UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

PORTHOS RIBEIRO DE ALBUQUERQUE MOTTA

Estudo Exploratório do Uso de Classificadores para a Predição de Desempenho e Abandono em Universidades

Goiânia
2016

sistema de bibliotecas ufg

UFG

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR AS TESES E DISSERTAÇÕES ELETRÔNICAS NA BIBLIOTECA DIGITAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio da Biblioteca Digital de Teses e Dissertações (BDTD/UFG), regulamentada pela Resolução CEPEC nº 832/2007, sem ressarcimento dos direi'abaixo, para fins de leitura, impressão e/ou *download*, a título de divulgação da produção científica brasileira, a partir desta data.

1. Identificação do material bibliográfico: **[x] Dissertação** **[ ] Tese**

2. Identificação da Tese ou Dissertação

Nome completo do autor: Porthos Ribeiro de Albuquerque Motta

Título do trabalho: Estudo Exploratório do Uso de Classificadores para a Predição de Desempenho e Abandono em Universidades.

3. Informações de acesso ao documento:

Concorda com a liberação total do documento [x] SIM [ ] NÃO[1]

Havendo concordância com a disponibilização eletrônica, torna-se imprescindível o envio do(s) arquivo(s) em formato digital PDF da tese ou dissertação.

Porthos Ribeiro de Albuquerque Motta

Assinatura do (a) autor (a)

Data: 23 / 11 / 2016

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. A extensão deste prazo suscita justificativa junto à coordenação do curso. Os dados do documento não serão disponibilizados durante o período de embargo.

PORTHOS RIBEIRO DE ALBUQUERQUE MOTTA

Estudo Exploratório do Uso de Classificadores para a Predição de Desempenho e Abandono em Universidades

Dissertação apresentada ao Programa de Pós–Graduação do Instituto de Informática da Universidade Federal de Goiás, como requisito parcial para obtenção do título de Mestre em Mestrado em Ciência da Computação.

Área de concentração: Informática e Educação.

Orientadora: Profa. Dra. Ana Paula Laboissière Ambrósio -INF/UFG

Goiânia
2016

Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema de Bibliotecas da UFG.

Ribeiro de Albquerque Motta, Porthos
Estudo Exploratório do Uso de Classificadores para a Predição de Desempenho e Abandono em Universidades [manuscrito] / Porthos Ribeiro de Albquerque Motta. - 2016.
CLVI, 156 f.

Orientador: Profa. Dra. Ana Paula Laboissière Ambrósio; co orientador Dr. Eduardo Simões de Albuquerque.
Dissertação (Mestrado) - Universidade Federal de Goiás, Instituto de Informática (INF), Programa de Pós-Graduação em Ciência da Computação, Goiânia, 2016.
Bibliografia. Apêndice.
Inclui siglas, abreviaturas, gráfico, tabelas, algoritmos, lista de figuras, lista de tabelas.

1. Mineração de Dados Educacionais. 2. Classificação. 3. Predição de desempenho e abandono. I. Laboissière Ambrósio, Ana Paula, orient. II. Título.

CDU 004

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

UFG

ATA Nº 20/2016

ATA DA SESSÃO DE JULGAMENTO DA DISSERTAÇÃO DE MESTRADO DE PORTHOS RIBEIRO DE ALBUQUERQUE MOTTA

Aos vinte dias do mês de outubro de dois mil e dezesseis, às dez horas, na sala 150 do Instituto de Informática da Universidade Federal de Goiás, Campus Samambaia, reuniu-se a banca examinadora designada na forma regimental pela Coordenação do Curso para julgar a dissertação de mestrado intitulada **"Estudo Exploratório do Uso de Classificadores para a Predição de Desempenho e Abandono em Universidades"**, apresentada pelo aluno Porthos Ribeiro de Albuquerque Motta como parte dos requisitos necessários à obtenção do grau de Mestre em Ciência da Computação, área de concentração Ciência da Computação. A banca examinadora foi presidida pela orientadora do trabalho de dissertação, Professora Doutora Ana Paula Laboissière Ambrósio (INF/UFG), tendo como membros o Professor Doutor Anderson da Silva Soares (INF/UFG) e o Professor Doutor Leandro da Silva Almeida (IE/Universidade do Minho). O professor Leandro da Silva Almeida participou da sessão por videoconferência. Aberta a sessão, o candidato expôs seu trabalho. Em seguida, o aluno foi arguido pelos membros da banca e:

(X) tendo demonstrado suficiência de conhecimento e capacidade de sistematização do tema de sua dissertação, a banca concluiu pela **aprovação** do candidato, sem restrições.

( ) tendo demonstrado suficiência de conhecimento e capacidade de sistematização do tema de sua dissertação, a banca concluiu pela **aprovação** do candidato, condicionado a satisfazer as exigências listadas na Folha de Modificação de Dissertação de Mestrado anexa à presente ata, no prazo máximo de 60 dias, a contar da presente data, ficando o professor orientador responsável por atestar o cumprimento dessas exigências.

( ) não tendo demonstrado suficiência de conhecimento e capacidade de sistematização do tema de sua dissertação, a banca concluiu pela **reprovação** do candidato.

Os trabalhos foram encerrados às treze horas. Nos termos do Regulamento Geral dos Cursos de Pós-Graduação desta Universidade, lavrou-se a presente ata que, lida e julgada conforme, segue assinada pelos membros da banca examinadora.

Profa. Dra. Ana Paula Laboissière Ambrósio Ana Paula L. Ambrosio

Prof. Dr. Anderson da Silva Soares Anderson S. Soares

Prof. Dr. Leandro da Silva Almeida Leandro Silva

Porthos Ribeiro de Albuquerque Motta

Graduou-se em Matemática pela Universidade Federal de Goiás - Goiânia (2004) e especializou-se em Controladoria e Finanças Corporativas pelas Faculdades Alves Faria - Goiânia (2006) e em Gestão Bancária com Ênfase em Finanças Corporativas - Escola Aberta do Brasil (2006) Vila Velha . Graduou-se em Informática pela Universidade Estadual de Goiás - Aparecida de Goiânia (2013) e atuou como professor de educação básica pela Secretaria Estadual de Educação de Goiás e Secretaria Municipal de Educação de Goiás. Trabalha como Analista de Tecnologia da Informação na Agência de Fomento de Goiás S.A. - Goiânia, com levantamento de requisitos, análise de sistemas, modelagem de banco de dados e de processos de negócios. No trabalho de pesquisa sendo desenvolvido, procura utilizar a mineração de dados para buscar informações que auxiliem na melhoria dos processos de negócios educacionais.

Dedico este trabalho ao meu falecido pai que apesar das dificuldades, sempre me orientou a seguir e a nunca desistir mesmo passando por adversidades.

Agradecimentos

Agradeço à Deus e à minha esposa, pela compreensão nos momentos de dificuldade e pelo incentivo de finalizar mais uma etapa acadêmica em minha vida. Ao meu pai, que me ensinou a ter responsabilidade, respeito e consideração pelas pessoas e pela família, e que infelizmente faleceu antes da conclusão deste trabalho. À Profa. Ana Paula Laboissière Ambrósio, pelos momentos de orientação e pelo apoio, e que me auxiliou de forma decisiva a repensar a forma de aprender e de estudar, incentivando e repassando a experiência do pensamento crítico, analítico e sistemático. Aos professores Dr. Eduardo Simões de Albuquerque e Dr. Anderson da Silva Soares, pelo tempo dedicado auxiliando em questões ímpares que possibilitaram o avanço do trabalho, e pelas aulas onde sempre tinham novidades, incentivos e questões que promoviam a reflexão, bem como a busca constante por novos conhecimentos.

"Quem procura ter sabedoria ama a vida, e quem age com inteligência encontra a felicidade"

Salomão,
Provérbios 19:8.

Resumo

de Albuquerque Motta, Porthos Ribeiro. **Estudo Exploratório do Uso de Classificadores para a Predição de Desempenho e Abandono em Universidades**. Goiânia, 2016. 154p. Dissertação de Mestrado. Instituto de Informática, Universidade Federal de Goiás.

A Mineração de Dados Educacionais, por meio da tríade melhoria da qualidade, redução do custo e eficácia do ensino, age e procura compreender melhor o processo de ensino-aprendizagem dos alunos. Neste contexto, o objetivo desta dissertação é o estudo exploratório de métodos de classificação para predizer o desempenho e o abandono de alunos a partir de dados existentes nas bases de dados acadêmicas das universidades. Neste trabalho foram usados dados demográficos, sócio-econômicos e resultados acadêmicos, oriundos do Vestibular e do banco de dados acadêmico da universidade para analisar diversas técnicas de classificação, assim como técnicas de balanceamento e seleção de atributos identificadas através de uma revisão sistemática da literatura. Seguindo uma tendência verificada nos artigos levantados, optou-se por utilizar como principal algoritmo de classificação o J48, apesar de estudos comparativos terem mostrado melhores resultados com técnicas de regressão logística e redes Bayesianas. Isto se deve ao fato das árvores de decisão permitirem uma análise dos atributos usados nos modelos gerados, mantendo níveis de acurácia aceitáveis, enquanto as outras técnicas funcionam como uma caixa preta. Neste sentido, a técnica de Resample, que escolhe um subconjunto balanceado dos dados, apresentou melhores resultados que a técnica de SMOTE, que gera dados sintéticos para balancear os dados. Quanto ao uso de técnicas de seleção de atributos, estas não trouxeram vantagens significativas. Dentre os atributos usados, notas e aspectos econômicos aparecem com frequência nos modelos gerados. Uma tentativa de prever desempenho por disciplina, com base em dados de disciplinas já cursadas em semestres anteriores foi menos bem sucedida, talvez pelo fato de usar classes preditoras ternárias. Apesar disto, as análises realizadas mostraram que o uso de classificadores é um caminho promissor para a predição de desempenho e abandono, mas estudos mais aprofundados ainda são necessários.

Palavras–chave

Mineração de Dados Educacionais, Classificação, Predição de desempenho e abandono

Abstract

de Albuquerque Motta, Porthos Ribeiro. **1**. Goiânia, 2016. 154p. MSc. Dissertation. Instituto de Informática, Universidade Federal de Goiás.

Educational Data Mining, by the triad of quality improvement, cost reduction and educational effectiveness, acts and seeks to better understand the teaching and learning process. In this context, the aim of this work is an exploratory study of classification methods to predict student performance and dropout from data in university academic databases. In this study we used demographic, socio-economic and academic results, obtained from the Vestibular and the university database to analyze several classification techniques, as well as balancing and attribute selection techniques, identified through a systematic review of the literature. Following a trend found in the selected articles, we chose to use decision trees as the primary classification algorithm, although comparative studies showed better results with logistic regression techniques and Bayesian networks. This is because decision trees allow an analysis of the attributes used in the generated models while maintaining acceptable levels of accuracy, while other techniques work as a black box. Through the tests we found that you get better results using balanced sets. In this sense, the Resample technique that selects a balanced subset of the data showed better results than SMOTE technique that generates synthetic data for balancing the dataset. Regarding the use of attribute selection techniques, these did not bring significant advantages. Among the attributes used, grades and economic factors often appear as nodes in the generated models. An attempt to predict performance for each subject based on data from previous courses was less successful, maybe because of the use of ternary predictive classes. Nevertheless, the analysis carried out showed that the use of classifiers is a promising way to predict performance and dropout, but further studies are still needed.

Keywords

Educational Data Minining, Classification, Outcome prediction

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

Listas de Abreviaturas e Siglas

AD AdaBoost

ACM Association for Computing Machinery

AI Artificial Intelligence

API Application Programming Interface

APV Aprovado

AUC Area Under the ROC Curve

CART Classification And Regression Trees

CBF Consistency-Base Filter

CC Ciências da Computação

CERCOMP Centro de Recursos Computacionais

CFS Correlation based Feature Selection

CM Confusion Matrix

CRISP-DM Cross Industry Standard Process For Data Mining

CT Classification Trees

DBLP Digital Bibliography & Library Project

DM DataMining

ED1 Estrutura de Dados 1

ED2 Estrutura de Dados 2

EDM Educational DataMining

ENEM Exame Nacional do Ensino Médio

ERIC Information Resources Information Center

FN False Negative

FNR False Negative Rate

FP False Positive

FPR False Positive Rate

IEEE Institute of Electrical and Electronics Engineers

INEP Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira

KDD Knowledge Discovery in Database

KNN K-Nearest Neighbor

LDA Linear Discriminante Analysis

LM Lógica Matemática

LR Logistic Regression

MCC Mattews Correlation Coefficient

MD Mineração de Dados

MDE Mineração de Dados Educacionais

ML Machine Learning

MLE Maximum likelihood estimation

MLP Multilayer Perceptron

NB Naive Bayes

OLAP On-line Analytical Processing

PC1 Programação de Computadores 1

PC2 Programação de Computadores 2

PCA Principal Componentes Analysis

PDF Probability Density Functions

PR Pattern Recognition

PRODIRH Pró-Reitoria de Desenvolvimento Institucional e recursos Humanos

POO Programação Orientada a Objeto

PSO Particle Swarm Optimization

QDA Quadratic Discriminante Analysis

REF Reprovado por Falta

REP Reprovado

REST Represental State Transfer

RF Random Forest

ROC Receiver operating Characteristic

RSL Revisão Sistemática da Literatura

SISU Sistema de Seleção Unificada

SMOTE Synthetic Minority Over-sampling Technique

SPAM Sending and Posting Advertisement in Mass

SQL Structured Query Language

SVM Suport Vector MachineSuport Vector Machine

TC Teoria da Computação

TN True Negative

TNR True Negative Rate

TP True Positive

TPR True Positive Rate

UCI UC Irvine Machine Learning Repositor

UFG Universidade Federal de Goiás

WEE Weka Experiment Environment

WEKA Waikato Environment for Knowledge Analysis

CAPÍTULO 1

Introdução

Com o intuito de conhecer a realidade do sistema educacional brasileiro, com vistas a elaboração de políticas públicas na área de educação, o Ministério da Educação, através do *Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira* INEP coleta dados e informações sobre diversos aspectos nos níveis da educação no Brasil. Pela sua magnitude e abrangência, as bases consideradas de maior importância são o Censo Escolar e o Censo do Ensino Superior. Essas bases têm uma atualização anual e dispõem de informações sobre as instituições escolares nos diversos níveis de ensino, como o número de matrículas, o volume de alunos, o movimento escolar, características básicas da instituição, equipamentos e edificações existentes, além de dados sobre o pessoal técnico e administrativo e as características dos docentes, entre outros.

As diversas bases de dados do INEP podem ser acessadas via Internet, no site www.inep.gov.br [84], onde se encontram também diversas informações e relatórios que podem ser consultados ou copiados (via download). Tal sistema fornece um amplo espectro de informações sobre o sistema educacional, com uma grande variedade de possibilidades de consulta de dados e indicadores educacionais, inclusive com séries históricas.

Além destas informações globais coletadas pelo INEP, as diferentes instituições educacionais também possuem bases de dados acadêmicas que armazenam a trajetória dos alunos em seus estabelecimentos, por meio de dados sobre os cursos e disciplinas cursadas, notas, frequências, entre outras. Não é comum o acesso externo a estes dados, porém a parte acessível acaba sendo disponibilizada através das bases de dados do INEP ou através de estatísticas cujo objetivo é expor o trabalho feito dentro das instituições.

A *Universidade Federal de Goiás* (UFG) disponibiliza em seu site *(www.prodirh.ufg.br)*[148], de forma sistematizada, informações estatísticas de suas atividades-fim:– Ensino, Pesquisa e Extensão, a fim de "democratizar o acesso aos dados do trabalho desenvolvido na UFG e prover uma fonte de busca e análises aos pesquisadores, estudantes e à sociedade em geral", com informações sobre a infraestrutura, sobre o quantitativo dos acadêmicos e dos servidores, e ainda sobre as questões financeiras da Instituição. Quando solicitado, os microdados referentes a unidades acadêmicas indivi-

duais podem ser obtidas por meio de downloads através de consultas aos bancos de dados acadêmicos da Instituição.

A estas informações acadêmicas mais formais, pode-se somar aquelas obtidas nos cursos. Com a adoção cada vez mais frequente de ambientes virtuais de aprendizagem e de gerenciamento de disciplinas como Moodle e Blackboard, muita informação acaba sendo armazenada no ambiente, incluindo documentos submetidos, histórico de acesso, etc. Isto fica ainda mais evidente quando o curso ou disciplina é ofertado na modalidade a distância. Nestes casos, praticamente toda iteração acaba sendo documentada localmente. Infelizmente, apesar de interessante, falta uma estrutura mais formal de coleta dessas informações, e assim, elas acabam sendo perdidas ao longo do tempo. De fato, somente parte desta grande quantidade de informações coletada acaba sendo vista pelos professores e muitas vezes são usadas apenas para compor as notas finais dos alunos.

A análise das informações acadêmicas oferece uma visão global do sistema educacional e serve como base para a tomada de decisões. No entanto, ela geralmente fica restrita a resumos que geram indicadores estatísticos que expressam aspectos particulares do sistema, como por exemplo o rendimento escolar, a distorção entre a idade e a série, bem como a progressão dos estudantes no decorrer dos anos. Além disto, apesar dos avanços tecnológicos, o tratamento das informações ainda é um processo complexo e demorado, atrasando o acesso aos relatórios estatísticos.

O aumento na quantidade de dados e informações acadêmicas disponíveis e os avanços nas áreas relacionadas ao tratamento de informações, levaram à criação de uma nova área de pesquisa denominada mineração de dados educacionais, ou EDM (do inglês, *Educational DataMining*), que trata da aplicação de técnicas de mineração de dados ou DM (do inglês, *DataMining*), aprendizado de máquina e estatística à informações geradas em ambientes educacionais. Trata-se de uma área recente de pesquisa que tem apresentado um rápido crescimento, com grande potencial para melhorar a qualidade de ensino. Apesar disto, no Brasil esta área de pesquisa ainda é pouco explorada [33].

A mineração de dados tem como objetivo descobrir “novas” informações através da análise de grandes quantidades de dados [111]. O termo “novas informações” refere-se ao processo de identificar relações entre dados que podem produzir novos conhecimentos e gerar novas descobertas científicas. Esta área de pesquisa é também conhecida como descoberta de conhecimentos em bancos de dados, ou KDD (do inglês, *Knowledge Discovery in Databases*) [48].

Existem várias linhas de pesquisa na área de EDM e muitas delas derivadas diretamente da área de mineração de dados, sendo que a taxonomia das principais subáreas de pesquisa em EDM são expostas a seguir [157]:

(1) Predição (*Prediction*)

Classificação (*Classification*)

Regressão (*Regression*)

Estimativa de Densidade (*Density Estimation*)

(2) Agrupamento (*Clustering*)

(3) Mineração de Relações (*Relationship Mining*)

Mineração de Regras de Associação (*Association Rule Mining*)

Mineração de Correlações (*Correlation Mining*)

Mineração de Padrões Sequenciais (*Sequential Pattern Mining*)

Mineração de Causas (*Causal Mining*)

(4) Destilação de Dados para Facilitar Decisões Humanas (*Distillation of Data for Human Judgment*)

(5) Descoberta com Modelos (*Discovery with Models*)

O estudo apresentado nesta dissertação ficou restrito ao estudo da predição, com foco na classificação baseado na linhas de pesquisas apresentadas por Baker *et al.*[157].

Aprendizagem de Máquina (do inglês, *Machine Learning*) é um subdomínio da Inteligência Artificial (do inglês, *Artificial Intelligence*) , que se baseia na natureza para representá-la por meio da criação de modelos artificiais, e tem como principal foco o desenvolvimento de técnicas e algoritmos para que um equipamento eletrônico (*i.e.* notebook, computador de mesa por exemplo), possa aprender um determinado comportamento ou padrão de forma multifacetada, a partir da experiência adquirida por meio de testes e treinamentos. Dentro deste contexto temos a área de Reconhecimento de Padrões, que junto com a Estatística, visa efetuar a análise e predição dos dados, e estuda a classificação e a descrição de objetos a partir de suas características, por meio da utilização do reconhecimento estatístico ou do reconhecimento linguístico.

Estas ferramentas podem ser aplicadas na chamada análise de aprendizagem (do inglês, *Learning Analytics*), que foca tanto na seleção, captura e processamento de dados educacionais que serão úteis para alunos e educadores, quanto no desenvolvimento de sistemas que possuem a capacidade de capturar dados e elaborar relatórios a partir de uma base de dados que cresce de forma contínua, minimizando o tempo de obtenção e utilização dos dados. Ela também visa a modelagem e análise a fim de prever comportamentos, agindo sobre previsões, com a capacidade de retroalimentação.

Dentro das opções oferecidas pelas ferramentas associadas à Mineração de Dados, temos aquelas que possuem a capacidade de efetuar previsões com base nos dados. Esta capacidade é particularmente interessante no contexto educacional por permitir que ações preventivas possam ser tomadas quando situações negativas forem identificadas. Por exemplo, com a predição da reprovação em determinada disciplina ou abandono do curso, ações poderão ser mapeadas, criadas e aplicadas para auxiliar alunos, professores e instituição de ensino.

Uma revisão sistemática da literatura mostrou um crescente interesse no uso de mineração de dados para a predição de desempenho acadêmico. No entanto esta área ainda está em estágio inicial, sendo grande parte dos trabalhos composta de estudos exploratórios, onde algoritmos de classificação são aplicados a dados disponíveis para verificar sua adequação para a predição de desempenho. Destes estudos têm surgido algumas tendências, que ainda não são suficientes para uma sistematização do processo. O uso de árvores de decisão como principal algoritmo de classificação tem se mostrado recorrente pela sua transparência no modelo gerado, o que facilita a compreensão do processo e a elaboração de medidas preventivas, sendo que alguns trabalhos fazem comparações entre classificadores. No entanto, estas comparações limitam-se a comparar resultados sem considerarem o test-t e o AUC. Além disto, os trabalhos não levam em consideração técnicas acessórias como o balanceamento dos dados e a seleção de atributos. Verificou-se também, que a análise de resultados foi feita basicamente na comparação do total de instâncias classificadas corretamente, sem levar em consideração a matriz de confusão, importante para a identificação dos casos mais críticos de alunos com mau desempenho.

Assim, neste trabalho optou-se por explorar alguns destes aspectos com o intuito de contribuir para o entendimento do processo de predição de desempenho, a fim de identificar melhores práticas que podem no futuro contribuir na sistematização do processo. Foram comprados vários tipos de classificadores para avaliar os resultados das classificações com o auxílio de medidas de desempenho como o AUC (Área sobre a Curva ROC), a precisão, entre outras, e aplicando o teste de correção de emparelhados (test-t) para a verificação da existência de resultados estatisticamente melhores ou piores ao nível de significância de 5% ou com confiança de 95%, para cada valor obtido pelas medidas de desempenho, considerando como classificador base o J48 (caixa branca) e com base de dados desbalanceadas, balanceadas por meio de algoritmos como o SMOTE e o Resample, e também filtradas a partir da aplicação do filtro de correlação CFS. Os procedimentos foram feitos considerando a classificação por abandono e por situação geral (desempenho), utilizando dados do ensino presencial dos alunos de ciências da computação.

Tendo em vista a comparação, utilizando vários tipos de classificadores em relação ao J48, em vários casos foi possível perceber que mesmo com classificadores estatisticamente melhores, a sua utilização não ficou impedida, e assim foi utilizado para a geração de árvores de decisão e para a extração de regras para o abandono e desempenho. Por meio das regras foi possível verificar quais características foram essenciais para a composição da classificação e quais não foram consideradas importantes. Diferente dos projetos da RSL, a análise do desempenho levou em consideração a matriz de confusão, para identificar os resultados que melhor classificam os alunos com mau desempenho, e com o menor índice de falso positivo, isto é, aqueles alunos que não são classificados

como tal, mas acabam abandonando ou reprovando. O problema com estes casos, é que eles acabem não sendo beneficiados pelas ações de prevenção desenvolvidas. Além da análise estática do desempenho, testou-se uma análise temporal do desempenho do aluno, onde este desempenho foi feito por semestre, levando em consideração apenas as notas das disciplinas já cursadas pelos alunos.

1.1 Objetivos

1.1.1 Objetivo Geral

Estudo exploratório de métodos de classificação desenvolvidos para a mineração de dados educacionais para predizer o desempenho e o abandono acadêmico.

1.1.2 Objetivos Específicos

(1) Estudo dos mecanismos de classificação que podem ser utilizados em mineração de dados educacionais;
(2) Preparação e transformação dos dados acadêmicos usados nos estudos de caso por meio de técnicas de mineração de dados;
(3) Identificação de um conjunto de predições educacionais que podem ser feitas usando estes mecanismos dado o conjunto de dados disponíveis na base acadêmica da UFG;
(4) Identificação de variáveis preditoras adequadas para a predições do desempenho e abandono acadêmico;
(5) Utilização de métricas adequadas para avaliação dos resultados;
(6) Definição de um ambiente computacional a ser implementado para a análise dos dados, através da identificação das ferramentas disponíveis para a análise de dados educacionais.
(7) Conversão de um problema real em tarefas de Mineração de Dados;

Esta dissertação encontra-se no campo da Mineração de Dados e suas aplicações em bases de dados Educacionais, sendo assim de forma mais específica de Mineração de Dados Educacionais na hipótese de que tais técnicas podem ser agregadas a um modelo multidimensional que leve à descoberta de fenômenos e ao entendimento dos dados relativos ao abandono e desempenho dos alunos e seus impactos, com base nas características extraídas da base de dados educacional. Foram utilizadas as abordagens de modelagem descritiva e preditiva, a fim de descobrir informações ocultas relativas aos aluno do curso de ciências da computação (CC) da Universidade Federal de Goiás (UFG).

Para as análises preditivas, foram utilizados 10 classificadores para abandono e 11 para o desempenho. O classificador J48 foi utilizado para exibição gráfica do processo de classificação e para a extração das regras utilizadas. Boas perguntas foram definidas:

- O Aluno abandona o curso?
- O Aluno terá um bom desempenho no final do curso?
- O Aluno terá um bom desempenho nas disciplinas cursadas e cada semestre letivo?

Para as perguntas definidas foram geradas estatísticas descritivas para avaliação das respostas das perguntas definidas. A boa notícia foi a possibilidade de responder à pergunta sobre o abandono e sobre o desempenho a partir da comparação entre os algoritmos de classificação de forma automática onde o classificador Floresta Aleatória retornou um excelente valor para a precisão e o melhor valor para a área sobre a curva ROC (AUC).

Para o abandono por disciplina considerando cada semestre, foi possível classificar a partir do J48 apenas as disciplinas de programação de computadores I, Lógica matemática, estrutura de dados I, Programação de computadores II, e Programação Orientada a objetos.

Dessa forma nesta dissertação apresentamos todo o procedimento para chegar aos resultados demonstrados. Acreditamos ser de interesse da comunidade educacional o conhecimento de um assunto de grande importância para a universidade. Os resultados são promissores, com modelos de classificação construídos que apresentam bons resultados de precisão e AUC.

1.2 Organização da Dissertação

O Capítulo 2 apresenta a área de Mineração de Dados e Mineração de Dados Educacionais. Os principais algoritmos de classificação, assim como as medidas de desempenho, são apresentados no Capítulo 3. Neste capítulo também é feita uma análise comparativa destas técnicas de classificação usando um arquivo do repositório UCI. Os trabalhos relacionados, analisados através de uma Revisão Sistemática da Literatura, são apresentados no Capítulo 4. No Capítulo 5 são apresentados os dados dos alunos de Ciências da Computação da UFG e o tratamento que foi realizado para adequá-los aos requisitos de entrada dos diferentes mecanismos de classificação. Também são definidos os atributos de classe, e como estes foram calculados. No Capítulo 6 é traçado um perfil dos alunos de Ciência da Computação (CC) da UFG usando estatística descritiva. Os resultados obtidos da aplicação dos algoritmos de classificação aos dados do curso de CC são apresentados no Capítulo 7. Por fim, no Capítulo 8 são apresentadas as considerações finais sobre este trabalho e trabalhos futuros.

CAPÍTULO 2

Mineração de Dados

Inteligência Artificial (do inglês, *Artificial Intelligence* - AI) é o estudo dos sistemas que agem de um modo que a um observador qualquer pareceria ser inteligente [28]. Esta definição não abrange o todo da AI, dessa forma uma definição mais adequada seria:

- A AI envolve utilizar métodos baseados no comportamento inteligente de humanos e outros animais para solucionar problemas complexos [28].
- Pensando como humano ela pode ser definida como "o novo e interessante esforço para fazer os computadores pensarem (...) máquinas com mentes, no sentido total e literal"[74], ou pensando de forma racional como o "o estudo das computações que tornam possível perceber, raciocinar e agir"[189].
- Agindo como humano pode ser definida como "O estudo de como computadores podem fazer tarefas que hoje são melhor desempenhadas pelas pessoas"' [46], ou agindo racionalmente "como estudo de projetos inteligentes"' [146].

O Reconhecimento de padrões (do inglês, *Pattern Recognition* - PR) trata da classificação e descrição de objetos em um determinado número de categorias ou classes a partir da observação de suas características [181]. Tem por objetivo representar de uma forma mais simples um determinado conjunto de dados a partir da relevância de suas características, proporcionando dessa forma a possibilidade de partição em classes.

Pode ser considerado como uma abordagem para o aprendizado de máquina, baseando-se para este fim na modelagem estatística dos dados, e dessa forma a partir de um modelo probabilístico e da teoria de decisão, obtém-se um algoritmo. Envolve problemas de extração de características de um determinado repositório de dados dos objetos que serão classificados, selecionando as mais relevantes, ou seja, aquelas que são mais discriminativas para a construção de um classificador ou preditor.

Aprendizagem de máquina que é um subcampo da AI, e tem por objetivo ou função desenvolver algoritmos e técnicas para que um computador possa aprender a executar determinado tipo de tarefa, melhorando gradativamente seu desempenho por meio do treinamento, e a partir da experiência [128].

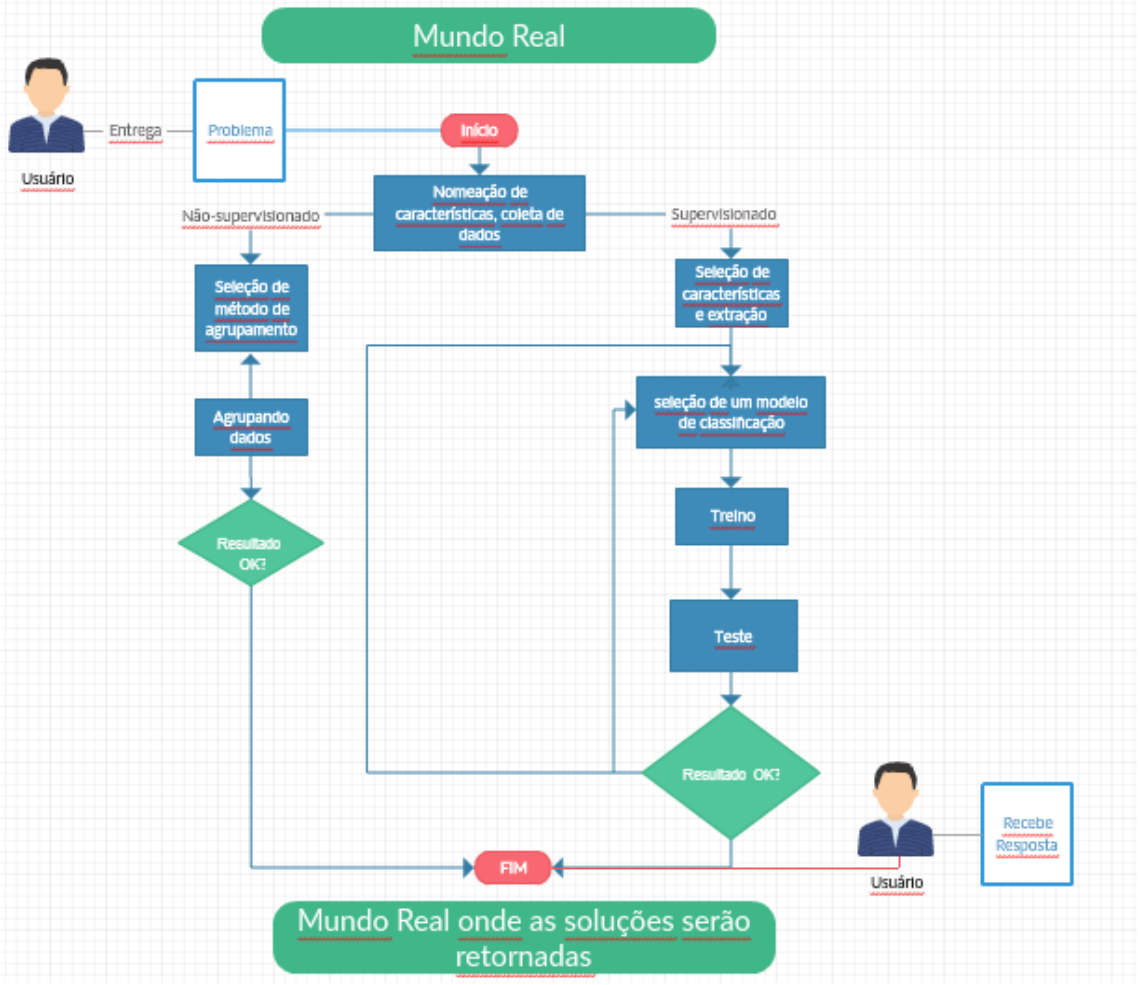

Figura 2.1: *Fases da aprendizagem de máquina adaptado de [108]*

A mineração de dados (MD) é uma etapa no processo de descoberta de conhecimento ou *Knowledge Discovery from Data* (KDD) [68], mas, pelo fato do termo ser utilizado com frequência ficou mais popular que o KDD. Seguindo esta tendência, neste trabalho será utilizado o termo mineração de dados para se referir ao processo ou método de extração ou mineração de conhecimento interessante ou de padrões a partir de um grande conjunto de dados [68]. É o processo de descoberta, de forma automática, de informações úteis em grandes repositórios de dados [176].

Assim, a MD pode ser vista como o resultado natural da evolução da tecnologia da informação ilustradas na Figura 2.2, e resulta da integração entre várias áreas do conhecimento que estão dispostas na Figura 2.3, usando técnicas de várias disciplinas como: banco de dados, estatística, aprendizado de máquina, computação de alta performance, computação paralela e distribuída, reconhecimento de padrões, redes neurais, visualização de dados, retorno de informação, imagem, processamento de sinais e análise de dados espaciais.

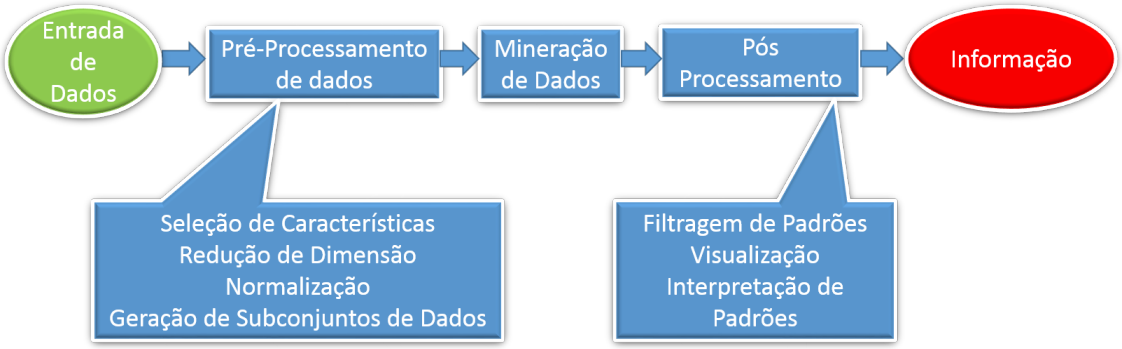

Figura 2.2: *Processo de Descoberta de conhecimento em Banco de Dados (KDD). Adaptado de Tan et al. [176]*

O uso das técnicas de mineração de dados dentro do contexto educacional ficou conhecida recentemente como Mineração de Dados Educacionais (do inglês, *Educational Data Mining* - EDM), pois como a própria MD, integra várias disciplinas e está relacionada ao desenvolvimento de métodos da MD para explorar dados educacionais, que podem ser utilizados para apoiar o processo de entendimento dos estudantes usando estatística, algoritmos provenientes da inteligência artificial, reconhecimento de padrões, aprendizado de máquina, entre outros [158].

Por meio da tríade melhoria da qualidade, redução do custo e eficácia do ensino, EDM age e procura compreender melhor o processo de ensino-aprendizagem dos alunos. De acordo com Romero *et al.* [160] existem três principais objetivos da EDM:

1 **Pedagógico**: Com o intuito de auxiliar na concepção de conteúdos didáticos e na melhoria da performance acadêmica dos alunos;
2 **Gerencial**: A fim de otimizar a instituição de ensino, e manutenção de infra-estruturas de ensino, as diversas áreas de interesse e de estudo.
3 **Comercial**: Para auxiliar estudantes e nas fases de recrutamento, especialmente em instituições de ensino privado.

EDM pode ser vista como um ciclo interativo de formação de hipóteses, testes e refinamento de acordo com a Figura 2.4.

O foco da EDM está nas informações oriundas de sistemas educacionais, sejam eles na modalidade à distância ou presencial, e utiliza técnicas de DM para mineração destes dados com o objetivo de descoberta de conhecimento. Ainda pela Figura 2.4, usando EDM o papel do educador neste contexto é de planejar, construir e manter sistemas de ensino. Já os alunos serão os atores que terão o papel de interação e de utilização desses sistemas.

EDM vista com foco interdisciplinar se mistura com a pedagogia, onde a primeira fornece meios de analisar e modelar os dados, e a segunda contribui com o conhecimento dos processos de aprendizagem, além de requisitos específicos que não são

Figura 2.3: *Data Mining como uma confluência de várias disciplinas. Adaptado de Tan et. al [176]*

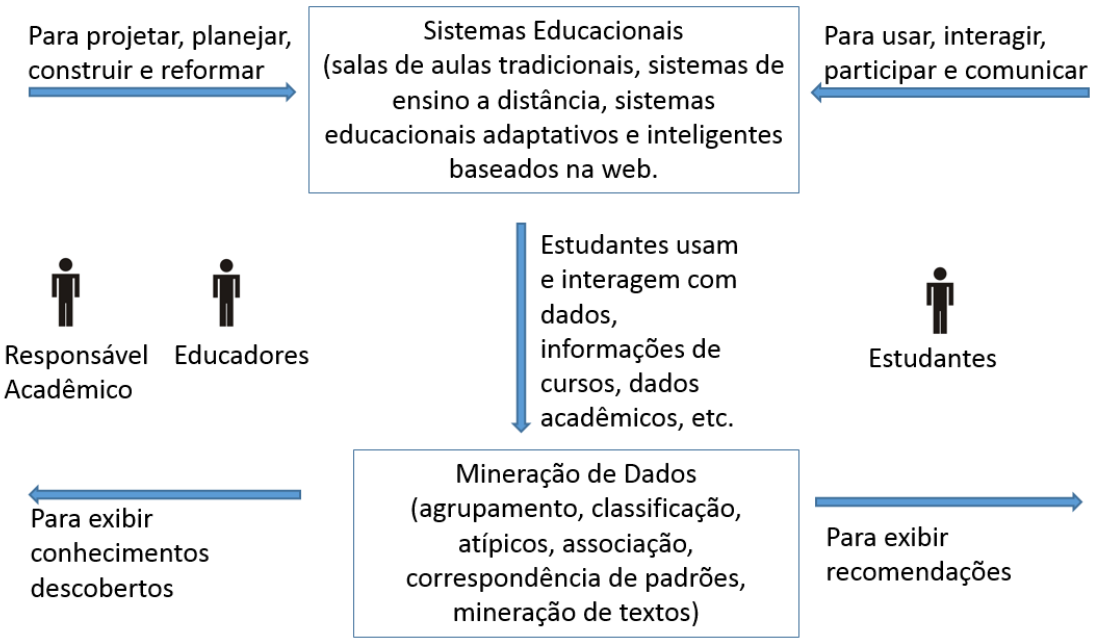

Figura 2.4: *Definição do ciclo de Mineração de Dados em Sistemas Educacionais, adaptado de Romero et. al. [160]*

encontrados em outros domínios, como os aspectos pedagógicos de educadores, educandos e do sistema.

Descobrir conhecimento a partir de dados possui um certo grau de dificuldade, não sendo uma tarefa trivial. É preciso conhecer o domínio a ser explorado, o que permite delimitar o problema, identificar dados relevantes e analisar o resultado dentro do contexto.

Por se tratar de um processo de busca exaustiva, a MD possui alguns desafios na metodologia e interação com o usuário, como, por exemplo a análise de diferentes tipos de dados que devem ser adequados e interpretados, e o uso de várias técnicas de extração de conhecimento, como caracterização, associação, agrupamento (*clustering*), tendência, análise de desvio e análise de similaridade. A mineração interativa em múltiplas bases de dados por meio de operações de On-line Analytical Processing - OLAP em cubos de dados, permite que os usuários se concentrem na busca de padrões, exibindo os dados de forma interativa.

Para o fim de extração e descoberta de conhecimento, e para converter dados em informações úteis, é necessário passar por diversas etapas, que são formalizadas pelo processo KDD [68] [48], que pode ser definido como o processo de extração de conhecimento a partir dos dados, onde a mineração de dados é apenas uma etapa particular do KDD. Já a metodologia CRISP-DM (*Cross Industry Standard Process For Data Mining, Processo Padrão Inter-Indústrias para Mineração de Dados*) [26], desenvolvido pelo consórcio formado por NCR *Systems Engineering Copenhagen*, *DaimlerChrysler AG*, *SPSS Inc.* e *OHRA Verzekeringen en Bank Groep B.V* em 1996 [26], é composta por fases e processos padrões que são utilizados para o desenvolvimento de projetos de Mineração de Dados, sem a preocupação de qual ferramenta computacional será utilizada. Seus principais objetivos são:

(1) Propor a transformação dos dados por meio das técnicas de mineração de dados;
(2) Utilizar métricas adequadas para a avaliação dos resultados;
(3) Converter problemas reais em tarefas de Mineração de Dados.

As fases do CRISP-DM [26] em resumo são: Entender o negócio; Entender os dados; Preparar os dados; Modelar os dados; Avaliar os dados; Desenvolver um modelo com a melhor performance.

A mineração de dados quando vista como um processo de descoberta de conhecimento envolve as seguintes etapas:

(1) *Limpeza de dados*: processo de remoção de dados inconsistentes ou incompletos;
(2) *Integração de dados*: Quando múltiplos bancos de dados são combinados;
(3) *Seleção de dados*: Quando dados interessantes são retornados do banco de dados para a tarefa de análise;

(4) *Transformação de dados*: Quando os dados são consolidados (por meio de agrupamento) e ou transformados (a partir de uma técnica de transformação) em uma forma apropriada para a mineração;
(5) *Mineração de dados*: um processo essencial, onde métodos eficientes e inteligentes são aplicados em determinada ordem para a extração de padrões;
(6) *Avaliação de padrões*: processo que visa identificar padrões verdadeiramente interessantes, representando uma base de conhecimento ou apenas medidas interessantes;
(7) *Representação de conhecimento*: Quando técnicas de visualização e representação de conhecimento são usadas para representar o conhecimento "minerado"pelo usuário.

A importância da mineração de dados se constitui pelo fato de ser possível descobrir conhecimento que não fica claro utilizando apenas a estatística ou consultas de dados. Por meio da estatística descritiva é possível obter resumos dos dados utilizando média, distribuição de frequências, localizando valores máximos e mínimos. Pelas consultas aos dados, utilizando a linguagem SQL (Structured Query Language) por exemplo, podemos efetuar um agrupamento utilizando a estatística descritiva, agrupamento pelo somatório de determinado atributo, aplicando em seguida a média por exemplo. Utilizando a mineração de dados é possível encontrar regras, geradas por meio de algoritmos, que permitem prever em que classe um sujeito será enquadrado dependendo do valor de seus atributos, o que não pode ser feito apenas por observação dos dados ou por meio de estatística ou consulta de dados.

Vale salientar que os dados se tornam relevantes a partir do momento que são de fato importantes para a descoberta de conhecimento e/ou para a geração de informação que de fato influencie no negócio ou nos processos de tomada de decisão. Já o conhecimento do contexto dos dados minerados contribui para entender como um dado poderá ser utilizado, seja para gerar rótulos de classes, métricas, ou para ser utilizado em determinado algoritmo.

2.1 Pré-Processamento

A etapa de pré-processamento é fundamental no processo de KDD, pois nela os dados são captados, selecionados, organizados, limpos, preparados e transformados. A captação é a forma de obtenção dos dados, seja ela por meio de conexão com algum repositório de dados, pelo acesso e download dos dados através de uma base de dados pública ou privada, ou outra forma qualquer. Após a captação é interessante selecionar apenas aqueles atributos que de fato serão utilizados para análise, seja pelos algoritmos de mineração de dados ou pelo estudo estatístico, e isso pode ser feito por meio de técnicas de seleção de atributos. As técnicas mais comuns são: Ganho de informação, Taxa

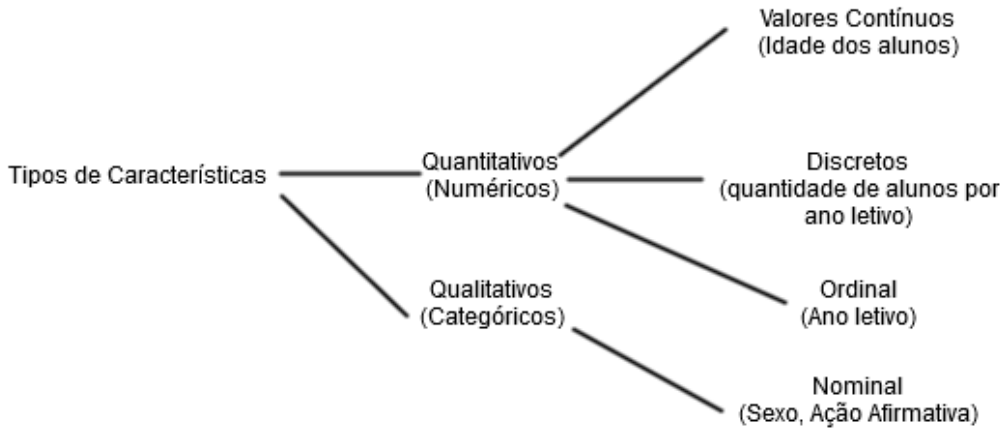

Figura 2.5: *Tipos de características [108]*

de Ganho e o índice de Gini, ChiSquare, Correlação, OneR, Componentes Principais. Existem algoritmos que utilizam estas técnicas já embutidas no próprio algoritmo, como é o caso das árvores de decisão [156].

Em sistemas de classificação, um dos atributos serve como *classificador*, identificando as classes nas quais o sujeito pode ser alocado pelos algoritmos. Este atributo é normalmente definido pelo objetivo da predição, e serve para guiar todo o processo de mineração. Os atributos selecionados para uso pelos algoritmos de MD serão aqueles relevantes para efetuar a predição desejada, por meio da construção de um modelo utilizando estes atributos.

Os atributos presentes nos banco de dados podem ser de diferentes tipos conforme ilustrado pela Figura 2.5. As variáveis discretas, que possuem um grande número de possibilidades, são considerados como quantitativas. Já as características categóricas possuem um pequeno número de possibilidades, ou pequenas variações.

Dentre os registros de um banco de dados, é comum ter atributos sem valores (do inglês, *missing values*), e estes devem ser tratados. Caso existam em pequena quantidade os registros onde eles aparecem podem ser omitidos. Contundo, para um conjunto de dados reduzido, uma pequena porção de valores inexistentes pode de fato afetar os resultados e também empobrecer o conjunto de treinamento [72]. Assim, antes de retirá-los, é preciso avaliar a importância de cada atributo, ou seja, e se ele não for crucial para a análise, então poderá ser retirado, caso contrário, a melhor alternativa é obter o valores que estão faltando.

De fato, um tempo significativo pode ser gasto no tratamento de valores dessa natureza, e nem sempre é interessante automatizar este processo. Os valores inexistentes costumam ser representados como nulo, *NaNs* ou sem valor apresentável. Para tratar estes casos, é preciso o julgamento humano na hora de decidir pela substituição ou exclusão dos dados. É preciso analisar caso a caso, antes da aplicação desses critérios. Existem vários

tipos de técnicas de substituição desses valores como: substituir um valor pela média arimética do conjunto de dados, valor máximo, mínimo, substituir com valores aleatórios, remover as linhas sem valores.

Não sendo possível obter os valores que faltam, deve-se optar por algoritmos de predição que não sejam sensíveis à dados com *missings*, e que ainda assim consigam retornar uma informação com qualidade e confiável. A maioria dos algoritmos e *Toolboxes* de mineração de dados conseguem lidar com valores inexistentes, com diferentes estratégias sendo adotadas pelos algoritmos de classificação.

Em outras situações, o analista precisa transformar valores contínuos em discretos, para a utilização em alguns classificadores, e isso pode ser feito por meio da transformação dos dados em conjunto de intervalo. Existem algumas técnicas para este fim como a *Entropy-MDL*, *Equal frequency*[134].

Em outras situações os dados obtidos para análise podem ter sido disponibilizados, ou até mesmo possuírem a natureza discreta. Assim caso o analista necessite utilizá-los de forma contínua, será preciso aplicar algum tipo de transformação nos dados para este fim. Algumas técnicas para a transformação de dados discretos em contínuos são: Substituição pelo valor mais frequente, utilizar um atributo por valor, remover atributos com múltiplos nomes, remover todos os atributos discretos, dividir cada atributo pelo números de valores.

A normalização de atributos também é de grande importância na área de mineração de dados. Ela se refere a uma técnica matemática que transforma os valores de cada atributo em um mesmo intervalo de variação a fim de minimizar a discrepância entre os valores de atributos distintos. Ela não pode ser confundida com a técnica de normalização de Banco de dados que possui as chamadas formas normais.

Uma normalização pode ser feita por meio das seguintes fórmulas [68]:

- normalização min-max:

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \tag{2-1}$$

- normalização z-score:

$$z_i = \frac{x_i - mean(x)}{std(x)} \tag{2-2}$$

- normalização por escala decimal

$$z_i = \frac{x}{(10)} \tag{2-3}$$

onde $x = (x_1, ..., x_n)$, e z_i é agora o seu i-ésimo dado normalizado.

2.2 Exploração de dados e Redução de Dimensão

Na mineração de dados existem situações onde encontramos um grande número de variáveis no banco de dados. Em algumas situações, subconjuntos de dados possuem alta correlação com outros dados. Em um modelo de classificação isso pode levar ao *overfitting* e acurácia insuficiente. Além disto, uma grande quantidade de variáveis podem ocasionar um problema de performance computacional. No desenvolvimento de um modelo, variáveis não representativas podem incrementar o custo na coleta e no processamento por parte dos algoritmos. Chegamos então na questão da dimensionalidade, que pode ser resumida como o número de variáveis independentes ou de entrada que serão utilizadas pelo modelo. Um dos pontos chaves da mineração de dados é o de sempre reduzir a dimensionalidade sem sacrificar a acurácia [169].

Um dos primeiros passos da análise de dados é a exploração, que pode ser efetuada por meio de resumo de dados e de gráficos. É um passo importante e que precisa ser considerado; para melhor compreensão dos dados, melhorando os resultados dos modelos e do processo de mineração de dados. Por exemplo, para verificar quanto os dados estão dispersos em relação à média pode-se utilizar o desvio padrão. Utilizando uma função para contar a quantidade de registros em branco é possível encontrar *missing values*. Caso não encontre valores faltantes, não será preciso tratá-los com as técnicas de remoção ou de substituição de valores.

Utilizando a visualização de dados é possível verificar se existe relação entre os dados e se estes possuem uma tendência parecida. Isto porém não é possível com um grande conjunto de dados. O que se recomenda nestes casos é utilizar um exemplo aleatório dos dados para a geração de visualização.

Por meio da análise de correlação é possível verificar o quanto as variáveis estão relacionadas. A matriz de correlação costuma ser utilizada para este fim. Ela também é útil para verificar a existência de redundância. A remoção de variáveis fortemente correlacionadas é importante para evitar problemas de multicolinearidade (presença de dois ou mais preditores que compartilham a mesma relação linear com a variável de resultado).

Se tratando de técnicas de redução de dimensão uma das mais utilizadas é a Análise de Componentes Principais (do inglês, *Principal Componentes Analysis* - PCA), que reduz o número de preditores do modelo analisando as variáveis de entrada, e é utilizada quando se possui variáveis quantitativas. Para variáveis categóricas (nominais ou ordinais) existem outros tipos de técnicas como a Análise de Correspondência.

2.3 Mineração de Dados

Após a captação, seleção e tratamento dos dados, é realizada a etapa de mineração dos dados. A Tabela 2.1 apresenta metodologias adotadas para o tratamento de diversos tipos de problemas de mineração de dados e para diversos tipos de dados.

Tabela 2.1: *Tabela que descreve Metodologias de Mineração de dados, destacando quais os tipos de dados cada metodologia utiliza, bem como o tipo de Problema de Mineração de Dados.*

Um guia para o uso típico de mineração de dados Metodologias para diversos problemas de mineração de dados e tipos de dados adaptado de [193]

	Tipo de Dado				*Problema de Mineração de Dados*		
Metodologia de Mineração de Dados	Dado Rotulado	Dado Não Rotulado	Registro de Dados Separados	Dados de Séries Temporais	Predição e Classificação	Descoberta de Padrões de Dados, Associações e Estrutura	Reconhecimento de Semelhanças e Diferenças nos Dados
Árvores de Decisão	x		x		x	x	x
Regras de Associação		x	x			x	x
Redes Neurais Artificiais	x	x	x	x	x		x
Analise Estatística de Dados Normais e Anormais		x	x		x		x
Análise Bayesiana dos Dados	x	x	x	x	x	x	x
Processos Escondidos de Markov e Mineração de Padrão Sequencial	x	x		x	x		x
Modelos de Predição e Classificação	x		x	x	x	x	x
Análise de Componentes Principais		x	x			x	x
Métodos Psicométricos de Modelagem Variável Latente	x	x	x		x	x	x
Agrupamento Escalonável		x	x			x	x
Similaridade e Indexação de Series Temporais	x	x		x	x		x
Análise de Séries Temporáis Não-Lineares	x	x		x	x	x	x

Dentre as metodologias mais usadas encontram-se as técnicas de Associação, Clusterização e Classificação. As técnicas de Associação visam a descoberta de regras de associação mostrando condições atributo-valor que ocorrem frequentemente em determinado conjunto de dados [68]. A verificação da existência de regras de associações entre registros de dados que estão ou devem estar relacionados de alguma forma a partir de expressões "se-então"são tarefas de associação. As regras de associação possuem um determinado grau de certeza, sendo composto por dois fatores: suporte e confiança. Como exemplo de algoritmos utilizados para a obtenção de regras de associação podemos citar o *Apriori* [8], *FP-growth* [70], *ECLAT* [196].

As técnicas de Clusterização analisam um objeto da classe sem utilizar um rótulo definido a priori (como acontece na classificação). Os objetos são agrupados de acordo com o princípio da maximização da similaridade intra-classe e minimizando a semelhança inter-classe. Cada conjunto formado pelo agrupamento pode ser visto como uma classe de objetos. A Clusterização (do inglês, Clustering) também pode facilitar a formação

de Taxonomia, isto é, a organização de informações em uma hierarquia de classes, em grupos que possuem um determinado conjunto de eventos semelhantes [68]. Além disso os objetos dentro de um cluster possuem alta similaridade em comparação com o outro, mas são muito dissimilares para várias abordagens para clusterização. Alguns métodos de clusterização são relacionados a seguir:

- Métodos de Particionamento: k-means [95]
- Métodos Hierárquicos: Clusterização Hierárquica (AGNES [173]). BIRCH [198] integra clusterização hierárquica com interatividade (baseado em distância) realocação.
- Métodos baseados em Densidade:DBSCAN [13] e OPTICS [12]
- Métodos baseados em Grid: STING [185]
- Métodos baseados em Modelos: COBWEB [50]
- Métodos baseados dados de Alta Dimensão: CLIQUE [7] e PROCLUS [6]

As técnicas de agrupamento também podem ser utilizadas para detecção de valores extremos (outliers). Por não serem supervisionadas, não se faz necessário efetuar qualquer hipótese sobre a distribuição dos dados (por exemplo, métodos baseados em densidade). Os valores extremos são pontos que não se enquadram em qualquer cluster.

Por ser o foco principal do trabalho, a parte que descreve a classificação e também os classificadores (baseados em redes neurais, os estatísticos, e de aprendizagem de máquina), será apresentada em capítulo separado.

2.4 Pós-Processamento

É a etapa onde efetua-se a avaliação dos modelos implementados e dos padrões interessantes e informações relevantes retornados pela análise dos dados na etapa de mineração de dados. e também a representação e visualização dos resultados oriundos do processo de mineração de dados [69].

Estes resultados podem ser expressos por meio de tabelas, gráficos gerados com os valores obtidos por medidas que fornecem informações relevantes sobre o modelo utilizado e também por árvores de decisão (considerando os algoritmos que possuem esta habilidade) e até mesmo por árvore filogenética.

CAPÍTULO 3

Técnicas de Classificação

Existem diversas técnicas de mineração de dados, dentre as principais, podemos citar, as de aprendizagem de máquina e as estatísticas. Algumas técnicas são supervisionadas e outras não-supervisionadas e realizam as mais diversas tarefas de classificação e predição, de análise de regras de associação, de padrões sequenciais, análise de clusters (agrupamentos) e análise de valores extremos (outliers) conforme exposto no Capitulo sobre Mineração de Dados. Utilizaremos neste estudo apenas técnicas de supervisionadas para a realização das tarefas de classificação.

3.1 Aprendizagem

Os problemas de aprendizagem podem ser classificados de forma grosseira como supervisionados e não-supervisionados [73]. Os problemas que se enquadram na categoria supervisionado são aqueles guiados pela presença de uma variável de interesse, de um rótulo ou de uma classe, com o objetivo de prever o valor de uma medida de resultado baseado em série de medidas de entrada. Já a aprendizagem não-supervisionada não possui intervenção humana, e nem uma designação significativa de entradas e saídas, ou seja, nenhuma medida de resultado, sendo o objetivo principal o de descrever associações e padrões de entrada num conjunto de medidas de entrada [73].

A tarefa de aprendizagem supervisionada é a seguinte[164]:

Dado um conjunto de treinamento de **n** partes de exemplos de entrada e saída

$$(x_1, y_1), (x_2, y_2), ...(x_n, y_n) \tag{3-1}$$

onde cada y_j foi gerado por uma função desconhecida $y = f(x)$, deseja-se descobrir uma função h que se aproxime da função verdadeira f. A aprendizagem é uma busca através de hipóteses possíveis por aquele algoritmo que poderá retornar um bom desempenho, mesmo utilizando novos exemplos além do conjunto de teste e treinamento.

Os algoritmos utilizados pela aprendizagem supervisionada necessitam de características e de uma classe de rótulos. As características são as informações que estão

disponíveis para resolução de determinado problema. Já a classe ou preditor é o conhecimento adquirido a partir de determinado conjunto de características. Como exemplo podemos citar a classe mão e classe pé. As duas classes possuem algumas características comuns, porém existem características determinantes e discriminantes das duas classes como a localização e a forma.

Se determinado indivíduo possuir o pé localizado no braço e a mão localizada na perna, então "esse ser"não será classificado corretamente na classe humano, caso estes elementos sejam características determinantes, será considerado um valor extremo (outlier), ou seja um indivíduo que possui determinadas características diferentes dos demais.

Exemplos de problemas que podem ser resolvidos pela aprendizagem supervisionada são:

(1) *Identificação de Spam* por meio de um o rótulo que será utilizado para identificar se determinado e-mail é ou não um *Spam*.

(2) *Identificação de câncer de próstata* a partir de características de determinado paciente que possui ou não câncer, sendo considerado um problema de regressão, pelo fato da medição dos resultados ser feita de forma quantitativa.

Na aprendizagem não-supervisionada não verificamos a presença de uma variável de interesse, apenas recursos e nenhuma medida de resultado, sendo que o foco é a formação de agrupamento para ser possível efetuar a criação de classes com rótulos. Uma tarefa comum da aprendizagem não-supervisionada é a descoberta de grupos de determinadas observações chamadas de clusters, obtidos por meio dos dados de treinamento, que são analisados grupo a grupo[64][89].

3.2 Dados de teste e treinamento

Na aprendizagem supervisionada, uma questão chave presente é a seguinte: Como será possível predizer ou classificar um modelo utilizando um novo conjunto de dados? Possuímos um interesse particular em comparar a performance de vários modelos, porém será escolhido apenas um modelo, ou seja, deseja-se identificar o modelo com melhor performance.

Em um primeiro momento é comum pensar que o melhor modelo é aquele que classifica as variáveis de interesse ou características de um determinado conjunto de dados. Porém, quando sâo utilizados os mesmos dados para o desenvolvimento do modelo, testar e treinar, superestima-se a performance, introduzindo então um bias. Quando um modelo trabalha bem com um determinado conjunto de dados, é preciso verificar se ele também vai ter a mesma performance e resultado positivo na classificação

utilizando outro conjunto de dados para encontrar e selecionar o melhor modelo dentre os modelos. Dessa forma uma solução consiste em particionar o conjunto de dados e desenvolver o modelo utilizando apenas uma das partições. Assim, com o modelo pronto, basta utilizar a outra partição para ver o que acontece. Essa divisão dos dados é chamada de etapa de treinamento e etapa de teste. Porém o ideal é dividir os dados em três etapas: Treinamento, teste e validação.

As etapas de teste e treinamento para a utilização e aprendizagem dos algoritmos de classificação são importantes e necessárias para evitar anomalias nos resultados com o *overfitting*, que trata da memorização dos resultados do treinamento, o que não é um bom sinal, sinalizando que o algoritmo não efetuou a tarefa de forma adequada, pelo fato de ter memorizado as relações e estruturas, bem como os ruídos ou coincidências.

Além das etapas de treinamento e teste existe uma outra etapa de observações que se chama validação ou hold-out set, que pode ser necessária, uma vez que ela é utilizada para o ajustamento e sintonia de variáveis chamadas hiper parâmetros que controlam a forma de aprendizado do modelo em questão.

Não há requisitos para a alocação do tamanho das partições de teste e treinamento, sendo que podem variar de acordo com a quantidade de dados disponíveis. Costuma ser uma prática comum a alocação de 50% dos dados ou mais para o treinamento, uns 25% para o conjunto de teste e os dados remanescentes para a validação do modelo. Pode-se também utilizar a validação cruzada para efetuar as etapas de teste e treinamento.

3.2.1 Balanceamento

Problemas como balanceamento dos dados, generalização de resultados, *overfitting* e *underfitting* são comuns em problemas reais, e precisam ser resolvidos por meio de estratégias. Dessa forma é preciso aplicação de técnicas adequadas nos modelos para dessa anomalias.

Existem situações onde a quantidade de elementos de classes distintas são desproporcionais. Dessa forma para minimizar a distinção entre as classes e evitar a tendência dos classificadores predizerem as majoritárias, procura-se utilizar técnicas de balanceamento [27]. Para o caso do balanceamento, uma das técnicas mais utilizadas para esta tarefa é a chamada *SMOTE* (Synthetic Minority Oversampling Technique) [27], onde a classe minoritária é sobre-amostrada através da criação de exemplos "sintéticos". Estes exemplos sintéticos são introduzidos ao longo da linha de segmentos de entrada de todos os k da classe minoritária de vizinhos mais próximos, que são escolhidos de forma aleatória dependendo da quantidade de sobre-amostragem, e não réplicas dos dados existentes. A principal limitação do *SMOTE* encontra-se na forma como os exemplos

são gerados, utilizando a interpolação dos dados minoritários existentes no escopo dos exemplos.

3.2.2 Seleção de Variáveis

Em mineração de dados, de acordo com [18], o número de classificadores que devem ser considerados aumenta exponencialmente com o número de atributos do conjunto de dados, ficando mais difícil para o algoritmo de aprendizagem encontrar um modelo preciso. Esse tipo de situação é chamada de Maldição da Dimensionalidade. Uma das formas de evitá-la é efetuando a redução do número de atributos. Esta redução pode ser feita por meio da retirada de atributos redundantes e irrelevantes a fim de contribuir com a legibilidade dos resultados gerados.

Por meio das técnicas de seleção de variáveis, representadas por métodos wrapper e filtros, torna-se possível efetuar a redução do número de atributos.

Métodos wrapper procuram um subconjunto ótimo de características, adaptado a um determinado algoritmo de classificação e um domínio [107]. O algoritmo de indução é utilizado como uma caixa preta pelo algoritmo de seleção de subconjunto.

Alguns exemplos de filtros existentes são: *Consistency-Base Filter* (CBF), *Correlation-based Feature Selection* (CFS), *InfoGain* e *Relief* e *GainRatio*.

O filtro *CBF* [32] avalia a relevância de um subconjunto de variáveis por um nível resultante de consistência das classes quando os exemplos são projetados em subconjuntos.

O *CFS* utiliza uma avaliação heurística baseada em correlação, referida de maneira formal como coeficiente de correlação de Pearson, para selecionar um subconjunto de características úteis para utilização nos algoritmos de aprendizagem, a fim de melhorar a precisão e o entendimento dos resultados [65].

O Ganho de Informação (do inglês, InfoGain) é considerado um critério de impureza que utiliza a entropia como medida de impureza [156] e está intimamente relacionada com a estimativa de máxima verosimilhança (do inglês, Maximum likelihood estimation - MLE) que é considerado um método estatístico popular utilizado para fazer inferências sobre os parâmetros da distribuição de probabilidade subjacente a partir de um determinado conjunto de dados.

O Algoritmo *Relief* [99] [98] é randômico e atribui um peso relevante para cada característica e destina-se a indicar a pertinência ao conceito alvo e não auxilia na remoção das características redundantes, irrelevantes e com alta correlação.

A Razão de Ganho (do inglês, *GainRation*) normaliza o ganho de informação por meio da divisão entre *InfoGain* e a *Entropia* e é utilizado para reduzir a tendência dos

	A	B	C	D	E
Validação Cruzada interação 1	Teste	Treino	Treino	Treio	Treino
Validação Cruzada interação 2	Treino	Teste	Treino	Treio	Treino
Validação Cruzada interação 3	Treino	Treino	Teste	Treio	Treino
Validação Cruzada interação 4	Treino	Treino	Treino	Teste	Treino
Validação Cruzada interação 5	Treino	Treino	Treino	Treio	Teste

Tabela 3.1: *Validação Cruzada com 5 Folds*

atributos com múltiplos valores, levando em consideração o número e tamanho dos ramos ao selecionar um atributo [150] [156].

3.2.3 Validação Cruzada

Nas etapas iniciais de desenvolvimento e quando os dados de treinamento são escassos, uma prática que costuma ser adotada é a chamada validação cruzada (*Cross Validation*). Ela pode ser utilizada para treinar e validar um algoritmo, pois os dados de treinamento são particionados em *k* conjuntos [69], e o algoritmo é treinado com todos os dados, sendo que uma das partições é testada sobre as demais. Dessa forma as partições são executadas várias vezes de acordo com a quantidade de interações definidas pelo utilizador de modo que o algoritmo fica treinado e avaliado considerando todos os dados. A tabela 3.1 mostra um conjunto de dados particionado em 5 subconjuntos de tamanho igual, rotulados de A até E. Inicialmente, o modelo é treinado em partições de B até E e testado na partição A. No próximo estágio de interação, o modelo é treinado nas partições A, C, D e E e testado na partição B. As partições passam por todos os modelos até finalizar as etapas de treinamentos e testes. Dessa forma a validação cruzada fornece uma estimativa mais precisa do modelo de testes de de desempenho do que uma única partição de dados.

O Leave-one-out é uma forma particular de validação cruzada, pelo fato de considerar o número de partições (*folds*) igual ao número de exemplos.

3.3 Classificadores

Classificadores são algoritmos utilizados para as tarefas automáticas de classificação de dados. Classificação é utiliza para a predição de classes de objetos e pode ser dita como o processo de generalização dos dados a partir de diferentes instâncias. Existe uma tendência de se referir a problemas com uma resposta qualitativa (classe) como problemas de classificação e aqueles com uma resposta quantitativas como problemas de regressão, apesar de nem sempre ser tão simples distinguir isso, pois podemos ter classes que retornam valores e não dados qualitativos.

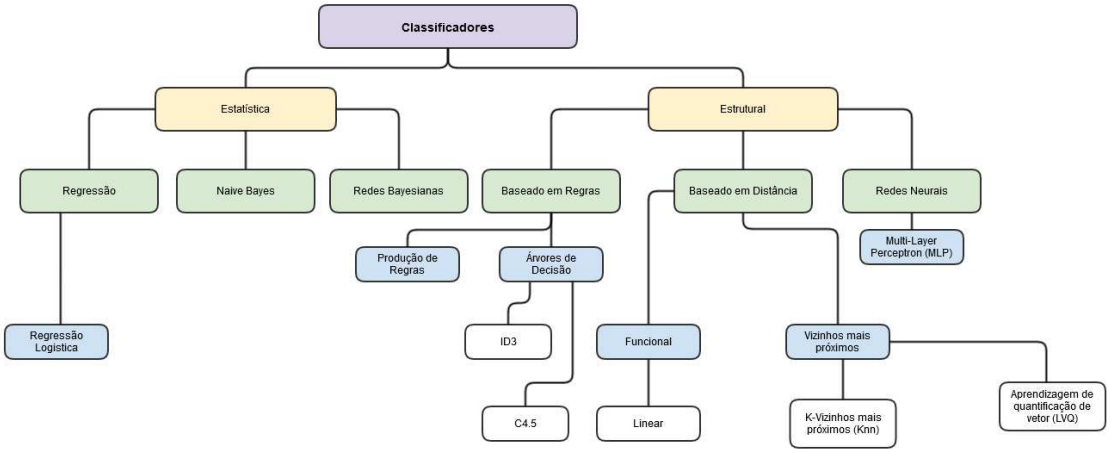

Figura 3.1: *Mapa com os principais classificadores.*

Não existe um único melhor classificador, pois os classificadores são aplicados a problemas distintos e são selecionados a partir do treinamento e comparação de performance de diferentes algoritmos, sendo que os estudos comparativos são baseados em experimentos extensivos, utilizando dados simulados e dados reais. A Figura 3.1 apresenta alguns tipos de classificadores.

De acordo com Dietterich [40] para a comparação de algoritmos de classificação é importante considerar a escolha do conjunto de teste bem como o conjunto de treinamento. É importante perceber também que alguns componentes aleatórios do algoritmo de treinamento, como os parâmetros de inicialização, são inicializados de forma aleatória, e além do mais, quando existem objetos rotulados de forma inconsistente, significa que existe erro aleatório na classificação.

A Tabela 3.3 exibe diferenças entre alguns algoritmos não-supervisionados em termos funcionais, oferecendo um guia para a escolha do classificador. As características e desempenho relativos a cada um dos algoritmos podem variar dependendo dos parâmetros passados para cada classificador, e o quanto os dados estão bem sincronizados.

3.4 Classificadores estatísticos Clássicos

Classificadores estatísticos clássicos são métodos baseados na máxima verossimilhança, que não requerem hipótese de probabilidade e são utilizados para separação de apenas duas classes. No caso básico, tenta-se utilizar um único separador linear para separar as classes utilizando LDA (Linear Discriminante Analysis, Análise Discriminante Linear). Não sendo possível, utiliza-se QDA (Quadratic Discriminant Analysis, Análise Discriminante Quadrática). A figura 3.2 mostra exemplos de LDA e QDA.

Algorithm	Problem Type	Results interpretable by you?	Easy to explain algorithm to others?	Average predictive accuracy	Training speed	Prediction speed	Amount of parameter tuning needed (excluding feature selection)	Performs well with small number of observations?	Handles lots of irrelevant features (separates signal from noise)?	Automatically learns feature interactions?	Gives calibrated probabilities of class membership?	Parametric?	Features might need scaling?	Algorithm
KNN	Either	Yes	Yes	Lower	Fast	Depends on n	Minimal	No	No	No	Yes	No	Yes	KNN
Linear regression	Regression	Yes	Yes	Lower	Fast	Fast	None (excluding regularization)	Yes	No	No	N/A	Yes	No (unless regularized)	Linear regression
Logistic regression	Classification	Somewhat	Somewhat	Lower	Fast	Fast	None (excluding regularization)	Yes	No	No	Yes	Yes	No (unless regularized)	Logistic regression
Naive Bayes	Classification	Somewhat	Somewhat	Lower	Fast (excluding feature extraction)	Fast	Some for feature extraction	Yes	Yes	No	No	Yes	No	Naive Bayes
Decision trees	Either	Somewhat	Somewhat	Lower	Fast	Fast	Some	No	No	Yes	Possibly	No	No	Decision trees
Random Forests	Either	A little	No	Higher	Slow	Moderate	Some	No	Yes (unless noise ratio is very high)	Yes	Possibly	No	No	Random Forests
AdaBoost	Either	A little	No	Higher	Slow	Fast	Some	No	Yes	Yes	Possibly	No	No	AdaBoost
Neural networks	Either	No	No	Higher	Slow	Fast	Lots	No	Yes	Yes	Possibly	No	Yes	Neural networks

Tabela 3.2: *Comparação de Classificadores www.dataschool.io/comparing-supervised-learning-algorithms]*

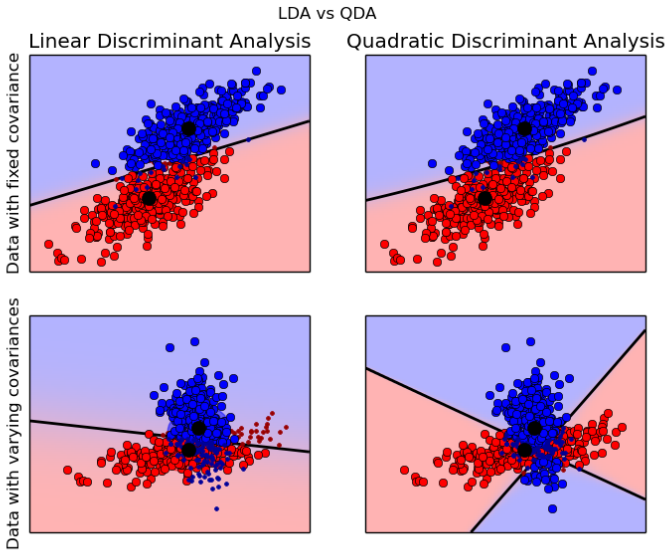

Figura 3.2: *LDA VS QDA*

3.4.1 Análise Discriminante Linear

O Linear Discriminante Analysis (LDA) [72] é representado por uma fronteira do tipo linear, e tem o propósito de classificar objetos em um ou mais grupos baseado num conjunto de características. Ele maximiza o raio entre a variância da classe para a classe de variância vencedora,garantindo assim a máxima separabilidade. E também não altera a localização, mas apenas providencia uma separação entre as classes por meio de uma região de decisão, que ajuda a entender melhor as características dos dados. Esta região, também chamada de fronteira de decisão linear é gerada pelo ajuste da densidade condicional dos dados da classe e usando a regra de Bayes.

O modelo ajusta uma densidade Gaussiana para cada classe, assumindo que todas as classes compartilham a mesma matriz de covariância e o modelo ajustado pode ser utilizado para a redução de dimensão dos dados de entrada, projetando em direções mais discriminativas.

A Função Densidade de Probabilidade (do inglês, Probability Density Functions - PDF) para uma gaussiana multivariada (onde x é o vetor aleatório, σ é uma matriz de covariância e μ é um vetor médio) é dada pela seguinte equação:

$$f(x) = ((2\pi)\frac{n}{2}\left|\sum\right|^{0}.5)^{-.5}\exp(-.5(x-\mu)^{\tau}\sum^{-1}(x-\mu)) \tag{3-2}$$

Os dados devem estar em uma das k classes, dessa forma, θ_i é a probabilidade de ponto de dados de uma classe I, assim, ele também segue $\sum_{i=1}^{k}\theta_{i=1}$ apenas por um simples axiomas da probabilidade.

Tenta-se encontrar a probabilidade de um ponto de dados de uma classe, dado o

que é visto. Pela regra de Bayes, da mesma forma que se deriva um estimador, E se G é a classe para o ponto X, sabe-se que

$$P(G=i|X=r)=\frac{P(X=x|G=i)P(G=i)}{P(X=x)}=\frac{f_i(x)\theta_i}{c} \tag{3-3}$$

, onde c é a constante em termos de G.

Para o LDA, assume-se a mesma matriz de covariância para cada classe. Assim tem-se uma função de probabilidades. Aplicando o logaritmo para as duas classes teremos:

$$\log\frac{P(G=i|X=x)}{P(G=j|X=x)}=\log\frac{f_i(x)}{f_i(x)}+\log\frac{\theta_i}{\theta_j}=\log\frac{\theta_i}{\theta_j}-.5(\mu_i+\mu_j)^{\tau}\sum^{-1}(\mu_i-\mu_j)+x^{\tau}\sum^{-1}(\mu_i-\mu_j) \tag{3-4}$$

,

A partir dessa equação é fácil ver que é possível derivar um classificador linear, assim a função discriminante linear para cada classe i é dada por:

$$y_i(x)=x^{\tau}\sum^{-1}\mu_i-.5\mu_k^{\tau}\sum^{-1}\mu_k+\log\theta_i \tag{3-5}$$

,

onde a classe calculada pertencente aos pontos de dados é a função LDA maximizada.

A complexidade do LDA depende da quantidade de características e dos registros. Se tiver mais características do a quantidade de registros, então a complexidade é dada por $O(d^3)$ caso contrário é $O(Nd^2)$.

3.4.2 Análise Discriminante Quadrática

Pode-se dizer que o Quadratic Discriminant Analysis (QDA) [72] é um tipo de classificador similar ao LDA, porém a fronteira entre duas regiões agora é representada por meio de uma superfície quadrática. Quando se assume que a igualdade de matriz de covariância falha, é obtida uma distribuição normal com a máxima verossimilhança em uma superfície quadrática representada por elipsóides e hyperbolóides por exemplo.

A função discriminante quadrática é obtida de forma simples a partir do logaritmo da função densidade de probabilidade e da substituição dos valores pela média dos exemplos e da matriz de covariância. Na classificação, o QDA é calculado para todas as classes, sendo que a classe com o maior discriminante é escolhida.

Utilizando a equação 3-4 é possível obter a QDA supondo que as matrizes de covariância são diferentes. Então após aplicar o logaritmo não há o cancelamento de

nenhuma matriz de covariância e portanto a equação do QDA será:

$$y_i(x) = -.5\log\left|\sum_i\right| - .5(x-\mu i)^{\tau}\sum_i^{-1}(x-\mu i) + \log\theta_i \tag{3-6}$$

3.4.3 Discriminante Logistico

O Discriminante Logístico (do inglês, Logistic Regression - LR) [72] é similar ao LDA, porém a diferença é apresentada quando os atributos possuem distribuições bem distantes da normal com várias covariâncias diferentes. Considera-se um método parcialmente paramétrico, pois as funções de densidade de probabilidade para as classes não são modeladas, mas sim as relações entre elas.

A maioria dos algoritmos discriminantes logísticos trabalham apenas com duas classes, porém existem alguns que conseguem resolver problemas com mais de duas classes, e a sua complexidade computacional é dada por $O(N.D^2)$ [2].

Um exemplo de algoritmo é a Regressão Logística que, apesar do nome, é um modelo utilizado para a classificação e não para a predição [20] [132]. É também um método para descrever relacionamento entre variáveis categóricas de resposta e um conjunto de variáveis de predição [111] A partir da aplicação de uma equação de regressão em uma função logística, que se transforma utilizando uma chamada Odds (Mudança) [72], obtem-se a equação de regressão logística.

Os tipos de discriminantes são: quadrático (curvado); linear (linha); logístico (linha pontilhada). Os dados são os primeiros dois componentes de Karhumen-Loeve, conforme 3.3 .

- Vantagens: Várias formas de regularizar o modelo, sem a necessidade de se preocupar com os dados que estão sendo correlacionados, com uma interpretação probabilista agradável, facilidade na atualização do modelo. Recomenda-se a utilização quando se deseja uma estrutura probabilística (ajustes de limiares de classificação, trabalhar com incerteza, obtenção de intervalos de confiança) e para incorporar dados de treinamento de forma mais fácil.

3.5 Classificadores estatísticos Modernos

3.5.1 Naive Bayes

Naive Bayes (NB) [44][55][109] é um classificador probabilístico baseado no teorema de Bayes e utilizada uma forte premissa de independência entre os preditores.São suposições raramente seguras para o mundo real. Não é para ser diferente, pois o próprio nome diz isso Naive (Ingênuo). Apesar disso este método é aplicado com sucesso em

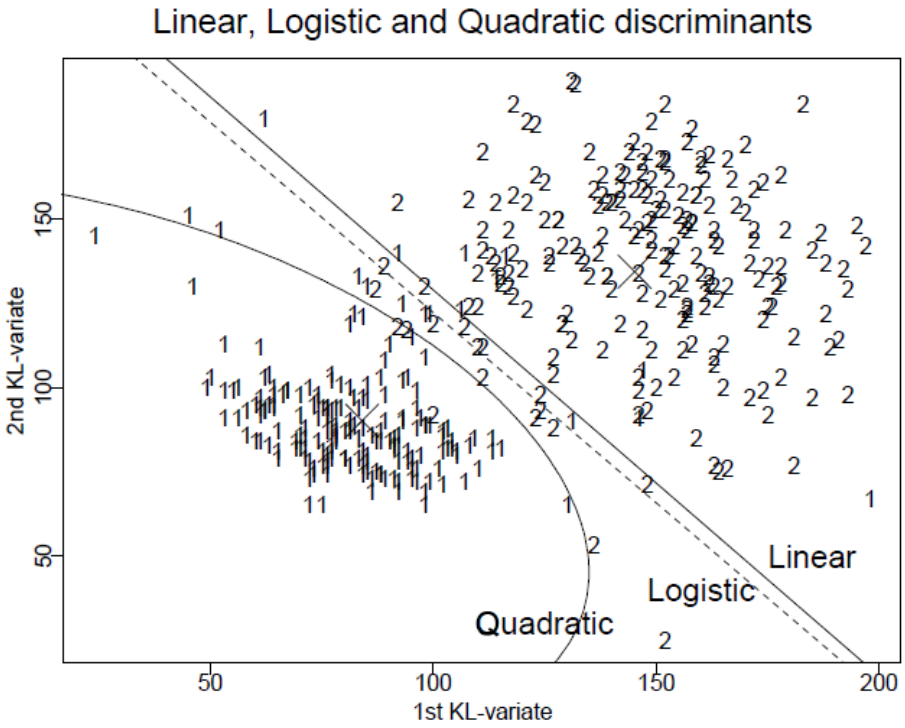

Figura 3.3: *Fronteira de decisão para os três discriminantes*

problemas do mundo real. Ele possui uma complexidade computacional dada por $O(N.D)$ [2].

Dessa forma a partir do teorema de Bayes dado pela probabilidade

$$P(A/B) = \frac{P(B|A)P(A)}{P(B)} \tag{3-7}$$

o classificador Bayesiano calcula a probabilidade por meio de um caso de testes de cada classe dado pela:

$$P(c|X_1,...,X_p) = \frac{P(c)P(X_1,...,X_p|c)}{P(X_1,...,X_p)} \tag{3-8}$$

onde c é a classe e

$$X_1,...,X_p \tag{3-9}$$

são os valores observados pelos preditores para os casos de teste. a probabilidade P(c) pode ser vista como uma expectativa a priori de se obter a classe

$$c.P(X_1,...,X_p|c) \tag{3-10}$$

é a probabilidade do casos de testes retornar a classe c. Finalmente o denominador é a probabilidade da evidência observada.

Esta equação é calculada para todas as possíveis classes de valores para determi-

nar a classe mais provável do caso de teste.

Este decisão depende somente do numerador da equação, sendo que o denominador será constante para todas as classes. Usando apenas definições da estatística na probabilidade condicional e assumindo de forma ingênua a independência condicional dos preditores, os numeradores da fração serão reduzidos para:

$$P(c)P(X_1,...,X_p|c) = P(c)\prod P(X_i|c) \tag{3-11}$$

Naive bayes implementa estimativa de probabilidades de exemplos de treinamento por meio de frequências relativas e a partir disso o método produz classes de probabilidades para todos os casos de teste.

3.5.2 K- Vizinhos Mais Próximos

O K-Nearest Neighbor (KNN) é uma técnica estatística que utiliza métricas de distância para a verificação da similaridade entre os dados, ou seja, baseado em um modelo de k-vizinhos mais próximos, ou em instância. O tempo necessário para classificar um instância de uma classe é alto, uma vez que as distâncias de uma instância a todas as demais dado um conjunto de treinamento deve ser calculada [45].

Umas das dificuldades na utilização do KNN é na decisão de escolha do valor de K, que não é uma tarefa simples, porém existem heurísticas que podem auxiliar nisto, ou pela própria variação do valor de k e verificação da acurácia retornada.

3.5.3 SVM

Máquinas de vetores de suporte ou Suport Vector Machine (SVM) foi introduzida por Vapnik e outros colaboradores em 1992. É uma técnica de classificação e que possui um mecanismo binário e de regressão baseada em aprendizado estatístico,supervisionado e tem por objetivo encontrar o hiperplano entre um conjunto de exemplos com valores positivo e negativos de saídas, assumindo que os dado são linearmente separáveis [29] [45]. Com o SVM é possível resolver problemas de otimização de programação quadrática a fim de maximizar a margem subjetiva para um conjunto de restrições do tipo linear [29] [45]. Na prática de estado da arte as implementações típicas do SVM apresentam uma complexidade de tempo de execução entre $O(m)$ e $O(m^{2.3})$conforme Platt [145].

É possível verificar algumas características que fazem do SVM um algoritmo atrativo para as tarefas de classificação [171]:

- Trabalha de forma robusta em com objetos de grandes dimensões.
- A base teórica está bem definida dentro da matemática e da estatística.

- Otimiza uma função quadrática com um único mínimo local, em contradição às redes neurais que possuem mais de um mínimo local dentro da função objetivo que deve ser minimizada.
- O SVM possui grande capacidade de generalização, sendo esta capacidade medida pela eficiência na classificação de dados que não pertençam ao conjunto de treinamento.

3.6 Aprendizado de máquinas de regras e árvores

A aprendizagem de máquina ou Machine Learning (ML) é um subcampo da ciência artificial que se dedica ao desenvolvimento de algoritmos. Ela é descrita com frequência como o aprendizado a partir da experiência, ou melhor sem a supervisão de humanos. [64]. Nos problemas que envolvem aprendizado supervisionado um programa pode prever a saída de determinada entrada por meio do aprendizado dos pares de entrada e saída.

Existem alguns riscos a cerca do uso de técnicas de aprendizagem de máquina e estes riscos são descritos a seguir [100]:

- Dados não estáveis;
- Underfitting (alto bias);
- Overfitting (sobreajuste) o modelo está ajustado acima do esperado em relação ao conjunto de dados;
- Futuro Imprevisível;

3.6.1 Árvores de Decisão ou classificação

Árvores de decisão ou classificação (classification trees - CT) são utilizadas para classificar objetos ou instâncias para um conjunto de classes pré-definidas baseada nos valores dos atributos. Porém não tem por objetivo substituir outros métodos tradicionais de classificação como os estatísticos e os de redes neurais artificiais, ou máquinas de vetores de suporte.

As árvores de decisão são construídas de acordo com os dados acumulados no banco de dados. E não há a real necessidade de elicitar manualmente o conhecimento, elas fazem este trabalho automaticamente. Assim, o conhecimento adquirido é referenciado como KDD e são técnicas muito populares em mineração de dados. Acredita-se que é pelo fato de serem simples e transparentes, e por serem auto-exploratórias, e por não haver a necessidade de ser um especialista em mineração de dados para utilização destes tipos de algoritmos.

São usualmente representadas por estruturas hierárquicas, sendo mais fáceis de interpretar do que as outras técnicas. Se a árvore de decisão vai se tornando complicada de interpretar, então a interpretação gráfica simples vai se tornando inútil, sendo que para os casos de árvores complexas outros procedimentos gráficos de interpretação poderão ser desenvolvidos para simplificar a interpretação.

3.6.2 C4.5

O C4.5 é um algoritmo de árvore de decisão desenvolvido por Quinlan [150] para utilização livre, descendente do algoritmo ID3 [149] e seguido do C5.0 que utiliza menos memória, sendo mais rápido, além de possuir baixa taxa de erro, apesar do algoritmo possuir uma licença proprietária para fins comerciais, existe uma versão OpenSource escrita em R, utilizando o código GPL escrito em C por Ross Quinlan, fornecido pela RuleQuest [151], por Max Kuhn, Steve Weston e Nathan Coulter.

O C4.5 utiliza *ganho de informação*, que não produz nada de novo, permitindo medir a taxa de ganho que é definida por:

$$GainRatio(p,T) = F(info(T) - info(p,T)) \tag{3-12}$$

onde,

$$Info(p,T) = \sum_{j=1}^{n} p_j x Entropie(p_j) \tag{3-13}$$

$$info(T) = Entropy(T) \tag{3-14}$$

onde,

F = número de exemplos da base de dados com o valor de conhecimento para um determinado número / total de amostras em um conjunto de dados de atributos.

Ele gerencia características com valores em intervalos contínuos e poda (pruning) por meio de um algoritmo baseado na estimativa pessimista da taxa de erro associada com um conjunto de N casos, e de E casos que não pertencem a classe mais frequente. Possui complexidade computacional dada por O(D.N log N) [2].

C5.0 é designado para analisar bases de dados contendo milhares a centenas de milhares de registros e dezenas a centenas de números, data ,hora ou campos nominais [151] é fácil de utilizar e não presume nenhum conhecimento especial de estatística ou aprendizagem de máquina.

3.6.3 Árvore de Classificação e Regressão

O CART (Classification And Regression Trees) é um algoritmo desenvolvido por Brieman, Friedman, Olshen, and Stone em 1984 [136]. Ele gera árvores binárias por meio da divisão de entradas nominais ou de intervalo para nominal, ordinal, intervalo ou de destino.Não é preciso efetuar pré-processamento dos dados (binning), pois os dados são tratados em estado bruto. Utiliza o índice de GINI para medir a impureza no nó da árvore. Para uma classe binária (ou seja não alfanumérica), o índice de GINI mede a impureza pela equação

$$GINI(t) = 1 - \sum [p(t|j)]^2 \tag{3-15}$$

Onde, $p(j|t)$ é a frequência relativa da classe j no nó t.

Quando o nó p é dividido em x partições, a qualidade da separação é dada por

$$GINI_{split} = \sum_{i=1}^{x} \frac{n_i}{n} GINI(t) \tag{3-16}$$

Onde, n_i = número de registro de filhos i n = número de registros do nó p

É possível utilizar CART para problema que envolvem múltiplas classes. Ele utiliza a poda de complexidade de custo mínimo para remover características dos classificadores que não são significantes, e também efetua o balanceamento das classes de forma automática, lidando com valores faltantes, e permite a aprendizagem de sensibilidade de custo e a estimação da probabilidade da árvore.

3.6.4 Florestas aleatória

Florestas aleatórias (do inglês, Random Forest - RF) são um tipo de conjunto de método que efetua predições calculando a média sobre as predições de vários modelos de base independentes [39], sendo um framework muito bem sucedido na classificação e regressão considerando vários campos de aplicação. São construídas a partir da combinação de predições de várias árvores, cada uma das quais é formada de forma isolada e com várias maneiras de construção, dependendo dos métodos de poda.

Para a construção da árvore cada no corresponde a um subconjunto retangular do R^D, e a cada passo da construção da árvore as células associadas com as folhas formam uma partição do R^D. A raíz da árvore corresponde a todo o R^D. A cada etapa da construção uma folha da árvore é selecionada para expansão. [39].

3.6.5 AdaBoost

Adaptive Boosting (também conhecido como AdaBoost) é uma técnica de aprendizagem máquina que pode ser utilizada em conjunto com vários outros algoritmos de aprendizagem. A noção é ter um conjunto de classificadores fracos (classificadores que realizam pelo menos melhor do que aleatório) e minimize o erro total encontrando o melhor classificador em cada etapa do algoritmo.

É um tipo de Ensemble Learning; que é um processo de escolha de múltiplos algoritmos de aprendizagem para uma melhor performance na classificação ou na resolução de um problema. O Algoritmo AdaBoost 1, proposto por Yoav Freund and Robert Shapire é um dos mais importantes métodos de combinação de algoritmos (ensemble), pelo fato de possuir uma sólida formação teórica; sendo muito preciso, simples e com vasto número de aplicações bem sucedidas. [190]. A complexidade computacional é dada por $O(O.R.I.F(X))$[2].

Algorithm 1 Algoritmo AdaBoost para classificação binária

INPUT:

1. Training data: $\mathcal{D} = \{(x_i, y_i), i = 1, \ldots, N\}$, where $x_i \in \mathbf{R}^d$ and $y_i \in \{-1, +1\}$
2. The number of sampled examples in each iteration: m
3. Weak learner: $\mathcal{L}$ that automatically learns a binary classifier $h(x) : \mathbf{R}^d \mapsto \{-1, +1\}$ from a set of training examples.
4. The number of iteration: T

OUTPUT:

1. The final classifier: $H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$

Algorithm

Initialize the distribution $D_0(i) = 1/N, i = 1, \ldots, N$
for $t = 1$ to T **do**
  Sample m examples with replacement from $\mathcal{D}$ according to the distribution $D_{t-1}(i)$.
  Train a binary classifier $h_t(x)$ using the sampled examples
  Compute the error rate $\varepsilon_t = \sum_{i=1}^{N} D_{t-1}(i) I(h_t(x_i) \neq y_i)$ where $I(z)$ outputs 1 when z is true and zero otherwise.
  Exit the loop if $\varepsilon > 0.5$.
  Compute the weight α_t as $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$
  Update the distribution as $D_t(i) = \frac{1}{Z_t} D_{t-1}(i) \exp(\alpha_t I(y_i \neq h_t(x_i))$ where $Z_t = \sum_{i=1}^{N} D_{t-1}(i) \exp(\alpha_t I(y_i \neq h_t(x_i))$.
end for
Construct the final classifier $H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$.

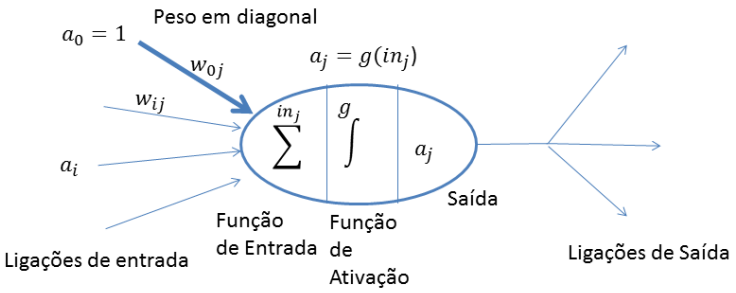

Figura 3.4: *Modelo matemático simples de um neurônio.*

3.7 Redes Neurais

As Redes neurais simulam sistemas biológicos, correspondentes ao cérebro humano. No cérebro humano os neurônios são conectados por uma via de pontos denominados de synapses. A unidade computacional básica de uma rede neural artificial é um neurônio que pode ser arranjada por diferentes tipos de arquiteturas conectadas entre eles. Sendo a arquitetura mais básica o perceptron que contém um conjunto de nós (neurônios) de entradas e nós de saídas.

Um modelo matemático simples de um neurônio desenvolvido por McCulloch e Pitts (1943), é dado pela figura 3.4, onde a ativação de saída da unidade é $a_j = g(\sum ni = 0 w_{i,j} a_i)$, onde a_i, é a ativação de saída da unidade $iew_{i,j}$ é o peso sobre a ligação da unidade i com essa unidade[164].

De forma grosseira, o neurônio da figura 3.4, "'dispara"' quando uma combinação linear de suas entradas excede algum limiar (rígido ou suave).

Os dados que são utilizados pelo perceptron devem ser do tipo numérico. Se por acaso for necessário utilizar dados categóricos, eles precisam ser transformados em dados na forma binária.

A figura abaixo mostra um modelo de uma rede neural simples com uma camada e uma rede de múltiplas camadas 3.5

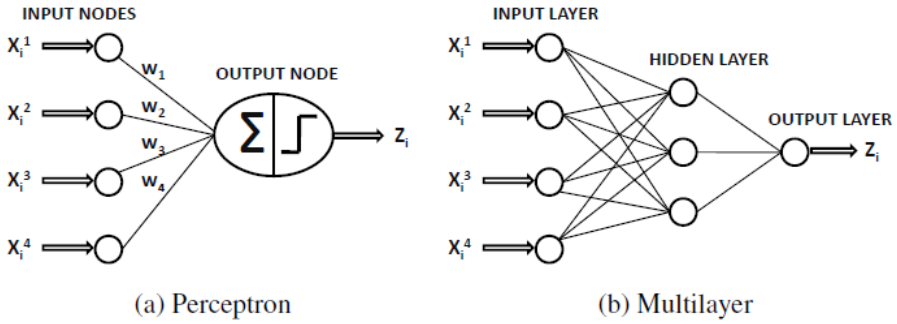

Figura 3.5: *Rede Neural de uma camada de múltiplas camadas*

3.7.1 MultiLayer Perceptron

Multilayer Perceptron (MLP) pode ser definido como uma rede neural de retro propagação com um gráfico acíclico finito, onde os nós são neurônios com ativações por intermédio de funções logísticas [75]. O algoritmo de retro propagação realiza a aprendizagem em uma rede neural de retro propagação de múltiplas camadas, e utiliza determinado número de neurônios, por meio de uma taxa de aprendizagem, com iterações pré-definidas, em relação a um limiar (thresold). No modelo de neurônio a função de ativação mais comum é a sigmoidal. Ele possui uma complexidade computacional dada por $O(D.N.K.I)$.

Pela Figura 3.5 pode-se visualizar a estrutura de um MLP.

3.8 Medidas de desempenho para classificação supervisionada

A partir da construção de modelos de classificação binária e das etapas de teste e treinamento, faz-se necessário verificar sua adequação a um determinado tipo de problema, pois um algoritmo não consegue resolver todos problemas com os melhores resultados possíveis. Dessa forma, nesta seção serão apresentados métodos e instrumentos para avaliar diferentes algoritmos de classificação supervisionada.

3.8.1 Matriz de confusão

A **Matriz de Confusão** (Tabela 3.3) é uma matriz que possui **n** dimensões quando **n** é o número de classes, onde são apresentados uma entrada constituída pelas classes desejadas e outra pelas classes previstas pelo modelo. As linhas correspondem à classe atual no conjunto de dados e as colunas representam classes preditoras. Os elementos da diagonal exibe o número de classificações corretas retornado por cada classe.

Tabela 3.3: *Matriz de Confusão exibe as amostras positivas e negativas que foram classificadas de forma correta ou incorreta.*

		Classe Atual	
		não	sim
Classe Prevista	não	TP	FN
	sim	FP	TN

	A	B
A	25	10
B	0	366

Tabela 3.4: ***Exemplo de Matriz de Confusão.***

onde:

(1) *FP (Falso Positivo)* Quando um exemplo negativo é classificado de forma incorreta como positivo;

(2) *FN (Falso Negativo)* Quando um exemplo positivo é classificado de forma incorreta como negativo

(3) *TP (Verdadeiro Positivo)* Quando um exemplo positivo é classificado corretamente.

(4) *TN (Verdadeiro Negativo)* Quando o exemplo negativo é classificado corretamente.

A Tabela 3.4 ilustra um exemplo de uma matriz de confusão, com a classe B retornando 366 (trezentos e sessenta e seis) instâncias classificadas de forma correta, e 10 (dez) incorretas, e a classe A, com todas as instâncias (vinte e cinco) classificadas corretamente. Nas células da matriz podem ser exibidos o número de instâncias das classes, a Proporção do resultado predito, ou a proporção do atual.

3.8.2 Taxas

Com base na classificação dos objetos da amostra, uma série de outras taxas podem ser calculadas:

(1) Taxa do Verdadeiro Positivo, Sensibilidade ou Recall

A Taxa do Verdadeiro Positivo (do inglês, True Positive Rate - TPR) [68] mais conhecida como taxa de acerto ou sensibilidade ou recall, corresponde à proporção de exemplos positivos que são rotulados corretamente pelo classificador.

$$TPR = \frac{TP}{TP + FN} \tag{3-17}$$

(2) Taxa do Verdadeiro Negativo ou Especificidade

A Taxa do Verdadeiro Negativo (do inglês, True Negative Rate - TNR), ou Especificidade (SPC), corresponde à proporção de exemplos negativos que são rotulados corretamente pelo classificador.

$$SPC = \frac{TN}{FP+TN} x100 \tag{3-18}$$

(3) Taxa de Falso Negativo

A Taxa de Falso Negativo (do inglês, False Negative Rate - FNR) [68], mais conhecida como taxa do falso alarme, corresponde a proporção de exemplos negativos que foram rotulados incorretamente. Esta é uma medida que é comumente usada em problemas de classificação em duas classes onde o foco do problema está em uma classe particular. Por exemplo, no caso de se desejar predizer o abandono, a FNR mostra a taxa de alunos que não abandonam mas são classificados erroneamente como se fossem abandonar. A FNR é calculada pela fórmula:

$$FNR = \frac{FP}{PF+TN} x100 \tag{3-19}$$

A FNR também pode ser expressa como (1-especificidade).

(4) Taxa de Falso Positivo

A Taxa de Falso Positivo (do inglês, False Positive Rate - FPR) [68], corresponde a proporção de exemplos positivos que foram rotulados incorretamente. No exemplo de se desejar predizer o abandono, a FPR mostra a taxa de alunos que abandonam o curso mas são classificados erroneamente como se não fossem abandonar, o que neste caso representa um problema, já que estes alunos não são identificados e portanto acabam não recebendo a atenção necessária. A FPR é calculada pela fórmula:

$$FPR = \frac{FP}{FP+TN} x100 \tag{3-20}$$

(5) Precisão

A Precisão ou Confiança pode ser dita como a proporção de previstos positivos que são de fato verdadeiramente positivos. É a medida de precisão de previsão de uma classe e para seu cálculo utiliza-se a equação a seguir:

$$Precisao = \frac{TP}{TP+FP} \tag{3-21}$$

(6) Gráficos de Elevação e de Ganho (*Lift charts and gain charts*):

Estes gráficos auxiliam na visualização da eficácia e desempenho de um determinado modelo de classificação. A Elevação é uma medida de eficácia, e o ganho avalia o modelo de desempenho utilizando a proporção da população.

(7) F-Measure:

A medida F (do inglês, F-Measure - F1) ou F-Score é uma média harmônica ou a média ponderada da precisão e do recall. Ela pode ser obtida por:

$$F1 = \frac{2TP}{2TP + FP + FN} \tag{3-22}$$

.

(8) Curva ROC (ROC Curve):

A curva ROC (*Receiver Operating Characteristic Curve*) é uma curva parametrizada por um parâmetro de um algoritmo de classificação, por meio dela é possível visualizar a performance de um classificador. Ao contrário da precisão, a curva ROC é insensível quando se trata de conjunto de dados com proporções de classes desbalanceadas. Ela ilustra também o desempenho do classificador para todos os valores do limiar (*threshold*) de discriminação em oposição da *precisão* e *recall*. Mostra um equilíbrio alcançado entre as taxas de TP (TPR) e FP (FPR) de um classificador. Dessa forma pode-se considerar a curva ROC como uma técnica padrão para resumir o desempenho do classificador em um intervalo de trocas entre o TP e FP.

O ideal para qualquer classificador seria obter um TPR = 1 e um FPR = 0, porém na prática isso nem sempre é possível. Os modelos localizados na diagonal são considerados modelos aleatórios, ou seja, possuem TPR = FPR, sendo preferível os modelos que estão localizados acima da diagonal, sendo melhores do que os que se estão abaixo dela.

Uma medida interessante chama-se área sobre a curva ROC (AUC) que pode ser definida com um resumo estatístico que indica que quanto maior a AUC, maior o desempenho retornado por um classificador, ou seja, ele pode ser utilizado para comparar a performance de classificadores. Ele pode ser computado pela seguinte fórmula:

$$A_{ROC} = \int_0^1 \frac{TP}{P} d\frac{FP}{N} = \frac{1}{PN}\int_0 NTPdFP \tag{3-23}$$

A área sob a curva (AUC) é uma métrica de desempenho de aceitação tradicional para uma curva ROC [71][23][112].

Em termos de verificação de valores para a comparação de performance, se o AUC retornado é igual a 1 então pode-se dizer que o classificador é perfeito e caso o AUC seja igual a 0.5, o classificador é considerado randômico. Na prática os classificadores devem estar entre 1 e 0.5 e de preferência o mais próximo o possível de 1. O AUC possui um custo computacional de $O(n^2 logn)$.

A lista abaixo[1], exibe um guia geral para classificar a acurácia de um teste de diagnóstico que considera os valores do AUC e retorna os seguintes qualificadores:

- 0.90 - 1 = Excelente (A)
- 0.80 - 0.90 = Bom (B)
- 0.70 - 0.80 = Fraco (C)
- 0.60 - 0.70 = Pobre (D)
- 0.50 - 0.60 = Falho (E)

3.8.3 Acurácia da classificação

A *acurácia* pode ser definida como a porcentagem de amostras positivas e negativas que foram classificadas corretamente sobre a soma de amostras positivas e negativas, sendo calculada pela seguinte equação:

$$Acurcia(\%) = \frac{TP+TN}{TP+TN+FP+FN}x100 \tag{3-24}$$

Por meio dessas medidas é possível calcular a taxa de erro (Err) de cada classe ou do classificador (Err_classe), e para esta finalidade utilizam-se as seguintes expressões:

$$Err\_classe_1(\%) = \frac{FN}{TP+FN}x100 \tag{3-25}$$

$$Err\_classe_2(\%) = \frac{FP}{FP+TN}x100 \tag{3-26}$$

$$Err(\%) = \frac{FP+FN}{n}x100 \tag{3-27}$$

Um método geralmente aceito para estimar o possível resultado da classificação para dados desconhecidos é utilizar uma parte dos dados para treinamento e outra para teste, sendo que a parte do teste, seria então a estimativa do resultado para os dados desconhecidos.

Então o procedimento seria utilizar parte substancial (*conjunto de treinamento*) dos dados fornecidos para o treinamento e os dados restantes (*conjunto de testes*) são testados e comparados com os resultados de classificação conhecida.

Dessa forma a proporção correta no conjunto de teste é uma estimativa imparcial da precisão da regra, desde que o conjunto de treinamento é amostrado aleatoriamente a partir dos dados fornecidos.

[1] http://gim.unmc.edu/dxtests/roc3.htm

Um importante problema relacionado à acurácia é o chamado *overfitting*, que mede o quando o modelo está sobre ajustado considerando os dados de treinamento e dessa forma o modelo fica especializado de tal maneira que impossibilita a generalização dos dados futuros. Isto acontece quando o modelo é muito complexo em comparação com o tamanho dos dados, e pode ocorrer pelo fato de modelos complexos possuírem maior poder representativo, e tal representação pode considerar até mesmos erros. Por outro lado modelos mais simples possuem um maior poder de generalização, ou seja, conseguem classificar melhor os dados futuros. Se o modelo é simples demais ele pode apresentar um problema de não considerar padrões essenciais nos dados, ou seja, ele não consegue se aproximar do modelo verdadeiro, o chamado *underfitting*.

3.9 Avaliação dos algoritmos

Uma prática comum na Mineração de dados é a realização de testes para avaliar o desempenho dos algoritmos em um dado contexto. Para isto utilizam-se repositórios de dados previamente testados e classificados. Nesta seção será realizada a avaliação dos algoritmos para o contexto educacional.

O UCI [2] é um repositório de dados ou repositório de aprendizagem de máquina que possui base de dados públicas de uso geral. Ela costuma ser utilizada para avaliação de algoritmos de mineração dados antes de serem aplicados a problemas reais, e possui centena de dados disponibilizados por fontes governamentais, ou universidades, contemplam várias áreas do conhecimento como Medicina, Agricultura, Finanças e Educação. A comunidade acadêmica tem utilizado este repositório, a nível mundial para testar algoritmos como visto por Soares [172]. Como que estes dados estão bem estruturados e preparados para o processo de DM e alguns nem precisam passar por etapas de pré-processamento.

Visto o interesse do trabalho em dados educacionais, buscamos no UCI bases relacionadas com este critério e que fossem direcionadas para resolução de problemas de classificação, ou seja, com rótulo pré-definido.

Selecionamos o *dataset Student Performance* fornecido por Paulo Cortez da Universidade do Minho de Portugal [30] em 27-11-2014 para tarefa de classificação e Regressão, com características multivariadas dos alunos, contendo 649 (seiscentos e quarenta e nove) instâncias e 33 (trinta e três) atributos onde um deles é o atributo de saída que classifica os alunos de acordo com o abandono ("sim"ou "não"). Na Tabela 3.6 é possível ver o nome, descrição e domínio de cada um dos atributos.

[2] https://archive.ics.uci.edu/ml/index.html

NúmeroAtributo		Descrição(Domínio)
1	school	student's school (binary: 'GP' - Gabriel Pereira or 'MS' - Mousinho da Silveira)
2	sex	student's sex (binary: 'F' - female or 'M' - male)
3	age	student's age (numeric: from 15 to 22)
4	address	student's home address type (binary: 'U' - urban or 'R' - rural)
5	famsize	family size (binary: 'LE3' - less or equal to 3 or 'GT3' - greater than 3)
6	Pstatus	parent's cohabitation status (binary: 'T' - living together or 'A' - apart)
7	Medu	mother's education (numeric: 0 - none, 1 - primary education (4th grade), 2 " 5th to 9th grade, 3 " secondary education or 4 " higher education)
8	Fedu	father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 " 5th to 9th grade, 3 " secondary education or 4 " higher education)
9	Mjob	mother's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')
10	Fjob	father's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')
11	reason	reason to choose this school (nominal: close to 'home', school 'reputation', 'course' preference or 'other')
12	guardian	student's guardian (nominal: 'mother', 'father' or 'other')
13	traveltime	home to school travel time (numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour)
14	studytime	weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours)
15	failures	number of past class failures (numeric: n if 1<=n<3, else 4)
16	schoolup	extra educational support (binary: yes or no)
17	famsup	family educational support (binary: yes or no)
18	paid	extra paid classes within the course subject (Math or Portuguese) (binary: yes or no)
19	activities	extra-curricular activities (binary: yes or no)
20	nursey	attended nursery school (binary: yes or no)
21	higher	wants to take higher education (binary: yes or no)
22	internet	Internet access at home (binary: yes or no)
23	romantic	with a romantic relationship (binary: yes or no)
24	famrel	quality of family relationships (numeric: from 1 - very bad to 5 - excellent)
25	freetime	free time after school (numeric: from 1 - very low to 5 - very high)
26	goout	going out with friends (numeric: from 1 - very low to 5 - very high)
27	Dalc	weekend alcohol consumption (numeric: from 1 - very low to 5 - very high)
28	Walc	weekend alcohol consumption (numeric: from 1 - very low to 5 - very high)
29	health	current health status (numeric: from 1 - very bad to 5 - very good)
30	absences	number of school absences (numeric: from 0 to 93
31	G1	first period grade (numeric: from 0 to 20)
32	G2	second period grade (numeric: from 0 to 20)
33	G3	final grade (numeric: from 0 to 20, output target)

Tabela 3.5: *Detalhamento das informações dos dados dos alunos da base UCI*

Tabela 3.6: *Resultado dos classificadores para predição de abandono de estudantes utilizando a base de dados UCI students-mat com validação cruzada com 10 subconjuntos.*

Algoritmo	Acurácia	Erro
DecisionTreeClassifier	87,88	12,12
LDA	88,5	11,5
QDA	79,28	20,72
Logistic Regression	**91,87**	8,13
KNN	88,7	11,3
SVM	88,73	11,27
AdaBoost	89,92	10,08
RandomForest	89,84	10,16
J48	87,87	12,13
MLPClassifier	88,07	11,93

A base de dados foi importada para um banco de dados SQLlite a fim de facilitar a análise. Efetuamos algumas alterações na base de dados, substituindo strings por valores inteiros, ou seja, o mesmo que utilizar o filtro de nominal para binário, pois alguns algoritmos precisam receber entradas numéricas como MLP, SVM, LR por exemplo. O estudo teve como objetivo comparar o desempenho dos 10 (dez) algoritmos de classificação (Logistic Regression (LR), AdaBoost (AD), Random Forest (RF), SVM, KNN, LDA, MPLClassifier, DecisionTree (CART), J48 (C4.5) e QDA) verificando a existência de mudanças de acurácia considerando diferentes tipos de técnicas de particionamento dos dados.

3.9.1 Experimento com 10 folds cross validation

A fim de avaliar o comportamento da aplicação da validação cruzada no conjunto reduzido de dados do UCI, utilizamos o ambiente Weka Experiment Environment (WEE) da ferramenta Weka [21] inserindo 10 algoritmos com 10 folds cross validation, sendo executados 10 vezes. Os resultados da acurácia da classificação estão disponíveis na Tabela 3.7.

O WEE gerou os valores médios da acurácia apresentados pelo Gráfico 3.6 após 10 execuções de cada um dos classificadores. Verifica-se que o algoritmo LR obteve melhor acurácia e o QDA a pior.

3.9.2 Experimento com Train/Test Split (data randomized)

Neste experimento também executado no WEE com a seleção do método de particionamento de forma randômica em Teste e Treinamento, onde 66% dos dados

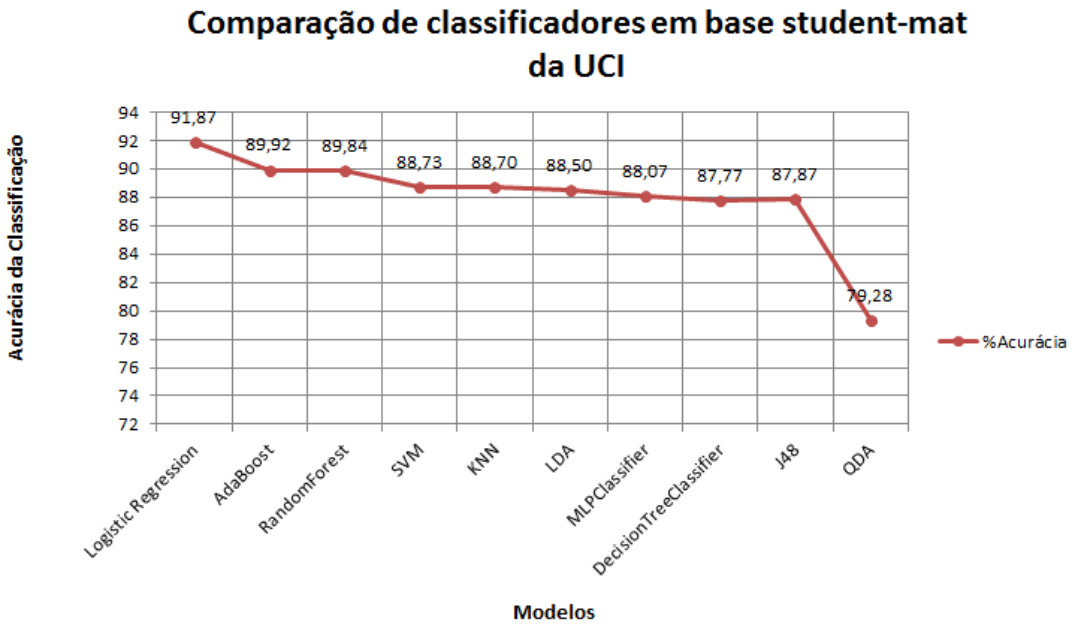

Figura 3.6: *Gráfico com os percentuais de acurácia gerados por cada classificador utilizando a base de dados studend-mat do UCI utilizando o método de validação cruzada particionando os dados em 10 subconjuntos.*

foram usados para o Treinamento e o valor restante para a etapa de teste. A configuração de 10 execuções para cada um dos classificadores foi considerada. Dessa forma queremos verificar se a escolha deste método de particionamento dos dados afetará o resultado final da classificação de cada um dos algoritmos influenciando nos resultados de acertos ou erros.

Os resultados gerados após 10 execuções de cada classificador, porém considerando uma partição de teste e outra de treinamento divididos de forma aleatória, podem ser observados no gráfico 3.7.

Considerando cada um dos métodos de partição: validação cruzada; teste e treinamento dos dados, foi possível observar diferentes valores de acurácia para cada um dos algoritmos, contudo, com baixa variação percentual, sendo que o classificador CART, que aparece na Figura 3.7 e na Figura 3.6, permaneceu com uma acurácia média mais estável após execução do método de validação cruzada. O método de validação cruzada possui um custo computacional maior, e que vai aumentando de acordo com o tamanho da base de dados, porém retorna um resultado mais realista. Logo, por meio do experimento utilizando dados educacionais disponíveis no repositório de dados UCI, foi possível verificar a partir da avaliação inicial dos resultados do desempenho dos classificadores definidos neste capítulo para posterior aplicação na base de dados dos alunos da UFG que é possível utilizar estes algoritmos para classificação de estudantes de acordo o resultado final do aluno, sendo

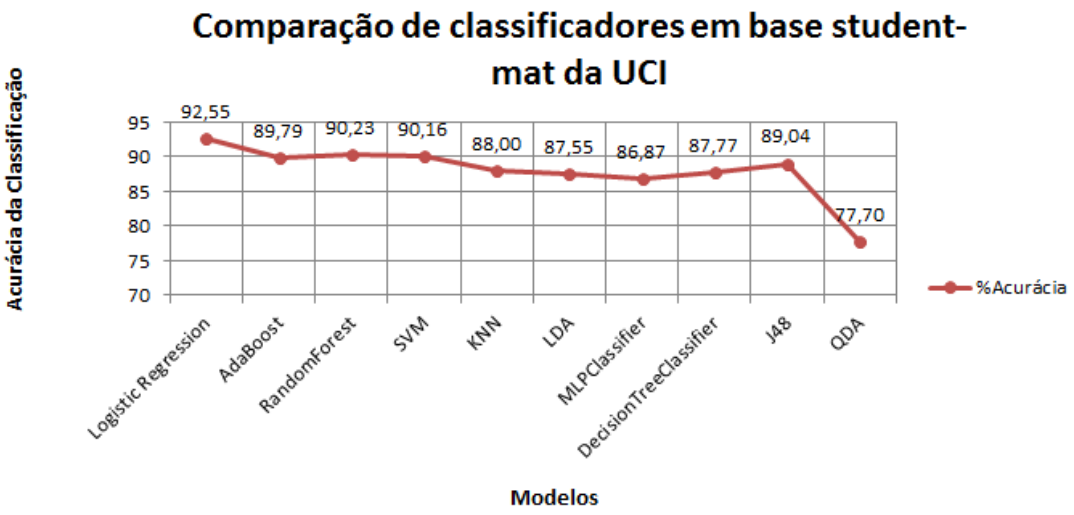

Figura 3.7: *Gráfico com os percentuais de acurácia gerados por cada classificador utilizando a base de dados studend-mat do UCI particionando os dados em teste e treinamento.*

estes experimentos inciais, uma referência para a avaliação final que será efetuada na seção de resultados, relatando os classificadores que obtiverem maior acurácia

CAPÍTULO 4

Mineração de Dados Educacionais utilizando classificadores: uma Revisão Sistemática da Literatura

Mineração de Dados Educacionais (do inglês, Educational Datamining - EDM), pesquisa a implementação de métodos e desenvolvimento de ferramentas para a análise de dados oriundos de instituições de ensino com o objetivo de descobrir "novas"informações que permitem melhor compreender os alunos e os ambientes nos quais apreendem [140]. É uma nova vertente na pesquisa relacionada à mineração de dados ou KDD. Levantamentos bibliográficos feitos por Romero e Ventura [160], cobrindo o período de 1995 a 2005, estendido em 2010 [157], Baker e Yacef [15], publicado em 2009, Jindal e Borah [92], cobrindo o período de 1998 a 2012, Peña-Ayala [143], com trabalhos publicados entre 2010 e 2013, e Thakar [178] (2002-2014), apresentam uma revisão do estado da arte em EDM e as tendências de pesquisa na área. Luan [115] discute as aplicações potenciais de EDM no ensino superior e explica como mineração de dados economiza recursos maximizando eficiência acadêmica.

Os métodos usados em EDM vêm de diversas áreas do conhecimento, incluindo Mineração de Dados (do inglês, *Data Mining*) e aprendizado de máquina, psicometria e outras áreas da estatística, visualização de informações, e modelagem computacional. Romero e Ventura [160] classificaram os trabalhos em EDM em duas categorias: *Estatística e Visualização*, e *Mineração da Web*, que incluem as técnicas de mineração de dados como classificação e clusterização.

Estes métodos são usados para abordar variados tipos de problemas em ambiente educacional, que podem ser acadêmicos ou administrativos como discute Jindal et al. [92]. Exemplos incluem, retenção de alunos, adaptação de ambientes de aprendizagem às necessidades dos alunos, criação de modelos de comportamento do aluno, formação de grupos, identificação da estrutura de domínios de conhecimento, apoio pedagógico para o aluno, contribuição para teorias educacionais, entre outros.

Para apoiar o trabalho sendo desenvolvido, foi realizada a presente Revisão Sistemática da Literatura (RSL), com o propósito de compreender quais técnicas de classificação e ferramentas (toolboxes) estão sendo mais utilizadas, verificando também a existência de trabalhos que tratam da utilização de técnicas de EDM de forma a predizer o abandono e o desempenho acadêmico. Como contribuição desse estudo podemos destacar:

1 Fornecer uma visão geral dos algoritmos de classificação utilizados para Mineração de Dados Educacionais.
2 Apresentar as principais ferramentas utilizadas para a Mineração de Dados Educacionais.
3 Identificar preditores e classificadores para o contexto de predição de desempenho e abandono

4.1 Métodos

A metodologia seguida qualifica esta RSL como uma avaliação qualitativa sistemática dos resultados empíricos da investigação acerca da classificação de abandono e desempenho de estudantes de acordo com Okoli et. al [135].
A fim de realizar a revisão da literatura foi definido um protocolo de avaliação, que consiste em três etapas distintas:

Etapa 1: Planejamento da revisão
Atividade 1.1: Identificação da necessidade de uma revisão
Atividade 1.2: Desenvolvimento de um protocolo de revisão
Etapa 2: Condução da revisão
Atividade 2.1: Identificação da busca
Atividade 2.2: Seleção de estudos primários
Atividade 2.3: Estudo de qualidade
Atividade 2.4: Extração de dados
Atividade 2.5: Sintetização de dados
Etapa 3: Relatando a revisão
Atividade 3.1: Comunicando os resultados

4.2 Planejamento e Condução da revisão

Para a realização desta revisão sistemática utilizou-se a abordagem básica proposta de acordo com Kitchenham et al.[102], a fim de realizar os objetivos mencionados anteriormente, baseando-se nas seguintes questões de pesquisa:

(1) QP1 - Algoritmos: Quais algoritmos de classificação são usados para predizer desempenho e abandono?
(2) QP2 - Classificadores: Quais classes são usadas para a análise de desempenho e abandono?
(3) QP3 - Preditores: Quais atributos são usados para predizer desempenho e abandono?
(4) QP4 - Ferramentas OpenSource: Quais ferramentas ou toolboxes Open-Source são utilizados para as tarefas de classificação de estudantes em Base de Dados Educacionais?

Os artigos, objetos de pesquisa, foram trabalhos publicados sobre a utilização de algoritmos de classificação para classificar alunos de acordo com o desempenho ou abandono. A questão principal norteou a definição das questões de pesquisa que serão respondidas a partir da leitura dos trabalhos selecionados nas etapas de extração e inclusão, sendo que os critérios de exclusão possuem um grau de importância maior que os de inclusão.
Para definir os termos usados na pesquisa, foi feita uma pesquisa nas bases de dados internacionais de recursos acadêmicos e editoras, como Scopus, ERIC, o Google Scholar, Science Direct, DBLP, ACM Digital Library, IEEE, SpringLink. Os termos de pesquisa incluíram "student dropout", "student performance"e "predict". O processo de busca se estendeu de janeiro de 2015 a março de 2016. Foram selecionados artigos publicados nos últimos dez anos (2007 a 2016).
A escolha das bases de dados acadêmicas para pesquisa levou em consideração, além da relevância para a área da pesquisa, critérios auxiliares como:

(1) O fornecimento de mecanismos de busca que implementam expressões lógicas utilizando uma *string* de busca;
(2) Disponibilizar mecanismos de busca utilizando a web;
(3) Aceitar resultados que se relacionem a temas da área de Computação em específico na área de Mineração de Dados;
(4) Facilitar a pesquisa por meio de filtro de acordo com o ano de publicação dos trabalhos.

Para o propósito desse estudo, foram usadas as seguintes bases de dados internacionais: (a) ACM Digital Library, (b) IEEE Xplore, (c) ScienceDirect e (d) Scopus. As buscas foram restritas a trabalhos publicados em língua inglesa e/ou portuguesa entre 2007 e 2016. A *string* de busca utilizada foi: ((student dropout* OR student peformance OR predict)). A Tabela 4.1 apresenta o protocolo utilizado em cada base de dados.

1. idiomas

Tabela 4.1: *Strings de busca.*

Fonte	***String* de busca utilizadas nas fontes ACM, IEEE, ScienceDirect e Scopus**	**Nota**
ACM	query: (student dropout student performance predict) filter: publicationYear: gte:2007 , owners.owner=GUIDE"	Busca em "Advanced Search", filtro de data adicionado manualmente
IEEE	(((student dropout) AND student performance) AND predict)	Busca em "Command Search", filtro de data adicionado manualmente
ScienceDirect	pub-date > 2007 and pub-date < 2016 and student dropout and student performance and predict[All Sources(Computer Science)].	Busca em "Advanced search", filtros "pub-date" e "All Sources" adicionados manualmente
Scopus	(TITLE-ABS-KEY(student dropout) AND TITLE-ABS-KEY (student performance) AND TITLE-ABS-KEY(predict)) AND PUBYEAR > 2007 AND PUBYEAR <2016	Busca em "Advanced search", filtros "LIMIT-TO" adicionados manualmente

Português e inglês

2. filtro de busca

   Publicações entre 2007 e 2016. Utilizar a *string* de busca definida.

3. **Fase de extração**: Serão aplicados critérios de inclusão e exclusão por meio da leitura completa de cada um dos artigos selecionados.

   Critérios para exclusão

   E1: Não tratam de desempenho ou abandono

   E2: Não possui dados de estudantes

   E3: Publicado fora do período de 2007 a 2016

   E4: Trabalho sem utilização de algoritmo de classificação

   E5: Não tratam de Mineração de Dados Educacionais

   Critérios para inclusão

   I1: Os documentos devem estar na Web

   I2: Estudos conceituais sobre Mineração de Dados Educacionais baseados em evidência

   I3: Tratam de desempenho ou abandono de alunos

   I4: Publicações escritas em português ou inglês

Figura 4.1: *Gráfico Radial dos arquivos rejeitados na etapa de seleção.*

I5: Publicado no período de 2007 a 2016

4. **Etapa de seleção**: Foram aplicados os critérios de exclusão e inclusão na leitura do título, palavras chaves e resumo.

4.3 Execução

Trata-se da etapa de aplicação dos itens definidos no protocolo a fim de dar continuidade ao processo de pesquisa. Para auxiliar nesta etapa, utilizou-se o software StArt [1][77], que auxiliou na organização das etapas descritas a seguir:

(1) Pesquisa da literatura - coleta dos dados.
(2) Revisão e avaliação dos resultados da pesquisa - seleção dos estudos primários.
(3) Análise, codificação e síntese dos resultados.
(4) Relatório da avaliação.

4.3.1 Seleção

A aplicação da string de busca nas bases acadêmicas retornou 156 (cento e cinquenta e seis) artigos, oriundos das diversas fontes pela Figura 4.2, sendo 78 (50%)

[1] http://lapes.dc.ufscar.br/tools/start_tool

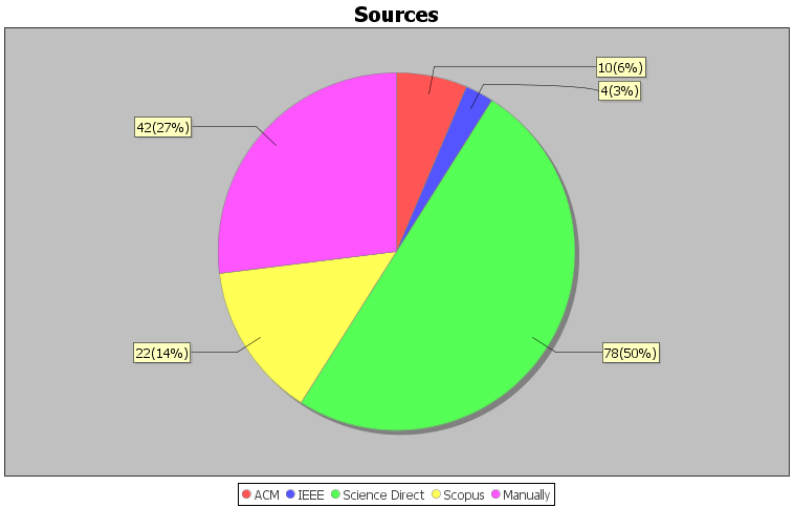

Figura 4.2: *Fontes de pesquisa dos artigos selecionados.*

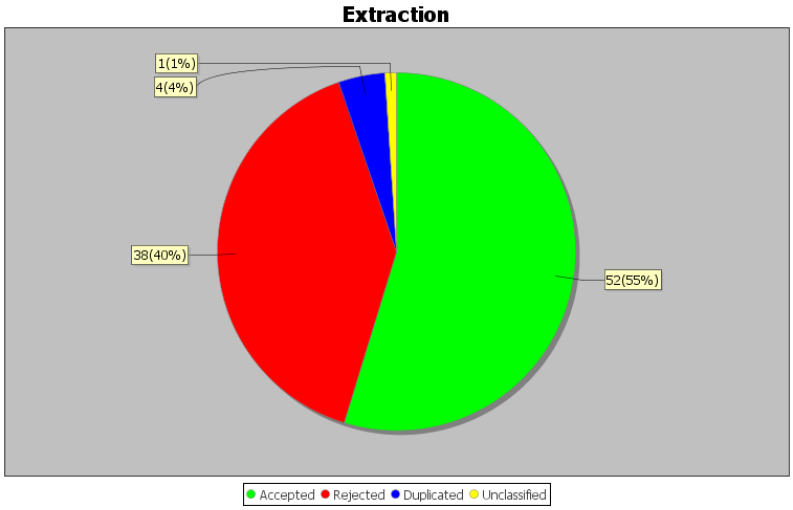

Figura 4.3: *Análise dos artigos selecionados*

retornados pela ScienceDirect, 42 (27%) obtidos manualmente de pesquisas durante o decorrer do mestrado, 22(14%) obtidos na Scopus, 10 (6%) obtidos na ACM e 4 (3%) obtidos na IEEE.

A aplicação dos filtros iniciais, conforme os critérios de inclusão e exclusão apresentados anteriormente, levou à seleção de 95 (noventa e cinco) deles, e exclusão de 57 (cinquenta e sete), sendo que destes 3 (três) eram duplicados e 1 (um) não classificado. Um gráfico Radial dos artigos rejeitados pode ser visto por meio da Figura 4.1.

Os 95 (noventa e cinco) artigos selecionados foram lidos e classificados de acordo com os critérios de inclusão e exclusão. Como pode ser observado na Figura 4.3, 52 (55%) dos artigos foram aceitos, 38 (40%) foram rejeitados, 4 (4%) eram duplicados e 1 (1%) não-classificado.

Os critérios para aceitar os artigos podem ser vistos na Figura 4.4. Percebe-se que

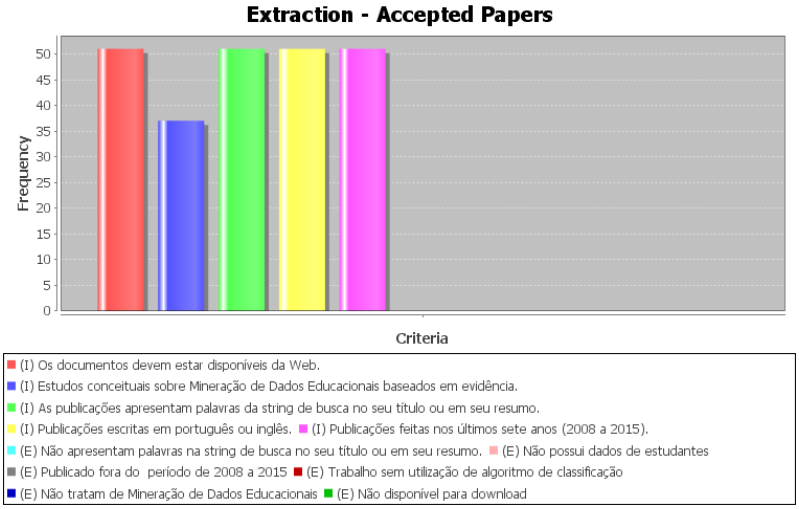

Figura 4.4: *Artigos aceitos na etapa de extração.*

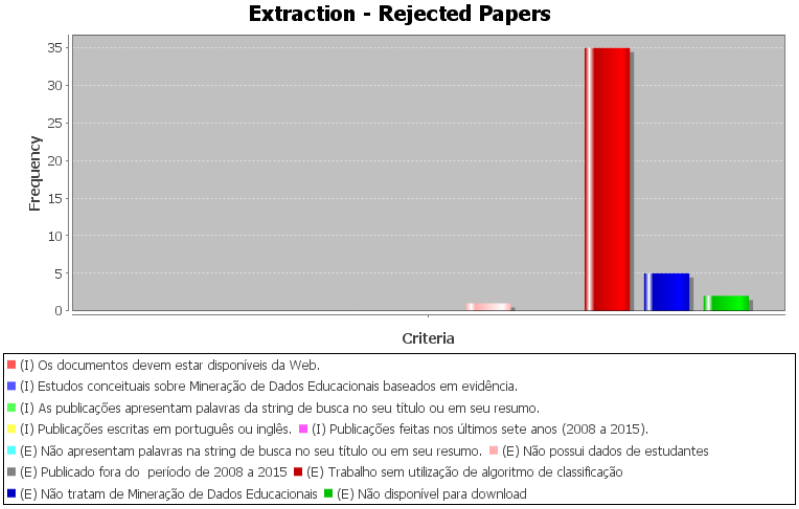

Figura 4.5: *Artigos rejeitados na etapa de extração.*

alguns artigos não se enquadram em todos os critérios e não tratam de estudos conceituais sobre Mineração de Dados Educacionais baseados em evidência.
Considerando a Figura 4.5, percebe-se que um dos maiores motivos da rejeição dos artigos é o fato de não utilizarem algoritmos de classificação, sendo que os outros critérios de exclusão aparecem, porém em menor quantidade.

4.4 Resultados

Durante a leitura dos artigos aceitos foram feitos os levantamentos relativos às informações relacionadas às questões de pesquisa apresentadas. Vale salientar a diversidade encontrada na origem dos dados, que são provenientes de instituições de ensino do mundo todo, incluindo Estados Unidos, Alemanha, Reino Unido, Espanha, Paquistão, Palestina, India, Malásia, Austrália, Arábia Saudita, México,

Colômbia, Turquia, entre outros. Quanto aos conjuntos de dados utilizados no trabalhos, estes variam entre 500-20.000 linhas, com uma média de 7.200 linhas, e foram coletados de uma ou várias universidades, podendo abranger mais de um instituto ou departamento e em alguns casos durante vários anos.

A seguir serão analisados os resultados seguindo as questões de pesquisa definidas.

QP1 - Algoritmos: Quais algoritmos de classificação são usados para predizer desempenho e abandono?

A Tabela 4.2 mostra quais técnicas de classificação estão sendo utilizadas nos artigos. Verifica-se que vários trabalhos utilizam mais de uma técnica, e portanto encontram-se referenciados em mais de uma linha da tabela, sendo que 41 artigos utilizaram mais de uma técnica de classificação e apenas 9 (nove) utilizaram apenas uma técnica para as tarefas de classificação.

Dos 52 (cinquenta e dois) artigos aceitos, 39 (trinta e nove) artigos utilizaram Árvores de Decisão (J48 e outras versões do C4.5, CART, ID3, CHAID, JRip, ADTree, REPTree, entre outros), isto é, 75% dos artigos. Outras técnicas bastante utilizadas incluem Naive Bayes (16 artigos), Redes Neurais (14 artigos), e Suport Vector Machine (14 artigos). Técnicas menos utilizadas incluem Boost (6 artigos), Nearest Neighbour (6 artigos), Bagging (4 artigos), Regressão Logística (3 artigos), Lógica Fuzzy (2 artigos), e Algoritmos Genéticos (1 artigo). A preferência pelo uso de árvores de decisão está na facilidade de interpretação das regras utilizadas para se chegar até o resultado final de classificação, já que as regras são exibidas por meio de scores de decisão e também pela forma "se-então", através da extração das regras. Outras técnicas, como SVM, apresentam-se como "caixas-pretas", onde o usuário não consegue identificar de maneira clara como foi feita a classificação.

Nos trabalhos considerados não foi verificada a utilização do método de exame de partículas(do inglês, Particle Swarm Optimization - PSO) e nem classificadores que utilizam estratégias evolucionárias.

Além dos algoritmos de classificação propriamente ditos, as técnicas de mineração de dados oferecem outras ferramentas para o tratamento dos dados. Entre elas ferramentas para o balanceamento dos dados e seleção de atributos. Apenas 6 (seis) trabalhos analisados utilizaram a técnica de balanceamento SMOTE, um utilizou Oversampling, e um MetaCost. Os demais não utilizaram nenhuma técnica de balanceamento. Em 19 (dezenove) trabalhos foram localizadas técnicas de seleção de características e a técnica mais utilizada foi a InfoGain com 10 (dez). A Tabela 4.3 mostra um resumo das técnicas de seleção de atributos que foram localizadas nos objetos de pesquisa.

Os trabalhos aceitos para a RSL apresentaram resultados de acurácia acima de 50%, sendo que em média a acurácia obtida foi de 86,36%. O melhor resultado obtido

Tabela 4.2: *Algoritmos Classificadores encontrados nos artigos*

Algoritmos Classificadores	**Artigos**
Decison Tree	Thammasiri *et al.*, Shaleena *et al.*, López textitet al., Pradeep *et al.*, Hu *et al.*, Zafra *et al.*, Kotsiantis *et al.*, Nandeshwar *et al.* ,Dursun Delen, Tamhane *et al.*, Guruler *et al.*, Dejaeger *et al.*, Ashutosh *et al.*, Gogaa *et al.*, Guptaa *et al.*, Kasih *et al.*, Márquez-Vera *et al.*, Tomida *et al.*,Baradwaj *et al.*,Yadav *et al.*, Adhatrao *et al.*, Abu-Oda *et al.*, Regha *et al.*, Márquez-Vera *et al.*, Pradeep *et al.*,Guarín *et al.*,Ahmad *et al.*,Iam-On *et al.*,Dekker *et al.*,Yukselturk *et al.*,Pal *et al.*,Asif *et al.*,Dragicevic *et al.*,Thammasiri *et al.*,Kovacic *et al.*,Jeevalatha *et al.*,Romero *et al.*,AL-Malaise *et al.*,Ahmed *et al.*
Neural Network	Thammasiri *et al.*, Shaleena *et al.*, Dursun Delen,Tekin, Lykourentzou *et al.*, Huang *et al.*, Dejaeger *et al.*, Gogaa *et al.*, Ramaswami *et al.*, Martinho *et al.*, Yukselturk *et al.*, Asif *et al.*, Thammasiri *et al.*, Romero *et al.*
Fuzzy	Lykourentzou *et al.*, Martinho *et al.*
SVM	*Thammasiri et al.*, Zafra *et al.*, Kotsiantis *et al.*, Dursun Delen, Gayner *et al.*, Lykourentzou *et al.*, Huang *et al.*, Juan *et al.*, Dejaeger *et al.*, Rahman *et al.*, Ifenthaler *et al.* , Márquez-Vera *et al.*, Zimmermann *et al.*, Thammasiri *et al.*
SVM	*Thammasiri et al.*, Zafra *et al.*, Kotsiantis *et al.*, Dursun Delen, Gayner *et al.*, Lykourentzou *et al.*, Huang *et al.*, Juan *et al.*, Dejaeger *et al.*, Rahman *et al.*, Ifenthaler *et al.*, Márquez-Vera *et al.*, Zimmermann *et al.*, Thammasiri *et al.*
NaiveBayes	Shaleena *et al.*, López *et al.*, Pradeep *et al.*, Zafra *et al.*, Kotsiantis *et al.*, Nandeshwar *et al.*, Tamhane *et al.*, Márquez-Vera *et al.*, Ramaswami *et al.*, Pal *et al.*, Abu-Oda *et al.*, Guarín *et al.*, Ahmad *et al.*, Iam-On *et al.*, Yukselturk *et al.*,Asif *et al.*
LR	Dursun Delen, Tamhane *et al.*, Thammasiri *et al.*
NN	Zafra *et al.*, Kotsiantis *et al.*,Iam-On *et al.*, Yukselturk *et al.*,Asif *et al.*,Romero *et al.*
GA Bayes Net	Yukselturk *et al.*, Nandeshwar *et al.*, Gayner *et al.*, Huang *et al.*, Dekker *et al.*, Romero *et al.*,
Boost	Hu *et al.*, Zafra *et al.*, Kotsiantis *et al.*, Dursun Delen, Romero et al., AL-Malaise et al.
Bagging	Zafra *et al.*, Dursun Delen, Gogaa *et al.*, Márquez-Vera *et al.*

SVM - Suport Vector Machine, LR - Logistic Regression, NN - Nearest Neighbour

Tabela 4.3: *Resumo das técnicas de seleção de características utilizadas como filtros e que foram localizadas nos artigos expostos nesta RSL*

Filtros	*Artigos*
GiniIndex	Asif et al.
Information Gain	Asif et al., Ramaswami et al.,Gulati, Nandeshwar et al., Márquez-Vera et al., Osmanbegovic et al.
CfsSubsetEval	Pradeep et al., Ramaswami et al., Hina, Nandeshwar et al, Márquez-Vera et al.
ChiSquared-AttributeEval	Pradeep et al., Ramaswami et al., Márquez-Vera et al.,Osmanbegovic et al.
Consistency-SubsetEval	Pradeep et al., Márquez-Vera et al.
Filtered-AttributeEval	Pradeep et al., Márquez-Vera et al.
GainRatio-AttributeEval	Asif et al., Pradeep et al., Ramaswami et al., Hina, Márquez-Vera et al., Osmanbegovic et al.
OneRAttributeEval	Pradeep et al., Ramaswami et al., Nandeshwar et al., Márquez-Vera et al.,Osmanbegovic et al.
ReliefFAttributeEval	Pradeep et al, Ramaswami et al., Márquez-Vera et al.
SymmetricalUncertAttributeEval	Ramaswami et al., Márquez-Vera et al.
Non-negative Matrix Factorization Clustering based Feature Selection(NMFCFS)	Reghaand et al.

foi de Ioanna et. al [116] que utilizou uma combinação de 3 (três classificadores (FFNNs[2], SVMs[3], PESFAM[4]), obtendo 100% de acurácia na classificação. A menor acurácia apresentada foi no artigo de Yadav et al. [191] que utilizou o classificador CART, com 56,25% de acurácia. Em alguns artigos o autor não deixou claro qual o valor da acurácia retornada pelo classificador. Para as árvores de decisão a média da acurácia da classificação foi de 86,75%. Nos conjuntos com uma maior quantidade de dados acima de 15.000 linhas, verificou-se uma acurácia entre 93% e 94% por cento.

QP2 - Classificadores: Quais classes são usadas para a análise de desempenho e abandono?

A análise dos artigos mostrou que existem diferentes interpretações de desempenho e abandono. Pal [139], por exemplo, considera o abandono daqueles alunos que não renovaram matricula após o primeiro ano. Guarin et al. [58] analisam o abandono devido ao desempenho acadêmico, separando-os daqueles abandonos devidos a fatores não acadêmicos, como por exemplo problemas financeiros. O

[2]FFNNs - Feed-Forward Neural Networks
[3]SVMs - Support Vector Machines
[4]Probabilistic Ensemble Simplified Fuzzy ARTMAP

Tabela 4.4: *Abandono*

Tipo	Autores
Abandono no ensino superior	Stoessel et al., Abu-Oda & El-Halees, Pal, Amaya *et al.*, Sango-Diah *et al.*, Zhang *et al.*, Yadav *et al.*, Nandeshwar *et al.*, Guarín *et al.*, Kovacic
Abandono no ensino fundamental, médio ou tecnológico	Tomida & Yamaguchi, Márquez-Vera *et al.*, Martinho *et al.*
Abandono no início do curso	Dekker *et al.*, Delen, Thammasiri *et al.*, Lykourentzou *et al.*, Iam-On & Boongoen, Pal
Abandono em EAD	Gulati, Lara *et al.*,Almeida Neto *et al.*, Ahmad *et al.*, Yukselturk *et al.*, Lykourentzou et al., Rigo *et al.*

termo "retenção"também tem sido usado nos artigos onde a análise do abandono visa a retenção dos alunos no curso [199] [192] [130].

Alguns artigos tratam de abandono em disciplinas ou no curso como um todo, muitos focando no inicio do curso, sendo a análise feita para os diferentes níveis de ensino e para a Educação à Distância conforme Tabela 4.4. Artigos que tratam do abandono incluem os que tentam identificar as características relevantes na desistência do educando no início dos estudos como tratado por [36] [78] [174], e se a graduação foi concluída no tempo determinado [17] sem interrupção e de acordo com a quantidade de anos determinada para o curso. Iam-On et al.[82] propõem um framework de transformação de dados para melhorar a acurácia das previsões. Como estudo de caso, tentam predizer o abandono com dados disponíveis no início e no fim do primeiro ano. Jeevalatha et al. [91] tentam prever se a alocação de estágio para os alunos terá sucesso ou não. Todos os artigos analisados utilizaram classes binárias, indicando se o aluno abandonou ou não.

Já o desempenho do aluno visa predizer o resultado final (nota) de uma disciplina ou do curso como um todo como pode ser visto pela Tabela 4.5. Enquanto alguns utilizaram classes binárias (aprovado/reprovado), outros utilizaram classes mais numerosas, chegando a oito valores possíveis (excelente, muito bom, bom, regular, suficiente, ruim, muito ruim, não presente como relatado por Pradeep et al.[147]. Os artigos incluem cursos presenciais e ensino a distância.

Gupta et al.[61] propõem o uso de preditores, incluindo abandono e rendimento no processo de avaliação institucional.

QP3 - Preditores: Quais atributos são usados para predizer desempenho e abandono?

Tabela 4.5: *Desempenho*

Tipo Desempenho	Classes	Autores
Desempenho na graduação	Notas A,B,C,D,E	Asif *et al.*
Desempenho no ensino fundamental e média	Excellent,Very good,Good, Regular, Sufficient,Poor, Very Poor, Not Presented	Pradeep *et al.*
Desempenho baseado no perfil do aluno e no perfil de aprendizagem	study unit outcomes	Ifenthaler *et al.*
Desempenho no HighSchool	HSCGrade(marks/grade obtained at HSc Level)	Ramaswami & Bhaskaran
Reprovação em EAD	Sim, Não	Detoni & Araújo
Bom desempenho do aluno	Sim, Não	Regha & Rani
Desempenho com base na nota final	High, Medium, Low	AL-Malaise *et al.*
Desempenho na Graduação	Excellent, Good, Average	Ahmed & Elaraby
Desempenho na graduação	Nota Final	Al-Barrak & Al-Razgan
Sucesso na Graduação. Duas situações: sucesso com GPA >=2 e sucesso com GPA >= 3 (honors)	Sim, Não	Guruler *et al.*
Desempenho na disciplina Dynamics	Nota do Exame Final	Huang & Fang
Sucesso de alunos espanhois no Exame do PISA, onde sucesso é estar no percentil 25% superior	Sim, Não	Gorostiaga & Rojo-Álvarez
Bom desempenho do aluno	Sim, Não	Goga *et al.*
Passou no curso (Nota final)	Sim, Não	Márquez-Vera *et al.*
Desempenho no fim do semestre (ESM - End semester Marks)	First >= 60%, Second >= 45% and <60%, Third >= 36% and < 45%, Fail < 36%.	Yadav *et al.*
Desempenho com base no resultado final universitário (total university score) obtido pelo aluno.	excellent, very good, good, average and bad	Kabakchieva
Desempenho no fim do semestre (ESM - End semester Marks)	First >= 60%, Second >= 45% and <60%, Third >= 36% and < 45%, Fail < 36%.	Baradwaj & Pal
Sucesso do aluno com base na Media Global do Aluno (Grade Point Average); e tempo para completar o curso (time-to-degree).	Sim, Não	Dragicevic *et al.*
Passou no curso (Nota final)	Sim, Não	Osmanbegovic & Suljic
Resultado final do curso	1 - Extraordinary (Cum Laude); 2 - Very Satisfactory; 3 - Satisfactory	Kasih *et al.*
Passou no primeiro semestre	Sim, Não	Adhatrao *et al.*

Verifica-se uma tendência de usar todas as informações disponíveis como entrada de dados para os sistemas de classificação. O número de atributos nos artigos lidos variam entre 40 e 375. Porém estes atributos são peneirados por algoritmos de seleção, onde alguns utilizados pela RSL são apresentados na Tabela 4.3, ou pelos próprios algoritmos de classificação, onde apenas aqueles com maior grau de relevância são usados no modelo final. A análise dos atributos usados permite uma classificação dos mesmos em alguns grandes grupos:

(1) Dados demográficos (gênero, idade, residência, etc.)
(2) Dados sócio-econômicos (renda, grau de instrução dos pais, membros na família, se tem geladeira ou tv, etc.)
(3) Notas obtidas (em outras disciplinas do curso, em provas de admissão, em atividades, média geral do curso, etc.)
(4) Comportamento em Ambiente Virtual de Aprendizagem (número de acessos, participação em fórum, acesso a conteúdo, etc.)
(5) Motivação (obtida através de questionários ou testes).

Em vários casos se verificou o uso de pré-processamento dos dados antes da utilização dos mesmos como preditores. Muito do pré-processamento aplicado aos dados visa reduzir o número de valores associados aos atributos. Isto é principalmente verificado em atributos nominais, com várias opções, como é o caso de itens de questionários. No entanto, a definição do tipo de pré-processamento usado não segue regras, e fica a cargo do usuário do sistema, que define quais atributos devem ser trabalhados e o tipo de transformação a ser aplicada.
Também, não se verificou um conjunto de atributos que se destacam como bons preditores de desempenho e abandono de maneira geral. Isto se deve à grande variedade de informação usada e de características dos algoritmos de classificação, que em muitos casos funcionam como "caixa-preta", não permitindo a verificação dos atributos de fato usados.
QP4 - Ferramentas OpenSource: Quais ferramentas ou toolboxes OpenSource são utilizados para as tarefas de classificação de estudantes em Base de Dados Educacionais?
Na maior parte dos trabalhos os autores utilizaram toolboxes OpenSource para efetuarem as tarefas de classificação, pois estas fornecem vários algoritmos supervisionados e não-supervisionados. As toolboxes e softwares utilizados incluem: *Weka* [21], *RapidMiner*, *IBMSPSS Modeler*, *SPSS*, *MatLab*, *SqlServer*, *Clementine*, *R project* ,e *KEEL framework*. No entanto, *MATLAB*, *SPSS*, *SqlServer* e *Clementine* não se enquadram na categoria *OpenSource*. A toolbox mais utilizada foi a *Weka* que foi encontrada em 22 trabalhos, seguida pela *RapidMiner* em 5 trabalhos, o *KEEL* e *R project* em 1 trabalho cada.

Weka, *RapidMiner* e *KEEL*, utilizam interface gráfica e implementam vários classificadores, sendo que o *KEEL* e o *RapidMiner* são baseados no *Weka* e escritos na linguagem *Java*. Porém não é necessário conhecimento prévio dessa linguagem para utilização das ferramentas. O *Weka* além de interface gráfica fornece um ambiente para execução de linha de comando e de modelagem via Workflow, como também aceita plugins que podem integrar toolboxes escritas em outras linguagens de programação como o *Scikit-learn* escrito em *Python* e o *MLP* escrito em *R*. Já o *R project* fornece apenas a interface via linha de comando e para utilizá-lo é preciso ter conhecimento mínimo dos comandos que devem ser utilizados.
Nesta RSL não foram retornados trabalhos utilizando as toolboxes *Scikit-learn* [141] e *Orange*, o que pode ser uma oportunidade para trabalhos futuros.

CAPÍTULO 5

Dados

Os dados utilizados neste trabalho referem-se aos alunos do curso de Ciências da Computação da Universidade Federal de Goiás, e são provenientes de duas fontes: CS-UFG (Centro de Seleção da Universidade Federal de Goiás) e CERCOMP-UFG (Centro de Recursos Computacionais da Universidade Federal de Goiás). Os dados foram disponibilizados, em 2013/2, após solicitação formal, com anuência do Instituto de Informática. Por motivos de segurança da informação, não foi possível obter acesso direto ao banco de dados, sendo os dados fornecidos via planilhas excel. Apesar de atualizações terem sido solicitadas mais recentemente, estas ainda não foram fornecidas.

Vale ressaltar que em 2008 foi aprovado um novo currículo para o curso de Ciências da Computação com mudanças significativas no projeto pedagógico. Além disto, em 2007 o governo criou o Plano de Reestruturação e Expansão das Universidades Federais - decreto 6.096/2007, com o objetivo de ampliar o número de vagas na graduação nas Universidades Federais. O Instituto de Informática, como parte deste esforço, criou uma nova turma do curso de Ciências da Computação (CC), com 40 alunos, passando a ter duas entradas por ano a partir de 2009. Isto implicou na realização de dois vestibulares por ano para prover estas vagas. Dada a mudança no projeto pedagógico optou-se por trabalhar com dados de alunos que ingressaram depois de 2008.

Além disto, no período de 2009 a 2013, o processo de seleção na Universidade Federal de Goiás sofreu diversas modificações. Em 2011, a UFG definiu que os candidatos que fizeram as provas do Exame Nacional do Ensino Médio (Enem) poderiam aproveitar as notas das provas de conhecimentos gerais na primeira fase do Vestibular. Em 2012, o Ministério da Educação (MEC) criou o Sistema de Seleção Unificada (SiSU), um sistema informatizado, no qual instituições públicas de ensino superior oferecem vagas para candidatos participantes do Exame Nacional do Ensino Médio (Enem). Com a instituição do SiSU, em 2012/1 a UFG passou a oferecer 20% das vagas em seus cursos através desse sistema. Em 2014/2 foi realizado o último vestibular na UFG. A partir de 2015 o sistema de ingresso passou a ser

Variável	*Tipo*	*Tamanho*	*Descrição*	*Intervalo*
id	Inteiro	-	Identificador no vestibular	0 - 999999
curso	Alfanumérico	4	Identificador do curso	(A110,A131,A152)
lingua_portuguesa	Inteiro	2	Nota de Portugues	0 a 10
biologia	Inteiro	2	Nota de Biologia	0 a 10
fisica	Inteiro	2	Nota de Física	0 a 10
literatura_brasileira	Inteiro	2	Nota de Literatura	0 a 10
matematica	Inteiro	2	Nota de Matemática	0 a 10
historia	Inteiro	2	Nota de História	0 a 10
geografia	Inteiro	2	Nota de Geografia	0 a 10
quimica	Inteiro	2	Nota de Química	0 a 10
lingua_estrangeira	Inteiro	2	Nota de Ling.Est.	0 a 10
sexo	Texto	1	Sexo do aluno	M,F
nascimento	Alfanumérico	10	Data de Nascimento	-
enem	Inteiro	5	Nota no Exame Nacional do Ensino Médio	0,000 a 100,000

Tabela 5.1: *Tabela com informações do vestibular e do Exame Nacional do Ensino Médio(ENEM)*

exclusivamente através do SiSU.

O CS-UFG forneceu os dados referentes aos candidatos do vestibular aos cursos do Instituto de Informática dos anos de 2009 a 2013 conforme Tabela 5.1, separados por semestre. A base de dados incluem dados dos candidatos, do curso selecionado (C. da Computação (A110), Engenharia de Software (A131) ou Sistemas de Informação (A152)), dos resultados dos candidatos nas provas, assim como as respostas do questionário socio-econômico aplicado pela CS-UFG. Quando disponível também forneceram a nota do candidato no Exame Nacional do Ensino Médio (ENEM). Os dados dos anos anteriores não se encontravam em formato digitalizado e portanto não foram disponibilizados. Isto implica que os alunos que ingressaram antes de 2009/2, assim como aqueles que entraram através do SiSU a partir de 2012, não possuem dados do vestibular. Além disto, apesar de manter grande parte do questionário sócio-econômico aplicado ao longo dos anos, algumas poucas modificações foram feitas. Para poder acompanhar estas modificações nos questionários, uma cópia dos mesmos foi fornecida pelo CS-UFG.

O CERCOMP-UFG forneceu os dados dos alunos de Ciências da Computação que ingressaram a partir de 2008, armazenados no sistema acadêmico da UFG. O arquivo, com 391 alunos, contém variáveis demográficas, como sexo, e data de nascimento; resultados globais, como nota global e nota do curso; e informações do ingresso, como ação afirmativa e percentagem de conclusão do curso. A lista completa encontra-se na Tabela 5.3.

Variável	*Tipo/Tamanho*	*Descrição*	*Intervalo*
codigo	inteiro(4)	codigo da disciplina	154, 162, 168, 185, 4201, 4202, 4204
disciplina	texto(31)	descrição da disciplina	ED1,ED2,LM,PC1,PC2,POO,TC
matricula	inteiro(5)	número da matrícula	-
ano_oferta	inteiro(4)	ano da oferta do curso	2008 a 2013
semestre_oferta	inteiro(1)	semestre em que o curso foi ofertado	1 a 2
nota	inteiro(2)	Nota do aluno em determinada disciplina	0 a 10
frequencia	inteiro(2)	Frequência do aluno em determinada disciplina	0 a 64
situacao	texto(3)	Situação do aluno no final de cada semestre	valores:'APV','REP','REF'

ED1-Estrutura de Dados1, ED2-Estrutura de Dados2, LM-Lógica Matemática PC1-Programação de computadores1, PC2-Programação de Computadores2 TC-Teoria da Computação1, POO-Programação Orientada a Objetos

Tabela 5.2: *Tabela com dados das notas, frequência e situação dos alunos em cada disciplina (desempenho dos alunos)*

Além disto, o CERCOMP-UFG também disponibilizou um arquivo com informações referentes às disciplinas cursadas pelos alunos com informações relacionadas ao ano/sem no qual a disciplina foi cursada, nota, frequência e situação final (reprovado por falta (REF), aprovado (APV), reprovado (REP)). A estrutura deste arquivo encontra-se na Tabela 5.2. Foram disponibilizadas somente as disciplinas do curso de Ciências da Computação ligadas à área de programação de computadores: programação de computadores 1 (pc1), programação de computadores 2 (pc2), programação orientada a objetos (poo), lógica matemática (lm), teoria da computação (tc), estrutura de dados 1 (ed1), estrutura de dados 2 (ed2).

Observa-se que as disciplinas Programação de Computadores 1 e 2, junto com Estrutura de Dados 1 e 2 formam o núcleo principal de programação, onde os conceitos básicos de algoritmos são apresentados.São objetos naturais para análises temporais. Programação Orientada a Objetos é um paradigma diferente e pode ser usado para comparações nas análises. Lógica Matemática e Teoria da Computação, junto com Programação de Computadores 1, são disciplinas com alto nível de reprovação e abandono, e baixa média global, o que as tornam objetos naturais para análise.

Variável	*Tipo/Tamanho*	*Descrição*	*Intervalo*
curso	texto(22)	Nome do Curso	tipo único:CIÊNCIAS DA COMPUTAÇÃO
turno	texto(8)	turno do curso	tipo único:integral
modalidade	texto(10)	modalidade do curso	único registro:presencial
grau_acadêmico	texto(11)	Grau a ser obtido	único registro:BACHARELADO
matricula	inteiro(22)	Número da matrícula	0 - 999999
ano_ingresso	inteiro(4)	Ano em que ingressou no curso	2008 - 2013
semestre_ingresso	inteiro(2)	Semestre em que ingressou no curso	1 - 2
forma_ingresso	texto(55)	Como o aluno ingressou no curso	valores:'INGRESSO POR PROCESSO SELETIVO', 'INGRESSO POR TRANSFERÊNCIA',...'INGRESSO PORTADOR DE DIPLOMA'
acao_afirmativa	texto(19)	Ação para ingresso	valores:'Não','DC - Renda Inf','DC - Renda Sup','Escola Pública','Negro Escola Pública'
dt_nascimento	texto(10)	Quando o aluno nasceu	-
cidade_nascimento	texto(24)	Cidade onde o aluno nasceu	-
uf_nascimento	texto(2)	Estado em que o aluno Nasceu	-
sexo	texto(9)	Sexo do aluno	valores: MASCULINO,FEMININO
uf	texto(2)	Unidade da Federação	Código da UF
media_global_aluno	decimal(3)	Média do aluno durante o curso	variando de 0 a 10
media_global_curso	decimal(3)	Média das notas dos alunos no curso	variando de 0 a 10
percentual_integralizacao	texto(3)	Percentual de integração	0 - 100%
situacao_vinculo	texto(12)	Situação do aluno no curso	valores:Desvinculado, Trancado, Vinculado
ano_desvinculo	inteiro(4)	Ano em que o aluno saiu do curso	Nulo, 0 a 2013
semestre_desvínculo	inteiro(1)	Semestre em que o aluno saiu do curso	Nulo,0 a 2013
dt_desvinculacao	alfanumérico(19)	Data em que saiu do curso	Nulo e dados com data e hora
motivo_desvinculacao	inteiro(64)	Se o aluno saiu do curso	valores:Nulo, CURRICULO INTEGRALIZADO...OPÇÃO POR OUTRO CURSO

Tabela 5.3: *Dados pessoais dos alunos*

5.1 Tratamento dos Dados

Em cada uma das bases de dados foi efetuada uma verificação de consistência, abrindo cada planilha no Microsoft Excel e analisando a situação dos dados. Neste caso foi uma tarefa manual, porém espera-se no futuro automatizar este processo. Foram encontradas inconsistências na forma de preenchimento dos dados como por exemplo, CURRICULO INTEGRALIZAO em vez de CURRICULO INTEGRALIZADO, ou nos nomes das cidades algumas estavam acentuadas e outras não. Para facilitar a utilização dos registros, decidimos retirar os acentos dos nomes das cidades.

Para facilitar a manipulação dos dados, optou-se por carregar as informações em um banco de dados SQLite que suporta até 140TB [1] de dados para efetuar operações de consulta SQL utilizando o padrão ANSI-SQL, o que facilita uma futura migração para um outro SGBD (Sistema Gerenciador de Banco de Dados) em ambiente de produção.

Após a importação dos arquivos xls para o banco de dados foi preciso efetuar uma padronização nos nomes das variáveis, que estavam com nome maiúsculo e com espaços para os casos com nome composto, variáveis com ponto, acentuação, a primeira letra Maiúscula. A padronização considerou apenas nomes em letra minúscula, sem acentuação e para os nomes compostos utilizamos o underline no lugar do espaço em branco.

Além dos atributos originais dos arquivos fornecidos, outros atributos foram incluídos para facilitar a organização no banco de dados, como o ano e o semestre referentes ao arquivo para que esta informação não fosse perdida no processo.

No caso dos questionários, foi feita uma análise antes da importação para a base de dados, a fim de alinhar as perguntas de todos os anos. Apesar da grande maioria das perguntas se repetirem ao longo dos anos, algumas foram incluídas ou retiradas. A tabela de questionário contém todas as questões que apareceram em pelo menos um questionário. Quando a pergunta não fazia parte do questionário daquele ano, ela teve o valor nulo atribuído.

5.2 Organização dos dados

A organização dos dados no banco de dados possibilitou definir *views* (perspectivas do banco de dados) que extraem e formatam os dados que serão utilizados nos diver-

[1] https://www.sqlite.org/limits.html

sos processos de análise, seja na análise estatística ou na execução dos algoritmos de mineração de dados.

Ao definir as *views*, é possível guardar as consultas na base de dados para que sejam repetidas para gerar as tabelas desejadas a partir dos dados na base por meio de uma consulta SQL simplificada. Assim, caso novos dados sejam acrescentados na base, a geração das tabelas fica fácil. A definição de *views* também permite a manipulação dos dados, efetuando agrupamentos, transformações e criação de novos tipos de dados.

Inicialmente foi criada uma tabela completa, contendo todos os atributos existentes no banco de dados. Para cada um dos 391 alunos matriculados no curso foi gerada uma linha na tabela contendo os dados demográficos (contidos no primeiro arquivo fornecido pelo CERCOMP-UFG), os dados do vestibular daquele aluno incluindo o questionario sócio econômico (fornecido pelo CS-UFG), e as disciplinas cursadas pelos alunos.

Visto que um aluno pode cursar a mesma disciplina mais de uma vez, foi preciso fazer uma transformação para linearizar as disciplinas. Pode-se visualizar este processo como a criação de colunas na tabela contendo as informações relevantes para cada disciplina cursada. Seria o equivalente a transformar as linhas da tabela desempenho em colunas. Como houve caso de aluno cursar a mesma disciplina quatro vezes, este processo torna-se inviável. Assim optou-se por um resumo, para cada disciplina foram acrescentados na tabela informações a respeito da primeira vez que o aluno cursou a disciplina (ano, semestre, nota, frequência, conceito, idade), e a vez em que foi aprovado (ano, semestre, nota, frequência, conceito, idade), além de um contador para informar quantas vezes o aluno cursou aquela disciplina. Caso tenha sido aprovado na primeira vez que cursou, os dados da primeira e da aprovação encontram-se repetidos, e o contador de vezes será igual a um. Observa-se que os dados referentes às vezes intermediárias não são usados. Para calcular a idade do aluno no ano/semestre em que cursou a disciplina definiu-se uma função usando a data de nascimento do aluno.

No caso dos questionários sócio-econômicos observamos que as respostas de cada pergunta estavam no formato numérico e as colunas estavam com nomes que não explicavam o dado da coluna. Assim foi necessário criar uma view para efetuar a tradução das respostas numéricas de acordo com as respostas do questionário, e para nomear as colunas com termos legíveis e que explicam de fato o conteúdo da coluna. Por exemplo, a questão 2 que estava com o nome da coluna q2 foi renomeado para estado_civil e as respostas que variavam de 0 a 5 foram rotuladas respectivamente para solteiro, casado, separado, viúvo, divorciado e outros. Dessa forma a view renomeou todas as colunas e todas as respostas do questionário.

O processo de ETL (Extract Transform and Load) foi feito utilizando a linguagem *Python* com a biblioteca de análise de dados *Pandas*. Por meio da criação de um *script* foi possível conectar à base de dados existente, extrair os dados que seriam transformados por meio de uma consulta *SQL*, efetuar operações de agrupamento, transposição de colunas, aplicações de funções, gerando dessa forma uma tabela geral do banco de dados. O *script* com o código fonte utilizado para estas transformações encontra-se no Apêndice A. Este processo gerou uma tabela com 113 atributos e com 391 linhas.

Após a análise desta tabela geral, foram definidas as classes para o processo de predição. Optou-se por definir 3 tipos de predição: *abandono*, que possui duas classes: sim e não; a *situação geral* que possui duas classes: positiva e negativa; e *situação por disciplina*, que possui 3 classes: APV (aprovado), REF (Reprovado por Falta), REP (Reprovado por nota).

Os valores da variável *abandono* foram definidos para cada aluno utilizando a regra:

- Se o percentual de integralização do curso for igual a 100% e a situação vínculo for igual a *desvinculado* então o aluno não abandonou o curso e a variável *abandono* recebe o valor *não*.
- Se a situação for igual a *vinculado* então o aluno não abandonou e a variável *abandono* recebe o valor *não*.
- Se o percentual de integralização do curso for inferior a 100% e a situação vínculo for igual a *desvinculado* então o aluno abandonou o curso e a variável *abandono* recebe o valor *sim*.
- Se a situação vínculo for igual a *trancado* então o aluno *abandonou* e a variável recebe o valor *sim*.

O valor *positiva* foi atribuido à variável *situacao_geral* caso a *media_global_aluno* seja maior ou igual a *media_global_curso*, caso contrário o valor da variável *situacao_geral* será *negativa*.

A variável *situacao_por_disciplina* (poo, ed1, ed2, pc1, pc2, lm) já existe na base de dados e denota o conceito recebido pelo aluno na primeira vez que cursou determinada disciplina. Pode receber os valores aprovado, reprovado por falta ou reprovado por nota. O conceito é definido pela Resolução Cepec nº 1122 da UFG, que determina que se o aluno obtiver nota final igual ou superior a seis vírgula zero (6,0) e frequência igual ou superior a setenta e cinco por cento (75%) da carga horária da disciplina então o aluno será aprovado na disciplina (APV), se a frequência for inferior a que 75% então será reprovado por falta (REF) independente da nota. E se tiver uma nota inferior a seis vírgula zero (6,0) com frequência igual ou superior a setenta e cinco por cento, será reprovado por nota (REP).

CAPÍTULO 6

Perfil dos alunos

Antes de aplicar as técnicas de mineração de dados, utilizando os algoritmos de classificação a fim de gerar regras para melhor compreender o processo de abandono do curso e da performance dos alunos, é interessante e conveniente entender e conhecer os dados que serão usados no processo, traçando um perfil dos alunos do curso de Ciências da Computação da UFG. Nesta investigação utilizaremos a amostra composta de 391 estudantes da população de alunos de graduação conforme dados descritos no Capitulo V.

Antes de iniciar a análise, vale observar que existem variáveis com mais de 15% das instâncias sem valor atribuído (missing data). Quando estes dados vêm da base do Vestibular, a razão pode ser porque o aluno entrou no curso por outra forma de ingresso além do vestibular, ou não prestou ENEM, ou porque não respondeu o questionário em parte ou no todo. Quando vêm das disciplinas cursadas, isto é de se esperar visto que existem alunos em diversas fases do curso, e portanto, existem aqueles que ainda não cursaram as disciplinas de semestres mais adiantados. A Tabela 6.1 mostra a quantidade de registros de alunos em cada categoria por ano. Na categoria "outra forma de ingresso"aparece o total de alunos que entraram no curso por outra forma de seleção distinta do Vestibular, incluindo transferências, ordem judicial e SiSU. Na categoria "Ação Afirm", consta o total de alunos que ingressaram através de cotas alocadas a alunos de escola pública e negros (Figura 6.1).

Análise dos 391 estudantes por gênero mostrou que 90% (351) são do gênero masculino e 10% (40) do feminino, conforme esperado para um curso de computação [184]. A Figura 6.2 ilustra o resultado obtido. A Tabela 6.2 mostra a distribuição por ano de ingresso.

De uma perspectiva etária Tabela 6.3, verifica-se que os alunos de Ciência da Computação são majoritariamente jovens, sendo que 46% (179) dos estudantes que ingressaram no curso tinham menos de 18 anos, e 95% menos que 24 anos. Apenas 5 alunos do total ingressaram com mais de 24 anos, isto é, 1% pertence à faixa-etária de 28 a 38 anos, dos quais 1,02% (4) são do gênero masculino e 0,25% (1)

Ano	*Total alunos*	*Outra forma de ingresso*	*Ação Afirmativa*	*Vestibular*	*Enem*	*Questionário*
2008	39	5	0	8	5	6
2009	78	10	16	35	34	39
2010	73	0	12	68	62	66
2011	79	3	17	72	69	72
2012	73	8	14	69	68	69
2013	49	17	7	44	41	38
Total	391	43	66	296	279	290

Tabela 6.1: *Total registros na base de dados por ano*

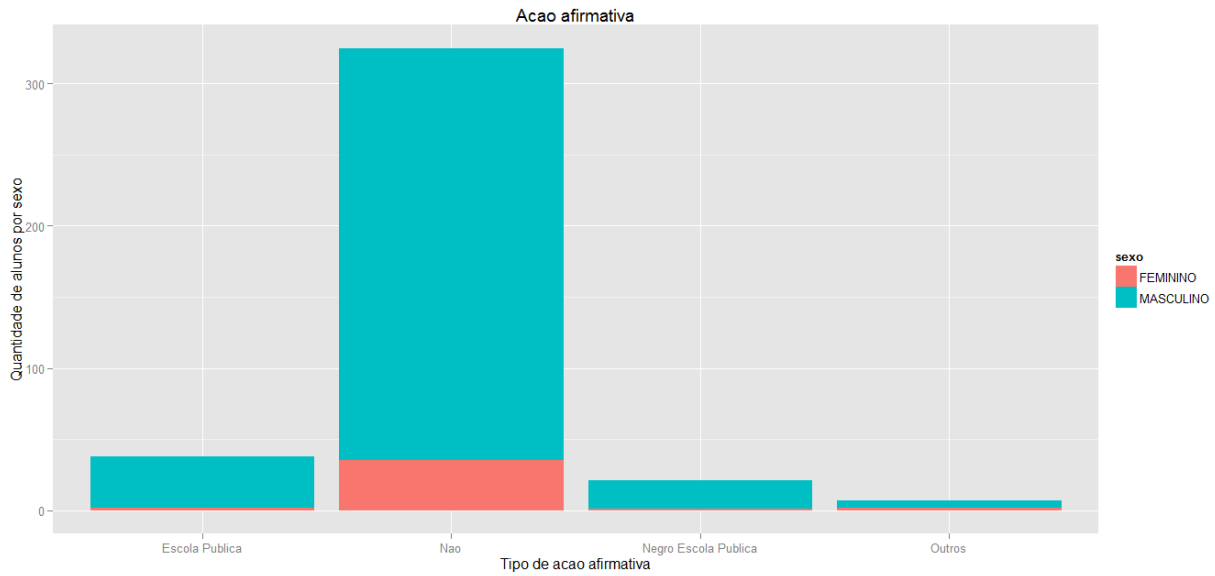

Figura 6.1: *Distribuição na ação afirmativa por gênero*

Ano	*Feminino*	*Masculino*	*Total*
2008	2	37	39
2009	12	66	78
2010	6	67	73
2011	9	70	79
2012	5	68	73
2013	6	43	49

Tabela 6.2: *Distribuição por ano de ingresso, por gênero*

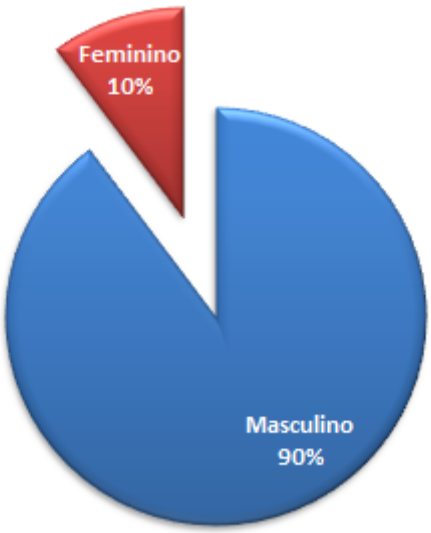

Figura 6.2: *Alunos por sexo*

Faixa_Etária	*Homens*		*Mulheres*		*Total*	
	%Perc	*Amostra*	*%Perc*	*Amostra*	*%Perc*	*Amostra*
15 a 18	40,15%	157	5,6%	22	46%	179
18 a 24	45%	**176**	4%	16	49%	**192**
24 a 28	3,6%	14	0,25%	1	4%	15
28 a 38	1,02%	4	0,25%	1	1%	5
Total	89,77%	351	10,23%	40	100%	391

Tabela 6.3: *Tabela da quantidade de alunos do gênero masculino e feminino pela faixa etária.*

do feminino.

Para entender a distribuição dos estudantes por classe usada na mineração de dados, foi feita uma análise dos dados para cada uma das predições propostas. Verificou-se que 74,68% (292) dos estudantes não abandonaram o curso. Isto quer dizer que se formaram ou estão matriculados no curso. 25,32% (99) estudantes abandonaram o curso, isto é, desistiram do curso ou estão com a matricula trancada. A Tabela 6.5 apresenta os dados de forma mais detalhada.

O estado civil predominante foi o de solteiros com 284 (duzentos e oitenta e quatro), sendo 5 (cinco) casados e 1 (um) aluno se enquadrou como outros Figura 6.3. A Tabela 6.4 mostra a distribuição dos alunos por raça. Visto que estes dados fazem parte do questionário do Vestibular, esta informação não estava disponível para todos os estudantes.

A partir da Tabela 6.6 é possível observar que para o desempenho geral do aluno existe um equilíbrio entre as duas classes pela própria construção da classe, lembrando que o aluno tem uma situação positiva se sua média geral no curso é maior ou igual à media geral da turma, e negativa caso contrário.

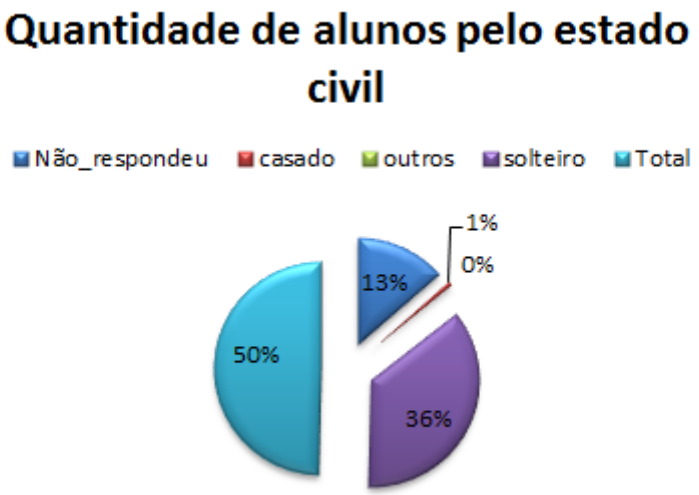

Figura 6.3: *Estado civil de acordo como o questionário sócio-econômico considerando os alunos no período de 2008 a 2013.*

Raça	*Frequencia*	*%*
amarelo	18	6.23%
branco	150	51.90%
pardo	89	30.80%
negro	32	11.07%
Total	289	100%

Tabela 6.4: *Distribuição por Raça*

Abandono	*Homens*		*Mulheres*		*Total*	
	%Perc	*Amostra*	*%Perc*	*Amostra*	*%Perc*	*Amostra*
nao	67,52%	**264**	7,16%	28	74,68%	**292**
sim	22,25%	87	3,07%	12	25,32%	99
Total	89,77%	351	10,23%	40	100%	391

Tabela 6.5: *Distribuição de alunos por gênero de acordo com o abandono*

Desempenho	*Homens*		*Mulheres*		*Total*	
	%Perc	Amostra	%Perc	Amostra	%Perc	Amostra
negativa	45,78%	**179**	4,09%	16	49,87%	**195**
positiva	43,99%	172	6,14%	24	50,12%	196
Total	89,77%	351	10,23%	40	100%	391

Tabela 6.6: *Distribuição de alunos por gênero de acordo com a situação geral no curso*

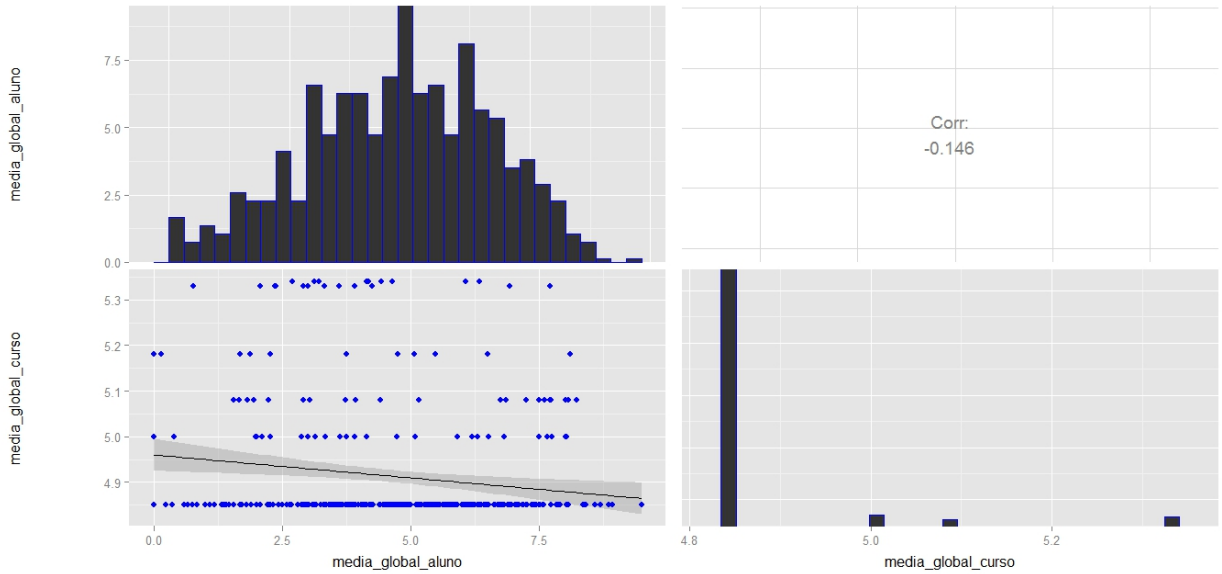

Figura 6.4: *Correlação entre a média global do curso e a média global do aluno.*

Na análise do desempenho por disciplina foram utilizados os dados das disciplinas dos semestres anteriores para prever o desempenho de uma dada disciplina. Como pode ser observado na Tabela 6.7 verifica-se uma diminuição progressiva do número de registros a medida que aumenta o semestre como esperado.

	Quantidade	*Media*	*DesvPad*	*Min*	*Max*	*Qtd_APR*	*Qtd_REP*	*Qtd_REF*
pc1_nota_primeira	315	5,33	2,67	0,00	10,00	250	36	29
lm_nota_primeira	324	5,31	2,79	0,00	10,00	229	47	48
pc2_nota_primeira	238	6,23	2,55	0,00	10,00	210	18	10
poo_nota_primeira	210	5,16	2,83	0,00	10,00	167	18	25
ed1_nota_primeira	209	4,92	2,66	0,00	9,60	169	17	23
ed2_nota_primeira	156	6,00	2,26	0,00	10,00	139	10	7
tc_nota_primeira	70	5,30	2,53	0,00	10,00	65	1	4

Tabela 6.7: *Resumo descritivo com as notas da primeira vez que um aluno cursou determinada disciplina.*

Verificar-se pela Figura 6.5 a existência de uma forte correlação entre a nota da primeira vez que o aluno cursou programação de computadores I (pc1) e a média global do aluno, porém isso não quer dizer que a nota de pc1 é um indicador de causa para a média global do aluno.

A existência de uma forte correlação entre a nota da primeira vez que o aluno cursou a disciplina pc1 e lógica matemática (lm) pode ser observada pela Figura 6.6. Para os demais casos a correlação vai de moderada até desprezível.

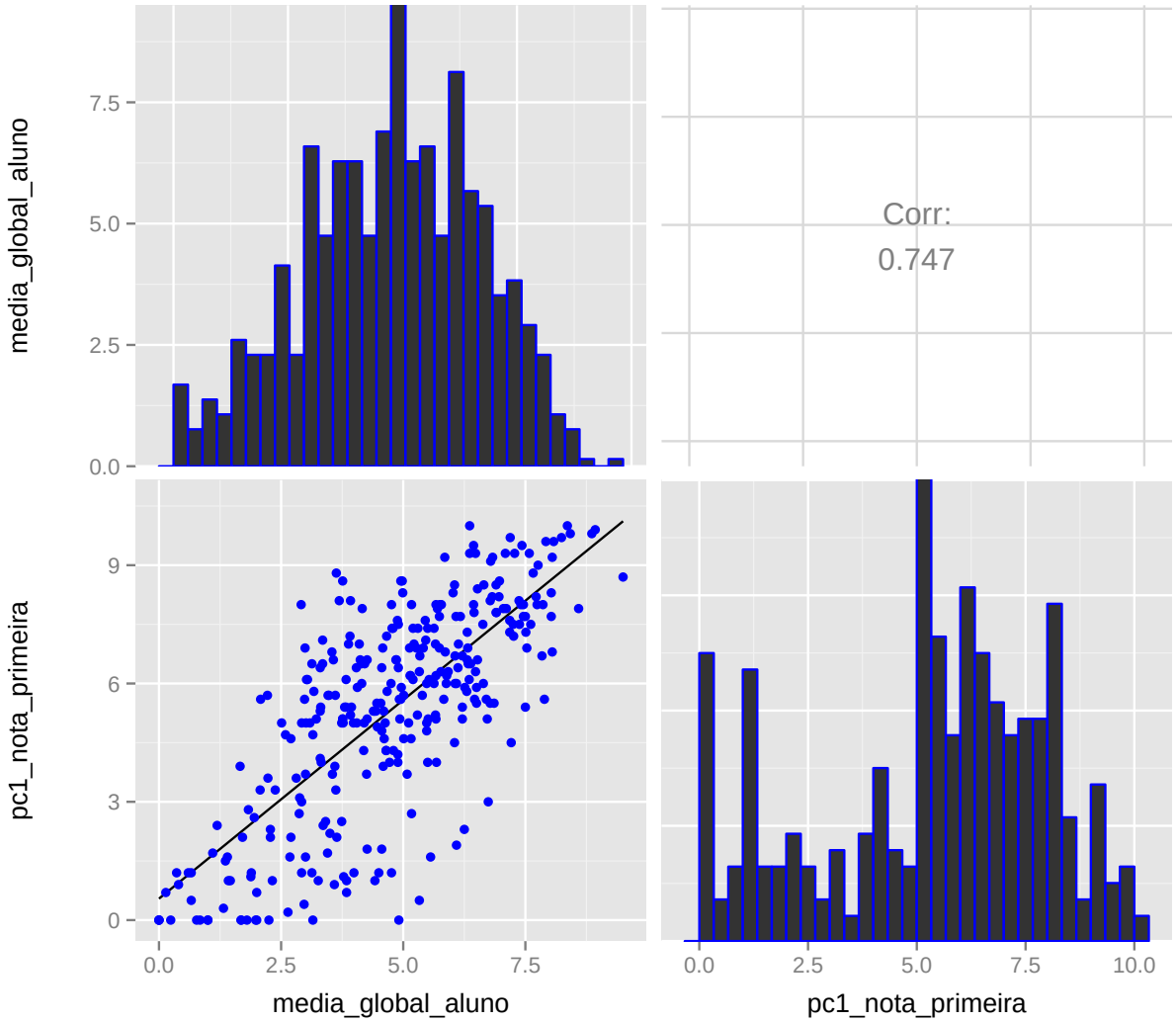

Figura 6.5: *Correlação entre a nota da primeira vez que o aluno cursou pc1 e a média global do aluno*

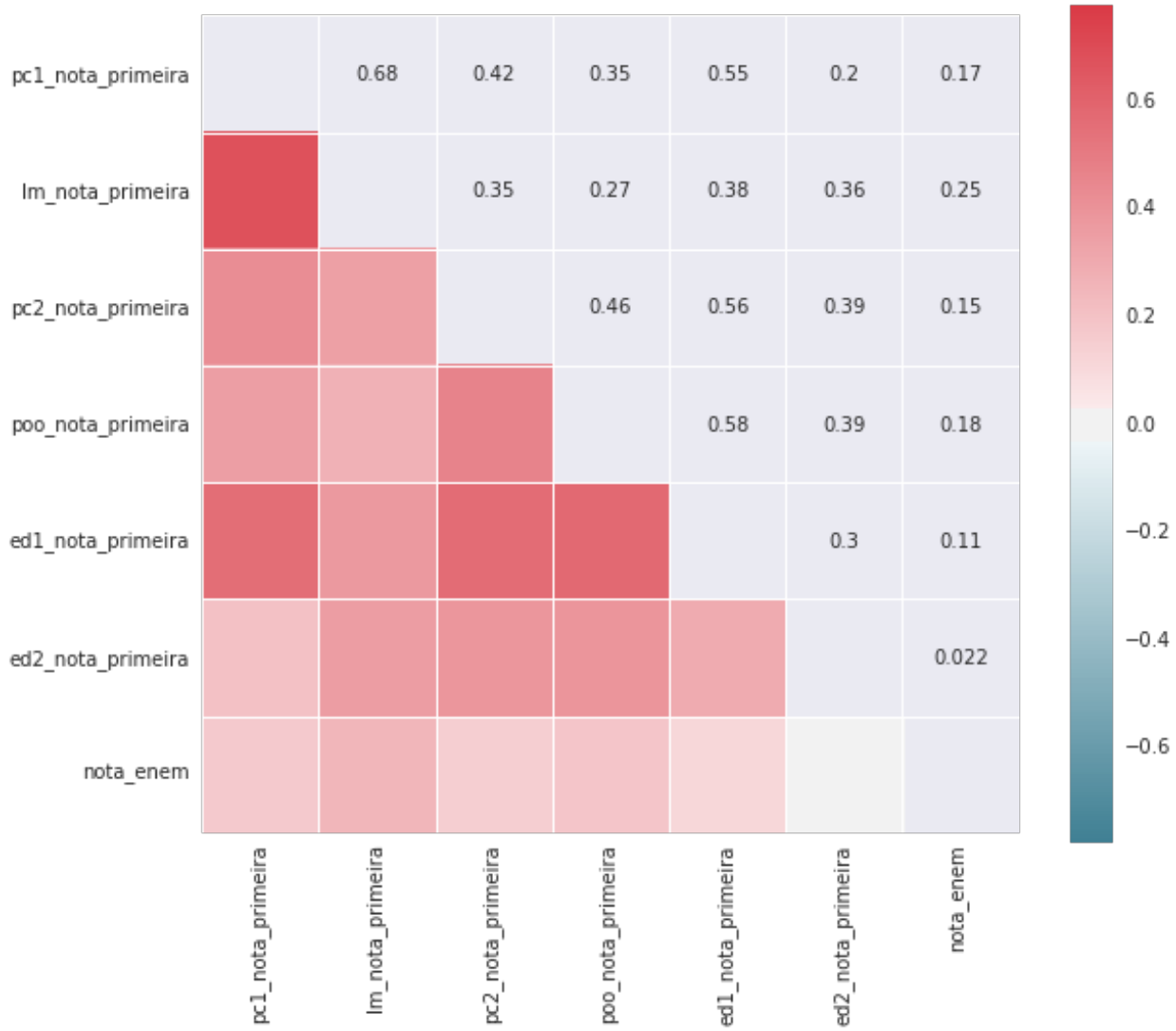

Figura 6.6: *Correlação entre as notas da primeira vez que os alunos cursaram as disciplinas e a nota do enem*

CAPÍTULO 7

Mineração dos Dados da UFG

Os experimentos conduzidos foram baseados nos artigos relatados na RSL, onde foram utilizadas técnicas variadas de filtragem, balanceamento, além de diferentes algoritmos de classificação para dados educacionais. O universo considerado para os experimentos possui 391 (trezentos e noventa e um) *objetos* e 113 (cento e treze) *características*.

Serão efetuadas 3 análises: *abandono*, *situação-geral* e *situação por disciplina*, onde serão utilizadas características que abrangem os dados de desempenho dos alunos nas disciplinas, do vestibular, do questionário sócio-econômico e pessoais dos alunos como sexo, idade, etc.

Para o abandono, teremos a classe binária abandono (*sim* ou *não*), para a situação-geral a classe binária situação (*positiva*, *negativa*) e para a situação por disciplina a classe não-binária (*APV*, *REF*, *REP*).

Para comparação das medidas de desempenho serão utilizadas 10 (dez) execuções da validação cruzada com 10 (dez) partições, além disso o teste de significância estatística baseado no *teste-t de Student* para auxiliar na comparação dos resultados dos classificadores.

Nos estudos estatísticos é importante salientar que nem sempre um resultado de uma relação estatisticamente significante estabelecida entre duas variáveis implica que a variável explicativa provoca mudança na variável de resposta.

Os resultados dos ensaios vão gerar vários números que não poderão ser resumidos usando a média e nem mesmo o desvio padrão, pois mesmo que o resultados destas medidas sejam diferentes, por meio do teste de significância estatística será possível responder que amostras retiradas de uma mesma população não possuem diferenças estatisticamente significativas.

Dessa forma o teste de significância será utilizado para dar sentido às diferenças entre os resultados dos classificadores executados várias vezes com vários números aleatórios produzidos pela validação cruzada. Por esta razão vamos utilizar este teste pois será preciso afirmações mais precisas e maior rigor no que diz respeito aos resultados.

Existem vários testes de significância como o *Teste z*, *Teste Qui-Quadrado* (teste X^2), *Teste F*, porém não é o foco comparar os diferentes testes estatísticos, o que pode ser utilizado para trabalhos futuros.
Para o *Teste de Correção de Emparelhados* ou *Teste de Significância* serão utilizados os parâmetros padrões fornecidos pelo WEE, 5 (cinco) bases de dados, 10 (dez) execuções com um nível de confiança de 0.05.

7.1 Abandono

A fim de selecionar quais classificadores utilizar na análise de dados, foi realizado um levantamento, por meio da RSL, para verificar quais classificadores são utilizados para mineração de dados educacionais. Além disso, procura-se por um classificador que retorne um bom desempenho, um resultado de fácil compreensão das regras utilizadas para a classificação e que efetue a filtragem do conjunto de características.
Dessa forma para verificar quais algoritmos classificadores melhor se adaptam ao contexto estudado foi feita uma comparação de diversos classificadores. Para os experimentos foram selecionados os classificadores *LDA, QDA, NB, RL, SVM, CART, RF, AB, KNN* da toolbox *Scikit-learn*, *J48 (C4.5)* e *MLP* do *Weka*. Utilizamos o *WEE* (Weka Experiment Environment) em conjunto com o plugin para utilizar a toolbox *Scikit-learn* desenvolvido em *Python*.
Na primeira etapa de comparação foram usados os dados da base que classifica os alunos de acordo com o abandono no final do curso. Efetuamos a importação do arquivo *abandono.csv* para o *WE* (Weka Explorer) a fim de gerar vários arquivos para serem utilizados no WEE utilizando ferramentas para o balanceamento dos arquivos e seleção de atributos. Assim, os seguintes arquivos foram gerados:
A base *abandono* com as 113 *características* e 391 *objetos* sem nenhum tipo de filtragem dos dados.
Neste arquivo foi aplicado o filtro supervisionado de instância *SMOTE* com os parâmetros padrões (-C 0 –K 5 –P 100.0 –S 1, onde C = 0) para detectar de forma automática a classe minoritária não vazia que será utilizada pelo *SMOTE* (K =5 utiliza cinco vizinhos mais próximos para efetuar a interpolação dos atributos para a geração dos exemplos sintéticos, P=100 para criar 100% de instâncias por meio do *SMOTE*, S=1 o valor utilizado para a geração da amostragem aleatória) para a geração de novos exemplos sintéticos para a classe minoritária por meio da interpolação entre eles a fim de obter o balanceamento dos dados, gerando o arquivo *abandono-s* com 113 *características* e 490 *objetos*.
Utilizando a base *abandono-s*, foi aplicado o filtro supervisionado para a seleção

de atributos chamado *CfsSubsetEval* (Correlation-based Feature Subset Selection for Machine Learning) com os parâmetros P-1 –E1, número de threads e tamanho do pool do thread que pode ser o número de cores do CPU respectivamente e para o avaliador e para os parâmetros de busca foi utilizado o BestFirst (Melhor Primeiro) para procurar o espaço de subconjuntos de atributos por meio de subidas gananciosas por meio de um mecanismo de retrocesso com –D 1 que considera o tamanho máximo do laço do cache dos subconjuntos avaliados e N 5 para o número consecutivo de nós utilizados antes do fim do processo de busca. Dessa forma o novo conjunto de dados está constituído de 490 *objetos* e 23 *características*.
A base *abandono-r* foi gerada a partir da base abandono após a aplicação do filtro supervisionado *Resample* para a produção de uma subamostragem aleatória do conjunto de dados sem substituição, com o Bias para a classe uniforme igual a -B 0.0 com –S 1 para o valor do número aleatório e para a subamostragem com –Z 100.0 para o tamanho do subconjunto como 100% do conjunto original. Não houve variação no número de objetos e características.
Em seguida foi aplicado o filtro CfsSubsetEval gerando a base *abandono-rcfs*, com 391 *objetos* e 9 *características*.

Tabela 7.1: *Teste de Correção de emparelhados para a precisão*

Dataset	(10)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
abandono	0.81	0.81	0.83	0.82	0.82	0.84	0.75 •	0.86 ○	**0.90** ○	0.84
abandono-r	0.88	0.91	0.92 ○	0.85 •	0.84 •	0.91	0.88	0.92 ○	**0.93** ○	0.92 ○
abandono-rcfs	0.86	0.91 ○	0.89 ○	0.87	0.86	0.84	0.82	0.87	0.86	**0.92** ○
abandono-s	0.83	0.81	0.84	0.82	0.84	0.85	0.75 •	**0.86**	0.85	0.84
abandono-scfs	0.81	0.80	**0.84**	**0.84**	0.83	**0.84**	0.78 •	0.83	0.80	0.83

○, • Diferença estatisticamente significativa ou não
(1)-RandomForest (2) - AdaBoost (3) - SVM (4) - KNN (5) - LogisticRegression
(6) - QDA (7)-LDA (8)-GaussianNB (9) - Cart (10) - J48

O *J48* foi selecionado como base para a seleção por se tratar de um classificador que usa árvore de decisão e por ter sido o mais utilizado na mineração de dados educacionais conforme especificado na RSL, e pelo fato de ser um classificador que exibe as regras utilizadas na classificação, um classificador da linha caixa branca, demonstrando quais as características que foram de fato utilizadas para a predição.
A partir do classificador *J48* como base para a comparação e utilizando o conjunto de dados *abandono*, verificamos que o classificador *NB* retornou o maior valor de precisão, sendo que o *QDA* (6) demonstrou possuir um resultado estatisticamente pior que o *J48*, e os algoritmos (7 e 8) retornaram resultados estatisticamente melhores que o *J48* ao nível de significância especificado. Já os demais algoritmos (1,2,3,4,5 e 9) não apresentaram diferença estatística significativa conforme Tabela 7.1.

Para a base *abandono-r* que utiliza o Resample para o balanceamento dos dados, de acordo com a Tabela 7.1 o NB continua retornando o maior valor para a precisão. O pior valor foi retornado pelo classificador (4), seguido do (3) que são piores ao nível de significância estatística, e os algoritmos (2), (7), (8) e (9) retornaram resultados estatisticamente melhores que o J48 (10) com nível de 5% de significância.
Os classificadores (1), (5), (6) não apresentaram diferenças estatisticamente significativas de acordo com o teste de correção para amostras emparelhadas para a precisão.
Considerando a base *abandono-rcfs* observa-se que o preditor (9) Decision Tree Classifier - CART retornou maior valor para a precisão, e que seguido do (1) e (2) apresentaram resultado significativamente melhor que o (10). Já os preditores (3), (4), (5), (7) e (8) não apresentaram diferenças estatísticas significativas em relação ao *J48*.
Olhando pra a base *abandono-s* que que utiliza o *SMOTE*, foi possível verificar que o classificador (7) retornou o melhor resultado, já o (6) retornou pior resultado. Não houve diferenças estatisticamente significativas entre o (10) e os algoritmos (1), (2), (3), (4), (5), (7), (8) e (9).
O QDA classificou de forma imprecisa os dados da base *abandono-scfs* e foi estatisticamente pior. Não houve diferenças estatísticas significativas entre o (10) e os demais classificadores, apesar dos classificadores (2), (3) , (5) demonstrarem maiores valores para a precisão em relação ao *J48*.

Tabela 7.2: *Teste de Correção de emparelhados para o percentual de classificados corretamente (Acurácia)*

Dataset	(10)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
abandono	78.13	77.80	76.45	**79.59**	76.13	78.29	74.94	78.70	59.67 •	76.09
abandono-r	86.49	**90.13** ○	89.51	83.70	78.58 •	88.00	89.41	86.98	66.68 •	88.08
abandono-rcfs	85.01	**90.89** ○	86.47	82.37	81.45	79.00 •	77.85 •	81.23 •	81.10 •	90.69 ○
abandono-s	83.06	82.45	81.39	84.00	78.47 •	**84.02**	79.39 •	82.92	73.86 •	79.65
abandono-scfs	82.84	81.47	82.12	**84.98**	83.35	83.69	76.33 •	83.73	82.39	79.63

○, • Diferença estatisticamente significativa ou não
(1) - RandomForest (2) - AdaBoost (3) - SVM (4) - Knn (5) - LogisticRegression
(6) - QDA (7) - LDA (8) - NaiveBayes (9) - Cart (10) - J48

Considerando o dataset *abandono* foi possível observar que o classificador (8) possui a menor acurácia, porém a medida que balanceamentos de dados e filtros vão sendo aplicados o resultado os resultados dos outros datasets vão melhorando, contudo a diferença estatística continua.
Já o classificador (3) retornou a melhor acurácia, e nenhum algoritmos retornou resultado estatisticamente melhor que o *J48* ao nível de significância especificado.
Para a base *abandono-r* que utiliza o *Resample* para o balanceamento dos dados,

de acordo com a Tabela 7.2 o classificador (1) retornou a melhor acurácia. Já a pior foi retornada pelo classificador (8), seguido do (4) e o algoritmo (1) retornou o resultado estatisticamente melhor que o J48 (10) com nível de 5% de significância. Os classificadores (2), (3), (5), (6), (7) e (9) não apresentaram diferenças estatisticamente significativas de acordo com o teste de correção para amostras emparelhadas para a acurácia.

Considerando a base *abandono-rcfs* observa-se que o preditor (1) retornou a pior acurácia, seguido do (9) apresentaram resultado significativamente melhor que o (10). Já os preditores (2), (3), (4) não apresentaram diferenças estatísticas significativas em relação ao *J48*, porém os preditores (5), (6), (7) e (8) apresentaram diferenças estatísticas significativas.

Olhando pra a base *abandono-s* que que utiliza o *SMOTE*, não houve diferenças estatisticamente significativas entre o algoritmo (10) e os (1), (2), (3), (5), (7) e (9), sendo que os classificadores (4), (6) e (8) mostraram diferença estatística significativa.

O *QDA* demostrou a mais baixa acurácia com diferença significativa a um nível de significância de 5% pior que o *J48* (10) utilizando a base *abandono-scfs*. Os demais classificadores não apresentaram diferenças significativas.

Tabela 7.3: *Teste de Correção de emparelhados para os classificados incorretamente*

Dataset	(10)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
abandono	21.87	22.20	23.55	20.41	23.87	21.71	25.06	21.30	40.33 ○	23.91
abandono-r	13.51	9.87 ●	10.49	16.30	21.42 ○	12.00	10.59	13.02	33.32 ○	11.92
abandono-rcfs	14.99	9.11 ●	13.53	17.63	18.55	21.00 ○	22.15 ○	18.77 ○	18.90 ○	9.31 ●
abandono-s	16.94	17.55	18.61	16.00	21.53 ○	15.98	20.61 ○	17.08	26.14 ○	20.35
abandono-scfs	17.16	18.53	17.88	15.02	16.65	16.31	23.67 ○	16.27	17.61	20.37

○, ● Diferença estatisticamente significativa ou não
(1) - RandomForest (2) - AdaBoost (3) - SVM (4) - Knn (5) - LogisticRegression
(6) - QDA (7) - LDA (8) - Naive Bayes (9) - Cart (10) - J48

Os valores classificados de forma incorreta são exibidos na Tabela 7.3 considerando os percentuais por base de dados e por algoritmos de classificação, além disso exibe a significância estatística a partir do *J48*, que foi escolhido como base para a comparação, em relação aos demais classificadores.

É possível notar que para o dataset *abandono* o classificador (8) retornou a maior taxa de classificados incorretamente com diferenças estatísticas significativas. Os demais classificadores não mostraram diferenças estatísticas significativas para a taxa classificados incorretamente.

Para a base *abandono-r* que utiliza o *Resample* para o balanceamento dos dados, de acordo com a Tabela 7.3 o classificador (8) retornou a maior taxa de classificados

incorretamente, seguido do classificador (4). A menor taxa foi retorna pelo classificador (1), e os demais classificadores não retornaram resultados com diferenças estatísticas significativas em relação ao *J48* (10) com nível de 5% de significância.
Para a base *abandono-r* que utiliza o *Resample* para o balanceamento dos dados, de acordo com a Tabela 7.3 o classificador (8) retornou a maior taxa de classificados incorretamente, seguido do classificador (4). A menor taxa foi retorna pelo classificador (1), e os demais classificadores não retornaram resultados com diferenças estatísticas significativas em relação ao *J48* (10) com nível de 5% de significância.
Considerando a base *abandono-rcfs* observa-se que os preditores (5),(6) e (8) retornaram as maiores taxas de classificados incorretamente e resultados com significância estatística melhor que o do preditor (10). Já os preditores (1) e (3), são significativamente piores que o classificador (10), porém os demais preditores (2), (3), (4) não apresentaram diferenças estatísticas significativas.
Olhando pra a base abandono-s que que utiliza o *SMOTE*, verificou-se que a base (8), (6) e (4) apresentaram resultados com nível de 5% de significância estatística melhores que o algoritmo *J48* (10). Os demais não apresentaram diferenças estatísticas significativas.
O classificador (6) retornou a melhor significância estatística em relação aos classificados incorretamente considerando a base abandono-scfs. Os demais não demonstraram diferenças estatísticas significativas.

Tabela 7.4: *Teste de Correção de emparelhados para Area_under_ROC (AUC)*

Dataset	(10)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
abandono	0.71	0.79 ○	0.80 ○	0.66	0.75	**0.81** ○	0.50 ●	0.78	0.76	0.68
abandono-r	0.86	**0.94** ○	0.91 ○	0.72 ●	0.82	0.89	0.79	0.87	0.81	0.84
abandono-rcfs	0.81	**0.94** ○	0.89 ○	0.75 ●	0.84	0.85	0.68 ●	0.86	0.82	0.90 ○
abandono-s	0.84	0.89 ○	**0.90** ○	0.82	0.87	**0.90** ○	0.75 ●	0.88 ○	0.86	0.79 ●
abandono-scfs	0.84	0.88 ○	0.89 ○	0.83	0.87	**0.90** ○	0.81	**0.90** ○	0.86	0.79 ●

○, ● Diferença estatisticamente significativa ou não
(1) - RandomForest (2) - AdaBoost (3) - SVM (4) - Knn (5) - LogisticRegression
(6) - QDA (7) - LDA (8) - Naive Bayes (9) - Cart (10) - J48

Para os valores do *AUC* (Area Under Curve) a Tabela 7.4 exibe os resultados que variam de 0 a 1, por base de dados e por algoritmos de classificação, além disso exibe a significância estatística a partir do *J48* , que foi escolhido como base para a comparação, em relação aos demais classificadores. Quanto maior o valor do *AUC* melhor, indicando se o classificador efetua a classificação de forma aleatória ou não.
Os classificadores (6),(3) para os dados abandono de acordo com o valor do *AUC* falharam ao efetuar a classificação. Considerando as bases de dados *abandono-rcfs* e *abandono-s* o resultado com *AUC* que resulta em falha na classificação foi

retornado pelo classificador (6), já para as bases *abandono-s* e *abandono-scfs* o classificador (9) se enquadrou como Bom na tarefa de classificação. O melhor classificador para o arquivo abandono foi o classificador *Regressão Logística* (5) com *AUC* de 0,90 se enquadrando como Excelente , para as bases *abandono-r* e *abandono-rcfs* o *AdaBoost* (2) foi o melhor classificador e assim por diante.
Aplicando os filtros e algoritmos de balanceamento é possível perceber uma alteração no valor da *AUC*, e dessa forma, no resultado final retornado pelos classificadores. Dessa forma os classificadores com melhores resultados formam o *Random Forest*, *Logistic Regression*, *AdaBoost* e *LDA*.
Assim, para o estudo do abandono é possível utilizar o classificador *J48* que retornou bons resultados de desempenho e que possui algumas características que são interessantes retornando o conjunto de regras da classificação, gerando uma árvore de decisão que facilita na interpretação da classificação, além de utilizar o ganho de informação para efetuar a filtragem dos dados. Dessa forma o *J48* será utilizado para a classificação de abandono geral a fim de verificar e analisar os resultados retornados por ele.

7.1.1 Abandono com J48

Para os experimentos considerando o Abandono utilizamos o arquivo csv exportado da base de dados chamado *abandono.csv*, que foi importado por meio do *Weka Explorer* (WE) para efetuar a geração de arquivo no formato arff e para aplicação de técnicas de balanceamento (*SMOTE* e *Resample*) e de filtragem de dados utilizando o filtro (csf). Os arquivos gerados foram: *abandono-r*, *abandono-s*, e *abandono-rcfs* e *abandono-scfs*. O classificador *J48* foi executado considerando as configurações padrões e utilizando para a etapa de teste e de treinamento a validação cruzada com 10 partições.
A árvore de decisão gerada a partir da base *abandono* sem aplicação de balanceamento de dados e filtragem por meio de cfs, utilizando a filtragem por meio do ganho de informação é apresentada pela Figura 7.1, sendo que os resultados de ganho de informação são exibidos na Tabela 7.5 que utiliza como método de busca o ranking do atributo e característica de avaliação supervisionada com classe do tipo nominal com 113 características.
Apesar de qualquer valor de ganho de informação acima de zero exibir algum tipo de significado, limitamos a exibição do ganho das 15 melhores características classificadas no caso do abandono geral. Sendo que a Tabela 7.5 indica que pc1_situacao_primeira obteve o maior valor para o ganho de informação, e pela árvore de decisão da figura 7.1 foi possível observar que este atributo foi considerado

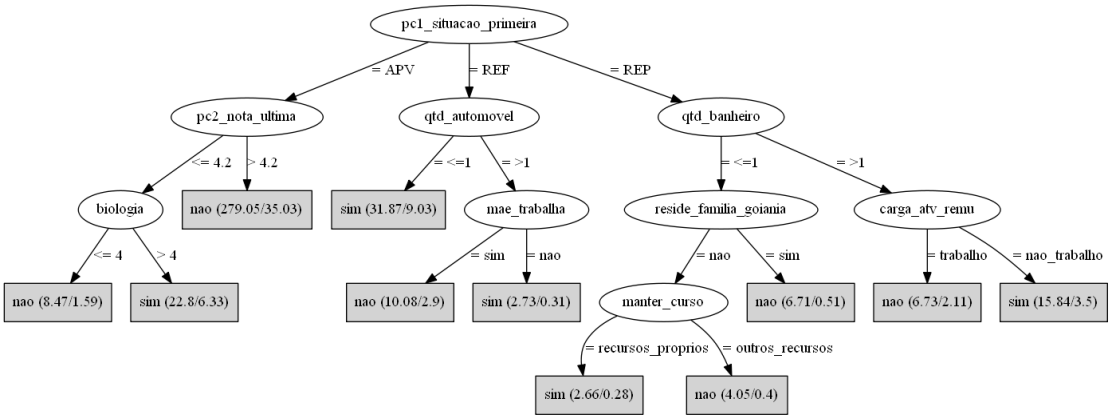

Figura 7.1: *Abandono sem balanceamento utilizando a base abandono*

Tabela 7.5: *Tabela com o resultado do filtro ganho de informação*

posição média	Ganho de Informação	Nr. Atributo	Descrição
1 +- 0	0.092 +- 0.007	15	pc1_situacao_primeira
2 +- 0	0.089 +- 0.006	18	pc1_nota_ultima
3.3 +- 0.46	0.063 +- 0.007	17	pc1_nota_primeira
5 +- 1.18	0.046 +- 0.007	24	pc2_nota_ultima
5.5 +- 1.91	0.047 +- 0.012	42	lm_nota_ultima
5.9 +- 1.3	0.042 +- 0.005	39	lm_situacao_primeira
7.1 +- 1.81	0.039 +- 0.009	23	pc2_nota_primeira
7.1 +- 1.3	0.039 +- 0.003	41	lm_nota_primeira
8.1 +- 0.83	0.034 +- 0.004	21	pc2_situacao_primeira
12 +- 2.28	0.013 +- 0.003	74	possui_renda_mensal
12.8 +- 2.52	0.012 +- 0.001	53	poo_nota_primeira
14.8 +- 5.42	0.011 +- 0.002	55	poo_qtd_cursou
15.1 +- 3.81	0.011 +- 0.002	35	ed2_nota_primeira
17 +- 4.1	0.01 +- 0.002	71	manter_curso
17.9 +- 3.3	0.01 +- 0.001	30	ed1_nota_ultima

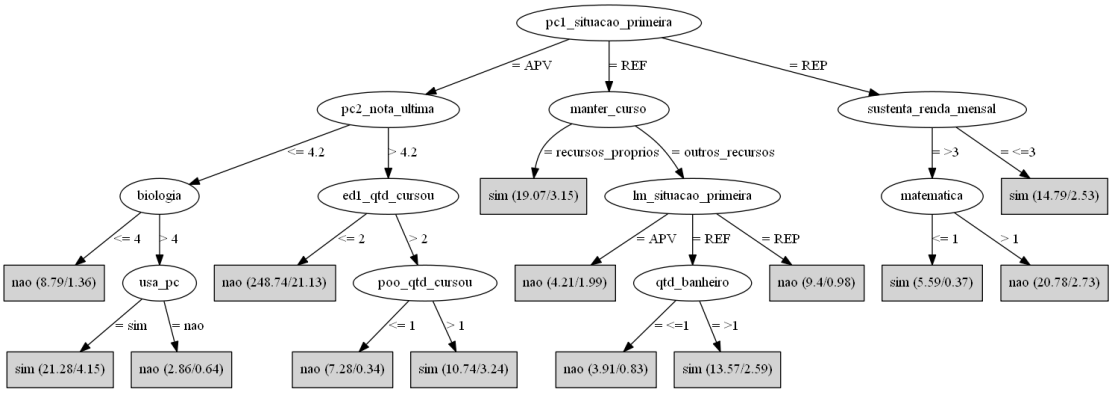

Figura 7.2: *Abandono com balanceamento de dados Resample utilizando a base abandono-r*

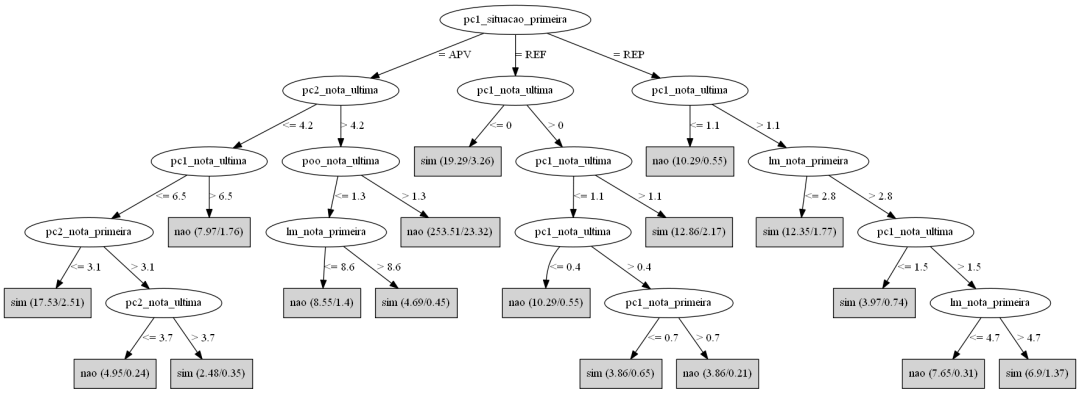

Figura 7.3: *Árvore de decisão do abandono geral com balanceamento de dados Resample e filtro cfs*

raiz.

Utilizando o Filtro Correlation Feature Selection (CFS) nos dados da base *abandono-r*, foi possível observar a geração da procura através de todas as combinações de características do conjunto de dados, criando um subconjunto de características que possuem boa capacidade preditiva. A Tabela 7.6, lista os atributos selecionados por este método.

Percebe-se ainda pela Tabela 7.6, que 8 (oito) atributos (pc1_situacao_primeira, pc1_nota_ultima,pc1_nota_primeira, pc2_nota_primeira, lm_nota_primeira,lm_nota_ultima, pc2_situacao_primeira,pc2_nota_ultima) são originários da tabela de desempenho de alunos, e 2 (atributos) atributos oriundos do questionário sócio-econômico, e não foram retornados dados demográficos ou do vestibular.

Observando as árvores das 7.2, 7.3, 7.4 e 7.5 percebe-se diferenças entre os formatos de cada árvore, com relação a quantidade de folhas e de ramos, onde a menor quantidade foi retornada pela figura 7.3 e a maior quantidade de folhas é gerada pela figura 7.1.

Para verificar o abandono a partir das árvore de decisão geradas é preciso observá-

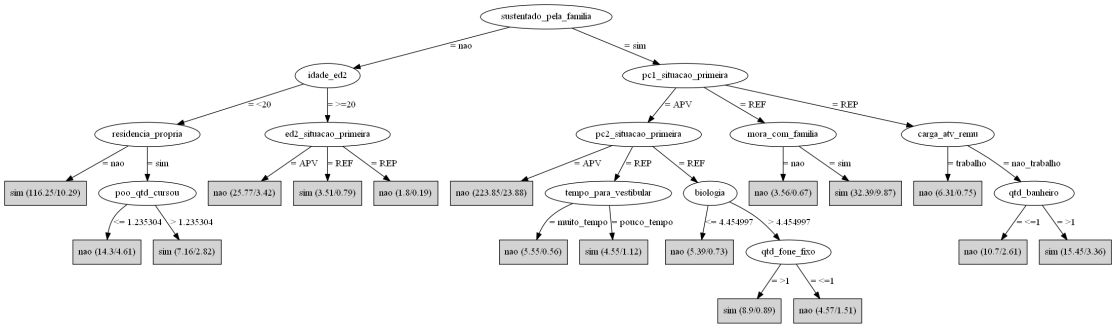

Figura 7.4: *Abandono com balanceamento de dados utilizando smote utilizando a base abandono-s*

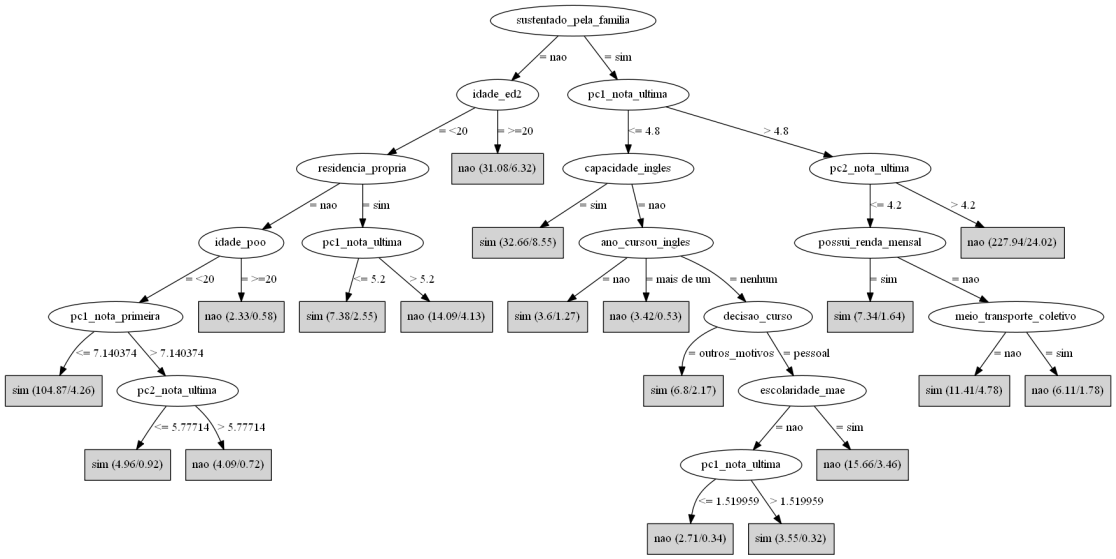

Figura 7.5: *Abandono com balanceamento de dados utilizando smote e com filtro cfs utilizando a base-scfs*

Tabela 7.6: *Ranking com as características após a aplicação do filtro CFS*

Posição	Nr. Atributo	Descrição	folds
1	15	pc1_situacao_primeira	10(100 %)
2	18	pc1_nota_ultima	10(100 %)
3	17	pc1_nota_primeira	8(80 %)
4	23	pc2_nota_primeira	8(80 %)
5	41	lm_nota_primeira	8(80 %)
6	42	lm_nota_ultima	7(70 %)
7	21	pc2_situacao_primeira	6(60 %)
8	24	pc2_nota_ultima	5(50 %)
9	74	possui_renda_mensal	4(40 %)
10	101	qtd_banheiro	3(30 %)

la de cima para baixo utilizando a estratégia (*top-down*), verificando os valores dos ramos no nó raiz e em cada nó até chegar no valor final que está na folha e retorno a classe sim ou não.

Para a escolha da raiz da árvore o algoritmo *J48(C4.5)* seleciona os atributos com ganho de informação com valores acima da média. O atributo que tiver a maior razão de ganho é selecionado como raiz, sendo a razão de ganho calculada dividindo o ganho pela entropia.

O atributo com o maior ganho de informação foi o pc1_situacao_primeira e dessa forma ele será escolhido como raiz da árvore de decisão. Os atributos do segundo nível tiveram um ganho de informação menor que o raiz porém maior que os demais e assim por diante.

Tabela 7.7: *Resultado da execução do classificador J48 para os arquivos de abandono*

base/media abandono	a b <–CM	Taxa VP	Taxa FP	Precisão	Recall	F-Measure	MCC	ROC Area	PRC Area	Classe	Acuracia	Tamanho Arvore	Nr. Folhas
	276 16 | a	0,945	0,667	0,807	0,945	0,871	0,366	0,702	0,834	nao	79%	20	11
	66 33 | b	0,333	0,055	0,673	0,333	0,446	0,366	0,701	0,487	sim			
	Weighted Avg.	0,79	0,512	0,773	0,79	0,763	0,366	0,702	0,746				
abandono-r													
	280 12 | a	0,959	0,404	0,875	0,959	0,915	0,626	0,86	0,929	nao	**87%**	25	14
	40 59 | b	0,596	0,041	0,831	0,596	0,694	0,626	0,86	0,712	sim			
	Weighted Avg.	0,867	0,312	0,864	0,867	0,859	0,626	0,86	0,874				
abandono-rcfs													
	278 14 | a	0,952	0,424	0,869	0,952	0,908	0,595	0,828	0,900	nao	86%	32	17
	42 57| b	0,576	0,048	0,803	0,576	0,671	0,595	0,828	0,698	sim			
	Weighted Avg.	0,857	0,329	0,852	0,857	0,848	0,595	0,828	0,849				
abandono-s													
	268 24 | a	0,918	0,313	0,812	0,918	0,862	0,633	0,832	0,797	nao	82%	30	17
	62 136 | b	0,687	0,082	0,85	0,687	0,76	0,633	0,832	0,85	sim			
	Weighted Avg.	0,824	0,22	0,827	0,824	0,821	0,633	0,832	0,818				
abandono-scfs													
	272 20 | a	0,932	0,308	0,817	0,932	0,87	0,656	0,829	0,8	nao	83%	34	18
	61 137 | b	0,692	0,068	0,873	0,692	0,772	0,656	0,83	0,847	sim			
	Weighted Avg.	0,835	0,211	0,839	0,835	0,831	0,656	0,829	0,819				

Pela tabela 7.7 percebe-se que a maior acurácia é retornada utilizando a base abandono-r que utiliza o Resample sem aplicação de filtros, apresentando a menor taxa de falso positivo para a classe minoritária e com maior AUC, sendo considerado um bom classificador.

A partir do exposto e analisando a árvore de decisão da Figura 7.2, e considerando o interesse na classe minoritária, ou seja, aquela que retorna o sim, que significa que o aluno vai abandonar o curso, foi possível extrair as regras descritas a seguir:

- Abandono-r

  Se (pc1_situacao_primeira = APV e pc2_nota_ultima <= 4.2 e biologia > 4 e usa_pc=sim) então = sim.

Se (pc1_situacao_primeira = APV e pc2_nota_ultima > 4.2 e ed1_qtd_cursou > 2 e poo_qtd_cursou > 1) então = sim.

Se (pc1_situacao_primeira = REF e manter_curso = recursos_proprios) então = sim.

Se (pc1_situacao_primeira = REF e manter_curso = outros_recursos_proprios e lm_situacao_primeira = REF e qtd_banheiro >= 1) então = sim.

Se (pc1_situacao_primeira = REP e sustenta_renda_mensal >= 3 e matematica <= 1) então = sim

Se (pc1_situacao_primeira = REP e sustenta_renda_mensal <= 3) então = sim

Analisando uma das regras extraídas é possível perceber um sentido lógico nos resultados, visto que um aluno que é reprovado em PC1 e que se mantém com recursos próprios abandona o curso. Viver com recursos próprios significa que o aluno vive de recursos provavelmente provenientes de seu trabalho, e portanto reprovar em PC1 implica em gastos maiores que as vezes ele não consegue bancar.

7.2 Situação Geral

A Situação Geral dos estudantes indica se ele terá um resultado positivo ou negativo no final do curso de bacharelado em Ciências da Computação. O resultado é considerado *positivo* se a média das notas do aluno é maior ou igual à media geral do curso. O resultado é considerado *negativo* se a média do aluno é menor que a media geral do curso. Vale ressaltar que a media geral do curso é calculada para cada turma de aluno ingressante no curso, e é composta da média das notas dos alunos daquela turma. Os experimentos a seguir serão efetuados considerando esta situação como classe. Serão utilizados os mesmos algoritmos mencionados na Seção 7.1 com as bases de dados *situacao-geral*, *situacao-geral-r*, *situacao-geral-rcfs*, *situacao-geral-s*, e *situacao-geral-scfs*, obtidas seguindo os mesmos procedimentos descritos para Abandono.

O *J48* foi selecionado como base para a seleção por se tratar de um classificador de árvore de decisão e por ter sido mais utilizado na mineração de dados educacionais conforme especificado na RSL, e pelo fato de ser um classificador que exibe as

Tabela 7.8: *Teste de Correção de emparelhados para a precisão*

Dataset	(10)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(11)
situacao-geral	0.79	0.77	0.81	0.82 ○	0.78	0.80	**0.95** ○	0.78	0.82	0.74	0.79
situacao-geral-r	0.83	0.88○	0.90 ○	0.88 ○	0.79 ●	0.90 ○	**0.99** ○	0.85	0.82	0.89 ○	0.91 ○
situacao-geral-rcfs	0.83	0.83	0.88 ○	0.87	0.88	0.82	0.74 ●	0.85	0.71 ●	**0.92** ○	0.89 ○
situacao-geral-s	0.82	0.78	0.81	0.88 ○	0.83	0.81	**0.91** ○	0.79	0.79	0.74 ●	0.80
situacao-geral-scfs	0.76	0.76	0.79	0.81 ○	0.75	**0.84** ○	0.76	0.82	0.77	0.70	0.76

○, ● Diferença estatisticamente significativa ou não
(1) - RandomForest (2) -AdaBoost (3) - SVM (4) - knn (5) - LogisticRegression
(6) - QDA (7) - LDA (8) - NaiveBayes (9) - Cart (10) - J48 (11) - MLP

regras utilizadas na classificação, um classificador da linha caixa branca, demonstrando quais as características que foram de fato utilizadas para a predição.

A partir do classificador *J48* como base para a comparação e utilizando o conjunto de dados *situacao-geral*, verificamos que o classificador *QDA* (6) retornou o maior valor de precisão em conjunto com o classificador (3) *SVM* (Suporte Vector Machine Classifier) também apresenta-se estatisticamente melhor que o J48 (10). Já os demais algoritmos (1, 2 ,3 ,4 ,5 ,6 ,8 ,9 e 11) não apresentaram diferença estatística significativa conforme Tabela 7.8 para a *precisão*.

Para a base *situacao-geral-r* que utiliza o *Resample* para o balanceamento dos dados, de acordo com a Tabela 7.8, verificou-se que o *QDA* (6), retornou o maior valor para a *precisão*, e os classificadores (1, 2,3 , 5, 6, 9 e 11) respectivamente apresentaram melhor resultado estatisticamente significativos em relação ao *J48* ao nível de significância de 5%. O classificador (4) apresentou o menor valor para precisão e estatisticamente pior ao nível de significância de 5% em relação ao *J48*. Não foi verificada diferença estatística significativa para a precisão considerando os classificadores (7 e 8).

Considerando a base *situacao-geral-rcfs* observa-se que o preditor (9) *Cart* retornou maior valor para a precisão, e que seguido do (11 e 2) apresentaram resultado significativamente melhor que o (10). Já os preditores (3), (4), (5), (7) e (11) não apresentaram diferenças estatísticas significativas em relação ao J48 no que diz respeito a precisão.

Olhando pra a base *situacao-geral-s* que que utiliza o SMOTE para o balanceamento dos dados, foi possível verificar que o classificador *QDA* (6) retornou o melhor resultado, seguido do classificador (3) que mostrou-se resultado estatisticamente significativa ao nível de significância de 5% melhor que o (10) para a precisão. Não houve diferenças estatisticamente significativas entre o *J48* (10) e os algoritmos (1), (2), (3), (4), (5), (7), (8) e (9) e (11).

Os classificadores (2), (5), (11) seguidos dos classificadores (3) e (4) apresentaram resultados estatisticamente significativos melhores que o *J48*, considerando a

base *situacao-geral-scfs*. Já os demais classificadores não demonstraram resultados estatisticamente significativos a um nível de significância de 5% para a precisão.
Foi possível então verificar pela Tabela 7.8 que no geral a melhor precisão foi apresentada pelo classificador QDA com 0.99.

Tabela 7.9: *Teste de Correção de emparelhados para a taxa de verdadeiro positivo*

Dataset	(10)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(11)
situacao-geral	0.85	0.82	0.79	0.82	0.86	0.82	0.49 •	0.83	**0.87**	0.76 •	0.80
situacao-geral-r	0.88	**0.94**	0.92	0.93	0.90	0.90	0.71 •	0.87	0.88	0.92	0.91
situacao-geral-rcfs	0.91	0.90	0.88	0.83 •	0.85 •	0.84 •	0.84	0.84 •	**0.95**	0.90	0.82 •
situacao-geral-s	0.73	0.84 ○	0.77	0.77	0.79	0.80	0.55 •	0.79	**0.88** ○	0.75	0.79
situacao-geral-scfs	0.58	0.70 ○	**0.72** ○	0.71 ○	0.69 ○	0.68 ○	0.69 ○	0.66 ○	0.73 ○	0.71 ○	0.71 ○

○, • Diferença estatisticamente significativa ou não
(1) - RandomForest (2) -AdaBoost (3) - SVM (4) - Knn (5) - LogisticRegression
(6) - QDA (7) - LDA (8) - NaiveBayes (9) - Cart (10) - J48 (11) - MLP

Em termos da taxa de *VP* que retorna o percentual de casos positivos que são classificados como verdadeiros, ou seja, os casos em que os alunos que vão abandonar o curso de fato abandonam, onde a Tabela 7.9 exibe estes percentuais por base de dados e por algoritmos de classificação, além disso exibe a significância estatística a partir do *J48*.
É possível notar que para o dataset *situacao-geral* o classificador (6) *QDA* retornou a mais baixa taxa de *VP*, ou seja ele tende a classificar os alunos que deveriam ter uma situação geral *positiva* em alunos que com situação geral *negativa*, porém a medida que balanceamentos de dados e filtros vão sendo aplicados o resultado do *QDA* vai melhorando, e em alguns casos a diferença estatística desaparece. O classificador (9) retornou diferença estatisticamente significativas pior que o *J48* ao nível de significância de 5%, os demais classificadores não demonstraram diferenças estatísticas significativas para a taxa de *VP*.
Para a base *situacao-geral-r* que utiliza o *Resample* para o balanceamento dos dados, de acordo com a Tabela 7.9 o classificador (6) retornou a pior taxa de *VP* e pior resultado estatisticamente significativo. Os demais classificadores não apresentaram resultados estatisticamente significativos.
Por meio da base *situacao-geral-rcfs* os algoritmos (3, 4, 5, 7 e 11) apresentaram resultados estatisticamente significativos piores que o *J48* ao nível de significância de 5%. Os demais classificadores não apresentaram resultados estatisticamente significativos.
Olhando pra a base *situacao-geral-s* que que utiliza o *SMOTE*, não houve diferenças estatisticamente significativas entre o algoritmo (10) e os algoritmos (2, 3, 4, 5, 7, 9 e 11) sendo que o classificador 6 mostrou a pior diferença estatística significativa a

um nível de significância de 5% em relação ao *J48* para a taxa de *VP*. Os algoritmos (1 e 8) foram considerados estatisticamente melhores que o *J48*.

Para a base de dados *situacao-geral-scfs*, todos os classificadores foram estatisticamente melhores que o J48 ao nível de significância de 5% considerando a taxa de verdadeiro positivo. Além disso o J48 retornou a pior taxa para a taxa de VP.

Tabela 7.10: *Teste de Correção de emparelhados para a taxa de falso positivo*

Dataset	(10)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(11)
situacao-geral	0.23	0.25	0.19	0.19	0.26	0.21	0.03	0.23	0.19	0.27	0.22
situacao-geral-r	0.20	0.13	0.11 •	0.13	0.25	0.10 •	0.01 •	0.16	0.19	0.12 •	0.10 •
situacao-geral-rcfs	0.19	0.20	0.12 •	0.13	0.12	0.19	0.31 ○	0.15	0.40 ○	0.08 •	0.10 •
situacao-geral-s	0.09	0.12	0.09	0.06 •	0.09	0.10	0.03 •	0.11	0.12	0.14 ○	0.10
situacao-geral-scfs	0.10	0.12	0.10 ○	0.08	0.12	0.07 ○	0.11	0.08	0.11	0.16 ○	0.12

○, • Diferença estatisticamente significativa ou não
(1) - RandomForest (2) - AdaBoost (3) - SVM (4) - Knn (5) - LogisticRegression
(6) - QDA (7) - LDA (8) - NaiveBayes (9) - Cart (10) - J48 (11) - MLP

Em termos da taxa de *FP* que retorna o percentual de casos positivos que são classificados como falso na Tabela 7.10, ou seja, os casos em que os alunos que obtiveram situação *negativa* e são classificados como alunos que obtiveram situação *positiva*, exibe estes percentuais por base de dados, por algoritmos de classificação, e pela significância estatística a partir do *J48*, que foi escolhido como base para a comparação, em relação aos demais classificadores por motivos expostos anteriormente.

Para o dataset *situacao-geral* não houve diferença significativa do J48 em relação aos demais classificadores.

Para a base *situacao-geral-r* que utiliza o *Resample* para o balanceamento dos dados, de acordo com a Tabela 7.10 os classificadores (2,5,6,9 e 10) apresentaram resultados estatisticamente piores que o J48 a um nível de significância de 5%. Porém quando se trata de falsos positivos, percebemos que quando menor a taxa melhor. Então para a base *situacao-geral-r* o classificador (6) retornou o melhor resultado para a taxa de *FP*, 0.01 ou seja quase zero.

Considerando a base *situacao-geral-rcfs* dois algoritmos (6 e 8) apresentaram resultados estatisticamente melhores que o J48,porém como dito anteriormente, para o caso do FP, quer dizer que estes algoritmos possuem taxas mais altas, ou seja, estão classificando os positivos como falsos. Os casos estatisticamente piores que o J48, algoritmos (2, 9 e 11) porém possuem menores taxas de FP.

Olhando pra a base *situacao-geral-s* que que utiliza o *SMOTE*, verificou-se que os algoritmos (3), e (6) apresentaram resultados com nível de 5% de significância estatística piores que o algoritmo *J48* (10). Os demais não apresentaram diferenças

estatísticas significativas, com exceção do algorimto (9).

Tem termos da taxa de *FP*, os classificadores (2, 5, e 9) retornaram resultados com significância estatística melhores que o J48. Os demais não demonstraram diferenças estatísticas significativas para a base de dados *situacao-geral-scfs*.

Tabela 7.11: *Teste de Correção de emparelhados para Area_under_ROC (AUC)*

Dataset	(10)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(11)
situacao-geral	0.85	0.90 ○	0.89 ○	0.82	0.88	0.91 ○	0.89	0.87	0.90 ○	0.76 ●	0.87
situacao-geral-r	0.92	0.97 ○	0.95	0.91	0.91	0.96 ○	0.85 ●	0.92	0.92	0.89	0.95
situacao-geral-rcfs	0.91	0.96 ○	0.95 ○	0.86 ●	0.93	0.94 ○	0.89	0.94 ○	0.94 ○	0.91	0.94
situacao-geral-s	0.92	0.95 ○	0.94	0.85 ●	0.93	0.95 ○	0.92	0.92	0.92	0.84 ●	0.92
situacao-geral-scfs	0.91	0.94 ○	0.94 ○	0.89	0.92	0.96 ○	0.94 ○	0.96 ○	0.94 ○	0.86 ●	0.95 ○

○, ● Diferença estatisticamente significativa ou não
(1) - RandomForest (2) - AdaBoost (3) - SVM (4) - Knn (5) - LogisticRegression
(6) - QDA (7) - LDA (8) - NaiveBayes (9) - Cart (10) - J48 (11) - MLP

Para os valores do AUC (Area Under Curve) a Tabela 7.11 exibe os resultados que variam de 0 a 1, por base de dados e por algoritmos de classificação, além disso exibe a significância estatística a partir do J48 , que foi escolhido como base para a comparação, em relação aos demais classificadores. Quanto maior o valor do AUC melhor, indicando se o classificador efetua a classifica de forma aleatória ou não. Os classificadores (6),(3) para os dados situacao-geral pelo AUC, sendo que falharam ao efetuar a classificação. Considerando os dados do arquivo situacao-geral-rcfs e abandono-s o resultado do AUC que resulta em falha na classificação foi retornado pelo classificador (6), já para as bases situacao-geral-s e situacao-geral-scfs o classificador (9) se enquadrou como Bom na tarefa de classificação. O melhor classificador para o arquivo abandono foi o classificador Regressão Logística (5) com AUC de 0,90 se enquadrando como Excelente, para o arquivo situacao-geral-r e situacao-geral-rcfs o AdaBoost (2) foi o melhor classificadores assim por diante.

Aplicando os filtros e algoritmos de balanceamento é possível perceber uma alteração no valor da AUC, e dessa forma, no resultado final retornado pelos classificadores. Dessa forma os classificadores com melhores resultados formam o Random Forest, Logistic Regression, AdaBoost e LDA.

Assim, para o estudo da situação-geral é possível utilizar o classificador J48 que retornou bons resultados de desempenho e que possui algumas características que são interessantes retornando o conjunto de regras da classificação, gerando uma árvore de decisão que facilita na interpretação da classificação, além de utilizar o ganho de informação para efetuar a filtragem dos dados. Dessa forma o J48 será utilizado para a classificação da situação-geral a fim de verificar e analisar os resultados retornados por ele.

7.2.1 Situação Geral utilizando J48

Tabela 7.12: *Resultado da execução do classificador J48 para os arquivos de situação geral*

base/media													
situacao-geral	a b <–CM	Taxa VP	Taxa FP	Precisão	Recall	F-Measure	MCC	ROC Area	PRC Area	Classe	Acuracia	Tamanho Arvore	Nr. Folhas
	152 44 | a	0,776	0,267	0,745	0,776	0,760	0,509	0,806	0,768	positiva	75%	21	12
	52 143 | b	0,733	0,224	0,765	0,733	0,749	0,509	0,806	0,797	negativa			
	Weighted Avg.	0,754	0,246	0,755	0,754	0,754	0,509	0,806	0,782				
situacao-geral-r													
	173 23 | a	0,883	0,179	0,832	0,883	0,856	0,705	0,912	0,910	positiva	85%	27	15
	35 160 | b	0,821	0,117	0,874	0,821	0,847	0,705	0,912	0,884	negativa			
	Weighted Avg.	0,852	0,148	0,853	0,852	0,852	0,705	0,912	0,897				
situacao-geral-rcfs													
	181 15 | a	0,923	0,195	0,826	0,923	0,872	0,734	0,909	0,889	positiva	86%	13	7
	38 157 | b	0,805	0,077	0,913	0,805	0,856	0,734	0,909	0,875	negativa			
	Weighted Avg.	0,864	0,136	0,870	0,864	0,864	0,734	0,909	0,882				
situacao-geral-s													
	136 60 | a	0,694	0,074	0,824	0,694	0,753	0,650	0,881	0,752	positiva	84%	37	21
	29 361 | b	0,926	0,306	0,857	0,926	0,890	0,650	0,881	0,918	negativa			
	Weighted Avg.	0,848	0,229	0,846	0,848	0,845	0,650	0,881	0,863				
situacao-geral-scfs													
	119 77 | a	0,607	0,113	0,730	0,607	0,663	0,520	0,821	0,710	positiva	79%	31	16
	44 346 | b	0,887	0,393	0,818	0,887	0,851	0,520	0,821	0,868	negativa			
	Weighted Avg.	0,794	0,299	0,789	0,794	0,788	0,520	0,821	0,815				
hline													

Comparando as árvores geradas é possível perceber que para a árvore exposta na Figura 7.6 que utilizou a base situacao-geral-n aparecem duas questões do questionário sócio-econômico: se o aluno usa computador e de a família reside em Goiânia, se não reside o aluno é classificado com a situação negativa, o que nos leva a pensar que o fato de residir fora de Goiânia pode gerar mais desgaste no aluno, pelo fato de gastar mais tempo para chegar na universidade.

Quando olhando para a árvore gerada pela Figura 7.9 utilizando base situacao-geral-r que foi balanceada com o *Resample* mais características diferentes das notas nas disciplinas aparecem e são interessantes. Por exemplo, a quantidade de empregadas que pode ser considerada um indicador da situação econômica do estudante faz com que apareça uma situação do aluno negativa caso tenha uma ou nenhuma empregada. A ação afirmativa também aparece, sendo que para os alunos que foram beneficiados por esta ação existe a possibilidade do retorno de situação positiva e negativa, dependendo da renda mensal, do tempo que demorou para prestar vestibular e do tempo que cursou inglês.

Para a árvore da Figura 7.7 algumas novas características aparecem como se o aluno mora ou não com a família, se possui residência própria e se participou de processo seletivo com determinada idade quando cursou a disciplina poo e quando cursou

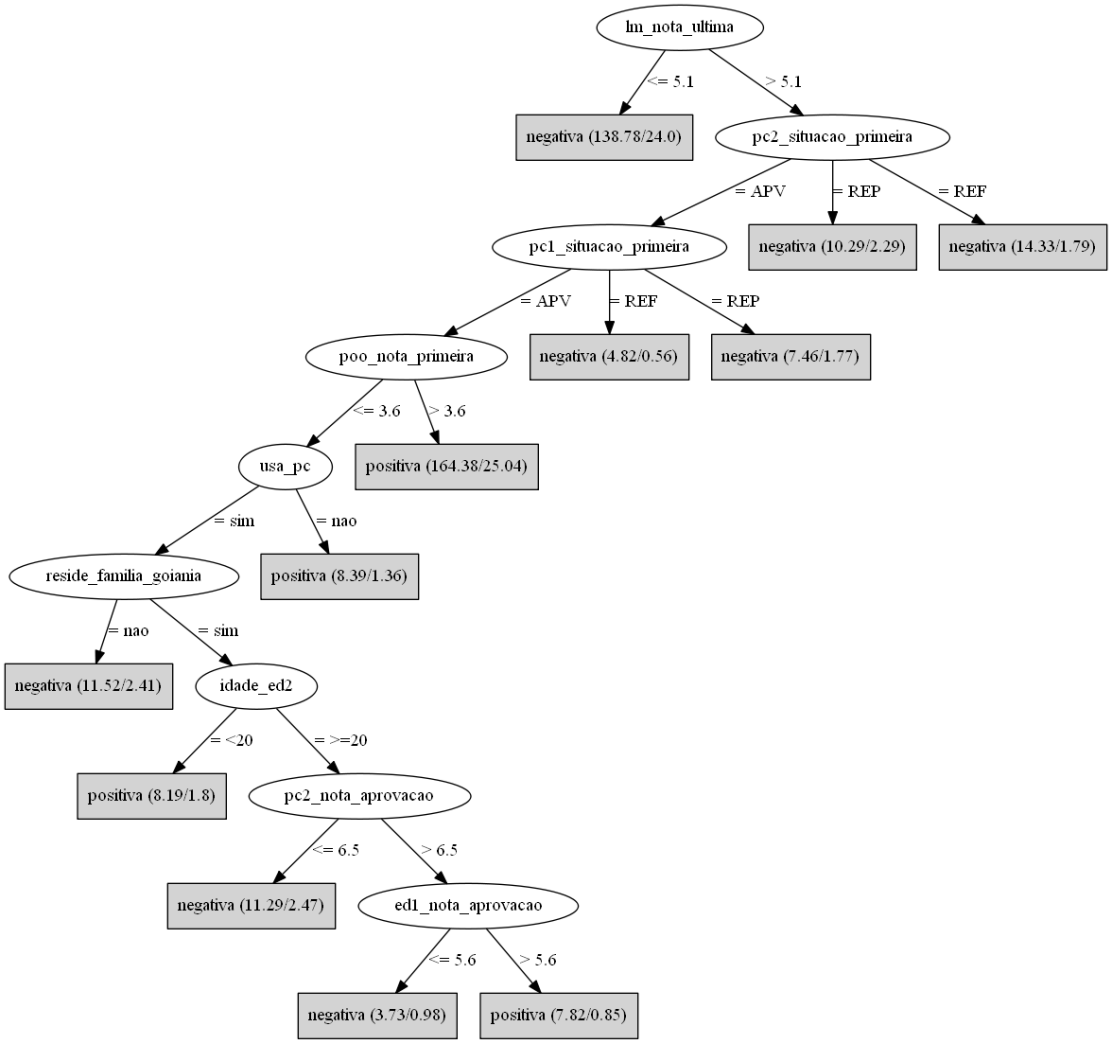

Figura 7.6: *Situação geral sem utilizar balanceamento de dados e filtro*

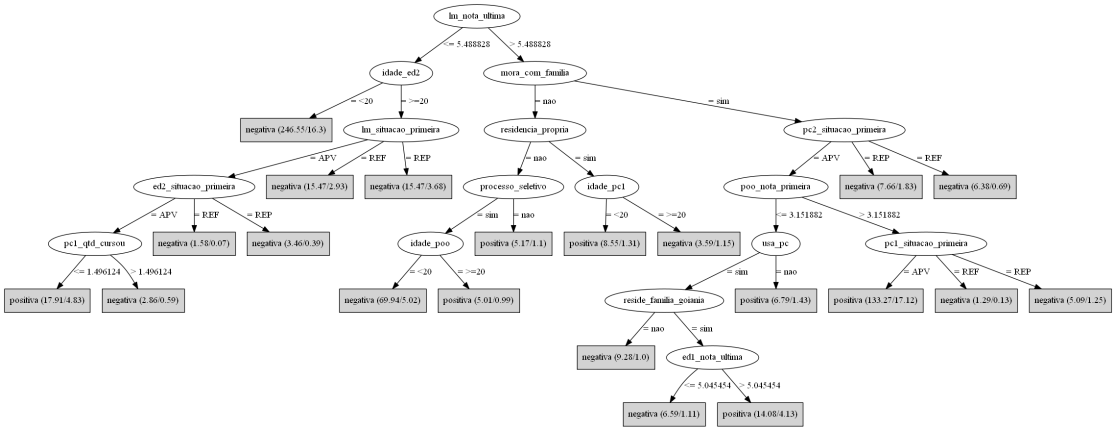

Figura 7.7: *Situação geral utilizando o SMOTE para balanceamento dos dados*

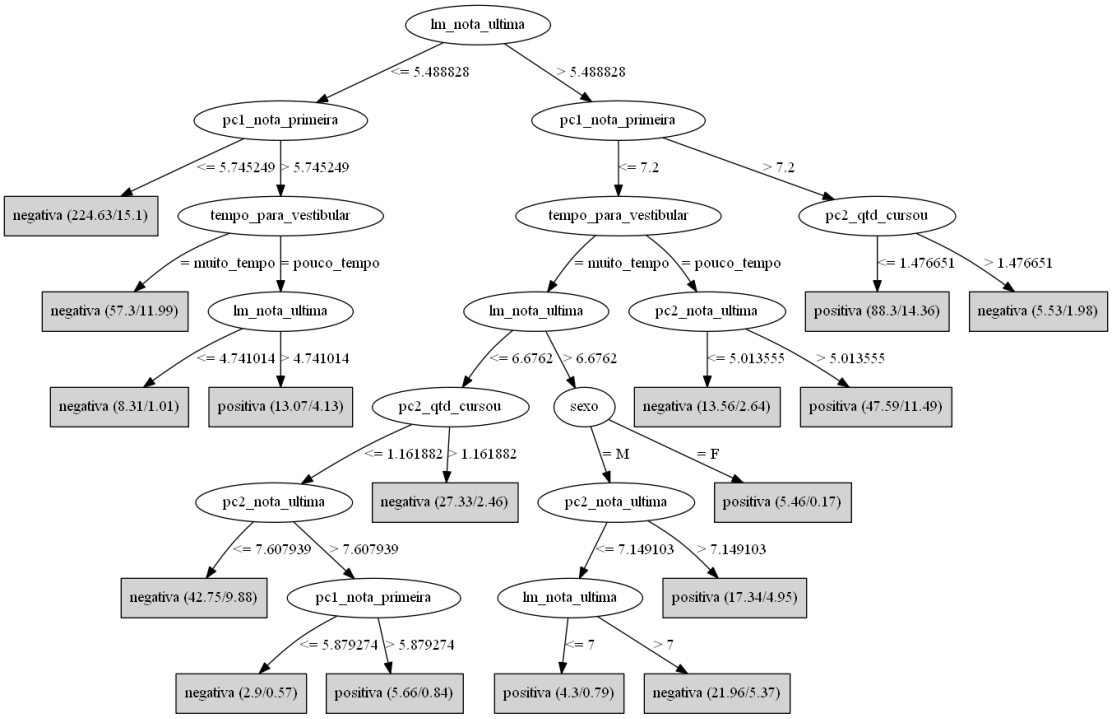

Figura 7.8: *Situação geral utilizando o SMOTE para balanceamento dos dados e filtro CFS para seleção de características*

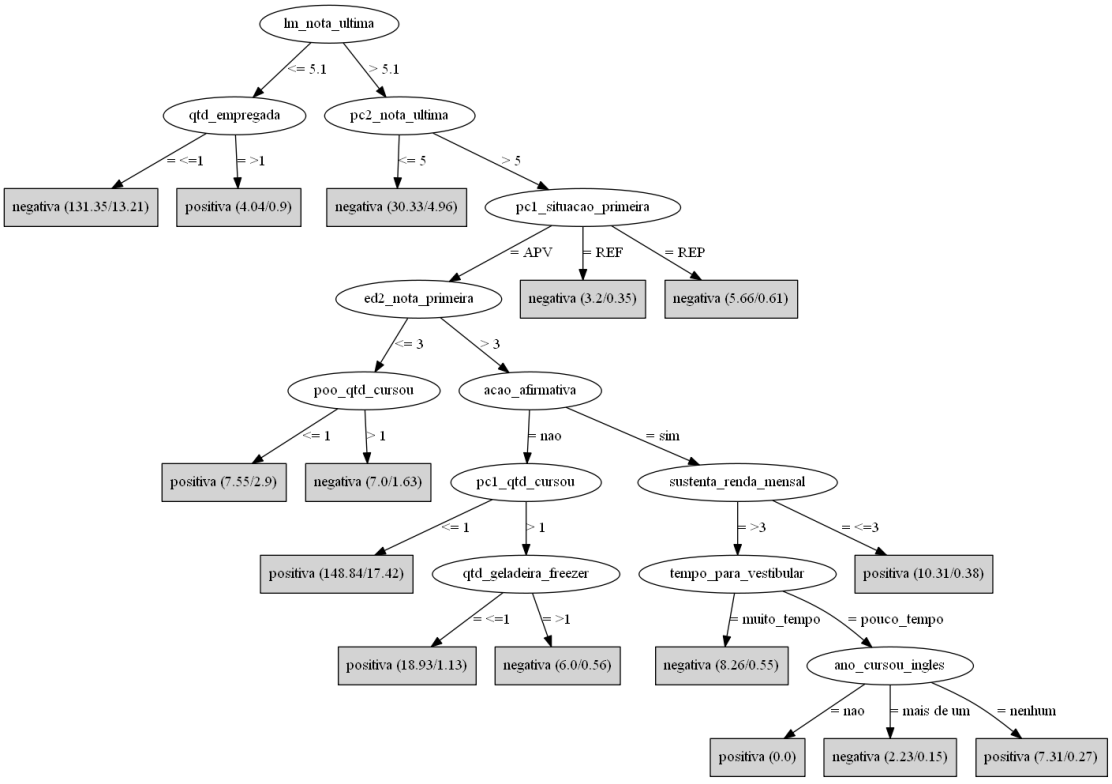

Figura 7.9: *Situação geral utilizando o Resample para balanceamento dos dados*

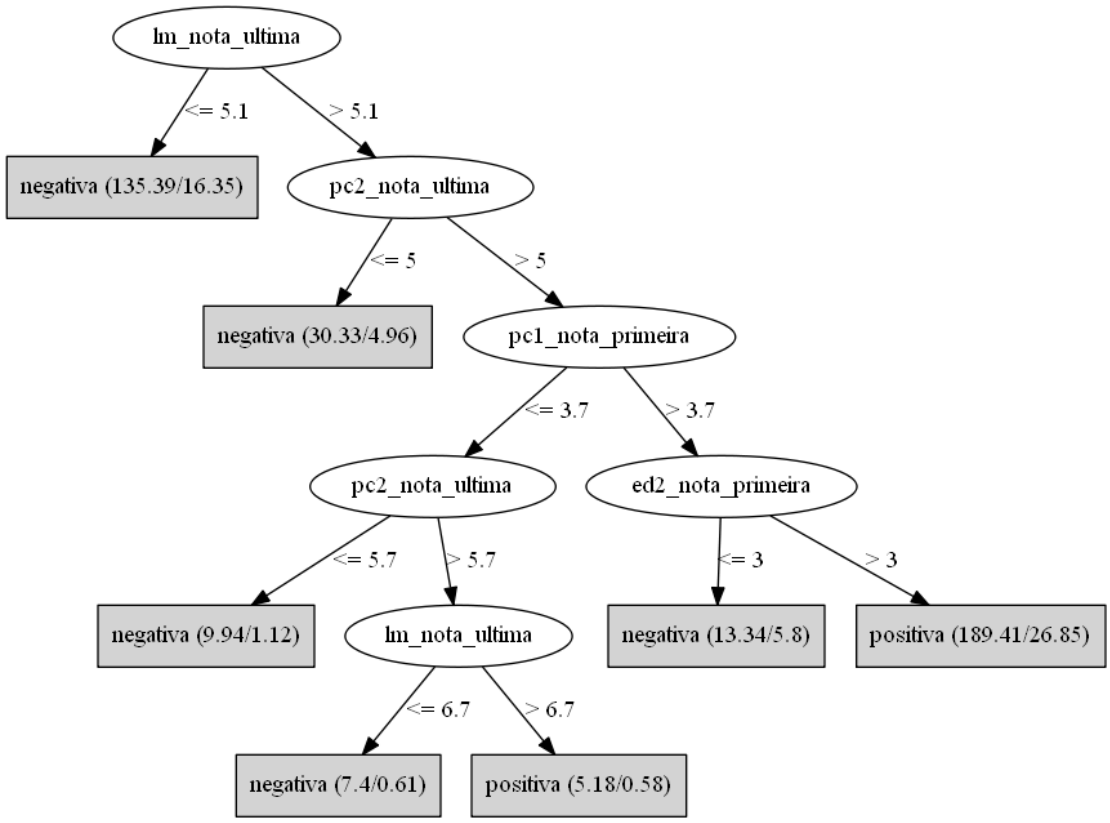

Figura 7.10: *Situação geral utilizando o Resample para balanceamento dos dados e o filtro CFS para seleção de características*

a disciplina pc1. Dependendo do valor de cada característica o aluno pode se enquadrar em situação negativa.

Pela Tabela 7.12 percebe-se que a maior acurácia é retornada utilizando a base situacao-geral-rcfs que utiliza o *Resample* com aplicação de filtros, apresentando a menor taxa de falso positivo para a classe minoritária e com maior AUC, sendo considerado um modelo excelente. Pela Matriz de Confusão é possível verificar que a quantidade de FN é bem menor que TN o que indica que que o classificador conseguiu classificar na maior parte dos casos os exemplos negativos com verdadeiramente negativos. Verificamos um considerável valor para precisão para a situação *positiva* e *negativa*, sendo que na média retornou um valor de 0,87, sendo que o ideal seria 1, porém sabemos que na maioria dos modelos não chega-se a esse valor. Pelo AUC o modelo se mostrou excelente para a tarefa de classificação. Todos os modelos de situação geral com o J48 podem ser utilizados para as tarefas de classificação.

Pela estatística descritiva foi possível perceber que a quantidade de valores para a situação *positiva* e para a *negativa*os dados não apresentaram desbalanceamento.

A partir do exposto e analisando a árvore de decisão da Figura 7.10, e considerando o interesse na classe minoritária, ou seja, aquela que retorna a situação *negativa* e

que significa que o aluno vai ter nota geral menor que a nota do curso, foi possível extrair as regras descritas a seguir:

- situacao-geral-rcfs

  Se (lm_nota_primeira <= 5.1) então = negativa

  Se (lm_nota_primeira > 5.1 e pc2_nota_ultima <= 5) então = negativa

  Se (lm_nota_primeira > 5.1 e pc2_nota_ultima > 5 e pc1_nota_primeira <= 3.7 e pc2_nota_ultima <= 5.7) então = negativa

  Se (lm_nota_primeira > 5.1 e pc2_nota_ultima > 5 e pc1_nota_primeira <= 3.7 e pc2_nota_ultima > 5.7 e lm_nota_ultima <= 6.7) então = negativa

  Se (lm_nota_primeira > 5.1 e pc2_nota_ultima > 5 e pc1_nota_primeira > 3.7 e ed2_nota_primeira <= 3) então = negativa

7.3 Situação Disciplinas utilizando J48

Esta análise tem como objetivo verificar se é possível predizer o desempenho do aluno (aprovado, reprovado ou reprovado por falta) para dada disciplina, usando como características preditoras informações sobre seu desempenho nas disciplinas de programação já cursadas em semestres anteriores. Lembrando que as disciplinas PC1 e LM são ofertadas no primeiro semestre do curso, PC2 no segundo, ED1 e POO no terceiro, e ED2 no quarto. Assim, para predizer o desempenho de PC2, do segundo semestre, utilizam-se os dados de PC1 e LM, ministradas no primeiro semestre. Para predizer o resultado de ED2, utilizam-se todos os dados das disciplinas anteriores.

Nesta seção são apresentados os resultados dos experimentos utilizando o classificador J48. Não utilizamos a técnica de *Resample* porque a variável alvo é formada por 3 classes, ou seja, não se trata mais de uma classificação binária. Assim serão usados os arquivos básicos, com aplicação do *SMOTE* e seleção de variáveis.

Considerando a Tabela 7.13 é possível verificar uma distribuição melhor para todos os indicadores de desempenho. Para as bases que contemplam o SMOTE e o SMOTE com filtro verificamos que não houve uma melhora na acurácia e nem mesmo na precisão. Os modelos retornam um AUC médio acima de 0,764, sendo que o modelo situacao-pc1-scfs chegou próximo dos 0,80, sendo classificado ainda como *Fraco*, porém pode ser utilizado como modelo para classificação da disciplina de programação de computadores I.

A partir da árvore de decisão situacao-pc1-scfs é possível extrair as seguintes regras para as classes minoritárias:

Tabela 7.13: *Resultado da execução do classificador J48 para os arquivos de situação pc1*

base/media situacao-pc1	a b c <–CM	Taxa VP	Taxa FP	Precisão	Recall	F-Measure	MCC	ROC Area	PRC Area	Classe	Acuracia	Tamanho Arvore	Nr. Folhas
	226 9 15 |a	0,904	0,508	0,873	0,904	0,888	0,419	0,781	0,830	APV	80.31%	34	18
	17 17 2 | b	0,472	0,043	0,586	0,472	0,523	0,472	0,808	0,464	REP			
	16 3 10 | c	0,345	0,059	0,370	0,345	0,357	0,295	0,633	0,211	REF			
	Weighted Avg.	0,803	0,413	0,794	0,803	0,797	0,414	0,771	0,731				
situacao-pc1-s													
	225 8 17 |a	0,900	0,383	0,862	0,900	0,881	0,538	0,773	0,773	APV	78.48%	41	22
	15 13 8 | b	0,361	0,042	0,500	0,361	0,419	0,369	0,784	0,401	REP			
	21 5 32 | c	0,552	0,087	0,561	0,552	0,557	0,468	0,711	0,562	REF			
	Weighted Avg.	0,785	0,297	0,773	0,785	0,778	0,509	0,764	0,698				
situacao-pc1-scfs													
	223 10 17 |a	0,892	0,447	0,842	0,892	0,866	0,472	0,785	0,815	APV	77.32%	22	12
	13 17 6 |b	0,472	0,042	0,567	0,472	0,515	0,467	0,782	0,370	REP			
	29 3 26 |c	0,448	0,080	0,531	0,448	0,486	0,394	0,795	0,554	REF			
	Weighted Avg.	0,773	0,343	0,760	0,773	0,765	0,458	0,786	0,725				

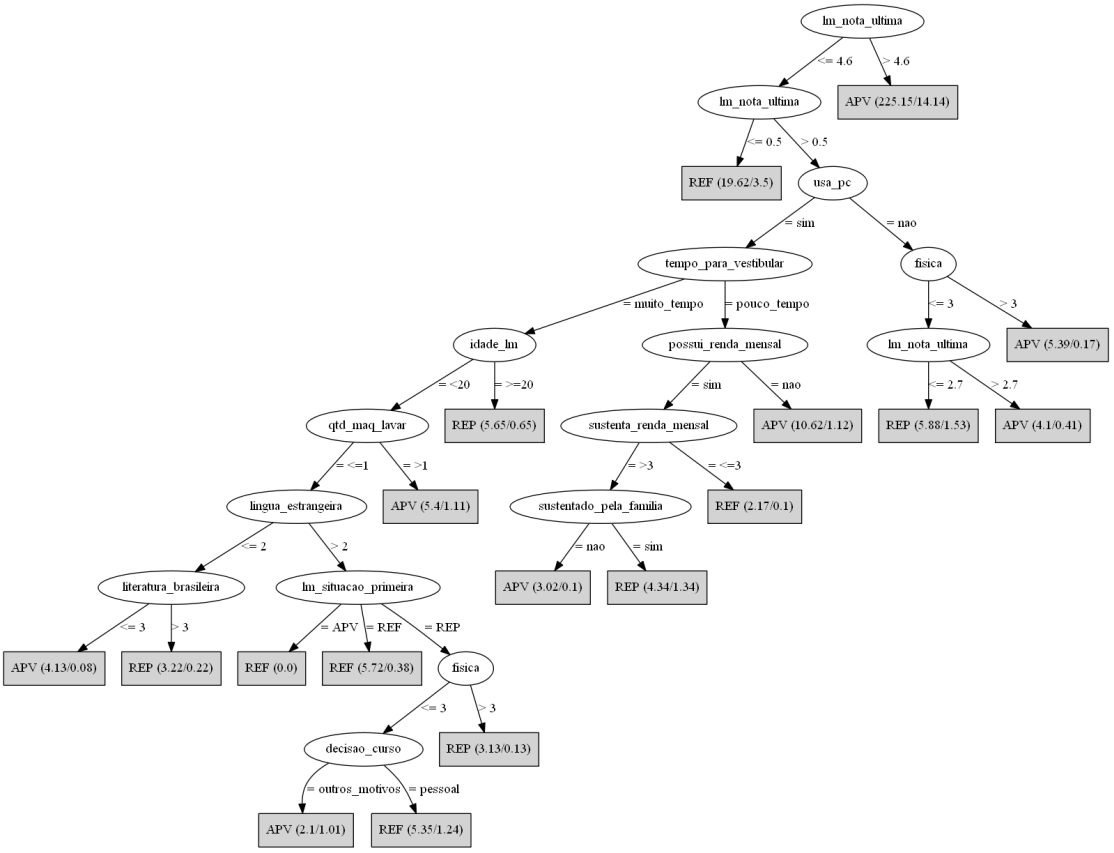

Figura 7.11: *Situação da disciplina pc1 sem balanceamento*

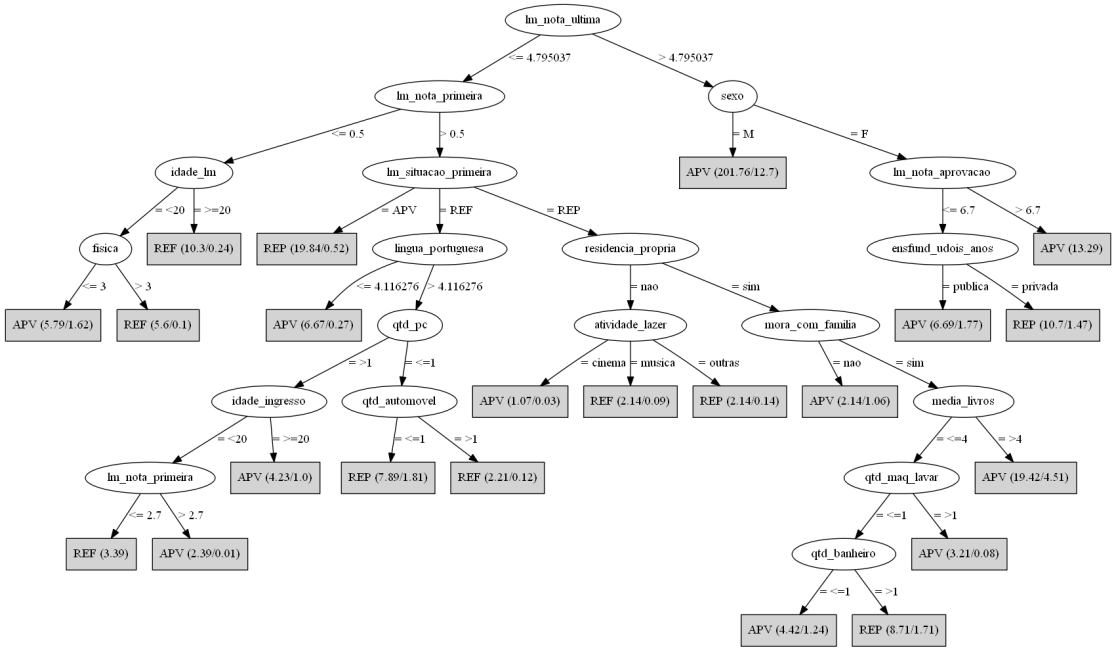

Figura 7.12: *Situação da disciplina pc1 utilizando SMOTE*

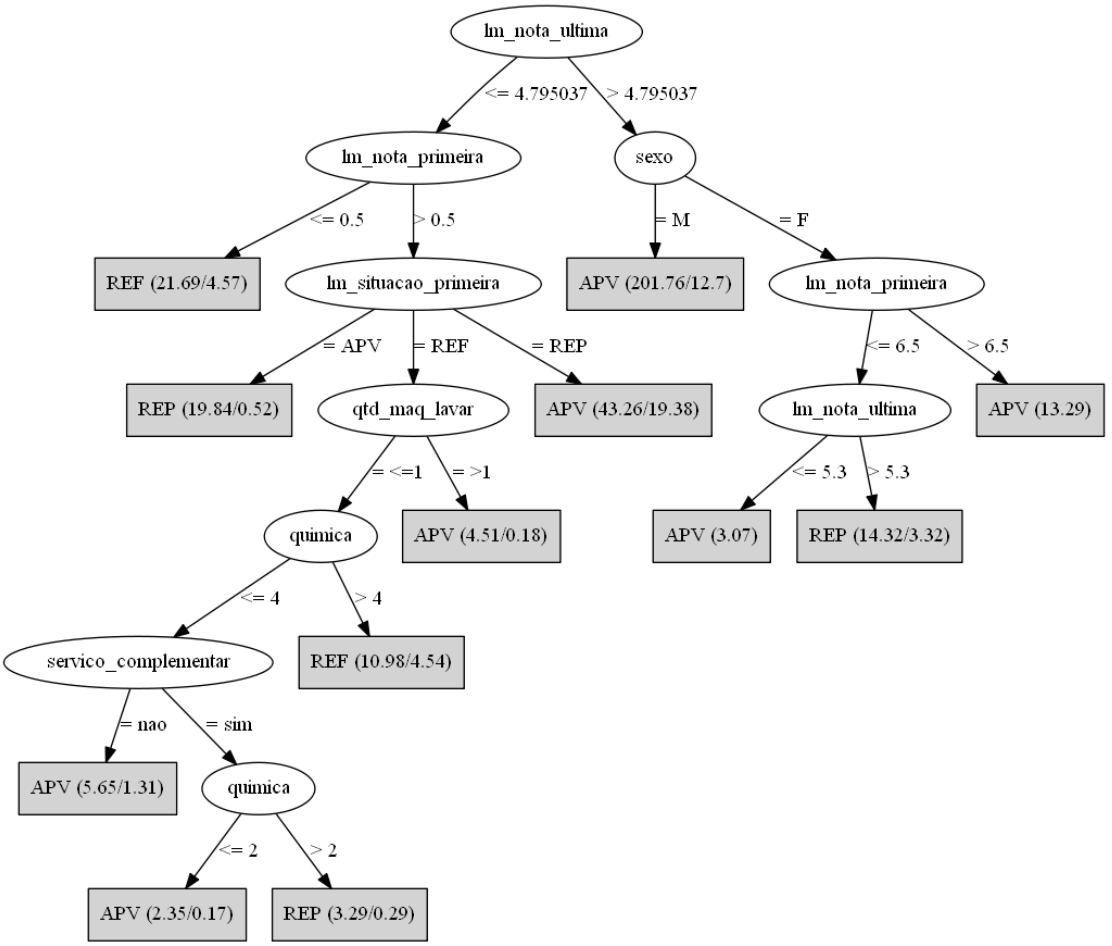

Figura 7.13: *Situação da disciplina pc1 utilizado SMOTE e filtro CFS para seleção de características*

- situacao-pc1-scfs

  Se (lm_nota_ultima <= 4.795037 e lm_nota_primeira <= 0.5) então = REF

  Se (lm_nota_ultima <= 4.795037 e lm_nota_primeira > 0.5 e lm_situacao_primeira =APV) então = REP

  Se (lm_nota_ultima <= 4.795037 e lm_nota_primeira > 0.5 e lm_situacao_primeira =REF e qtd_maq_lavar <=1 e quimica <= 4 e servico_complementar = sim e quimica > 2) então = REP

  Se (lm_nota_ultima <= 4.795037 e lm_nota_primeira > 0.5 e lm_situacao_primeira =REF e qtd_maq_lavar <=1 e quimica > 4) então = REF

  Se (lm_nota_ultima > 4.795037 e sexo = F e lm_nota_primeira <= 6.5 e lm_nota_ultima > 5.3) então = REP

Tabela 7.14: *Resultado da execução do classificador J48 para os arquivos de situação lm*

base/media situacao-lm	a b c <–CM	Taxa VP	Taxa FP	Precisão	Recall	F-Measure	MCC	ROC Area	PRC Area	Classe	Acuracia	Tamanho Arvore	Nr. Folhas
	222 5 2 |a	0,969	0,516	0,819	0,969	0,888	0,558	0,794	0,766	APV	77%	27	15
	22 17 8 |b	0,362	0,051	0,548	0,362	0,436	0,373	0,607	0,328	REP			
	27 9 12 |c	0,250	0,036	0,545	0,250	0,343	0,302	0,483	0,273	REF			
	Weighted Avg.	0,775	0,377	0,739	0,775	0,742	0,493	0,721	0,629				
situacao-lm-s													
	201 19 9 |a	0,878	0,345	0,804	0,878	0,839	0,552	0,786	0,738	APV	72%	51	28
	23 58 13 |b	0,617	0,112	0,652	0,617	0,634	0,514	0,746	0,586	REP			
	26 12 10 |c	0,208	0,068	0,313	0,208	0,250	0,168	0,521	0,182	REF			
	Weighted Avg.	0,725	0,250	0,702	0,725	0,711	0,493	0,742	0,628				
situacao-lm-scfs													
	210 10 9 |a	0,917	0,380	0,795	0,917	0,852	0,576	0,808	0,756	APV	73%	40	22
	26 56 12 |b	0,596	0,079	0,718	0,596	0,651	0,551	0,757	0,636	REP			
	28 12 8 |c	0,167	0,065	0,276	0,167	0,208	0,127	0,504	0,196	REF			
	Weighted Avg.	0,739	0,263	0,709	0,739	0,718	0,512	0,756	0,653				

Considerando a Tabela 7.14 a maior acurácia foi retornada pela base situacao-lm com 77%. As outras duas bases situacao-lm-s e situacao-lm-scfs retornaram 72% e 73% para a acurácia respectivamente. Aplicando o filtro SMOTE e SMOTE com CFS, verificamos que a Taxa de VP para a classe REP melhora, a classe REF não tem uma melhora significativa para esta taxa. Em termos de AUC para a classe REF os modelos na média ainda são considerados fracos e com baixa imprecisos.Percebe-se também que as árvores geradas são grandes e possuem número de folha acima de 15. Assim, as regras geradas pelas árvores de decisão para a situacao-lm não serão consideradas.

Considerando a Tabela 7.15 o valor da acurácia para todas as três bases (situacao-pc2,situacao-pc2-s e situacao-pc2-scfs) são respectivamente 87%, 84% e 85%. Isso gera

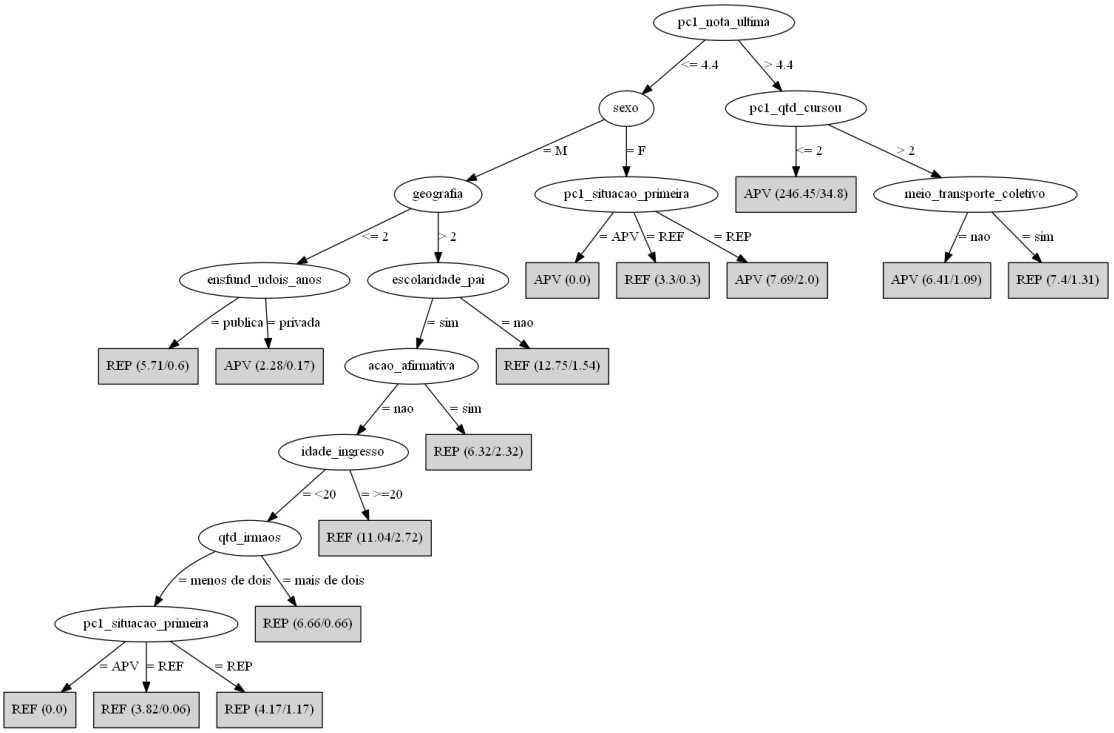

Figura 7.14: *Situação da disciplina lm sem balanceamento*

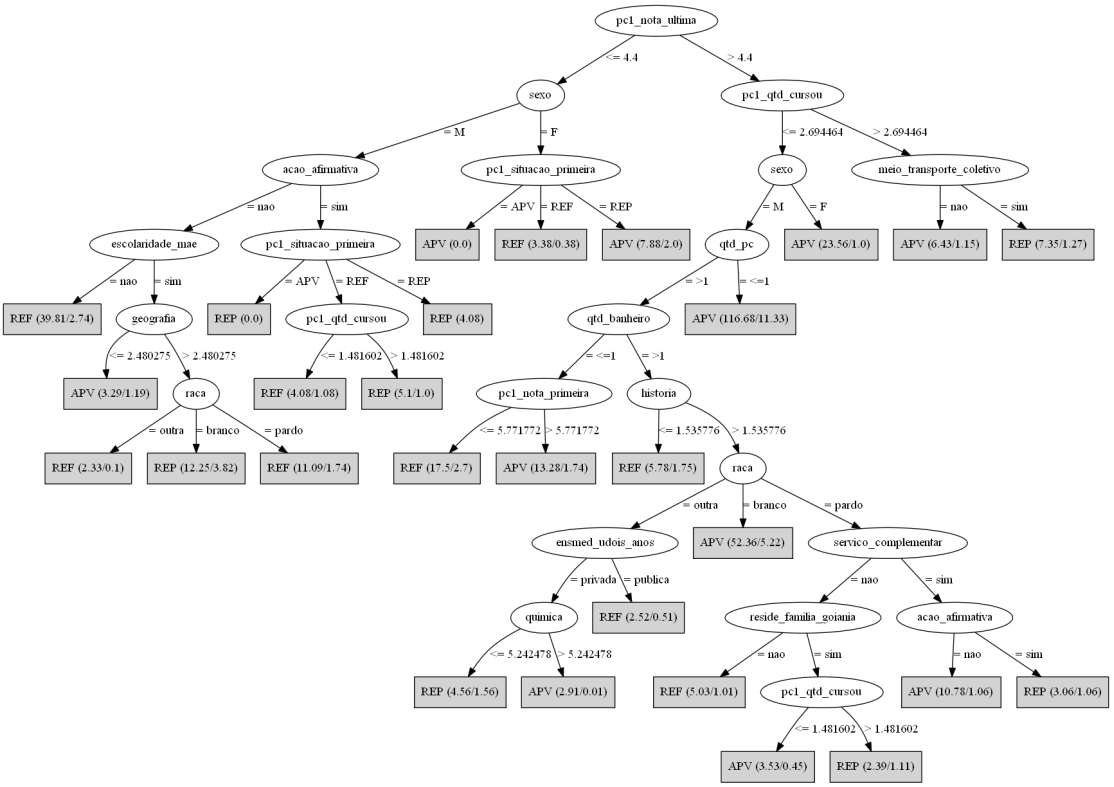

Figura 7.15: *Situação da disciplina lm utilizando SMOTE*

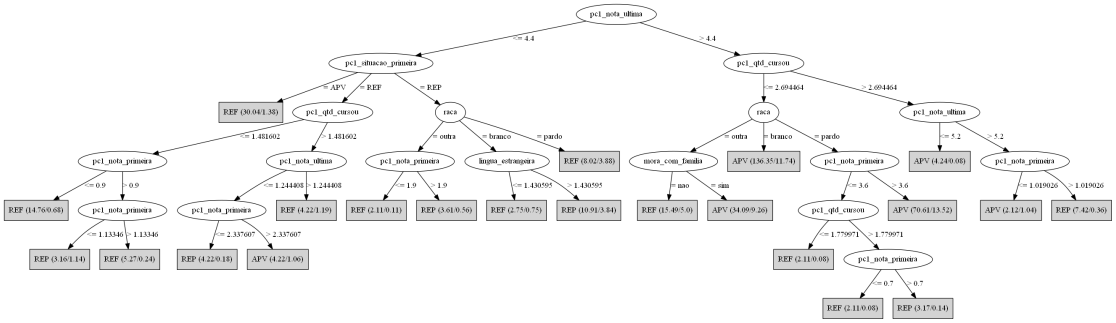

Figura 7.16: *Situação da disciplina lm utilizando SMOTE com filtro CFS para seleção de características*

Tabela 7.15: *Resultado da execução do classificador J48 para os arquivos de situação pc2*

base/media situacao-pc2	a b c <–CM	Taxa VP	Taxa FP	Precisão	Recall	F-Measure	MCC	ROC Area	PRC Area	Classe	Acuracia	Tamanho Arvore	Nr. Folhas
	210 0 0|a	1,000	1,000	0,882	1,000	0,938	0,000	0,499	0,537	APV	88%	1	1
	10 0 0 |b	0,000	0,000	0,000	0,000	0,000	0,000	0,500	0,026	REP			
	18 0 0 |c	0,000	0,000	0,000	0,000	0,000	0,000	0,454	0,044	REF			
	Weighted Avg.	0,882	0,882	0,779	0,882	0,827	0,000	0,496	0,478				
situacao-pc2-s													
	202 6 2|a	0,962	0,842	0,863	0,962	0,910	0,187	0,590	0,619	APV	83%	25	13
	17 3 0 |b	0,150	0,031	0,300	0,150	0,200	0,165	0,684	0,129	REP			
	15 1 2 |c	0,111	0,009	0,500	0,111	0,182	0,211	0,562	0,089	REF			
	Weighted Avg.	0,835	0,716	0,791	0,835	0,800	0,187	0,595	0,541				
situacao-pc2-scfs													
	205 4 1|a	0,976	0,789	0,872	0,976	0,921	0,302	0,683	0,699	APV	85%	7	4
	14 6 0 |b	0,300	0,026	0,500	0,300	0,375	0,347	0,769	0,181	REP			
	16 2 0 |c	0,000	0,004	0,000	0,000	0,000	-0,018	0,418	0,039	REF			
	Weighted Avg.	0,851	0,671	0,779	0,851	0,810	0,282	0,671	0,610				

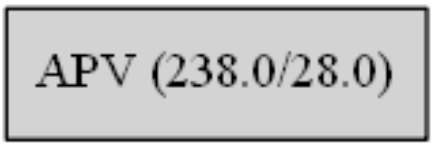

Figura 7.17: *Situação da disciplina pc2 sem balanceamento*

a expectativa de que este modelo poderia ser utilizado, porém olhando para a classe minoritária REF percebe-se que apresenta valores muito baixos para várias medidas, chegando a zero para VP, precisão, recall F-Measure e com um valor de -0,018 para o MCC considerando a base situacao-pc2-scfs. Olhando para a base situacao-pc2-s é possível verificar baixas Taxas de VP,Precisão, Recall, F-Measure e MCC para a classe REF. Os modelos são imprecisos e não classificam de forma adequada as classes minoritárias. Assim para a situação do aluno na disciplina pc2 os modelos não classificam as características de forma adequada e não são confiáveis para esta tarefa por tenderem a classificar de forma aleatória.

Pela Tabela 7.16 tanto a situacao-ed1-s e situacao-ed1-scfs retornaram taxas de acurácia de 82% e 83% respectivamente, sendo melhores que a taxa retornada pela base situacao-ed1. Olhando para a classe *REP* considerando todas as três bases verifica-se uma baixas taxas para todas as medidas, visto que no caso do AUC, os valores não chegam a

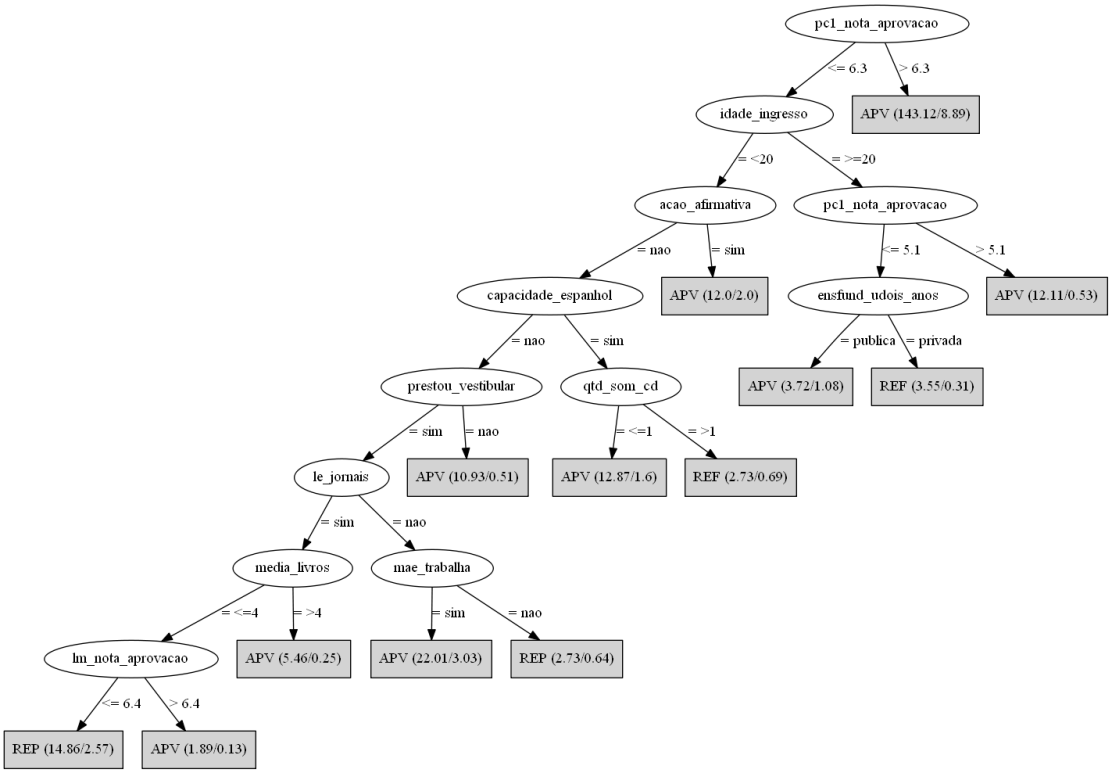

Figura 7.18: *Situação da disciplina pc2 utilizado SMOTE*

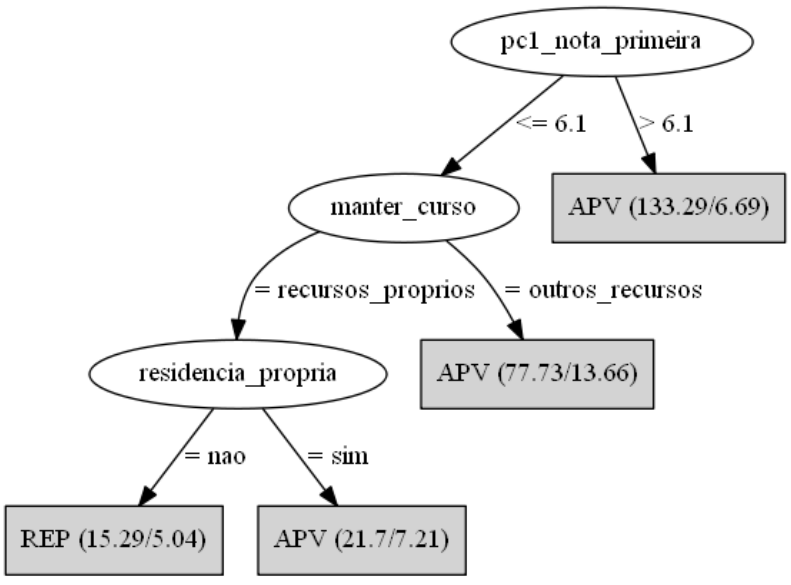

Figura 7.19: *Situação da disciplina pc2 utilizado SMOTE e filtro CFS para seleção de características*

Tabela 7.16: *Resultado da execução do classificador J48 para os arquivos de situação ed1*

base/media situacao-ed1	a b c <–CM	Taxa VP	Taxa FP	Precisão	Recall	F-Measure	MCC	ROC Area	PRC Area	Classe	Acuracia	Tamanho Arvore	Nr. Folhas
	162 5 2 |a	0,959	0,675	0,857	0,959	0,905	0,379	0,861	0,781	APV	81%	14	9
	18 3 2 |b	0,130	0,038	0,300	0,130	0,182	0,136	0,364	0,111	REP			
	9 2 6 |c	0,353	0,021	0,600	0,353	0,444	0,425	0,655	0,153	REF			
	Weighted Avg.	0,818	0,552	0,775	0,818	0,788	0,356	0,790	0,656				
situacao-ed1-s													
	158 6 5 |a	0,935	0,439	0,863	0,935	0,898	0,549	0,869	0,797	APV	82%	14	9
	17 5 1 |b	0,217	0,039	0,385	0,217	0,278	0,231	0,454	0,163	REP			
	8 2 24 |c	0,706	0,031	0,800	0,706	0,750	0,711	0,813	0,447	REF			
	Weighted Avg.	0,827	0,337	0,805	0,827	0,812	0,541	0,818	0,679				
situacao-ed1-scfs													
	162 3 4 |a	0,959	0,509	0,848	0,959	0,9	0,54	0,894	0,861	APV	83%	8	5
	19 3 1 |b	0,13	0,02	0,429	0,13	0,2	0,193	0,479	0,183	REP			
	10 1 23 |c	0,676	0,026	0,821	0,676	0,742	0,706	0,866	0,462	REF			
	Weighted Avg.	0,832	0,386	0,801	0,832	0,805	0,53	0,848	0,732				

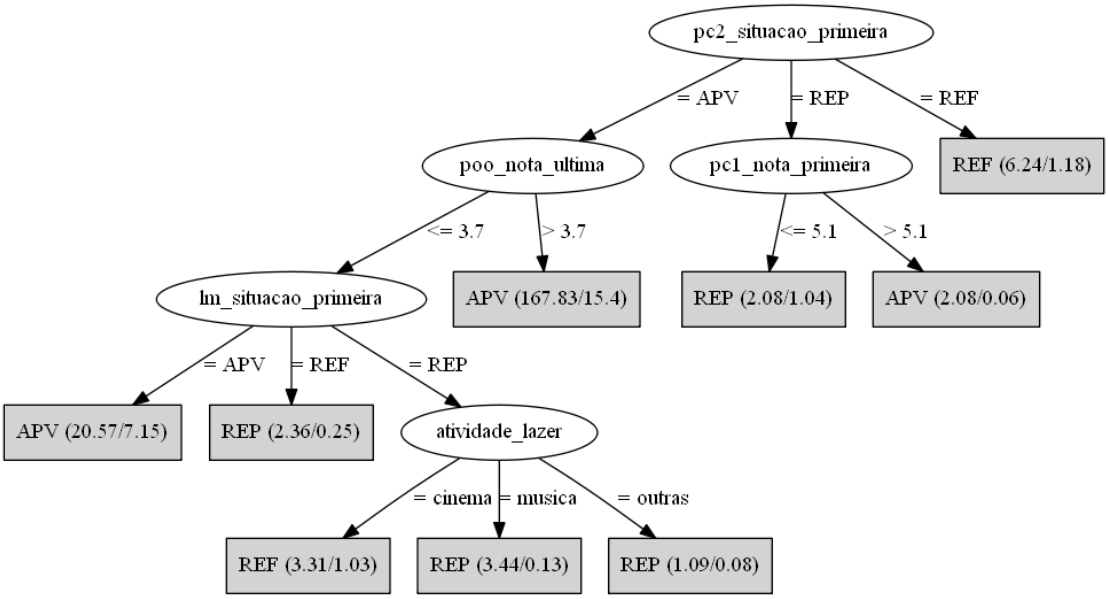

Figura 7.20: *Situação da disciplina ed1 sem balanceamento dos dados*

0,5, apesar de que na média retorna um valor acima de 0,79. Considerando a classe *REF*, após a aplicação do SMOTE e do filtro CFS, percebe-se um aumento em todas as taxas.

Para a disciplina de Estrutura de Dados I foi possível verificar a partir da Tabela 7.16 que não houve grande variação na acurácia. Mesmo após a aplicação do SMOTE e posteriormente do filtro CFS para as classes minoritárias continuam com poucos objetos, houve pequena variação na precisão.

Dessa forma considerando a árvore da situacao-ed1-s é possível extrair as seguintes regras para as classes minoritárias:

- situacao-ed1-scfs

  Se (pc2_situacao_primeira = APV e poo_nota_ultima <= 3.7 e lm_nota_ultima <= 4.812301) então = REP

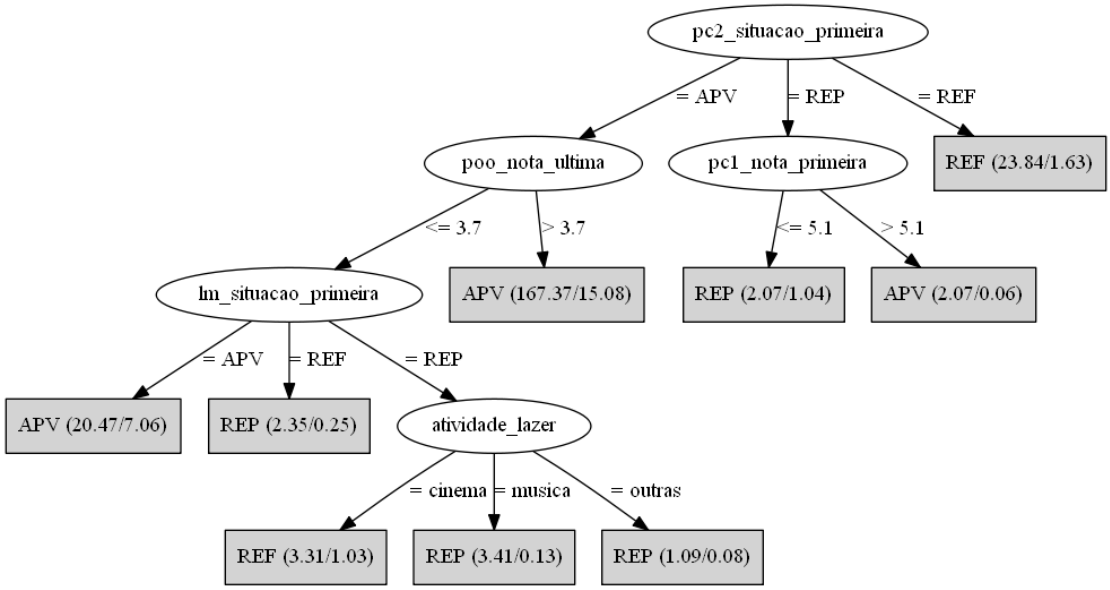

Figura 7.21: *Situação da disciplina ed1 utilizando o SMOTE*

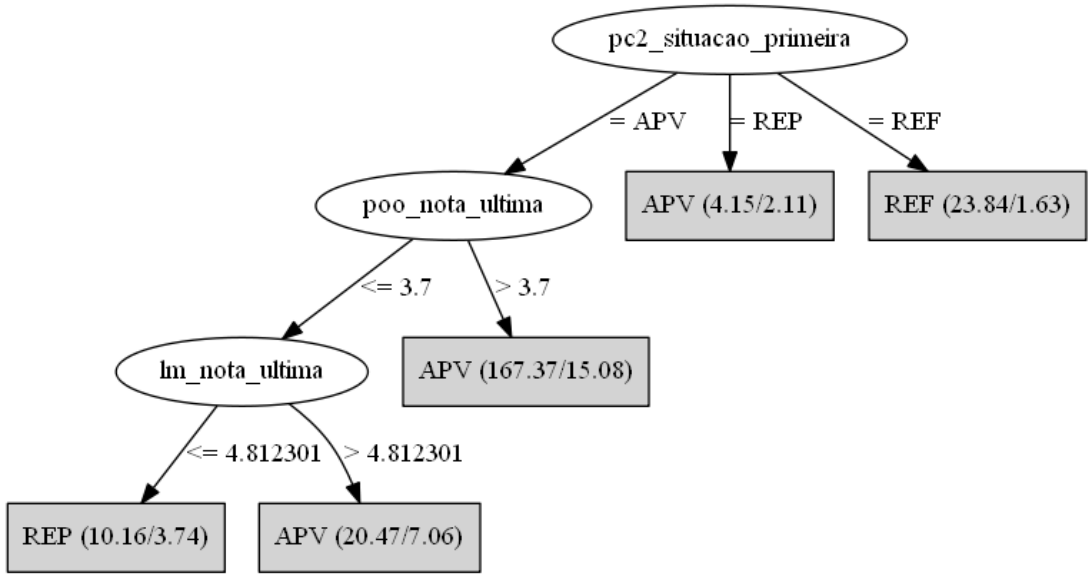

Figura 7.22: *Situação da disciplina ed1 utilizando o SMOTE e filtro CFS para seleção de características*

Se (pc2_situacao_primeira = REF) então = REF

Tabela 7.17: *Resultado da execução do classificador J48 para os arquivos de situação poo*

base/media situacao-poo	a b c <–CM	Taxa VP	Taxa FP	Precisão	Recall	F-Measure	MCC	ROC Area	PRC Area	Classe	Acuracia	Tamanho Arvore	Nr. Folhas
	160 5 2la	0,958	0,581	0,865	0,958	0,909	0,469	0,895	0,823	APV	80%	21	12
	17 4 4 lb	0,160	0,054	0,286	0,160	0,205	0,138	0,376	0,092	REP			
	8 5 5 lc	0,278	0,031	0,455	0,278	0,345	0,310	0,539	0,146	REF			
	Weighted Avg.	0,805	0,471	0,761	0,805	0,777	0,416	0,803	0,678				
situacao-poo-s													
	155 9 3la	0,928	0,590	0,812	0,928	0,866	0,406	0,860	0,776	APV	75%	21	12
	16 4 5 lb	0,160	0,059	0,250	0,160	0,195	0,123	0,433	0,144	REP			
	20 3 13lc	0,361	0,042	0,619	0,361	0,456	0,403	0,664	0,384	REF			
	Weighted Avg.	0,754	0,445	0,720	0,754	0,728	0,374	0,782	0,645				
situacao-poo-scfs													
	157 4 6la	0,940	0,475	0,844	0,940	0,890	0,531	0,868	0,797	APV	78%	18	10
	14 7 4 lb	0,280	0,044	0,438	0,280	0,341	0,288	0,500	0,200	REP			
	15 5 16 lc	0,444	0,052	0,615	0,444	0,516	0,450	0,727	0,417	REF			
	Weighted Avg.	0,789	0,361	0,763	0,789	0,770	0,491	0,805	0,672				

De acordo com a Tabela 7.17 os melhores resultados foram apresentados utilizando a base situacao-poo-scfs, onde a acurácia foi de 80%, AUC médio de 0,805, melhores taxas de VP, Precisão e MCC para as classes minoritárias REF e REP. As taxas da classe majoritária APV não sofreram grande variação. Apresentou uma árvore menor que as dos outros dois modelos (situacao-poo e situacao-poo-s) e com menor quantidade de folhas também. Dessa forma os modelos podem ser utilizados para classificar a disciplina poo.

Observando os resultados retornados pelas árvores da Figura 7.23, Figura 7.24 e da Figura 7.25, foi possível verificar algumas características não comuns a todos os modelos como idade_ed1 e residencia_propria por exemplo, porém considerando as outras características que constam nas árvores de decisão não se percebe uma grande alteração estrutural.

A partir da árvore de decisão situacao-poo-scfs é possível extrair as seguintes regras para as classes minoritárias:

- situacao-poo-scfs

  Se (ed1_nota_ultima <=4.749403 e ed1_situacao_primeira = APV) então = REF

  Se (ed1_nota_ultima <=4.749403 e ed1_situacao_primeira = REP e residencia_propria = nao) então = REP

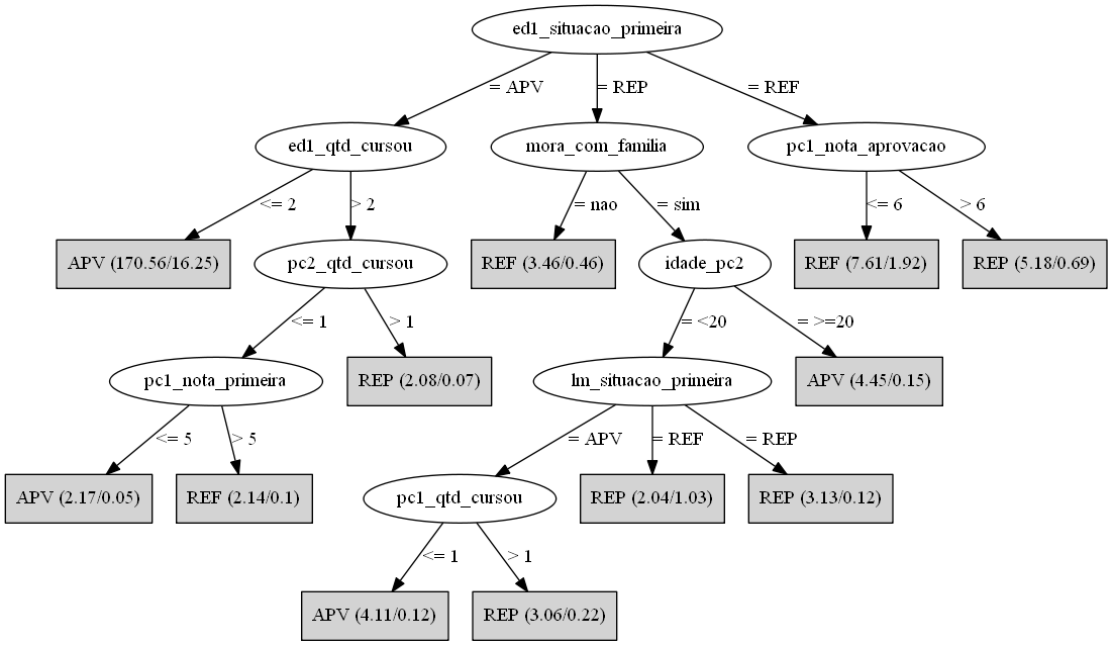

Figura 7.23: *Situação da disciplina poo sem balanceamento*

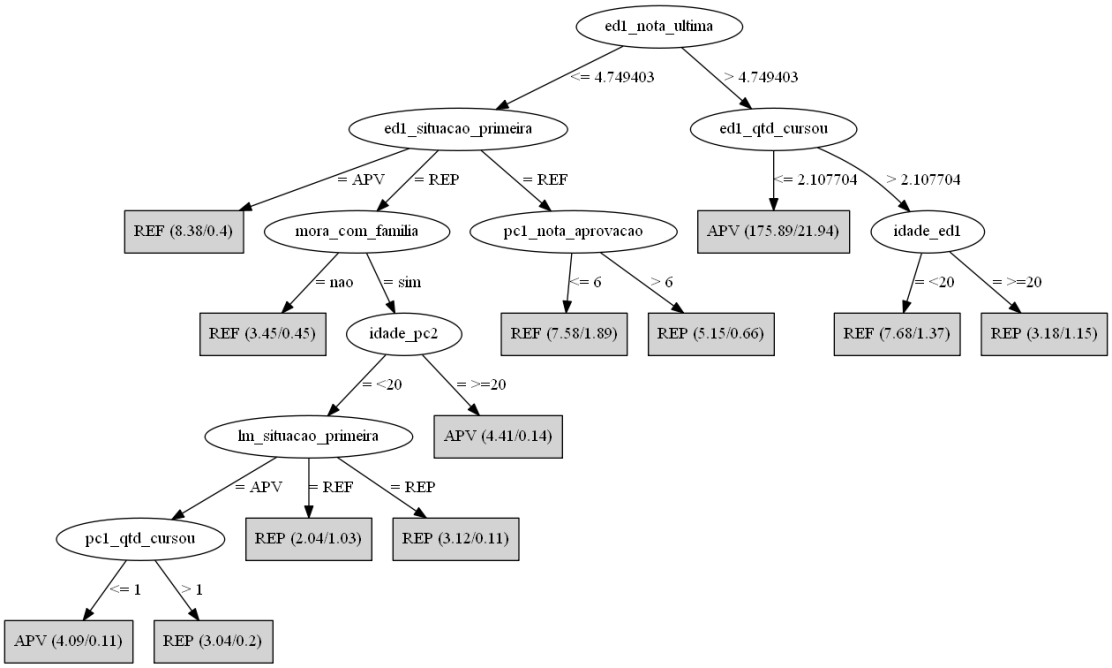

Figura 7.24: *Situação da disciplina poo utilizado SMOTE*

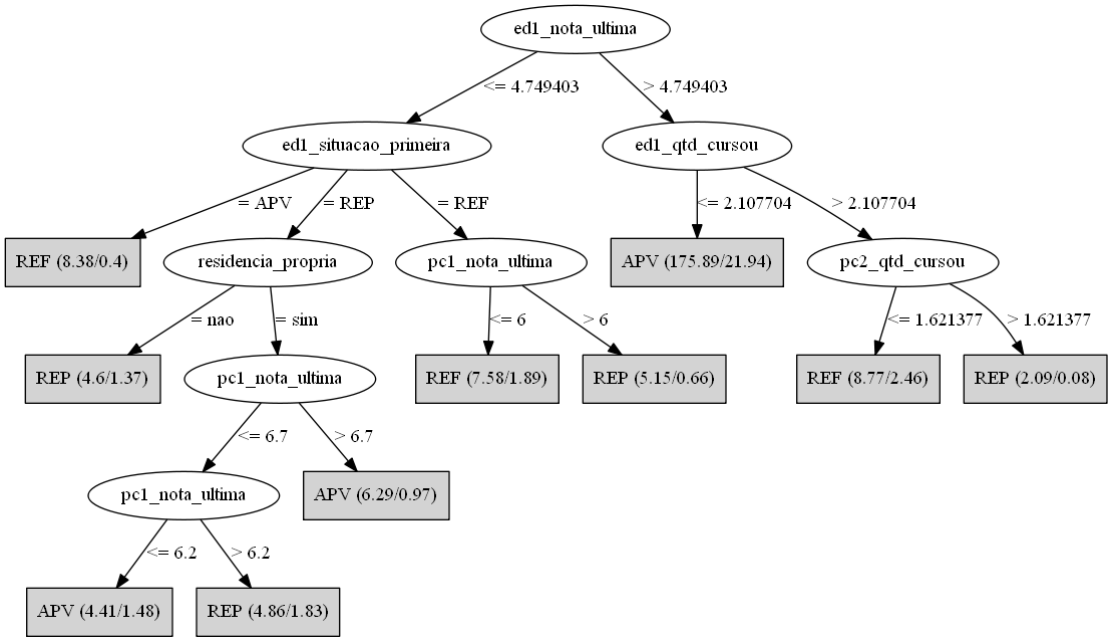

Figura 7.25: *Situação da disciplina poo utilizado SMOTE e filtro CFS para seleção de características*

Se (ed1_nota_ultima <=4.749403 e ed1_situacao_primeira = REP e residencia_propria = sim e pc1_nota_ultima <=6.7 e pc1_nota_ultima >6.2) então = REP

Se (ed1_nota_ultima <=4.749403 e ed1_situacao_primeira = REF e pc1_nota_ultima <= 6) então = REF

Se (ed1_nota_ultima <=4.749403 e ed1_situacao_primeira = REF e pc1_nota_ultima > 6) então = REP

Se (ed1_nota_ultima > 4.749403 e ed1_qtd_cursou >2.107704 e pc2_qtd_cursou <=1.621377) então = REF

Se (ed1_nota_ultima > 4.749403 e ed1_qtd_cursou >2.107704 e pc2_qtd_cursou >1.621377) então = REP

Pela Tabela 7.18 percebe-se que a acurácia tanto para a base situacao-ed2 quanto para a base situacao-ed2 retornaram valores entre 86% e 89%, sendo que a acurácia para a situação-ed2-scfs retornou uma acurácia de 87%. Considerando todas as base de situacao-ed2 verifica-se que a taxa VP, Precisão, Recall, F-Measure retornaram zero para a classe REF e valores abaixo de 0,4 para a classe REP, o que indica que o classificador não conseguiu classificar as classes minoritárias.

Tabela 7.18: *Resultado da execução do classificador J48 para os arquivos de situação ed2*

base/media situacao-ed2	a b c <–CM	Taxa VP	Taxa FP	Precisão	Recall	F-Measure	MCC	ROC Area	PRC Area	Classe	Acuracia	Tamanho Arvore	Nr. Folhas
	139 0 0 |a	1,000	1,000	0,891	1,000	0,942	0,000	0,495	0,353	APV	89%	1	1
	10 0 0 |b	0,000	0,000	0,000	0,000	0,000	0,000	0,499	0,026	REP			
	7 0 0 |c	0,000	0,000	0,000	0,000	0,000	0,000	0,347	0,017	REF			
	Weighted Avg.	0,891	0,891	0,794	0,891	0,840	0,000	0,488	0,317				
situacao-ed2-s													
	136 0 3 | a	0,978	0,750	0,883	0,978	0,928	0,354	0,492	0,340	APV	87%	9	5
	9 1 0 | b	0,100	0,000	1,000	0,100	0,182	0,307	0,470	0,049	REP			
	9 0 5 | c	0,357	0,020	0,625	0,357	0,455	0,437	0,720	0,294	REF			
	Weighted Avg.	0,871	0,641	0,868	0,871	0,842	0,359	0,510	0,318				
situacao-ed2-scfs													
	133 0 6 |a	0,957	0,792	0,875	0,957	0,914	0,233	0,507	0,344	APV	84%	7	4
	10 0 0 |b	0,000	0,000	0,000	0,000	0,000	0,000	0,475	0,074	REP			
	9 0 5 |c	0,357	0,040	0,455	0,357	0,400	0,354	0,835	0,246	REF			
	Weighted Avg.	0,847	0,679	0,785	0,847	0,814	0,229	0,533	0,319				

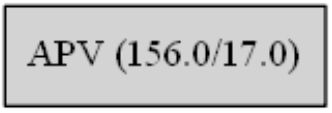

Figura 7.26: *Situação da disciplina ed2 sem balanceamento*

Dessa forma para a disciplina de estrutura de dados 2 não foi possível criar um modelo adequado para efetuar a classificação considerando o algoritmo J48.

7.4 Discussão

A classificação utilizando a situação geral retornou os melhores resultados, gerando modelos excelentes de acordo com o AUC. Para o Abandono, o modelo com *Resample* e que utiliza a Floresta Aleatória foi o que retornou melhores resultados para Taxa de Verdadeiro Positivo, Precisão e AUC, sendo dessa forma considerado o melhor classificador para o conjunto de dados Abandono. Vale observar que os melhores resultados foram obtidos com arquivos que possuem um certo equilíbrio entre o número de objetos pertencentes às classes alvo. Entre as técnicas de balanceamento adotadas, a de*Resample* obteve mais sucesso. Isto quer dizer que neste contexto é melhor escolher um subconjunto equilibrado dos dados, em vez de recorrer à criação de objetos sintéticos para o equilíbrio dos arquivos como é feito com o *SMOTE*. Na análise da situação por disciplina, em muitos casos não foi possível gerar um modelo. Isto pode se dar devido ao desequilíbrio das classes, agravado pelo fato de ser um classificador ternário. Nota-se que neste caso não foi possível usar a técnica de *Resample*. O uso do *SMOTE* também não trouxe ganho significativo.

As técnicas de seleção de atributo selecionam os atributos de maior relevância para serem utilizados pelos classificadores. Em vários trabalhos citados nas RSL foram utilizadas técnicas de seleção de atributos. A que retornou melhores resultados foi

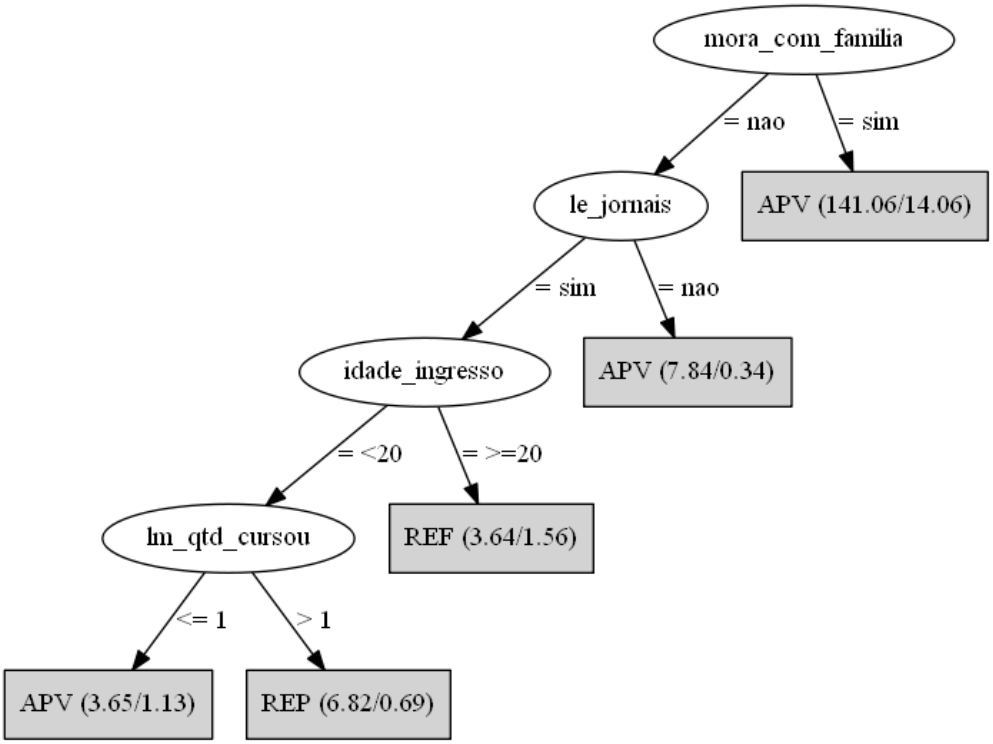

Figura 7.27: *Situação da disciplina ed2 utilizado SMOTE*

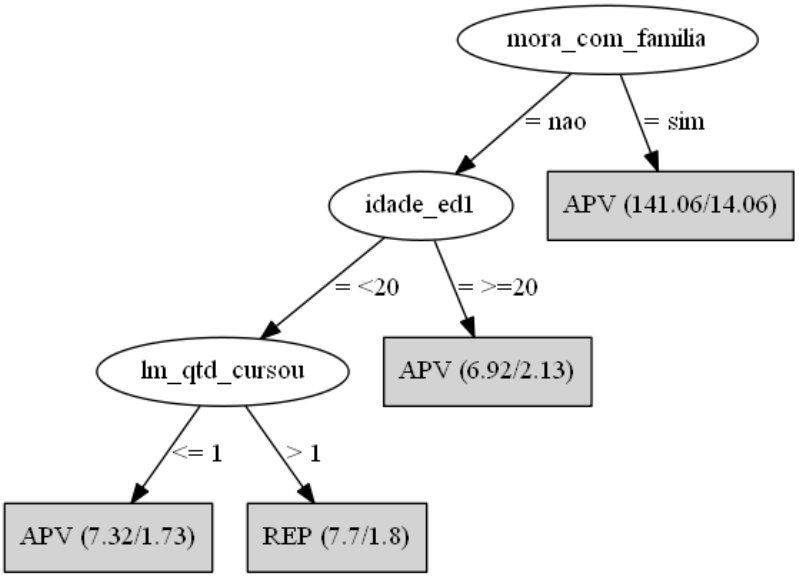

Figura 7.28: *Situação da disciplina ed2 utilizado SMOTE e filtro CFS para seleção de características*

a CFS e por este motivo foi utilizada para filtrar os dados considerando as classes abandono, situação geral e situação por disciplina logo após a aplicação dos algoritmos de balanceamento de dados.

A análise comparativa dos algoritmos de classificação mostrou que existe uma ligeira vantagem no uso de classificadores baseados em redes bayesianas e regressão logística. No entanto, esta vantagem nos índices de acurácia não compensa a transparência oferecida pelas árvores de decisão. Esta parece ser a conclusão alcançada pela maioria dos trabalhos da RSL, que utilizam árvore de decisão como principal método de classificação, e que foi confirmada pelos resultados obtidos.

Os dados usados da UFG cobrem a maioria dos tipos de dados encontrados na RSL. Foram usados dados demográficos, socio-econômicos, notas de avaliações anteriores ao ingresso no curso e durante o curso. Dentre os 113 atributos da base de dados usados no processo de mineração dos dados usando o Classificador J48, que permite uma analise dos modelos gerados, foi possível verificar que o preditor que mais aparece nos modelos é a nota da primeira vez que o aluno cursou PC1.

Analisando o abandono a partir do classificador J48 e olhando para o Ganho de informação, percebe-se que o atributo da disciplina PC1 (pc1_situacao_primeira,pc1_nota_ultima, pc1_nota_primeira) aparecem nas primeiras posições no ranking mesmo após a aplicação do filtro CFS. Na extração de regras percebemos também que o atributo pc1_situacao_primeira aparece na raíz da árvore de decisão para todas as bases de abandono. Este dado vem reforçar uma percepção informal dos professores que esta disciplina é chave no desempenho do aluno no curso. O aluno que reprova em PC1 muitas vezes acredita que se ele não dá conta desta disciplina que é básica do curso, então ele está no curso errado, o que leva ao abandono e troca de curso.

Olhando para os resultados apresentados pelas árvores de decisão, verificamos o aparecimento de características, antes não vistas no decorrer dos experimentos. Por exemplo, para a classificação dos alunos de acordo com o abandono, foi possível verificar por meio da árvore de decisão gerada pela Figura 7.1 o aparecimento de questões oriundos do questionário sócio-econômico como a quantidade de automóveis, a quantidade de banheiros se a família reside ou não em Goiânia, a nota de biologia se o aluno se mantem com recursos próprios e se trabalha ou não. Estes dados fazem sentido, uma vez que retornam a situação econômica da família considerando os bens patrimoniais e dependendo da falta de cada um, isso pode levar a uma situação de abandono comprovada pelo modelo gerado pelo J48.

Para a análise do desempenho nas disciplinas não efetuamos uma comparação entre classificadores pelo fato de serem subconjuntos dos dados da situação geral e por utilizarem 3 classes. Optou-se então pela utilização apenas do J48 para efetuar os experimentos. Foi possível predizer a situação de pc1 com bons resultados, com uma

acurácia de 80%, precisão média de 0,80 e taxa de VP média de 0,80. Porém mesmo após aplicar algoritmos de balanceamento, olhando a classe majoritária e as minoritárias, não houve alterações relevantes em relação aos resultados.

Em alguns casos não foi possível efetuar a classificação, como o caso pc2, pois para a classe REF foram retornadas baixas taxas de VP, Precisão, Recall, F-Measure e MCC, sendo o oposto para a classe majoritária APV. Considerando a base situacao-pc2-s não houve melhora nos resultados. Alias para a base situacao-pc2-scfs a classe REF retornou zero para o VP, sendo que a acurácia 85% nada mais é que um reflexo da classe majoritária.

CAPÍTULO 8

Conclusão

O objetivo da dissertação é o estudo exploratório de métodos de classificação desenvolvidos para a mineração de dados educacionais na previsão de resultados acadêmicos. Para atingir este objetivo foi feito um levantamento de técnicas de mineração de dados, com enfoque nos algoritmos de classificação. Ferramentas que implementam estes algoritmos foram estudadas.

Em paralelo foi realizado um levantamento bibliográfico para identificar trabalhos correlatos na área de Mineração de Dados Educacionais que utilizam algoritmos de classificação para a previsão de resultados acadêmicos. Verificou-se que esta é uma área de pesquisa recente e bastante ativa, mas que no Brasil ainda está pouco desenvolvida. A RSL mostrou que várias técnicas de classificação vêm sendo usadas, sendo que a maioria dos trabalhos explora mais de uma técnica. A mais usada é a Árvore de Decisão, que aparece em 75% dos artigos, seguida por Naive Bayes (16 artigos), Redes Neurais (14 artigos), e Suport Vector Machine (14 artigos). A opção por Árvores de Decisão dá-se pela sua transparência do modelo gerado, o que permite identificar claramente os atributos usados e as regras geradas. Os outros algoritmos apresentam-se como caixas pretas, onde é possível analisar o resultado final, mas não o processo. Assim, a não ser que os indicadores de desempenho da classificação sejam significativamente maiores, não vale a pena optar por estes classificadores. A execução dos algoritmos e feita através de ferramentas opensource disponíveis, sendo WEKA e RapidMiner as mais usadas.

Com relação ao tipo de desempenho de aluno analisados, verificou-se que estes tratam principalmente de abandono e o desempenho no curso ou na disciplina, apesar de existem diferentes interpretações de desempenho e abandono. Enquanto a classe abandono geralmente é binária, indicando se o aluno abandonou ou não, o desempenho pode apresentar uma gama maior de opções, classificando em até sete classes distintas. Quanto ao conjunto de atributos preditores, verificou-se que a tendência é inciar o processo com todos os atributos disponíveis, e fazer um refinamento durante a elaboração do modelo. Estes preditores abordam diversos aspectos relacionados aos alunos, incluindo dados demográficos, socio-econômicos, e notas de ingresso ou do curso.

Na base de qualquer estudo deste tipo estão os dados. Dados educacionais vêm

sendo coletados de forma sistemática por diversos órgãos. Ao nível Federal, o principal é o Instituto de Estudos e Pesquisas em Educação (INEP), que coleta dados e informações sobre os diversos aspectos nos diversos níveis da educação no Brasil, incluindo o Censo Escolar e o Censo do Ensino Superior. Institutos de Educação também coletam dados acadêmicos dos alunos, como rendimento escolar e percurso acadêmico. Além disto, com a expansão da oferta de cursos a distância e do uso de ambientes virtuais de aprendizagem, informações sobre as atividades desenvolvidas acabam sendo armazenadas, incluindo trabalhos submetidos, logs de acesso, entre outros. Análise estatísticas destes dados geram indicadores acadêmicos que muitas vezes servem como base para a definição de políticas educacionais. No entanto, com o uso de mecanismos de mineração de dados, outros tipos de análises podem ser aplicadas, e novas informações obtidas, abrindo o leque de possibilidades na análise educacional.

Neste trabalho foram utilizados dados acadêmicos e de ingresso dos alunos de Ciência da Computação da Universidade Federal de Goiás. Os dados foram obtidos através do CERCOMP e do Centro de Seleção da UFG. O processo de obtenção dos dados mostrou-se burocrático e demorado, o que pode comprometer projetos na área de mineração de dados educacionais. A fase de obtenção de dados foi complicada, pois apesar dos dados estarem em uma base de dados, não foi possível o acesso ao repositório, que é restrito. Por isso foram disponibilizados apenas parte dos dados, por meio de planilhas excel. Caso haja interesse por parte das instituições de ensino de promoverem projetos na área de análise de dados educacionais elas devem disponibilizar mecanismos de acesso aos dados mais práticos. Existem várias formas de disponibilizar acesso aos dados, como acesso via webservice (Método de comunicação e integração de diferentes aplicações) com certificado digital, API (Application Programming Interface - Interface de Programação de Aplicativos) REST(Represental State Transfer - Transferência de Estado Representacional).

Para que os dados pudessem ser usados nas análises eles tiveram que ser organizados e armazenados em um banco de dados, facilitando o acesso e a geração de visões dos dados de acordo com o algoritmo sendo aplicado. Caso ocorra alguma atualização nos dados será preciso efetuar a importação dos dados para algumas tabelas modeladas novamente, preservando os dados anteriores. O ideal seria ter um sistema com conexão ao Data Warehouse para efetuar estes processos.

O conjunto de dados obtido mostrou-se bastante incompleto, fazendo com que um número significativo de atributos não tenham valores atribuídos. Por estarmos trabalhando apenas com os alunos do curso de Ciência da Computação, o total de alunos foi de 391 alunos, sendo que por estes estarem em diversos semestres do curso, com percentagens variadas de conclusão do curso, muitas disciplinas ainda não foram cursadas. Isto acaba por diminuir o número de registros que foram efetivamente utilizados nas análi-

ses. A isto soma-se as lacunas existentes nos dados do Vestibular. Foram disponibilizados dados do vestibular de 290 dos alunos matriculados. Hoje o ingresso na UFG é feito através do SiSU com base nas notas do ENEM. Apesar disto, as análises realizadas podem ser facilmente adaptadas a esta nova realidade, já que as notas do ENEM para os alunos ingressantes agora estão armazenadas no banco de dados acadêmicos da universidade, assim como informações socio-econômicas coletadas através de questionário aplicado aos alunos após seu ingresso e que contempla as principais perguntas encontradas no questionário do vestibular.

Após o início do mestrado foram solicitadas informações atualizadas, mas estas não foram disponibilizadas. Existe um projeto dentro da UFG de tornar todos os dados disponíveis de forma anonimizada, mas até a presente data isto ainda não aconteceu. No futuro, caso o projeto se concretize, uma validação dos resultados obtidos na dissertação poderá ser feita com dados atualizados.

Uma análise dos dados obtidos mostrou que os alunos do curso de Ciências da Computação são predominantemente do sexo masculino, jovem e solteiro. Seguindo uma tendência mundial, apenas 10% dos alunos são mulheres. Aproximadamente 25% abandonam o curso. No entanto, com os dados disponíveis não foi possível fazer uma análise mais detalhada sobre o abandono, para identificar informações como em qual semestre este abandono geralmente acontece, os motivos apresentados, etc. Para isto, é preciso recolher e disponibilizar estas informações.

Antes de proceder à análise dos dados da UFG foi feito um estudo comparativo dos algoritmos de classificação usando dados do *dataset Student Performance* fornecido por Paulo Cortez da Universidade do Minho de Portugal disponível no repositório UCI.

Os testes considerando a base do UCI mostraram que o algoritmo de Regressão Logística, seguidos pelo AdaBoost e pelo RandomForest, obteve o melhor resultado, sendo que o classificador CART (árvore de decisão) permaneceu com uma acurácia média mais estável após execução do método de validação cruzada. Este resultado reforça a escolha de algoritmos baseados em árvores de decisão para a análise dos dados pelo fato de serem métodos whitebox e que possuem várias formas de visualização dos resultados da classificação, como as árvores de decisão e por meio de regras "se-então-senão", o que facilita a avaliação dos resultados gerados por pessoas que não possuem conhecimento técnico em mineração de dados. Além disso, em relação aos outros algoritmos que retornaram um bom resultado na classificação, não percebeu-se uma grande diferença estatística entre eles, assim, deixando a opção pela utilização dos algoritmos de árvores de decisão ainda mais pertinente. Verificou-se também a utilização de 10 execuções da validação cruzada e do particionamento do conjunto de dados em teste e treinamento para utilização dos classificadores. Para alguns classificadores a execução por meio de teste e treino retornou maiores acurácias em comparação à execução de 10 folds cross validation.

Para a análise dos resultados foram utilizados os toolboxes do Weka em conjunto com a biblioteca Scikit-Learn, que não foi relatada na RSL porém é de grande utilidade para as tarefas de mineração de dados, sendo possível assim efetuar comparações com classificadores escritos em Java e Python respectivamente, aproveitando o ambiente gráfico do Weka, porém utilizando bibliotecas escritas em Python. Foram realizadas análises para três contextos distintos. A primeira tentou predizer o abandono do aluno, a segunda seu desempenho geral no curso e a terceira, o desempenho por disciplina, semestre a semestre, levando em consideração as notas das disciplinas já cursadas.

Para as duas primeiras análises foi feita uma comparação de dez classificadores (LDA, QDA, NB, RL, SVM, CART, RF, AD, KNN da toolbox Scikit-learn, J48 (C4.5) e MLP do Weka), utilizando o WEE (Weka Experiment Environment). Esta comparação confirmou os resultados obtidos com os dados do *dataset Student Performance*, sendo que a Regressão Logistica e Floresta Aleatória obtiveram os melhores resultados, porém não tão melhores que justificassem seu uso em vez de árvores de decisão. Assim a análise foi feita usando o algoritmo J48 para a extração de regras e descoberta de padrões. Além disto, o desempenho levou em consideração a matriz de confusão, para a identificação mais precisa dos alunos com provável desempenho ruim. Verificou-se que esta abordagem muitas vezes implica em resultados distintos quando da avaliação de desempenho dos classificadores.

A comparação foi feita em cima de cinco arquivos: o arquivo original, o arquivo com Resample, o arquivo com Resample e CFS, o arquivo com SMOTE, e o arquivo com SMOTE e CFS, sendo Resample e SMOTE técnicas de balanceamento e CFS de seleção de atributos. Vale observar que os melhores resultados foram obtidos com arquivos que possuem um certo equilíbrio entre o número de objetos pertencentes às classes alvo. Entre as técnicas de balanceamento adotadas, a de Resample obteve mais sucesso. Na análise da situação por disciplina, em muitos casos não foi possível gerar um modelo. Isto pode se dar devido ao desequilíbrio das classes, agravado pelo fato de ser um classificador ternário.

A partir do Ganho de Informação foi possível perceber que a disciplina de PC1 apareceu em várias árvores de decisão, muitas vezes na raiz, indicando a importância da disciplina para o processo de classificação de acordo como desempenho e com o abandono. Este dado vem reforçar uma percepção informal de que esta disciplina é chave no desempenho do aluno no curso e na sua decisão de abandonar ou não. O aluno que reprova em PC1 muitas vezes acredita que se ele não dá conta desta disciplina que é básica do curso, então ele está no curso errado, o que leva ao abandono e troca de curso. Atributos que indicam situação econômica, como a quantidade de automóveis, a quantidade de banheiros, se a família reside ou não em Goiânia, também aparecem nas árvores de decisão indicando que o aspecto financeiro tem influência no desempenho dos

alunos.

Os resultados obtidos neste trabalho demonstram que técnicas de mineração de dados aplicadas à mineração de dados educacionais contribuem para a predição de abandono e desempenho (situação do aluno no curso e nas disciplinas) dos alunos. O estudo das características extraídas que influenciam no abandono e desempenho podem auxiliar na construção de possíveis propostas de intervenção que objetivam melhorar os resultados obtidos pelos alunos e para que permaneçam no curso de CC.

Apresentamos neste trabalho a comparação de diversos classificadores, onde a maior parte dos modelos de classificação obtidos apresentaram um bom valor para o AUC. Os algoritmos utilizados para o balanceamento dos dados e para a seleção de características foram de grande valia, melhorando os resultados e para identificar as características que de fato possuem um maior e menor grau de importância para a classificação. No entanto, estes resultados se aplicam ao contexto e dados do curso de Ciências da Computação usados. Apesar de haver um princípio de confluência quanto ao algoritmo de classificação e técnicas de tratamento de dados a serem usados, os dados ainda permanecem como a grande incógnita. Ainda não existe um consenso quanto ao tipo de dado (notas, demográficos, socio-econômicos, etc.) que melhor modela o desempenho, o que dificulta uma sistematização do processo. Maiores estudos serão necessários para identificar os dados relevantes e padronizá-los.

8.0.1 Trabalhos futuros

Para trabalhos futuros é desejável um conjunto maior de dados para melhorar a precisão dos modelos gerados. Além disso neste trabalho não foram consideradas todas as disciplinas do curso de Ciências da Computação, apenas aquelas ligadas à área de programação. Sugere-se a análise de todas as disciplinas do curso de Ciências da Computação da UFG para verificar se de fato alguma disciplina, que não consta neste trabalho, tem maior influência nos resultados finais. Pode-se também considerar outros cursos da área de informática existentes na UFG e até mesmo utilizar dados de outros institutos e se possível de toda a universidade para análise.

Visto que os classificadores Regressão Linear e Redes Bayesianas tiveram bons desempenhos nas análises comparativas, estes devem ser investigados mais a fundo apesar de funcionarem como caixa-preta, não permitindo uma análise do modelo. No entanto, pode-se verificar se ao se analisar os resultados destes classificadores em conjunto com a árvore de decisão, melhores resultados podem ser obtidos. Pode-se também utilizar outros classificadores, como aqueles que utilizam a estratégia evolucionária para avaliar se de fato podem ser utilizados no contexto educacional.

Caso o projeto da UFG de disponibilizar os dados dos alunos da universidade de

maneira anonimizada se concretize, isto irá abrir um leque de oportunidades para análise, inclusive análises comparativas.

Sugere-se também a utilização da análise de aprendizagem (Learning Analytics) para a partir dos resultados obtidos pelos classificadores estudados neste trabalho, modelar e desenvolver sistemas que possuem a capacidade de capturar dados e elaborar relatórios a partir de uma base de dados que cresce de forma contínua para que a instituição possa planejar intervenções, ações por meio do conhecimento prévio dos alunos retornados pelos classificadores.

Além do exposto, acredita-se ser importante considerar outros aspectos, obtidos por meio de estudos e testes psicológicos, para aliar com o descrito neste estudo para verificar se eles influenciam nos resultados relativos à predição do desempenho e do abandono dos estudantes.

Referências Bibliográficas

[1] **Comparison of machine learning methods for intelligent tutoring systems**. In: Ikeda, M.; Ashley, K.; Chan, T.-W., editors, *Intelligent Tutoring Systems*, volume 4053 de **Lecture Notes in Computer Science**. Springer Berlin Heidelberg, 2006.

[2] **Big Data: Techniques and Technologies in Geoinformatics**. CRC Press, 2014.

[3] ABDULLAH, A.; MALIBARI, A.; ALKHOZAE, M. **Students'performance prediction system using multi agent data mining technique**. *International Journal of Data Mining & Knowledge Management Process*, 4(5):1, 2014.

[4] ADHATRAO, K.; GAYKAR, A.; DHAWAN, A.; JHA, R.; HONRAO, V. **Predicting students' performance using id3 and c4. 5 classification algorithms**. *arXiv preprint arXiv:1310.2071*, 2013.

[5] AGARWAL, S.; PANDEY, G.; TIWARI, M. **Educational data mining: A review of the state of the art**. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 40(6):601–618, Nov. 2010.

[6] AGGARWAL, C. C.; WOLF, J. L.; YU, P. S.; PROCOPIUC, C.; PARK, J. S. **Fast algorithms for projected clustering**. *SIGMOD Rec.*, 28(2):61–72, June 1999.

[7] AGRAWAL, R.; SRIKANT, R. **Fast algorithms for mining association rules in large databases**. In: *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, p. 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.

[8] AGRAWAL, R.; SRIKANT, R.; OTHERS. **Fast algorithms for mining association rules**. In: *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, p. 487–499, 1994.

[9] AHMAD, F.; ISMAIL, N. H.; AZIZ, A. A. **The prediction of studentsâ^{TM} academic performance using classification data mining techniques**. *Applied Mathematical Sciences*, 9(129):6415–6426, 2015.

[10] AHMED, A. B. E. D.; ELARABY, I. S. **Data mining: A prediction for student's performance using classification method**. *World Journal of Computer Application and Technology*, 2(2):43–47, 2014.

[11] ANDERSON, D. R.; SWEENEY, D. J.; WILLIAMS, T. A. **Estatística Aplicada a Administracão e Economia**. Thomson Learning, 2007, 2007.

[12] ANKERST, M.; BREUNIG, M. M.; KRIEGEL, H.-P.; SANDER, J. **Optics: Ordering points to identify the clustering structure**. In: *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, SIGMOD '99, p. 49–60, New York, NY, USA, 1999. ACM.

[13] ASHOUR, W.; SUNOALLAH, S. **Multi Density DBSCAN**, p. 446–453. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[14] ASIF, R.; MERCERON, A.; PATHAN, M. K. **Predicting student academic performance at degree level: a case study**. *International Journal of Intelligent Systems and Applications*, 7(1):49, 2014.

[15] BAKER, R. S.; YACEF, K. **The state of educational data mining in 2009: A review and future visions**. *JEDM-Journal of Educational Data Mining*, 1(1):3–17, 2009.

[16] BALDI, P.; BRUNAK, S.; CHAUVIN, Y.; ANDERSEN, C. A. F.; NIELSEN, H. **Assessing the accuracy of prediction algorithms for classification: an overview**. *Bioinformatics*, 16(5):412–424, 2000.

[17] BARKER, K.; TRAFALIS, T.; RHOADS, T. R. **Learning from student data**. In: *Systems and Information Engineering Design Symposium, 2004. Proceedings of the 2004 IEEE*, p. 79–86. IEEE, 2004.

[18] BELLMAN, R. **Adaptative control processes**, 1961.

[19] BERTHOLD, M.; (EDS.), D. J. H. **Intelligent Data Analysis**. Springer-Verlag Berling Heidelberg, 2010.

[20] BISHOP, C. **Pattern Recognition and Machine Learning**. Information Science and Statistics. Springer, 2006.

[21] BOUCKAERT, R.; FRANK, E.; HALL, M.; KIRKBY, R.; REUTEMANN, P.; SEEWALD, A.; SCUSE, D. **â^{TM}weka manualâ^{TM}, university of wakiato**. Technical report, version 3-6-0, 2008.

[22] BOULIC, R.; RENAULT, O. **3d hierarchies for animation**. In: Magnenat-Thalmann, N.; Thalmann, D., editors, *New Trends in Animation and Visualization*. John Wiley & Sons ltd., 1991.

[23] BRADLEY, A. P. **The use of the area under the roc curve in the evaluation of machine learning algorithms**. *Pattern recognition*, 30(7):1145–1159, 1997.

[24] BRESFELEAN, V. P.; BRESFELEAN, M.; GHISOIU, N.; COMES, C.-A. **Determining studentsâ^{TM} academic failure profile founded on data mining methods**. In: *Information Technology Interfaces, 2008. ITI 2008. 30th International Conference on*, p. 317–322. IEEE, 2008.

[25] CARNAHAN, B.; ÉRARD MEYER.; KUNTZ, L.-A. **Comparing statistical and machine learning classifiers: Alternatives for predictive modeling in human factors research.** p. 408–423, 2003.

[26] CHAPMAN, P.; CLINTON, J.; KERBER, R.; KHABAZA, T.; REINARTZ, T.; SHEARER, C.; WIRTH, R. **Crisp-dm 1.0 step-by-step data mining guide**. Technical report, The CRISP-DM consortium, August 2000.

[27] CHAWLA, N. V.; BOWYER, K. W.; HALL, L. O.; KEGELMEYER, W. P. **Smote: synthetic minority over-sampling technique**. *Journal of artificial intelligence research*, 16:321–357, 2002.

[28] COPPIN, B. **InteligÃªncia Artificial**. Jones Barlett Publishers, 2004.

[29] CORTES, C.; VAPNIK, V. **Support-vector networks**. *Mach. Learn.*, 20(3):273–297, Sept. 1995.

[30] CORTEZ, P.; SILVA, A. M. G. **Using data mining to predict secondary school student performance**. 2008.

[31] CUNNINGHAM, S. J. **Machine learning and statistics: A matter of perspective**. *Waikato*, 1995.

[32] DASH, MANORANJAN ADN LIU, H. **Consistency-based search in feature selection**. *Artificial Intelligence*, 151:155–176, 2003.

[33] DE BAKER, R. S. J.; ISOTANI, S.; DE CARVALHO, A. M. J. B. **Mineraçao de dados educacionais: Oportunidades para o brasil**. *Revista Brasileira de Informática na Educação*, 19(2), 2011.

[34] DEJAEGER, K.; GOETHALS, F.; GIANGRECO, A.; MOLA, L.; BAESENS, B. **Gaining insight into student satisfaction using comprehensible data mining techniques**. *European Journal of Operational Research*, 218(2):548–562, 2012.

[35] DEKKER, G.; PECHENIZKIY, M.; VLEESHOUWERS, J. **Predicting students drop out: A case study**. In: *Educational Data Mining 2009*, 2009.

[36] DEKKER, G.; PECHENIZKIY, M.; VLEESHOUWERS, J. **Predicting students drop out: a case study**. In: Barnes, T.; Desmarais, M.; Romero, C.; Ventura, S., editors, *Proceedings of the 2nd International Conference on Educational Data Mining (EDM'09)*, p. 41–50, Cordoba, Spain, July 2009.

[37] DELEN, D. **A comparative analysis of machine learning techniques for student retention management**. *Decision Support Systems*, 49(4):498–506, 2010.

[38] DEMŠAR, J. **Statistical comparisons of classifiers over multiple data sets**. *Journal of Machine learning research*, 7(Jan):1–30, 2006.

[39] DENIL, M.; MATHESON, D.; DE FREITAS, N. **Narrowing the gap: Random forest in theory and in practice.** In: *ICML*, p. 665–673, 2014.

[40] DITTERICH, T. G. **Approximate statistical tests for comparing supervised classification learning algorithms**. *Neural Comput.*, 10(7):1895–1923, Oct. 1998.

[41] DOANE, D. **Aesthetic frequency classifications**. *The American Statistician*, 30(4):181 – 183, 1976.

[42] DRAGICEVIC, M.; BACH, M. P.; Å IMICEVIC, V. **Improving university operations with data mining: Predicting student performance**. *International Journal of Social, Management, Economics and Business Engineering*, 8(4):1094 – 1099, 2014.

[43] DRAGIČEVIĆ, M.; BACH, M. P.; ŠIMIČEVIĆ, V. **Improving university operations with data mining: Predicting student performance**. *environment*, 1:2, 2014.

[44] DUDA, R. O.; HART, P. E.; OTHERS. **Pattern classification and scene analysis**, volume 3. Wiley New York, 1973.

[45] DUDA, R.; HART, P.; STORK, D. **Pattern Classification**. Wiley, 2012.

[46] E., R.; KNIGHT, K. **Artificial Intelligence (second edition)**. McGraw-Hill, 1991.

[47] FANGMIN, N.; BINGHUI, H. **A Cobweb Model and it's Application**, p. 337–338. Springer London, London, 1991.

[48] FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. **The kdd process for extracting useful knowledge from volumes of data**. *Commun. ACM*, 39(11):27–34, Nov. 1996.

[49] FERREIRA, D. F. **Estatística Básica**. Editora UFLA, 2009.

[50] FISHER, D. H. **Knowledge acquisition via incremental conceptual clustering**. *Machine Learning*, 2(2):139–172, 1987.

[51] FOX, J. **The R Commander: A basic statistics graphical user interface to R**. *Journal of Statistical Software*, 14(9):1–42, 2005.

[52] GANSNER, E. R.; NORTH, S. C. **An open graph visualization system and its applications to software enginnering**. *SOFTWARE - PRACTICE AND EXPERIENCE*, 30:1203–1233, 2000.

[53] GARCÍA-SAIZ, D.; ZORRILLA, M. E. **Comparing classification methods for predicting distance students' performance**. In: *Proceedings of the Second Workshop on Applications of Pattern Analysis, WAPA 2011, Castro Urdiales, Spain, October 19-21, 2011*, p. 26–32, 2011.

[54] GOGA, M.; KUYORO, S.; GOGA, N. **A recommender for improving the student academic performance**. *Procedia-Social and Behavioral Sciences*, 180:1481–1488, 2015.

[55] GOOD, I. J.; HACKING, I.; JEFFREY, R.; T(Ö)RNEBOHM, H. **The estimation of probabilities: An easy on modern bayesina methods**. 1966.

[56] GROTHENDIECK, G. **gsubfn: Utilities for strings and function arguments.**, 2014. R package version 0.6-6.

[57] GROTHENDIECK, G. **sqldf: Perform SQL Selects on R Data Frames**, 2014. R package version 0.4-10.

[58] GUARÍN, C. E. L.; GUZMÁN, E. L.; GONZÁLEZ, F. A. **A model to predict low academic performance at a specific enrollment using data mining**. *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, 10(3):119–125, 2015.

[59] GUNDUZ, N.; FOKOUE, E. **Uci machine learning repository**, 2013.

[60] GÜNER, N.; YALDIR, A.; GÜNDÜZ, G.; ÇOMAK, E.; TOKAT, S.; İPLIKÇI, S. **Predicting academically at-risk engineering students: A soft computing application**. *Acta Polytechnica Hungarica*, 11(5):199–216, 2014.

[61] GUPTA, P.; MEHROTRA, D.; SHARMA, T. **Identifying knowledge indicators in higher education organization**. *Procedia Computer Science*, 46:449 – 456, 2015. Proceedings of the International Conference on Information and Communication Technologies, {ICICT} 2014, 3-5 December 2014 at Bolgatty Palace amp; Island Resort, Kochi, India.

[62] GUPTA, P.; MEHROTRA, D.; SHARMA, T. **Identifying knowledge indicators in higher education organization**. *Procedia Computer Science*, 46:449–456, 2015.

[63] GURULER, H.; ISTANBULLU, A.; KARAHASAN, M. **A new student performance analysing system using knowledge discovery in higher educational databases**. *Computers & Education*, 55(1):247–254, 2010.

[64] HACKELING, G. **Mastering Machine Learning with scikit-learn**. Packt Publishing, 2014.

[65] HALL, M. A.; SMITH, L. A. **Practical feature subset selection for machine learning**. 1998.

[66] HÄMÄLÄINEN, W.; SUHONEN, J.; SUTINEN, E.; TOIVONEN, H. **Data mining in personalizing distance education courses**. In: *Proceedings of the 21st ICDE World Conference on Open Learning and Distance Education*, p. 18–21, 2004.

[67] HÄMÄLÄINEN, W.; VINNI, M. **Comparison of machine learning methods for intelligent tutoring systems**. In: *Intelligent Tutoring Systems*, p. 525–534. Springer, 2006.

[68] HAN, J.; KAMBER, M.; PEI, J. **Data Mining:Concepts and Techniques: Concepts andd Techniques**. The Morgan Kaufman Series in Data Management Systems. Elsevier Science, 2011.

[69] HAN, J.; KAMBER, M. **Data Mining: Concepts and Techniques, 2nd ed.The Morgan Kaufmann Series in Data Management Systems, Jim Gray, Series Editor**. Morgan Kaufmann Publishers, 2006.

[70] HAN, J.; PEI, J.; YIN, Y. **Mining frequent patterns without candidate generation**. In: *ACM Sigmod Record*, volume 29, p. 1–12. ACM, 2000.

[71] HART, P. E.; STORK, D. G.; DUDA, R. O. **Pattern classification**. *John Willey & Sons*, 2001.

[72] HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. **The Elements of Statistical Learning: Dataminig, Inference, and Prediction**. Springer, 2009.

[73] HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. **Mastering Machine Learning with scikit-learn**. Springer, 2013.

[74] HAUGELAND, J. **Artificial Intelligence: the Very Idea.** MIT Press, 1985.

[75] HAYKIN, S. **Neural Networks and Learning Machines**. Número v. 10 em Neural networks and learning machines. Prentice Hall, 2009.

[76] HEREDIA, D.; AMAYA, Y.; BARRIENTOS, E. **Student dropout predictive model using data mining techniques**. *IEEE Latin America Transactions*, 13(9):3127–3134, Sept 2015.

[77] HERNANDES. **AvaliaÃ§Ã£o da ferramenta start utilizando o modelo tam e o paradigma gqm**. In: *Proceedings of 7th Experimental Software Engineering Latin American Workshop (ESELAW 2010*, p. 30–39, 2010.

[78] HERZOG, S. **Estimating student retention and degree-completion time: Decision trees and neural networks vis-à-vis regression**. *New Directions for Institutional Research*, 2006(131):17–33, 2006.

[79] HU, Y.-H.; LO, C.-L.; SHIH, S.-P. **Developing early warning systems to predict studentsâ™ online learning performance**. *Computers in Human Behavior*, 36:469–478, 2014.

[80] HUANG, S.; FANG, N. **Predicting student academic performance in an engineering dynamics course: A comparison of four types of predictive mathematical models**. *Computers & Education*, 61:133–145, 2013.

[81] HUNTER, J. D. **Matplotlib: A 2d graphics environment**. *Computing In Science & Engineering*, 9(3):90–95, 2007.

[82] IAM-ON, N.; BOONGOEN, T. **Improved student dropout prediction in thai university using ensemble of mixed-type data clusterings**. *International Journal of Machine Learning and Cybernetics*, p. 1–14, 2015.

[83] IFENTHALER, D.; WIDANAPATHIRANA, C. **Development and validation of a learning analytics framework: Two case studies using support vector machines**. *Technology, Knowledge and Learning*, 19(1-2):221–240, 2014.

[84] INEP. **Instituto nacional de estudos e pesquisas**, nov 2015.

[85] INEP. **Taxas de rendimento escolar**. Inep, Distrito Federal, Brasil, 2016.

[86] INF. **Ciências da computação | instituto de informática - ufg**. http://portal.inf.ufg.br, Apr. 2015. April 05, 2015.

[87] INF. **Ciências da computação | instituto de informática - ufg**. http://portal.inf.ufg.br, Apr. 2015. April 05, 2015.

[88] INSTITUTE, A. N. S. **American national standard for information systems: database language — SQL: ANSI X3.135-1992**. pub-ANSI, pub-ANSI:addr, 1992. Revision and consolidation of ANSI X3.135-1989 and ANSI X3.168-1989, Approved October 3, 1989.

[89] JAMES, G.; WITTEN, D.; HASTIE, T.; TIBSHIRANI, R. **An Introduction to Statistical Learning: with Applications in R**. Springer Texts in Statistics. Springer New York, 2013.

[90] JARMAN, K. H. **The Art of Data Analysis: How to Answer Almost Any Question Using Basic Statistics**. John Wiley & Sons, Inc., 2013.

[91] JEEVALATHA, T.; ANANTHI, N.; KUMAR, D. S. **Performance analysis of undergraduate students placement selection using decision tree algorithms**. *International Journal of Computer Applications*, 108(15), 2014.

[92] JINDAL, R.; BORAH, M. D. **A survey on educational data mining and**. *International Journal of Database Management Systems(IJDMS)*, 5(3), Jun 2013.

[93] JOHNSON, R.; WICHER, D. **Applied multivariate statistical analysis**. Person Education, Inc., 2007.

[94] JONES, E.; OLIPHANT, T.; PETERSON, P.; OTHERS. **SciPy: Open source scientific tools for Python**, 2001–. [Online; accessed 2015-06-22].

[95] KANUNGO, T.; MOUNT, D. M.; NETANYAHU, N. S.; PIATKO, C. D.; SILVERMAN, R.; WU, A. Y. **An efficient k-means clustering algorithm: Analysis and implementation**. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):881–892, July 2002.

[96] KASIH, J.; AYUB, M.; SUSANTO, S. **Predicting studentsâ^{TM} final passing results using the classification and regression trees (cart) algorithm**. *World Transactions on Engng. and Technol. Educ*, 11(1):46–49, 2013.

[97] KATES, L.; PETZOLDT, T. **proto: Prototype object-based programming**, 2012. R package version 0.3-10.

[98] KIRA, K.; RENDELL, L. A. **The feature selection problem: Traditional methods and a new algorithm**. In: *AAAI*, volume 2, p. 129–134, 1992.

[99] KIRA, K.; RENDELL, L. A. **A practical approach to feature selection**. In: *Proceedings of the ninth international workshop on Machine learning*, p. 249–256, 1992.

[100] KIRK, M. **Thoughtful Machine Learning: A Test-Driven Approach**. O'Reilly Media, 2014.

[101] KITCHENHAM, B. **Procedures for performing systematic reviews**. *keele,UK,keele University*, 33(2004):1–26, 2004.

[102] KITCHENHAM, B.; CHARTERS, S. **Guidelines for performing systematic literature reviews in software engineering**. Technical Report EBSE 2007-001, 2007.

[103] KNUTH, D. E. **The TEX Book**. Addison-Wesley, 15th edition, 1984.

[104] KOTSIANTIS, S.; PATRIARCHEAS, K.; XENOS, M. **A combinational incremental ensemble of classifiers as a technique for predicting studentsâ™ performance in distance education**. *Knowledge-Based Systems*, 23(6):529–535, 2010.

[105] KOTSIANTIS, S.; PIERRAKEAS, C.; PINTELAS, P. **Preventing student dropout in distance learning using machine learning techniques**. In: Palade, V.; Howlett, R.; Jain, L., editors, *Knowledge-Based Intelligent Information and Engineering Systems*, volume 2774 de **Lecture Notes in Computer Science**, p. 267–274. Springer Berlin Heidelberg, 2003.

[106] KOVACIC, Z. **Early prediction of student success: Mining students' enrolment data.** 2010.

[107] KOVAVI, R.; JOHN, G. H. **Wrappers for feature subset selection**. *Artificial Intelligence*, 97(1):273–324, 1997.

[108] KUNCHEVA, L. I. **Combining pattern classifiers: methods and algorithms**. John Wiley & Sons, 2004.

[109] LANGLEY, P.; IBA, W.; THOMPSON, K. **An analysis of bayesian classifiers**. In: *Aaai*, volume 90, p. 223–228, 1992.

[110] LARA, J. A.; LIZCANO, D.; MARTÍNEZ, M. A.; PAZOS, J.; RIERA, T. **A system for knowledge discovery in e-learning environments within the european higher education area–application to student data from open university of madrid, udima**. *Computers & Education*, 72:23–36, 2014.

[111] LAROSE, D. T. **Discovering Knowledge in Data: An Introduction to Data Mining**. NJ:Wiley, 2005.

[112] LEE, S. S. **Noisy replication in skewed binary classification**. *Computational statistics & data analysis*, 34(2):165–191, 2000.

[113] LONGNECKER, M. **An Introduction to Statistical Methods and Data Analysis**. Brooks/Cole Cengage Learning, 2010.

[114] LU, Z.; SZAFRON, D.; GREINER, R.; LU, P.; WISHART, D.; POULIN, B.; ANVIK, J.; MACDONELL, C.; EISNER, R. **Predicting subcellular localization of proteins using machine-learned classifiers**. *Bioinformatics*, 20(4):547–556, 2004.

[115] LUAN, J. **Data mining and its applications in higher education**. *New directions for institutional research*, 2002(113):17–36, 2002.

[116] LYKOURENTZOU, I.; GIANNOUKOS, I.; NIKOLOPOULOS, V.; MPARDIS, G.; LOUMOS, V. **Dropout prediction in e-learning courses through the combination of machine learning techniques**. *Comput. Educ.*, 53(3):950–965, Nov. 2009.

[117] LYKOURENTZOU, I.; GIANNOUKOS, I.; NIKOLOPOULOS, V.; MPARDIS, G.; LOUMOS, V. **Dropout prediction in e-learning courses through the combination of machine learning techniques**. *Computers & Education*, 53(3):950–965, 2009.

[118] MA, Y.; LIU, B.; WONG, C. K.; YU, P. S.; LEE, S. M. **Targeting the right students using data mining**. In: *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, p. 457–464. ACM, 2000.

[119] MÁRQUEZ-VERA, C.; CANO, A.; ROMERO, C.; NOAMAN, A. Y. M.; MOUSA FARDOUN, H.; VENTURA, S. **Early dropout prediction using data mining: a case study with high school students**. *Expert Systems*, 33(1):107–124, 2016.

[120] MARQUEZ-VERA, C.; MORALES, C. R.; SOTO, S. V. **Predicting school failure and dropout by using data mining techniques**. *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, 8(1):7–14, 2013.

[121] MARROCO, J. **AnÃ¡lise estatÃstica com utilizaÃ§Ã£o do SPSS.** EdiÃ§ÃμesS-$labo$, 2007.

[122] MARTINHO, V. R. D. C.; NUNES, C.; MINUSSI, C. R. **An intelligent system for prediction of school dropout risk group in higher education classroom based on artificial neural networks**. In: *2013 IEEE 25th International Conference on Tools with Artificial Intelligence*, p. 159–166. IEEE, 2013.

[123] MARTINS, C. **Manual de Análise de Dados Quantitativos com recurso ao IBM SPSS: Saber decidir, fazer ,interpretar e redigir**. psiquilibriosedições, 1th edition, 2011.

[124] MCKINNEY, W. **Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython**. O'Reilly Media, 2012.

[125] Michie, D.; Spiegelhalter, D. J.; Taylor, C.; Campbell, J., editors. **Machine Learning, Neural and Statistical Classification**. Ellis Horwood, Upper Saddle River, NJ, USA, 1994.

[126] MILLER, T. W. **Modeling Techniques in Predictive Analitycs with Python and R: A Guide to Data Science**. Person Education, Inc., 2014.

[127] MINAEI-BIDGOLI, B.; KASHY, D. A.; KORTEMEYER, G.; PUNCH, W. **Predicting student performance: an application of data mining methods with an educational web-based system**. In: *Frontiers in education, 2003. FIE 2003 33rd annual*, volume 1, p. T2A–13. IEEE, 2003.

[128] MITCHELL, T. **Machine Learning**. McGraw-Hill International Editions. McGraw-Hill, 1997.

[129] MUEHLENBROCK, M. **Automatic action analysis in an interactive learning environment**. In: *The 12th international conference on artificial intelligence in education, AIED*, p. 73–80, 2005.

[130] NANDESHWAR, A.; MENZIES, T.; NELSON, A. **Learning patterns of university student retention**. *Expert Systems with Applications*, 38(12):14984–14996, 2011.

[131] NGHE, N. T.; JANECEK, P.; HADDAWY, P. **A comparative analysis of techniques for predicting academic performance**. In: *Frontiers In Education Conference-Global Engineering: Knowledge Without Borders, Opportunities Without Passports, 2007. FIE'07. 37th Annual*, p. T2G–7. IEEE, 2007.

[132] NISBET, R.; ELDER, J.; MINER, G. **Statistical Analysis and Data Mining Applications**. Elsevier, 2009.

[133] OBLINGER, D.; CAMPBEL, J. **Academic analytics**. *educase*, 2007.

[134] ØHRN, A. **Rosetta technical reference manual**. *Department of Computer and Information Science, Norwegian University of Science and Technology (NTNU), Trondheim, Norway*, p. 1–66, 2000.

[135] OKOLI, C.; SCHABRAM, K. **A guide to conducting a systematic literature review of information systems research**. *Sprouts Work. Pap. Inf. Syst*, 10:26, 2010.

[136] OLSHEN, L.; STONE, C. J.; OTHERS. **Classification and regression trees**. *Wadsworth International Group*, 93(99):101, 1984.

[137] ON DATABASES, R. S. I. G. **DBI: R Database Interface**, 2014. R package version 0.3.1.

[138] OTT, R.; LONGNECKER, M. **An Introduction to Statistical Methods and Data Analysis**. Available 2010 Titles Enhanced Web Assign Series. Cengage Learning, 2008.

[139] PAL, S. **Mining educational data using classification to decrease dropout rate of students**. *arXiv preprint arXiv:1206.3078*, 2012.

[140] PAVLIK JR, P. I.; CEN, H.; KOEDINGER, K. R. **Learning factors transfer analysis: Using learning curve analysis to automatically generate domain models**. In: *Educational Data Mining 2009*, 2009.

[141] PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. **Scikit-learn: Machine learning in Python**. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[142] PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. **Scikit-learn: Machine learning in Python**. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[143] PEÑA-AYALA, A. **Educational data mining: A survey and a data mining-based analysis of recent works**. *Expert systems with applications*, 41(4):1432–1462, 2014.

[144] PÉREZ, F.; GRANGER, B. E. **IPython: a system for interactive scientific computing**. *Computing in Science and Engineering*, 9(3):21–29, May 2007.

[145] PLATT, J. C. **12 fast training of support vector machines using sequential minimal optimization**. *Advances in kernel methods*, p. 185–208, 1999.

[146] POOLE, D.; MACKWORTH, A.; GOEBEL, R. **Computacional intelligence. A logical approach**. Oxford University Press, 1998.

[147] PRADEEP, A.; DAS, S.; KIZHEKKETHOTTAM, J. J. **Students dropout factor prediction using edm techniques**. In: *Soft-Computing and Networks Security (ICSNS), 2015 International Conference on*, p. 1–7. IEEE, 2015.

[148] PRODIRH. **Pró-reitoria de desenvolvimento institucional e recursos humanos**, nov 2015.

[149] QUINLAN, J. R. **induction of decision trees**. *Machine Learning*, 1(1):81–106, 1986.

[150] QUINLAN, J. R. **C4.5: Programs for Machine Learning**. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[151] QUINLAN, R. **Data mining tools see5 and c5. 0**. 2004.

[152] R CORE TEAM. **R: A language and Environment for Statistical Computing**. R Foundation for Statistical Computing, Vienna, Autria, 2014.

[153] RAHMAN, T. A.; RAHMAN, A. **Minimizing student attrition in higher learning institutions in malaysia using support vector machine**. *Journal of Theoretical and Applied Information Technology*, 71(3), 2015.

[154] RAMASWAMI, M.; BHASKARAN, R. **A study on feature selection techniques in educational data mining**. *arXiv preprint arXiv:0912.3924*, 2009.

[155] REGHA, R. S.; RANI, R. U. **A novel clustering based feature selection for classifying student performance**. *Indian Journal of Science and Technology*, 8(S7):135–140, 2015.

[156] ROKACH, L.; MAIMON, O. **Data mining with decision trees: theory and applications**. World scientific, 2014.

[157] ROMERO, C.; VENTURA, S. **Educational Data Mining: A Review of the State of the Art**. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 40(6):601–618, Nov. 2010.

[158] ROMERO, C.; VENTURA, S. **Educational data mining: A review of the state of the art**. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 40(6):601–618, Nov. 2010.

[159] ROMERO, C.; VENTURA, S. **Educational data mining: A review of the state of the art**. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 40(6):601–618, Nov. 2010.

[160] ROMERO, C.; VENTURA, S. **Educational data mining: A survey from 1995 to 2005**. *Expert systems with applications*, 33(1):135–146, 2007.

[161] ROMERO, C.; VENTURA, S.; ESPEJO, P. G.; HERVÁS, C. **Data mining algorithms to classify students**. In: *Educational Data Mining 2008*, 2008.

[162] ROSSUM, G. **Python reference manual**. Technical report, Amsterdam, The Netherlands, The Netherlands, 1995.

[163] RUMSEY, D. **Statistics Essentials for Dummies**. Wiley Publishing, Inc., 2010.

[164] RUSSELL, S. J.; PETER, N. **Artificial Intelligence**. Elsevier, 2013.

[165] SCOTT, D. W. **On optimal and data-based histograms**. *biometrika*, 66(3):605 – 610, 1979.

[166] SEABOLD, J.; PERKTOLD, J. **Statsmodels: Econometric and statistical modeling with python**. In: *Proceedings of the 9th Python in Science Conference*, 2010.

[167] SHALEENA, K. P.; PAUL, S. **Data mining techniques for predicting student performance**. In: *Engineering and Technology (ICETECH), 2015 IEEE International Conference on*, p. 1–3, March 2015.

[168] SHAW, E.; MARINI, J.; MATTERN, K. **Exploring the utility of advanced placement participation and performance in college admission decisions**. *Educational and Psychological Measurement*, 73(2):229 – 253, 2013.

[169] SILIPO, R.; ADAE, I.; HART, A.; BERTHOLD, M. **Seven techniques for dimensionality reduction**, 2014.

[170] SMITH, A.; JONES, B. **On the complexity of computing**. In: Smith-Jones, A. B., editor, *Advances in Computer Science*, p. 555–566. Publishing Press, 1999.

[171] SMOLA, A. **Advances in Large Margin Classifiers**. Neural information processing series. MIT Press, 2000.

[172] SOARES, C. **Is the uci repository useful for data mining?** In: *Progress in Artificial Intelligence*, p. 209–223. Springer, 2003.

[173] SOBCZAK, G.; PIKUŁA, M.; SYDOW, M. **AGNES: A Novel Algorithm for Visualising Diversified Graphical Entity Summarisations on Knowledge Graphs**, p. 182–191. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[174] SUPERBY, J.-F.; VANDAMME, J.; MESKENS, N. **Determination of factors influencing the achievement of the first-year university students using data mining methods**. In: *Workshop on Educational Data Mining*, p. 37–44, 2006.

[175] TAMHANE, A.; IKBAL, S.; SENGUPTA, B.; DUGGIRALA, M.; APPLETON, J. **Predicting student risks through longitudinal analysis**. In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, p. 1544–1552. ACM, 2014.

[176] TAN, P.-N.; STEINBACH, M.; KUMAR, V. **Introduction to Data Mining, (First Edition)**. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[177] TEKIN, A. **Early prediction of students' grade point averages at graduation: A data mining approach.** *Eurasian Journal of Educational Research*, 54:207–226, 2014.

[178] THAKAR, P. **Performance analysis and prediction in educational data mining: A research travelogue**. *arXiv preprint arXiv:1509.05176*, 2015.

[179] THAMMASIRI, D.; DELEN, D.; MEESAD, P.; KASAP, N. **A critical assessment of imbalanced class distribution problem: The case of predicting freshmen student attrition**. *Expert Systems with Applications*, 41(2):321–330, 2014.

[180] THEODORIDIS, S.; KOUTROUMBAS, K. **Pattern Recognition**. Pattern Recognition Series. Elsevier Science, 2006.

[181] THEODORIDIS, S.; KOUTROUMBAS, K. **Pattern Recognition Fourth Edition**. Elsevier, 2009.

[182] TOMIDA, K.; YAMAGUCHI, Q. **Using data mining to identify patterns of student voltage drop**. *KUROSHIO*, 8(2):138–146, 2015.

[183] UNMWE. **A university of national and world economy**, nov 2015.

[184] VEIGA, D.; AMBRÓSIO, A.; OLÍMPIO, N. **As mulheres no curso de ciência da computação da universidade federal de goiás**. In: *Actas del II Congreso de la Mujer Latinoamericana en la Computación*, 2010.

[185] WANG, W.; YANG, J.; MUNTZ, R. R. **Sting: A statistical information grid approach to spatial data mining**. In: *Proceedings of the 23rd International Conference on Very Large Data Bases*, VLDB '97, p. 186–195, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[186] WASKOM, M.; BOTVINNIK, O.; HOBSON, P.; COLE, J. B.; HALCHENKO, Y.; HOYER, S.; MILES, A.; AUGSPURGER, T.; YARKONI, T.; MEGIES, T.; COELHO, L. P.; WEHNER, D.; CYNDDL.; ZIEGLER, E.; DIEGO0020.; ZAYTSEV, Y. V.; HOPPE, T.; SEABOLD, S.; CLOUD, P.; KOSKINEN, M.; MEYER, K.; QALIEH, A.; ALLAN, D. **seaborn: v0.5.0 (november 2014)**, Nov. 2014.

[187] WICKHAM, H. **ggplot2: elegant graphics for data analysis**. Springer New York, 2009.

[188] WICKHAM, H.; JAMES, D. A.; FALCON, S. **RSQLite: SQLite Interface for R**, 2014. R package version 1.0.0.

[189] WINSTON, P. **Artificial Intelligence (Third Edition)**. Addison-Wesley, 1992.

[190] WU, X.; KUMAR, V. **The Top Ten Algorithms in Data Mining**. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series. CRC Press, 2009.

[191] YADAV, S. K.; BHARADWAJ, B.; PAL, S. **Data mining applications: A comparative study for predicting student's performance**. *arXiv preprint arXiv:1202.4815*, 2012.

[192] YADAV, S. K.; BHARADWAJ, B.; PAL, S. **Mining education data to predict student's retention: a comparative study**. *arXiv preprint arXiv:1203.2987*, 2012.

[193] YE, N.; OTHERS. **The handbook of data mining**, volume 24. Lawrence Erlbaum Associates, Publishers Mahwah, NJ/London, 2003.

[194] YUKSELTURK, E.; OZEKES, S.; TÜREL, Y. K. **Predicting dropout student: an application of data mining methods in an online education program**. *European Journal of Open, Distance and E-learning*, 17(1):118–133, 2014.

[195] ZAFRA, A.; VENTURA, S. **Multi-instance genetic programming for predicting student performance in web based educational environments**. *Applied Soft Computing*, 12(8):2693–2706, 2012.

[196] ZAKI, M. J. **Scalable algorithms for association mining**. *IEEE Transactions on Knowledge and Data Engineering*, 12(3):372–390, 2000.

[197] ZANG, W.; LIN, F. **Investigation of web-based teaching and learning by boosting algorithms**. In: *Information Technology: Research and Education, 2003. Proceedings. ITRE2003. International Conference on*, p. 445–449. IEEE, 2003.

[198] ZHANG, T.; RAMAKRISHNAN, R.; LIVNY, M. **Birch: An efficient data clustering method for very large databases**. *SIGMOD Rec.*, 25(2):103–114, June 1996.

[199] ZHANG, Y.; OUSSENA, S.; CLARK, T.; HYENSOOK, K. **Using data mining to improve student retention in he: a case study.** *Neurocomputing*, 1:190 – 197, 2010.

[200] ZIMMERMANN, J.; BRODERSEN, K. H.; HEINIMANN, H. R.; BUHMANN, J. M. **A model-based approach to predicting graduate-level performance using indicators of undergraduate-level performance**. *JEDM-Journal of Educational Data Mining*, 7(3):151–176, 2015.

APÊNDICE A

Códigos-Fonte

Segue neste apêndice os códigos-fonte utilizados e produzidos no decorrer deste trabalho. São expostosos os códigos fontes em Python para transformação dos dados de linhas para coluna.

A.1 Códigos-Fonte em python para transformação dos dados de linhas para colunas

Nesta seção apresenta-se o código-fonte para um pivot mais avançado para transformar linhas para colunas da tabela de desempenho de estudantes. Utilizando o SQL-ANSI puro não seria possível efetuar esta transformação, dessa forma optamos por utilizar a linguagem python em conjunto com a biblioteca pandas para efetuar esta tarefa.

Listing A.1: *pivot*

```
#Pacote para manipulação de dataframes
import pandas as pd
#Efetuar a conexao com o banco sqlite
import sqlite3
#Executar comandos sql
from pandas.io import sql
#Manipulação de matrizes
import numpy as np
# Geração de Gráficos
import matplotlib.pyplot as plt
pd.options.display.mpl_style = 'default'
from matplotlib import pyplot
%matplotlib inline

#formatação de número decimal
from decimal import Decimal
```

```
#Criação da conexão com a base de dados
conn =
    sqlite3.connect("C:\\Users\\admin\\Documents\\porthos\\DadosCS\\arquivo\\dados.db")

pc2 = sql.read_sql('select␣
    disciplina,codigo,nr_matricula,ano_oferta,semestre_oferta,nota_disciplina,\
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣frequencia,situacao␣from␣
    programacao_computadores_II␣where␣nr_matricula', conn)
pc1 = sql.read_sql('select␣
    disciplina,codigo,nr_matricula,ano_oferta,semestre_oferta,nota_disciplina,\
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣frequencia,situacao␣from␣
    programacao_computadores_I␣where␣nr_matricula', conn)
lm = sql.read_sql('select␣
    disciplina,codigo,nr_matricula,ano_oferta,semestre_oferta,nota_disciplina,\
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣frequencia,situacao␣from␣logica_matematica␣
    where␣nr_matricula', conn)
ed1 = sql.read_sql('select␣
    disciplina,codigo,nr_matricula,ano_oferta,semestre_oferta,nota_disciplina,\
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣frequencia,situacao␣from␣estrutura_dados_I␣
    where␣nr_matricula', conn)
ed2 = sql.read_sql('select␣
    disciplina,codigo,nr_matricula,ano_oferta,semestre_oferta,nota_disciplina,\
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣frequencia,situacao␣from␣estrutura_dados_II␣
    where␣nr_matricula', conn)
poo = sql.read_sql('select␣
    disciplina,codigo,nr_matricula,ano_oferta,semestre_oferta,nota_disciplina,\
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣frequencia,situacao␣from␣programacao_oo␣where␣
    nr_matricula', conn)

tc = sql.read_sql('select␣
    disciplina,codigo,nr_matricula,ano_oferta,semestre_oferta,nota_disciplina,\
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣frequencia,situacao␣from␣teoria_computacao␣
    where␣nr_matricula', conn)

def agrupa(x):


    d=''
    c=int(x['codigo'].values[0])
    if c==4202:
```

```
        d = 'pc1_'
    elif c==4204:
        d ='pc2_'
    elif c==154:
        d ='lm_'
    elif c==162:
        d ='poo_'
    elif c==168:
        d ='ed2_'
    elif c==4201:
        d ='ed1_'
    elif c==185:
        d = 'tc_'

    resultado=dict()
    semestre_aprovacao=None
    ano_aprovacao =None
    nota_primeira = x['nota_disciplina'].values[0]
    nota_aprovacao=None
    frequencia = x['frequencia'].values[0]

    if x['situacao'].values[len(x)-1]=='APV':
        nota_aprovacao=x['nota_disciplina'].values[len(x)-1]
        semestre_aprovacao=int(x['semestre_oferta'].values[len(x)-1])
        ano_aprovacao = int(x['ano_oferta'].values[len(x)-1])

        nova_sit = 1
    elif x['situacao'].values[len(x)-1]=='REF':

        nova_sit= 2
        ano_aprovacao = None
        nota_aprovacao = None
    elif x['situacao'].values[len(x)-1]=='REP':

        nova_sit= 3
        ano_aprovacao = None
        nota_aprovacao = None
    else:
        nova_sit = 4 # O aluno nao cursou nenhuma vez a disciplina
        ano_aprovacao = None
        nota_aprovacao = None
```

```
    #semestre_segunda=int(x['semestre_oferta'].values[len(x)-1])
    if len(x)>1:
        nota_segunda=(x['nota_disciplina'].values[1])
    else:
        nota_segunda=(x['nota_disciplina'].values[0])

    resultado.update({d+'ano_primeira': int(x['ano_oferta'].min()),
            d+'nota_primeira':nota_primeira,
            d+'ano_aprovacao':ano_aprovacao,
            d+'semestre_primeira':int(x['semestre_oferta'].values[0]),
            d+'semestre_aprovacao':semestre_aprovacao,
            d+'nota_aprovacao':nota_aprovacao,
            d+'qtd_cursou':x['situacao'].count(),
            'nr_matricula':x['nr_matricula'].values[0],
            d+'disciplina':str(x['disciplina'].values[0]),
            d+'codigo':int(x['codigo'].values[0]),
            d+'situacao_primeira':nova_sit,
            d+'frequencia':int(x['frequencia'].values[0])
            })
    return pd.Series(resultado, name='agrupa')

cursou_pc1=pc1.groupby(['nr_matricula'] ).apply(agrupa)
cursou_lm=lm.groupby(['nr_matricula'] ).apply(agrupa)
cursou_ed1=ed1.groupby(['nr_matricula'] ).apply(agrupa)
cursou_ed2=ed2.groupby(['nr_matricula'] ).apply(agrupa)
cursou_tc=tc.groupby(['nr_matricula'] ).apply(agrupa)
cursou_poo=poo.groupby(['nr_matricula'] ).apply(agrupa)

table_pc2 = pd.pivot_table(cursou_pc2,index='nr_matricula')
table_pc1 = pd.pivot_table(cursou_pc1,index='nr_matricula')
table_lm = pd.pivot_table(cursou_lm,index='nr_matricula')
table_ed1 = pd.pivot_table(cursou_ed1,index='nr_matricula')
table_ed2 = pd.pivot_table(cursou_ed2,index='nr_matricula')
table_tc = pd.pivot_table(cursou_tc,index='nr_matricula')
table_poo = pd.pivot_table(cursou_poo,index='nr_matricula')

a=pd.concat([table_pc1,table_pc2,table_lm,table_ed1,table_ed2,
table_tc,table_poo],axis=1)

#exportar o resultado para o excel
#a.to_excel('d://Desenvolvimento//todos.xls')
```

```
#exportar para a base de dados
a.tosql('teste',conn)
#print table_pc2
```