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

BRENO OLIVEIRA

Algoritmos de Aprendizado de Máquina na Predição e Avaliação de Evasão de Clientes em Ambiente de Produção

Goiânia
2021

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO (TECA) PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TESES

E DISSERTAÇÕES 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 direitos autorais, de acordo com a Lei 9.610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo das Teses e Dissertações disponibilizado na BDTD/UFG é de responsabilidade exclusiva do autor. Ao encaminhar o produto final, o autor(a) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do material bibliográfico

[X] Dissertação [ ] Tese

2. Nome completo do autor

Breno Oliveira

3. Título do trabalho

Algoritmos de Aprendizado de Máquina na Predição e Avaliação de Evasão de Clientes em Ambiente de Produção

4. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador)

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

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante:
a) consulta ao(à) autor(a) e ao(à) orientador(a);
b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo da tese ou dissertação.
O documento não será disponibilizado durante o período de embargo.
Casos de embargo:
- Solicitação de registro de patente;
- Submissão de artigo em revista científica;
- Publicação como capítulo de livro;
- Publicação da dissertação/tese em livro.

Obs. Este termo deverá ser assinado no SEI pelo orientador e pelo autor.

Documento assinado eletronicamente por **Anderson Da Silva Soares**, **Professor do Magistério Superior**, em 26/07/2021, às 10:47, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **BRENO OLIVEIRA**, **Discente**, em 27/07/2021, às 09:34, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de

outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **2232858** e o código CRC **C7EC1FB8**.

Referência: Processo nº 23070.032165/2021-55 SEI nº 2232858

BRENO OLIVEIRA

Algoritmos de Aprendizado de Máquina na Predição e Avaliação de Evasão de Clientes em Ambiente de Produção

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

Área de concentração: Ciência da Computação.

Orientador: Prof. Anderson da Silva Soares

Goiânia
2021

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

OLIVEIRA, BRENO
Algoritmos de Aprendizado de Máquina na Predição e Avaliação de Evasão de Clientes em Ambiente de Produção [manuscrito] / BRENO OLIVEIRA. - 2021.
82 f.

Orientador: Prof. Dr. ANDERSON DA SILVA SOARES.
Dissertação (Mestrado) - Universidade Federal de Goiás, Instituto de Informática (INF), Programa de Pós-Graduação em Ciência da Computação, Goiânia, 2021.

1. desvio de conceito. 2. auto machine learning. 3. dados em stream. 4. churn. I. DA SILVA SOARES, ANDERSON, orient. II. Título.

CDU 004

UNIVERSIDADE FEDERAL DE GOIÁS

INSTITUTO DE INFORMÁTICA

ATA DE DEFESA DE DISSERTAÇÃO

Ata nº **15** da sessão de Defesa de Dissertação de **Breno Oliveira**, que confere o título de Mestre em Ciência da Computação, na área de concentração em Ciência da Computação.

Aos dois dias do mês de julho de dois mil e vinte e um, a partir das catorze horas, via sistema de webconferência da RNP, realizou-se a sessão pública de Defesa de Dissertação intitulada **"Algoritmos de Aprendizado de Máquina na Predição e Avaliação de Evasão de Clientes em Ambiente de Produção"**. Os trabalhos foram instalados pelo Orientador, Professor Doutor Anderson da Silva Soares (INF/UFG) com a participação dos demais membros da Banca Examinadora: Professora Doutora Telma Woerle de Lima Soares (INF/UFG), membro titular interno; Professor Doutor Rafael Teixeira Sousa (UFMT), membro titular externo. A realização da banca ocorreu por meio de videoconferência, em atendimento à recomendação de suspensão das atividades presenciais na UFG emitida pelo Comitê UFG para o Gerenciamento da Crise COVID-19, bem como à recomendação de isolamento social da Organização Mundial de Saúde e do Ministério da Saúde para enfrentamento da emergência de saúde pública decorrente do novo coronavírus. Durante a arguição os membros da banca não fizeram sugestão de alteração do título do trabalho. A Banca Examinadora reuniu-se em sessão secreta a fim de concluir o julgamento da Dissertação, tendo sido o candidato **aprovado** pelos seus membros. Proclamados os resultados pelo Professor Doutor Anderson da Silva Soares, Presidente da Banca Examinadora, foram encerrados os trabalhos e, para constar, lavrou-se a presente ata que é assinada pelos Membros da Banca Examinadora, aos dois dias do mês de julho de dois mil e vinte e um.

TÍTULO SUGERIDO PELA BANCA

Documento assinado eletronicamente por **Anderson Da Silva Soares**, **Professor do Magistério Superior**, em 02/07/2021, às 15:17, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Telma Woerle De Lima Soares**, **Professora do Magistério Superior**, em 02/07/2021, às 15:17, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **RAFAEL TEIXEIRA SOUSA**, **Usuário Externo**, em 02/07/2021, às 15:18, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **BRENO OLIVEIRA**, **Discente**, em 02/07/2021, às 15:49, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Fábio Moreira Costa**, **Coordenador de Pós-graduação**, em 05/07/2021, às 10:30, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do

Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **2158316** e o código CRC **AAAC0A11**.

Referência: Processo nº 23070.032165/2021-55 SEI nº 2158316

BRENO OLIVEIRA

Algoritmos de Aprendizado de Máquina na Predição e Avaliação de Evasão de Clientes em Ambiente de Produção

Dissertação defendida no Programa de Pós–Graduação do Instituto de Informática da Universidade Federal de Goiás como requisito parcial para obtenção do título de Mestre em Ciência da Computação, aprovada em 08 de Junho de 2021, pela Banca Examinadora constituída pelos professores:

Prof. Anderson da Silva Soares
Instituto de Informática – UFG
Presidente da Banca

Prof. Dr. Telma Woerle de Lima Soares
Instituto de Informática – UFG

Prof. Dr. Rafael Teixeira Sousa
ICET – UFMT

Breno Oliveira

Graduou-se em Administração pela Faculdade Padrão, especializou-se em Planejamento de Negócios pela Universidade Estadual de Goiás e em Gestão de Negócios, Controladoria e Finanças pelo Instituto de Pós Graduação de Goiás. Durante o Mestrado, na UFG - Universidade Federal de Goiás, foi bolsista da iniciativa privada, participou de projetos junto ao Centro de Excelência em Inteligência Artificial, conquistou o primeiro lugar, trabalho eleito por júri convidado, no 1o Workshop de Inteligência Artificial organizado pelo Instituto de Informática da UFG, com o projeto para identificação de clientes com propensão ao cancelamento de serviço de assinatura.

Dedico este trabalho ao meu irmão e amigo de uma vida inteira, Ulisses Cabral de Moura, por se mostrar o maior incentivador durante toda esta evolução, se tornando essencial a cada momento de dificuldade, de alegria e de conquista.

Agradecimentos

Agradeço primeiramente ao professor e orientador Anderson Soares, que serviu de farol direcionador de todo o esforço despendido durante o processo de aprendizagem e evolução, pela paciência e zelo atencioso em todos os momentos de dúvida e busca por conhecimento. Agradeço ao Grupo Jaime Câmara pela disponibilização dos dados necessários para o desenvolvimento deste projeto e, aos gestores Breno Machado, Marisol Sanchez Lloris e Paulo César Pansini, por permitirem que o ambiente de trabalho se tornasse fonte de conhecimento e aprimoramento pessoal e profissional. Agradeço aos demais professores e colegas com quem tive o prazer de compartilhar as aulas durante esse percurso. Ademais, agradeço à todos que, de forma direta ou indireta, contribuíram para a conclusão deste projeto, que finda mais um importante ciclo pessoal e acadêmico.

Resumo

Oliveira, Breno. **Algoritmos de Aprendizado de Máquina na Predição e Avaliação de Evasão de Clientes em Ambiente de Produção**. Goiânia, 2021. 81p. Dissertação de Mestrado. Instituto de Informática, Universidade Federal de Goiás.

O desenvolvimento de soluções de aprendizado de máquina prevê diversas etapas bem estabelecidas, sendo que os estudos científicos possuem uma concentração em etapas como engenharia de dados, treinamento do modelo e métricas de avaliação de desempenho. O advento da implantação de soluções de aprendizado de máquina em ambientes empresariais em um nível sem precedentes inspira a revisitação de alguns problemas anteriormente apontados na literatura, porém pouco explorados como o monitoramento e avaliação da deterioração da solução ao longo do tempo. Durante o treinamento dos modelos de aprendizado de máquina, supõe-se que os dados não vistos pelo modelo em produção apresentem a mesma distribuição dos dados utilizados durante a etapa de treinamento. Modelos em produção podem perder desempenho à medida que os dados sofram alterações com o passar do tempo. Este fenômeno é definido na literatura como desvio de conceito. Nesse contexto, este trabalho propõe uma metodologia que utiliza *Auto Machine Learning* com aprendizado de dados em *stream* capaz de mitigar eventuais desvios de conceito que possam surgir nos modelos implementados em ambiente de produção. Foram utilizados dados reais de um problema de evasão de clientes (*Churn*) de um jornal de grande circulação regional. Foram implementados três modelos de aprendizado de máquina utilizando duas metodologias: a metodologia proposta denominada *autoML-DS* e a metodologia de referência que faz uso de retreinamento convencional dos modelos. Os resultados demonstraram que a metodologia de referência apresenta perdas de desempenho dos modelos implementados enquanto o *autoML-DS* tem sua capacidade preditiva preservada. O *autoML-DS* foi capaz de adaptar os modelos ao longo do tempo, sem a necessidade da realização de um retreino completo, mantendo pequenas variações na proporção de erros.

Palavras–chave

<desvio de conceito, auto *machine learning*, dados em *stream*, *churn*>

Abstract

Oliveira, Breno. **Machine Learning Algorithms in Predicting and Evaluating Customer Evasion in a Production Environment**. Goiânia, 2021. 81p. MSc. Dissertation. Instituto de Informática, Universidade Federal de Goiás.

The development of machine learning solutions involves several well-established stages. However, scientific studies have a concentration on stages such as data engineering, model training, and performance evaluation metrics. The advent of machine learning solutions implementation in business environments at an unprecedented level inspires the revisiting of some problems previously mentioned in the literature, but little explored. Among them, monitoring and evaluating the deterioration of the solution over time. During machine learning models training, it is assumed that the data not seen by the model in production presents the same distribution as the data used during the training stage. However, production models can decrease/lose performance as data changes over time. This phenomenon is defined in the literature as concept deviation. In this context, this work proposes a methodology that uses Auto Machine Learning with data stream learning capable of mitigating eventual concept deviations that may arise in the models implemented in a production environment. Real data from a customer avoidance problem (Churn) of a large-circulation regional newspaper were used. Three machine learning models were implemented using two methodologies: the proposed methodology called autoML-DS and the reference methodology that makes use of conventional model retraining. The results showed that the reference methodology presents performance losses of the implemented models, while the autoML-DS has its predictive capacity preserved. AutoML-DS was able to adapt the models over time, without having to perform a complete retraining, keeping small variations in the error rate.

Keywords

<concept drift, auto machine learning, data stream, churn>

Sumário

CAPÍTULO 1

Introdução

A era digital e o avanço dos algoritmos de aprendizado de máquina tem impulsionado a implantação de soluções em ambientes de produção de diversas organizações, sendo utilizados como fonte geradora de informação que direcionam ações, [Lorica e Paco 2018]. De acordo com [Lorica e Paco 2018], o uso de soluções de Aprendizado de Máquina (ML - *Machine Learning*) em produção começou perto da virada do século, mas levou cerca de 20 anos para que a prática se tornasse popular em todo o setor.

O aprendizado de máquina (*Machine Learning*) tornou-se uma ferramenta essencial para diversos setores da economia. Cada vez mais utilizada em organizações públicas e privadas, técnicas de *Machine Learning* (ML) vem contribuindo para a jornada de transformação digital, e consequentemente para o desenvolvimento de soluções de negócio a partir dos dados disponíveis [Polyzotis et al. 2018]. Com o poder computacional atualmente disponível, a grande quantidade de dados gerados e armazenados e o surgimento de algoritmo mais eficientes, o aprendizado de máquina vem ganhando maior espaço dentro das organizações para otimizar as operações existentes e adicionar novos serviços, dando à elas uma vantagem competitiva relevante. Esse aumento no uso de ML ajuda a estabelecer a necessidade de um entendimento maior das etapas de modelagem de aplicações de aprendizado de máquina [Clark 2018].

As etapas de desenvolvimento de uma solução comercial de aprendizado de máquina pode ser divididas em pelo menos cinco estágios:

1. Preparação e modelagem de dados;
2. Construção do modelo;
3. Avaliação;
4. Implantação;
5. Monitoramento e reavaliação.

A literatura de aprendizado de máquina apresenta uma concentração de trabalhos nos três primeiros estágios citados anteriormente. Os esforços científicos estavam centrados na evolução de técnicas de seleção de variáveis, transformação de dados [Kumar e Minz 2014], novos modelos de otimização para aprendizado

[Kotsiantis, Zaharakis e Pintelas] e novas métricas, metodologias e processos de avaliação [Hossin e Sulaiman 2015]. Para alguns tipos de problemas envolvendo aprendizado de máquina, os dados podem sofrer uma alteração significativa ao longo do tempo, levando os modelos obtidos a partir de um treino a se tornarem obsoletos. No aprendizado de máquina e na mineração de dados, o autor [Žliobaitė, Pechenizkiy e Gama 2016] define esse fenômeno como desvio de conceito.

A solução tradicionalmente comum na deterioração de modelos de aprendizado de máquina é o retreino em um conjunto de dados atualizado a partir de técnicas estatísticas de detecção de deterioração. Normalmente, usa-se uma abordagem de janela deslizante, por exemplo removendo a informação mais antiga e adicionando informação atualizada. Contudo, para algumas aplicações, o processo de construção, treino, atualização de dados, implantação e avaliação é artesanal. Apesar de haver uma quantidade relativamente mínima em trabalhos propondo técnicas de detecção de deterioração de modelos de aprendizado de máquina [Baena-Garcıa et al. 2006, Sobhani e Beigy 2011, Maciel, Santos e Barros 2015], a etapa de parametrização ou reparametrização no caso do retreino requer um conjunto de habilidades e percepções sensíveis que dependem muito da experiência do engenheiro de aprendizado de máquina a frente do processo. Contudo, recentemente houveram avanços relevantes nos estudos de auto-ML [Feurer et al. 2015, Kotthoff et al. 2017, Feurer et al. 2019, Madrid et al. 2019].

Nesse contexto, este trabalho apresenta a pesquisa e o desenvolvimento de modelos de *Auto-ML* com dados em *stream* que sejam capazes de suprir possíveis desvios de conceito de soluções de aprendizado de máquina de forma automatizada, sem a necessidade de retreino periódico. De forma a contribuir para o incremento de esforços investigativos de autocorreção de desvios de conceito em implementações de soluções já em ambiente de produção. Dito isto, o trabalho apresenta a discussão sob a ótica de duas hipóteses, a saber:

H1) Modelos de *AutoML* com dados em *stream* são capazes de se adaptar aos desvios de conceito que eventualmente surgem nas distribuições dos dados, enquanto o modelo está em produção;

H2) Modelos de *AutoML* com dados em *stream* possuem performance equivalente aos modelos que servem como metodologia de referência para este trabalho, ou seja, modelos que necessitam de retreino total de forma periódica.

1.1 Organização do documento

Este trabalho está estruturado em quatro blocos. No primeiro bloco é apresentada a Fundamentação Teórica que servirá de base para todo o estudo, abrangendo as etapas de projetos envolvendo *Machine Learning* (ML); trazendo as definições de conceito e

desvio de conceito, além de fundamentação para aplicação de *Auto-Machine Learning*. No segundo bloco, o experimento proposto é apresentado e comentado, com a motivação para o estudo, contexto de aplicação e detalhamento do experimento realizado. O terceiro bloco apresenta os resultados obtidos com a implementação de uma solução utilizando *Auto-ML* com dados em *stream* e sua comparação com os resultados obtidos da metodologia de referência. A conclusão do trabalho será apresentada no quinto bloco. E por fim, as referências bibliográficas utilizadas como base para todo o trabalho.

CAPÍTULO 2

Fundamentação Teórica

2.1 Etapas de Modelagem em Machine Learning

As etapas de modelagem em aplicações inteligentes de ML podem ser ilustradas utilizando uma técnica de mineração de dados conhecida como CRISP-DM (*Cross Industry Standard Process for Data Mining*). [Clark 2018] faz uso do modelo CRISP-DM para a auditoria de aplicações de ML, o adaptando de acordo com as necessidades do processo de aprendizado de máquina.

Para [Wirth e Hipp 2000] o CRISP-DM aborda partes de um problema definindo um modelo de processo que fornece uma estrutura para a execução de projetos de mineração de dados independentemente do setor industrial ou da tecnologia utilizada. O CRISP-DM visa tornar grandes projetos de mineração de dados menos dispendiosos, mais confiáveis, mais replicáveis, mais gerenciáveis e mais rápidos.

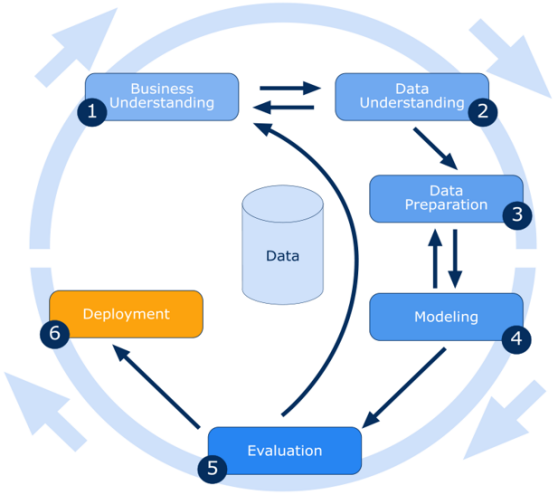

Figura 2.1: Fases do processo CRISP-DM para mineração de dados [Wirth e Hipp 2000].

.

[Wirth e Hipp 2000] detalha, ainda, as fases do CRISP-DM, conforme abaixo:

1. *Business Understanding* (Entendimento do negócio): Fase inicial que foca no entendimento dos objetivos e requisitos do projeto, traçando um plano preliminar para que tais objetivos sejam alcançados;
2. *Data Understanding* (Entendimento dos dados): A coleta inicial de dados seguida de atividades de exploração e conhecimento deles, com a finalidade de identificar problemas na qualidade dos dados, descobrir *insights* ou reconhecer padrões refletidos no conjunto;
3. *Data Preparation* (Preparação dos dados): A fase de preparação de dados inclui seleção de variáveis, registro de atributos, limpeza de dados, construção de novos atributos e transformação de dados para ferramentas de modelagem;
4. *Modeling* (Modelagem): Nesta fase, técnicas de modelagem são selecionadas e aplicadas, e seus parâmetros são calibrados para valores ideais. Frequentemente, percebe-se problemas de dados durante a modelagem ou obtém-se ideias para a construção de novos dados, sendo necessário revisitar a fase de preparação;
5. *Evaluation* (Avaliação): Avaliar o modelo de forma detalhada e revisar as etapas executadas durante sua construção é de vital importância para garantir que seus resultados sejam adequados aos objetivos de negócios;
6. *Deployment* (Implementação): O novo conhecimento adquirido precisará ser organizado e apresentado de forma que seja possível aplicá-lo e usá-lo em problemas reais, fornecendo uma resposta ao objetivo traçado durante o entendimento do negócio.

[Polyzotis et al. 2018] traz uma visão geral de um pipeline de aprendizado de máquina de ponta a ponta, considerando um ponto de vista de uso de dados. Pode-se notar fases do CRISP-DM na visão apresentada por [Polyzotis et al. 2018], como partes fundamentais do processo cíclico de *machine learning*.

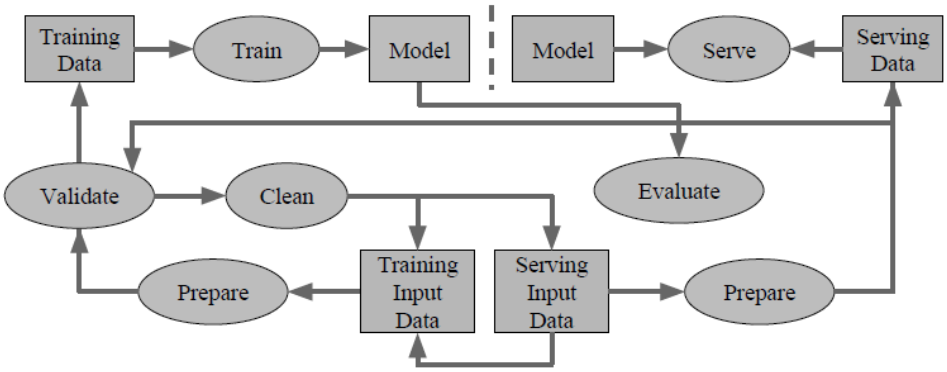

Figura 2.2: Ciclo de vida de projeto de ML [Polyzotis et al. 2018].

O ciclo de vida de um projeto de *machine learning* começa com a geração de dados de treinamento, de forma que os dados podem ser estruturados, semiestruturados ou não estruturados. [Polyzotis et al. 2018] enumera as etapas seguintes como:

- Preparação;
- Treino e Avaliação;
- Validação;
- Limpeza; e
- Entrega.

2.1.1 Preparação e modelagem de dados

A etapa inicial de qualquer projeto envolvendo *machine learning* é a compreensão dos dados disponíveis para o trabalho. Para que os modelos de ML sejam confiáveis é necessário que o conjunto de treinamento seja capaz de descrever o problema considerado. Espera-se que esse conjunto de dados contenha indícios de causas ou padrões de comportamento do problema a ser modelado, sendo necessário que os dados utilizados em produção sejam semelhantes aos dados do conjunto de treino. Dados reais nunca são perfeitos, visto que erros de dados são inevitáveis e podem ocorrer de maneiras criativas e inesperadas [Chai 2020].

A preparação dos dados para projetos de *machine learning* geralmente consome a maior parte do esforço investido em todo o processo [Witten, Frank e Hall 2005], pois dados reais têm uma qualidade baixa, sendo necessário uma verificação cuidadosa. Ao iniciar um projeto de *machine learning*, é necessário reunir todos os dados relacionados ao problema a ser solucionado em um conjunto de instâncias. A integração de dados de diferentes fontes geralmente apresenta muitos desafios, dentre eles o processo de limpeza de dados.

Para [Polyzotis et al. 2018], executar verificações de sanidade (*sanity checks*) é a necessidade de identificar se os dados têm a "forma" esperada antes do primeiro modelo ser treinado. Ainda para [Polyzotis et al. 2018], ferramentas de visualização ajudam a entender a forma dos dados, descobrindo suas propriedades e assim, desenvolvendo melhores verificações de seu estado.

A maioria dos *datasets* disponíveis para trabalhos de aprendizado de máquina apresentam valores ausentes, sendo necessários tratá-los. [Witten, Frank e Hall 2005] diz que valores ausentes são frequentemente indicados por entradas *out-of-range*, ou seja, fora dos padrões atribuídos à determinado atributo. Existem vários tipos de valores ausentes, que devem receber tratamentos diferentes. [Makaba e Dogo 2019] distingue pelo menos três tipos diferentes de valores de atributos desconhecidos: Considerando que A seja um atributo e X um exemplo para o qual o atributo valor está ausente:

- Valor ausente: X deve ter um valor para A, mas não está disponível;
- Valor não aplicável: o valor de A não pode ser medido para X;
- Valor sem importância: o atributo A pode assumir qualquer valor para X sem alterar sua classificação.

[Makaba e Dogo 2019] apresentam três grupos de estratégias para tratar valores ausentes, sendo elas descritas abaixo.

1. Ignorar exemplos com valores ausentes:
    - **Estratégia de exclusão:** a estratégia mais simples trata-se de ignorar os exemplos com valores ausentes, não sendo necessárias alterações no algoritmo de aprendizado.
2. Tratar valores ausentes uniformemente para todos os casos:
    - **Estratégia de valor ignorado:** ignorar os atributos que possuem valores ausentes, onde todo recurso que testar o valor desconhecido receberá um *truth value false*;
    - **Estratégia de qualquer valor:** corresponde essencialmente à interpretação de valores ausentes como valores sem importância;
    - **Estratégia de valor especial:** corresponde a inclusão de um novo atributo indicando se o valor do é conhecido ou não;
    - **Estratégia de valor comum:** se considerarmos que existe um valor "verdade" para cada valor ausente, pode-se tentar estimar esse valor com base nas informações conhecidas. Podendo assim, substituir valores ausentes dos atributos discretos pelo valor mais comum e, dos atributos contínuos pelo valor médio.
3. Tratar de valores ausentes de acordo com o caso:
    - **Estratégia de valor pessimista:** a ideia principal é impedir a utilização de atributos com muitos valores ausentes, ou seja, exemplos positivos com um valor ausente não são cobertos por nenhum teste com base nesse atributo e, inversamente, exemplos negativos são cobertos por todos os testes baseados neste atributo;
    - **Estratégia de valor previsto:** a estratégia requer um classificador para prever o valor ausente para cada atributo, sendo comumente usado o KNN (*k-nearest neighbor*;
    - **Estratégia de valor distribuído:** prever uma distribuição de probabilidade sobre todos os valores possíveis, em cada atributo.

Machine Learning depende da qualidade dos dados usados durante o treinamento para que se obtenha um bom desempenho. Para demonstrar um erro comum em *datasets*,

[Polyzotis et al. 2018] usa como exemplo um atributo 'País' onde seu valor é "US" (EUA), no entanto se em uma próxima carga de dados vier como "us" (nós), sem validação ou pré-processamento, o modelo simplesmente pensará que há um novo país. Para lidar com esse tipo de problema, [Polyzotis et al. 2018] sugere que se um valor de um atributo não for consistente, correções automáticas poderão ser adotadas; para que sejam o mais assertivas possíveis, os usuários devem ser notificados por meio de alertas, mesmo que muitas técnicas de limpeza de dados sejam relevantes.

Quando coletados originalmente, muitos dos campos ou atributos utilizados para o treinamento de modelos de *machine learning* provavelmente não tinham um propósito ao serem mapeados ou eram pouco importantes e foram deixados em branco ou desmarcados.

[Witten, Frank e Hall 2005] afirma que erros tipográficos ou de medição em valores numéricos geralmente causam discrepâncias que podem ser detectadas através de uma análise gráfica de cada um dos atributos que compõe o modelo. Valores errados, geralmente se desviam significativamente do padrão que é aparente nos valores restantes. Às vezes, porém, é difícil encontrar valores imprecisos, principalmente sem o conhecimento especializado do domínio.

Outro erro comum em *datasets* ocorre quando há dados duplicados, e a maioria das ferramentas de aprendizado de máquina produzirá resultados diferentes se algumas das instâncias nos *datasets* de treino forem duplicadas, pois a repetição lhes dará mais influência no resultado [Witten, Frank e Hall 2005]. Portanto identificar registros duplicados e/ou incorretos ajuda na limpeza dos dados, por isso é importante remover registros indesejados no início do pipeline [Chai 2020].

Os dados podem se tornar obsoletos, visto que muitos itens mudam conforme as circunstâncias mudam, portanto, é necessário levar em consideração se os dados utilizados em qualquer modelo de *machine learning* ainda estão atualizados e tem importância para o objetivo traçado para o modelo preditivo.

Como relatado, a limpeza de dados é de vital importância para preparação de dados para análise e é importante lidar com possíveis erros de dados antes de apresentar os resultados. Segundo [Chai 2020] a comparação dos resultados do modelo com e sem os erros de dados permite a apresentação de evidências gráficas para mostrar que a limpeza dos dados vale o tempo gasto. Embora a correção de erros nem sempre seja perfeita, o processo ainda melhora a qualidade dos dados utilizados para o treinamento.

2.1.2 Construção do modelo

Para [Polyzotis et al. 2018] um dos aspectos mais artísticos de *machine learning* é a preparação de dados. Durante o desenvolvimento inicial de um modelo de *machine*

learning, a preparação de dados se resume na projeção de um conjunto de atributos que são mais preditivos para o *target* de saída. Quando os modelos estiverem mais maduros, o foco poderá mudar para otimização de atributos e redução de latência, selecionando um subconjunto dentre todos os atributos disponíveis, mantendo a mesma precisão. Tal mudança de foco é comumente chamada de seleção de atributos.

Cada instância que fornece a entrada para o aprendizado de máquina é caracterizada por seus valores em um conjunto fixo de atributos predefinidos [Witten, Frank e Hall 2005], onde as linhas são chamadas de instâncias e as colunas recebem o nome de atributos.

[Witten, Frank e Hall 2005] afirma que o valor de um atributo para uma instância específica é uma medida da quantidade à qual o atributo se refere, podendo ser um atributo numérico ou nominal. Atributos numéricos, às vezes chamados de atributos contínuos, medem números - com valor real ou inteiro; já atributos nominais assumem valores em um conjunto finito pré-especificado de possibilidades e às vezes são chamados categóricos.

A engenharia de dados é o processo de construção de atributos, que inclui tarefas como a extração de atributos a partir de dados brutos, por meio da seleção de atributos [Kumar e Minz 2014]. [Kumar e Minz 2014] afirma ainda que enquanto a comunidade de *machine learning* estudou a seleção algorítmica de recursos, os esforços auxiliares na engenharia de dados foram amplamente ignorados. Já para [Polyzotis et al. 2018] a qualidade de um atributo normalmente está ligada ao poder preditivo deste, embora seja difícil estimar o poder preditivo, um bom caminho é entender a correlação do atributo com o *target* de saída.

Após a estruturação dos dados em instâncias e atributos, um conjunto de transformações é aplicado antes de serem inseridos no pipeline de um modelo preditivo, tal transformação depende do algoritmo de *machine learning* utilizado para a predição. No entanto, aprender as representações e o objetivo pode exigir recursos e dados significativos, portanto, a engenharia manual de recursos ainda é usada na maioria dos casos [Polyzotis et al. 2018].

À medida que os modelos se tornam mais maduros, os desenvolvedores geralmente experimentam a adição e remoção de novos atributos. As técnicas de seleção de atributos eliminam dados não úteis para reduzir a complexidade do modelo resultante. Onde o objetivo final é um modelo parcimonioso, mais rápido de calcular e com pouca ou nenhuma degradação na precisão preditiva. A seleção de atributos não se trata de reduzir o tempo de treinamento de fato, algumas técnicas aumentam o tempo geral de treino, mas de reduzir o tempo de pontuação do modelo [Zheng e Casari 2018].

[Zheng e Casari 2018] agrupa as técnicas de seleção de atributos em 3 grupos:

- **Técnicas de Filtragem:** As técnicas de filtragem pré-processam os recursos para remover aqueles que dificilmente serão úteis para o modelo. São muito mais baratas

em relação aos outros 2 grupos, mas elas não levam em consideração o modelo que está sendo empregado. Portanto, eles podem não conseguir selecionar os recursos certos para o modelo. É necessário cuidado para não eliminar inadvertidamente atributos que sejam úteis antes que eles cheguem à etapa de treinamento do modelo;

- **Método *Wrapper:*** Essas técnicas permitem que experimentos com subconjuntos de atributos sejam criados, o que reduz o risco de exclusão de um atributo com pouca contribuição para o modelo, mas com relevância quando utilizada em conjunto à outras. O método *wrapper* trata o modelo como uma "caixa preta"que fornece um índice de qualidade de um subconjunto proposto para os atributos do *dataset*.
- **Método *Embedded:*** Esses métodos executam a seleção de atributos como parte do processo de treinamento do modelo. Eles não são tão poderosos quanto os métodos *wrapper*, mas não são nem de longe tão caros. Comparados à filtragem, os métodos *embedded* selecionam atributos específicos do modelo, encontrando um equilíbrio entre despesa computacional e qualidade dos resultados.

A tendência contínua de crescimento exponencial dos dados disponíveis torna o processo de descoberta de conhecimento em dados ainda mais importante, sendo cada vez mais possível se obter resultados relevantes por meio de aplicações de *machine learning*, [Deshpande, Kamath e Joglekar 2019]. Assim, os problemas mais desafiadores estão no campo da classificação. Problemas de classificação no mundo real resultaram no grande número de casos em que o aprendizado da classificação é ainda mais difícil devido a conjuntos de dados desequilibrados.

Em aplicações de *machine learning* onde o problema de previsão pode ser solucionado utilizando um classificador, caso uma determinada classe possua mais exemplos sendo mais prevalente frente às outras, uma excelente precisão é obtida independentemente dos valores dos atributos. De fato, pode ser muito difícil apresentar uma previsão numericamente mais precisa [Witten, Frank e Hall 2005].

A questão fundamental do problema de aprendizado por meio de classes desequilibradas é a capacidade de dados desequilibrados comprometerem significativamente o desempenho dos algoritmos de aprendizado mais avançados. Tais algoritmos falham em representar adequadamente as características distributivas dos dados e, consequentemente, fornecem precisões desfavoráveis nas classes de dados [Deshpande, Kamath e Joglekar 2019].

Uma prática comum para lidar com conjuntos de dados desequilibrados é reequilibrá-los artificialmente, utilizando técnicas de *sampling*, onde o processo chamado de *upsampling* visa replicar casos da minoria, e processo chamado *downsampling* ignorar casos da maioria. No entanto, [Provost 2000] afirma que ainda está em aberto a questão de saber se simplesmente alterar a inclinação da distribuição (sem realmente examinar dados diferentes) pode melhorar sistematicamente o desempenho do modelo preditivo.

[Chawla, Japkowicz e Kotcz 2004] trazem o questionamento de "Qual é a distribuição correta para um algoritmo de aprendizado?", onde salienta que a distribuição natural, ou real, nem sempre é a distribuição ideal. Além disso, o desequilíbrio nos dados pode ser mais característico da “escassez” no espaço de atributos do que o desequilíbrio de classe.

Assim, [Chawla, Japkowicz e Kotcz 2004] afirma que o processo de *undersampling* aleatório pode potencialmente remover alguns exemplos importantes, e o *oversampling* aleatório pode levar ao ajuste excessivo. Além disso, *oversampling* pode introduzir uma tarefa computacional adicional se o conjunto de dados já for bastante grande, mas desequilibrado.

Ao tentar entender como os algoritmos de aprendizado de máquina se comportam com conjuntos de dados desequilibrados, podemos encontrar uma maneira de operar em determinados pontos desse espectro, [Provost 2000].

O aprendizado de máquina é aplicado com sucesso a muitas tarefas de análise de dados, desde o reconhecimento de imagens até a previsão de compras no varejo. Inúmeras bibliotecas de *machine learning* e serviços *on-line* estão disponíveis e novas aparecem a cada ano [Song, Ristenpart e Shmatikov 2017]. A seleção de modelos preditivos é um processo abrangente que visa obter modelos de *machine learning* satisfatórios, incluindo a seleção de algoritmos e ajustes de hiperparâmetros [Kumar e Minz 2014].

[Song, Ristenpart e Shmatikov 2017] define que um *pipeline* de aprendizado de máquina consiste em várias etapas, mostradas na figura 2.3. Tal *pipeline* será utilizado para futuras explicações e exemplificações.

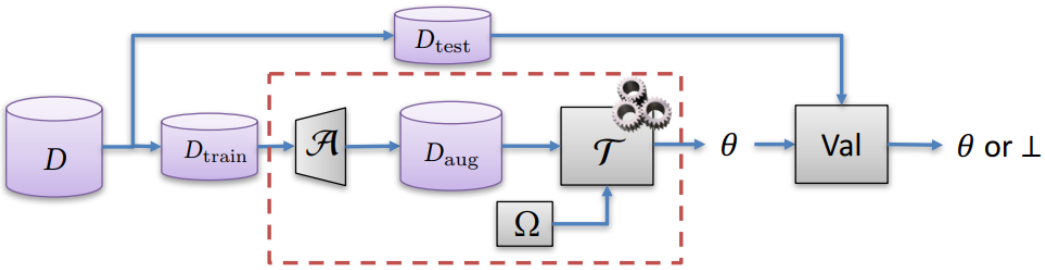

Figura 2.3: Pipeline de treinamento de aprendizado de máquina [Song, Ristenpart e Shmatikov 2017].

Na figura 2.3, explicada por [Song, Ristenpart e Shmatikov 2017], os dados D são divididos no conjunto de treinamento D_{train} e no conjunto de testes D_{test}. Os dados de treinamento podem ser aumentados usando um algoritmo A e, em seguida, os parâmetros

são calculados usando um algoritmo de treinamento T que usa um regularizador Ω. Os parâmetros resultantes são validados usando o conjunto de testes e aceitos ou rejeitados (um erro $\perp$ é gerado). Se os parâmetros θ forem aceitos, eles podem ser publicados (modelo de caixa branca) ou implantados em um serviço de previsão ao qual o usuário tem acesso de entrada / saída (modelo de caixa preta). A caixa tracejada indica as partes do pipeline que podem ser controladas pelo usuário.

Uma estratégia comum que visa melhorar a capacidade geral de modelos de *machine learning*, ou seja, aumentar seu poder preditivo em entradas fora de seus conjuntos de dados de treinamento, é usar o aumento de dados como uma etapa opcional de pré-processamento antes de treinar o modelo. Os dados de treinamento D_{train} são expandidos com novos pontos de dados gerados usando transformações determinísticas ou aleatórias. Sendo demonstrado na figura 2.3, por [Song, Ristenpart e Shmatikov 2017], o conjunto de dados expandido resultante D_{aug} é então usado para treinamento.

O conjunto de dados D_{aug} é usado como *input* por um algoritmo de treinamento T (geralmente randomizado), que também recebe como entrada uma cadeia de configuração γ chamada hiperparâmetros. O algoritmo de treinamento T gera um conjunto de parâmetros θ, que define um modelo $f_\theta : X \to Y$.

O problema da otimização de hiperparâmetros aparece quando um modelo é governado por hiperparâmetros, ou seja, parâmetros que não são aprendidos pelo modelo, mas devem ser escolhidos pelo usuário [Bertrand 2019]. A otimização de hiperparâmetros é o processo de pesquisar e ajustar uma série de combinações possíveis para uma melhora incremental do modelo de *machine learning*.

[Zheng e Casari 2018] afirma que em um fluxo de trabalho de aprendizado de máquina, escolhemos não apenas os atributos, mas também o modelo, conforme figura 2.4. Esta é uma alavanca de dupla articulação e a escolha de uma afeta a outra. Bons atributos facilitam a etapa de modelagem subsequente e o modelo resultante é mais capaz de concluir a tarefa desejada. Atributos ruins podem exigir um modelo mais complexo para atingir o mesmo nível de desempenho.

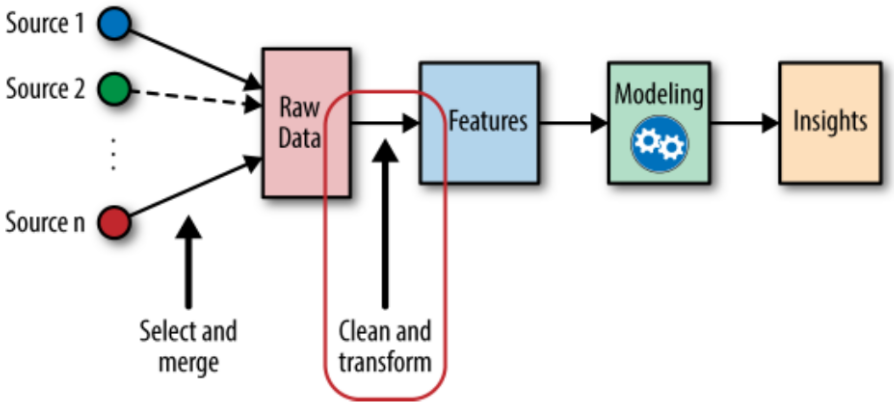

Figura 2.4: O lugar da engenharia de dados no fluxo de trabalho de aprendizado de máquina [Zheng e Casari 2018].

A popularidade do aprendizado de máquina, sendo possível graças a capacidade computacional, o volume de dados gerados associados a capacidade armazenamento, ocasionou a uma explosão no número de bibliotecas, estruturas e serviços de *machine learning* [Song, Ristenpart e Shmatikov 2017]. Esses serviços automatizam grande parte do *pipeline* de *machine learning* moderno, onde os usuários podem fazer *upload* de conjuntos de dados (*datasets*), realizar treinamento e disponibilizar os modelos resultantes para uso.

Os algoritmos de aprendizado de máquina visam otimizar o desempenho de uma determinada tarefa usando exemplos e/ou experiências anteriores. De acordo com [Schmidt et al. 2019], quando o processo de modelagem é concluído, tendo o sido modelo treinado, otimizando seu desempenho, geralmente medido através de algum tipo de função de custo, é necessário que o modelo seja avaliado em dados não vistos anteriormente, indicados como conjunto de teste, para estimar sua capacidade de generalização e extrapolação.

2.1.3 Avaliação

A avaliação é a chave para fazer progressos reais em projetos de *machine learning*. [Song, Ristenpart e Shmatikov 2017] demonstra na figura 2.3 que um modelo treinado é validado medindo sua precisão de teste (θ, D_{test}). Se a precisão do teste for muito baixa, a validação poderá rejeitar o modelo, gerando algum erro que representamos com um símbolo distinto $\perp$. Uma métrica relacionada é a diferença entre a métrica de validação do teste versus a métrica de validação do treino, onde essa lacuna mede o quão adaptado o modelo está em seu conjunto de dados de treinamento.

Para problemas de classificação, é natural avaliar o desempenho de um classificador em termos da taxa de erro. O classificador prevê a classe de cada instância: se estiver correta, isso é contado como um sucesso; caso contrário, é um erro. A taxa de erro é apenas a proporção de erros cometidos em todo um conjunto de instâncias e mede o desempenho geral do classificador [Witten, Frank e Hall 2005].

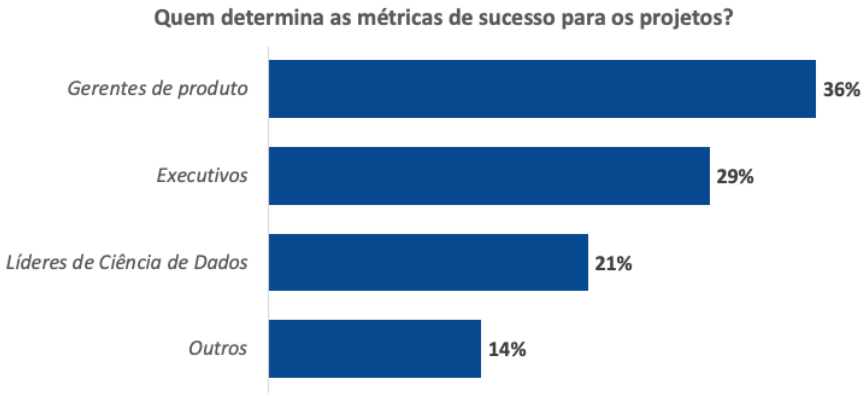

Figura 2.5: Quem determina as métricas de sucesso para os projetos? [Lorica e Paco 2018].

Sessenta e cinco por cento das organizações entrevistadas por [Lorica e Paco 2018] colocam a responsabilidade de definição de métricas de avaliação de seus projetos de *machine learning* em gerentes ou executivos de produtos, enquanto 21% deixam essa tarefa para os líderes de ciência de dados, podendo ser visto na figura 2.5.

Em sua pesquisa [Lorica e Paco 2018] ainda constatam que 73% das organizações utilizam métricas de negócios (*business*) e que 48% adotam métricas de avaliação voltadas para *machine learning*, podendo ser visto na figura 2.6.

Figura 2.6: Quais as métricas usadas para a avaliação dos projetos? [Lorica e Paco 2018].

2.1.4 Implantação, monitoramento e reavaliação

O uso produtivo do aprendizado de máquina não é apenas uma questão de encontrar alguns dados e aplicar cegamente os algoritmos de aprendizado [Witten, Frank e Hall 2005]. Para o autor, há evidências que autores executam uma infinidade de algoritmos de aprendizado em um determinado conjunto de dados e, em seguida, escrevem um artigo alegando que esse método de aprendizado de máquina é o melhor para esse problema - com pouco entendimento aparente do que esses algoritmos fazem, da natureza dos dados ou da consideração da significância estatística.

[Witten, Frank e Hall 2005] ainda salienta que a escolha entre o uso de modelos de *machine learning* simples ou modelos mais sofisticados deve ser feita com cuidado, pois um modelo extremamente simples - que na maioria dos casos faz escolha da classe majoritária durante a classificação - define uma linha de base sobre a qual qualquer método de aprendizado deve ser capaz de melhorar. Dito isto, o autor considera a melhoria da linha de base alcançada por um método simples como uma proporção da melhoria da linha de base alcançada por um método sofisticado.

Para que um modelo de aprendizado de máquina tenha o desempenho esperado, os conjuntos de dados de entrada precisam atender às especificações de qualidade pré-definidas, onde o processo de avaliação verifica se os dados necessários existem e estão atualizados. Tais verificações evidenciam a intenção de correlacionar as alterações nos conjuntos de dados de entrada com as variações dos resultados do aprendizado de máquina, para que mais estudos possam ser feitos para melhorar a precisão do aprendizado de máquina, onde esse tipo de informação pode ser importante para derivação e seleção de novos atributos, [Wu et al. 2011].

A maioria dos modelos de classificação possui um ou mais parâmetros que tem por finalidade controlar a complexidade do modelo. Quanto maior a complexidade do

modelo, maior o poder discriminador que ele possui, embora o risco de *overfitting* também aumente. *Overfitting* é um fenômeno observado quando um modelo treinado apresenta um desempenho extremamente bom nas amostras usadas para treinamento, mas apresenta um desempenho ruim em novas amostras ainda não conhecidas pelo modelo; isto é, o modelo não generaliza bem [Xu e Goodacre 2018].

[Xu e Goodacre 2018] afirmam que para encontrar um conjunto ideal de parâmetros do modelo, que tenham um equilíbrio apropriado entre esses dois aspectos, é necessário dividir os dados em um conjunto de treinamento e validação, onde:

- O conjunto de treinamento é usado para construir o modelo com várias configurações de parâmetros do modelo e, em seguida, cada modelo treinado é desafiado pelo conjunto de validação;
- O conjunto de validação contém amostras com proveniência conhecida, no entanto não são utilizadas para modelar; portanto, as previsões no conjunto de validação permitem avaliar a precisão do modelo.

.

Com base nos erros no conjunto de validação, o(s) parâmetro(s) do modelo ideal é resultado do conjunto usando com o menor erro de validação. É importante ter uma boa estimativa do desempenho do modelo treinado e otimizado em amostras desconhecidas em geral, ou seja, para avaliar o desempenho da generalização.

Para [Xu e Goodacre 2018] uma única divisão do conjunto de dados, entre treinamento e teste, pode fornecer estimativas errôneas acerca do desempenho do modelo, e que o desempenho medido pela validação cruzada é super otimista. O autor ainda apresenta estudos que destacam a importância de se ter um segundo conjunto de testes adicional, não visto durante a modelagem ou a validação do modelo, com a finalidade de se obter uma melhor estimativa do desempenho da generalização do modelo.

O desempenho estimado do modelo pode ser afetado por muitos fatores, como o algoritmo de modelagem, a sobreposição entre os dados, o número de amostras disponíveis para treinamento e, talvez o mais importante, o método usado para dividir os dados. [Xu e Goodacre 2018] apresenta os métodos de separação de dados relatados e utilizados na literatura, categorizados em três tipos diferentes:

1. Validação cruzada, técnica que divide os dados em k partes diferentes, onde uma parte é mantida como o conjunto de validação. O modelo é treinado nas partes restantes do $k-1$ e depois aplicado ao conjunto de validação, registrando seu desempenho preditivo. Esse processo se repete k vezes para que cada parte tenha sido usada como um conjunto de validação uma vez. As performances preditivas registradas são então calculadas como média, o parâmetro do modelo ideal é determinado como aquele que teve o melhor desempenho preditivo médio;

2. Selecionar aleatoriamente uma proporção de amostras, separando-as para a validação e utilizando as amostras restantes para treinamento;
3. Com base na distribuição dos dados, selecionar sistematicamente um determinado número das amostras mais representativas do conjunto de dados e usar as amostras restantes para validação.

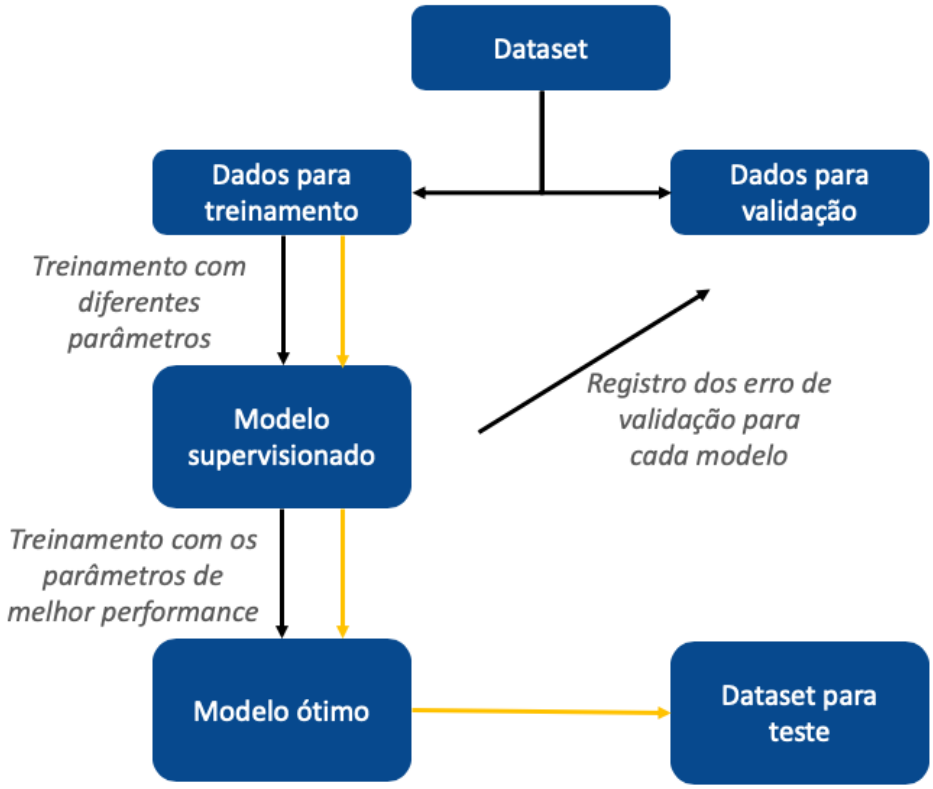

Figura 2.7: Fluxo geral usado para seleção de modelo. As setas pretas indicam o processo de validação, enquanto as setas amarelas indicam o treinamento final e o teste no processo de teste cego, [Xu e Goodacre 2018].

Um fluxo geral de um processo típico de validação de modelo é apresentado na figura 2.7. [Xu e Goodacre 2018] salienta que mesmo seguindo o procedimento por ele proposto, ainda é impossível dizer até que ponto o desempenho preditivo estimado do modelo do conjunto de testes corresponde à verdadeiro distribuição subjacente dos dados. Isso ocorre porque, em aplicações do mundo real, o último conjunto de testes, geralmente, é desconhecido, e é preciso presumir que o desempenho medido usando o conjunto é um estimador preciso e não-aproximado para o desempenho do modelo em todas as amostras

desconhecidas provenientes da mesma distribuição do conjunto de dados de treinamento e teste.

2.2 Desvio de Conceito (*Concept Drift*)

Como apresentado seção 2.1, a modelagem de problemas com soluções de aprendizado de máquinas usa atributos ou características de entrada e um *target* de saída, sendo mapeados utilizando uma função de custo para minimizar o erro. Tais modelos são construídos visando fazer previsões de dados futuros, que ainda não foram vistos pelo modelo preditivo, e espera-se que o modelo seja capaz de prever tais dados com a mesma precisão obtida no treinamento.

Dados são o principal componente utilizado para treinamento de um modelo preditivo, tais dados devem trazer indícios sobre a condição prevista pelo modelo. Dito isto, entende-se que os novos dados do modelo em produção serão sempre semelhantes aos dados utilizados no treinamento, assumindo que os atributos de entrada e a variável de saída serão sempre constantes. O que, na realidade, não acontece.

Independentemente do tipo de aprendizado envolvido, chama-se o que deve ser aprendido de "conceito" e a saída produzida pelo modelo de "descrição do conceito" [Witten, Frank e Hall 2005]. Já a mudança na distribuição dos dados ao longo do tempo, produz o fenômeno chamado de "desvio de conceito" [Gama et al. 2014].

Os dados podem mudar com o tempo. Isso pode resultar em desempenho preditivo ruim e degradante em modelos preditivos que assumem um relacionamento estático entre variáveis de entrada e saída. Esse problema das mudanças nos relacionamentos subjacentes nos dados é chamado de desvio de conceito (*concept drift*) no campo do aprendizado de máquina.

O real desvio de conceito refere-se a mudanças na distribuição condicional da saída (variável de destino), dada uma entrada (atributos), enquanto a distribuição da entrada pode permanecer inalterada [Gama et al. 2014].

Um modelo preditivo consegue identificar padrões a partir dos dados nos quais ele entrou em contato, por meio do treinamento. Focando sua predição com base em dados essenciais e desprezando outros que não tem certa expressividade, sendo capaz de generalizar a predição para dados nos quais ainda não entrou em contato. Sendo assim, é comum que modelos em produção comecem a perder desempenho, visto que os dados são parte fundamental de qualquer modelagem preditiva e que esses dados sofram alterações durante o passar do tempo.

Modelos de *machine learning* analisam um conjunto de dados históricos e, em seguida, desenvolvem um modelo que reflete o mundo como era quando foi formado. Mas o mundo é dinâmico, e as distribuições complexas que os modelos de um modelo

provavelmente não são estacionárias e, portanto, mudam com o tempo, levando à deterioração do desempenho do modelo. Para resolver esse problema, é necessário desenvolver mecanismos para detectar e lidar com desvios de conceitos [Webb et al. 2015].

O mundo é dinâmico, em constante fluxo. Mas o aprendizado de máquina geralmente cria modelos estáticos a partir de dados históricos. À medida que o mundo muda, esses modelos podem se tornar cada vez mais confiáveis. Para falar sobre desvio de conceito, [Webb et al. 2015] traz o conceito de fluxo de dados (*data stream*) como um *dataset* onde os elementos possuem um registro de data e hora (*time stamps*). [Webb et al. 2015] ainda salienta que o treinamento de um modelo de aprendizado de máquina só possui acesso aos dados com registro de data e hora antes de um ponto específico, no entanto quando colocados em produção, tais modelos devem ser aplicados à elementos de dados com registro de data e hora subsequentes. Em resumo, um algoritmo de aprendizado analisa os dados de treinamento para criar um modelo para dados de testes futuros.

Um problema difícil com o aprendizado de máquina em muitos domínios do mundo real é que o conceito de interesse pode depender de algum contexto oculto [Tsymbal 2004], não fornecido explicitamente na forma de atributos preditivos. Na maioria dos casos a causa ou razão da mudança de conceito é desconhecida a priore, o que dificulta a tarefa de aprendizagem pelos modelos preditivos. Mudanças no contexto podem induzir mudanças mais ou menos radicais no conceito alvo, sendo esse a definição de desvio de conceito apresentada por [Tsymbal 2004]. Um modelo preditivo eficaz deve ser capaz de rastrear essas mudanças e se adaptar rapidamente a elas.

Um problema difícil ao lidar com o desvio de conceito é distinguir entre desvio de conceito verdadeiro e um ruído. Alguns algoritmos podem reagir exageradamente ao ruído, interpretando-o erroneamente como desvio de conceito, enquanto outros podem ser altamente robustos ao ruído, ajustando-se às mudanças muito lentamente. Um modelo ideal deve combinar robustez ao ruído e sensibilidade à deriva do conceito [Kubat e Widmer 1994].

Na visão de [Žliobaitė, Pechenizkiy e Gama 2016] o aprendizado supervisionado tradicional pressupõe que os dados de treinamento e aplicação provêm da mesma distribuição, conforme ilustrado na Figura 2.8 (a). Na vida real, as previsões precisam ser feitas de forma instantânea, muitas vezes em tempo real, trazendo desafios adicionais. Nesse contexto, pode ser esperado que a distribuição de dados mude ao longo do tempo. Assim, a qualquer momento, os dados de teste podem ser provenientes de uma distribuição diferente da distribuição utilizada no treinamento do modelo preditivo, conforme ilustrado na Figura 2.8 (b).

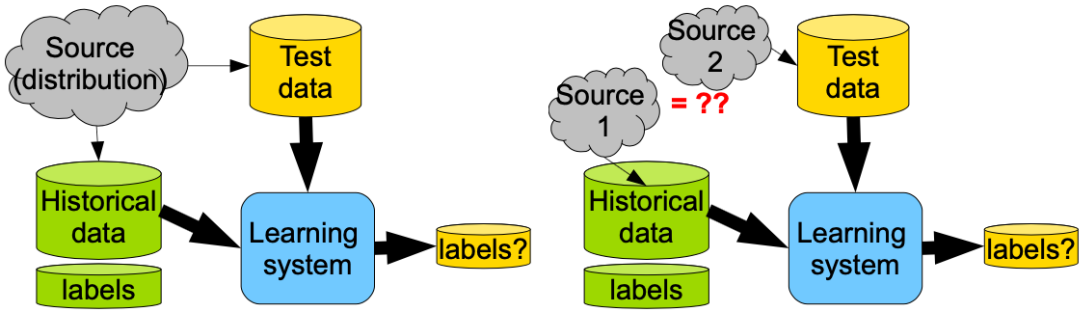

Figura 2.8: Aprendizado supervisionado estacionário (a) e aprendizado sob o desvio de conceito (b), [Žliobaitė, Pechenizkiy e Gama 2016].

O entendimento sobre desvio de conceito é de crescente importância, visto, à medida que mais e mais dados são organizados na forma de fluxos de dados (*data stream*), conceito apresentado acima por [Webb et al. 2015], em vez de bancos de dados estáticos, e não é realista esperar que as distribuições de dados permaneçam estáveis por um longo período de tempo.

[Gama et al. 2014] distingue dos tipos de desvios de conceito (figura 2.9) , sendo eles:

- Desvio de conceito real: ocorre quando há mudanças conceito alvo ou variável de saída, tendo ou não variações nas distribuições utilizadas para o treinamento;
- Desvio de conceito virtual: acontece se a distribuição dos dados recebidos e utilizados em treinamento mudar sem afetar o conceito alvo ou variável de saída.

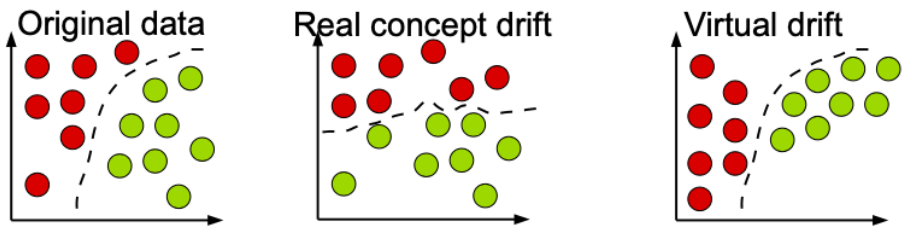

Figura 2.9: Tipos de desvios de conceito: círculos representam instâncias, cores diferentes representam classes diferentes. [Gama et al. 2014].

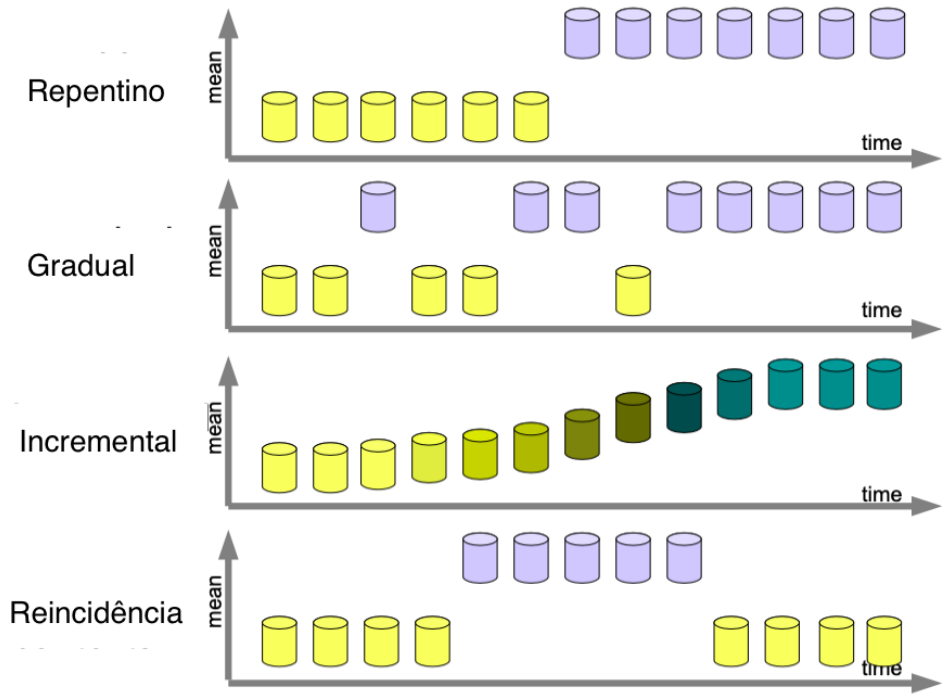

Figura 2.10: Ilustração dos quatro tipos estruturais de desvio de conceito, [Žliobaitė 2010].

Por meio da figura 2.10, [Žliobaitė 2010], apresenta 4 tipos estruturais de desvios que podem ocorrer nos conceitos. Pra explicá-los, o autor restringe o número de fontes possíveis ao longo do tempo a duas: S_I e S_{II}.

O padrão mais simples de um desvio de conceito é a que ocorre de forma repentina, quando no momento t_0 uma fonte S_I é subitamente substituída pela fonte S_{II}.

O desvio de conceito gradual pode ocorrer de duas maneiras, um onde as fontes S_I e S_{II} estão ativas por um determinado período. Com o passar do tempo, a probabilidade de amostragem da fonte S_I diminui, aumenta a probabilidade de amostragem da fonte S_{II}. Um ponto de atenção levantado pelo autor é que no início desse tipo de desvio gradual, antes que mais instâncias sejam vistas, uma instância da fonte S_{II} pode ser facilmente mal interpretada com ruído aleatório. Outro tipo de desvio também referido como gradual inclui mais de duas fontes, no entanto, a diferença entre as fontes é muito pequena, portanto, o desvio é percebido apenas quando se observa um período de tempo mais longo. Tal forma de desvio gradual recebe o nome de desvio incremental.

Por fim, há outro grande tipo de desvio de conceito conhecido como recorrente ou reincidente. Tal desvio ocorre quando o conceito anteriormente ativo reaparece após algum tempo. Difere da noção comum de sazonalidade de uma maneira que não é

certamente periódica, não está claro quando a fonte pode reaparecer.

[Gama et al. 2014] afirma que alterações na distribuição de dados ao longo do tempo podem se manifestar de diferentes formas, demonstradas na figura 2.11. De acordo com o autor, o desvio de conceito pode ocorrer de forma repentina, alternando de um conceito para outro (exemplo: substituição de um sensor por outro sensor que tenha uma calibração diferente em determinada aplicação); ou de forma incremental, sofrendo com uma suave alteração de conceito durante o tempo (o sensor vai perdendo acurácia, se tornando menos preciso). O desvio pode acontecer repentinamente (por exemplo, os tópicos de interesse pesquisados como analista de crédito podem mudar repentinamente, por exemplo, dos preços da carne para o transporte público) ou gradualmente (por exemplo, os tópicos de notícias relevantes mudam de habitações para casas de férias, enquanto o usuário alterne abruptamente, mas continue voltando ao interesse anterior por algum tempo).

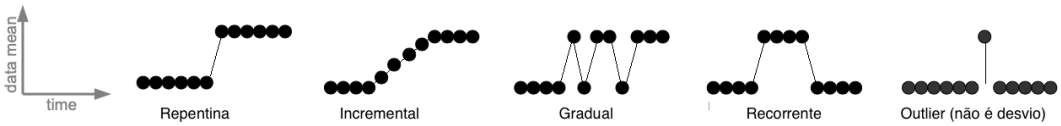

Figura 2.11: Padrões de mudanças ao longo do tempo [Gama et al. 2014].

O desvio pode acontecer repentinamente (por exemplo, ao analisar crédito o hábito de pesquisa de tópicos relacionados ao preço de alimentos pode mudar, abruptamente, para interesse em valores de transporte público) ou gradualmente (por exemplo, os tópicos de notícias relevantes mudam de habitações para casas de férias, enquanto o usuário alterne abruptamente, mas continue voltando ao interesse anterior por algum tempo).

Por fim, [Gama et al. 2014], afirma que desvios podem introduzir novos conceitos que não foram apresentados anteriormente, ou conceitos vistos anteriormente podem ocorrer depois de algum tempo (por exemplo, na moda). Tais mudanças podem ser caracterizadas por gravidade, previsibilidade e frequência.

Um dos desafios de modelos de aprendizado de máquina ao se tratar desvios de conceito levantados por [Gama et al. 2014], e anteriormente apresentado por [Kubat e Widmer 1994], está na capacidade de distinção de uma deriva de conceito real com um outlier ou ruído que se refere a um desvio ou anomalia aleatória única, figura 2.11.

Os modelos preditivos que necessitam de mecanismos capazes de detectar e se adaptar à evolução dos dados ao longo do tempo, caso contrário, sua precisão ficará comprometida tendendo a redução. Para isso, é necessário que o modelo seja atualizado

levando em consideração os novos dados ou ser completamente substituído para atender à ao novo conceito *target*, [Gama et al. 2014].

[Tsymbal 2004] afirma que um sistema ideal para se tratar de desvios de conceito deve ser capaz de:

1. adaptar-se rapidamente ao desvio de conceito;
2. ser robusto ao ruído e diferenciá-lo do desvio de conceito;
3. reconhecer e tratar contextos recorrentes.

Corroborando com a ideia de [Tsymbal 2004], [Gama et al. 2014] acrescenta que modelos preditivos são necessários para reconhecer tais mudanças a medida que o novos exemplos ocorram usar não mais do que uma quantidade fixa de memória para qualquer armazenamento.

2.2.1 Generalização

As técnicas de aprendizado de máquina giram em torno de maneiras de transformar, construir ou impor algum tipo de forma nos dados e usá-los para descobrir, decidir, classificar, ranquear, agrupar, recomendar, rotular ou prever o que está acontecendo ou o que acontecerá. Dentre as técnicas de aprendizado de máquina o principal ponto de convergência é que todos eles podem ser usados para classificar as coisas. Sua capacidade de classificar depende de aprender a reconhecer as diferenças entre categorias que permanecem fixas. Essas categorias podem ser numerosas, como na mineração de dados para reconhecimento facial, onde há muitos exemplos a serem classificados, ou podem ser poucas, como na classificação de e-mail que requer a classificação positiva ou negativa para spam. A previsão usando o aprendizado de máquina pressupõe a existência de classificações relativamente estáveis, podendo ser arbitrárias ou altamente artificiais. A combinação de indiferença às diferenças reais e presunção de classificações estáveis é uma característica distintamente problemática do aprendizado de máquina [Mackenzie 2015]. Partindo desse conceito, um bom modelo preditivo deve ser capaz de se adaptar adequadamente a dados novos e inéditos, extraídos da mesma distribuição usada para criar o modelo.

A generalização é um problema onde os questionamentos giram em torno de saber se um determinado modelo é significativo ou válido, ou se tal modelo encontrou nos dados de treinamento indícios suficientes para ser aplicados a eventos subsequentes.

A capacidade de generalizar a partir de exemplos é amplamente reconhecida como uma capacidade essencial de qualquer modelo de aprendizagem. A generalização envolve a observação de um conjunto de exemplos utilizados no treinamento de algumas características gerais do conceito comuns a esses exemplos, e depois a formulação de uma definição de conceito com base nessas características comuns

[Mitchell, Keller e Kedar-Cabelli 1986]. [Mackenzie 2015] ainda salienta que o aprendizado de máquina busca explorar padrões de alta dimensão, onde os espaços vetoriais justapõem quase qualquer número de características. A generalização de um modelo depende dos *trade-offs* entre *overfitting* e *underfitting* e entre a modelagem de previsões feitas com muitos ou poucos indícios nos dados utilizados para o treinamento.

[Hastie, Tibshirani e Friedman 2009] afirma que com muito ajuste de parâmetros, o modelo se adapta bem aos dados de treinamento resultando em uma acurácia de teste alta, porém perde a capacidade de generalização. Por outro lado, se o modelo não for complexo o suficiente, possuindo os ajustes de parâmetros necessários, ficará desajustado e poderá apresentar um grande viés, resultando novamente em uma generalização deficiente.

[Mitchell, Keller e Kedar-Cabelli 1986] afirma que, nos últimos anos, surgiram propostas de métodos de generalização que contrastam fortemente com os métodos baseados em similaridade e com uso intensivo de dados; tais métodos deixam de confiar na quantidade de dados utilizados para o treinamento da solução em *machine learning* e em um *bias* indutivo, métodos mais recentes restringem a busca, baseando-se no conhecimento do domínio da tarefa e do conceito do aprendizado de máquina.

Existem várias técnicas para melhorar a generalização dos modelos de aprendizado de máquina, técnicas que processam dados com mais cuidado e atenção, como validação cruzada, utilização de técnicas de re-amostragem, criação, comparação e utilização de modelos em conjuntos (as chamadas técnicas de *ensemble*), utilização de algoritmos *random forest*, técnicas de penalização ou técnicas de regularização. Em outros casos, o aumento do poder computacional a utilização de um conjunto de dados maior são meios usados para incrementar o poder de generalização do modelo. Em outros casos, muito esforço é dedicado para encontrar e refinar os atributos ou fontes de dados que parecem dar melhor suporte às previsões [Hastie, Tibshirani e Friedman 2009].

Após analisar um único exemplo de treinamento em termos de aprendizagem, esses métodos são capazes de produzir uma generalização válida do exemplo, juntamente com uma justificativa dedutiva da generalização em termos do conhecimento do conceito buscado. Mais precisamente, esses métodos baseados em explicações analisam o exemplo de treinamento construindo uma explicação de como o exemplo satisfaz a definição do conceito de aprendizagem. As características do exemplo identificado por esta explicação são então usadas como base para a formulação da definição do conceito geral. A justificativa para esta definição de conceito segue a explicação construída para o exemplo de treinamento [Mitchell, Keller e Kedar-Cabelli 1986].

2.2.2 Avaliação de Modelos

Métricas de avaliação de modelos de *machine learning* são ferramentas capazes de medir a performance dos modelos preditivos, de forma que métricas diferentes avaliam características diferentes do classificador induzido pelo algoritmo de classificação.

[Hossin e Sulaiman 2015] afirma que as métricas de avaliação pode ser categorizada em três tipos, que são *threshold* (métricas de separação limiar, em português), métricas probabilísticas e métricas de ranqueamento, onde cada um desses tipos de métricas avalia o classificador com objetivos diferentes. O autor ainda salienta que todos esses tipos de métricas são um método de grupo escalar, em que toda performance é medida e apresentada através de um único valor, facilitando a comparação e análise, embora possa mascarar detalhes sutis de seus comportamentos.

Uma das práticas mais comuns atualmente é focar em apenas um critério, ou seja, usar apenas uma métrica de avaliação ou projetar um esquema de avaliação personalizado com vários critérios para a aplicação específica. Além disso, ao aplicar um algoritmo de aprendizado para resolver um problema do mundo real, muitas vezes existem outros fatores importantes que precisam ser considerados [Lavesson e Davidsson 2008].

[Lavesson e Davidsson 2008] distinguem três tipos de candidatos para avaliação, ideia sustentada também por [Hossin e Sulaiman 2015]:

- As métricas de avaliação são utilizadas para avaliar a capacidade de generalização do classificador treinado; sendo usadas para medir e resumir a qualidade do modelo quando aplicado à dados não vistos durante o treinamento. Nesse caso, a métrica de avaliação é usada para medir e resumir a qualidade do classificador treinado quando testado com dados não vistos;
- As métricas de avaliação podem ser empregadas como avaliador na seleção de modelos, sendo usada para determinar o melhor classificador entre os diferentes tipos de classificadores treinados que se concentram no melhor desempenho futuro (modelo ideal) quando testados com dados invisíveis;
- As métricas de avaliação foram empregadas como discriminador para discriminar e selecionar a solução ótima (melhor solução) entre todas as soluções geradas durante o treinamento de classificação, onde somente a melhor solução que se acredita ser o modelo ideal será testada com os dados não vistos.

[Hossin e Sulaiman 2015] afirma que na primeira e na segunda aplicação de métricas de avaliação, quase todos os tipos de *threshold*, métricas probabilísticas e métricas de ranqueamento podem ser aplicados para avaliar o desempenho e a eficácia dos classificadores. Por outro lado, apenas alguns tipos de métricas poderiam ser empregados como discriminadores para discriminar e selecionar a solução ideal durante o treinamento de classificação.

Em aplicações de aprendizagem de máquina onde tem-se um problema classificatório, métricas de avaliação são utilizadas em dois estágios dentro do ciclo de *machine learning*: utilizada durante o treinamento (processo de aprendizado) para otimizar o algoritmo de classificação, empregada para discriminar e selecionar a solução capaz de produzir uma previsão mais precisa da avaliação futura do modelo; e, utilizada na fase de teste, medindo a eficácia do classificador quando aplicado à dados não conhecidos pelo modelo [Hossin e Sulaiman 2015].

O monitoramento de modelos preditivos em produção deve ser contínuo. Nessa abordagem, assume-se que a perda de desempenho do modelo preditivo acontece e deve-se quantificar tal decaimento. Avaliar o desempenho de modelos em produção é uma tarefa complicada, visto que é necessário possuir a predição do modelo e a variável *target* de saída real, comparando assim o previsto com o realizado.

As métricas de avaliação de desempenho podem ser selecionadas a partir das medidas tradicionais de precisão, como as citadas na tabela 3.5. No entanto, é importante considerar pontos de referências apropriados ou abordagens de *baseline* em configurações específicas [Gama et al. 2014]. Dito isso, o autor apresenta métricas para a medida de precisão dos modelos que utilizam *data streams*:

- Medida para o custo computacional do processo de mineração: RAM-Hours é uma medida unidimensional dos recursos computacionais usados pelos algoritmos de streaming, com base nas opções de custo de aluguel dos serviços de computação em nuvem. Cada GB de RAM implantado por 1 hora é igual a uma hora de RAM.
- Estatística para a classe levando em consideração o desbalanceamento de classes: A estatística é muito conveniente para calcular fluxo de *data stream*, como comparar, por exemplo, com uma alternativa do ROC.

[Gama et al. 2014] afirma que além de avaliar o desempenho da estratégia de aprendizado, deve-se avaliar a precisão da detecção de desvios de conceito separadamente para as estratégias que empregam detecção explícita de deriva como parte da estratégia de manipulação de conceito. Os critérios a seguir são relevantes para avaliar os métodos de detecção de alterações.

- Probabilidade de verdadeira detecção de desvios: capacidade do modelo de aprendizado de detectar desvios quando eles ocorrem, podendo ser computada em dados sintéticos, onde os desvios são conhecidos;
- Probabilidade de falsos alarmes: ao invés de relatar a taxa de falsos positivos comumente usada para detecções de desvios, nas configurações de streaming, é mais conveniente usar o inverso do tempo para detecção ou a duração média da execução, que é o tempo esperado entre as detecções de falsos positivos. Essa medida pode ser calculada tanto em dados sintéticos em que os desvios são

conhecidos ou em dados reais que não têm desvios; nesse caso, todas as detecções são contadas como alarmes falsos;

- Atraso na detecção: visa fornecer uma estimativa de quantas novas instâncias são necessárias para detectar um desvio de conceito após a ocorrência real de uma deriva (ou quanto tempo se passaria antes que a alteração fosse detectada). Normalmente, o tempo médio para detecção é usado. Os desvios precisam ser conhecidos, dados sintéticos são adequados para avaliar esse aspecto.

2.2.3 Detecção de desvio de conceito

A detecção de desvio refere-se às técnicas e mecanismos que caracterizam e quantificam o desvio do conceito através da identificação de pontos de mudança ou intervalos de tempo de mudança [Gu 2019]. O autor ainda apresenta uma estrutura geral para detecção de desvio contém quatro estágios, demonstrado na figura 2.12.

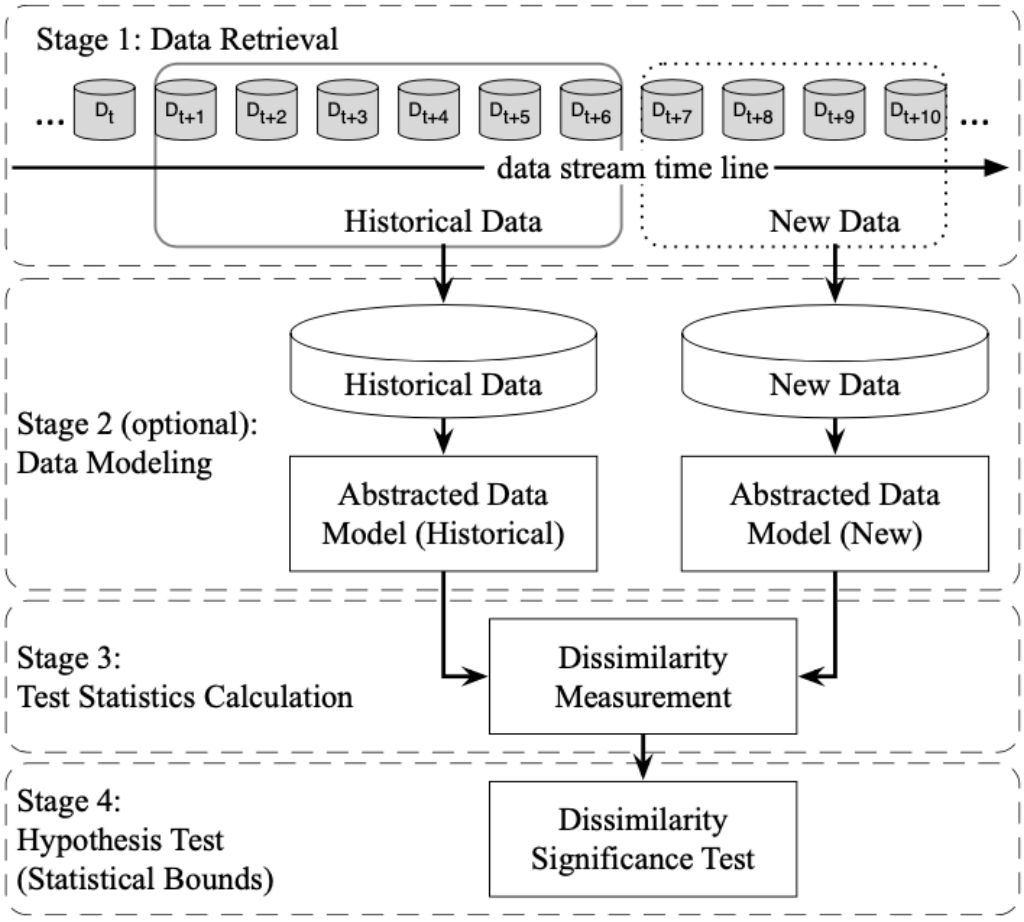

Figura 2.12: Uma estrutura geral para detecção de desvio de conceito [Gu 2019]

.

- Estágio 01 - Recuperação de dados: visa recuperar dados dentro dos fluxos de dados (*data streams*), como uma única instância não pode carregar informações suficientes para inferir a distribuição geral, ter uma boa organização dos dados para encontrar um padrão ou conhecimento significativo é importante nas tarefas de análise do fluxo de dados.
- Estágio 02 - Modelagem de dados: o objetivo é abstrair os dados recuperados e extrair os principais atributos que possuem informações sensíveis ao modelo, ou seja, os atributos com maior relevância durante o treino.
- Estágio 03 - Teste estatístico: quantifica a gravidade do desvio de conceito e forma estatísticas para o teste de hipótese. É considerado o aspecto mais desafiador da detecção de desvio de conceito, pois o problema de como definir uma métrica precisa e robusta da dissimilaridade ainda é uma questão em aberto.

- Estágio 04 - Teste de hipóteses: usa um teste de hipótese específico para avaliar a significância estatística da mudança observada no estágio 03. São usados com a finalidade de determinar a precisão da identificação do desvio de conceito no modelo, capaz de estimar a probabilidade de a alteração ser causada por um desvio de conceito e não por um ruído ou viés nas amostras usadas.

[Gu 2019] ainda aponta 3 alternativas de detecção de desvios de conceito por meio de algoritmos existentes, abaixo listadas:

- Detecção de desvio com base na taxa de erro: os algoritmos de detecção de desvio com base na taxa de erro do modelo treinado se concentram no rastreamento de alterações na taxa de erro on-line dos classificadores de base, caso seja identificado uma variação da taxa de erro (aumento ou redução) estatisticamente significantes, um processo de atualização será acionado;
- Detecção de desvio com base na distribuição de dados: os algoritmos de detecção de desvio baseada na distribuição de dados usam uma função ou métrica de distância para quantificar a diferença entre a distribuição dos dados usados no treinamento do modelo e os dados que o modelo ainda não conhece. Se a dissimilaridade for estatisticamente significativamente diferente, o sistema acionará um processo de atualização do modelo de aprendizagem;
- Detecção de desvio por teste de hipóteses múltiplos: algoritmos deste grupo aplicam técnicas semelhantes às mencionadas nas categorias anteriores, além de usar vários testes de hipóteses para detectar o desvio de conceito.

2.2.4 Adaptação ao Desvio de Conceito

Os algoritmos de aprendizado geralmente precisam operar em ambientes dinâmicos, que estão mudando inesperadamente; uma propriedade desejável desses algoritmos é a capacidade de incorporar novos dados [Gama et al. 2014]. A adaptação, ou reação, à desvios de conceito são estratégias utilizadas para atualizar os modelos de aprendizado existentes de acordo com a desvio [Gu 2019].

2.2.4.1 Retreino de modelos

A maneira mais comum de se tratar desvios de conceito é por meio do retreino dos modelos, sendo uma estratégia de fácil implementação e podendo ser aplicada a qualquer modelo preditivo. A reciclagem deve ser feita utilizando dados mais recentes que os utilizados previamente no treino, substituindo o modelo obsoleto por um novo. É necessário um detector de desvio de conceito explícito para decidir quando treinar novamente o modelo, conforme citado na seção 2.2.3.

Para esse tipo de reação à desvios de conceito, deve-se adotar uma estratégia de janelamento de forma a preservar os dados mais recentes para o retreino e os dados mais antigos para testes. Ao adotar uma estratégia baseada em janela, testes devem ser feitos para decidir o tamanho de janela apropriado para o modelo a ser retreinado. Uma janela pequena pode refletir melhor a distribuição de dados mais recente, mas uma janela grande fornece mais dados para o treinamento de um novo modelo [Gu 2019].

A principal limitação é que a estratégia de retreino de modelos traz é o custo computacional para lidar com os tamanhos das amostras em tempo real.

2.2.4.2 Modelos adaptativos

Uma alternativa para treinar um modelo de aprendizado de máquina está no desenvolvimento de um modelo que aprenda de forma adaptável a partir dos dados alterados, de forma que esse modelo tenha a capacidade de se atualizar parcialmente quando a distribuição de dados subjacentes é alterada. Essa abordagem é sem dúvida mais eficiente quando a deriva ocorre apenas em regiões locais [Gu 2019].

Os algoritmos de aprendizado adaptável podem ser vistos como algoritmos avançados de aprendizado incremental, capazes de se adaptar à evolução do processo de geração de dados ao longo do tempo [Gama et al. 2014]. Para [Gu 2019] modelos adaptativos oferecem bom desempenho apenas com modificações locais, sendo comumente usados na prática. A única limitação é que esses métodos são projetados especificamente para modelos de árvore de decisão, pois tais modelos são capazes de examinar e se adaptar a cada sub-região separadamente.

Os métodos de *ensembles* (compreendem um conjunto de classificadores de base que podem ter tipos ou parâmetros diferentes) utilizam as saídas de cada classificador base, combinando-as por meio de regras de peso e votação para prever os dados recém-chegados. Muitos métodos de *ensembles* adaptativos foram desenvolvidos com o objetivo de lidar com a deriva de conceitos, estendendo os métodos clássicos de *ensembles* ou criando regras específicas de votação adaptativa [Gu 2019]. *Bagging*, *Boosting* e *Random Forests* são métodos clássicos de *ensemble* usados para melhorar o desempenho de classificadores únicos, sendo eles passiveis de extensão para lidar com dados de *streaming* com desvio de conceito.

2.3 Aprendizado de Máquina com Dados em *Stream* (*AutoML-DS*)

A adoção do aprendizado de máquina nas empresas atingiu um estágio de maturidade em que as metodologias estão sendo debatidas. À medida que o aprendizado

de máquina se torna mais amplamente usado, muitas organizações estão adaptando os processos que eles usaram no desenvolvimento de software também para criar produtos de dados [Lorica e Paco 2018].

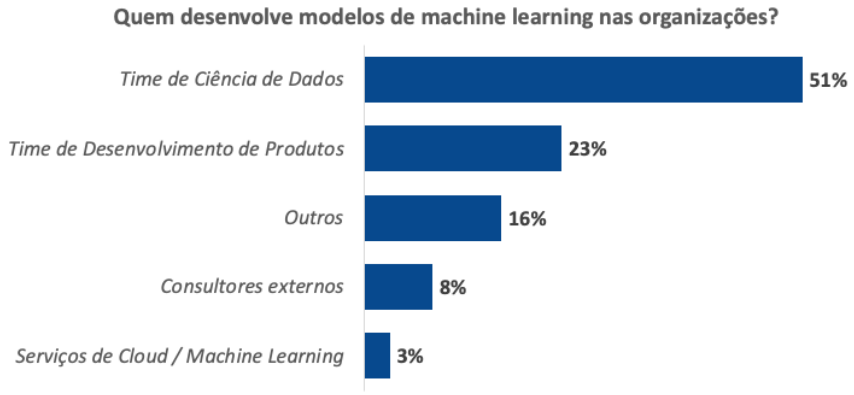

Figura 2.13: Quem desenvolve modelos de *machine learning* nas organizações? [Lorica e Paco 2018].

Em sua pesquisa "*The State of Machine Learning Adoption in the Enterprise*", [Lorica e Paco 2018] demonstra na figura 2.13 que cerca da metade das organizações entrevistadas possuem um time de ciência de dados que é responsável pela criação de modelos de *machine learning*, 8% das organizações contam com consultores externos; enquanto poucos entrevistados pertenciam a organizações que dependem dos serviços Auto *machine learning* oferecidos por provedores de nuvem.

As técnicas de mineração de dados e, mais especificamente, a prática preditiva da mineração de dados, conhecida como aprendizado de máquina, não são novas. O reconhecimento de padrões, a modelagem estatística, a descoberta de conhecimento e o aprendizado de máquina são campos ativos de pesquisa, embora principalmente em ambientes bastante específicos, próximos a pesquisas científicas, governamentais e da indústria, bem como em certos domínios de negócios [Mackenzie 2015].

O aprendizado de máquina utiliza dados históricos para o treinamento de modelos preditivos capazes de solucionar problemas de estimação e classificação. No entanto, tais modelos podem perder desempenho, sofrendo de um processo definido como desvio de conceito (demonstrado na sessão 2.2). Para tentar suprir a lacuna de treinamento entre dois instantes no tempo, apresenta-se o conceito de aprendizado com dados em *streaming* ou aprendizado *online*.

As fontes de dados estão se tornando cada vez mais onipresentes, mais velozes e mais acessíveis em relação à década anterior. Esse fato impulsionou o desenvolvimento de algoritmos e técnicas de aprendizado de máquina capazes de trabalhar com o fluxo

constante de chegada de novos dados em tempo real, e agora faz-se necessário que tais métodos sejam transferidos dos laboratórios e centros de pesquisas para o mercado profissional, como foi aconteceu com os métodos tradicionais de aprendizado de máquina [Gomes et al. 2019].

É certo que a análise de *big data* é capaz de fornecer *insights* importantes para os negócios [Benczúr, Kocsis e Pálovics 2018]. No entanto, são gerados dados a partir de várias fontes, em um volume extremo e o fato de que esses dados chegam continuamente em fluxos múltiplos, rápidos, variando no tempo, possivelmente imprevisíveis e ilimitados pode gerar alguns problemas de fundamentalmente novos.

A classificação é apontada como um dos problemas mais amplamente estudados quando o assunto é mineração de dados e aprendizado de máquina [Hoens, Polikar e Chawla 2012], onde os modelos preditivos devem tentar aprender conceitos de um conjunto de dados estático, cujas instâncias pertencem a uma distribuição subjacente definida por uma função de custo. Desse modo, assume-se que o conjunto de dados contém informações relevantes que são necessárias para aprendizado pertencente à função custo aplicada em dados não vistos.

[Benczúr, Kocsis e Pálovics 2018] compara o processamento de dados tradicional, onde pressupõe que os dados fiquem disponíveis para acesso, mesmo que em alguns casos seu processamento seja realizado em blocos maiores, sendo realizado em *batchs*; já no processamento de dados em *streaming*, os dados chegam continuamente em um fluxo, que deve ser processado por um sistema com recursos limitados, tendo a restrição de memória como principal dificultador do processo.

Uma das premissas da mineração de dados tradicional é que cada conjunto de dados é gerado a partir de uma única função oculta estática, sendo ela utilizada para o treinamento e para o teste dos modelos preditivos. Já em modelos preditivos que utilizam dados de *streaming*, isso não precisa ser verdade, visto que a função que gera instâncias em um instante de tempo t, não precisa ser a mesma função que irá gerar instâncias no instante de tempo $t+1$ [Hoens, Polikar e Chawla 2012].

[Gomes et al. 2019] aborda em seu trabalho, que o aprendizado de máquina para dados em *streaming* teve, nos últimos anos, foco dedicado ao aprendizado supervisionado. Tal fato vem mudando, sendo possível encontrar trabalhos voltados para a clusterização, mineração, detecção de anomalias, seleção de atributos, aprendizado com *multi-label*, aprendizado semi-supervisionado, dentre outros.

A preparação de dados é uma parte essencial de uma solução de aprendizado de máquina, conforme demonstrado na sessão 2.1.1. Tendo como principais objetivos a aprendizagem de algoritmos capazes de lidar com tais dados, visando a melhoria do aprendizado, extraindo ou mantendo apenas os dados relevantes para o modelo. Já no mundo real, os problemas exigem transformação de dados brutos, por meio de etapas

de pré-processamento e que envolvem a seleção de dados relevantes antes que possam ser usados para construir modelos preditivos. No entanto, o pré-processamento de dados em *streaming* pode ser complicado, pois as estatísticas sobre os dados são desconhecidas a priori, por exemplo, os valores médios, valores de desvios padrão, valores mínimo e máximo que um determinado atributo pode conter [Gomes et al. 2019].

As mesmas operações de pré-processamento e seleção de atributos realizadas em modelos preditivos treinados com conjunto de dados tradicionais são necessárias para conjuntos de dados em *streaming*. A principal diferença, apontada por [Gomes et al. 2019], é que em um conjunto de dados em streaming faz-se necessária a aplicação contínua de todo o pipeline enquanto o modelo está em uso. Com isso, as etapas descritas na sessão 2.1 são intercaladas em um processo *online*, o que, idealmente, não depende da execução de algumas tarefas *offline*.

[Benczúr, Kocsis e Pálovics 2018] afirma que ao utilizar modelos de aprendizado de máquina para dados em *streaming*, deve-se ter atenção as considerações algorítmicas e estatísticas. O autor aponta que um problema comum em tais modelagens é a restrição do modelo computacional, sendo que estes não armazenam as entradas e com isso torna-se inviável desfazer uma decisão ou previsão tomada com base em dados históricos.

Um segundo problema apontado pelo autor, que corrobora com a ideia de [Gomes et al. 2019], está na mudança dos dados ao longo do tempo que podem sofrer com as alterações em suas distribuições, causadas em sua maioria por desvios de conceito. Entretanto, para esse cenário, os modelos de aprendizado de máquina treinados com dados em *streaming* são favoráveis, sendo capazes de se adaptar a nova distribuição.

[Hoens, Polikar e Chawla 2012] levanta, em sua obra, outro desafio para modelos de aprendizado de máquina, que surge quando se assume que a prevalência de uma classe no conjunto de treino e teste existe, e que tal classe, em modelos em produção, permanecerá equivalente. Segundo o autor, essa suposição não deve ser aplicada à modelos preditivos que utilizam dados em *streaming*, pois suas distribuições podem se tornar altamente desequilibradas. Desse modo, uma classe minoritária em um conjunto de dados estáticos, pode se tornar mais sub-representada em conjunto de dados em *streaming*.

O avanço do poder computacional estimulou o crescente número de algoritmos de aprendizado de máquina com dados em *streaming*, no entanto as métricas e os meio de avaliação dos modelos que utilizam deste recurso ainda é um problema em aberto [Gama, Sebastiao e Rodrigues 2013]. Os principais dificultadores do processo de avaliação são:

- ter um fluxo contínuo de dados ao invés de uma amostra fixa de exemplos independentes e distribuídos de forma idêntica (i.d.i.);
- os modelos preditivos passam a evoluir com o tempo, não sendo estáticos; e

- os dados são gerados por distribuições não estacionárias, e não por uma amostra de distribuição única.

A tabela 2.1 detalha as diferenças do aprendizado de máquina quando o treinamento é realizado com dados em *batch*, aprendizado tradicional, e o aprendizado com dados em *streaming*. Tais diferenças influenciam na forma de avaliação de ambos os modelos [Gama, Sebastiao e Rodrigues 2013].

	BATCH	**STREAM**
Tamanho dos dados	Dataset finito	Fluxo contínuo
Distribuição dos dados	i.d.i.	Não-i.d.i.
Evolução dos dados	Estático	Não estacionário
Construção do modelo	Em lotes	Incremental
Estabilidade do modelo	Estático	Em evolução
Outras observações	Independente	Dependente

Tabela 2.1: Diferenças entre o aprendizado com dados em *batch* e o aprendizado com dados *streaming*, [Gama, Sebastiao e Rodrigues 2013].

[Gama, Sebastiao e Rodrigues 2009] afirma que em modelos de aprendizado de máquina que utilizam dados em *streaming*, onde os dados usados como *input* são potencialmente infinitos, as técnicas de re-amostragem e validação cruzada não são aplicáveis; sendo técnicas apropriadas para avaliação e melhora de desempenho em modelos treinados tradicionalmente utilizando conjuntos de dados finitos.

Dito isto, [Gama, Sebastiao e Rodrigues 2009], levantam duas possibilidades de avaliação dos modelos treinados utilizando dados em *streaming*:

- Manter um conjunto de teste independente: onde o modelo preditivo é aplicado em um conjunto de dados de teste, em intervalos de tempos regulares;
- Predição sequencial: por meio de avaliação pré-sequencial, onde o erro do modelo preditivo é calculado a partir de uma sequência de exemplos. O erro pré-sequencial é calculado com base em uma soma acumulada de uma função de perda entre a previsão e os valores reais, definido pela função:

$$S = \sum_{i=1}^{n} L(y_i, \hat{y}_i) \tag{2-1}$$

[Gama, Sebastiao e Rodrigues 2009] ressaltam que o valor verdadeiro de y_i não é necessário para todos os pontos dentro de um conjunto de dados em *streaming*, sendo

possível calcular a função de perda S_i para os pontos onde y_i é conhecido, utilizando modelos de *feedback*.

A perda média é dada por $M = \frac{1}{n} x S$, sendo possível estimar um intervalo de confiança para qualquer função de perda $M \pm \varepsilon$, usando limites de Chernoff:

$$\varepsilon_c = \sqrt{\frac{3 x \overline{\mu}}{n} ln(2/\delta)}, \tag{2-2}$$

onde δ é o coeficiente de confiança definido pelo usuário. No caso de funções de perda limitadas, como os limites entre 0-1, o limite de Hoeffding pode ser usado:

$$\varepsilon_h = \sqrt{\frac{R}{2n} ln(2/\delta)}, \tag{2-3}$$

sendo R é o intervalo da variável aleatória. Em ambos os casos, é utilizado a soma de variáveis aleatórias independentes e fornecem uma aproximação relativa ou absoluta do desvio de X de sua expectativa. Sendo ele independentes da distribuição da variável aleatória [Gama, Sebastiao e Rodrigues 2009].

CAPÍTULO 3

Trabalho Proposto

3.1 Formulação do problema

O *churn* do consumidor é definido como o ato de pedido de interrupção da realização de negócio com uma empresa em um determinado período [Ma, Tan e Shu 2015]. Este problema pode ser formulado como um problema de classificação que utiliza diversos tipos de dados financeiros, de consumo do produto e de níveis de satisfação para prever o ato do pedido de cancelamento.

No contexto deste trabalho, a previsão de evasão dos clientes utilizando algoritmos de aprendizado de máquina são modelados como problemas de classificação binária [Stripling et al. 2015]. A construção de um classificador deve ser capaz de discriminar as instâncias das duas classes consideradas a partir dos atributos utilizados como entrada. No contexto da previsão de *churn*, as instâncias são clientes e os rótulos de classe são *churn* e *não churn*, [Nguyen 2011].

Segundo [Nguyen 2011], a entrada para um classificador é um conjunto de instâncias (x^1, y^1), (x^2, y^2), ..., (x^n, y^n). Cada instância $x \in \mathbb{R}^d$ é apresentada na forma de vetor de características d-dimensionais $x = [x_1, x_2, ..., x_d]$ e a classe, em classificações binárias, $y \in \{0, 1\}$. A notação construída por [Nguyen 2011] é utilizada como base no contexto deste trabalho.

No contexto deste trabalho, o cliente é considerado *churner* quando ele interrompe o contrato de assinatura antes do prazo pré-determinado no momento da contratação do serviço. Dessa forma os atributos devem refletir indícios do desejo de cancelamento em qualquer período de tempo, entre a data de início e de término contratual, que antecede à interrupção do serviço. A saída de um classificador capaz de prever *churn* pode ser definida formalmente da seguinte forma:

$$\text{Churn}(\mathbf{x}, c) = P(Class = Churner|\mathbf{x}) \tag{3-1}$$

Onde:

- $\mathbf{x}$: conjunto de atributos no momento do cancelamento ou término do contrato;

- c: contrato/período de fornecimento do serviço;

3.2 Fonte e descrição de dados

Para este experimento foi utilizado um conjunto de dados de serviços de assinatura. Sendo um problema comum em empresas que não possuem cultura voltada para a utilização de dados, o *dataset* é fruto de um trabalho de análise, limpeza e correções que duraram cerca de 8 meses. Foram utilizados dados a parir de 01/01/2019, tendo o conjunto de dados resultante 18.354 instâncias (clientes), cada uma contendo 54 atributos (variáveis), sendo uma delas o rótulo de saída (variável dependente) e 53 são preditores (variáveis independentes).

A tabela 3.1 descreve cada uma das variáveis do conjunto de dados usadas neste trabalho, agrupados conforme o tipo de origem.

NOME DA VARIÁVEL	DESCRIÇÃO
CLUBE DO ASSINANTE	
qtd_rec_clube_aces_site	# Reclamação referente a acesso ao site do clube do assinante
qtd_rec_clube_impr_cart	# Reclamação referente a impressão de cartão
qtd_rec_clube_impr_voucher	# Reclamação referente a impressão de voucher
qtd_rec_clube_info_clube	# Reclamação referente a informações do clube
qtd_rec_clube_n_rec_cart	# Reclamação referente a não recebimento de cartão do assinante
qtd_rec_clube_prob_emp_conv	# Reclamação referente as empresas conveniadas
qtd_rec_clube_qual_brinde	# Reclamação referente a qualidade de brindes
qtd_rec_clube_retir_brinde	# Reclamação referente a retirada de brindes
qtd_solic_club_ped_brinde	# Solicitação de brinde
qtd_solic_club_pedi_info_clube	# Solicitação de informação sobre o clube
CONTRATUAL	
canal_venda	Canal de venda
qtd_meses_periodo_atendido	# Meses corridos do período de contrato
qtd_meses_relacionamento	# Meses relacionamento desde o primeiro contrato
tipo_venda	Tipo de venda
tp_pessoa	Tipo de pessoa - física ou jurídica
FINANCEIRO	
ds_forma_pag	Forma de pagamento do plano

esta_em_atraso	Se possui parcelas em atraso
qtd_dias_atraso	# Dias em atraso
qtd_parcelas	# Parcelas totais do contrato
qtd_parcelas_atrasadas	# Parcelas atrasadas do contrato
IMPRESSÃO	
qtd_rec_impr_qualid_foto	# Reclamação referente a qualidade das fotos no jornal
qtd_rec_impr_qualid_text	# Reclamação referente a qualidade dos textos no jornal
DISTRIBUIÇÃO E LOGÍSTICA	
qtd_rec_distrib_entreg_atras	# Reclamação referente a atrasos na entrega
qtd_rec_distrib_jorn_danif	# Reclamação referente ao recebimento de jornal danificado
qtd_rec_distrib_jorn_incomp	# Reclamação referente ao recebimento de jornal incompleto
qtd_rec_distrib_jorn_n_entreg	# Reclamação referente a não entrega do jornal
qtd_rec_distrib_loc_err	# Reclamação referente a entrega realizada em local errado
qtd_rec_distrib_protec_jorn	# Reclamação referente a proteção do jornal entreguE
PUBLICIDADE	
qtd_rec_public_qtd_anunc	# Reclamação referente a quantidade de anúncios no jornal
qtd_rec_public_tp_anunc	# Reclamação referente aos tipos de anúncios no jornal
CONTEÚDO EDITORIAL	
qtd_insat_polit_jorn	# Reclamação referente a insatisfação com a posição política do jornal
qtd_rec_redac_colunistas	# Reclamação referente aos colunistas do jornal
qtd_rec_redac_cont_editori	# Reclamação referente ao conteúdo editorial
qtd_rec_redac_cont_fotos	# Reclamação referente ao conteúdo das imagens
qtd_rec_redac_formato	# Reclamação referente ao formato
RELACIONAMENTO	
qtd_reclamacao	# Reclamações totais
VENDA	
qtd_rec_vend_acess_online	# Reclamação referente ao acesso online

qtd_rec_vend_insist_vendedor_reten	# Reclamação referente a insistência da equipe de retenção
qtd_rec_vend_insist_vendedor_vend	# Reclamação referente a insistência de venda pelo vendedor
qtd_rec_vend_pos_venda	# Reclamação referente a pós-venda
qtd_rec_vend_qualid_atend_venda	# Reclamação referente a qualidade da venda
qtd_rec_vend_qualid_brinde	# Reclamação referente a qualidade do brinde
qtd_rec_vend_retir_brinde	# Reclamação referente a retirada de brindes
qtd_solic_vend_agend_parc	# Solicitação de agendamento de parcela
qtd_solic_vend_cancel	# Solicitação de cancelamento de contrato
qtd_solic_vend_contr_assin	# Solicitação de informação sobre o contrato vigente
qtd_solic_vend_emiss_2_via_carn	# Solicitação de segunda via de carnê
qtd_solic_vend_info_agreg	# Solicitação de agregados
qtd_solic_vend_info_brind	# Solicitação de informações de brindes
qtd_solic_vend_info_classif	# Solicitação de informações sobre o classificados
td_solic_vend_nf_assin	# Solicitação de nota fiscal
qtd_solic_vend_ped_brind_jorn	# Solicitação de brinde
qtd_solic_vend_suspen	# Solicitação de suspensão temporária de contrato
VARIÁVEL TARGET	
situacao_periodo*	Situação do período do contrato

Tabela 3.1: Visão geral das variáveis na análise.

A variável *target* possui duas classes, abaixo discriminadas:

- Cancelado: contrato finalizado antes da data de término prevista, com a intervenção do assinante (classe de *churn*, assumindo valor 1). O dataset utilizado possuí 7.005 instâncias presentes nesta classe;
- Atendido: contrato finalizado na data de término prevista, sem a intervenção do assinante (classe de não-*churn*, assumindo valor 0). O dataset utilizado possuí 11.349 instâncias presentes nesta classe.

3.3 Engenharia de dados e seleção de atributos

Em modelos de predição de *churn* é necessário que os dados reflitam os indícios do desejo de cancelamento por parte dos clientes, para que se obtenha uma boa predição.

Sendo assim o processo de limpeza de base se deu por meio das atividades abaixo listadas:

- Corte na base histórica: foram considerados para o treinamento contratos com data de ativação igual ou superior a 01/01/2019, levando em consideração que os hábitos de consumo sofreram alterações durantes os anos;
- Exclusão de pré contratos: contratos onde não houve o pagamento da primeira parcela na data de corte foram desconsiderados;
- Exclusão de contratos governamentais: contratos de governo foram excluídos por não serem passíveis de ações, como os contratos não governamentais;
- Exclusão de contratos promocionais: contratos cortesias e degustação foram retirados do dataset por serem promoções permitidas para a atração de novos clientes, com data de início e término definidas, com no máximo 7 dias de duração.

Para a construção do modelo preditivo faz-se necessário re-codificar os atributos preditores categóricos, transformando-os em numéricos. Tal codificação não altera o teste do efeito geral da preditora. Atributos com datas foram transformados em contínuos, como por exemplo 'data_ativação' foi utilizada para calcular o atributo 'qtd_meses_relacionamento'.

[Zhu, Baesens e Broucke 2017] afirma que o desequilíbrio de classe é uma característica importante de muitos conjuntos de dados usados para modelagem de previsão de *churn*. A maioria dos algoritmos de classificação frequentemente possui um viés em relação à classe majoritária; portanto, o desequilíbrio de classe dificulta o desempenho dos classificadores, especialmente em termos da classe minoritária de interesse. No *dataset* de treinamento a classe dominante possuí 61,8% das instâncias e apenas 38,2% para a classe de interesse (contratos cancelados), não sendo necessário técnicas de balanceamento.

A seleção de atributos fornece uma maneira eficaz de resolver problemas decorrentes do número excessivo de variáveis preditoras, removendo dados irrelevantes e redundantes, o que pode otimizar o tempo/custo computacional, visando a melhora da precisão do aprendizado de forma a facilitar o entendimento do modelo e/ou dos dados do aprendizado [Cai et al. 2018]. Tendo esse conceito em vista, o enxugamento no número de atributos preditores foi realizado sem que a performance do modelo fosse prejudicada, o resultado desse processo encontra-se na tabela 3.2.

NOME DA VARIÁVEL	DESCRIÇÃO
PREDITORES	
VL_ASS	Valor total do plano contratado
QTD_MESES_RELACIONAMENTO	# de meses relacionamento com o cliente, desde seu primeiro contrato
QTD_ACESSO_30	# acessos ao serviço digital disponível no contrato

QTD_PARCELAS_ATRASADAS	# de parcelas em atraso
TT_RECLAMACOES	# total de reclamação registradas na Central de Relacionamento
TT_SOLICITACOES	# total de solicitações registradas na Central de Relacionamento
FPGT_BOLETO	Se a forma de pagamento = boleto bancário, então 1, senão 0
FPGT_CARTAO	Se a forma de pagamento = cartão de crédito, então 1, senão 0
FPGT_DEBITO	Se a forma de pagamento = débito em conta, então 1, senão 0
VARIÁVEL TARGET	
CHURN*	Situação do período do contrato

Tabela 3.2: Visão geral das variáveis na análise, após engenharia de dados.

3.4 Solução implementada

3.4.1 Conjuntos de treinos e testes

Foram adotados carregamentos periódicos de novos dados com o intuito de avaliar o desempenho da solução implementada com aprendizado utilizando dados em *stream* (*auto machine learning with data stream - AutoML-DS*). Por meio de carregamentos semanais, os novos dados para os treinamentos incrementais são inseridos no modelo *AutoML-DS* há cada 7 dias. A tabela 3.3 demonstra o período de tempo compreendido por cada *dataset* incremental e a quantidade de instâncias.

Dataset	Período de carregamento	N. Instâncias
T0	01/01/2019 - 31/10/2020	14.017
T1	01/11/2020 - 07/11/2020	238
T2	08/11/2020 - 14/11/2020	218
T3	15/11/2020 - 21/11/2020	153
T4	22/11/2020 - 28/11/2020	202
T5	29/11/2020 - 05/12/2020	173
T6	06/12/2020 - 12/12/2020	184
T7	13/12/2020 - 19/12/2020	188
T8	20/12/2020 - 26/12/2020	180
T9	27/12/2020 - 02/01/2021	190

PRED1	03/01/2021 - 15/03/2021	2.611

Tabela 3.3: Períodos de divisão do *dataset* e número de instâncias

Modelos de *AutoML-DS* aprendem de maneira sequencial conforme os dados se tornam disponíveis em tempo real, ao contrário do treinamento de modelos convencionais, onde o treinamento deve ser feito com dados históricos. Portanto, no *AutoML-DS*, não há treinamento em dados de histórico estático, visto que o modelo consegue aprender na hora.

3.4.2 Modelos implementados

Foram utilizados três modelos preditivos que permitem a aplicação de *AutoML-DS*, a saber:

- Modelo linear: *Regressão Logística*;
- Modelo de *ensemble*: *AdaBoost*; e
- Modelo árvore de decisão: *Extremely Fast (Árvore de Decisão EF*).

3.4.2.1 Metodologia de Referência

O método de referência com treinamento convencional dos modelos está ilustrado na figura 3.1.

A figura 3.1 ilustra a metodologia convencional de referência para este trabalho. O *pipeline* de um projeto envolvendo aprendizado de máquina é dividido em duas grandes etapas, onde a primeira etapa é fase de treinamento e, posteriormente, a fase em que o modelo preditivo é posto em produção. Utilizando-se do *Dataset T0* realiza-se o treinamento de modelos de aprendizado de máquina. Em um ambiente de produção, os dados não vistos pelo modelo são preparados e imputados no modelo já treinado com a finalidade de classificar, nesse experimento, se o cliente terá chances de evadir da carteira de assinaturas antes do período de término contratual ou não. Após a inferência do modelo no conjunto de dados T1 - Pred1, a predição obtida passará por avaliação, sendo necessário determinado tempo para reconhecer se o cliente, de fato, evadiu. Com os dados de performance em mãos é possível avaliar o comportamento do modelo em produção, que recebe dados reais a todo momento. Apresentando performance aderente, os resultados podem ser utilizados até que o conjunto de dados a ser imputado tenha sua distribuição alterada e o modelo comece a perder desempenho, o que resulta em desvios de conceito. No experimento, os *datasets T1* à *pred1*, detalhados na tabela 3.3, foram

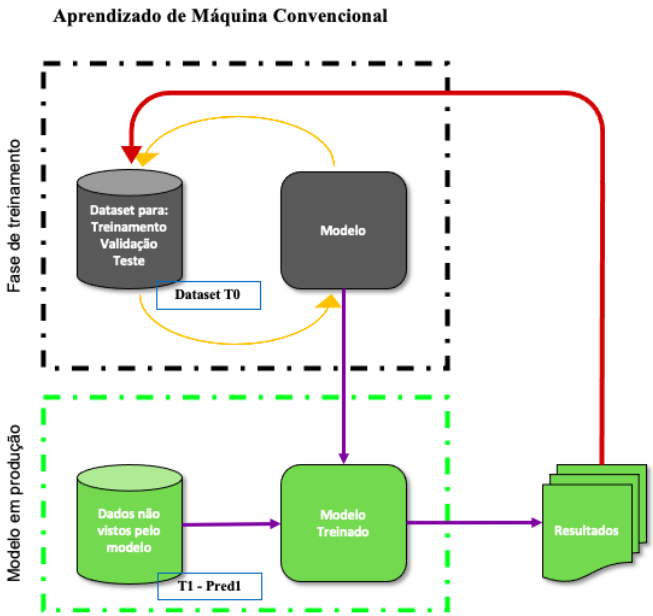

Figura 3.1: Esquema de treinamento convencional.

utilizados para predição do modelo treinado, destacados na etapa em que o modelo está em produção na figura 3.1.

Em ambientes de negócios, perca de desempenho de um modelo preditivo em produção, pode influenciar diretamente as decisões acerca do problema modelado e acarretar em perda de recursos financeiros, prejudicar a credibilidade da equipe responsável por ciência de dados, além de comprometer o relacionamento junto à clientes classificados de forma errônea. Quando o desvio de conceito é identificado, o modelo de aprendizado de máquina tradicional deve ser direcionado para a fase de treinamento, para que possa aprender com a distribuição dos dados mais recentes e ter a performance aprimorada novamente. Este processo está representado pela seta vermelha na figura 3.1.

3.4.2.2 Metodologia com *AutoML-DS*

A figura 3.2 representa o aprendizado de máquina com dados em *stream*. A fase de treinamento é idêntica à metodologia de referência, utilizando, inclusive o mesmo conjunto de dados (*dataset T0)*. Os modelos são colocados em ambiente de produção e espera-se que sejam capazes de reconhecer os padrões, aos quais foram submetidos na fase anterior, em dados não vistos pelos modelos treinados. Nessa etapa, existe a diferenciação da metodologia de referência e a metodologia proposta com aprendizado utilizando dados em *AutoML-DS*.

Conforme ilustrado na figura 3.2, o aprendizado em *AutoML-DS* permite que

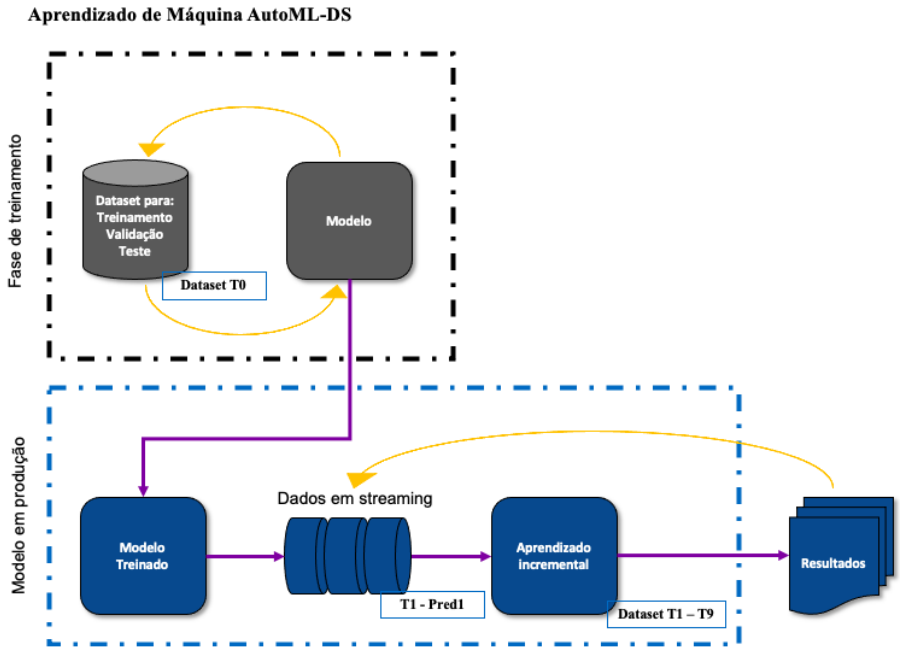

Figura 3.2: Esquema de treinamento utilizando *AutoML-DS*.

os novos dados possam ser acessados na medida em que chegam dentro de um fluxo de inferência e são utilizados para realizar aprendizado incremental ao modelo treinado na etapa anterior (*datasets T1 a T9*). Recebendo novos dados na medida em que chegam em um fluxo contínuo, o aprendizado incremental identifica as possíveis novas distribuições dos dados de modo a ajustar o modelo preditivo. Espera-se que essa dinâmica possa corrigir desvios de conceito que podem surgir ao longo do tempo. Independente dos resultados obtidos após a predição e *outputs* do modelo, o treinamento incremental é contínuo e sem interrupção e não necessita de retreino total, como é feito com modelos de aprendizado de máquina convencional.

A figura 3.3, proposta por [Montiel et al. 2018], ilustra a sequência de treinamento de um modelo que utiliza *AutoML-DS* (*StreamModel*) e monitora o desempenho, utilizando avaliação pré-sequencial, demonstrada por [Gama, Sebastiao e Rodrigues 2009] na sessão 2.3. O *Stream* fornece os dados dentro de um fluxo contínuo, mediante a solicitação do usuário; já o avaliador (*StreamEvaluator*) executa a consulta dos dados em *stream*, além de ser responsável pelo treinamento e teste do modelo preditivo com os dados recebidos, monitorando continuamente a performance do modelo.

O treinamento com *AutoML-DS* se inicia na fase de treinamento ilustradas na figura 3.2, sendo o modelo criado e avaliado pelo usuário (figura 3.3, marcador 1). Após o processo de modelagem, um *loop* de avaliação contínua é executado enquanto houver

dados disponíveis dentro do fluxo. Modelos criados utilizando *AutoML-DS* analisam exemplo a exemplo, de modo a consultar os dados que estão chegando no fluxo *Stream* (figura 3.3, marcador 2), submetendo-os ao pré-processamento de dados, definindo as variáveis dependentes (X) e independente (y_true) (figura 3.3, marcador 3). A partir de então, dá-se início ao *loop* de avaliação pré-sequencial, que é executado enquanto houver exemplos válidos para serem submetidos ao modelo treinado.

O *loop* de avaliação pré-sequencial submete ao *StreamModel* as variáveis preditoras (X) de cada instância, uma a uma (figura 3.3, marcador 4). O *StreamModel*, por sua vez, faz a predição de X, devolvendo a instância classificada (y_predicted) ao *StreamEvaluator* (figura 3.3, marcador 5). Ao *StreamEvaluator* é atribuído a função de avaliar a performance da predição, comparando o y_predicted com a real classe da instância (y_true) (figura 3.3, marcador 6). As métricas são atualizadas (figura 3.3, marcador 7) com o resultado da avaliação da instância classificada, bem como os gráficos referentes a cada métrica (figura 3.3, marcador 8). O *StreamEvaluator* submete a nova instância a ser classificada ao *StreamModel* (figura 3.3, marcador 9). Durante todo o *loop* de avaliação o *StreamEvaluator* devolve ao usuário um modelo já treinado e atualizado conforme as novas distribuições dos dados recebido em *stream* (figura 3.3, marcador 10).

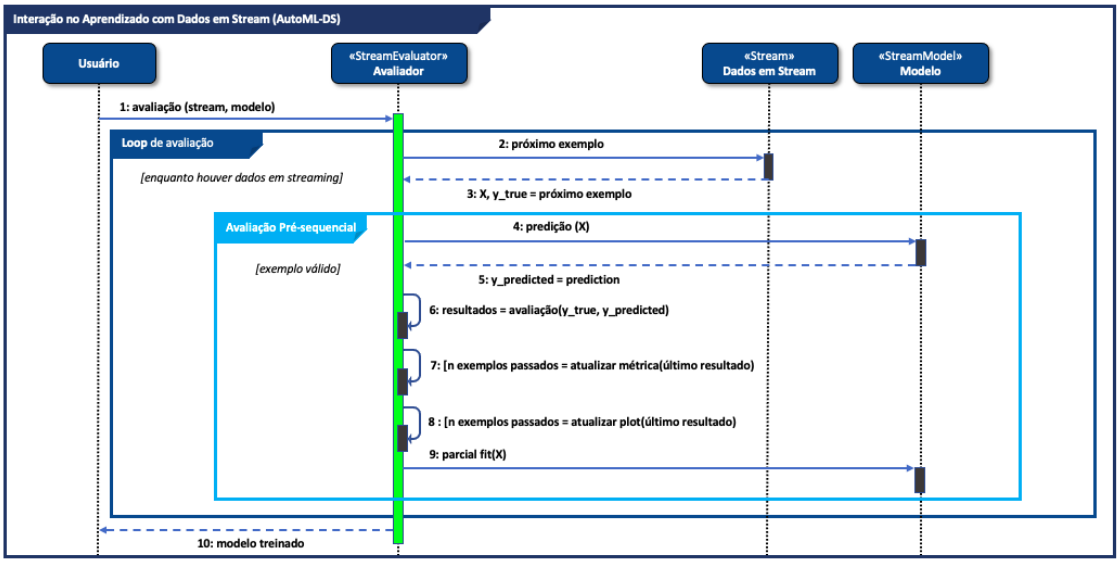

Figura 3.3: Loops de interação no *AutoML-DS*.

O detalhamento do uso dos dados no ciclo de aprendizado dos modelos *AutoML-DS*, da Figura 3.2, está na tabela 3.4. Enquanto o *Dataset* T0 foi utilizando para o primeiro treino de forma offline, os *datasets T1* à *T9* foram usados de forma incremental em cada

instante *T*, sendo eles reconhecidos como dados em *stream*, conforme demonstrado na figura 3.2.

FASE DE TREINAMENTO	MODELO EM PRODUÇÃO	PREDIÇÃO
Treinamento, Teste e Validação		
Dataset T0	–	Dataset T1 - PRED1
	Aprendizado Incremental	
–	Dataset T1	Dataset T2 - PRED1
–	Dataset T2	Dataset T3 - PRED1
–	Dataset T3	Dataset T4 - PRED1
–	Dataset T4	Dataset T5 - PRED1
–	Dataset T5	Dataset T6 - PRED1
–	Dataset T6	Dataset T7 - PRED1
–	Dataset T7	Dataset T8 - PRED1
–	Dataset T8	Dataset T9 - PRED1
–	Dataset T9	Dataset PRED1

Tabela 3.4: *Datasets* utilizados nos ciclos de aprendizado em modelos *AutoML-DS*

3.4.2.3 Métricas de Avaliação

A matriz de confusão é utilizada para demonstrar a eficiência de modelos preditivos usados para classificação. Para cada classe avaliada é comparado o valor resultante do modelo preditivo com o valor conhecido da classe dentro do conjunto de teste. A partir de uma matriz de confusão é possível avaliar os casos Positivos Verdadeiros (TP), Falsos Positivos (FP), Negativos Verdadeiros (TN) e Falsos Negativos (FN), que servem como base para outras métricas de avaliação. A tabela 3.5 foi construída com base nos trabalhos de [Mateen et al. 2020] [Hossin e Sulaiman 2015], apresenta as métricas de avaliação de desempenho dos modelos preditivos usadas neste trabalho.

MÉTRICAS	FOCO DE AVALIAÇÃO/ FÓRMULA
Acurácia	A acurácia mede a proporção de previsões corretas sobre o número total de instâncias avaliadas. $$Acurácia = \frac{TP+TN}{TP+FP+TN+FN} x100$$

Precisão	A precisão é usada para medir os padrões positivos que são previstos corretamente a partir do total de padrões previstos em uma classe positiva. $$Precisão = \frac{TP}{FP + TP}$$
Recall	O recall é resultado do número de exemplos classificados como pertencentes a uma classe, que são de fato desta classe (Positivos verdadeiros) dividido pela quantidade total da classe em questão. $$Recall = \frac{TP}{(TP + FN)}$$
Taxa de Falsos Positivos (FPR)	Representa a taxa de falsos positivos obtidos. $$FPR = \frac{FP}{(FP + TN)}$$
Taxa de Falsos Negativos (FNR)	Representa a taxa de falsos negativos obtidos. $$FNR = \frac{FN}{(FP + TN)}$$

Tabela 3.5: Métricas de avaliação de desempenho utilizadas neste trabalho.

A análise de curva ROC (*Receiver Operating Characteristic*) é uma das abordagens populares que podem ser usadas para medir o desempenho dos classificadores em conjuntos de dados desequilibrados. O gráfico ROC representa taxas positivas verdadeiras versus taxas positivas falsas. Os classificadores podem ser selecionados com base em suas trocas entre verdadeiros positivos e falsos positivos [Zhu, Baesens e Broucke 2017]. Em vez de comparar visualmente as curvas, a área sobre a curva (AUC - *Area Under the Curve*) agrega o desempenho de um modelo de classificação em um único número, o que facilita a comparação do desempenho geral de vários modelos de classificação. Um classificador aleatório tem uma AUC de 0,5 e um classificador perfeito possui uma AUC igual a 1.

CAPÍTULO 4

Performance e Resultados

Este capítulo apresenta os resultados da metodologia de referência e da solução proposta denominada *AutoML-DS*. A compilação estatística dos resultados são provenientes da execução de cada algoritmo nos nove conjuntos de dados (T1-T9) para que se possa avaliar possíveis degradações dos modelos na metodologia de referência e no *AutoML-DS*.

4.1 Resultados da metodologia de referência

A tabela 4.1 apresenta as medidas estatísticas dos resultados gerados pelos modelos preditivos utilizando a metodologia de referência. As medidas estatísticas foram compiladas a partir da execução do mesmo algoritmo nos diferentes conjuntos de dados (T1-Pred1).

MÉTRICA	MÉDIA(%)	DESVIO P.	MÍN.(%)	MÁX.(%)
Regressão Logística				
Acurácia	80,44	2,9221	76,58	85,22
Precisão	55,03	4,6180	49,25	63,20
Recall	93,43	0,3912	92,54	93,75
Taxa de Falsos Positivos	23,48	3,8427	17,1	28,48
Taxa de Falsos Negativos	6,57	0,3912	6,25	7,46
AUC	95,72	0,1795	95,45	96,02
Adaboost				
Acurácia	95,33	0,0372	95,26	95,38
Precisão	92,11	0,3283	91,59	92,71
Recall	87,45	0,1463	87,19	87,66
Taxa de Falsos Positivos	2,27	0,0504	2,19	2,36
Taxa de Falsos Negativos	12,55	0,1463	12,34	12,81
AUC	94,78	0,1477	94,55	95,02
Árvore de Decisão EF				

Acurácia	96,27	0,0234	96,23	96,30
Precisão	96,57	0,1680	96,29	96,82
Recall	87,09	0,1561	86,8	87,28
Taxa de Falsos Positivos	0,94	0,0274	0,9	0,99
Taxa de Falsos Negativos	12,91	0,1561	12,72	13,2
AUC	95,21	0,0979	95,03	95,33

Tabela 4.1: Avaliação dos modelos de metodologia de referência com carregamentos semanais.

Analisando a tabela 4.1, percebe-se que os modelos que utilizam a metodologia de referência são aderentes ao problema de predição de evasão de clientes, tendo uma acurácia acima de 80% nos três modelos utilizados. No entanto, o modelo que adota regressão logística como classificador possui uma precisão relativamente baixa (55,03%) quando comparado aos demais classificadores. Este fato, implica em um classificador com maior número de falsos positivos (23,48%). A taxa de falsos negativos dos modelos que utilizam Adaboost e Árvore de Decisão EF apresenta resultados relativamente melhores com cerca de 12% do total de classificações.

Nota-se que para o modelo que adota regressão logística como classificador, os desvios padrões da acurácia, precisão e taxa de falsos positivos são relativamente mais altos quando comparados aos desvios do *recall*, taxa de falsos negativos e *AUC*. Consequentemente, os valores de mínimo e máximo são mais distantes da média, o que indica uma dispersão maior dos dados.

Os modelos que adotam *Adaboost* e árvore de decisão EF possuem valores mínimo e máximo mais próximos ao valor médio de cada métrica, bem como baixo desvio padrão. Fato que indica que os modelos não se degradam ao longo do tempo, tendo uma constância nas predições de *churn*.

4.2 Resultados do *AutoML-DS*

A tabela 4.2 apresenta as medidas estatísticas dos resultados gerados pelos modelos preditivos utilizando a metodologia proposta por este trabalho, *AutoML-DS*, nos diferentes conjuntos de dados (T1-Pred1).

MÉTRICA	**MÉDIA(%)**	**DESVIO P.**	**MÍN.(%)**	**MÁX.(%)**
Regressão Logística				
Acurácia	95,93	0,0892	95,69	95,98
Precisão	95,24	0,2351	94,59	95,38

Recall	86,86	0,1465	86,59	87,08
Taxa de Falsos Positivos	1,32	0,0970	1,25	1,58
Taxa de Falsos Negativos	13,14	0,1465	12,92	13,41
AUC	95,58	0,0983	95,40	95,69
Adaboost				
Acurácia	96,08	0,2298	95,50	96,24
Precisão	95,74	0,9332	93,37	96,41
Recall	87,08	0,2318	86,73	87,56
Taxa de Falsos Positivos	1,19	0,3088	0,96	1,97
Taxa de Falsos Negativos	12,92	0,2318	12,44	13,27
AUC	94,85	0,1433	94,55	95,02
Árvore de Decisão EF				
Acurácia	96,37	0,0803	96,22	96,47
Precisão	96,84	0,1106	96,64	97,01
Recall	87,26	0,0478	87,17	87,34
Taxa de Falsos Positivos	0,87	0,0480	0,79	0,94
Taxa de Falsos Negativos	12,74	0,0478	12,66	12,83
AUC	95,86	0,3950	95,10	96,35

Tabela 4.2: Avaliação dos modelos *AutoML-DS* com carregamentos semanais.

Por meio da tabela 4.2, é possível observar que os valores médios de cada métrica são próximos para os três modelos implementados. Observa-se que os valores mínimos e máximos estão próximos aos valores médios de cada métrica.

Em cada um dos instantes de tempo, o modelo recebe dados em *stream* para que seja realizado o treinamento incremental do modelo já treinado. Quando observado os desvios padrão de cada métrica, em cada modelo, é possível inferir que os modelos são capazes de suprir os desvios de conceito que surgem ao longo do tempo, mantendo uma constância no aprendizado e na predição.

4.3 Metodologia de Referência *versus* do *AutoML-DS*

Os modelos *AutoML-DS* possuem uma performance média geral mais apurada que a metodologia de referência. Tal performance é medida pela métrica acurácia (96,12%, média entre os 3 modelos implementados), que indica que os modelos possuem uma proporção de previsões corretas maior que os modelos que necessitam de retreino periódico (90,68%, média entre os 3 modelos implementados). O resultado positivo é

observado nos três modelos que utilizam *AutoML-DS*, regressão logística, Adaboost e Árvore de Decisão EF, com acurácia média de 95,93%, 96,08% e 96,37%, respectivamente.

Por meio da precisão, podemos notar que os modelos *AutoML-DS* possuem vantagem sobre os modelos convencionais, onde as predições corretas apresentam melhores médias individuais, onde o modelo que utiliza regressão logística possuí uma precisão de 95,24%, o modelo com Adaboost 95,74% e o modelo com árvore de decisão EF com 96,84% de precisão.

A proporção de falsos positivos também é melhor aferida nos resultados gerados pelos modelos *AutoML-DS*. Enquanto o modelo convencional utilizando regressão logística possuí uma taxa de falsos positivos de 23,48%, o modelo que utiliza *AutoML-DS* possuí uma taxa de falso positivos significativamente menor (1,32%); já o modelo *AutoML-DS* Adaboost possuí uma taxa de falso positivo de 1,19% e o modelo de referência possuí 2,27% de taxa de falsos positivos; e, por fim, o modelo convencional que utiliza árvore de decisão EF apresentou uma taxa de falso positivos de 0,94%, sendo também superado pelo modelo que utiliza *AutoML-DS* (0,87%).

Por outro lado, os modelos da metodologia de referência se sobressaem ao avaliarmos o *recall*. O *recall* médio para o modelo convencional que utiliza regressão logística é de 93,43%, já o adaboost possuí *recall* médio de 87,45%, sendo a média estatística para o modelo *AutoML-DS* com regressão logística 86,86% e 87,08% para o modelo *AutoML-DS* com Adaboost. O modelo *AutoML* utilizando árvore de decisão EF possuí um resultado levemente melhor para o *recall*, 87,26%, quando comparado ao modelo que utiliza a metodologia de referência, 87,09%. Os resultados inferiores dos modelos propostos neste trabalho frente a metodologia de referência não comprometem a qualidade dos classificadores, visto que o *recall* de ambos é superior a 86%, e possuem *gap* inferior a 7% quando comparados aos resultados do modelo convencional.

Ao analisarmos os resultados médios da taxa de falsos negativos é possível observar que o modelo que utiliza a metodologia de referência com regressão logística (6,57%) se sobressai ao modelo proposto por este trabalho, *AutoML-DS*, (13,14%). O modelo *AutoML-DS* com Adaboost também possuí taxas de falsos negativos superior aos resultados dos modelos de referência para a mesma métrica, sendo o resultado do modelo *AutoML-DS* com adaboost 12,92% e do modelo convencional 12,55%. Já o modelo *AutoML-DS* com árvore de decisão EF possuí taxa de falsos negativos de 12,74%, um pouco melhor que o modelo convencional, que possuí 12,91% de taxa de falsos negativos.

Os modelos *AutoML-DS* que utilizam técnicas de *ensemble* (Adaboost) ou árvore de decisão EF possuem valores de *AUC* (94,85% e 95,86%, respectivamente). Estes valores são próximos dos resultados utilizando modelos convencionais (94,78% e 95,21%, respectivamente). O modelo *AutoML-DS* que utiliza regressão logística possuí 95,58% de *AUC* e fica atrás do modelo convencional com a mesma técnica de classificação (95,72%).

No entanto o modelo *AutoML-DS* possuí menor dispersão nos resultados de tais métricas, o que indica uma maior constância no desempenho do modelo ao longo dos instantes T.

4.3.1 Análise de Desvio de Conceito

A tabela 4.3 apresenta os resultados de cada métrica utilizada para a avaliação dos modelos nos instantes *T0* e *T9*. A tabela traz as respectivas variações dos resultados entre os dois pontos, para os modelos criados utilizando a metodologia de referência.

Por meio da tabela 4.3 é possível notar que o modelo da metodologia de referência que utiliza regressão logística como classificador, apresenta perda de desempenho entre os instantes *T0* e *T9* nas métricas acurácia (variação de -8,64), precisão (-13,95), taxa de falsos positivos (+11,38) e AUC (-0,57). Em contrapartida o modelo teve uma leve melhora na performance de *recall* (+1,18) e, consequentemente, na taxa de falsos negativos (-1,18). Já os modelos que utilizam Adaboost e Árvore de Decisão EF apresentam perda de desempenho em todas as métricas avaliadas, visto que a variação de performance em tais modelos são pequenas, este resultado pode indicar que os modelos não se degradam ao longo do tempo.

MÉTRICA	INSTANTE T0(%)	INSTANTE T9(%)	VARIAÇÃO
Regressão Logística			
Acurácia	85,22	76,58	-8,64
Precisão	63,20	49,25	-13,95
Recall	92,54	93,72	1,18
Taxa de Falsos Positivos	17,10	28,48	11,38
Taxa de Falsos Negativos	7,46	6,28	-1,18
AUC	96,02	95,45	-0,57
Adaboost			
Acurácia	95,37	95,26	-0,11
Precisão	92,71	91,59	-1,12
Recall	87,66	87,19	-0,47
Taxa de Falsos Positivos	2,19	2,36	0,17
Taxa de Falsos Negativos	12,34	12,81	0,47
AUC	95,02	94,55	-0,47
Árvore de Decisão EF			
Acurácia	96,24	96,23	-0,01
Precisão	96,82	96,29	-0,53
Recall	87,27	86,80	-0,47
Taxa de Falsos Positivos	0,91	0,99	0,08

Taxa de Falsos Negativos	12,73	13,20	0,47
AUC	95,30	95,03	-0,27

Tabela 4.3: Variação do desempenho (instante T0 para instante T9) dos modelos da metodologia de referência com carregamentos semanais.

A tabela 4.4 apresenta os resultados referentes ao modelo proposto, *AutoML-DS*. Os modelos *AutoML-DS* são treinados por meio do aprendizado incremental, detalhados no capítulo 3, sendo os instantes T0 e T9 os pontos de incremento de dados aos modelos já treinado no instante T0. Por meio da tabela 4.4, nota-se que a acurácia, a precisão e as taxas de falso positivos tiveram melhora de desempenho entre os 2 pontos, nos 3 modelos implementados. Este fato indica que os treinamentos incrementais são capazes de suprir os possíveis desvios de conceitos que eventualmente surgem, pois o desempenho nas três métricas citadas não sofre degradação ao contrário dos modelos da metodologia de referência.

Os modelos convencionais possuem um *recall* médio mais preciso que os modelos *AutoML-DS*, excetuando o modelo em que o classificador é uma árvore de decisão, classificando corretamente os casos positivos pertencentes a classe de *churn*. Sendo este um indicador importante para problemas de classificação, onde a classe positiva deve servir para direcionar e orientar ações pelas organizações. Em contrapartida, modelos *AutoML-DS* têm menor variação entre os instantes T0 e T9, mostrando ser mais constante nas predições. Como a taxa de falsos negativos é inversamente proporcional ao *recall*, a performance nesse indicador também é mais precisa nos modelos da metodologia de referência.

MÉTRICA	**INSTANTE T0(%)**	**INSTANTE T9(%)**	**VARIAÇÃO**
Regressão Logística			
Acurácia	95,69	95,98	0,29
Precisão	94,59	95,35	0,73
Recall	87,08	86,59	-0,49
Taxa de Falsos Positivos	1,58	1,27	-0,33
Taxa de Falsos Negativos	12,92	13,41	0,49
AUC	95,62	95,40	-0,22
Adaboost			
Acurácia	95,50	96,24	0,74
Precisão	93,37	96,38	3,01
Recall	87,56	86,73	-0,83

Taxa de Falsos Positivos	1,97	0,96	-1,01
Taxa de Falsos Negativos	12,44	13,27	0,83
AUC	94,55	95,02	0,47
Árvore de Decisão EF			
Acurácia	96,22	96,47	0,25
Precisão	96,71	97,01	0,30
Recall	87,27	87,17	-0,10
Taxa de Falsos Positivos	0,94	0,79	-0,15
Taxa de Falsos Negativos	12,73	12,83	0,10
AUC	95,10	96,35	1,25

Tabela 4.4: Variação do desempenho (instante T0 para instante T9) dos modelos *AutoML-DS* com carregamentos semanais.

4.3.2 Síntese Comparativa

A tabela 4.5 apresenta um resumo comparativo entre o modelo proposto, *AutoML-DS*, e a metodologia de referência. Por meio da tabela, apresenta-se qual dos modelos possuí melhor performance média em cada uma das métricas de desempenho. Considera-se neste julgamento somente os valores absolutos que foram medidos. As últimas duas colunas indicam se os modelos tiveram ganho ou perda de desempenho do instante T0 ao instante T9.

O modelo *AutoML-DS* que utiliza regressão logística como classificador apresente melhor desempenho em 3 métricas (acurácia, precisão e taxa de falsos positivos) frente aos modelos que adotam a metodologia de referência, que, por sua vez, está a frente das métricas de *recall*, taxa de falsos negativos e AUC. A variação dos resultados dos modelos *AutoML-DS* são positivas nas mesmas métricas de melhor desempenho. Já o modelo de referência apresenta melhora no *recall* e, consequentemente, na taxa de falsos positivos.

Ao observarmos o modelo *AutoML-DS* com adaboost, notamos que, comparado à metodologia de referência, se sobressaí na acurácia, precisão, taxa de falsos positivos e AUC, tendo ganho de performance nas mesmas métricas ao longo dos instantes T0 e T9. O modelo que utiliza a metodologia de referência, necessitando de retreino periódico, apresenta melhor desempenho frente ao modelo com dados em *stream* no *recall* e na taxa de falsos negativos, porém o modelo apresenta perda de desempenho em todas as métricas avaliadas, entre o instante T0 e T9.

Já o *AutoML-DS* com árvore de decisão EF apresenta melhor performance em

todas as métricas avaliadas, quando comparados ao método de referência. O modelo apresenta perda de desempenho apenas no *recall* e na taxa de falsos negativos. Por outro lado, o modelo convencional apresenta perda de performance em todas as métricas utilizadas para avaliar o desempenho dos modelos.

MÉTRICA	**MELHOR DESEMPENHO**	**VARIAÇÃO ENTRE T0 E T9**	
		AUTOML-DS	**REFERÊNCIA**
Regressão Logística			
Acurácia	AutoML-DS	Positiva	Negativa
Precisão	AutoML-DS	Positiva	Negativa
Recall	Met. Referência	Negativa	Positiva
Taxa de Falsos Positivos	AutoML-DS	Positiva	Negativa
Taxa de Falsos Negativos	Met. Referência	Negativa	Positiva
AUC	Met. Referência	Negativa	Negativa
Adaboost			
Acurácia	AutoML-DS	Positiva	Negativa
Precisão	AutoML-DS	Positiva	Negativa
Recall	Met. Referência	Negativa	Negativa
Taxa de Falsos Positivos	AutoML-DS	Positiva	Negativa
Taxa de Falsos Negativos	Met. Referência	Negativa	Negativa
AUC	AutoML-DS	Positiva	Negativa
Árvore de Decisão EF			
Acurácia	AutoML-DS	Positiva	Negativa
Precisão	AutoML-DS	Positiva	Negativa
Recall	AutoML-DS	Negativa	Negativa
Taxa de Falsos Positivos	AutoML-DS	Positiva	Negativa
Taxa de Falsos Negativos	AutoML-DS	Negativa	Negativa
AUC	AutoML-DS	Positiva	Negativa

Tabela 4.5: Comparação de performance e sentido de variação de performance entre os instantes T0 e T9, entre o modelo proposto *AutoML-DS* e a metodologia de referência

4.3.3 Análise das Curvas ROC e Matrizes de Confusão

As curvas ROC são uma medida importante dos modelos que auxiliam os profissionais na utilização prática da solução em ambiente de produção. As figuras 4.1, 4.2 e 4.3 apresentam as curvas obtidas dos modelos treinados para a metodologia de referência o para o *AutoML-DS*.

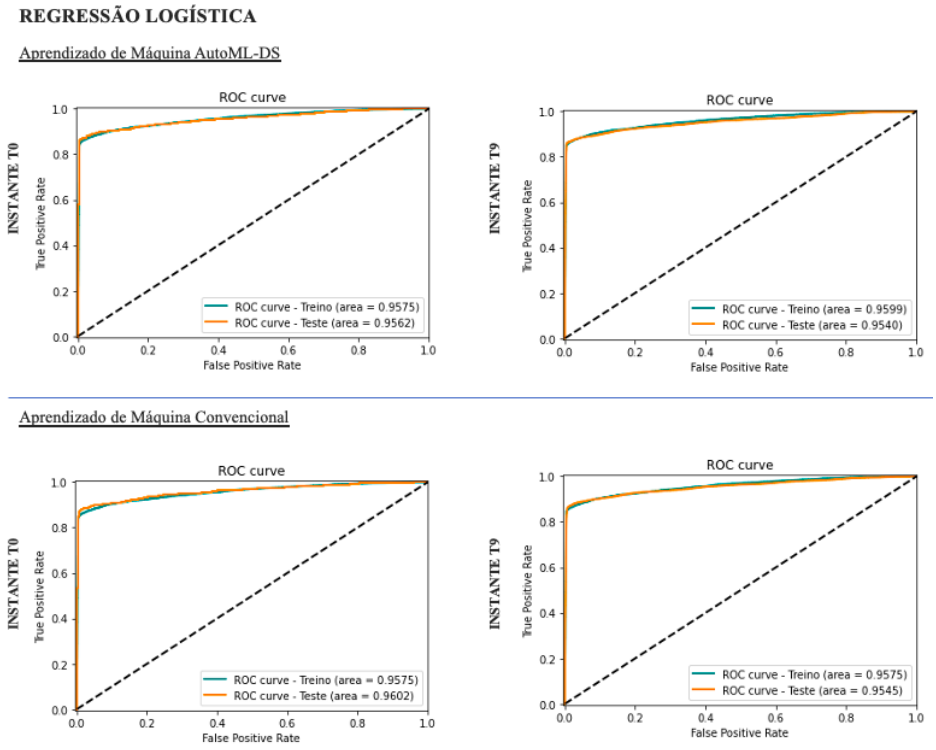

Figura 4.1: Curvas ROC dos modelos regressão logística, carregamentos semanais

.

Em uma avaliação visual das Curvas ROC, nota-se que a incerteza maior está predição da classe positiva de *churn*. Já os casos de Falso Negativo (classe de não-*churn*) são relativamente mais fáceis de acerto. A utilização de AUC leva em consideração o desempenho da classe individual para todos os limites possíveis, comparando a classe prevista de um evento com a classe real desse evento, considerando todos os possíveis valores de corte para a classe prevista.

O AUC médio é demonstrado nas tabelas 4.1 e 4.2, onde nota-se que o modelo convencional com separação linear, regressão logística, possui melhor desempenho com base nesta métrica, no entanto, a variação entre o resultado mínimo e máximo é maior, o que indica maior dispersão no resultado. Já nas tabelas 4.3 e 4.4, observa-se que o modelo *AutoML-DS* tem perda de AUC ao longo dos instantes T0 e T9 menor que o modelo convencional. Já o modelo *AutoML-DS* que utiliza Adaboost possuí um AUC superior ao

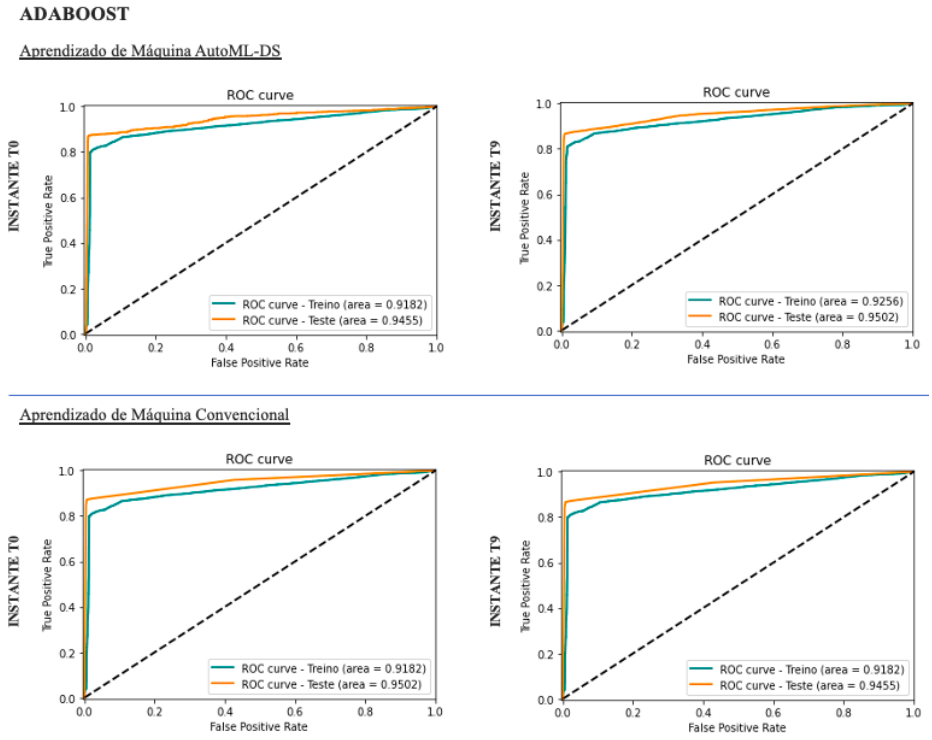

Figura 4.2: Curvas ROC dos modelos Adaboost, carregamentos semanais

modelo de referência, além de variar positivamente; o modelo que tem como classificador o Árvore de Decisão EF tem AUC superior ao modelo convencional, além de possuir uma variação dos valores do AUC entre os instantes T0 e T9 positiva, indicando melhora na performance.

A matriz de confusão é uma fonte relevante para diversas métricas de avaliação, conforme visto na seção 2.2.2. Em problemas de classificação em que há relevância na identificação de falsos positivos e falsos negativos, esses números podem resultar em impacto financeiro diante da atuação de retenção de clientes. Neste ponto, considera-se a utilização do valor padrão de threshold (0.5) para separação das classes para obtenção da matriz de confusão.

A figura 4.4 demonstra como os modelos convencionais que utilizam regressão logística como classificador perde a capacidade de predição ao longo do tempo, fazendo-se necessário um retreino constante para suprir os desvios de conceito. O modelo utilizando a metodologia de referência tem um aumento considerável de falsos positivos entre o instante T0 e o instante T9, passando de 13% para 22%. Por outro lado, o modelo *AutoML-DS* que utiliza regressão logística tem sua capacidade de predição conservada, pois recebe, a cada instante, novos dados que fazem com que os modelos se adaptem às mudanças que acontecem durante o percurso. O modelo *AutoML-DS* apresenta uma leve

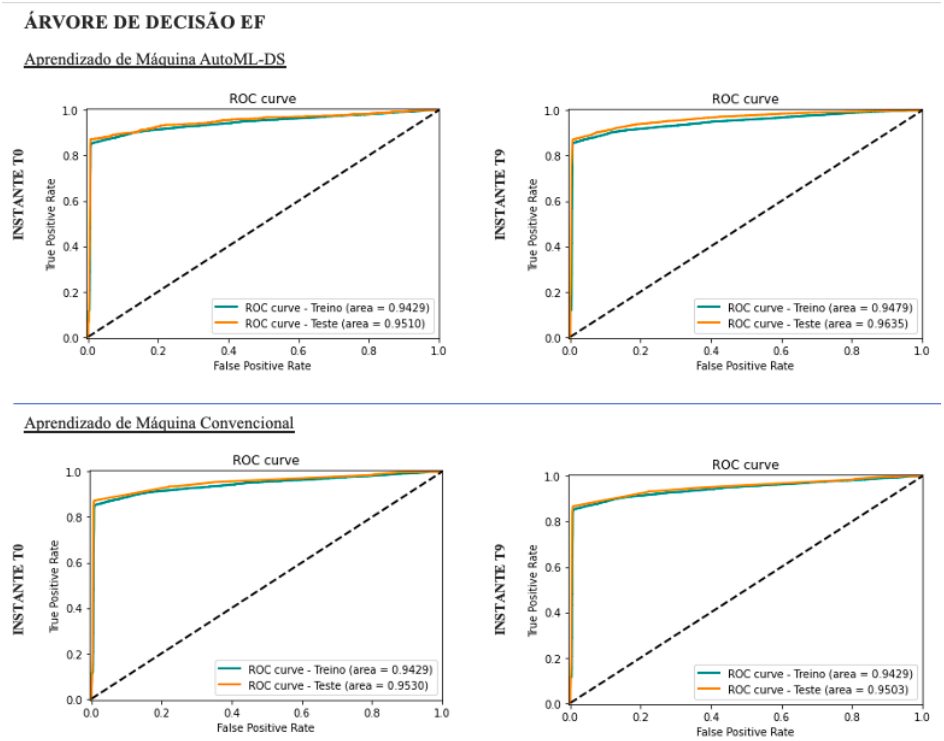

Figura 4.3: Curvas ROC dos modelos Árvore de Decisão EF, carregamentos semanais

.

melhora no volume de falsos positivos, passando de 1,2% no instante T0 para 1,0% no instante T9.

Quando compara-se o volume de falsos positivos em ambos os modelos, nota-se que há garantia na predição de casos classificados como *churn* e que provavelmente não deixaram de ser clientes do serviço em questão, é mais adequada para os modelos *AutoML-DS*.

Vale ressaltar que o volume de predições corretas no modelo que utiliza metodologia proposta por este trabalho também sofre uma pequena alteração, sendo o volume de verdadeiros negativos 74,7% no instante T0 e 76,3% no instante T9. Já o volume de verdadeiros positivos, classe *churn*, tem uma pequena queda, onde o volume de predições corretas nessa classe no instante T0 é de 21,0% e 19,7% no instante T9.

Nota-se que a partir dos resultados da figura 4.4, ambos os modelos que utilizam de regressão logística para classificação são aderentes ao ambiente de teste no momento em que foram treinados no instante T0. No entanto, a aplicação do modelo com a metodologia de referência no ambiente de produção e uso no problema considerado, implica em um dispêndio financeiro com os clientes com falso positivo. O modelo possui volume de falsos positivos aproximadamente 10 vezes superior ao volume de

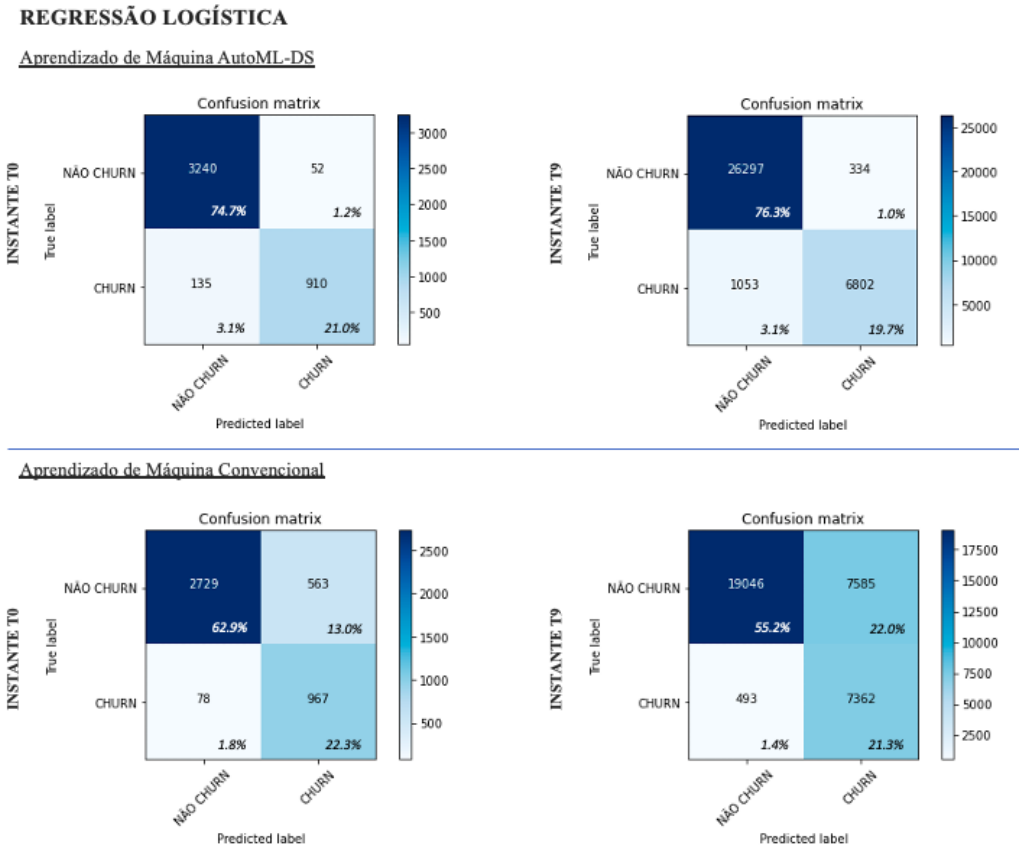

Figura 4.4: Matrizes de Confusão dos modelos regressão logística nas metodologias de referência e no *autoML-DS*.

.

falsos positivos do modelo *AutoML-DS*. Entretanto, o modelo utilizando a metodologia de referência tem uma performance superior ao modelo *AutoML-DS* quando observado o volume de falsos negativos. Este resultado influencia diretamente no volume da carteira de clientes, visto que os modelos *AutoML-DS* (regressão logística) foram menos capazes de identificar os clientes que, de fato, evadiram com mais precisão. No entanto, vale ressaltar que a taxa de falsos positivos e a taxa de falsos negativos não consideram o desempenho individual da classe de um classificador.

As figuras 4.5 e 4.6 apresentam as matrizes de confusão para metodologia de referência e *AutoML-DS*. Nota-se que os modelos da metodologia de referência não se deterioram ao longo dos instantes T0 e T9. O volume de acertos e erros dos modelos possuem pouca variação entre os instantes T0 e T9.

O modelo *AutoML-DS* com treinamento incremental e utilizando Adaboost como classificador apresenta uma melhoria no volume de falsos positivos, reduzindo de 1,5%

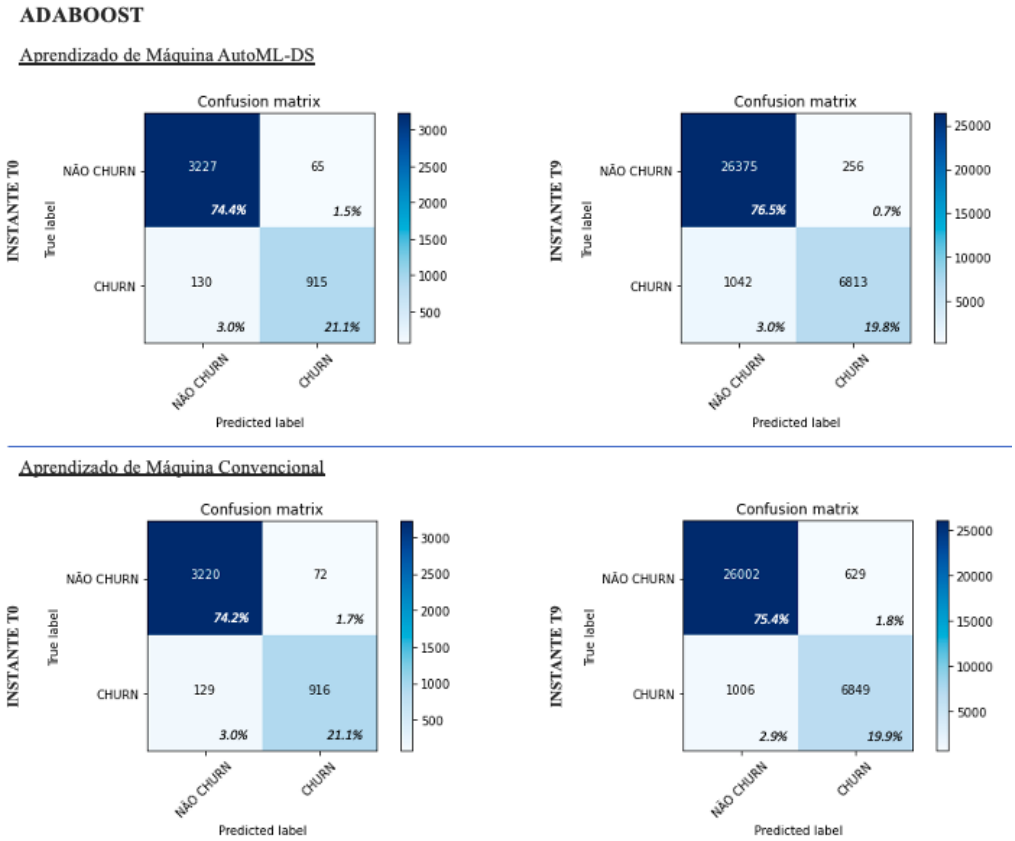

Figura 4.5: Matrizes de Confusão dos modelos Adaboost, carregamentos semanais

.

no instante T0 para 0,7% no instante T9. O que no modelo que utiliza metodologia convencional não ocorre, tendo volume de falsos positivos de 1,7% no instante T0 e 1,8% no instante T9.

No modelo *AutoML-DS* que utiliza árvore de decisão EF o volume de falsos positivos tem uma redução modesta de 0,7% no instante T0 para 0,6% no instante T9. O inverso acontece ao modelo que utiliza a metodologia de referência, onde o volume de falsos positivos sai de 0,7% no instante T0 para 0,8% no instante T9.

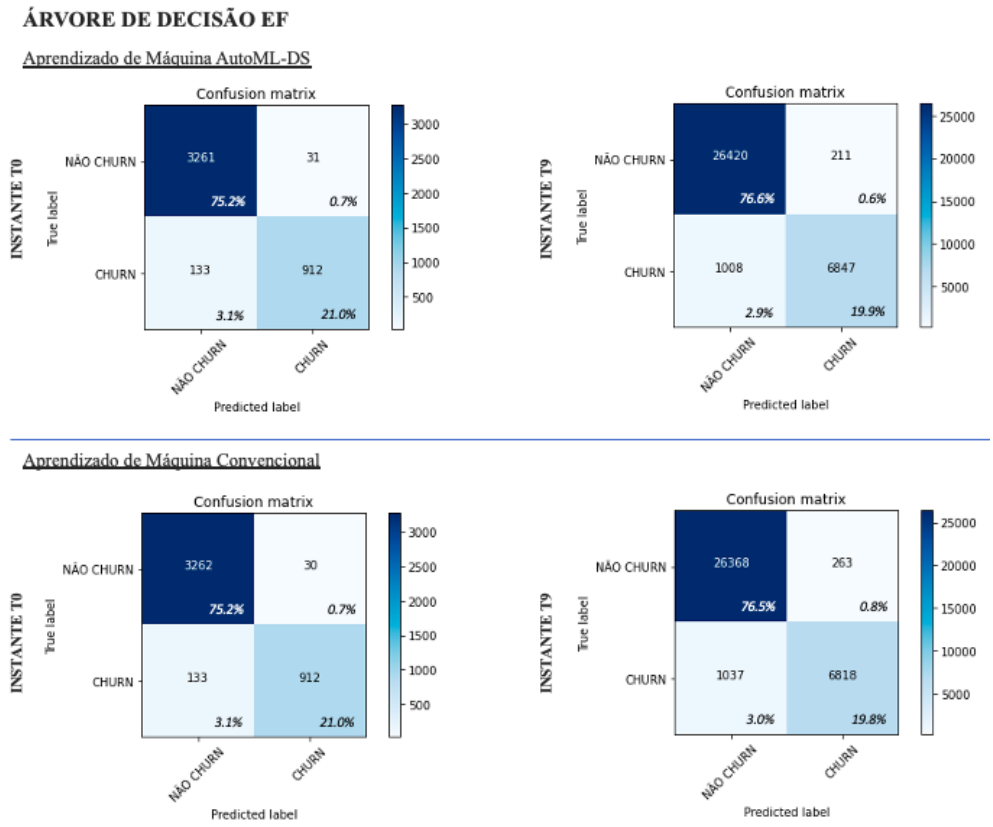

Figura 4.6: Matrizes de Confusão dos modelos Árvore de Decisão EF, carregamentos semanais

.

4.4 Ganhos Econômicos do Emprego da Solução em Ambiente de Produção

O projeto de prevenção de *churn* está implementado e em uso contínuo há aproximadamente um ano e meio, totalizando dezoito meses em ambiente de produção. O modelo preditivo implementado, capaz de identificar os possíveis clientes que deixarão a carteira, ou seja possíveis *churners*, passou a direcionar ações de *marketing* e relacionamento.

Para avaliar a performance das ações de relacionamento foram estabelecidos 2 grupos, onde um grupo aleatório recebe ações e um grupo de controle não é atingido pelas ações da equipe de *marketing* e relacionamento.

Com ações específicas para este grupo de cliente, tenta-se suprir o possível desejo de cancelamento do contrato junto à empresa antes que ele venha a existir. A figura

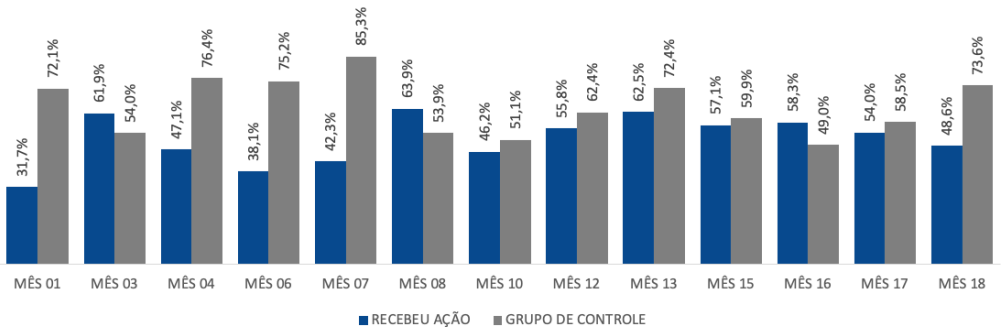

Figura 4.7: *Churn Rate* aferido para o grupo de assinantes que receberam ações direcionadas e para o grupo de controle

.

4.7 apresenta o *churn rate* (taxa média de evasão de clientes) de clientes indicados pelo modelo que receberam ações de relacionamento e para o grupo de controle, assinantes que não receberam intervenção da equipe de *marketing* e relacionamento. Nota-se que apenas em três pontos de avaliação (meses 03, 08 e 16) o *churn rate* do grupo que recebeu ação de relacionamento superou o *churn rate* do grupo de controle. Analisando a figura 4.7 obtém-se um *churn rate* médio para os assinantes atingidos por ações de retenção ativa de aproximadamente 51,3%, o que indica uma performance superior quando observado o grupo de controle, que possuí um *churn rate* médio de 64,9%.

O resultado financeiro obtido pelo projeto implementado é demonstrado na figura 4.8, onde as barras em azul indicam o valor retido dos assinantes que receberam ações de relacionamento e a linha verde é resultado acumulado dos valores retidos, já descontados os custos de implementação e manutenção do projeto, além do custo dispendido com ações de relacionamento. Financeiramente, o projeto apresenta bons resultados. Apurando a receita advinda dos contratos que não cancelaram o contrato de assinatura vigente e que foram atingidos por alguma ação de relacionamento soma-se aproximadamente 779 mil reais, já os custos de implementação e manutenção, incluindo os custos com ações, está em torno de 114 mil reais. Resultados em uma receita líquida de aproximadamente 665 mil reais.

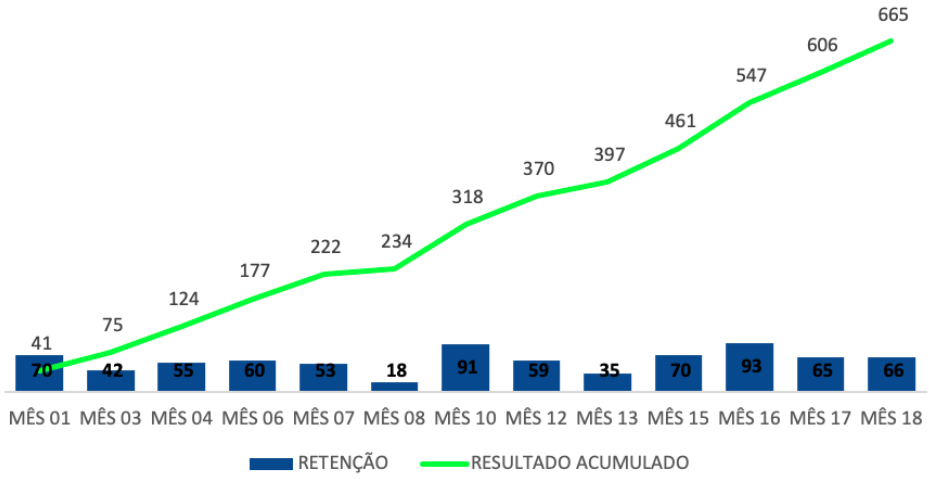

Figura 4.8: Resultado financeiro do projeto

.

Como resultados não financeiros, a adoção de modelos preditivos para a análise e manutenção de carteira de clientes acarretou na melhoria do direcionamento da equipe de relacionamento, consolidação de atividades relacionadas a retenção ativa de clientes, estabelecimento de mapa de risco de cada cliente com quem a empresa possuí contrato de assinatura por meio do *churn score*, dentre outros ganhos intangíveis.

A tabela 4.6 apresenta a simulação de gastos com ações de relacionamento com falsos positivos indicados pelo modelo preditivo que utiliza a metodologia de referência e pelo modelo utilizando dados em *stream*, *AutoML-DS*. A tabela ainda apresenta a variação de dispêndio financeiro com a utilização do modelo proposto por este trabalho comparada ao dispêndio financeiro efetivo da utilização da metodologia de referência.

	REFERÊNCIA	**AUTOML-DS**	**∆**
Regressão Logística			
Custo de ações - Falsos Positivos	R$ 33.083,60	R$1.456,81	-95,6%
Receita evadida - Falsos Negativos	R$ 3.442.07	R$ 7.351,93	113,6%
Dispêndio Financeiro Total	**R$ 36.525,67**	**R$ 8.808,75**	**-75,9%**
Adaboost			
Custo de ações - Falsos Positivos	R$ 2.743,52	R$ 1.116,60	-59,3%
Receita evadida - Falsos Negativos	R$ 7.023,78	R$ 7.275,13	3,6%
Dispêndio Financeiro Total	**R$ 9.767,30**	**R$ 8.391,73**	**-14,1%**
Árvore de Decisão EF			
Custo de ações - Falsos Positivos	R$ 1.147,13	R$ 920,32	-19,8%
Receita evadida - Falsos Negativos	R$ 7.240,22	R$ 7.037,75	-2,8%

Dispêndio Financeiro Total	**R$ 8.387,22**	**R$ 7.958,07**	**-5,1%**

Tabela 4.6: Simulação de utilização de recursos financeiros, considerando falsos positivos e falsos negativos

O custo médio de ações de relacionamento e *marketing* (crm) é de R$48,92 (quarenta e oito reais e noventa e dois centavos) por cliente. Em um cenário onde a equipe de relacionamento é capaz realizar ações com 40,13% dos assinantes indicados (atingimento) e leva-se em consideração que os dados de predição utilizados para o experimento compreende um período de quatro meses e meio (m). Dito isto, o custo de ações estimado (CFP) para os falsos positivos do modelo é calculado por meio da seguinte fórmula: $CFP = crm * a/m * FP$.

Na tabela 4.6 nota-se que os custos de ações com os assinantes classificados como falsos positivos, ou seja assinantes foram classificados como potenciais *churners* e que não deixaram de ser clientes, é inferior nos modelos que utilizam a metodologia proposta por este trabalho (*AutoML-DS*) quando comparada aos custos que seriam decorrentes da utilização dos modelos que adotam a metodologia de referência. Quando observado os modelos que adotam regressão logística, o custo de ações *AutoML-DS* é 95,6% inferior ao modelo da metodologia de referência. O modelo Adaboost com *AutoML-DS* possuí um custo de ações com falsos positivos 59,3% inferior ao custo do modelo da metodologia de referência. Por fim, o modelo *AutoML-DS* que adota árvore de decisão EF como classificador possuí um resultado mais modelos, apresentando um custo 19,1% menor em relação ao modelo que necessita de retreino periódico.

A perda de receita (RE) com os clientes classificados como os falsos negativos também foi estimada para comparar os resultados dos modelos preditivos que utilizam a metodologia de referência e os modelos *AutoML-DS*. Para o cálculo, leva-se em consideração um valor preço médio mensal de assinatura (ass) de R$48,40 (quarenta e oito reais e quarenta centavos); um *churn rate* médio (cr) para o grupo de controle, que não recebem ações de relacionamento, (64,9%); e considera-se que os dados de predição utilizados para o experimento compreende um período de quatro meses e meio (m). Dessa forma, a receita evadida decorrente dos falsos negativos indicados pelos modelos é calculado usando a fórmula: $RE = ass * cr/m * FN$.

Quando observada a receita que é perdida quando o modelo aponta um cliente como falso negativo, ou seja, aquele cliente que abandonou a carteira de assinantes e o modelo não foi capaz de identificá-lo, nota-se que os modelos que adotam a metodologia de referência possuem uma performance superior aos modelos que utilizam de treinamento com dados em *stream* em dois dos três classificadores utilizados. No modelo que adota regressão logística há uma grande diferença, onde o modelo *AutoML-DS* possuí uma evasão de receita 113,6% superior ao modelo que adota a metodologia de referência; o

modelo *AutoML-DS* possuí uma perda de receita 3,6% superior ao modelo da metodologia de referência. Por outro lado, há um ganho modesto de performance quando o classificador é árvore de decisão EF, onde apura-se uma perda de receita 2,8% inferior no modelo *AutoML-DS* quando comparado ao modelo que adota a metodologia de referência.

O dispêndio financeiro é o resultado da soma do custo de ações realizadas com os falsos positivos e da receita evadida com a perda dos falso negativos. Os modelos que adotam *AutoML-DS* se sobressaem com um dispêndio financeiro menor em todos os modelos implementados quando comparados aos modelos da metodologia de referência. No modelo que utiliza regressão logística como classificador, o dispêndio financeiro é 75,9% inferior; no modelo com Adaboost o dispêndio financeiro é 14,1% inferior; mesmo com resultados semelhantes o modelo *AutoML-DS* com classificador árvore de decisão EF é 5,1% superior ao modelo de referência quando observamos o dispêndio financeiro.

CAPÍTULO 5

Conclusões

Este trabalho teve por objetivo o desenvolvimento e implementação de algoritmos de aprendizado de máquina na predição e avaliação de evasão de clientes em ambiente de produção. Considerou-se um problema real de identificação precoce de saída de clientes *churn* de um jornal de grande circulação regional.

Apesar de relatado na literatura de aprendizado de máquina, um dos maiores desafios em ambiente de produção é o desvio de conceito dos modelos após a implantação. Nesse sentido, foi proposta uma metodologia que além de considerar o desenvolvimento, a pesquisa e a implantação do ciclo completo de projeto envolvendo aprendizado de máquina em um problema real, fosse capaz de mitigar o problema de desvio de conceito.

Os resultados obtidos dos modelos preditivos de *churn* mostram que o problema pode ser satisfatoriamente tratado como um problema de predição a partir dos dados disponíveis na empresa considerada. No entanto, observou-se que em diferentes níveis, os modelos treinados sofrem degradação de desempenho ao longo do tempo em um ambiente de produção. A metodologia proposta que incluía o uso de uma atualização de dados de forma incremental foi capaz de adaptar os modelos ao longo do tempo, sem a necessidade da realização de um retreino completo.

Nos modelos *AutoML-DS* obtém-se constância no volume de falsos positivos, tendo ganhos modestos nas predições. O volume de falsos positivos reflete um grupo de clientes que necessita de ações de retenção que consomem recursos financeiros. Como os modelos propostos apresentam melhores resultados quando comparados à metodologia de referência, existe uma economia de recursos que se a metodologia convencional estivesse em produção necessitaria.

O projeto está em uso na empresa considerada no estudo a cerca de um ano e meio e até o momento proporcionou ganhos de aproximadamente R$ 647.000,00 além de ganhos intangíveis tais como melhoria da percepção de marca, satisfação dos clientes e aumento da audiência.

Como limitação, aponta-se que não foi possível desenvolver um método que permita que quaisquer classificadores possam ser utilizados na metodologia *AutoML-DS*. Também não foi implementado uma verificação automática de desvio de conceito.

Esta premissa foi verificada apenas experimentalmente de forma offline. Estas limitações podem ser explorados para trabalhos futuros.

Referências Bibliográficas

[Baena-Garcıa et al. 2006]BAENA-GARCIA, M. et al. Early drift detection method. In: *Fourth international workshop on knowledge discovery from data streams*. [S.l.: s.n.], 2006. v. 6, p. 77–86.

[Benczúr, Kocsis e Pálovics 2018]BENCZÚR, A. A.; KOCSIS, L.; PÁLOVICS, R. Online machine learning in big data streams. *arXiv preprint arXiv:1802.05872*, 2018.

[Bertrand 2019]BERTRAND, H. *Hyper-parameter optimization in deep learning and transfer learning : applications to medical imaging*. Tese (Doutorado) — Université Paris-Saclay, 01 2019.

[Cai et al. 2018]CAI, J. et al. Feature selection in machine learning: A new perspective. *Neurocomputing*, Elsevier, v. 300, p. 70–79, 2018.

[Chai 2020]CHAI, C. P. The importance of data cleaning: Three visualization examples. *CHANCE*, Taylor & Francis, v. 33, n. 1, p. 4–9, 2020.

[Chawla, Japkowicz e Kotcz 2004]CHAWLA, N. V.; JAPKOWICZ, N.; KOTCZ, A. Editorial: Special issue on learning from imbalanced data sets. *SIGKDD Explor. Newsl.*, Association for Computing Machinery, New York, NY, USA, v. 6, n. 1, p. 1–6, jun. 2004. ISSN 1931-0145. Disponível em: <https://doi.org/10.1145/1007730.1007733>.

[Clark 2018]CLARK, A. The machine learning audit—crisp-dm framework. 2018.

[Deshpande, Kamath e Joglekar 2019]DESHPANDE, A.; KAMATH, C.; JOGLEKAR, M. A comparison study of classification methods and effects of sampling on unbalanced data. In: IEEE. *2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)*. [S.l.], 2019. p. 1056–1063.

[Feurer et al. 2015]FEURER, M. et al. Efficient and robust automated machine learning. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2015. p. 2962–2970.

[Feurer et al. 2019]FEURER, M. et al. Auto-sklearn: efficient and robust automated machine learning. In: *Automated Machine Learning*. [S.l.]: Springer, 2019. p. 113–134.

[Gama, Sebastiao e Rodrigues 2009]GAMA, J.; SEBASTIAO, R.; RODRIGUES, P. P. Issues in evaluation of stream learning algorithms. In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.: s.n.], 2009. p. 329–338.

[Gama, Sebastiao e Rodrigues 2013]GAMA, J.; SEBASTIAO, R.; RODRIGUES, P. P. On evaluating stream learning algorithms. *Machine learning*, Springer, v. 90, n. 3, p. 317–346, 2013.

[Gama et al. 2014]GAMA, J. et al. A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, ACM New York, NY, USA, v. 46, n. 4, p. 1–37, 2014.

[Gomes et al. 2019]GOMES, H. M. et al. Machine learning for streaming data: state of the art, challenges, and opportunities. *ACM SIGKDD Explorations Newsletter*, ACM New York, NY, USA, v. 21, n. 2, p. 6–22, 2019.

[Gu 2019]GU, F. *Concept Drift Detection for Machine Learning with Stream Data*. Tese (Doutorado) — Faculty of Engineering and Information Technology, University of Technology Sydney, 2019.

[Hastie, Tibshirani e Friedman 2009]HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The elements of statistical learning: data mining, inference, and prediction*. [S.l.]: Springer Science & Business Media, 2009.

[Hoens, Polikar e Chawla 2012]HOENS, T. R.; POLIKAR, R.; CHAWLA, N. V. Learning from streaming data with concept drift and imbalance: an overview. *Progress in Artificial Intelligence*, Springer, v. 1, n. 1, p. 89–101, 2012.

[Hossin e Sulaiman 2015]HOSSIN, M.; SULAIMAN, M. A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, Academy & Industry Research Collaboration Center (AIRCC), v. 5, n. 2, p. 1, 2015.

[Kotsiantis, Zaharakis e Pintelas]KOTSIANTIS, S. B.; ZAHARAKIS, I. D.; PINTELAS, P. E. Aprendizado de máquina: uma revisão das técnicas de classificação e combinação. *Revisão de Inteligência Artificial*, v. 26, p. 159–190.

[Kotthoff et al. 2017]KOTTHOFF, L. et al. Auto-weka 2.0: Automatic model selection and hyperparameter optimization in weka. *The Journal of Machine Learning Research*, JMLR. org, v. 18, n. 1, p. 826–830, 2017.

[Kubat e Widmer 1994]KUBAT, M.; WIDMER, G. Adapting to drift in continuous domains technical report õfai-tr-94-27. *Vienna: Austrian Research Institute for Artificial Intelligence*, 1994.

[Kumar e Minz 2014]KUMAR, V.; MINZ, S. Feature selection: a literature review. *SmartCR*, v. 4, n. 3, p. 211–229, 2014.

[Lavesson e Davidsson 2008]LAVESSON, N.; DAVIDSSON, P. Generic methods for multi-criteria evaluation. In: SIAM. *Proceedings of the 2008 SIAM International Conference on Data Mining*. [S.l.], 2008. p. 541–546.

[Lorica e Paco 2018]LORICA, B.; PACO, N. *The State of Machine Learning Adoption in the Enterprise*. [S.l.]: O'Reilly Media, 2018.

[Ma, Tan e Shu 2015]MA, S.; TAN, H.; SHU, F. When is the best time to reactivate your inactive customers? *Marketing Letters*, Springer, v. 26, n. 1, p. 81–98, 2015.

[Maciel, Santos e Barros 2015]MACIEL, B. I. F.; SANTOS, S. G. T. C.; BARROS, R. S. M. A lightweight concept drift detection ensemble. In: IEEE. *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*. [S.l.], 2015. p. 1061–1068.

[Mackenzie 2015]MACKENZIE, A. The production of prediction: What does machine learning want? *European Journal of Cultural Studies*, SAGE Publications Sage UK: London, England, v. 18, n. 4-5, p. 429–445, 2015.

[Madrid et al. 2019]MADRID, J. G. et al. Towards automl in the presence of drift: first results. *arXiv preprint arXiv:1907.10772*, 2019.

[Makaba e Dogo 2019]MAKABA, T.; DOGO, E. A comparison of strategies for missing values in data on machine learning classification algorithms. In: IEEE. *2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*. [S.l.], 2019. p. 1–7.

[Mateen et al. 2020]MATEEN, M. et al. Automatic detection of diabetic retinopathy: A review on datasets, methods and evaluation metrics. *IEEE Access*, IEEE, v. 8, p. 48784–48811, 2020.

[Mitchell, Keller e Kedar-Cabelli 1986]MITCHELL, T. M.; KELLER, R. M.; KEDAR-CABELLI, S. T. Explanation-based generalization: A unifying view. *Machine learning*, Springer, v. 1, n. 1, p. 47–80, 1986.

[Montiel et al. 2018]MONTIEL, J. et al. Scikit-multiflow: A multi-output streaming framework. *The Journal of Machine Learning Research*, JMLR. org, v. 19, n. 1, p. 2915–2914, 2018.

[Nguyen 2011]NGUYEN, E. H. X. *Customer churn prediction for the Icelandic mobile telephony market*. Tese (Doutorado) — Faculty of Industrial Engineering, Mechanical Engineering and Computer Science University of Iceland, 2011.

[Polyzotis et al. 2018]POLYZOTIS, N. et al. Data lifecycle challenges in production machine learning: a survey. *ACM SIGMOD Record*, ACM New York, NY, USA, v. 47, n. 2, p. 17–28, 2018.

[Provost 2000]PROVOST, F. Machine learning from imbalanced data sets 101. In: AAAI PRESS. *Proceedings of the AAAI'2000 workshop on imbalanced data sets*. [S.l.], 2000. v. 68, p. 1–3.

[Schmidt et al. 2019]SCHMIDT, J. et al. Recent advances and applications of machine learning in solid-state materials science. *npj Computational Materials*, Nature Publishing Group, v. 5, n. 1, p. 1–36, 2019.

[Sobhani e Beigy 2011]SOBHANI, P.; BEIGY, H. New drift detection method for data streams. In: SPRINGER. *International conference on adaptive and intelligent systems*. [S.l.], 2011. p. 88–97.

[Song, Ristenpart e Shmatikov 2017]SONG, C.; RISTENPART, T.; SHMATIKOV, V. Machine learning models that remember too much. In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. [S.l.: s.n.], 2017. p. 587–601.

[Stripling et al. 2015]STRIPLING, E. et al. Profit maximizing logistic regression modeling for customer churn prediction. In: IEEE. *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. [S.l.], 2015. p. 1–10.

[Tsymbal 2004]TSYMBAL, A. The problem of concept drift: definitions and related work. *Computer Science Department, Trinity College Dublin*, v. 106, n. 2, p. 58, 2004.

[Webb et al. 2015]WEBB, G. et al. Characterizing concept drift. *Data Mining and Knowledge Discovery*, v. 30, 11 2015.

[Wirth e Hipp 2000]WIRTH, R.; HIPP, J. Crisp-dm: Towards a standard process model for data mining. In: SPRINGER-VERLAG LONDON, UK. *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*. [S.l.], 2000. p. 29–39.

[Witten, Frank e Hall 2005]WITTEN, I. H.; FRANK, E.; HALL, M. A. Practical machine learning tools and techniques. *Morgan Kaufmann*, p. 578, 2005.

[Wu et al. 2011]WU, L. L. et al. Evaluating machine learning for improving power grid reliability. 2011.

[Xu e Goodacre 2018]XU, Y.; GOODACRE, R. On splitting training and validation set: A comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning. *Journal of Analysis and Testing*, Springer, v. 2, n. 3, p. 249–262, 2018.

[Zheng e Casari 2018]ZHENG, A.; CASARI, A. *Feature engineering for machine learning: principles and techniques for data scientists*. [S.l.]: "O'Reilly Media, Inc.", 2018.

[Zhu, Baesens e Broucke 2017]ZHU, B.; BAESENS, B.; BROUCKE, S. K. vanden. An empirical comparison of techniques for the class imbalance problem in churn prediction. *Information sciences*, Elsevier, v. 408, p. 84–99, 2017.

[Žliobaitė 2010]ŽLIOBAITĖ, I. Learning under concept drift: an overview. *arXiv preprint arXiv:1010.4784*, 2010.

[Žliobaitė, Pechenizkiy e Gama 2016]ŽLIOBAITĖ, I.; PECHENIZKIY, M.; GAMA, J. An overview of concept drift applications. In: *Big data analysis: new algorithms for a new society*. [S.l.]: Springer, 2016. p. 91–114.