MANUTENÇÃO AUTOMÁTICA DE UM SISTEMA DE AUDITORIA INTELIGENTE EM UMA SEGURADORA DE SAÚDE

LUCAS DE MIRANDA BASTOS

MANUTENÇÃO AUTOMÁTICA DE UM SISTEMA DE AUDITORIA INTELIGENTE EM UMA SEGURADORA DE SAÚDE

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Orientador: Nivio Ziviani
Coorientador: Adriano Alonso Veloso

Belo Horizonte
Abril de 2020

Bastos, Lucas de Miranda

B327m Manutenção automática de um sistema de auditoria inteligente em uma seguradora de saúde [manuscrito] / Lucas de Miranda Bastos. — Belo Horizonte, 2020.
xii, 39 f. : il. ; 29cm

Orientador: Nivio Ziviani
Coorientador: Adriano Alonso Veloso

Dissertação (mestrado) - Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação.
Referências: f. 37-39

1. Computação – Teses. 2. Aprendizado do computador - Teses. 3. Algoritmos de computador – Teses. 4. Teoria do controle – Teses. 5. Seguro-saúde – Auditoria - Teses. I. Ziviani, Nivio. II. Veloso, Adriano Alonso. III. Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação. IV. Título.

CDU 519.6*82(043)

Ficha catalográfica elaborada pela bibliotecária Irénquer Vismeg Lucas Cruz - CRB 6ª Região nº 819.

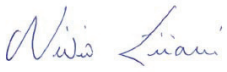
UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Manutenção Automática de um Sistema de Auditoria Inteligente em uma Seguradora de Saúde

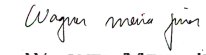
LUCAS DE MIRANDA BASTOS

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

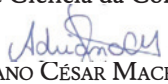


PROF. NIVIO ZIVIANI - Orientador
Departamento de Ciência da Computação - UFMG

PROF. ADRIANO ALONSO VELOSO
Departamento de Ciência da Computação - UFMG



PROF. WAGNER MEIRA JÚNIOR
Departamento de Ciência da Computação - UFMG



PROF. ADRIANO CÉSAR MACHADO PEREIRA
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 1 de Abril de 2020.

Dedico este trabalho à minha mãe, Giselle. Você é e sempre será a minha maior inspiração!

Agradecimentos

Primeiro, agradeço ao meu avô Paulo, minha avó Maria Goretti (*in memorian*) e minha mãe Giselle por todos os ensinamentos e pelo investimento que fizeram na minha educação, esse trabalho também é uma conquista de todos vocês.

À minha namorada Lorena e aos meus amigos e familiares por fazerem a vida valer a pena e por terem me escutado falar tanto deste trabalho.

Por fim, agradeço aos meus orientadores Nivio e Adriano e também as empresas Kunumi e Unimed BH por permitirem a execução desse projeto e fornecerem o conhecimento e infraestrutura necessária para sua conclusão.

"Que não te falte forças pra lutar e motivos pra sorrir
Fé pra acreditar e amor pra dividir"
(Fábio Brazza - Heróis Invisíveis)

Resumo

A utilização do aprendizado de máquina para resolução de problemas reais proporciona aumento da velocidade na análise de dados e sobrepõe o viés humano com maior assertividade em atividades, tais como predição, classificação e otimização. O modelo gerado por um algoritmo de aprendizado de máquina necessita de ajustes periódicos para manter a efetividade, mesmo que novos dados de entrada mudem com o tempo. A principal motivação para o desenvolvimento deste trabalho é o fato de que os modelos de aprendizado de máquina tornam-se defasados com o passar do tempo. O objetivo deste trabalho é automatizar o processamento de solicitações na Unimed-BH que não podem ser auditadas automaticamente e necessitam de intervenção humana. Isso acontece quando: (i) não existem regras de negócio claras para essas situações ou (ii) a requisição é complexa ou cara. Atualmente esses tipos de solicitações são direcionadas para um grupo de auditores especialistas que fazem análise manual das informações apresentadas. A empresa possui cerca de 1,2 milhões de clientes e audita cerca de 500 mil solicitações de exames e procedimentos médicos por mês. A principal contribuição deste trabalho inclui a criação de um mecanismo de manutenção do sistema de auditoria eletrônica da Unimed-BH capaz de monitorar uma ou mais métricas de riscos e calibrar parâmetros do modelo automaticamente. A aplicação do mecanismo de manutenção monitora dinamicamente a efetividade da análise de solicitações originalmente enviadas para auditoria humana, objetivando manter parâmetros mínimos de efetividade que minimizem os impactos da automação e atuem de acordo com o cenário atual. Caso o modelo aprove uma solicitação que deveria ser negada é gerado um prejuízo financeiro que envolve uma métrica de risco que regula a taxa de automação. A relação entre a taxa de automação e o risco financeiro utiliza a Teoria dos Controladores Proporcionais Integrados Derivativos para que o sistema proposto seja mais permissivo nos momentos de pouco prejuízo financeiro e permita aumentar a taxa de automação.

Palavras-Chaves: Aprendizado de Máquina, Teoria de Controle, Sistema em Produção, Seguradora de Saúde

Abstract

The use of machine learning to solve real problems provides an increase in the speed of data analysis and overlaps the human bias with greater assertiveness in activities such as prediction, classification and optimization. The model generated by a machine learning algorithm requires periodic adjustments to maintain effectiveness, even if new input data changes over time. The main motivation for the development of this work is the fact that the machine learning models become outdated over time. The objective of this work is to automate the processing of requests at Unimed-BH that cannot be audited automatically and require human intervention. This happens when: (i) there are no clear business rules for these situations or (ii) the request is complex or expensive. Currently, these types of requests are directed to a group of expert auditors who perform manual analysis of the information presented. The company has about 1.2 million customers and audits around 500,000 requests for medical exams and procedures per month. The main contribution of this work includes the creation of a maintenance mechanism for the Unimed-BH electronic audit system capable of monitoring one or more risk metrics and calibrating model parameters automatically. The application of the maintenance mechanism dynamically monitors in real time the effectiveness of the analysis of requests originally sent for human audit, aiming to maintain minimum effectiveness parameters that minimize the impacts of automation and act according to the current scenario. If the model approves a request that should be denied, a financial loss is generated that involves a risk metric that regulates the automation rate. The relationship between the automation rate and financial risk uses the Theory of Derivative Integrated Proportional Controllers so that the proposed system is more permissive in times of little financial loss and allows to increase the automation rate. Similarly, it is desirable that the system becomes more prohibitive in cases where the performance of the model deteriorates, which allows to dynamically control the performance of the model.

Keywords: Machine Learning, Control Theory, System in Production, Health Insurance

Lista de Figuras

Lista de Tabelas

Sumário

Capítulo 1

Introdução

A evolução da Web é marcada pela alta capacidade de geração de dados, onde os usuários consomem conteúdo de grandes portais e também produzem uma quantidade massiva de informação. Concomitantemente, empresas transferem parte das suas operações para o mundo digital e levam seus consumidores a uma nova era de praticidade e conteúdo personalizado, onde não é preciso mais sair de casa para assistir filmes ou fazer compras.

Essa migração dos processos para o mundo *online*, aliada ao barateamento e melhoria das tecnologias de armazenamento e processamento de dados, faz com que as empresas tenham cada vez mais interesse em armazenar o máximo de informação sobre o usuário. Por exemplo, uma loja virtual é capaz de contabilizar o tempo gasto pelo usuário na visualização de um determinado produto. Essa informação pode ser utilizada estrategicamente para recomendar ou não aquele produto para o cliente, situação que não é possível em uma grande loja física.

Todavia, o aumento de informação armazenada dificultou o trabalho dos analistas de dados tradicionais. Isso aconteceu porque a computação clássica não funciona bem para o processamento de dados de alta dimensionalidade, pois a mesma exige que os analistas saibam explicitamente como programar as rotinas de processamento e análise dos dados. Esse tipo de programação exige que cada caso seja tratado individualmente pela criação de regras manuais.

Dado esse cenário não escalável, abre-se espaço para a utilização de novos tipos de algoritmos,[1] que ao invés de submeterem os analistas à criação de regras e associações manuais, trazem o foco para o desenvolvimento de modelos matemáticos capazes de inferir essas regras automaticamente em função da distribuição histórica dos dados.

[1] https://medium.com/@karpathy/software-2-0-a64152b37c35

Esse campo de estudo é denominado Aprendizado de Máquina, que é uma subárea da Inteligência Artificial (IA).

Posto isso, nota-se a implantação de soluções utilizando aprendizado de máquina por muitas empresas nos últimos anos [McKendrick, 2019]. Os ganhos vão desde o aumento na velocidade com que se consegue analisar os dados até a obtenção de *insights* que sobrepõe o viés humano, fazendo com que os analistas também aprendam com a máquina. Além disso, como disponível em NITRD [2019], observa-se um investimento bilionário dos Estados Unidos nas pesquisas em IA nos últimos anos.

1.1 Motivação

O ciclo de vida de um modelo de aprendizado de máquina normalmente consiste em ajustar um modelo matemático em função de uma base de dados utilizando algum algoritmo de aprendizado visando aprimorar uma função objetivo. A saída desse processo gera um modelo matemático capaz de avaliar novas instâncias de dados e predizer uma resposta com base naquilo que foi observado na etapa de ajuste.

Entretanto, um modelo de aprendizado de máquina só se torna útil após a sua implantação dentro de um sistema com a finalidade de realizar a tarefa para o qual foi ajustado. Historicamente, a maior parte da pesquisa na área de aprendizado de máquina é focada na etapa de ajuste dos modelos, tendo como objetivo criação de novos algoritmos e técnicas de processamento que produzam resultados de melhor qualidade do que aqueles já criados.

A principal motivação para o desenvolvimento deste trabalho é o fato de que os modelos de inteligência artificial tornam-se defasados com o passar do tempo. Fato é que assim como os softwares de computação clássica precisam de manutenção periódica, os novos modelos de aprendizado também carecem desse mesmo cuidado, visto que muitos problemas de efetividade surgem apenas após a implantação do sistema [Shimodaira, 2000] [Snoek et al., 2019] [Lakshminarayanan et al., 2017] [Brier, 1950].

Quando um segurado deseja utilizar algum serviço médico, como consultas ou exames, ele faz uma solicitação para que a seguradora cubra os custos desses serviços. Na maioria dos casos, essas requisições são auditadas automaticamente por um sistema que utiliza computação clássica. Entretanto, existem casos onde as requisições não podem ser auditadas automaticamente e necessitam de intervenção humana. Isso acontece quando:

- Não existem regras de negócios claras para essas situações;
- A requisição é complexa ou cara.

Solicitações que se encaixem em pelo menos uma das situações supracitadas são direcionadas para um grupo de auditores humanos que fazem análises manuais mais profundas das informações apresentadas.

1.2 Objetivos e Contribuições do Trabalho

O principal objetivo deste trabalho é criação de um sistema de manutenção automática de modelos de inteligência artificial de forma a manter patamares mínimos de efetividade. Para isso, pressupõe-se que o desempenho do modelo está associado a alguma métrica de risco que possa ser medida dinamicamente, uma vez que a mesma será utilizada como referência para controlar a atuação do modelo de aprendizado de máquina.

As principais contribuições deste trabalho são:

- Criação de um mecanismo de manutenção do sistema de auditoria eletrônica capaz de monitorar uma ou mais métricas de riscos e calibrar parâmetros do modelo dinamicamente.

- Aplicação do mecanismo de manutenção para monitorar a efetividade da análise de solicitações originalmente enviadas para auditoria humana, objetivando manter parâmetros mínimos de efetividade que minimizem os impactos da automação e atuem de acordo com o cenário atual.

- O sistema proposto utiliza conceitos da Teoria dos Controladores Proporcionais Integrativos Derivativos (PID) [Ziegler et al., 1942] para calibrar automaticamente alguns parâmetros do modelo. Até onde sabemos, esse é o primeiro trabalho que utiliza a teoria do controlador PID com a finalidade de calibrar modelos de aprendizado de máquina.

- Implantação e experimentação do sistema proposto em um ambiente de produção real.

Cabe ressaltar que o sistema completo de auditoria processa as solicitações antes que as mesmas sejam encaminhadas para auditoria humana. O modelo audita cada item (i.e., cirurgia, exame, consulta) individualmente. Se e somente se todos os itens são aprovados, a solicitação é aprovada. Do contrário, a solicitação é enviada para a auditoria humana, para que seja submetido ao processo de auditoria padrão. Assim, é importante notar que esse novo sistema utilizando modelo de aprendizado de má-

quina nunca irá cometer injustiça com um segurado, pois não será capaz de negar uma solicitação diretamente.

Quando o modelo de aprendizado de máquina aprova uma solicitação que deveria ser negada, é gerado um prejuízo financeiro para a empresa, pois a mesma deve pagar por uma solicitação que não deveria ser aprovada, criando uma relação entre a automação e a perda financeira (métrica de risco).

Para aumentar a taxa de automação, a empresa pode assumir o risco de um possível prejuízo financeiro. Essa abordagem é interessante porque no mundo ideal, os auditores deveriam gastar seus esforços em solicitações mais complexas, que envolvem fazer visitas aos pacientes nos hospitais, ligar para outras empresas e até mesmo fazer revisões em literatura científica ou nas leis que regem os processos. Portanto, aumentar a taxa de automação faz com que os auditores tenham mais tempo para se dedicar ao trabalho mais difícil e humano, gerando um retorno financeiro indireto.

Graças à relação entre a taxa de automação e o risco financeiro, é possível aplicar os conceitos do controlador PID no sistema para que ele seja mais permissivo nos momentos de pouco prejuízo financeiro, fazendo com que a taxa de automação aumente e assumindo, assim, o risco de ter mais solicitações aprovadas erroneamente. Analogamente, é desejável que o sistema se torne mais proibitivo nos casos onde a performance do modelo está ruim. Assim seremos capazes de controlar o desempenho do modelo dinamicamente.

1.3 Estudo de Caso

Este trabalho foi realizado em parceria com a Unimed-BH, uma grande empresa de seguros de saúde no Brasil, que possui cerca de 1,2 milhões de clientes e audita cerca de 500 mil solicitações por mês. À vista disso, este trabalho automatiza uma parte importante do processamento automatizado de solicitações de segurados da empresa.

Existe na Unimed-BH um sistema de auditoria eletrônica que utiliza modelo de aprendizado de máquina para automatizar a análise de solicitações. O modelo foi criado a partir de um conjunto de solicitações prévias, que foram auditadas por médicos especialistas de modo que o sistema desenvolvido possa generalizar as análises de auditoria para as próximas solicitações reais. Como o volume dessas requisições é grande, cerca de 60 mil por mês, existem problemas relacionados ao tempo de resposta à solicitações e custo devido a erros de análises e viés humano.

1.4 Organização

O restante dessa dissertação está organizado da seguinte forma: O Capítulo 2 contém uma revisão sobre os trabalhos relacionados ao tema. O Capítulo 3 descreve a metodologia utilizada, bem como a abordagem utilizada para calibrar o modelo e a forma como a teoria de controle foi aplicada na resolução do problema. O Capítulo 4 contém a descrição dos experimentos e os resultados obtidos. Por fim, o Capítulo 5 conclui o trabalho e aponta oportunidades de trabalhos futuros.

Capítulo 2

Conceitos Básicos e Trabalhos Relacionados

2.1 Controlador Proporcional Integrativo Derivativo (PID)

O Controlador Proporcional Integrativo Derivativo (PID) é um mecanismo de controle em sistemas fechados capaz de medir uma ou mais variáveis de interesse e comparar seu valor atual com um valor de referência desejado [Doyle et al., 2013]. A diferença entre o valor medido e o valor de referência é chamada de erro de referência, através dele, um atuador é capaz de alterar os sinais enviados para o sistema controlado e aplicar as correções derivadas do erro de referência.

A Figura 2.1 ilustra o diagrama da arquitetura de um controlador PID genérico. No diagrama, observa-se que o sistema a ser controlado possui como entrada um sinal atuante e tem como saída uma variável controlada. Esses sinais são emitidos e medidos, respectivamente pelo controlador.

Para automatizar o controle, a variável controlada é enviada pelo caminho de *feedback* para o controlador e comparada com a variável de referência para produzir o termo do erro. Através da análise do erro, o controlador modifica o sinal atuante de forma a tentar aproximá-lo do valor esperado de saída dando início a um novo ciclo deste processo.

O controlador possui três estratégias diferentes para calcular o sinal atuante, a saber:

- A estratégia proporcional utiliza apenas o valor corrente do erro para calcular o sinal atuante.

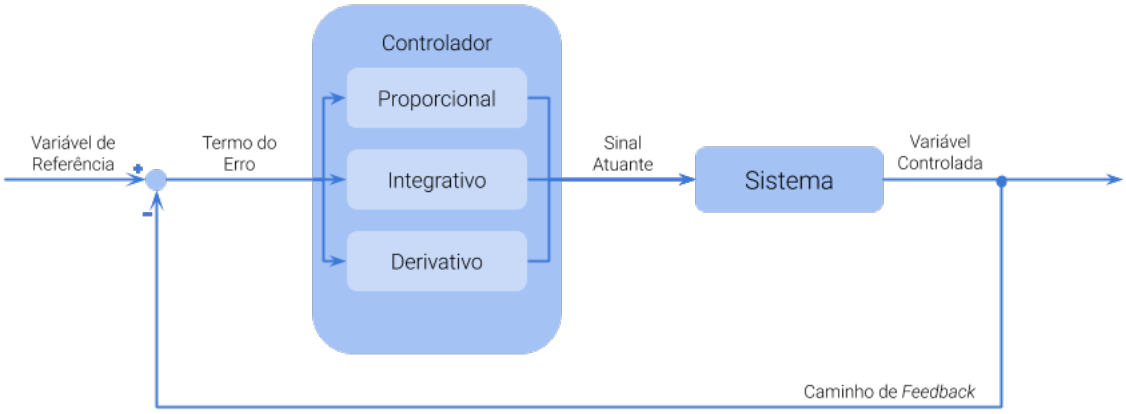


Figura 2.1. Diagrama de um controlador PID genérico. Diagrama adaptado do vídeo em `https://youtu.be/wkfEZmsQqiA`

- A estratégia integrativa, consolida erros passados mantendo um histórico agregado do erro.
- A estratégia derivativa analisa a tendência do erro para inferir a dinâmica futura do erro.

Sejam $u(t)$ e $e(t)$, o sinal gerado como saída pelo controlador e o valor do erro em um tempo t, respectivamente e k_P, k_I e k_D, os pesos dados a cada uma das estratégias, a equação 2.1 sumariza um sistema PID genérico, conforme apresentado no diagrama.

$$u(t) = k_P\, e(t) + k_I \int_0^t e(t)\mathrm{d}t + k_D \frac{\mathrm{d}e(t)}{\mathrm{d}t} \tag{2.1}$$

Ao zerar valores de k_P, k_I ou k_D, removemos algumas das estratégias , como controladores P ou controladores PI. O controlador P utiliza apenas a informação corrente para gerar o sinal atuante, em razão disso, os sistemas podem ficar mais instáveis em alguns casos por não considerarem a evolução ou o passado do sistema. Porém, essa estratégia pode ser interessante em casos onde o valor da variável controlada pode ser relaxado em relação à variável de referência.

Normalmente as aplicações da Teoria de Controle são voltadas para problemas de controle de equipamentos em ambientes industriais. O controlador PID em específico é muito utilizado em problemas que envolvem controle de temperatura, como câmaras de resfriamento ou fornalhas industriais [Grassi & Tsakalis, 2000] [Dequan et al., 2012]. Posto isso, até onde sabemos este é o primeiro trabalho que utiliza a teoria do controlador PID para realizar a manutenção de modelos de aprendizado de máquina em ambientes de produção. Na literatura é possível encontrar alguns trabalhos que seguem

a lógica inversa, onde utilizam algoritmos de aprendizado de máquina para substituir controladores PID [Cao et al., 2005] [Jun & Safonov, 1999] [Cheon et al., 2015].

Em aprendizado de máquina, na área de aprendizado por reforço, do inglês *reinforcement learning* [Sutton et al., 1998], existe uma vertente que é focada em controlar os agentes através do cálculo de políticas ótimas, como o *Q-learning* [Watkins & Dayan, 1992]. Nessas abordagens, os agentes otimizam suas sequências de ações através da avaliação contínua da função de recompensa, porém as políticas normalmente são baseadas em estados discretos e finitos, o que não se adequaria ao problema resolvido por esse trabalho, uma vez que a variável metrificada é contínua e teoricamente pode assumir qualquer valor real.

2.2 Interação do Humano com a Máquina em IA

Alguns trabalhos exploram a interação do ser humano com modelos de aprendizado de máquina de maneira *online*, isto é, enquanto o modelo está em fase operacional. A área do Aprendizado Ativo, do inglês *Active Learning*, estuda a capacidade de um modelo de aprendizado de máquina decidir dentro de um conjunto de opções, quais são aquelas instâncias que mais irão trazer resultados para a qualidade do modelo, ao serem enviadas para o conjunto de treino [Cohn et al., 1994].

Dentre as várias abordagens de aprendizado, pode-se listar o aprendizado semissupervisionado, do inglês *Semi-supervised Learning*, que se aplica em cenários onde uma parte expressiva dos dados do conjunto de treino não se encontram rotulados e deseja-se incrementar a qualidade do modelo rotulando àqueles que mais irão trazer o ganho desejado [Zhu & Goldberg, 2009]. As semelhanças do aprendizado semissupervisionado e aprendizado ativo com o sistema proposto neste trabalho são encontradas na etapa de produção, uma vez que as novas instâncias enviadas são instâncias reais e ainda não foram rotuladas por um especialista. Dessa forma, faz-se necessário a criação de um processo de amostragem contínua desses dados que serão enviados para a análise e rotulação manual dos especialistas da seguradora. As implicações dessa abordagem serão trabalhadas na seção 2.3 e no capítulo 3.

Por fim, o trabalho de Holzinger [2016] utiliza o termo *iML (interactive Machine learning)* para se referir às soluções que necessitam da avaliação humana no "pós-treino". O trabalho também foca no contexto médico e tem como uma de suas motivações o fato de que normalmente os conjuntos de dados médicos normalmente são incompletos e possuem muitos dados dúbios ou corrompidos, fazendo com que seja produzido um modelo de aprendizado de máquina de qualidade ruim. Dentre os exemplos

citados pelo trabalho, está a análise de *clusters*, que é um processo que naturalmente depende expertise humana; dobramento de proteína que é um processo de estruturação de proteínas de modo que fiquem funcionais; anonimização de dados de pacientes e por fim; laudo provisório, que é de extrema importância em muitos casos de urgência e emergência onde o tempo de atendimento é crucial para o sucesso do atendimento.

2.3 Mitigando Riscos de implantação de sistemas de IA em Ambientes de Produção

Quando um sistema que utiliza modelos de aprendizado de máquina é enviado para um ambiente de produção existem algumas variáveis que devem ser levadas em consideração que normalmente não são analisadas quando são feitos apenas experimentos. Essas variáveis aparecem tanto no contexto de infraestrutura (como memória, quantidade de núcleos e velocidade do processador) quanto em contextos intrínsecos a esses sistemas (como degradação de eficiência ao longo do tempo por mudanças na distribuição das características dos novos indivíduos e tolerância a dados incompletos ou novos).

Na literatura, o fenômeno de degradação do modelo em função da passagem do tempo é conhecido como *Covariate Shift*[Shimodaira, 2000], esse campo de estudo vem ganhando importância nos últimos anos graças ao aumento das aplicações de IA nas indústrias. O trabalho de Snoek et al. [2019] compara diferentes métodos de calibragem de incerteza de modelos de aprendizado de máquina através de experimentos empíricos. Os resultados obtidos mostram que para manter a estabilidade dos conjuntos de dados com características categóricas ao longo do tempo a melhor saída é fazer um *ensemble* de vários modelos treinados independentemente e com inicializações aleatórias diferentes, conforme apresentado em Lakshminarayanan et al. [2017]. A validação é feita utilizando a Área sob a curva ROC (AUROC) e o *Brier Score* [Brier, 1950].

2.4 Analogia com o Funcionamento de um Ar Condicionado

Como visto, as teorias do controlador PID são muito aplicadas em sistemas de aquecimento e resfriamento. Afim de tornar mais claro o entendimento da aplicação da teoria no contexto deste trabalho, esta seção utilizará as Figuras 2.2 e 2.3 para fazer uma analogia do funcionamento de um sistema de ar condicionado com o sistema desse

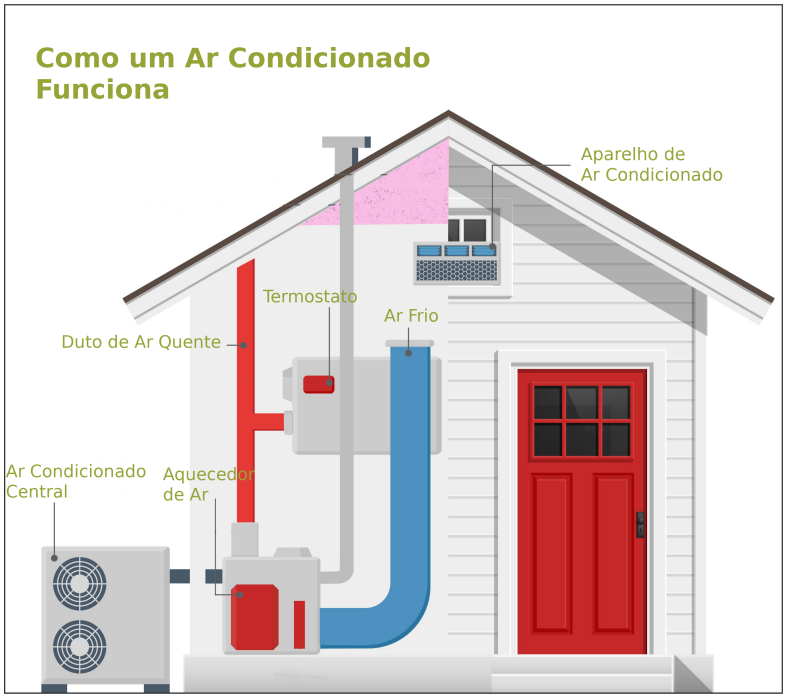


Figura 2.2. Esquema de funcionamento de um ar condicionado. Figura retirada do site da empresa Arronco Comfort Air e modificada para ilustrar o trabalho.

trabalho. A figura original foi retirada do sistema web da empresa Arronco Comfort Air[1] e foi simplificada e traduzida para a utilização nesse trabalho.

A Figura 2.2 ilustra o funcionamento de um sistema de ar condicionado: O ar condicionado central é responsável por retirar o ar externo e enviá-lo para dentro do sistema de ar condicionado. Uma vez dentro do sistema, o ar passa pelo aquecedor, fazendo com que a troca de temperatura gere também o ar frio. Por fim, o ar frio é enviado para o aparelho de ar condicionado e distribuído no interior da casa.

Para controlar a temperatura, o aparelho de ar condicionado possui um termômetro que mede constantemente a temperatura do ambiente e a compara com o valor desejado pelos usuários. Caso a temperatura esteja acima do desejado, então o sistema injeta um ar mais frio do que o desejado, para que o calor se distribua no ambiente até que a temperatura se estabilize no valor desejado.

Neste exemplo, a teoria do controlador PID atua na comparação entre a temperatura desejada e real para então calibrar o quão frio deve ser o ar quer será injetado no ambiente.

Já a Figura 2.3 utiliza o exemplo anterior para fazer uma analogia com a atuação do sistema PID. Na figura, o sistema de manutenção automática utiliza um conjunto

[1] `https://arronco.com/cincinnati`

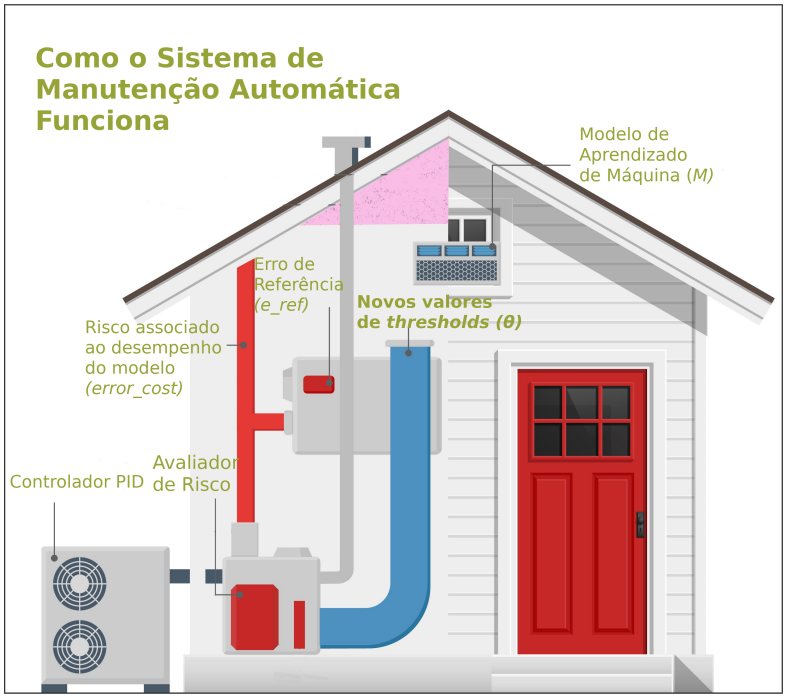


Figura 2.3. Analogia da utilização do controlador PID no contexto do trabalho com o funcionamento de um ar condicionado. Figura retirada do site da empresa Arronco Comfort Air e modificada para ilustrar o trabalho.

de solicitações auditadas pelo modelo e avaliadas por especialistas para calcular o desempenho do modelo. O desempenho do modelo está diretamente associado a uma métrica de risco associada a qualidade das predições e por isso, é possível compará-la com um valor de risco desejado para saber se o modelo funciona como esperado.

Por fim, a análise do desempenho do modelo quando comparada com o valor de erro de referência faz com que seja possível variar certos hiperparâmetros relacionados à confiança de auditoria. Ou seja, é possível calibrar modelo para que o mesmo automatize mais solicitações em momentos de baixo risco e, caso contrário, assuma um comportamento mais proibitivo.

Neste caso, espera-se que haja oscilação nos valores dos hiperparâmetros relacionados á confiança do modelo até que os mesmos se estabilizem naqueles que mais se adéquem ao cenário de risco. Tal qual o sistema de ar condicionado injeta um ar mais frio do que o desejado afim de que o resfriamento estabilize a temperatura em algum momento.

Capítulo 3

Sistema Proposto

Nesta seção, discutimos o sistema proposto neste trabalho. Inicialmente, o antigo cenário de auditoria de contas é descrito para melhor contextualização do problema. Depois, apresentamos a solução de auditoria eletrônica utilizando um modelo de aprendizado de máquina. Finalmente, expomos o sistema de controle e manutenção do sistema de auditoria automática utilizando o controlador PID.

3.1 Cenário: Auditoria de Contas

A Figura 3.1 ilustra o fluxo de vida das solicitações no sistema de auditoria exclusivamente humana.

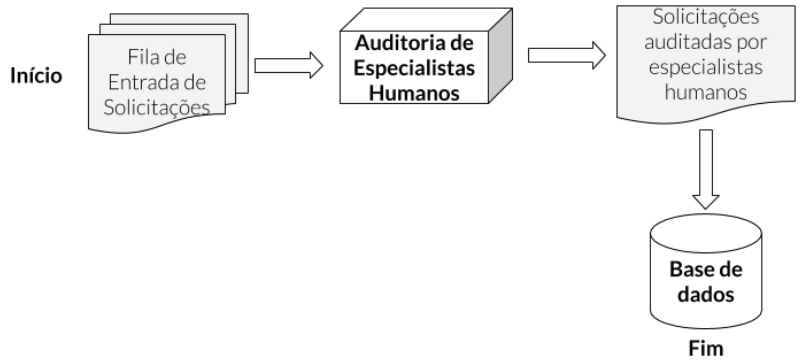


Figura 3.1. Fluxo das solicitações no cenário de auditoria exclusivamente humana

O Fluxo de solicitações é constituído pelas seguintes fases, a saber:

- A **Fila de Entrada de Solicitações** representa a origem dos dados e início do fluxo. A fila contém todas as solicitações que ainda não foram auditadas e estão disponíveis para serem analisadas pelos auditores. A fila é populada através da criação de solicitações pelos clientes da seguradora e é consumida pelos auditores especialistas.
- A **Auditoria de Especialistas Humanos** é a equipe de auditores humanos responsáveis pela análise das solicitações. Para que um especialista realize o processo de auditoria, ele precisa remover uma solicitação da Fila de Entrada e dar início ao processo manual de análise, que envolve consultar contratos, conferir históricos, dentre outras tarefas. Uma solicitação é composta de vários itens (exame de sangue, consulta médica, etc.) que são analisados separadamente pelo especialista e podem ser aprovados ou reprovados individualmente.
- As **Solicitações Auditadas por Especialistas Humanos** são o resultado da auditoria humana. Nessa etapa, as solicitações possuem indicadores de quais itens foram reprovados e quais itens foram aprovados pelo auditor. Vale ressaltar que o nosso trabalho considera que a decisão dos auditores está sempre correta.
- Por fim, o resultado é salvo no **Banco de Dados** da seguradora e ficam disponíveis para a consulta dos clientes.

O processo de auditoria humana, apesar de eficaz, tem muitos problemas relacionados ao tamanho da demanda. Como o volume de solicitações é muito maior que a equipe de auditores, algumas solicitações podem ficar aguardando na fila por dias até serem auditadas e os impactos dessa demora são transmitidos para o cliente que precisa esperar até que a solicitação seja auditada para usufruir do que foi solicitado, caso aprovado.

Para resolver esse problema, foi proposto o sistema de auditoria inteligente, cujo objetivo é automatizar as solicitações utilizando um modelo de aprendizado de máquina para reduzir o tempo de espera dos clientes.

3.2 Sistema de Auditoria Inteligente

A Figura 3.2 ilustra o fluxo de auditoria do sistema de auditoria com a intervenção do modelo de aprendizado de máquina. O diagrama não ilustra o processo de ajuste do modelo, portanto não apresenta a retroalimentação do sistema inteligente pelos dados auditados por especialista. O sistema foi nomeado de Sistema de Auditoria Inteligente.

O Sistema de auditoria inteligente é constiuído pelas seguintes fases, a saber:

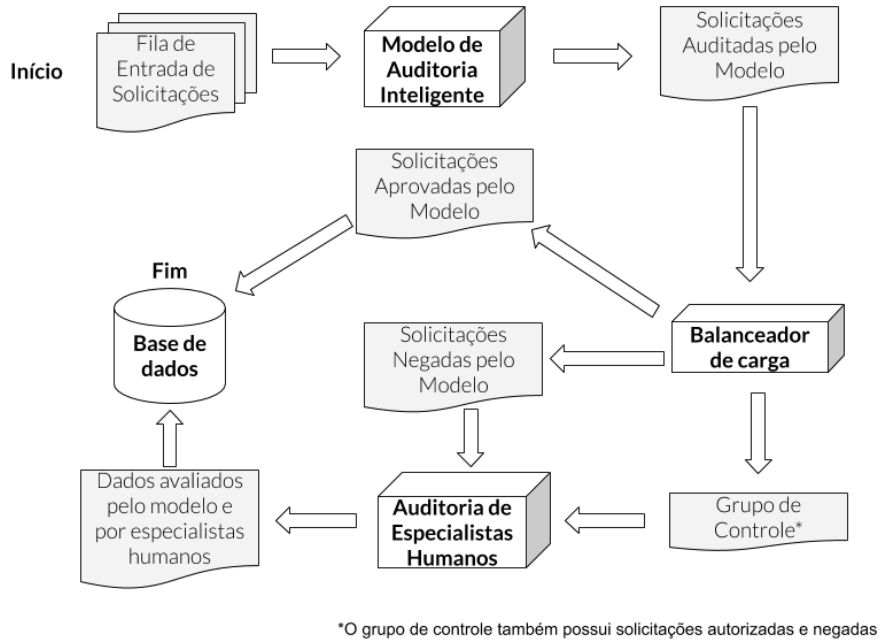


Figura 3.2. Fluxo de dados no sistema de auditoria inteligente.

- O **Modelo de Auditoria Inteligente** é o modelo de aprendizado de máquina criado com a finalidade de automatizar as solicitações. Seu ajuste é feito utilizando as solicitações auditadas previamente pelos auditores humanos.
- O **Balanceador de Carga** é responsável por direcionar as solicitações auditadas pelo modelo em 3 fluxos com finalidades distintas, conforme descrito abaixo:
  - As **Solicitações Aprovadas pelo Modelo** são as solicitações automatizadas pelo modelo. Neste caso, não existe a necessidade da intervenção de um especialista humano na auditoria da conta e por isso, esses dados já podem ser armazenados diretamente no banco de dados.
  - As **Solicitações Negadas pelo Modelo** são solicitações que o modelo não teve confiança suficiente para aprová-las. Para não haver nenhuma injustiça com clientes que deveriam ter suas solicitações aprovadas, mas tiveram-nas negadas pelo modelo, todas as solicitações negadas são enviadas para o grupo de Auditoria de Especialistas Humanos.
  - Por fim, o **Grupo de Controle** representa uma amostragem das solicitações auditadas pelo modelo que são enviadas para auditoria humana, mesmo que tenham sido aprovadas pelo modelo. A amostragem é feita de maneira aleatória com uma probabilidade σ definida manualmente. O grupo de controle tem como funções: manter um conjunto de dados atualizado não

enviesado pelo modelo para fins de reajuste do mesmo; Informar métricas dinamicamente, através do cruzamento do resultado da auditoria do modelo e da auditoria humana e como consequência, é utilizado para a manutenção automática do sistema.

- A **Auditoria de Especialistas Humanos** é responsável por analisar as solicitações que o modelo negou e aquelas enviadas para o grupo de controle. É importante ressaltar que o auditor humano não sabe qual é a resposta da auditoria do modelo para as solicitações no grupo de controle.

Ao comparar o resultado da auditoria do modelo com a auditoria humana é possível conferir a assertividade do modelo e com isso utilizar métricas comuns em problemas de aprendizado de máquina.

A Figura 3.3 mostra uma matriz de contingência, através dela é possível extrair métricas importantes para a avaliação de modelos. O canto inferior esquerdo contém as solicitações aprovadas pelo modelo e pela auditoria, esses itens estão corretamente classificados (cor verde). No canto superior esquerdo, encontram-se as solicitações aprovadas pelo modelo, mas negados pela auditoria (cor vermelha). Essas solicitações geram um prejuízo financeiro para a empresa pela liberação indevida de recursos.

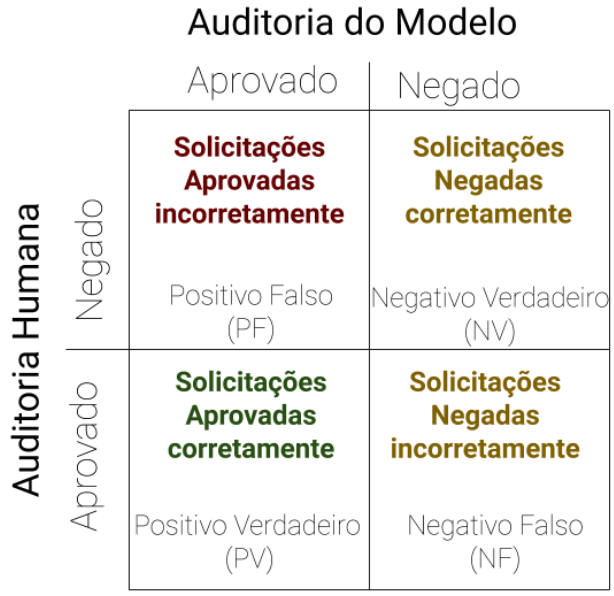


Figura 3.3. Matriz de contingência utilizada para avaliação do modelo.

Do lado direito, na cor amarela, encontram-se os itens negados pelo modelo. Independentemente do que diz a auditoria, esses itens não foram automatizados e portanto

não fornecem riscos atrelados à automação. Porém, sua divisão ainda é importante para o cálculo de algumas das métricas abaixo:

- A **precisão** é calculada pelo resultado da divisão do valor das solicitações aprovadas corretamente pelo modelo pelo número total de solicitações aprovadas pelo modelo. Pela figura: $\frac{PV}{PV+PF}$.
- A **revocação** é calculada pelo resultado da divisão do valor das solicitações aprovadas corretamente pelo modelo pelo número total de solicitações aprovadas pela auditoria humana. Pela figura: $\frac{PV}{PV+NF}$.
- O $F1$ é calculado pela média harmônica da precisão com a revocação. Pela figura: $2 * \frac{precisao * revocacao}{precisao + revocacao}$. O valor de $F1$ é um indicador de que o modelo tem boa cobertura e boa assertividade.
- Por fim, a **taxa de automação** é calculada pelo resultado da divisão do valor das solicitações aprovadas pelo modelo pelo somatório total de solicitações. Pela figura: $\frac{PV+PF}{PV+PF+NV+NF}$.

A auditoria do modelo inteligente ocorre da seguinte maneira: cada solicitação R é composta de vários itens r, conforme descrito anteriormente. A solicitação é aprovada pelo modelo se e somente se todos os itens pertencentes a ela forem aprovados. Isso significa que se uma solicitação contém um pedido para um exame de sangue e uma consulta médica, onde o primeiro foi negado e o segundo foi autorizado, então o resultado final será a negação da solicitação de ambos os itens.

Todos os itens das solicitações são avaliados pelo mesmo modelo, porém diferentes tipos de itens possuem diferentes peculiaridades. Por exemplo, um exame de sangue é mais barato e é solicitado com mais frequência do que uma cirurgia cardiovascular, por isso, a auditoria da cirurgia é muito mais difícil e arriscada do que a auditoria do exame de sangue. No conjunto de teste utilizado para fazer os experimentos do trabalho existem mais de 4000 tipos diferentes de itens, no Capítulo 4 serão apresentados mais detalhes sobre os dados.

Em modelos de classificação, a avaliação de um item s_r, onde $0 \geq s_r \geq 1$, deve ser maior que um limiar th, onde $0 \geq th \geq 1$ para que o item seja aprovado. Entretanto, note que seria um erro se fosse considerado apenas um único limiar para todos os tipos de item, uma vez que, como dito anteriormente, eles possuem diferentes riscos e impactos na seguradora. Assim, para cada tipo de item $i \in I$, onde I é o conjunto de todos os tipos de itens presentes nas solicitações, são calculados os valores de limiares

ótimos th_i^* específicos para cada tipo de item. Então a condição final para que um item seja aprovado se torna $s_r \geq th_{i_r}^*$.

Assim, se todos os itens de uma solicitação são aprovados, a solicitação é aprovada. Uma solicitação aprovada é considerada automatizada pelo sistema de auditoria inteligente. Para que seja possível medir o sistema dinamicamente, é necessário estabelecer algumas métricas importantes para o problema.

Para calcular os valores dos limiares ótimos $th_i^* \forall i \in I$, o sistema executa um algoritmo após a etapa de ajuste do modelo, onde realiza os seguinte processo iterativo:

- Inicializa-se um *threshold* $th_i = 1$;
- Utilizamos o modelo previamente treinado para calcular cada instância do item i no conjunto de otimização;
- Calculamos o *score* $F1_{th_i}$ utilizando th_i como limiar para os itens e comparando-os com os *labels* verdadeiros;
- Se o th_i^* não fora calculado anteriormente ou o novo $F1_{th_i}$ é maior que o $F1_{th_i^*}$ encontrado para o th_i^*, então $th_i^* = th_i$;
- $th_i = th_i - 0.01$ se $th_i \geq 0$;

Ao final desses passos, para cada $i \in I$ vai existir um limiar th_i^*, que será utilizado como referência para avaliar se um item será aprovado ou não. Note que $0 \geq th_i^* \geq 1$ e quanto maior for o valor desse limiar mais difícil será para que o item seja aprovado. Essa propriedade é fundamental para a criação do **controlador** (θ), que será utilizado para tornar o sistema mais ou menos permissivo.

Para assegurar que os tipos de itens avaliados de forma insatisfatória pelo modelo não interfiram na qualidade no sistema foi estabelecido um valor de segurança $F1_{min}$, onde:

$$th_i^* = \begin{cases} th_i^* & if\ F1_{th_i^*} \geq F1_{min} \\ 1 & otherwise \end{cases} \quad \forall i \in I \tag{3.1}$$

Entretanto, mesmo calculando todos os valores ótimos dos limiares, ainda não é possível controlar automaticamente o risco do modelo. Por isso, também implementamos um sistema de manutenção automática do sistema utilizando os conceitos do controlador PID.

3.3 Sistema de Manutenção Automática

Uma mudança abrupta na distribuição das características, por causa de de um efeito de temporada ou até mesmo um *bug* em algum dos sistemas da seguradora, pode fazer com que o sistema produza uma grande quantidade de respostas falso positivas. Caso essas situações venham a acontecer, pode ocorrer um risco financeiro que chega em valores muito perigosos para o orçamento da seguradora.

Por isso, é necessária a realização de manutenção periódica no sistema cuja finalidade é assegurar que o modelo continue estável ao longo do tempo. A Figura 3.4 ilustra a adição desse mecanismo de manutenção no sistema.

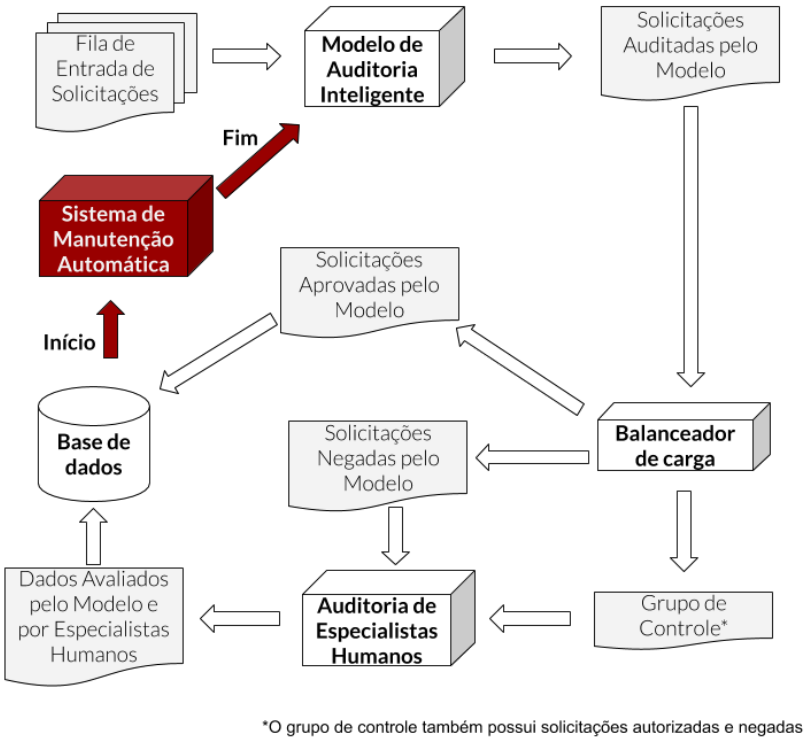


Figura 3.4. Fluxo de dados no sistema de auditoria inteligente.

O primeiro passo para a criação do mecanismo de manutenção automática é a criação de um outro parâmetro (batizado de **controlador**) que seja capaz de fazer o ajuste fino dos valores ótimos dos limiares.

Então, seja $th_i^* \forall i \in I$, os valores ótimos dos limiares calculados conforme descrito na Equação 3.1: existe um valor $th_i' \forall i \in I$, onde $th_i^* + th_i' = 1$. E uma vez que $0 \geq th_i^* \geq 1$, então $0 \geq th_i' \geq 1$. Assim, seja θ um valor do **controlador** escolhido arbitrariamente onde $0 \geq \theta \geq 1$. Assim, o valor final do limiar para que os itens sejam aprovados (th_i) pode ser definido como:

$$th_i = th_i^* + \theta * th_i' \quad \forall i \in I \tag{3.2}$$

ou

$$th_i = th_i^* + \theta * (1 - th_i^*) \quad \forall i \in I \tag{3.3}$$

Assim, se o sistema está produzindo muitos falso positivos, é possível aumentar o valor do **controlador**, fazendo com que a taxa de automação caia para evitar riscos. Em um caso extremo, onde $\theta = 1$ tem-se.

$$\begin{aligned} th_i &= th_i^* + 1 * (1 - th_i^*) \\ &th_i^* + 1 - th_i^* \\ &1 \quad \forall i \in I \end{aligned} \tag{3.4}$$

E assim, nenhum item vai ser aprovado pelo sistema e como consequência, nenhuma solicitação será aprovada.

A simples adição desse controlador demandará mais esforço de trabalho humano, além do que já é despendido no grupo de controle. Uma vez que o valor do controlador deve ser definido manualmente e analisado empiricamente.

Para saber se o modelo está performando corretamente, é necessária a análise humana que é feita dentro do grupo de controle. Além disso, note que o valor do controlador ainda precisa ser definido manualmente, demandando um esforço humano extra. Como saber o melhor valor possível para θ em função do custo de erro gerado pelo sistema? Com que frequência ele deve ser atualizado?

Seja f_R, o custo pago pela seguradora por uma requisição R autorizada pelo modelo. Se R deveria ser aprovada, então o sistema economizou dinheiro liberando as horas dos auditores. Se R deveria ser negada, então o modelo desperdiçou dinheiro ao fazer com que a seguradora pague por algo que não deveria pagar. Seja t_R, o resultado real de uma requisição que foi aprovada, onde $t_R = 1$ se a solicitação deveria realmente ser aprovada e $t_R = 0$ caso contrário, considere uma janela temporal de W dias no passado, onde o último dia é a data atual, e seja $\mathbf{R}_W$ o conjunto das solicitações que foram autorizadas pelo modelo dentro desta janela temporal W. É possível estimar uma taxa de erro e em função do custo das aprovações erradas por:

$$e_W = \frac{\sum_R^{\mathbf{R}_W} (f_R * (1 - t_R))}{\sum_R^{\mathbf{R}_W} (f_R)} \tag{3.5}$$

Dessa forma, e_W vai possuir um valor alto quando o número de aprovações erradas

for grande em relação ao número de aprovações totais. Pode-se traduzir esse valor como uma regra, onde o sistema pode cometer mais erros à medida que o sistema acerta mais. Além disso, pela Equação 3.5, é possível ver que $0 \geq e_W \geq 1$.

Por fim, uma vez que é possível calcular a taxa de erro dinâmica, nós podemos definir um valor de referência (e_{ref}) para o erro, onde e_{ref} representa a taxa de erro que é aceita pela seguradora. Assim, aplicamos os conceitos dos controladores PID para controlar o valor de θ, utilizando o erro e o valor de referência. Se $e_W << e_{ref}$, podemos decrescer o valor de θ, permitindo que o modelo assuma mais riscos e aumente sua taxa de automação. À medida que o valor de e_W vai se aproximando de e_{ref}, aumentamos o valor do controlador θ, fazendo com o que modelo precise de mais confiança para aprovar uma solicitação, fazendo assim com que a taxa de automação caia, reduzindo o número de aprovações indevidas.

Assim, a equação que atualiza o valor do controlador θ, considerando ambos os valores e_W e e_{ref} é:

$$\theta = \theta - \alpha(e_{ref} - e_W) \tag{3.6}$$

Se $e_W < e_{ref}$, então o novo valor de θ é menor que o antigo, isso significa que o sistema vai ser mais permissivo na automação. Contrariamente, se o valor de θ crescer, o sistema terá um comportamento mais proibitivo, fazendo com que as taxas de automação caiam. A atualização do valor do controlador é chamada periodicamente (o intervalo também é definido como um hiperparâmetro).

Note que estamos utilizando apenas o fator proporcional do controlador PID, fazendo com que o sistema seja classificado como *Controlador Proporcional*.

3.4 Pseudocódigo

Afim de tornar mais claro a explicação do sistema proposto, serão apresentados dois algoritmos em pseudocódigo que descrevem de maneira simples o processo de Auditoria Inteligente e Manutenção automática.

O Algoritmo 1 demonstra o processo de auditoria automática. Nele o modelo treinado de aprendizado de máquina (M), o conjunto dos limiares ótimos calculados (Th), o valor do controlador (θ) e a solicitação (R), que é composta de vários itens (r), são utilizados para obter a resposta final.

Primeiramente o modelo de aprendizado de máquina avalia todos os itens (r) dentro da solicitação, produzindo um conjunto de resultados S onde $0 \geq s \geq 1 \forall s \in S$ para cada um dos itens da solicitação.

Algorithm 1 Auditoria Inteligente de Solicitações

Require: $M \vee Th \vee \theta \vee R$
  $S \leftarrow score(M, R)$
  for $s \in S$ **do**
    $r \leftarrow 1$
    if $s.value < (Th|s.type| + \theta * (1 - Th|s.type|))$ **then**
      $r \leftarrow 0$
    end if
  end for
  if $r\ is\ 1$ **then**
    $R.model\_score \leftarrow 1$
    return R was approved by model
  else
    $R.model\_score \leftarrow 0$
    return R needs human auditioning
  end if

Em seguida, os resultados são comparados com os valores de referência dos limiares Th de acordo com o tipo de cada item. Nessa comparação é possível ver a adição do complemento do limiar ótimo do item que é proporcional ao valor do controlador.

Por fim, se todos os resultados dos itens foram maiores que os valores de limiares necessários para aprovação, então a solicitação é aprovada automaticamente, caso contrário, ela é enviada para auditoria humana.

Algorithm 2 Manutenção Automática dos Modelos

Require: $R_{cg} \vee \theta \vee \alpha \vee reference$
  $error\_cost \leftarrow 0$
  for $R \in R_{cg}$ **do**
    if $R.model\_score\ is\ 1 \vee R.human\_score\ is\ 0$ **then**
      $error\_cost \leftarrow error\_cost + R.cost$
    end if
  end for
  $total\_cost \leftarrow \sum(r.cost)\ \forall R \in R_{cg}\ if\ R.model\_score\ is\ 1$
  $cost \leftarrow error\_cost/total\_cost$
  $\theta \leftarrow \theta - \alpha * (error - reference)$
  return θ

O processo de manutenção automática consiste em calcular novos valores do controlador θ em função do valor atual da métrica de risco em comparação com seu valor de referência, para que o modelo se adéque automaticamente a uma situação de alto ou baixo risco. O Algoritmo 2 demonstra o processo do cálculo do valor do controlador.

Nele, utilizamos as solicitações do grupo de controle R_{cg}, pois elas possuem ambas as avaliações humana e automática. Como consideramos que a auditoria humana está

sempre correta, é possível saber se o modelo acertou ou errou sua avaliação.

Para cada solicitação dentro do grupo de controle ($R \in R_{cg}$), comparamos o resultado da auditoria humana com a automática. Cada solicitação autorizada erroneamente pelo modelo gera um custo ($R.cost$) que deve ser arcado pela empresa ($error\_cost$).

Por outro lado, cada solicitação autorizada automaticamente economiza tempo dos auditores e dos clientes. Nós utilizamos o custo total das solicitações autorizadas ($total\_cost$) para converter o custo do erro em um fator proporcional de forma a aplicar as teorias de controle PID.

Em posse do valor proporcional do custo ($cost$), pode-se comparar com o erro de referência ($reference$). Caso o custo seja menor que a referência, então podemos diminuir o valor do controlador para que o modelo tenha mais liberdade de autorizar. Em caso contrário, o valor do controlador irá aumentar fazendo com que os limiares de aprovação dos itens também aumentem e com isso, a automação caia.

Capítulo 4

Experimentos e Análises

Neste capitulo, discutimos os experimentos e as análises realizadas.

4.1 Ferramental Prático

O sistema foi programado utilizando a linguagem *Python*[1] e fica hospedado na infraestrutura interna da empresa. Os algoritmos utilizados no modelo de aprendizado de máquina são o *XGBoost* [Chen & Guestrin, 2016] e o *LightGBM* [Ke et al., 2017], sendo que o primeiro obteve resultados melhores que o segundo. As entradas que são enviadas para classificação do modelo possuem mais de 1000 características (Após o pré-processamento).

4.2 Base de Dados

A base de dados utilizada para os experimentos foi coletada no ambiente de produção da seguradora, portanto os dados das solicitações utilizadas são reais e passaram por todo pipeline do sistema descrito no Capítulo 3.

Os dados foram coletados no período de 1º de outubro de 2018 até o dia 19 de agosto de 2019. A Tabela 4.2 possui a volumetria dos dados utilizados no experimento. É possível observar que os volumes se mantém constante ao longo dos meses, com picos nos meses de janeiro, maio e julho. O grupo de controle foi maior no mês de outubro devido à uma regra interna e o mês de agosto teve volume menor pois a coleta se interrompeu no meio desse mês.

[1] `https://www.python.org/`

A escolha desses meses é compatível com as boas práticas de validação cruzada em séries temporais [Bergmeir & Benítez, 2012], uma vez que o ajuste do modelo de aprendizado de máquina do sistema foi feito com dados coletados em meses anteriores ao período supracitado.

Tabela 4.1. Volumetria dos dados obtidos da base em produção.

Mês	# de Solicitações	# de Itens	% do Grupo de Controle
Outubro/2018	18.205	53.312	28,23%
Novembro/2018	16.842	48.850	15,20%
Dezembro/2018	14.321	43.838	14,37%
Janeiro/2019	17.259	50.998	14,24%
Fevereiro/2019	17.147	48.557	14,86%
Março/2019	16.795	47.098	14,30%
Abril/2019	17.874	46.463	14,63%
Maio/2019	18.533	46.564	14,96%
Junho/2019	16.227	42.702	14,56%
Julho/2019	18.466	48.921	14,67%
Agosto/2019	10.415	28.127	14,59%

Como os dados foram extraídos do sistema já em produção, observa-se um viés onde as solicitações que foram aprovadas pelo modelo (automatizadas) não passaram pela auditoria humana, enquanto que as solicitações negadas pelo modelo passaram pelo processo humano de auditoria. Esse viés deve ser levado em consideração, uma vez que queremos medir o tempo economizado pela automação. Ou seja, gostaríamos de descobrir o impacto real na vida dos clientes e na vida dos auditores.

Para calcular o tempo economizado pela automação do sistema, é necessário saber quanto tempo uma solicitação demora para ser auditada completamente. E para isso, foram consultados os *logs* das aplicações da seguradora e coletados os tempos de abertura e fechamento dessas solicitações. Dessa forma, é possível ter uma medida razoável do tempo de espera do cliente.

Porém note que, conforme supracitado, existe um viés na base de dados, onde as solicitações não autorizadas pelo modelo tem um tempo de processamento maior as autorizadas pelo mesmo. Para resolver esse problema e obter cálculos mais justos, foram utilizados apenas os dados do grupo de controle para a realização dos experimentos. Dessa forma, retiramos todo esse viés de temporal introduzido pelo modelo em produção, uma vez que o grupo de controle obrigatoriamente é auditado por humanos independentemente da resposta do modelo.

Além disso, consideramos que a auditoria humana é o *ground-truth* das respostas, ou seja, assumimos que a resposta dos humanos é sempre correta. Portanto, estamos desconsiderando a existência de um viés humano nesse julgamento.

Por fim, ao utilizar apenas os dados do grupo de controle, é possível obter algumas métricas mais precisas, como o próprio tempo de auditoria e acurácia real do sistema. Nesta nova configuração, foram sorteadas algumas solicitações para fazer parte do grupo de controle que será utilizado para que o controlador PID ajuste o valor do controlador em prol do erro de automação. A Tabela 4.2 mostra a nova distribuição dos dados utilizando apenas o grupo de controle.

Tabela 4.2. Volumetria dos dados utilizados na realização dos experimentos. Note que a redução dos dados em relação a tabela anterior é de aproximadamente 85% (exceto no mês de outubro), correspondente ao grupo de controle real. Nos novos dados é possível observar as horas totais gastas com auditoria em cada mês (Tempo Auditoria) e o número de solicitações que foram enviadas para o grupo de controle artificial, que é utilizado para operar o controlador PID.

	# Solicitações	# Itens	Tempo Auditoria (em Horas)	# Solicitações no Grupo de Controle
Outubro/2018	7.440	22.034	2.454.111,99	1.586
Novembro/2018	3.660	10.233	1.121.494,70	707
Dezembro/2018	2.965	8.801	1.136.124,18	562
Janeiro/2019	3.592	10.558	981.260,68	692
Fevereiro/2019	3.704	10.465	1.153.244,63	682
Março/2019	3.445	9.371	1.158.581,54	676
Abril/2019	3.766	9.560	1.679.295,91	762
Maio/2019	3.978	9.725	1.709.283,16	762
Junho/2019	3.413	8.630	1.315.429,96	675
Julho/2019	3.884	10.489	1.554.204,20	760
Agosto/2019	2.239	5.950	965.378,08	455

4.3 Sistema Oráculo

O sistema oráculo utiliza a classificação da auditoria humana como resultado da auditoria automática, ou seja, o modelo sempre concorda com o auditor. Como o trabalho considera que o auditor humano está sempre certo em seu julgamento, então o sistema também está sempre correto.

Os gráficos da Figura 4.1 mostram as métricas de interesse na avaliação do oráculo. Na imagem, é possível ver que a automação do oráculo é em torno de 70% em todos os meses de experimentação. O custo relativo é sempre 0 uma vez que o oráculo

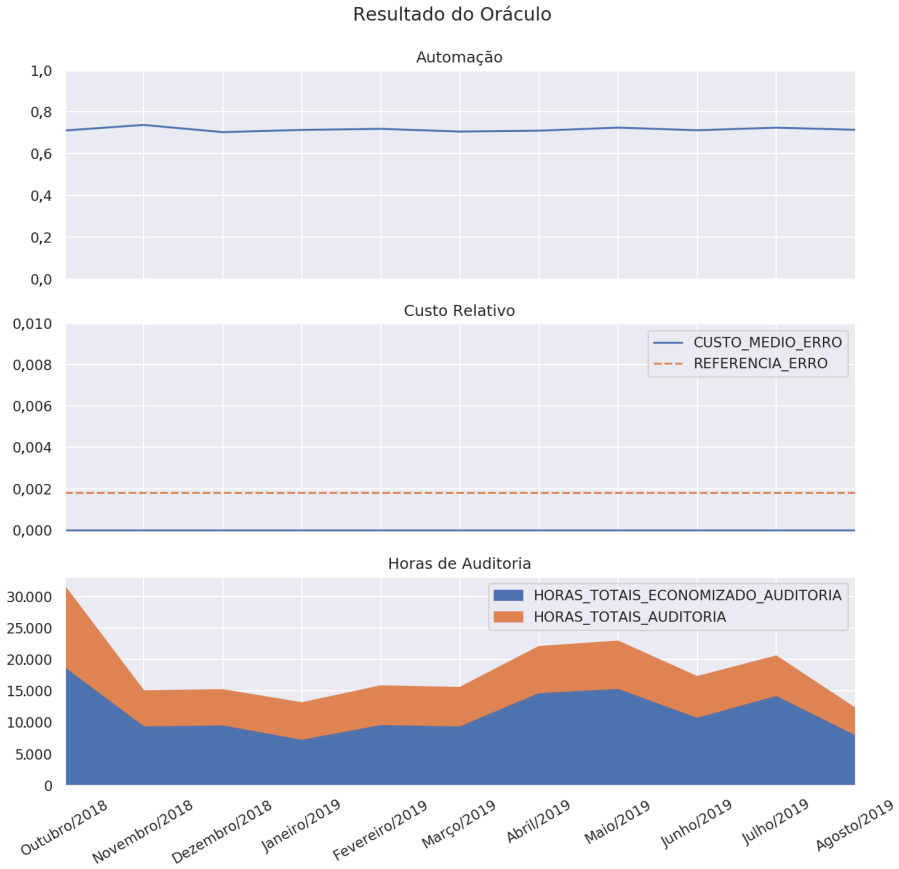


Figura 4.1. Métricas coletadas no experimento utilizando o sistema oráculo.

não comete erros de classificação. Por fim, é possível ver que a economia de horas de auditoria humana pelo sistema oráculo representa cerca de 55% até 60%. Essa queda em relação ao valor da automação pode ser explicado pela existência de solicitações complexas que acabam sendo negadas no fim do processo de auditoria.

Ao criar o sistema oráculo, definimos assim um *baseline* para comparação com as diferentes variações dos hiperparâmetros, pois será possível descobrir o quão longe está a configuração escolhida do mundo ideal, que é o sistema oráculo. Outras abordagens, como *Random Forests* [Liaw et al., 2002] ou técnicas de detecção de anomalias, poderiam ser utilizadas também como *baseline*, entretanto mantivemos a utilização do *XGBoost* por ter sido o algoritmo que obteve melhor desempenho no ajuste do modelo.

Nas próximas seções, apresentamos as análises do comportamento do sistema de acordo com a variação de cada hiper-parâmetro.

4.4 Experimentos

Conforme dito no Capítulo 3, o sistema de controle possui vários hiper-parâmetros que devem ser ajustados por um ser humano para que o mesmo funcione, a saber:

- **Intervalo de Atualização do Controlador.** Define o intervalo de tempo em que o sistema irá acionar a rotina de controle para atualizar o valor de controlador. Um intervalo pequeno demais pode tornar o sistema instável, principalmente em períodos de tempo com poucas solicitações. Já um intervalo muito grande pode fazer com que o sistema não seja efetivamente controlado.

- **Limiar Mínimo de F1 ($F1_{min}$).** Valor do corte de F1 (Equação 3.1) que define o valor mínimo necessário para que um item seja automatizado. Itens com F1 menores que o Limiar são automaticamente enviados para auditoria humana.

- **Custo de referência (e_{ref}).** (Equação 3.5) É o valor proporcional de dinheiro que a companhia aceita perder com a automação.

- **Taxa Alfa (α).** Parâmetro de aprendizado do controlador proporcional, assim como o Intervalo de Atualização, exerce a função de acelerar ou retardar a velocidade com que o controlador muda.

- **Valor Inicial do Controlador (θ_0).** Valor inicial para o controlador.

- **Valor Mínimo do Controlador** e **Valor Máximo do Controlador.** os limiares inferiores e superiores do controlador. Lembrando que se $\theta = 0$, o sistema opera confiando totalmente no julgamento do modelo de aprendizado de máquina. Já em $\theta = 1$, o sistema envia todas as solicitações para auditoria humana e a automação do sistema é 0.

Além de explicação dos hiper-parâmetros, ressalta-se também que estamos interessados em avaliar algumas métricas específicas com foco no nosso estudo de caso (Seção 1.2), são elas:

- **Taxa de automação.** é a razão entre as solicitações aprovadas pelo sistema pelo número de solicitações totais. Está diretamente relacionada com o volume de informação que estamos redirecionando para o modelo e liberando dos auditores humanos, de forma que estes últimos possam trabalhar em solicitações mais complexas.

- **Tempo Economizado pela Automação.** O processo de auditoria humana demanda análise das informações presentes nas solicitações. Como o número de auditores é bem menor que o número de solicitações, elas demoram um tempo para serem auditadas e logo o solicitante (cliente) tem que esperar até que sua solicitação seja aprovada. Por isso é interessante saber quanto tempo o sistema está economizando em relação a espera do cliente, uma vez que a solicitação é processada pelo sistema em no máximo dois minutos.

- **Dinheiro Gasto com Automação Incorreta.** No Capítulo 3, dissemos que o sistema possui como referência, um gasto financeiro aceitável com automações incorretas, porém dadas as características do sistema PID, não é garantido que o erro real seja sempre menor que a referência e portanto, é interessante obter o valor real para análise dos gastos.

Os experimentos contém variações de alguns desses hiper-parâmetros para que seja possível avaliar as consequências nas métricas avaliadas. Além disso, analisamos o comportamento de um sistema "oráculo", que classifica corretamente qualquer solicitação observada.

Por fim, também estamos interessados em responder as seguintes perguntas de pesquisas (PP) a respeito da implantação da solução proposta no ambiente citado:

1. É possível implantar um sistema de manutenção automática em modelos de aprendizado de máquina de forma a retardar a queda de eficiência do mesmo e que seja estável?

2. Existe um valor mínimo para a escolha do erro de referência utilizado no controlador PID nesse contexto?

4.4.1 Taxa Alfa

A taxa alfa (α) controla controla a intensidade do fator proporcional de um controlador PID. Caso o valor seja muito pequeno, o sistema não responderá com agilidade às alterações do risco financeiro e o controlador tenderá a ficar estático. Do contrário, caso o valor seja alto, o sistema pode ficar instável, fazendo com que o controlador fique oscilando nos extremos.

A Figura 4.2 ilustra o comportamento do controlador médio e o custo médio do erro para diferentes valores de alfa ao longo dos meses. Pelos gráficos, é possível ver que para os valores mais baixos $\alpha = 0,0001$ (linha azul) e $\alpha = 0,001$ (linha amarela), o controlador não responde com agilidade à variação do custo do erro e por isso, tende a

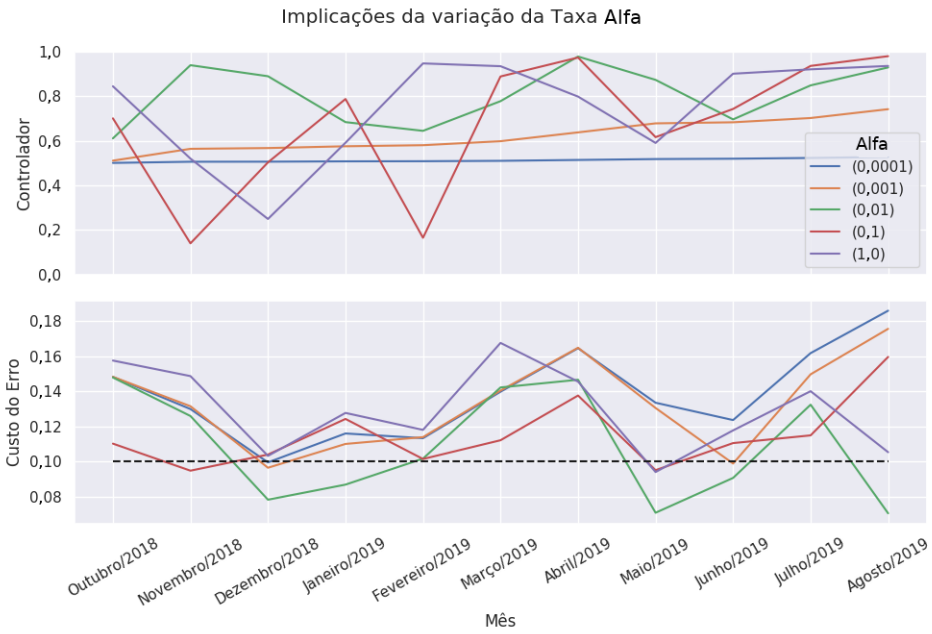


Figura 4.2. Variação do valor do controlador e custo do erro para diferentes valores da taxa alfa. Cores iguais representam o mesmo valor de alfa conforme a legenda. A linha hachurada preta no segundo gráfico indica o custo de referência.

fazer com que esse custo atinja valores altos. Como consequência, vê-se que nos meses de Julho/2019 e Agosto/2019 o custo do erro é quase o dobro da referência.

Em alternativa, para o valor mais alto $\alpha = 1$ (linha roxa) nota-se instabilidade do valor do valor do controlador, os valores oscilam em regiões próxima das bordas (0 e 0,98). Esse comportamento faz com que a operação do sistema fique muito permissiva em alguns momentos (fazendo o custo crescer muito rápido), o crescimento acelerado do custo, por sua vez, joga o valor do controlador pro outro extremo, fazendo com que o mesmo opere em modo proibitivo, diminuindo a automação.

Os valores $\alpha = 0,01$ (linha verde) e $\alpha = 0,1$ mantiveram o valor do controlador mais estável ao longo do tempo, fazendo com que o custo oscilasse pouco ao redor da referência, sendo os valores mais indicados para a operação nesse conjunto de dados.

4.4.2 Custo de Referência

O custo de referência é o hiperparâmetro mais sensível do sistema, pois ele está relacionado com o risco tolerável. Se o custo de referência for muito alto, o sistema irá operar em um modo permissivo, fazendo com que mais requisições sejam aprovadas erroneamente. Do outro lado, se o risco tolerável for pequeno, qualquer aprovação indesejada fará com que o sistema entre no modo proibitivo, prejudicando a automação.

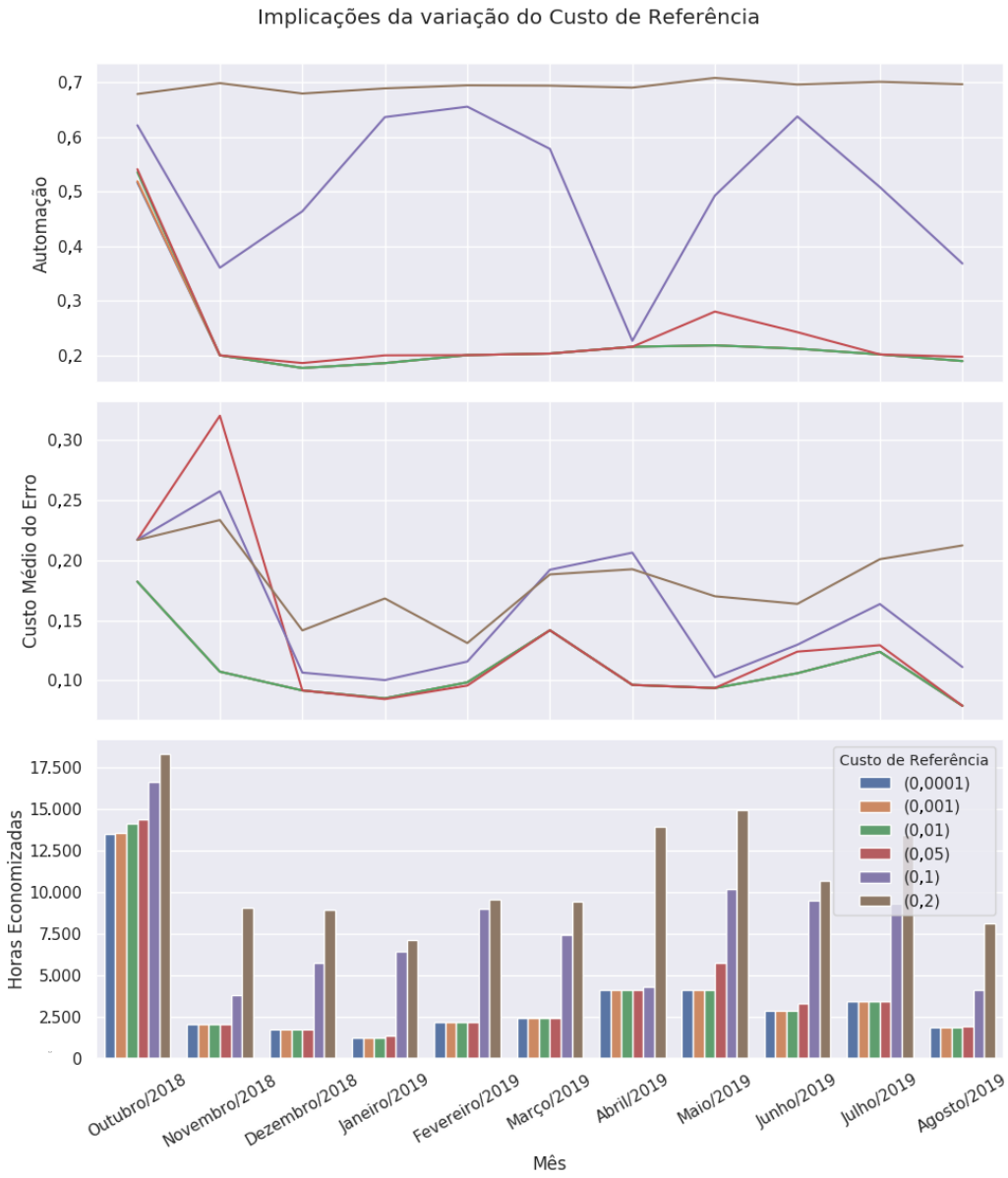


Figura 4.3. Variação das métricas desejadas para diferentes valores de custo de referência. Cores iguais representam o mesmo valor custo de referência conforme a legenda.

A Figura 4.3 ilustra o comportamento das métricas importantes para cada valor do custo de referência. No geral, como o esperado, um alto valor de referência implica em mais automação, que por sua vez aumenta o número de horas economizadas.

O custo de referência depende muitas vezes de decisões estratégicas e orçamentárias. Uma heurística para decidir um bom valor do custo de referência pode ser obtida ao se comparar o recurso gasto com o risco *versus* o recurso ganho com a automação.

Dependendo do cenário, uma análise agrupando os recursos pelo seu custo também pode fornecer informações para a escolha do custo de referência.

No gráfico, para os valores mais baixos, $e_{ref} = 0,0001$ (linha azul), $e_{ref} = 0,001$ (linha amarela), $e_{ref} = 0,01$ (linha verde) e $e_{ref} = 0,5$ (linha vermelha), observa-se um desempenho extremamente ruim do sistema, onde a automação fica presa em 20% e os custos chegam a ultrapassar em até 100 vezes o custo de referência ($e_{ref} = 0,0001$). Além disso, note que as curvas do custo médio do erro são extremamente correlacionadas, sugerindo a existência um custo de referência mínimo para que o sistema funcione. Esse comportamento se deve a característica intrínsecas do problema, como por exemplo o valor dos itens das requisições.

O valor mais alto do custo médio do erro é encontrado quando $e_{ref} = 0,5$ no mês de Novembro de 2018. Esse valor é 6 vezes maior do que o desejado. Essa característica mostra que o sistema está sujeito a cometer erros em diferentes solicitações dependendo dos valores das variáveis naquele instante de tempo. Esse pico se deve a uma solicitação extremamente cara que foi aprovada de forma indevida pelo sistema em um momento onde havia-se economizado muito pouco com as solicitações aprovadas corretamente, mas ainda sim o controlador estava muito baixo. Essa mesma solicitação foi aprovada nos experimentos onde o $e_{ref} > 0,5$, mas o sistema havia conseguido aprovar um número maior de requisições corretas. Fazendo com que o impacto fosse menor.

Em seguida, ao analisarmos os experimentos com $e_{ref} = 0,1$ (Linha Roxa) e $e_{ref} = 0,2$ (Linha Marrom), vemos o sistema funcionando com boa taxa de automação. Para $e_{ref} = 0,1$, o sistema ultrapassa consideravelmente o custo de referência nos meses de Novembro/2018, Março/2019, Abril/2019, Junho/2019 e Julho/2019, fazendo com que a automação tenha quedas nesses meses chegando a 23% em Abril/2019. Nesse cenário, o sistema parece funcionar mais próximo do desejado, porém ainda apresenta picos de custo de aprovações erradas que podem não ser desejadas. O comportamento quando $e_{ref} = 0,2$ temos um comportamento próximo do visto no oráculo, porém com o custo do erro alto na maioria dos meses. Nesse caso, não seria necessário utilizar esse sistema de controle, dado que a empresa estaria disposta a arcar com o alto custo.

Por fim, notamos um valor alto para as horas auditadas no primeiro mês, visto que o conjunto de dados tem mais instâncias em Outubro/2018. No geral, para valores maiores de e_{ref}, economiza-se mais horas de auditoria (com diferença notável a partir de $e_{ref} \geq 0,1$). Além disso, é possível perceber que para os valores mais baixos do custo de referência, o sistema de manutenção automática fica instável, fazendo com que o desempenho do modelo fique ruim. Logo, parece sim haver um valor mínimo de referência para o erro (respondendo positivamente a PP 2) que é derivado de características de como o risco é associado às solicitações deste problema.

4.4.3 Parâmetros do Controlador

Como dito no Capítulo 3, o controlador (θ) é um número real onde $0 \geq \theta \geq 1$. Entretanto, essa última condição só precisa ser verdade se quisermos um cenário onde o limiar de aprovação de um item seja no máximo o valor que foi calculado em função do *F1* (quando $\theta = 0$) e de que seja possível fazer com que todos os limiares sejam iguais a 1 (quando $\theta = 1$).

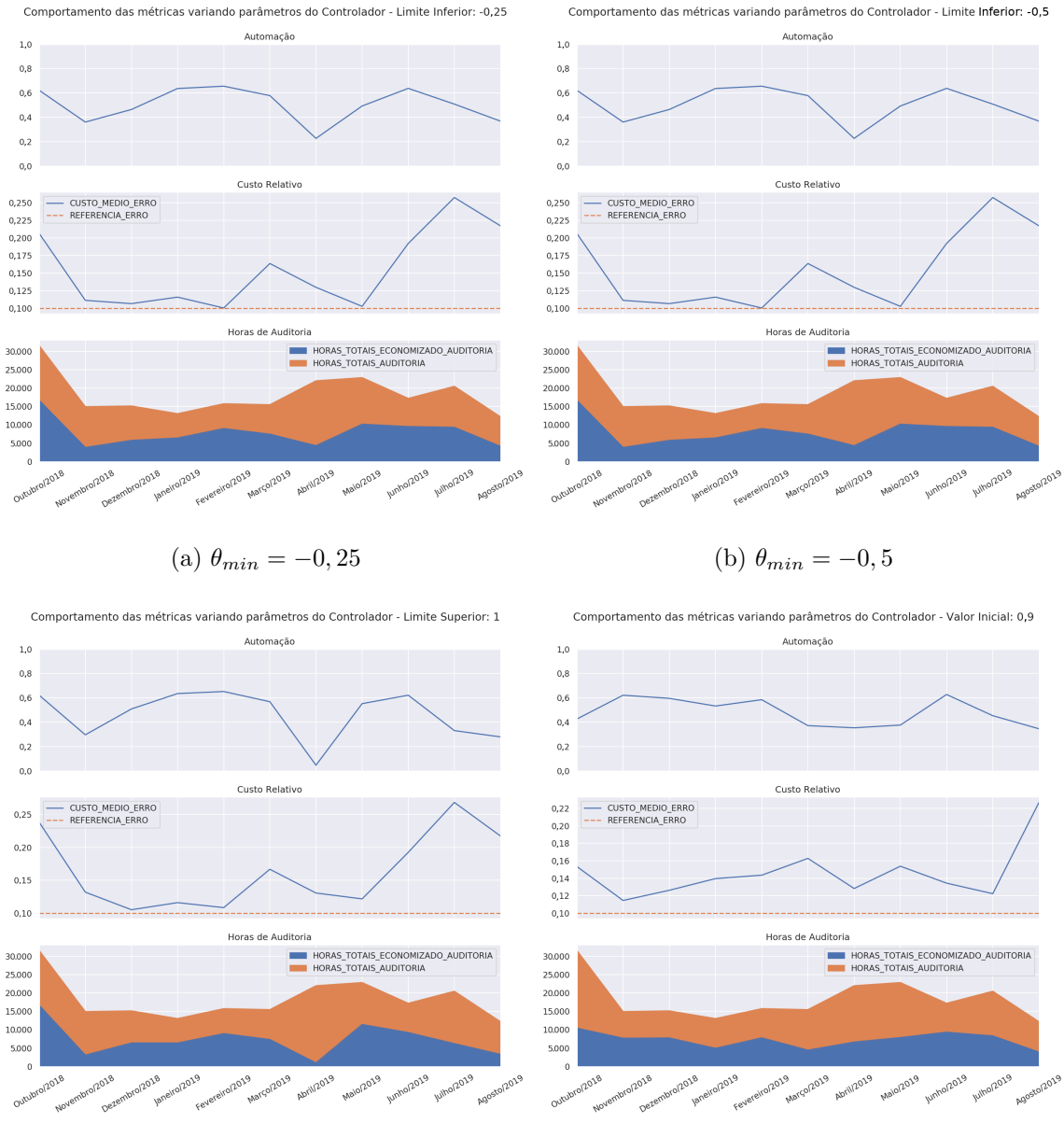
De fato, o limiar inferior (θ_{min}) $0 \geq \theta$ faz sentido, uma vez que temos o parâmetro $F1_{min}$ que, segundo a Equação 3.1, funciona como uma trava de segurança para que o sistema só processe itens com alta confiança. Uma vez que $\theta < 0$, estaremos indiretamente relaxando os limiares, tornando-os mais propensos a serem automatizados.

Entretanto, relaxar a condição onde $\theta \geq 1$ (θ_{max}) faz sentido em um ambiente de produção do sistema de automação, uma vez que pode ser desejável que o sistema continue automatizando algumas requisições mesmo depois de estourar o custo de referência. Ao fazer com que o limite superior seja um número próximo de 1, fazemos com que o sistema continue automatizando os itens mais fáceis de serem automatizados. Prontamente, vale observar que um valor $\theta > 1$ não faz sentido dada a modelagem do sistema.

Outro parâmetro relacionado diretamente ao controlador é o seu valor inicial (θ_0). Se escolhermos um valor inicial muito pequeno, então o sistema pode cometer muitos erros no início de sua execução, fazendo com que o sistema entre em modo proibitivo cedo demais. Note também que os erros cometidos no início do funcionamento tem maior impacto no sistema uma vez que o sistema pode não ter auditado uma quantidade razoável de requisições corretamente para que o erro relativo seja menor.

A Figura 4.4 ilustra o comportamento do sistema de acordo com variações nos hiperparâmetros relacionados diretamente com o controlador. Nos experimentos ilustrados em (a) e (b), o valor mínimo do controlador (θ_{min}) foi relaxado para $-0,25$ e $-0,50$ respectivamente e o comportamento em ambos é bem semelhante com o evidenciado no experimento da Figura 4.3 quando $e_{ref} = 0,1$. Portanto, no cenário onde os bons hiperparâmetros foram escolhidos, esse tipo de relaxamento não produz nenhum efeito.

Já no experimento (c), fixamos o valor máximo do controlador θ_{max} em 1 e com isso, o sistema pode chegar na situação onde nenhum item será aprovado. Neste experimento é possível observar um baixo valor de automação no mês de Abril/2019, porém, essa baixa taxa de automação não produz nenhum efeito de redução de custos quando comparado com outros cenários. Este efeito se deve ao fato de que ao fazer com que o autorizador não aprove nenhuma requisição, estamos deixando passar muitas



(a) $\theta_{min} = -0,25$ (b) $\theta_{min} = -0,5$

(c) $\theta_{max} = 1$ (d) $\theta_0 = 0,9$

Figura 4.4. Resultado dos experimentos variando os hiperparâmetros do controlador (θ).

requisições "simples"e dessa forma, deixando de diminuir o erro relativo.

Por fim, o experimento (d) ilustra o cenário onde o valor inicial do controlador (θ_0) é um valor maior do que o escolhido normalmente ($\theta_0 = 0,5$). Neste cenário é possível observar baixa automação no primeiro mês de execução do sistema, o que é esperado dado o aumento do valor inicial. Entretanto nota-se que o sistema opera de maneira melhor em alguns meses subsequentes em relação à outros experimentos (Junho/2019 e Julho/2019), mas também tem desempenho pior nos meses de Janeiro/2019 e Fevereiro/2019, fazendo com que a escolha desse hiperparâmetro dependa de características do problema.

Com isso, os experimentos mostram que é possível obter um conjunto de hiperparâmetros que faz com que o sistema funcione próximo do esperado e prolongue a boa eficiência do modelo (respondendo positivamente a PP 1). Entretanto, esses valores devem ser cuidadosamente escolhidos via experimentação e testes, uma vez que as peculiaridades de cada caso podem requerer diferentes ajustes do sistema.

4.5 Valores-Padrão dos Hiperparâmetros nos Experimentos

A Tabela 4.5 descreve os valores dos hiperparâmetros utilizados nos experimentos dessa seção para fins de reprodutibilidade e comparação dos resultados.

Tabela 4.3. Lista dos hiperparâmetros utilizados nos experimentos. 1- Oráculo. 2- Taxa Alfa. 3- Custo de Referência. 4- Parâmetros do Controlador

Experimento	1	2	3	4
Intervalo de Atualização	60 Minutos	60 Minutos	60 Minutos	60 Minutos
$F1_{min}$	0,5	0,9	0,9	0,9
e_{ref}	0,0018	0,1	VARIADO	0,1
α	0,01	VARIADO	0,01	0,01
θ_0	0,5	0,5	0,5	0,5 e 0,9
θ_{min}	0	0	0	0,0, -0,25 e -0,50
θ_{max}	0,98	0,98	0,98	0,98 e 1

Capítulo 5

Conclusões e Trabalhos Futuros

Neste trabalho, foi discutido um sistema que utiliza conceitos da teoria de controle para controlar a automação de um modelo de aprendizado de máquina em função do risco financeiro cometido por erros de classificação do modelo. Nosso estudo de caso utilizou dados reais de uma grande empresa de seguros de saúde, onde o sistema já opera em produção.

O primeiro problema atacado pelo trabalho se origina da característica dos processos da seguradora. *Como é possível automatizar um processo de auditoria complexo, onde cada requisição possui diferentes itens com várias diferenças?* Para endereçar essa questão, fora apresentada uma maneira de calibrar diferentes *tresholds* (th) individuais que refletem a qualidade de análise dos itens individualmente. Nesse cálculo, o sistema utiliza a métrica do *F1-Score* para medir o quão proficiente o modelo é em analisar cada item. Esse processo é muito interessante, pois dispensa a criação de múltiplos modelos cujo processo pode ser difícil ou até mesmo inviável.

Em seguida fora apresentado o conceito do controlador (θ), que é uma ferramenta que pode ser utilizada para flexibilizar os *thresholds* supracitados tornando o sistema mais permissivo ou mais proibitivo. Esse tipo de controle é interessante quando consideramos cenários onde o modelo vai degradando ao longo do tempo em função da mudança da distribuição de suas características. Dessa forma, a empresa pode a qualquer momento ajustar o sistema em função da sua situação atual.

Por fim, aplicamos os conceitos do Controlador PID para calibrar automaticamente o controlador em função do gasto financeiro cometido pelos erros do modelo. Esse sistema faz com que o modelo tenha cada vez mais influência no processo em caso de alta acurácia e do contrário, tenha cada vez menos espaço para atuar em cenários onde cometeu muitos erros. Essa estratégia é interessante em ambientes de produção, pois serve como balizador dos impactos da automação.

Nos experimentos, foi possível observar que o valor do erro de referência (e_{ref}) possui um valor mínimo para que o sistema funcione bem. O sistema de controle funcionou bem apenas nos casos onde $e_{ref} \geq 0,1$, e também vimos que para valores $e_{ref} \geq 0,2$ não faria sentido utilizar o controle de risco, uma vez que os resultados foram muito parecidos com o oráculo. Por isso ao implementar o sistema, deve-se analisar a viabilidade do risco mínimo necessário para operação.

Para os outros parâmetros, vimos que a taxa alfa segue exatamente o mesmo princípio dos parâmetros do controlador PID, onde se o valor é baixo demais, o sistema não responde com agilidade. Do contrário, o sistema fica instável oscilando entre os valores θ_{min} e θ_{max}. Os limites do controlador, bem como seu valor de inicialização mostraram-se coerentes com o esperado do funcionamento. No geral, a velocidade com que se deseja atualizar o sistema, e o número de dias da janela de erro são decisões de negócio.

Como trabalhos futuros, seria interessante adicionar os termos integrativos e derivativos no sistema. Esses dois termos fazem com que o erro fique mais estabilizado em relação à referência e podem ser boas alternativas para casos onde deseja ter mais controle sobre o erro. Também seria interessante testar com outra base de dados, em cenários distintos ao caso de estudo, para entender o quão viável é aplicar o sistema em outros ambientes. Expandir o conjunto de dados utilizado atualmente também poderia fornecer novas visões.

Referências Bibliográficas

Bergmeir, C. & Benítez, J. M. (2012). On the use of cross-validation for time series predictor evaluation. *Inf. Sci.*, 191:192--213.

Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1--3.

Cao, J.-Y.; Liang, J. & Cao, B.-G. (2005). Optimization of fractional order pid controllers based on genetic algorithms. Em *2005 international conference on machine learning and cybernetics*, volume 9, pp. 5686--5689. IEEE.

Chen, T. & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. Em Krishnapuram, B.; Shah, M.; Smola, A. J.; Aggarwal, C. C.; Shen, D. & Rastogi, R., editores, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pp. 785--794. ACM.

Cheon, K.; Kim, J.; Hamadache, M. & Lee, D. (2015). On replacing pid controller with deep learning controller for dc motor system. *Journal of Automation and Control Engineering*, 3(6):452--456.

Cohn, D. A.; Ghahramani, Z. & Jordan, M. I. (1994). Active learning with statistical models. Em Tesauro, G.; Touretzky, D. S. & Leen, T. K., editores, *Advances in Neural Information Processing Systems 7, [NIPS Conference, Denver, Colorado, USA, 1994]*, pp. 705--712. MIT Press.

Dequan, S.; Guili, G.; Zhiwei, G. & Peng, X. (2012). Application of expert fuzzy pid method for temperature control of heating furnace. *Procedia Engineering*, 29:257--261.

Doyle, J. C.; Francis, B. A. & Tannenbaum, A. R. (2013). *Feedback control theory*. Courier Corporation.

Grassi, E. & Tsakalis, K. (2000). PID controller tuning by frequency loop-shaping: application to diffusion furnace temperature control. *IEEE Trans. Contr. Sys. Techn.*, 8(5):842--847.

Holzinger, A. (2016). Interactive machine learning for health informatics: when do we need the human-in-the-loop? *Brain Informatics*, 3(2):119--131.

Jun, M. & Safonov, M. G. (1999). Automatic PID tuning: An application of unfalsified control. Em *Proceedings of the 1999 IEEE International Symposium on Computer Aided Control System Design (Cat. No. 99TH8404)*, pp. 328--333. IEEE.

Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q. & Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. Em *Advances in Neural Information Processing Systems*, pp. 3146--3154.

Lakshminarayanan, B.; Pritzel, A. & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. Em Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N. & Garnett, R., editores, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pp. 6402--6413.

Liaw, A.; Wiener, M. et al. (2002). Classification and regression by randomforest. *R news*, 2(3):18--22.

McKendrick, J. (2018 (Acessado no dia 17 de Setembro de 2019)). *How Fast Is Artificial Intelligence Growing?* `https://bit.ly/39A2DRv`.

NITRD (2019). *Supplement to the President's FY2020 Budget*, volume 1. National Science & Technology Council, 1 edição.

Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227--244.

Snoek, J.; Ovadia, Y.; Fertig, E.; Lakshminarayanan, B.; Nowozin, S.; Sculley, D.; Dillon, J. V.; Ren, J. & Nado, Z. (2019). Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift.

Sutton, R. S.; Barto, A. G. et al. (1998). *Introduction to reinforcement learning*, volume 2. MIT press Cambridge.

Watkins, C. J. C. H. & Dayan, P. (1992). Technical note Q-Learning. *Machine Learning*, 8:279--292.

Zhu, X. & Goldberg, A. B. (2009). *Introduction to Semi-Supervised Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.

Ziegler, J. G.; Nichols, N. B. et al. (1942). Optimum settings for automatic controllers. *trans. ASME*, 64(11).