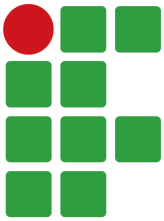


INSTITUTO FEDERAL
Goiás
Câmpus Anápolis

Thauan da Silva Cruz

Automatizando a fiscalização de gastos públicos por meio da classificação automática de empenhos utilizando Aprendizado de Máquina.

Anápolis-GO

2021

Thauan da Silva Cruz

Automatizando a fiscalização de gastos públicos por meio da classificação automática de empenhos utilizando Aprendizado de Máquina.

Projeto de Pesquisa apresentado ao curso de Bacharelado em Ciências da Computação, como requisito para obtenção do grau final na disciplina de Projeto Final de Curso 2.

Instituto Federal de Goiás - Câmpus Anápolis

Orientador: Prof. Dr. Hugo Vinícius Leão e Silva

Coorientador: Prof. Dr. Daniel Xavier de Sousa

Anápolis-GO

2021

Dados Internacionais de Catalogação na Publicação (CIP)

Cruz, Thauan da Silva

C957a Automatizando a fiscalização de gastos públicos por meio da classificação automática de empenhos utilizando aprendizado de máquina./ Thauan da Silva Cruz. – Anápolis: IFG, 2021.
79 f. : il. color.

Orientador: Prof. Dr. Hugo Vinícius Leão e Silva.

Trabalho de Conclusão de Curso – Instituto Federal de Educação, Ciência e Tecnologia de Goiás: Câmpus Anápolis – Bacharelado em Ciências da Computação, 2021.

1. Aprendizado de máquina. 2. classificação. 3. fiscalização orçamentária.
I. Silva, Hugo Vinícius Leão e (orient.). II. Sousa, Daniel Xavier de (coorient.). III. Título.

CDD 004

Ficha catalográfica elaborada pela bibliotecária Claudineia Pereira de Abreu CRB1/1956
IFG Câmpus Anápolis

TERMO DE AUTORIZAÇÃO PARA DISPONIBILIZAÇÃO NO REPOSITÓRIO DIGITAL DO IFG - ReDi IFG

Com base no disposto na Lei Federal nº 9.610/98, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia de Goiás, a disponibilizar gratuitamente o documento no Repositório Digital (ReDi IFG), sem ressarcimento de direitos autorais, conforme permissão assinada abaixo, em formato digital para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IFG.

Identificação da Produção Técnico-Científica

[ ]	Tese	[ ]	Artigo Científico
[ ]	Dissertação	[ ]	Capítulo de Livro
[ ]	Monografia – Especialização	[ ]	Livro
[X]	TCC - Graduação	[ ]	Trabalho Apresentado em Evento
[ ]	Produto Técnico e Educacional - Tipo: _____		

Nome Completo do Autor: Thauan da Silva Cruz
Matrícula: 20171060140105
Título do Trabalho: Automatizando a fiscalização de gastos públicos por meio da classificação automática de empenhos utilizando Aprendizado de Máquina.

Restrições de Acesso ao Documento

Documento confidencial: [X] Não [ ] Sim, justifique: _____

_____

Informe a data que poderá ser disponibilizado no ReDi/IFG:__/__/__
O documento está sujeito a registro de patente? [ ] Sim [X] Não
O documento pode vir a ser publicado como livro? [ ] Sim [X] Não

DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA

O/A referido/a autor/a declara que:

i. o documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade;
ii. obteve autorização de quaisquer materiais inclusos no documento do qual não detém os direitos de autor/a, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia de Goiás os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;
iii. cumpriu quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia de Goiás.

____Anápolis____, 09/09/2021.
Local Data

Thauan da Silva Cruz

_____
Assinatura do Autor e/ou Detentor dos Direitos Autorais

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE GOIÁS
CÂMPUS ANÁPOLIS

ATA DA SESSÃO PÚBLICA DE APRESENTAÇÃO E DEFESA DO TRABALHO DE CONCLUSÃO DE CURSO

Aos **doze** dias do mês de **agosto** de **2021**, às **09h30**, em sessão pública por meio de webconferência via Google Meet, foi realizada a sessão pública de apresentação e qualificação do Trabalho de Conclusão de Curso do Graduando **Thauan da Silva Cruz (matrícula 20171060140105)** do curso de Bacharelado em Ciência da Computação. A banca foi composta pelos seguintes membros: **Dr. Hugo Vinicius Leão e Silva, Dr. Daniel Xavier Sousa, Ms. Maurício Barros de Jesus e Dr. Eduardo Noronha de Andrade Freitas**, sob a presidência do primeiro. O trabalho de conclusão de curso tem como título **Aplicação de Aprendizado de Máquina no Processo de Automatização da Classificação da Natureza de Despesa**, sob orientação de **Hugo Vinicius Leão e Silva** e co-orientação de **Daniel Xavier de Sousa**. Após a apresentação, tendo sido o autor arguido pela Banca Examinadora, a nota obtida foi **9,7** e o Trabalho de Conclusão de Curso foi considerado **APROVADO** pela banca examinadora.

Encerra-se a presente sessão às **11** horas e **23** minutos. Eu, **Hugo Vinicius Leão e Silva** , dato e assino a presente ata que segue assinada por todos os membros da Banca e pelo graduando.

Dr. Hugo Vinicius Leão e Silva
(Assinado Eletronicamente)

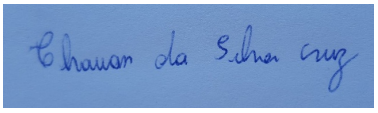
Dr. Daniel Xavier Sousa
(Assinado Eletronicamente)

Dr. Eduardo Noronha de Andrade Freitas
(Assinado Eletronicamente)

MAURICIO BARROS DE JESUS:01047638177

Assinado de forma digital por MAURICIO BARROS DE JESUS:01047638177
Dados: 2021.08.18 10:03:49 -03'00'

Ms. Maurício Barros de Jesus



Thauan da Silva Cruz
(Graduando)

Documento assinado eletronicamente por:

- **Hugo Vinicius Leao e Silva**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 18/08/2021 10:20:54.
- **Sergio Daniel Carvalho Canuto**, **COORDENADOR - FUC1 - ANA-CCSCC**, em 18/08/2021 11:01:08.
- **Daniel Xavier de Sousa**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 18/08/2021 14:48:09.
- **Eduardo Noronha de Andrade Freitas**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 18/08/2021 17:25:12.

Este documento foi emitido pelo SUAP em 13/08/2021. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse https://suap.ifg.edu.br/autenticar-documento/ e forneça os dados abaixo:

Código Verificador: 189952
Código de Autenticação: f73fc86fd8

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

TCE-GO	Tribunal de Contas do Estado de Goiás;
SIOFINET	Sistema de Programação e Execução Orçamentária e Financeira;
IFG	Instituto Federal de Goiás;
TF-IDF	Frequência do termo e o inverso da frequência nos documentos ou *term frequency–inverse document frequency*;
OHE	One-Hot Encoding;
TF	Frequência do termo ou *Term Frequency*;
IDF	Inverso da Frequência nos documentos ou *inverse document frequency*;
SVM	Máquina de Vetores de Suporte ou *Support Vector Machine*;
RF	Florestas Aleatórias ou *Random Forest*;
K-NN	*K-Nearest Neighbors*;
SGD	Gradiente Descendente Estocástico ou *Stochastic gradient descent*;
URLs	Localizador Uniforme de Recursos ou *Uniform Resource Locators*;
TP	Verdadeiro Positivo ou *True Positive*;
FN	Falso Negativo ou *False Negative*;
FP	Falso Positivo ou *False Positive*;
TN	Verdadeiro Negativo ou *True Negative*;
EOF	Execução Orçamentária e Financeira;
CSV	*Comma-Separated Values* ou Valores Separados por Vírgula.

Resumo

A fiscalização orçamentária do Estado de Goiás está a cargo do Tribunal de Contas. Para realizar essa tarefa, quando há a necessidade, especialistas são designados para manualmente avaliar cada empenho. Porém, esse processo é custoso visto a grande quantidade de empenhos mensais. Portanto, faz-se necessária a automatização desse processo de fiscalização de empenhos. Uma estratégia para lidar com esse problema é fazer a análise por amostragem, selecionando apenas um subconjunto dos empenhos para a fiscalização. O problema disso é que muitos empenhos potencialmente errados ficariam de fora dessa análise. Outra estratégia é a de utilizar técnicas tradicionais de programação, porém, seria necessário entender todos os padrões dos dados para se criar um programa para lidar com todos eles. Como o problema proposto não é trivial isso seria inviável. Tendo isso em mente, este trabalho se propôs a utilizar técnicas de aprendizado de máquina nesse processo de fiscalização, que têm a capacidade de entender padrões complexos presentes nos dados. Com isso, foi possível alcançar uma taxa de acerto de 90% segundo a métrica de avaliação Micro F1. Além disso, um objetivo secundário deste trabalho é fornecer a iniciantes dessa área de estudos um guia, apresentando os passos a serem seguidos para a execução de um projeto de aprendizado de máquina e, em complemento a isso, aplicar esses passos em um problema real, através dos dados fornecidos pelo Tribunal de Contas do Estado de Goiás (TCE-GO).

Palavras-chave: Aprendizado de máquina, Classificação, Fiscalização orçamentária.

Abstract

The budget oversight of the State of Goiás is in charge of the Court of Accounts. To accomplish this task, when the need arises, experts are assigned to manually assess each expense. However, this process is costly given the large amount of monthly expenses. Therefore, it is necessary to automate this inspection process. One strategy for dealing with this problem is by performing sampling analysis, selecting only a subset of the expenses. Thus, many potentially wrong expenses would be unconsidered on this analysis. Another strategy is to use traditional programming techniques. However, the understandment of all data patterns is needed to create a program to deal with them. As the proposed problem is not trivial, this would be unfeasible. With this in mind, this work proposes using machine learning techniques on this inspection process, which have the ability to understand complex patterns present in the data. In result, it was possible to achieve a hit rate of 90 % according to the Micro F1 evaluation metric. In addition, this work's secondary objective is to provide beginners in this field of study with a guide, presenting the steps to be followed and applying them to a real problem, through the data provided by the Court of Accounts of the State of Goiás.

Keywords: Machine learning, Classification, Budget Oversight.

Sumário

1 Introdução

Empenho é a etapa da execução das despesas públicas em que o governo reserva o dinheiro que será pago quando um bem for entregue ou um serviço concluído. Isso ajuda o governo a organizar os gastos pelas diferentes áreas do governo, evitando que se gaste mais do que foi planejado (TRANSPARÊNCIA, 2021).

A próxima etapa na execução das despesas públicas é denominada liquidação, que é quando se verifica que o governo recebeu aquilo que comprou. Ou seja, quando se confere que o bem foi entregue corretamente ou que a etapa da obra foi concluída como acordado. Por fim, se estiver tudo certo com as fases anteriores, o governo pode fazer o pagamento, repassando o valor ao vendedor ou prestador de serviço contratado (TRANSPARÊNCIA, 2021).

Essas três etapas de empenho, liquidação e pagamento se referem aos estágios necessários para a execução dos gastos públicos. Para auxiliar na correta execução desses gastos existe o Tribunal de Contas do Estado de Goiás (TCE-GO) ao qual uma de suas competências é realizar a fiscalização financeira e orçamentária do Estado e das entidades da administração direta e indireta (CONTAS DO ESTADO DE GOIÁS, 2020). Esse controle é realizado através do Sistema de Programação e Execução Orçamentária e Financeira (SIOFINET).

Nesse sistema, a classificação de um empenho é feita manualmente pelo órgão que realizou o empenho. Portanto, não há garantia de que esteja correta, seja por erro humano ou na intenção de cometer uma fraude (JESUS et al., 2019). Quando um empenho é colocado em uma categoria de gastos errada, ele pode não ser publicado nos portais de transparência e isso pode ferir a lei, visto que alguns gastos de caráter público devem estar dispostos nesses portais. Ainda, a incorreta classificação de um gasto público pode fazer com que o dinheiro alocado para uma finalidade seja gasto com outra, desorganizando os gastos públicos e escondendo o real gasto.

Contudo, com a quantidade média de 5.112 empenhos gerados todos os meses torna-se difícil realizar a análise manual de todos eles visto a grande quantidade. Além disso, o número de profissionais capazes de realizar essa análise é insuficiente. Dessa forma, faz-se necessária a automatização desse processo de fiscalização.

Um caminho para isso é fazer a análise por amostragem, avaliando apenas um subconjunto dos empenhos. Porém, essa estratégia desconsidera muitos empenhos que podem estar errados. Além disso, mesmo avaliando um volume menor de empenhos, ainda seria necessário alocar uma equipe dedicada a realizar essa tarefa e isso pode ser bem custoso.

Outro caminho é por meio de técnicas de aprendizado de máquina, que permitem analisar os padrões presentes nos documentos de empenho, permitindo aprender a identificar as naturezas de gastos desses empenhos. A vantagem disso é que não é necessária uma equipe para avaliar manualmente os empenhos, pois o modelo criado pode fazer isso automaticamente nos próximos empenhos que forem gerados. Contudo, existem diversos desafios que devem ser levados em consideração para se adquirir um resultado efetivo, como por exemplo a escolha adequada de quais algoritmos, hiperparâmetros e técnicas para fazer o pré-processamento dos dados.

Ainda, este trabalho se propôs não só fazer uso dos algoritmos disponíveis para esse tipo de problema de forma individual, mas também utiliza-los em conjunto na construção de soluções mais complexas com vistas à obtenção de melhores resultados.

Nesse contexto, o objetivo principal deste trabalho é desenvolver um modelo de aprendizado de máquina que automaticamente identifica empenhos públicos que foram classificados em uma natureza de despesa incorreta. Devido à parceria entre o Instituto Federal de Goiás (IFG) e o TCE-GO, foi possível utilizar os dados públicos do SIOFINET, contendo todos os empenhos ocorridos no período de janeiro de 2015 a abril de 2020.

1.1 Justificativa

O processo de verificação de todos os empenhos realizados pela administração pública do Estado de Goiás é inviável devido à grande quantidade de empenhos realizadas mensalmente. Além disso, o TCE-GO não possui um número suficiente de profissionais capacitados para realizar essa tarefa. Nesse contexto, a automatização desse processo é essencial para o auxílio do TCE-GO na tarefa de fiscalização dos gastos públicos, uma tarefa que traz benefícios tanto a sociedade quanto ao governo através da prevenção da ocorrência de fraudes e possíveis omissões de gastos públicos. Tendo isso em mente, este trabalho tem o objetivo de realizar essa tarefa de automatização fazendo uso de técnicas de aprendizado de máquina.

Para isso, foram utilizados diversos algoritmos estado-da-arte no problema abordado. Além da utilização desses algoritmos de forma individual, este trabalho também se propôs a fazer uso deles em conjunto, através de estratégias mais robustas de construção de modelos de aprendizado de máquina. Adicionalmente, foram construídos dois modelos de aprendizado de máquina independentes onde o primeiro realiza a classificação da natureza de despesa, ou seja, prediz qual natureza de despesa aquele empenho deve ser pertencer. Enquanto o segundo utiliza um conjunto de empenhos previamente validados por um especialista dos dados na criação de um modelo que faz predições quanto a corretude de cada empenho, ou seja, o quão corretos aqueles empenhos estão. Esses dois modelos foram criados para serem utilizados em conjunto, na intenção de aprimorar a confiabilidade da

resposta obtida pela solução criada.

1.2 Objetivos

O objetivo principal deste trabalho é desenvolver um modelo de aprendizado de máquina que automaticamente identifique as corretas naturezas de despesas dos empenhos públicos a fim de auxiliar o TCE-GO na tarefa de fiscalização orçamentária do Estado. Além disso, tem como objetivo secundário fornecer um passo a passo de como executar um projeto de aprendizado de máquina.

1.2.1 Objetivos específicos

O desenvolvimento deste trabalho fez uso de uma metodologia de projetos de aprendizado de máquina, que levou à execução dos seguintes passos:

1. Fazer a aplicação e a avaliação dos algoritmos *Support Vector Machine* (SVM), *Random Forest* (RF), *K-Nearest Neighbors* (*K*-NN) e *Stochastic Gradient Descent* (SGD) de forma individual utilizando a base de dados fornecida pelo TCE-GO, no que se refere às medidas de acurácia utilizadas na literatura da área *F1 Score* (SCIKIT-LEARN, 2021b);

2. Realizar um ajuste fino dos hiperparâmetros dos algoritmos avaliados com o objetivo de aprimorar a acurácia dos modelos;

3. Utilizar estratégias que fazem uso de mais de um algoritmo por vez na intenção de capturar diferentes aspectos dos dados;

4. Fornecer um material que sirva de guia para auxiliar iniciantes da área no desenvolvimento de projetos de aprendizado de máquina.

1.3 Estrutura do Trabalho

Este trabalho está estruturado da seguinte forma: o Capítulo 2 mostra a fundamentação teórica. Nesse sentido, esse capítulo apresenta maneiras de se aplicarem técnicas de aprendizado de máquina no problema de classificação de naturezas de despesa. Para isso, o Capítulo 2 explica as representações de dados utilizadas para permitir o processamento de dados textuais, assim como os algoritmos de aprendizado de máquina considerados. Finalmente, avaliam-se também diferentes estratégias de como utilizar esses algoritmos para permitir a criação de modelos mais robustos de aprendizado de máquina.

O Capítulo 3 descreve a metodologia utilizada na realização deste trabalho. Esse capítulo apresenta um *checklist* que pode ser aplicado a problemas de aprendizado de máquina e, também, apresenta a aplicação desse *checklist* ao problema abordado.

O Capítulo 4 descreve as métricas de avaliação utilizadas no trabalho, assim como os resultados dos experimentos realizados. Por fim, o Capítulo 5 apresenta as conclusões.

2 Fundamentação teórica

O Tribunal de Contas do Estado de Goiás possui como uma de suas competências a fiscalização de gastos efetuados por órgãos públicos. Para isso, é necessário que cada órgão informe com detalhes como seus gastos foram executados. De forma geral, isso ocorre quando os gestores desses órgãos preenchem no SIOFINET as informações detalhadas a respeito dos empenhos realizados.

Contudo, pode haver erros no registro de algumas informações. Como é grande a quantidade mensal de empenhos gerados pela administração pública, fiscalizar todos eles é inviável, especialmente pela falta de pessoal com capacidade técnica necessária para executar essa tarefa.

Diante desse problema, este trabalho propõe o desenvolvimento de um modelo de aprendizado de máquina para automatizar a verificação de empenhos públicos a fim de evitar a classificação errada de sua natureza de despesa e a consequente omissão daquele gasto nos portais de transparência.

Essa tarefa de verificação já foi abordada em (JESUS et al., 2019), porém considerando apenas os gastos de publicidade e propaganda. A proposta deste trabalho é expandir esse escopo para as demais categorias de gastos públicos, que totalizam cerca de 600 categorias distintas.

Por utilizar técnicas de aprendizado de máquina, este trabalho se baseia extensivamente no *checklist* disposto em (GÉRON, 2019). Esse *checklist* é composto por oito passos que compreendem desde o entendimento do problema até a entrega da solução criada.

Para isso, serão utilizados os dados públicos disponíveis no SIOFINET, composto por cerca de 324 mil empenhos de diversas naturezas de despesas ocorridas entre os anos de 2015 e 2020.

2.1 Aprendizado de Máquina

A fiscalização orçamentária do Estado de Goiás, quando necessária, é feita de forma manual, requerendo um profissional qualificado para ler e tentar identificar se o empenho sob análise possui alguma inconsistência. Um ponto avaliado é verificar se a natureza de despesa daquele empenho foi cadastrada corretamente, pois o preenchimento incorreto desse campo pode levar ao descumprimento da lei. Como existe um grande volume de empenhos a serem avaliados e há uma quantidade limitada de recursos humanos qualificados para essa tarefa, há a necessidade de uma alternativa para automatizar esse processo de fiscalização.

Ao utilizar técnicas tradicionais de programação, seria necessário analisar os dados e observar se existem padrões ou regras pré-definidas nos documentos que ajudem a separar os empenhos corretos dos incorretos. Para cada padrão identificado, escreve-se um algoritmo para detectá-lo. Após isso, é necessário testar esse programa para verificar a sua eficácia. Assim, esse processo é repetido até que se consiga um resultado satisfatório. Como esse problema não é trivial e não existem regras pré-definidas para tal tarefa, esse programa provavelmente se tornaria uma longa lista de regras complexas e muito difíceis de se manter (GÉRON, 2019).

Por outro lado, técnicas de aprendizado de máquina aprendem diversos padrões simples, complexos e, às vezes, até humanamente indetectáveis de forma automática. O programa é bem menor, fácil de manter e provavelmente mais preciso que técnicas tradicionais de programação (GÉRON, 2019). Por esse motivo, este trabalho utilizará essas técnicas para automatizar a tarefa de fiscalização orçamentária do Estado.

Aprendizado de Máquina é uma subárea da inteligência artificial que refere-se a habilidade de uma máquina de melhorar seu desempenho baseado em resultados anteriores. Em outras palavras é fazer o computador aprender com base na sua experiência e modificar suas ações com base nesse conhecimento. Em contra partida, na computação convencional o computador não muda suas ações a não ser que um programador explicitamente as mude (YAO; LIU, 2014). Nesse sentido, Aprendizado de Máquina pode utilizar os dados disponíveis para um determinado problema aprendendo seus padrões e realizando predições de novos dados baseados nesses padrões aprendidos.

2.2 Representação dos dados

A representação dos dados é um ponto muito importante a ser avaliado, pois, através dela é possível expor as características dos dados e facilitar o entendimento dos padrões pelos algoritmos. O tipo de representação depende do tipo do dado-alvo. Contudo, é importante ressaltar que existem recomendações da literatura na aplicação dessas representações para certos tipos de dados. Dentre elas, as mais frequentemente utilizadas são o *One-Hot Encoding* (OHE) e o *Term Frequency-Inverse Document Frequency* (TF-IDF).

2.2.1 *One-Hot Encoding*

A maioria dos algoritmos de aprendizado de máquina não tratam de forma direta atributos categóricos, que são aqueles formados por uma lista definida e geralmente pequena de atributos textuais. Portanto, é necessário fazer a transformação de atributos desse tipo para o tipo numérico. Uma maneira para lidar com isso é fazer a conversão das categorias em números, sendo a primeira categoria representada pelo número um, a

segunda pelo número dois e assim por diante. O problema dessa maneira é que algoritmos de aprendizado de máquina podem identificar que a primeira categoria, representada pelo número um, é parecida ou próxima da segunda categoria pois o número um é próximo do número dois, mesmo que sejam categorias completamente diferentes (GÉRON, 2019).

Tendo isso em mente, uma alternativa é a utilização do *One-Hot Encoding* (OHE), que transforma as diferentes categorias presentes em um atributo em novas colunas. Em outras palavras, cada categoria se transforma em um novo atributo binário criado para representar se aquele documento possui ou não aquela categoria. Dessa forma, não existe mais a possibilidade do algoritmo comparar as categorias entre si, visto que elas foram transformadas em atributos independentes. A desvantagem dessa representação é que a dimensionalidade dela pode se tornar muito grande, dependendo da quantidade de categorias de cada atributo. A Figura 1 mostra um exemplo dessa representação.

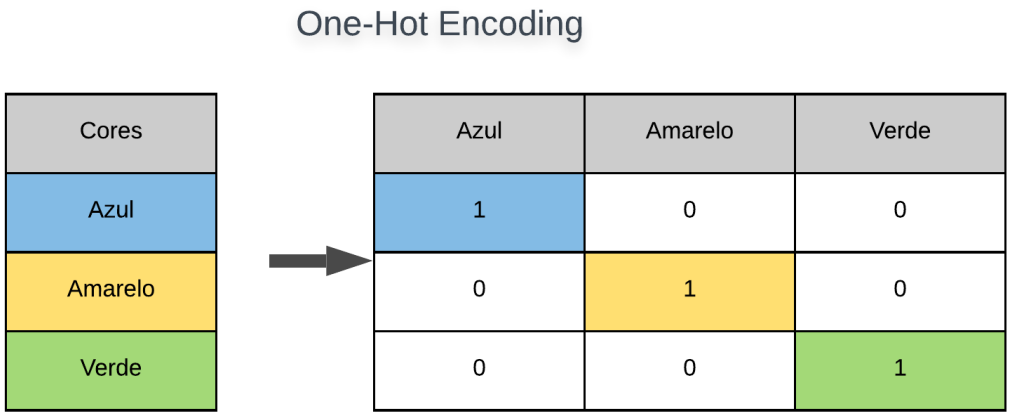


Figura 1 – Exemplo da aplicação da representação *One-Hot Encoding* para a coluna "Cores".

2.2.2 *Term Frequency-Inverse Document Frequency* (TF-IDF)

A representação OHE consegue lidar com atributos categóricos. Porém, se o atributo for de texto livre, deve-se preferencialmente escolher outras representações que explorem melhor a natureza desse tipo de dado. Dentre elas, uma das mais utilizadas é o TF-IDF. Ela separa as palavras do texto e atribui um valor a elas de acordo com sua capacidade discriminativa, observando a frequência com que essa palavra aparece (*"term frequency"* – TF) e em quantos documentos ela aparece (*"inverse document frequency"* – IDF).

O valor *term frequency* nos diz quantas vezes uma palavra aparece no texto, mas para essa medida ter um significado mais completo, é preciso observar também em quantos documentos essa palavra aparece, cujo valor é representado pelo *inverse document frequency*. Se uma palavra aparece muito frequentemente numa pequena quantidade de documentos,

ela tem um grande poder discriminativo e ajuda a separar esses documentos dos demais. Porém, se uma palavra aparece com bastante frequência mas em todos os documentos, ela perde o seu poder discriminativo. Por essa razão, a representação TF-IDF utiliza a combinação desses dois valores para quantificar o poder de discriminação de uma palavra utilizando a fórmula apresentada na Eq. (2.1).

$$W_{(i,j)} = T_{(i,j)} \times \log\left(\frac{N}{D_i}\right), \tag{2.1}$$

onde $W_{(i,j)}$ é o valor de TF-IDF calculado para a i-ésima palavra no j-ésimo documento; $T_{(i,j)}$ é a frequência dessa palavra no documento sob consideração, D_i é o número de documentos contendo essa palavra e N é o número total de documentos.

Então a representação TF-IDF primeiramente seleciona todas as palavras do *corpus*, o conjunto de todos os textos do nosso conjunto de dados, e utiliza a Eq. (2.1) para calcular um valor para cada palavra de cada documento. Isso tem como resultado uma matriz, onde as colunas são as palavras diferentes encontradas no *corpus*, e cada linha representa o texto de cada documento.

O conjunto de dados, ou também conhecido como *dataset* gerado a partir dessa representação geralmente possui algumas características, como a grande quantidade de colunas, devido à variedade de palavras encontradas nos textos. Ainda, os dados geralmente são esparsos, pois como as colunas são formadas por todas as palavras do *corpus*, dificilmente os documentos terão valores não-nulos para todas as colunas. A Figura 2 mostra um exemplo dessa representação.

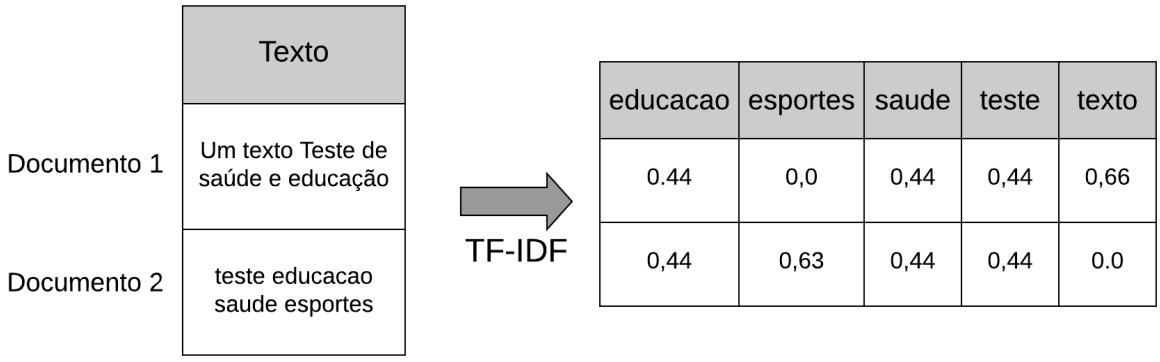


Figura 2 – Exemplo da aplicação da representação TF-IDF da biblioteca Scikit-learn no campo “Texto” de dois documentos.

2.3 Algoritmos

Pôde-se observar que as representações de dados apresentadas na Seção 2.2 permitem gerar um conjunto de novos atributos à partir de dados textuais, sejam eles categóricos ou

texto livre. Em outras palavras, essas representações de dados *reparametrizam* os atributos baseados em texto em um conjunto de valores numéricos que permitem que algoritmos de aprendizado de máquina possam aprender padrões. Esta seção traz os algoritmos preditores avaliados neste Trabalho de Conclusão de Curso na automatização da tarefa de classificação de empenhos públicos.

2.3.1 *Support Vector Machine* – SVM

O algoritmo *Support Vector Machine* (SVM) é um modelo de aprendizado de máquina poderoso e versátil, capaz de realizar diversas tarefas, dentre elas: classificação linear e não-linear, regressão e detecção de *outliers* (GÉRON, 2019).

É um dos modelos mais populares em aprendizado de máquina (GÉRON, 2019). Alguns dos motivos para isso é o fato dele ser particularmente apropriado para classificação em *datasets* complexos e pequenos, se manter efetivo mesmo quando o número de atributos dos dados é maior que o número de documentos, ser eficiente no uso de memória e possuir grande versatilidade. Essa versatilidade ocorre com o uso de *kernels*, que são funções responsáveis por representar o *dataset* usando uma dimensão superior, com o intuito de evidenciar uma separação dos dados. Além disso, é possível criar novos *kernels* ou usar um *kernel* apropriado já existente de acordo com o problema a ser tratado (SCIKIT-LEARN, 2021i).

A partir do *kernel*, inicialmente o SVM cria uma nova dimensão utilizando as dimensões já existentes. Então, ele constrói um ou mais hiper-planos a fim de separar os documentos em suas respectivas classes. Após isso, o SVM classifica um novo documento tendo como referência o lado do hiper-plano em que ele se encontra. Intuitivamente, uma boa separação alcançada pelo hiper-plano é aquela com a maior distância entre os pontos de treino mais próximos de cada classe (chamado de margem funcional), visto que, em geral, quanto maior for a margem menor será o erro de generalização do classificador. A Figura 3 mostra um exemplo de como a separação de classes pode ser realizada pelo SVM.

Pode-se observar na Figura 3 que existem duas classes diferentes e representadas respectivamente por um círculo e por um quadrado. Ainda, o hiper-plano é representado por uma linha sólida. Além disso, observa-se que três documentos, dois da primeira classe e um da segunda classe estão situados nos limites da margem funcional, que estão, por sua vez, representados por duas linhas tracejadas. Esses pontos nos limites da margem também são chamados de "*support vectors*" ou vetores de suporte.

2.3.2 *Random Forest* – RF

O algoritmo *Random Forest* (RF) tem um funcionamento simples. Suponha que uma pergunta complexa seja feita a mil pessoas aleatórias. Depois, as suas respostas

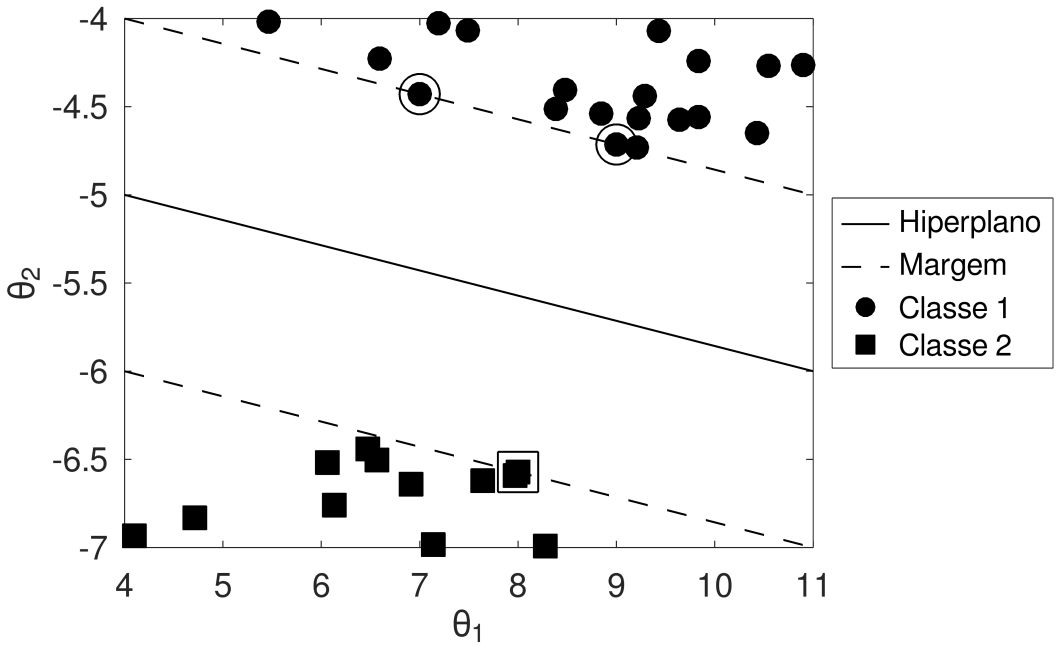


Figura 3 – Exemplo da aplicação do SVM na classificação de documentos de duas classes.

são agregadas. Em muitos casos, a resposta agregada é melhor do que a resposta de um *expert*. Isso se chama *wisdom of the crowd* ou, em outras palavras, conhecimento coletivo. Similarmente, se forem agregadas as predições de um grupo de algoritmos, provavelmente serão encontrados resultados ainda mais aprimorados do que se fossem utilizados individualmente os melhores algoritmos de aprendizado de máquina. Assim, pode-se construir um *ensemble* ao utilizar um grupo de algoritmos "*weak learners*" para produzir um modelo "*strong learner*". "*Weak learner*" é, nesse contexto, um algoritmo cuja acurácia está próxima ao de uma escolha aleatória (GÉRON, 2019).

No caso do Random Forest (BREIMAN, 2001), os "*weak-learners*" são representados por árvores de decisão, que são algoritmos que organizam os dados em uma estrutura de dados "árvore" de forma com que, percorrendo de sua raiz até a folha, obtém-se um resultado. Durante o treinamento, aplica-se a estratégia de *bagging*, na qual se constroem diferentes subconjuntos de treinamento para cada árvore, sendo um mecanismo para aperfeiçoar a diversidade dos *weak learners*. Essa construção é realizada pela seleção aleatória de instâncias e atributos para o treinamento de cada árvore. No caso das instâncias, a seleção aceita repetições em sua amostragem. A aplicação da estratégia *bagging* ao algoritmo RF faz com que cada árvore da floresta tenha acesso a uma sub região do universo de dados de treino.

Na fase de predição dos modelos RFs, os documentos percorrem as diversas árvores, da raiz até a folha, obtendo-se um rótulo ou um valor para determinado documento. Com cada árvore indicando uma classe específica, é feito um comitê para pontuar a classe mais indicada. O algoritmo *Random Forest* tem mostrado resultados bem efetivos, se mostrando

bastante consistente em *datasets* grandes e pequenos. (WANG et al., 2019).

2.3.3 K-*Nearest Neighbors* – K-NN

O algoritmo *K-Nearest Neighbors* (*K*-NN) implementa o aprendizado baseado em instâncias. Isso significa que a classificação de um documento cuja classe é desconhecida é realizada a partir da comparação desse documento com os dados de treino, sem necessariamente a construção de um modelo. O princípio usado nesse algoritmo é o de apresentar um novo documento e compará-lo aos documentos previamente armazenados e classificados. Esse estilo de processamento é conhecido como "avaliação preguiçosa" já que não há um trabalho prévio de treinamento de um modelo (SILVA; PERES; BOSCARIOLI, 2017).

A lógica que implementa esse algoritmo é simples. Um documento é classificado pela "votação da maioria", realizada junto aos seus vizinhos nos dados de treino. A classe da maioria dos K documentos mais próximos ao documento de teste é aquela que deve ser atribuída a ele. A relação de proximidade para identificação dos vizinhos é quantificada por uma métrica de distância pré-definida. Porém, vale ressaltar que diferentes métricas podem gerar diferentes resultados e a métrica a ser usada deve ser adequada ao tipo dos dados sob análise.

Um exemplo de métrica de proximidade e também a mais comumente utilizada é a distância euclidiana. Ela possui a propriedade de representar a distância física entre pontos em um espaço d-dimensional (SILVA; PERES; BOSCARIOLI, 2017). Na Figura 4, a representação da distância euclidiana se manifesta pela linha contínua entre os pontos i e j. De maneira genérica, a distância euclidiana entre esses dois pontos é calculada pela Eq. (2.2).

$$d_{i,j} = \sqrt{\sum_{k=1}^{n} (i_k - j_k)^2}, \tag{2.2}$$

onde $d_{i,j}$ é a distância d entre os pontos i e j. Ainda, i_k e j_k são os valores da variável k para os pontos i e j respectivamente.

2.3.4 Gradiente Descendente Estocástico – SGD

O método do Gradiente Descendente é um algoritmo de otimização capaz de encontrar soluções ótimas para um grande conjunto de problemas. A ideia geral desse algoritmo é ajustar o vetor de parâmetros $\dot{\boldsymbol{\theta}} = [\theta_1, \theta_2, \ldots, \theta_K]$ iterativamente a fim de minimizar uma determinada função $f(\dot{\boldsymbol{\theta}})$ de K dimensões (GÉRON, 2019).

Uma analogia sobre o funcionamento do método do Gradiente Descendente é a seguinte: considere que você está perdido em uma montanha e sob uma forte neblina. Você consegue sentir apenas o declive do chão sob seus pés. Uma boa estratégia para descer

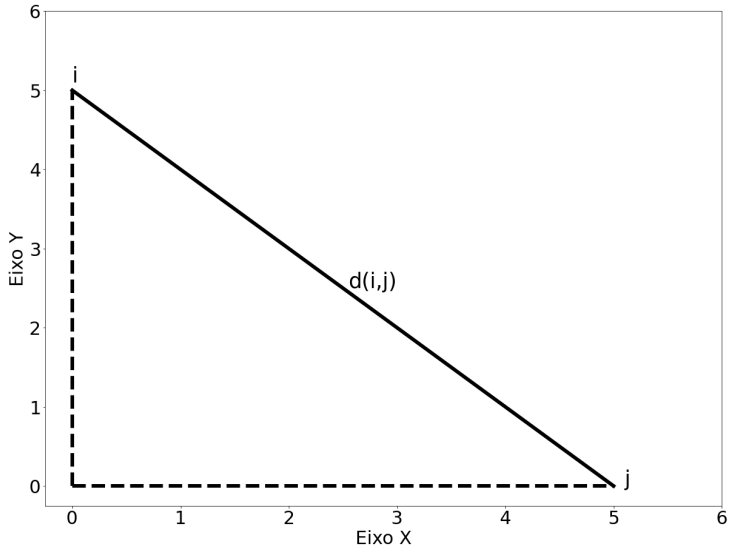


Figura 4 – Distância euclidiana $d(i, j)$ entre os pontos i e j.

essa montanha rapidamente é ir na direção em que a montanha mais se inclina para baixo (GÉRON, 2019).

Nesse sentido, o método do Gradiente Descendente realiza os seguintes passos:

1. Calcula o gradiente da função no ponto inicial $\dot{\boldsymbol{\theta}}_0$, produzindo $\nabla f(\dot{\boldsymbol{\theta}}_0)$. Esse vetor indica a direção de máximo aclive da função nesse ponto;

2. Corrige $\dot{\boldsymbol{\theta}}_0$ utilizando a equação $\dot{\boldsymbol{\theta}}_0 = \dot{\boldsymbol{\theta}}_0 - \eta \nabla f(\dot{\boldsymbol{\theta}}_0)$, onde η é um fator de ajuste do passo a ser dado na correção. Utiliza-se o passo $-\eta \nabla f(\dot{\boldsymbol{\theta}}_0)$, pois deseja-se minimizar $f(\dot{\boldsymbol{\theta}})$;

À medida em que o método do Gradiente Descendente se aproxima de um ponto mínimo, $\nabla f(\dot{\boldsymbol{\theta}}_0) \to \mathbf{0}$ e, consequentemente, os passos vão ficando cada vez menores. Assim, pode-se repetir os passos listados acima até atingir o critério de tolerância ou número máximo de iterações previamente estabelecido.

Esse algoritmo também pode ser aplicado em problemas de aprendizado de máquina, já que problemas de ajuste de curvas podem ser tratados como problemas de minimização. Nesse sentido, deseja-se encaixar uma reta a um conjunto de pontos de treino no plano cartesiano ajustando a sua inclinação e a sua altura. Para isso, utiliza-se uma função de perda, *loss function*, que calcula o resíduo ou, em outras palavras, a distância entre aqueles pontos e a reta considerada.

Sendo assim, o método do Gradiente Descendente pode ser aplicado em problemas de aprendizado de máquina ao definir o vetor de parâmetros da função de perda de forma a conter a inclinação e a altura de uma reta, além dos pontos de treino no plano cartesiano. A partir disso, inicializa-se o processo de otimização assumindo valores aleatórios para os

parâmetros da reta. Esse processo deve ser realizado até que se atinja a convergência ou atingir um número máximo de iterações.

Adicionalmente, existe o método do Gradiente Descendente Estocástico ou *Stochastic gradient descent* (SGD). Em vez de utilizar todo o *dataset* de treino no cálculo da função de perda e do respectivo gradiente, esse método utiliza apenas uma amostra aleatória desse *dataset* a cada iteração para diminuir o esforço computacional.

2.4 Estratégias

A Seção 2.3 apresentou alguns algoritmos utilizados para aprendizado de máquina. Esta seção apresenta algumas estratégias para usar esses algoritmos.

A primeira estratégia é utilizar qualquer um daqueles algoritmos de maneira isolada, podendo gerar bons resultados. Contudo, existem outras estratégias para lidar com os mais diversos tipos de problemas de aprendizado de máquina. Nesse sentido, é importante avaliar o problema analisado em função do uso de mais de um algoritmo em conjunto, com vistas à obtenção de melhores resultados.

2.4.1 Stacking

A estratégia *Stacking* é composta de duas etapas. A primeira delas consiste em treinar um ou mais algoritmos de aprendizado de máquina utilizando o mesmo conjunto de dados de treino. Após isso, a predição de cada um desses modelos nos dados de teste é coletada e são anexadas para criar um *outro dataset*. Nesse sentido considerando o uso literal do *Stacking*, é importante ressaltar que os atributos do *dataset* original não compõem esse novo *dataset*.

Logo em seguida, esse novo *dataset* é utilizado como parâmetro de entrada de outro algoritmo de aprendizado, conhecido como algoritmo de meta-aprendizado, a fim de retornar o resultado final do modelo (GRACZYK et al., 2010). A Figura 5 exemplifica o funcionamento dessa estratégia.

2.4.2 Oráculo

A estratégia Oráculo proposta neste trabalho tem o objetivo de agrupar os pontos fortes de diversos algoritmos. A ideia inicial é indicar para cada documento qual é potencialmente o melhor conjunto de algoritmos para fazer a sua predição. Em seguida, cada documento é avaliado sob seu conjunto de algoritmos e sua classe é definida através de um comitê. O maior objetivo da aplicação dessa estratégia, conhecida como meta-aprendizado, é tratar o alto desbalanceamento existente no *dataset* através do uso mais apropriado de algoritmos na predição de cada documento. Dessa forma, cada documento

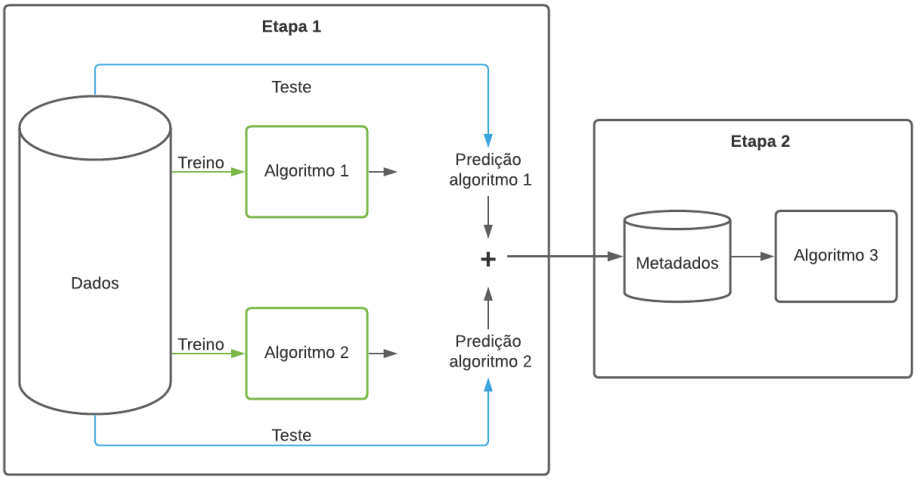


Figura 5 – Funcionamento da estratégia *Stacking*.

terá potencialmente o melhor algoritmo indicado para realizar a sua predição, fazendo com que obtenha-se melhores resultados para os diferentes tipos de classes presente nos dados.

Inicialmente, os dados são separados em dois conjuntos, treino e teste. Para a primeira parte dessa estratégia, utiliza-se o *cross-validation*, uma estratégia que divide os dados em K partes iguais e utiliza uma dessas partes como teste e as outras $K-1$ restantes como treino. A cada iteração, uma parte diferente é selecionada como teste e, por consequência, as demais como treino. Esse processo é repetido K vezes, obtendo-se assim K resultados em vez de apenas um (SCIKIT-LEARN, 2021a). A Figura 6 mostra um exemplo dessa estratégia considerando $K = 3$, onde se observa que o *dataset* é dividido em três partes iguais, em que o conjunto de teste assume um pedaço diferente em cada iteração do *cross-validation*.

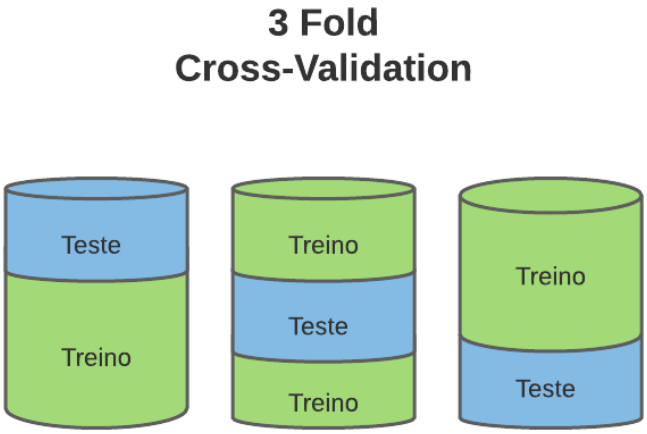


Figura 6 – Exemplo da estratégia *cross-validation* para um valor de $K = 3$.

Nesse sentido, o Oráculo aplica o *cross-validation* nos dados de treino de forma que, em cada iteração, ele avalia vários algoritmos. Dessa forma, ao fim das K execuções, cada documento do *dataset* de treino terá as predições dos diversos algoritmos avaliados. Uma desvantagem dessa estratégia é o alto custo computacional gerado tanto por conta das múltiplas execuções por parte do *cross-validation*, quando pelo uso de diversos algoritmos em conjunto.

Para cada documento de treino admitem-se apenas os algoritmos que acertaram a sua predição. A Figura 7 exemplifica essa etapa da estratégia Oráculo, onde se observa a inclusão de uma nova coluna ao *dataset*, contendo o conjunto de algoritmos para cada documento.

Figura 7 – Exemplo da primeira etapa da estratégia Oráculo.

O conjunto de algoritmos selecionados para cada documento pode admitir um ou mais algoritmos. Porém, essa escolha é tratada como um hiperparâmetro de tal forma que existem apenas duas opções. A primeira delas chamada de "Oráculo Singular" faz uso de maneira aleatória de um dos algoritmos selecionados previamente, enquanto a outra opção chamada de "Oráculo Múltiplo" utiliza todos eles. Por outro lado, vale ressaltar que há a possibilidade de que nenhum algoritmo seja selecionado para um documento em específico. Nesse caso, há dois caminhos distintos: ou seleciona-se um algoritmo aleatório ou selecionam-se todos os algoritmos avaliados.

Em seguida, é selecionado um outro algoritmo de aprendizado de máquina para treinar um novo modelo usando o conjunto de dados de treino. Porém, o rótulo a ser predito passa a ser a coluna de algoritmos selecionados. Após o treinamento, a predição é feita utilizando o conjunto de dados de teste que foi separado inicialmente. Após a predição, o conjunto de testes passa a ter um conjunto de algoritmos atrelados a cada um dos seus documentos.

Então, para cada documento, o conjunto de algoritmos selecionados realizarão a predição do rótulo original a qual ele pertence. Caso exista apenas um algoritmo para cada documento, a predição desse algoritmo será o rótulo da classificação. Já no caso em que houver mais de um algoritmo por documento torna-se necessário determinar qual das predições será escolhida. Quando atinge-se a concordância, ou seja, quando a maioria dos algoritmos predizem o mesmo rótulo para um determinado documento, esse é o rótulo selecionado. Entretanto, quando há divergência nas predições, o rótulo selecionado é dado por aquele com o maior valor de probabilidade determinado por cada algoritmo.

3 Metodologia

3.1 Fluxo de trabalho geral para aplicações de aprendizado de máquina

A metodologia a ser utilizada neste trabalho é a definida no *checklist* de projetos de aprendizado de máquina proposto em (GÉRON, 2019), que possui os seguinte passos:

1. Entender o problema e ter uma visão do todo;
2. Adquirir os dados;
3. Explorar os dados para obter ideias;
4. Preparar os dados;
5. Explorar diferentes modelos de aprendizado de máquina;
6. Refinar os modelos;
7. Apresentar sua solução;
8. Lançar, monitorar e manter o sistema.

A primeira etapa – *entender o problema e ter uma visão do todo* – tem como objetivo definir de forma clara o problema proposto, entender como a solução a ser desenvolvida será utilizada, qual a performance esperada dessa solução, dentre outras informações. Portanto, essa etapa exige diversas reuniões com os interessados para o melhor entendimento das necessidades do problema.

A próxima etapa é *Adquirir os dados*. Ela tem como objetivo listar os dados necessários para a criação da solução, assim como, analisar quantos documentos fazem parte do *dataset* e o quanto ele ocupará de espaço de armazenamento. Por fim, deve-se fazer de fato a aquisição dos dados e a conversão deles para um formato que seja de fácil manipulação pelos algoritmos de aprendizado de máquina.

A etapa seguinte, *explorar os dados para obter ideias*, tem o objetivo de aprofundar o entendimento sobre os dados e, para cada atributo, observar o seu nome, tipo, valores faltantes, distribuição dos valores, correlação entre atributos, entre outras características. Isso permite ter uma visão geral sobre cada atributo. Porém, isso por si só não é o suficiente, pois é necessário compreender esses atributos no contexto geral do negócio para auxiliar a tomada de decisões nas próximas etapas.

A etapa acima provê informações importantes sobre os atributos. Essas informações são então utilizadas na etapa denominada *preparar os dados*. Essas informações auxiliam desde tornar possível a identificação de atributos irrelevantes para o problema, assim como criar novos atributos que explicitam algumas características encontradas nos dados. Além disso, é possível escolher qual representação de dados serão utilizadas em cada um dos atributos, tendo em vista que o tipo de dado de cada um deles já foi fornecido na etapa anterior. Ainda, cada atributo deverá ter todos os seus valores faltantes preenchidos e, se o problema exigir, ter os seus *outliers* removidos. Nessa etapa, os atributos julgados como desnecessários são removidos e também são criados novos atributos. É importante ressaltar que, ao final desse processo, todos os atributos são numéricos e, para maior efetividade dos algoritmos de aprendizado de máquina, eles podem ser normalizados, ou seja, transformados de forma com que todos fiquem no mesmo intervalo de valores.

O objetivo da próxima etapa – *explorar diferentes modelos* – é testar diferentes modelos de aprendizagem de máquina, a fim de encontrar o que consegue explorar melhor as especificidades da base de dados. Nesse sentido, deve-se avaliar os algoritmos de maneira individual, assim como usá-los em conjunto, como ocorre nas estratégias apresentadas na Seção 2.4. Vale ressaltar que são escolhidos algoritmos de destaque que consideram distintas representações dos dados nesse processo de avaliação. A etapa *refinar os modelos* se refere a realizar ajustes finos nos hiperparâmetros para os algoritmos avaliados com o intuito de aprimorar os resultados obtidos.

Finalmente, as últimas etapas são *apresentar a sua solução* e *lançar, monitorar e manter o sistema*. Elas possuem o objetivo de documentar e apresentar tudo o que foi desenvolvido, explicar a sua solução e como ela atinge o objetivo de negócio e de fato lançar e monitorar o programa desenvolvido. Durante o funcionamento do sistema, é importante ressaltar que deve-se avaliar periodicamente a eficácia da solução na inferência dos dados a fim de verificar se o desempenho ainda é adequado. Isso decorre do fato de que ele pode diminuir à medida em que se tornam desatualizados os dados usados na criação do modelo de aprendizado de máquina. Caso isso ocorra, deve-se considerar a criação de um novo modelo, de forma que inclua o novo volume de dados.

3.2 Aplicação do fluxo de trabalho nos dados do TCE-GO

Considerando a primeira etapa do fluxo de trabalho, foram realizadas reuniões e conversas nas quais foi discutido qual era o problema a ser resolvido. Foi então identificada a necessidade da automatização da verificação da natureza de despesa dos empenhos públicos. Isso se justifica, pois atualmente esse é um processo manual e como tal é sujeito a falhas. Porém, a incorreta classificação de um empenho pode ferir a lei, no sentido de que alguns gastos de interesse público devem ser dispostos em portais de transparência. Sendo assim,

a classificação errada pode omitir esses gastos, ferindo a Lei n.º 12.527, de 18 de novembro de 2011 (REPÚBLICA, 2020). Portanto, a automatização desse processo de fiscalização é benéfica, pois, assim, em sua maioria, os empenhos incorretamente classificados poderão ser identificados e corrigidos.

A partir disso, o TCE-GO forneceu dados extraídos do SIOFINET, que se constituem dos empenhos realizados no Estado de Goiás durante o período que compreende de janeiro de 2015 a abril de 2020. Esse *dataset* é composto por 324.728 empenhos, cada um com 44 atributos, e disponibilizados no formato de uma planilha do Microsoft Excel. Além disso, o TCE-GO encaminhou o dicionário dos dados contendo informações relativas ao conteúdo de cada atributo para auxiliar no entendimento deles, cuja versão simplificada é ilustrada na Tabela 1.

Tabela 1 – Dicionário de dados considerando os 44 atributos para cada empenho.

N.	**Atributo**	**Descrição**
1	Exercício do orçamento (Ano)	Esse campo é do tipo numérico e representa o valor do ano do exercício orçamentário no formato "AAAA".
2	Órgão (Código/Nome)	Esse campo é composto pelo código e nome da unidade orçamentária do Órgão.
3	Órgão Sucessor Atual (Código/Nome)	Órgão sucessor atual representa o órgão sucessor mais recente de uma unidade, ou seja, a que está vigente no ano atual. Este campo é composto pelo código e nome do órgão sucessor atual.
4	Tipo Administração (Nome)	Esse campo representa o nome dos tipos de administração pública da unidade orçamentaria que podem ser direta ou indireta.
5	Tipo Poder (Nome)	Esse campo representa o nome do poder público. Poder público representa o conjunto dos órgãos com autoridade para realizar os trabalhos do Estado, são três os tipos de poder: o Poder Legislativo, Poder Executivo e Poder Judiciário.

N.	Atributo	Descrição
6	Classificação orçamentária (Descrição)	Representa a classificação da despesa orçamentária segundo a sua natureza. Ela é identificada por um conjunto de códigos, a seguir indicados: 1º Código: Ano de Exercício, 2º Código: Unidade Orçamentária, 3º Código: Função, 4º Código: Subfunção, 5º Código: Programa, 6º Código: Ação, 7º Código: Grupo de Despesa, 8° Código: Fonte do Recurso. Exemplo: 2017.0701.03.091.4001.4001.01.100 *Obs: se tiver um 9° Código com 2 dígitos: Modalidade de Aplicação.
7	Função (Cod/Nome)	A função representa o maior nível de agregação das diversas áreas de atuação do setor público. Ela se relaciona com a missão institucional do órgão, por exemplo, cultura, educação, saúde, defesa. Esse campo é composto do código e do nome da função.
8	Subfunção (Cod/Nome)	A subfunção é o próximo nível de agregação inferior à função e deve evidenciar cada área da atuação governamental, por intermédio da agregação de determinado subconjunto de despesas e identificação da natureza básica das ações que se aglutinam em torno das funções. Esse campo é composto do código e do nome da subfunção.
9	Programa (Cod/Nome)	Programa é usado para organizar a atuação governamental que articula um conjunto de ações que concorrem para a concretização de um objetivo comum preestabelecido, visando à solução de um problema ou ao atendimento de determinada necessidade ou demanda da sociedade. Esse campo é composto do código e nome do programa. A Subfunção não necessariamente guarda correlação com a função originária.

N.	Atributo	Descrição
10	Ação (Cod/Nome)	As ações são operações das quais resultam produtos (bens ou serviços), que contribuem para atender ao objetivo de um programa. As ações, conforme suas características podem ser classificadas como atividades, projetos ou operações especiais. Esse campo é composto pelo código e nome da ação.
11	Grupo Despesa (Cod/Nome)	É um agregador de elementos de despesa com as mesmas características quanto ao objeto de gasto Esse campo representa o código e nome do grupo de despesa.
12	Elemento Despesa (Cod/Nome)	Identifica os objetos de gasto, tais como vencimentos e vantagens fixas, juros, diárias, material de consumo, serviços de terceiros prestados sob qualquer forma, subvenções sociais, obras e instalações, equipamentos e material permanente, auxílios, amortização e outros que a administração pública utiliza para a consecução de seus fins. Esse campo representa o código e nome do elemento de despesa.
13	**Natureza Despesa (Cod)** **Rótulo**	É do tipo numérico, contém o código completo da natureza da despesa. Ela é identificada pelo conjunto de códigos, a seguir indicados: 1º Código: Categoria econômica, 2º Código: Grupo de despesa, 3º Código: Modalidade de aplicação, 4º Código: Elemento de despesa, 5º Código: Subelemento de despesa.
14	Natureza Despesa (Nome)	É do tipo string e seu valor representa o nome da natureza da despesa.
15	Formalidade (Nome)	Contêm valores de formalidade, exemplo, Folha de Pagamento – Líquido, Apropriação de Despesa e outros.
16	Modalidade Licitação (Nome)	Indica o procedimento que irá reger a licitação, são modalidades: concorrência, tomada de preços, convite, concurso, dispensa licitação, pregão e outros.

N.	Atributo	Descrição
17	Fonte Recurso (Cod)	Fonte de recurso é a classificação da receita segundo a destinação legal dos recursos arrecadados. As fontes de recursos constituem-se de determinados agrupamentos de naturezas de receitas, atendendo a uma determinada regra de destinação legal, e servem para indicar como são financiadas as despesas orçamentárias. Esse campo contém o código da fonte de recurso.
18	Fonte Recurso (Nome)	Esse campo contém o nome da fonte de recurso.
19	Beneficiário (CNPJ)	Beneficiário é um credor do ente público, representa uma pessoa física ou jurídica destinatária do empenho realizado. Esse campo contém o CNPJ do beneficiário, caso seja pessoa jurídica.
20	Beneficiário (CPF)	Esse campo contém o CPF do beneficiário, caso seja pessoa física.
21	Beneficiário (CPF/CNPJ)	É o número do CPF ou CNPJ dependendo do tipo do beneficiário (Físico ou Jurídico).
22	Beneficiário (Nome)	Esse campo contém o nome do beneficiário.
23	Período (Dia/Mes/Ano)	Esse campo contém a data de quando o valor foi empenhado, no formato dd/mm/aaaa.
24	Empenho (Número do Processo)	Empenho é o ato emanado de autoridade competente que cria para o Estado obrigação de pagamento pendente ou não de implemento de condição. Consiste na reserva de dotação orçamentária para um fim específico. Esse campo representa o número do processo que originou o empenho.
25	Empenho (Sequencial Empenho)	Esse campo representa o número completo do empenho, que é composto pelo ano de exercício, código do órgão, número sequencial de dotação e número do empenho. O número sequencial da dotação (3 dígitos), representa uma sintetização da classificação orçamentária do terceiro ao oitavo/nono códigos, com vigência no exercício a que se referem.
26	Empenho (Histórico)	Representa o histórico do empenho, contendo informações complementares sobre a finalidade do empenho.
27	Valor Empenhado	Representa o valor numérico do valor empenhado.
28	Valor Anulação Empenho	Representa o valor de anulação do empenho, que pode ser total ou parcial. Documento expedido no mesmo ano de emissão do empenho.

N.	Atributo	Descrição
29	Valor Estorno Anulação Empenho	Representa o valor de estorno do valor de anulação do empenho. Esse documento faz o inverso da Anulação de Empenho, devolvendo o saldo ao mesmo.
30	Valor Cancelamento Empenho	Representa o valor de cancelamento do empenho. Documento expedido em exercício diferente da emissão do empenho.
31	Valor Anulação Cancelamento Empenho	Representa o valor de anulação do valor de cancelamento do empenho.
32	Valor Saldo do Empenho	É o valor do saldo do empenho considerando todas as anulações, cancelamentos e estornos do empenho.
33	Valor Liquidação Empenho	Representa o valor de liquidação de empenho.
34	Valor Anulação Liquidação Empenho	Representa o valor de anulação de liquidação de empenho.
35	Valor Saldo Liquidado	É o valor do saldo liquidado, depois de considerar as anulação e estornos da liquidação.
36	Valor Ordem de Pagamento	Representa o valor numérico da ordem de pagamento.
37	Valor Guia Recolhimento	Representa o valor do guia de recolhimento.
38	Valor Anulação Ordem de Pagamento	É o valor de anulação da ordem de pagamento.
39	Valor Estorno Anulação O. Pagamento	Representa o valor que foi estornado do valor de anulação da ordem de pagamento.
40	Valor Estorno Guia Recolhimento	É o valor de estorno do guia de recolhimento.
41	Valor Saldo Pago	Representa o valor numérico do saldo pago.
42	Valor Saldo a Pagar	Representa o saldo que ainda falta pagar ou credor.
43	Valor a Liquidar	Representa o valor a liquidar.
44	Valor a Pagar Liquidado	É o valor a pagar liquidado.

No intuito de compreender o *dataset*, primeiramente foi realizada uma análise superficial dos dados usando a biblioteca Pandas – uma poderosa ferramenta *open source* para manipulação e análise de dados –, com o auxílio do dicionário de dados. A partir dessa análise, foram realizadas diversas reuniões com Analistas de Controle Externo do TCE-GO, que são especialistas nesses dados para o esclarecimento de dúvidas e melhor entendimento de cada atributo. Isso resultou na compilação do Apêndice A, que contém

as perguntas realizadas aos especialistas e as respostas obtidas deles. Resumidamente, pode-se observar naquele Apêndice perguntas com o objetivo de sanar dúvidas sobre o atributo em si, ou seja, entender o significado deles em termos de negócio e, além disso, descobrir quais atributos são derivados de outros.

Além disso, a exploração do *dataset* através do Pandas possibilitou a extração de informações que estão dispostas na coluna "Observações adicionais" da Tabela 12 presente no Apêndice B, que, de maneira resumida, se refere a quantidade de valores diferentes em cada atributo; a exemplos dos diferentes tipos de valores de cada campo e, finalmente, a determinadas características do atributo, como, por exemplo, a formatação dos dados ou alguma divergência de padrão em relação a maioria. Ainda, nessa mesma tabela a coluna "Decisão da Exploração de Dados" informa como cada atributo será tratado para que fique no formato exigido pelos algoritmos avaliados. Por fim, a última coluna, "Atributos para Geração dos Modelos", informa quantos novos atributos serão gerados após seu tratamento.

Para a etapa de *preparação dos dados*, foram realizadas diversas tarefas, como a criação de novos atributos a partir dos já existentes; a exclusão de atributos redundantes, pois já estavam presentes em outros atributos, a exclusão de atributos com pouco ou nenhum poder representativo, assim como a remoção de alguns empenhos com determinadas características. A Tabela 2 mostra quais novos atributos foram criados, indicando ainda quais colunas foram utilizadas para a sua criação, e quais atributos foram excluídos, incluindo o motivo da exclusão.

Isso se fez necessário pois apesar de algumas características serem óbvias para um observador humano, podem não ser para um algoritmo de aprendizado de máquina. Então, deve-se explicitar essas características para os algoritmos. Além disso, atributos não-relevantes ou pouco discriminativos podem *piorar* o desempenho do modelo treinado e por isso podem ser removidos.

As características observadas para a remoção dos empenhos foram: empenhos que não estavam mais vigentes; empenhos com saldo zerado, pois não possuem relevância para o TCE-GO; empenhos pertencentes a naturezas com apenas 3 documentos e, por fim, empenhos pertencentes ao elemento de despesa de código 92, "Despesas de Exercícios Anteriores", devido a sua peculiaridade em conter naturezas distintas e assim dificultar o trabalho de reconhecimento de padrões dos algoritmos.

Tabela 2 – Tabela de manipulação dos atributos.

N.	**Atributos criados**	**Atributos excluídos**	**Motivo da exclusão / Criação novo atributo**
1	acao_programa	Programa (Cod/Nome)(EOF)	"Ação (Cod/Nome)(EOF)" e "Programa (Cod/Nome)(EOF)" tiveram seus códigos removidos por terem códigos diferentes para o mesmo nome e então foram agregados gerando o novo campo "acao_programa"
2		Ação (Cod/Nome)(EOF)	
3	pessoa_juridica	Beneficiário (CNPJ)(EOF)	Todos esses campos foram removidos, gerando-se um novo campo denominado "pessoa_juridica", onde o valor 1 significa que a pessoa é jurídica e 0 caso seja pessoa física. Esses campos foram removidos, pois não julgamos necessários para classificação da natureza.
4		Beneficiário (CPF)(EOF)	
5		Beneficiário (CPF/CNPJ)(EOF)	
6	tfidf_beneficiario	Beneficiário (Nome)(EOF)	O campo "Beneficiário (Nome)(EOF)" é tratado como texto livre e a representação TF-IDF é aplicada a ele, gerando novas colunas que são acrescentadas aos dados.
7	orgao_sucedido	Órgão (Código/Nome)(EOF)	O campo "Órgão Sucessor Atual (Código/Nome)(EOF)" já nos diz qual é o órgão em vigência no momento e o campo "Órgão (Código/Nome)(EOF)" possui códigos diferentes para o mesmo nome. Por isso, o campo "Órgão (Código/Nome)(EOF)" foi substituído por um meta-atributo que possui valor 1 quando o órgão for diferente do órgão sucessor, ou seja, quando o órgão foi sucedido por outro, e 0 caso contrário.

N.	**Atributos criados**	**Atributos excluídos**	**Motivo da exclusão / Criação novo atributo**
8	empenho_por _processo	Empenho (Número do Processo)(EOF)	Campo removido por não ter uma padronização dos códigos, pois cada órgão utiliza uma forma diferente. A partir desse campo, foi calculada a quantidade de empenhos contidas em um processo, sendo adicionada a uma nova coluna denominada "empenho_por_processo".
9	Atributos criados a partir da representação TF-IDF	Empenho (Histórico)(EOF)	Como esse é um campo de texto livre, ele foi substituído por sua representação TF-IDF.
10		Classificação orçamentária (Descrição)(EOF)	Possui todos os valores de seus códigos já presentes em outros campos. Por isso, foi excluído.
11		Natureza Despesa (Nome)(EOF)	Campo excluído por já ser representado pelo rótulo.
12		Empenho (Sequencial Empenho)(EOF)	Esse campo é o identificador do empenho e, como tem valor diferente para cada empenho, ele não discretiza os documentos. Por isso, ele foi removido.
13		Valor Estorno Anulação Empenho(EOF)	Campo removido por ter apenas o valor 0 para todos os empenhos. Assim, não possui poder discriminativo.
14		Valor Anulação Cancelamento Empenho(EOF)	Campo removido por ter apenas o valor 0 para todos os empenhos. Assim, não possui poder discriminativo.
15		Fonte Recurso (cod)(EOF)	Campo excluído por já ser representado pelo campo "Fonte Recurso (Nome)(EOF)".
16		Elemento Despesa (Cod/Nome)(EOF)	Como é um atributo já presente no rótulo, ele precisa ser retirado das demais colunas para não interferir no aprendizado.
17		Grupo Despesa (Cod/Nome)	Como é um atributo já presente no rótulo, ele precisa ser retirado das demais colunas para não interferir no aprendizado.

N.	Atributos criados	Atributos excluídos	Motivo da exclusão / Criação novo atributo
18	periodo	Período (Dia/Mes/Ano)	Como esse campo é uma data, e datas não se repetem no contexto desses dados, esse campo em sua completude atrapalharia a criação de padrões. Por esse motivo, esse campo teve uma alteração onde apenas o mês extraído da data presente nele foi mantido.

Ainda no decorrer dessa etapa, os atributos 3, 4, 5, 7, 8, 15, 16 e 18 da Tabela 1 foram tratados utilizando a representação *One-Hot Encoding* por serem do tipo categórico, assim como o atributo "acao_programa". Por fim, o atributo de texto livre de número 26 da Tabela 1, "Empenho (Histórico)", foi tratado inicialmente passando o texto para caracteres minúsculos para evitar diferenças de grafia da mesma palavra. Em seguida, foram realizadas as seguintes ações:

1. A remoção de caracteres especiais, como cedilha e letras acentuadas;
2. O tratamento de palavras específicas identificadas a partir de observação geral dos dados do texto;
3. A remoção de possíveis URLs (*Uniform Resource Locators*), pontuações, números e *stopwords*, que são palavras frequentes mas sem poder discriminativo como artigos, preposições etc.

Por fim, o texto resultante foi transformado em formato numérico utilizando a representação TF-IDF, produzindo por volta de 60.000 novos atributos, que formaram outro *dataset* com uma nova visão dos dados. Esse *dataset* posteriormente foi utilizado tanto separadamente, como em conjunto com o *dataset* principal para avaliar qual a melhor representação de dados. Houve também a necessidade de se utilizar o atributo 22 da Tabela 1 e, devido à sua natureza, utilizou-se a representação TF-IDF nesse campo, assim como o tratamento inicial de dados citado acima. Além disso, esse campo possui tanto nomes de órgãos como de pessoas físicas e, afim de se capturar os nomes dos órgãos, todos os nomes de pessoas físicas foram substituídos por "PF". O resultado disso foi agregado ao *dataset* principal. Todas as ações tomadas na preparação dos dados são justificadas na coluna "Decisão da Exploração de Dados" na Tabela 12 do Apêndice B.

Adicionalmente, a preparação dos dados permitiu extrair ainda mais informações sobre os atributos que compõem os empenhos, como a correlação entre os atributos e a

distribuição estatística do rótulo. A correlação $-1{,}0 \leq c_{(i,j)} \leq 1{,}0$ permite identificar a similaridade entre os atributos i e j e nos diz o quanto esses atributos são parecidos entre si. Nesse sentido, atributos semelhantes possuem correlação próxima de 1,0, enquanto atributos distintos, com visões diferentes dos dados, possuem correlação próxima de $-1{,}0$. Por outro lado, atributos que possuem pouca relação entre si assumem valores próximos de zero.

A distribuição estatística, por sua vez, auxilia no entendimento da dificuldade do problema e na natureza dos dados. Dessa forma, uma distribuição desbalanceada das classes de um problema pode indicar um problema difícil, pois algumas classes terão poucos documentos para representá-las, enquanto outras classes terão muitos documentos. Uma análise aprofundada dessa distribuição estatística se faz útil para auxiliar na escolha do algoritmo mais apropriado para o problema em questão.

A partir do *feature importance*, um valor adquirido a partir do algoritmo *Random Forest*, que calcula a importância de cada atributo, foi possível estabelecer quais eram os 10 atributos mais importantes. A seguir, realizou-se um cálculo de correlação entre eles, gerando a imagem mostrada na Figura 8. Vale ressaltar que a quantidade de atributos selecionados pra realizar o cálculo neste exemplo teve como objetivo a melhor visualização dos valores, e não significa que apenas esses atributos foram selecionados para futuras análises.

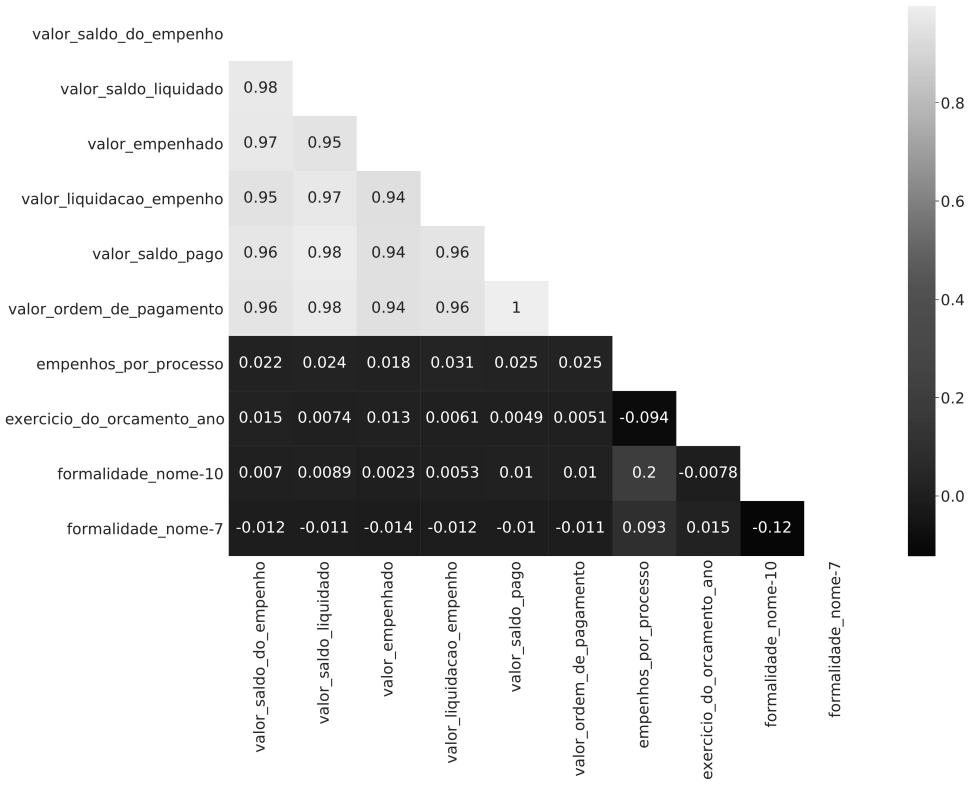


Figura 8 – Matriz de correlação dos 10 atributos mais importantes do *dataset* segundo o *feature importance*.

Na Figura 8, pode-se observar que existe uma forte correlação positiva entre os atributos relativos a valores monetários, como "valor_saldo_do_empenho", "valor_saldo_liquidado", "valor_empenhado" etc. Em outras palavras, os valores desses atributos possuem comportamentos parecidos. Por outro lado, é possível observar que os demais atributos possuem uma fraca correlação positiva ou negativa, significando que eles são razoavelmente independentes entre si.

Também, foi realizada a análise de distribuição estatística para o atributo tido como rótulo, produzindo a Figura 9. Pode-se observar nessa figura que o rótulo é bastante desbalanceado, pois a natureza de código "3.3.90.35.11" possui apenas dois empenhos, enquanto a natureza "3.3.90.18.05" possui 5.978. Além disso, analisando essa distribuição observou-se que a maioria das naturezas de despesa possuem menos de 100 empenhos, o que pode mostrar a dificuldade deste problema. Isso implica na dificuldade de entendimento dos padrões para aquelas naturezas de despesa, o que pode fazer com que a acurácia geral do modelo abaixe.

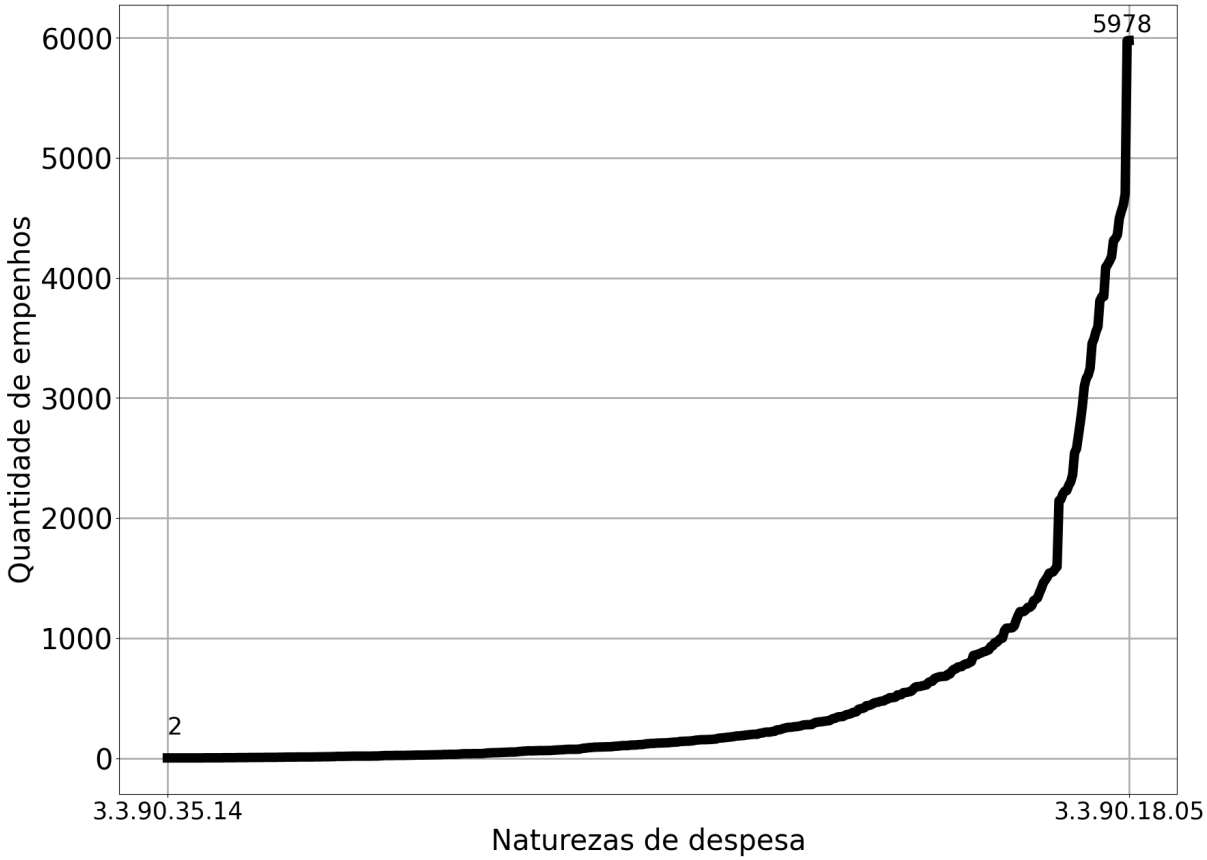


Figura 9 – Distribuição dos valores do atributo "Natureza Despesa(Cod)", o rótulo do problema proposto.

Para o quinto passo, *explorar diferentes modelos de aprendizado de máquina*, o problema proposto neste trabalho fez uso de diversos algoritmos de forma separada. Além disso, também foram aplicadas as estratégias denominadas *Stacking* e Oráculo, que permi-

tem utilizar uma quantidade maior de algoritmos por vez. Os algoritmos individualmente avaliados foram: *Support Vector Machine Classifier* (SCIKIT-LEARN, 2021i), *Random Forest* (SCIKIT-LEARN, 2021g), *K-Nearest Neighbor* (SCIKIT-LEARN, 2021e) e Gradiente Descendente Estocástico (SCIKIT-LEARN, 2021h). A ideia é determinar qual dessas estratégias traz melhores resultados para os dados analisados. Vale ressaltar que os algoritmos *CenKNN* (PANG; JIN; JIANG, 2015), *Radius Neighbors Classifier* (SCIKIT-LEARN, 2021f), *Naïve Bayes* (SCIKIT-LEARN, 2021c) e *Nearest Centroid* (SCIKIT-LEARN, 2021d) também foram avaliados de forma individual, entretanto, obtiveram resultados inferiores e por essa razão foram desconsiderados.

Ainda nessa etapa, um subconjunto de 5.899 documentos foi selecionado a partir do *dataset* para ser analisado e validado por uma especialista. A intenção foi separar um conjunto de documentos dos quais se tem certeza da sua corretude. Para isso, a especialista classificou cada um desses empenhos em "correto", "incorreto" ou "inconclusivo". A Tabela 3 mostra a porcentagem de documentos "corretos", "incorretos" e "inconclusivos" validados pela especialista de dados. Vale notar que os dados inconclusivos são aqueles que a especialista não conseguiu categorizar em "correto" ou "incorreto" devido a insuficiência de informações. Portanto, considerando os documentos validados pela especialista, temos que essa análise humana atingiu uma acurácia de cerca de 90%.

Tabela 3 – Dados validados pela especialista de dados quanto a sua corretude.

	Corretos	**Inconclusivos**	**Incorretos**	**Total**
Quantidade	5058	559	282	5899
Porcentagem	85.74%	9.47%	4.78%	

A partir desses dados construiu-se um segundo modelo de aprendizado de máquina, tendo essas avaliações como rótulo. Portanto, para cada empenho, são realizadas duas predições. A predição do primeiro modelo diz respeito à sua natureza de despesa, enquanto a segunda é relativa à sua possível corretude. A Figura 10 exemplifica a organização desses modelos.

O motivo da criação desse segundo modelo é dar maior robustez à classificação de um empenho, visto que agora existem dois modelos independentes realizando predições. Portanto, há seis possíveis combinações dos resultados que a combinação dos modelos pode produzir, que são apresentadas na Tabela 4.

Pode-se observar na Tabela 4 que o primeiro modelo – aquele responsável por predizer a natureza de despesa – pode concordar ou discordar com a natureza de despesa já presente no empenho. Além disso, o segundo modelo produz uma das três avaliações supracitadas. A combinação dessas predições geram seis possíveis conclusões.

Quando há a concordância dos dois modelos, ou seja, quando o primeiro modelo

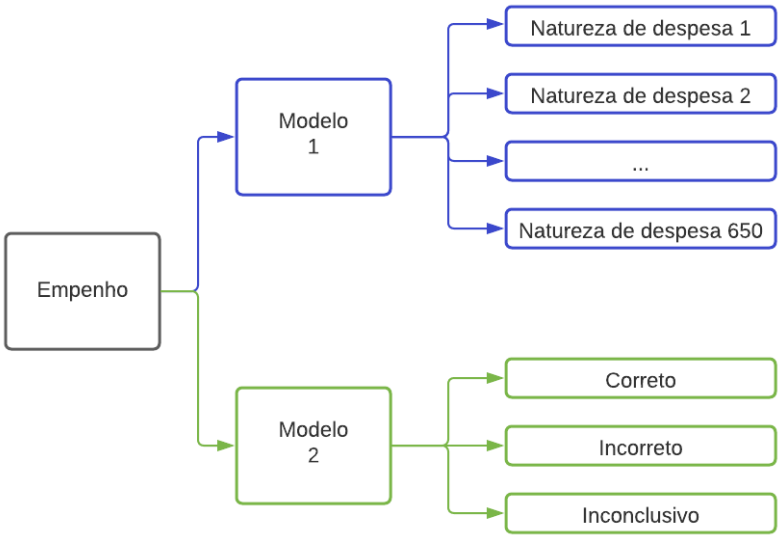


Figura 10 – Estrutura de predição para cada empenho avaliado.

Tabela 4 – Tabela com as possibilidades de resposta dos modelos.

Possibilidades	**Primeiro modelo**	**Segundo modelo**	**Conclusão**
Saída 1	Concorda com empenho	Empenho incorreto	Inconclusivo pelo segundo modelo (INCV_M2)
Saída 2	Concorda com empenho	Empenho correto	Correto em ambos os modelos (C_M1-M2)
Saída 3	Concorda com empenho	Empenho inconclusivo	Avaliação dúbia do segundo modelo (AD_M2)
Saída 4	Discorda do empenho	Empenho incorreto	Incorreto em ambos os modelos (INCT_M1-M2)
Saída 5	Discorda do empenho	Empenho correto	Inconclusivo pelo primeiro modelo (INCV_M1)
Saída 6	Discorda do empenho	Empenho inconclusivo	Inconclusivo pelo primeiro modelo e avaliação dúbia do segundo modelo (INCV_M1-AD_M2)

concorda com a natureza de despesa presente e o segundo prediz que o empenho está "correto", diz-se que o empenho está correto em ambos os modelos (C_M1-M2). Porém, quando o primeiro modelo discorda e o segundo modelo indica a avaliação "incorreto" ao empenho, diz-se que esse empenho está incorreto em ambos os modelos (INCT_M1-M2).

Na discordância entre as predições, ou seja, o primeiro modelo concorda com a predição mas o segundo considera incorreto ou vice-versa, há um resultado inconclusivo

por conta da discordância de um dos modelos, podendo resultar nas saídas "Inconclusivo pelo segundo modelo (INCV_M2)" ou "Inconclusivo pelo primeiro modelo (INCV_M1)".

Já nos casos onde o primeiro modelo concorda com a natureza de despesa presente no empenho mas o segundo modelo considera o empenho inconclusivo, temos como saída "Avaliação dúbia do segundo modelo (AD_M2)". Por fim, quando o primeiro modelo discorda com a natureza do empenho e o segundo modelo considera aquele empenho inconclusivo temos "Inconclusivo pelo primeiro modelo e avaliação dúbia do segundo modelo (INCV_M1-AD_M2)" como saída. Vale ressaltar que a conclusão de maior interesse por parte do TCE-GO é a "Saída 4" da Tabela 4, onde o empenho foi classificado como incorreto por ambos os modelos avaliados.

Seguindo o *checklist*, os algoritmos selecionados na etapa anterior passam pela etapa de *refinamento dos modelos*. Ressalta-se aqui que, em vez de se realizar uma busca manual do melhor conjunto de parâmetros para cada algoritmo, foi adotada uma metodologia de testes.

Isso envolveu o uso do *Grid Search*, que é um algoritmo utilizado em métodos de otimização de funções matemáticas, que, nesse caso, é o erro de predição do modelo gerado por um algoritmo preditor. Nesse sentido, objetiva-se minimizar esse erro, realizando uma busca exaustiva tendo como referência um intervalo de valores predefinidos pelo usuário para cada hiperparâmetro.

É importante ficar claro que um processo de busca exaustiva é, nesse contexto, equivalente a testar todas as combinações possíveis de valores contidos nos intervalos para cada um dos hiperparâmetros. O resultado desse processamento é o conjunto de hiperparâmetros que minimizou o erro de predição do modelo avaliado.

Ainda, para que esse processo produza resultados mais confiáveis, deve-se incluir o *cross-validation*. Dessa forma, o *Grid Search* escolhe o melhor conjunto de hiperparâmetros tendo como referência a média aritmética dos K valores do *cross-validation* para cada conjunto avaliado de hiperparâmetros.

Porém, o *Grid Search* possui elevado custo computacional e, por isso, o processo de busca foi realizado em dois estágios. O primeiro estágio realiza uma *coarse search*, que é responsável por realizar uma busca utilizando um intervalo predefinido. O segundo, por sua vez, realiza uma busca detalhada na vizinhança do resultado obtido no primeiro estágio (*fine search*). Isso permite obter valores mais precisos do melhor conjunto de hiperparâmetros enquanto mantém-se controlado o custo computacional.

A Tabela 5 mostra para cada representação de dado e para cada algoritmo individual qual foi o conjunto inicial de valores analisados pelo *Grid Search* para os hiperparâmetros avaliados.

Ainda, como escrito na Seção 2.4.1, a estratégia *Stacking* possui duas etapas. Para

Tabela 5 – Tabela de intervalo dos hiperparâmetros avaliados utilizando a estratégia *Grid Search*.

Representação	**Algoritmo**	**Hiperparâmetro**	**Valores**
OHE; TF-IDF; OHE+TF-IDF	RF	n_estimators	100; 300; 500; 700; 1000
	SVM	C	0,1; 1; 10; 100
	KNN	n_neighbors	1; 3; 5; 7
	SGD	max_inter	100; 300; 500; 700

este trabalho, a primeira delas faz uso das duas representações de dados extraídas do *dataset* para a aplicação dos algoritmos. Tanto a primeira como a segunda etapa avaliaram os algoritmos *Random Forest*, *K-Nearest Neighbors* e *Support Vector Machine*. A segunda etapa tem seu *dataset* composto a partir do resultado da aplicação dos algoritmos na etapa anterior. Vale ressaltar que, para cada representação de dados da primeira etapa, apenas o melhor algoritmo dentre os avaliados será executado. A Tabela 6 mostra os algoritmos que foram avaliados em cada representação de dados, assim como, os hiperparâmetros e seus intervalos de valores.

Tabela 6 – Tabela da organização das etapas do *Stacking*.

Etapa	**Representação**	**Algoritmo**	**Hiperparâmetro**	**Valores**
Etapa 1	OHE	RF	n_estimators	100; 300; 500; 700; 1000
		KNN	n_neighbor	1; 3; 5; 7
		SVM	C	0,1; 1; 10; 100
	TF-IDF	RF	n_estimators	100; 300; 500; 700; 1000
		KNN	n_neighbor	1; 3; 5; 7
		SVM	C	0,1; 1; 10; 100
Etapa 2	Metadados	RF	n_estimators	100; 300; 500; 700; 1000
		KNN	n_neighbor	1; 3; 5; 7
		SVM	C	0,1; 1; 10; 100

Por fim, nas duas últimas etapas, que são *apresentar sua solução* e *lançar, monitorar e manter o sistema*, o produto deste trabalho foi entregue ao TCE-GO, que são os *scripts* Python para o modelo de aprendizado de máquina criado, acompanhado da respectiva documentação com as explicações sobre as decisões tomadas durante a construção do produto. O produto entregue foi ajustado de acordo com as necessidades daquele Tribunal de Contas, que foram informadas através de reuniões e conversas.

Então, essa solução foi implantada no TCE-GO e está em uso de forma que os treinamentos dos modelos são realizados mensalmente, enquanto a predição pode ser feita diariamente conforme o surgimento de novos empenhos ou, de acordo com a necessidade do TCE-GO. A criação das interfaces que consumirão o resultado do modelo ficou a cargo daquele Tribunal de Contas. Futuramente, caso seja necessário, essa solução deverá

passar por um processo de *monitoramento e manutenção* para assegurar que o produto desenvolvido atenda às expectativas do cliente e funcione como o esperado.

4 Resultados

Este capítulo tem o objetivo de introduzir as métricas de avaliação que foram utilizadas neste Trabalho de Conclusão de Curso, assim como demonstrar e interpretar os resultados obtidos a partir dos experimentos realizados.

4.1 Métricas de avaliação

Antes dos resultados serem apresentados, essa seção aborda as métricas de avaliação que foram utilizadas para fazer as comparações dos modelos treinados usando os algoritmos abordados na Seção 2.3.

A métrica utilizada neste trabalho é denominada métrica F1, que pode ser interpretada como uma média ponderada da precisão ("*precision*") e da revocação ("*recall*") (SCIKIT-LEARN, 2021b).

Pode-se fazer uso de um simples exemplo para abordar os conceitos utilizados por essa métrica. Considere um indivíduo realizando predições sobre o estado de saúde de um conjunto de pessoas utilizando os valores "enferma" e "não-enferma". Nessa circunstância, existem quatro predições possíveis:

- Uma pessoa enferma recebe a predição "enferma";
- Uma pessoa enferma recebe a predição "não-enferma";
- Uma pessoa não-enferma recebe a predição "enferma";
- Uma pessoa não-enferma recebe a predição "não-enferma".

Respectivamente, essas predições podem ser consideradas como: verdadeira-positiva ("*True Positive*" – TP); falso-negativa ("*False Negative*" – FN); falso-positivo ("*False Positive*" – FP) e, por fim, verdadeiro-negativo ("*True negative*" – TN).

Levando isso em consideração, pode-se calcular a precisão em função das quantidades de verdadeiros-positivos e de falsos-positivos, como apresentado na Eq. (4.1) (SILVA; PERES; BOSCARIOLI, 2017):

$$\text{Precisão} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{4.1}$$

Semelhantemente, a revocação pode ser calculada como mostrado na Eq. (4.2):

$$\text{Revocação} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{4.2}$$

Por fim, a métrica F1 é calculada como apresentado na Eq. (4.3):

$$\text{F1} = 2 \times \frac{\text{Precisão} \times \text{Revocação}}{\text{Precisão} + \text{Revocação}}, \tag{4.3}$$

onde F1 assume valores no intervalo $[0, 1]$.

A implementação utilizada para essa métrica foi retirada da biblioteca Scikit-learn disponível no Python. Essa implementação possui várias opções para o cálculo de sua média, incluindo a Micro F1 e a Macro F1. A Micro F1 calcula a quantidade de verdadeiros-positivos e falsos-negativos sem fazer distinção de classe. Em outras palavras, a Micro F1 é relativa à taxa de acerto global do modelo preditor. Por outro lado, a Macro F1 é o resultado da média aritmética do cálculo da métrica F1 para cada uma das classes individualmente. Dessa forma, a Macro F1 não considera um possível desbalanceamento entre as classes (SCIKIT-LEARN, 2021b).

Este Trabalho de Conclusão de Curso utiliza tanto a Micro F1 quanto a Macro F1 na avaliação dos modelos preditores. Isso porque, um valor de Micro F1 próximo de um, que, *a priori* pode ser considerado um ótimo resultado, não necessariamente tem esse significado. A razão para isso é que o modelo preditor pode ser enviesado, possuindo alta taxa de acerto na predição de documentos de uma determinada classe dominante no *dataset*, mesmo que ele apresente baixa taxa de acerto nas demais classes. Nesse contexto, a Macro F1 atribui valores mais próximos a zero para modelos preditores que possuem essa tendência, já que ela avalia as classes de forma individual, sem levar em consideração a quantidade de documentos de cada uma delas.

4.2 Resultados

No decorrer deste trabalho foram executados diversos testes e experimentos para avaliar qual a melhor estratégia a ser utilizada considerando o *dataset* em questão. Esses testes compreendem desde a escolha de um algoritmo até o refinamento dos seus hiperparâmetros, que são mostrados nesta seção.

Assim como foi explicado na Seção 3.2, a etapa de *refinar os modelos* utilizou o algoritmo *Grid Search* numa estratégia em duas etapas, onde a primeira realiza uma busca mais ampla de valores dos hiperparâmetros, que são refinados pela segunda etapa. Isso pode ser observado na Tabela 7, que mostra, para cada representação de dados, os valores selecionados para os hiperparâmetros de cada algoritmo quando eles são avaliados individualmente. É importante ressaltar que, como essa é apenas uma busca do melhor conjunto de hiperparâmetros, ela abrangeu 40% dos dados, que foram selecionados de forma estratificada, onde a proporção é mantida nessa seleção. Além disso, a Tabela 7 mostra também os resultados obtidos em cada algoritmo segundo as métricas utilizadas para avaliação. Vale ressaltar que esses testes envolveram apenas o primeiro modelo de

Tabela 7 – Tabela de resultados dos hiperparâmetros, algoritmos avaliados e seus resultados.

Representação	**Algoritmo**	**Hiperparâmetro**	**Melhor Valor**	**Micro F1**	**Macro F1**
OHE	**RF**	**n_estimators**	**1020**	**0,61**	**0,42**
	K-NN	n_neighbor	1	0,49	0,34
	SGD	max_inter	80	0,41	0,18
	Média OHE			0,50	0,31
TF-IDF	RF	n_estimators	980	0,72	0,51
	SVM	**C**	**10**	**0,76**	**0,58**
	K-NN	n_neighbor	1	0,68	0,50
	SGD	max_inter	80	0,57	0,19
	Média TF-IDF			0,68	0,45
OHE+TF-IDF	RF	n_estimators	980	0,74	0,52
	SVM	**C**	**10**	**0,76**	**0,58**
	K-NN	n_neighbor	1	0,68	0,49
	SGD	max_inter	80	0,65	0,29
	Média OHE+TF-IDF			0,71	0,47

aprendizado de máquina. Isso se dá, pois os únicos documentos que possuem rótulo para o segundo modelo são aqueles que foram validados pela especialista. Por isso, não é possível utilizar as métricas de avaliação consideradas neste trabalho para o segundo modelo de aprendizado de máquina.

Na Tabela 7, os melhores algoritmos para cada representação de dados estão destacados em negrito. Pode-se observar que o *Random Forest* obteve os maiores valores de Micro F1 e Macro F1 considerando a representação *One-Hot Encoding*. Entretanto, para a representação TF-IDF e a representação OHE combinada à TF-IDF, o algoritmo que obteve os maiores valores de Micro F1 e Macro F1 foi o SVM. Vale ressaltar que o SVM não foi avaliado para a representação OHE devido ao elevado tempo de execução para a criação do modelo. Adicionalmente, dentre os três tipos de representação de dados, a representação combinada é aquela obteve o melhor resultado, levando em consideração os valores médios. Por essa razão, quando se tratam dos algoritmos avaliados individualmente, o SVM em conjunto com a representação combinada produz os melhores resultados.

Além disso, também foi realizada uma análise de hiperparâmetros para a estratégia *Stacking*. Contudo, vale ressaltar que não é necessário avaliar novamente na primeira etapa do *Stacking* todos os algoritmos preditores para cada representação de dados. Isso se dá, pois, a partir da Tabela 7, já se sabe qual é o melhor algoritmo para cada representação. A Tabela 8 mostra os valores selecionados para os hiperparâmetros de cada algoritmo avaliado na segunda etapa, assim como os seus resultados.

Observa-se na Tabela 8 que, para a segunda etapa da estratégia *Stacking*, o algoritmo *Random Forest* obteve o mesmo valor de Micro F1 que o SVM, porém, obteve o maior

Tabela 8 – Tabela de resultados dos hiperparâmetros para cada etapa do *Stacking*.

Etapa	**Representação**	**Algoritmo**	**Hiperparâmetro**	**Valor**	**Micro F1**	**Macro F1**
Etapa 1	OHE	RF	n_estimators	1020		
	TF-IDF	SVM	C	10		
Etapa 2	Meta-dados	SVM	C	10	0,88	0,75
		RF	**n_estimators**	**1000**	**0,88**	**0,76**
		K-NN	n_neighbor	7	0,87	0,73

valor considerando a métrica de avaliação Macro F1, cuja linha na tabela está destacada em negrito. Por essa razão, o *Random Forest* foi o algoritmo selecionado para a segunda etapa da estratégia *Stacking*.

A última estratégia considerada é o Oráculo, porém, como ele utiliza os algoritmos que já foram avaliados anteriormente, não se exigiram avaliações adicionais. Com isso, essa estratégia tem como referência para seus algoritmos os valores de hiperparâmetros presentes na Tabela 7. Vale ressaltar que, assim como mostrado nessa mesma tabela, a representações OHE combinada à TF-IDF obteve o melhor resultado médio. Portanto, a estratégia Oráculo utiliza essa representação.

Por fim, após a descoberta dos algoritmos e hiperparâmetros mais apropriados para cada experimento executado, foi então realizada uma última execução desses experimentos. Porém, considerando a totalidade dos dados, a fim de determinar o resultado final. Esses resultados estão dispostos na Tabela 9.

Tabela 9 – Tabela com os resultados finais dos experimentos avaliados.

Estratégia	**Micro F1**	**Macro F1**
Algoritmo SVM	0,86	**0,77**
Stacking	**0,90**	0,74
Oráculo Singular	0,86	0,70
Oráculo Múltiplo	0,87	0,72

Pode-se observar na Tabela 9 os resultados obtidos pelas abordagens adotadas neste trabalho, tendo os melhores resultados de acordo com cada métrica de avaliação adotada destacados em negrito. Enquanto a estratégia *Stacking* obteve o maior valor de Micro F1, o mesmo não pode ser dito para a Macro F1. Para essa métrica, a estratégia SVM obteve o maior valor.

Adicionalmente, existem dois resultados para a estratégia Oráculo. O primeiro deles, denominado “Oráculo Singular”, refere-se ao uso da estratégia Oráculo com apenas um algoritmo para cada documento. Além disso, o resultado denominado “Oráculo Múltiplo” é relativo à utilização de mais de um algoritmo por documento.

Isso sugere que a estratégia *Stacking* obteve melhor desempenho na predição de empenhos pertencentes a classes com maior quantidade de documentos. Por outro lado,

o SVM se saiu melhor na predição de empenhos relativos a classes menores, reforçando a afirmação de que esse algoritmo é bem adequado a *datasets* pequenos (GÉRON, 2019; CUNHA et al., 2021), que é o caso quando se consideram individualmente as classes menos dominantes.

Em se tratando do segundo modelo de aprendizado de máquina, adotou-se outro caminho. O motivo para isso é que apenas um subconjunto dos empenhos possui o rótulo determinado pela especialista, impossibilitando a realização de testes a fim de selecionar o melhor algoritmo para esse modelo. Como o *dataset* utilizado para a construção desse segundo modelo possui apenas 5.899 empenhos dentre o total de 324.728, decidiu-se por utilizar o SVM considerando o valor do hiperparâmetro "C" apresentado na Tabela 7.

Por fim, o resultado entregue ao TCE-GO contém a avaliação dos dois modelos através de um arquivo no formato "CSV" (*comma-separated values*) estruturado nos seguintes campos:

1. **Identificador**, relativo ao campo "Empenho (Sequencial Empenho)";
2. **Natureza de despesa**, relativo ao campo "Natureza Despesa (Cod)";
3. **Natureza predita**, relativo à predição do primeiro modelo;
4. **Corretude**, relativo à predição do segundo modelo;
5. **Data da predição**;
6. **Conclusão**, considerando o resultado da predição dos modelos de aprendizado de máquina tendo como referência as regras disposta na Tabela 4.

A Tabela 10 ilustra o resultado da predição de quatro empenhos.

Tabela 10 – Tabela contendo o resultado da predição de quatro empenhos.

Id. empenho	Natureza real	Natureza predita	Corretude	Data predição	Resultado
2016.6602.004.00010	3.3.90.30.33	3.3.90.30.33	CORRETO	30/06/2021	C_M1-M2
2019.2901.004.00220	3.1.90.13.01	3.1.90.13.07	INCORRETO	30/06/2021	INCT_M1-M2
2018.6601.033.00250	3.3.90.30.33	3.3.90.39.20	INCONCLUSIVO	30/06/2021	INCV_M1-AD_M2
2015.2201.001.00540	3.1.90.96.01	3.1.90.96.01	CORRETO	30/06/2021	C_M1-M2

A partir do arquivo gerado, é possível implementar duas novas funcionalidades. A primeira funcionalidade, interna ao TCE-GO, é disponibilizar esses resultados para análise por parte de um especialista. Esse especialista, por sua vez, terá o trabalho de avaliar não todos, mas apenas aqueles resultados considerados significativos por parte do TCE-GO. Portanto, os empenhos classificados como "INCT_M1-M2" devem ser validados por um especialista, pois foram apontados como incorretos por ambos os modelos. Vale ressaltar que esse é o resultado de maior interesse por parte daquele Tribunal.

A segunda funcionalidade é integrar esses modelos ao SIOFINET para que essa predição seja realizada logo ao cadastrar o empenho no sistema. Isso auxiliará o responsável por esse cadastro a classificar corretamente o empenho segundo a sua natureza de despesa.

Finalmente, com o objetivo de avaliar o impacto da ferramenta entregue ao TCE-GO, foi construída uma nova versão da solução de aprendizado de máquina, mas considerando os dados do SIOFINET que abrangem o período de 20 de janeiro de 2015 a 19 de maio de 2021. Em seguida, realizou-se a predição dos empenhos datados entre 20 de maio de 2021 a 16 de julho de 2021, cujos resultados são sumarizados na Tabela 11.

Tabela 11 – Tabela contendo a totalização de empenhos segundo cada saída.

Saída	**Quantidade**
C_M1-M2	7.557
INCV_M1	1.760
AD_M2	1.227
INCV_M1-AD_M2	900
INCV_M2	70
INCT_M1-M2	**46**
Total	11.560

Observa-se na Tabela 11 que, dos 11.560 documentos avaliados, 46 deles foram classificados como incorretos, que foram passados à especialista de dados do TCE-GO para serem verificados. A materialidade, ou seja, o valor em dinheiro associado a esses empenhos totalizam R$ 1.178.256,14.

5 Conclusões

A execução orçamentária do Estado de Goiás é fiscalizada pelo TCE-GO. Porém, existe um grande volume de documentos a serem fiscalizados e poucos profissionais com entendimento necessário para realizar essa tarefa. Por conta disso, faz-se necessária a automatização desse processo de fiscalização. Existem diversas formas de se abordar esse problema e a proposta neste trabalho faz uso de técnicas de aprendizado de máquina.

Para guiar o desenvolvimento da estratégia proposta, este trabalho fez uso de um *checklist* de projetos de aprendizado de máquina. Nele são dispostos os passos a serem executados para a realização de um projeto desse tipo, passando desde o entendimento do problema até a entrega do produto final. Além de mostrar esse *checklist* em sua forma genérica, este trabalho também o aplicou na resolução do problema de fiscalização das naturezas de despesas dos empenhos públicos. O objetivo disso é, além de trazer uma solução para o problema proposto neste trabalho, servir de guia para iniciantes da área de aprendizado de máquina.

Para isso, este trabalho avaliou diversas abordagens, incluindo desde a avaliação individual de algoritmos estado-da-arte até o uso em conjunto desses algoritmos através de estratégias mais robustas. A partir dessa avaliação, um modelo de aprendizado de máquina foi construído com o objetivo de realizar a predição da natureza de despesa. Nesse sentido, foram avaliados os algoritmos *Support Vector Machine*, *Random Forest*, *K-Nearest Neighbors*, *Stochastic Gradient Descent* e as estratégias *Stacking* e Oráculo.

A fim de se avaliar a corretude dos dados disponibilizados para este trabalho e a confiança que se poderia ter neles, foi realizada uma seleção de cerca de 6.000 documentos para a análise por parte de uma especialista. Esses dados foram verificados manualmente por essa especialista, que deu ênfase na verificação da classificação correta de suas naturezas de despesa. Isso permitiu realizar diversos testes utilizando dados validados. Além disso, foi possível construir um segundo modelo de aprendizado de máquina, que tem por objetivo inferir se um empenho público foi classificado corretamente segundo a sua natureza de despesa. Com isso, a intenção foi a de se obter maior robustez nas predições, já que são dois modelos independentes realizando predições sob o mesmo conjunto de dados.

Por mais que os dados disponibilizados pelo TCE-GO para esse projeto constituam um *dataset* complexo, com mais de 600 naturezas de despesa distribuídas de forma desbalanceada, observaram-se bons resultados. A taxa de acerto da solução criada foi quase idêntica a adquirida pela especialista nos dados, porém, despendendo apenas alguns minutos na predição, ao contrário da análise da especialista que levou mais de um mês.

O SVM obteve o melhor valor de Macro F1, enquanto a estratégia *Stacking* obteve

o melhor valor de Micro F1. Considerando esses resultados e a fim de se priorizar a classificação correta das naturezas de despesas com poucos empenhos, o SVM foi escolhido como algoritmo preditor. Isso é devido ao fato de que o SVM possui um melhor desempenho na predição de naturezas de despesas com essa característica.

Então, foi disponibilizado ao TCE-GO o produto deste trabalho, que é um *script* Python para a criação dos dois modelos de aprendizado de máquina em questão, auxiliando aquele Tribunal a realizar fiscalizações diárias dos empenhos públicos. Nesse contexto, a quantidade de empenhos a serem fiscalizados no período compreendido entre 20 de maio de 2021 a 16 de julho de 2021 caiu de 11.560 para apenas 46. Assim, as fiscalizações, que antes eram ocasionais e custosas, tornaram-se tarefas diárias e mais fáceis de serem gerenciadas. Com isso, agora é possível ao TCE-GO fiscalizar mais rigidamente os gastos públicos, proporcionando uma menor probabilidade de fraudes e descumprimentos da lei.

Referências

BREIMAN, Leo. Random Forests. **Machine Learning**, Springer Science e Business Media LLC, v. 45, n. 1, p. 5–32, 2001. DOI: 10.1023/a:1010933404324. Disponível em: <https://doi.org/10.1023/a:1010933404324>.

CONTAS DO ESTADO DE GOIÁS, Tribunal de. **Competências e Atribuições**. [S.l.: s.n.], 2020. Acessado em: 13/10/2020. Disponível em: <https://portal.tce.go.gov.br/competencias>.

CUNHA, Washington et al. On the cost-effectiveness of neural and non-neural approaches and representations for text classification: A comprehensive comparative study. **Information Processing & Management**, Elsevier, v. 58, n. 3, p. 102481, 2021.

GÉRON, Aurélien. **Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems**. [S.l.]: O'Reilly Media, 2019.

GRACZYK, Magdalena et al. Comparison of bagging, boosting and stacking ensembles applied to real estate appraisal. In: SPRINGER. ASIAN conference on intelligent information and database systems. [S.l.: s.n.], 2010. P. 340–350.

JESUS, Mauricio Barros de et al. Using Text Mining to Categorize the Purpose of Public Spending for the Benefit of Transparency and Accountability. In: IEEE. 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA). [S.l.: s.n.], 2019. P. 263–267.

PANG, Guansong; JIN, Huidong; JIANG, Shengyi. CenKNN: a scalable and effective text classifier. **Data Mining and Knowledge Discovery**, Springer, v. 29, n. 3, p. 593–625, 2015.

REPÚBLICA, Presidência da. **LEI Nº 12.527, DE 18 DE NOVEMBRO DE 2011**. [S.l.: s.n.], 2020. Acessado em: 13/10/2020. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2011/lei/l12527.htm>.

SCIKIT-LEARN. **Cross-validation**. [S.l.: s.n.], 2021. Acessado em: 04/02/2021. Disponível em: <https://scikit-learn.org/stable/modules/cross_validation.html>.

_____. **F1 Score**. [S.l.: s.n.], 2021. Acessado em: 26/05/2021. Disponível em: <https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html>.

SCIKIT-LEARN. **Gaussian Naive Bayes**. [S.l.: s.n.], 2021. Acessado em: 04/02/2021. Disponível em: <https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html?highlight=gaussiannb#sklearn.naive_bayes.GaussianNB>.

_____. **Nearest centroid classifier**. [S.l.: s.n.], 2021. Acessado em: 04/02/2021. Disponível em: <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.NearestCentroid.html>.

_____. **Nearest Neighbors**. [S.l.: s.n.], 2021. Acessado em: 04/02/2021. Disponível em: <https://scikit-learn.org/stable/modules/neighbors.html>.

_____. **Radius Neighbors Classifier**. [S.l.: s.n.], 2021. Acessado em: 04/02/2021. Disponível em: <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.RadiusNeighborsClassifier.html>.

_____. **random forest classifier**. [S.l.: s.n.], 2021. Acessado em: 04/02/2021. Disponível em: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>.

_____. **Stochastic Gradient Descent**. [S.l.: s.n.], 2021. Acessado em: 04/02/2021. Disponível em: <https://scikit-learn.org/stable/modules/sgd.html>.

_____. **Support Vector Machines**. [S.l.: s.n.], 2021. Acessado em: 26/01/2021. Disponível em: <https://scikit-learn.org/stable/modules/svm.html#svm-classification>.

SILVA, Leandro Augusto da; PERES, Sarajane Marques; BOSCARIOLI, Clodis. **Introdução à mineração de dados: com aplicações em R**. [S.l.]: Elsevier Brasil, 2017.

TRANSPARÊNCIA, Portal da. **Execução da despesa pública**. [S.l.: s.n.], 2021. Acessado em: 31/08/2021. Disponível em: <http://www.portaltransparencia.gov.br/entenda-a-gestao-publica/execucao-despesa-publica#:~:text=O%20empenho%20%C3%A9%20a%20etapa,mais%20do%20que%20foi%20planejado.>.

WANG, Haiyang et al. Dense adaptive cascade forest: a self-adaptive deep ensemble for classification problems. **Soft Computing**, Springer Science e Business Media LLC, v. 24, n. 4, p. 2955–2968, mai. 2019. DOI: 10.1007/s00500-019-04073-5. Disponível em: <https://doi.org/10.1007/s00500-019-04073-5>.

YAO, Xin; LIU, Yong. Machine learning. In: SEARCH Methodologies. [S.l.]: Springer, 2014. P. 477–517.

Appendices

A Questionário relativo ao processo de exploração dos dados do TCE-GO

Exercício do orçamento (Ano)(EOF)

Esse ano se refere ao ano do processo ou do empenho?

Refere-se à data de emissão do empenho.

Ele é data limite ou data de abertura?

Refere-se à data de abertura do empenho que está relacionado ao ano do orçamento. Os quatro primeiros dígitos do sequencial do empenho sempre coincidirá com esse ano.

Esse campo é relacionado com o processo?

Não tem correlação necessária com processo.

Órgão (Código/Nome)(EOF)

Esse campo órgão se refere ao órgão que abriu o processo original?

Como dito em reunião, o processo não tem tanta correlação com os códigos orçamentários e financeiros. Esse órgão corresponde a quem emitiu o empenho e que, portanto, tinha dotação orçamentária disponível para tal.

Esse campo é derivado de outro?

Eu diria que é derivado do orçamento.

Órgão Sucessor Atual (Código/Nome)(EOF)

Por que existem órgãos deste campo que não estão presentes no campo "Órgão (Código/Nome)(EOF)"

Quando há alguma reforma administrativa no Estado alguns órgãos deixam de existir e outros assumem as contas.

Esse campo é relacionado com o empenho ou o processo?

Esse campo está relacionado com o orçamento/empenho.

Tipo Administração (Nome)(EOF)

Constatou-se que geralmente está relacionado com o órgão, ou seja, o órgão faz parte da administração direta ou indireta com exceção do órgão "2953 - FUNEBOM", que tem 106 documentos indiretos e 837 diretos. Essa relação existe? Se sim, por que o órgão "2953 - FUNEBOM" não a segue?

Sim, os fundos normalmente seguem a classificação do órgão/entidade ao qual está vinculado: podem ter fundos da administração direta e fundos da administração indireta. Entendo que o órgão "2953 - FUNEBOM" é um fundo da administração direta, todavia não sei explicar o motivo de ter documentos nas duas classificações.

O órgão Detran possui dois códigos: 2961 e 5901. Por quê?

Com relação a um mesmo órgão possuir códigos diferentes, se relaciona com as constantes reformas administrativas que ocorrem no estado, muitas delas no decorrer de um exercício. Normalmente a busca no campo "Órgão Sucessor Atual (Código/Nome)(EOF)" elimina essa duplicidade e sempre apresenta o código válido para o órgão na atualidade.

Esse campo está relacionado com o empenho ou o processo?

Esse campo está relacionado com o orçamento/empenho.

Tipo Poder (Nome)(EOF)

No dicionário se refere a apenas três tipos de poderes: Executivo, Judiciário e Legislativo, porém, os dados contem mais dois tipos de poder: Ministério Público e Defensoria Pública. Está correto?

Sim. Realmente tem mais esses órgãos autônomos.

Esse campo está relacionado com o empenho ou o processo?

Esse campo está relacionado com o orçamento/empenho.

Função (Cod/Nome)(EOF)

A função se relaciona com a missão institucional do órgão, então, quer dizer que a Secretaria de Saúde, por exemplo, só pode ter empenhos na função "Saúde"?

Não, pode ser que o órgão execute mais de uma função.

Esse campo está relacionado com o empenho ou o processo?

Esse campo está relacionado com o orçamento/empenho.

Subfunção (Cod/Nome)(EOF)

Qual é a relação entre função e subfunção, uma subfunção pertence somente a uma função?

A função tem subfunções típicas, mas a subfunção não está obrigatoriamente atrelada à função.

Existe uma relação hierárquica entre os campos "Função (Cod/Nome)(EOF)", "Subfunção (Cod/Nome)(EOF)", "Programa (Cod/Nome)(EOF)", "Ação (Cod/Nome)(EO "Grupo Despesa (Cod/Nome)(EOF)", "Elemento Despesa (Cod/Nome)(EOF)" e "Fonte Recurso (Cod)(EOF)" ?

Função, Subfunção, Programa, Ação, Grupo Despesa, Elemento Despesa, essa sequência está em ordem decrescente de níveis de agregação da despesa, não necessariamente uma hierarquia. A Fonte de Recurso também pode ser considerada um agregador macro.

Esse campo está relacionado com o empenho ou o processo?

Esse campo está relacionado com o orçamento/empenho.

Programa (Cod/Nome)(EOF)

O que é programa?

O programa é o terceiro nível de agregação da despesa. Ele é estabelecido no Plano Plurianual e tem validade de quatro anos. Está relacionado com o plano de governo do chefe do executivo e a funções de estado específicas dos demais poderes e órgãos autônomos.

Qual a relação entre Programa e Função/Subfunção ?

Normalmente desenvolve aspectos específicos de Funções/Subfunções, mas não tem necessária hierarquia/subordinação a essas.

Esse campo está relacionado com o empenho ou o processo?

Esse campo está relacionado com o orçamento/empenho.

Ação (Cod/Nome)(EOF)

Qual a relação entre ação e programa?

A Ação é o desdobramento direto do Programa, ela sempre está relacionada a um Programa que pode ter mais de uma Ação vinculada a ele.

Uma ação pode ser de mais de um programa?

Normalmente uma ação pertence a um programa somente.

Esse campo está relacionado com o empenho ou o processo?

Esse campo está relacionado com o orçamento/empenho.

Grupo Despesa (Cod/Nome)(EOF)

Esse campo está relacionado com o empenho ou o processo?

Esse campo está relacionado com o orçamento/empenho.

Elemento Despesa (Cod/Nome)(EOF)

Esse campo está relacionado com o empenho ou o processo?

Esse campo está relacionado com o orçamento/empenho.

Formalidade (Nome)(EOF)

A descrição do campo ainda não deixou claro o que ele significa

Trata-se de agrupador gerencial que auxilia no registro das transações no SIOFINET e no Sistema de Contabilidade Geral.

Esse campo está relacionado com o empenho ou o processo?

Esse campo está relacionado com o orçamento/empenho.

Modalidade Licitação (Nome)(EOF)

226172 dos 324728 documentos têm a modalidade "Não Aplicavel". O que isso significa?

Muitas despesas não dependem de licitação. Todas das despesas com pessoal, por exemplo, independem de licitação. Por isso a quantidade de "Não aplicável".

Esse campo está relacionado com o empenho ou o processo?

Esse campo está relacionado com o orçamento/empenho.

Fonte Recurso (Cod)(EOF)

Esse campo está relacionado com o empenho ou o processo?

Esse campo está relacionado com o orçamento/empenho.

Fonte Recurso (Nome)(EOF)

Existe uma discrepância entre a quantidade de códigos de fontes de recursos (50) e a quantidade de nomes de fontes de recursos (30). Sendo assim, como inferir o nome da fonte de recurso através do seu código?

Verificar os dados encaminhados com o pessoal de TI. Pode ser um erro da carga de dados

Esse campo está relacionado com o empenho ou o processo?

Esse campo está relacionado com o orçamento/empenho.

Valor Estorno Anulação Empenho(EOF) e Valor Anulação Cancelamento Empenho(EOF)

No período de 2015 a parte de 2020, não houve nenhuma ocorrência desses campos, isso é raro mesmo ou é atípico nesses dois campos e em outros campos referentes a valores?

É raro, mas é necessário prever a operação caso seja necessário.

De forma geral sobre os valores do empenho sabemos que os valores dos empenhos podem ser alterados, mas não sabemos ainda como isso é feito, nem em que momento. Por exemplo: quando há alguma alteração nesses campos, onde essa alteração fica registrada e, mais importante, o valor anterior é substituído?

Os documentos de empenho não são alterados, aí incluindo valores e demais campos. Uma vez emitido ele permanece da mesma forma. O saldo do empenho pode ser alterado pelos documentos de: "Valor Anulação Empenho(EOF)" (negativo), "Valor Estorno Anulação Empenho(EOF)" (positivo), "Valor Cancelamento Empenho(EOF)" (negativo), "Valor Anulação Cancelamento Empenho(EOF)" (positivo).

Se existirem, quais são as fórmulas para se chegar nos campos de valores seguintes?

Valor Saldo do Empenho(EOF)

Fórmula: Empenho - Anulação Empenho + Estorno Anulação Empenho - Cancelamento Empenho + Anulação Cancelamento Empenho = Valor Saldo do Empenho.

Valor Saldo Liquidado(EOF)

Fórmula: Liquidação Empenho - Cancelamento Liquidação Empenho = Saldo Liquidado

Valor Saldo Pago(EOF)

Fórmula: Ordem de Pagamento - Guia Recolhimento - Anulação Ordem de Pagamento + Estorno Anulação O. Pagamento + Estorno Guia Recolhimento = Saldo Pago.

Valor Saldo a Pagar(EOF)

Fórmula: Saldo do Empenho - Saldo Pago = Saldo a Pagar.

Valor a Liquidar(EOF)

Fórmula: Saldo do Empenho - Saldo Liquidado = Valor a Liquidar.

Valor a Pagar Liquidado(EOF)

Fórmula: Saldo Liquidado - Saldo Pago = Valor a Pagar Liquidado.

Os demais campos de valores listados a seguir não possuem código e se tratam da soma dos documentos referentes a cada campo.

- Valor Guia Recolhimento(EOF)
- Valor Anulação Ordem de Pagamento(EOF)
- Valor Estorno Anulação O. Pagamento(EOF)
- Valor Estorno Guia Recolhimento(EOF)
- Valor Anulação Empenho(EOF)
- Valor Estorno Anulação Empenho(EOF)
- Valor Cancelamento Empenho(EOF)
- Valor Anulação Cancelamento Empenho(EOF)
- Valor Liquidação Empenho(EOF)
- Valor Anulação Liquidacao Empenho(EOF)

O que é esse "(EOF)" no final de todos os nomes dos campos?

EOF – Execução Orçamentária e Financeira. Trata-se de um Universo criado no B.O. da seção economia para a geração de relatórios gerenciais da execução orçamentária e financeira. Para o presente caso outros universos também podem ser utilizados como: "Empenho", "Liquidação" e "Ordem de Pagamento", os quais contêm basicamente a mesma estrutura, mas são tratados separadamente e passíveis de conjugação mediante chaves específicas.

B Observações dos dados através da utilização da biblioteca Pandas.

Tabela 12 – Tabela das análises realizadas com o auxilio da biblioteca Pandas.

Campos	**Observações adicionais**	**Decisão da Exploração de Dados**	**Atributos para Geração dos Modelos**
Exercício do orçamento (Ano) (EOF)	2015 a 2020.	Campo removido devido a ser um campo de data que não se repetirá, ou seja, não forma padrões com os dados.	0
Órgão (Código/Nome) (EOF)	165 valores diferentes. 2850 - FUNDO ESTADUAL DE SAUDE- FES; 2201 - GAB.DO SEC. DE EDUCAÇÃO, CULTURA E ESP; 0452 - FUNDESP-PJ; 6606 - UNIVERSIDADE ESTADUAL DE GOIÁS; 0701 - GAB. DO PROCURADOR GERAL DE JUSTICA.	O órgão sucessor será utilizado para representar essas duas colunas e quando houver uma divergência entre o órgão e órgão sucessor atual, essa informação será representada em um novo atributo "órgão_sucedido" que terá valor 1 caso o órgão sucessor atual seja diferente do órgão, ou seja. o órgão tenha sido sucedido	1

Campos	**Observações adicionais**	**Decisão da Exploração de Dados**	**Atributos para Geração dos Modelos**
Órgão Sucessor Atual (Código/Nome) (EOF)	80 valores diferentes sendo os mesmo órgãos do campo anterior com o exceção de 3 novos: 1752 - FUNDO DE APORTE À CELG D. S.A - FUNAC; 3151 - FUNDO ESP.IMPLA PROG.VEIC .LEVE S/TRILHOS; 3052 - FECAD.	O órgão sucessor foi transformado através da estratégia OHE gerando 80 novas colunas	80

Campos	**Observações adicionais**	**Decisão da Exploração de Dados**	**Atributos para Geração dos Modelos**
Tipo Administração (Nome)(EOF)	2 valores diferentes: "Direta"e "Indireta"; Há 106417 documentos com "Indireta"e 218311 com "Direta". Dos que possuem "Indireta", constatou-se que geralmente está relacionado com o órgão, ou seja, o órgão faz parte da administração direta ou indireta com exceção do órgão ``2953 - FUNEBOM", que tem 106 documentos indiretos e 837 diretos. Além disso, o Detran possui dois códigos: 2961 e 5901.	Transformado através da estratégia OHE gerando 2 novas colunas	2
Tipo Poder (Nome)(EOF)	5 tipos diferentes: EXECUTIVO; JUDICIÁRIO; LEGISLATIVO; MINISTÉRIO PÚBLICO; DEFENSORIA PÚBLICA.	Transformado através da estratégia OHE gerando 5 novas colunas	5

Campos	**Observações adicionais**	**Decisão da Exploração de Dados**	**Atributos para Geração dos Modelos**
Classificação orçamentária (Descrição)	4879 valores diferentes. Todos os 324.728 documentos possuem classificação orçamentária com oito códigos. Contudo, sabe-se que apenas o oitavo código possui comprimentos diferentes, com dois ou três dígitos. Existem empenhos com números de processo e naturezas de despesas diferentes porém com mesmo número de classificação orçamentária.Todos os códigos dessa classificação estão em outras colunas.	Campo removido devido a já ter todos seus valores presentes em outros campos	0
Função (Cod/Nome) (EOF)	25 valores diferentes 04 - ADMINISTRAÇÃO; 10 - SAÚDE; 19 - CIÊNCIA E TECNOLOGIA; 12 - EDUCAÇÃO; 06 - SEGURANÇA PÚBLICA.	Transformado através da estratégia OHE gerando 25 novas colunas	25

Campos	**Observações adicionais**	**Decisão da Exploração de Dados**	**Atributos para Geração dos Modelos**
Subfunção (Cod/Nome) (EOF)	78 valores diferentes 122 - ADMINISTRAÇÃO GERAL; 061 - AÇÃO JUDICIÁRIA; 272 - PREVIDENCIA DO REGIME ESTATUTARIO. 091 - DEFESA DA ORDEM JURÍDICA; 032 - CONTROLE EXTERNO.	Transformado através da estratégia OHE gerando 76 novas colunas	76

Campos	**Observações adicionais**	**Decisão da Exploração de Dados**	**Atributos para Geração dos Modelos**
Programa (Cod/Nome) (EOF)	226 valores diferentes. 4001 - PROGRAMA APOIO ADMINISTRATIVO; 0000 - ENCARGOS ESPECIAIS; 1028 - PROGRAMA PROMOÇÃO, PREVENÇÃO E PROTEÇÃO A ASSISTÊNCIA INTEGRAL À SAÚDE; 1057 - PROGRAMA ASSISTÊNCIA A SAÚDE DOS USUÁRIOS DO IPASGO; 1064 - PROGRAMA PESQUISA CIENTÍFICA, TECNOLÓGICA E DE INOVAÇÃO.	Ação e Programa tiveram seus códigos removidos devido a ter códigos diferentes para o mesmo nome e então foram concatenados juntos gerando o novo campo “ação_Programa” o novo campo gerado foi transformado através da estratégia OHE gerando 897 novas colunas.	0

Campos	**Observações adicionais**	**Decisão da Exploração de Dados**	**Atributos para Geração dos Modelos**
Ação (Cod/Nome) (EOF)	980 valores diferentes. 4001 - APOIO ADMINISTRATIVO; 2310 - IMPLEMENTAÇÃO DE SERVIÇOS DE ASSISTÊNCIA À SAÚDE; 7001 - ENCARGOS COM INATIVOS E PENSIONISTAS; 7006 - ENCARGOS JUDICIÁRIOS; 2023 - EXERCÍCIO DO CONTROLE EXTERNO DA ADMINISTRAÇÃO PÚBLICA ESTADUAL.		897

Campos	**Observações adicionais**	**Decisão da Exploração de Dados**	**Atributos para Geração dos Modelos**
Grupo Despesa (Cod/Nome) (EOF)	6 valores diferentes: 3 - OUTRAS DESPESAS CORRENTES; 1 - PESSOAL E ENCARGOS SOCIAIS; 4 - INVESTIMENTOS; 2 - JUROS E ENCARGOS DA DÍVIDA; 5 - INVERSÕES FINANCEIRAS; 6 - AMORTIZAÇÃO DA DÍVIDA.	Como elemento faz parte do Rótulo então ele é retirado para não influenciar o modelo	0

Campos	**Observações adicionais**	**Decisão da Exploração de Dados**	**Atributos para Geração dos Modelos**
Elemento Despesa (Cod/Nome) (EOF)	51 valores diferentes. 11 - VENCIMENTOS E VANTAGENS FIXAS - PESSOAL CIVIL; 39 - OUTROS SERVIÇOS DE TERCEIROS - PESSOA JURÍDICA; 30 - MATERIAL DE CONSUMO; 92 - DESPESAS DE EXERCÍCIOS ANTERIORES; 13 - OBRIGAÇÕES PATRONAIS; 41 - CONTRIBUIÇÕES.	campo excluido devido a esta associado com o rotulo	0
Natureza Despesa (Cod)	650 valores diferentes. 3.1.90.11.10; 3.3.90.18.05; 3.3.90.92.23; 3.3.90.30.09; 3.3.90.93.02.	A junção de todos os códigos deste campo representam a natureza de despesa portanto este campo será nosso RÓTULO	0

Campos	**Observações adicionais**	**Decisão da Exploração de Dados**	**Atributos para Geração dos Modelos**
Natureza Despesa (Nome)	547 valores diferentes. Vencimentos e Salários; Bolsa de Estudo para alunos de Graduação e Pós-Graduação; Indenizações e Restituições; Gêneros Alimentícios; Restituições Diversas.	Nome foi retirado pois já é representado pelo código do elemento e subelemento de despesa presente no campo ``Natureza Despesa (Cod)"	0
Formalidade (Nome)(EOF)	18 valores diferentes. Outras; Contratos; Folha de Pagamento - Regime Próprio - Descontos; Folha de Pagamento - Regime Próprio - Líquido; Folha de Pagamento - Outros.	Transformado através da estratégia OHE gerando 17 novas colunas	17
Modalidade Licitação (Nome)(EOF)	11valores diferentes Nao Aplicavel; Pregao; Dispensa Licitacao; Licitacao Inexigivel; Suprimento de Fundos.	Transformado através da estratégia OHE gerando 11 novas colunas	11

Campos	**Observações adicionais**	**Decisão da Exploração de Dados**	**Atributos para Geração dos Modelos**
Fonte Recurso (Cod)(EOF)	50 valores numéricos diferentes 100; 0; 220; 20; 8.	Como existe uma inconsistência entre o código e o nome utilizaremos apenas o nome da fonte de recurso	0
Fonte Recurso (Nome)(EOF)	30 valores diferentes RECEITAS ORDINARIAS; RECURSOS DIRETAMENTE ARRECADADOS; RECURSOS DO FUNDEB (E.C. Nº 53, DE 19/12/2006)); TRANSFERENCIAS CORRENTES (UNIAO); CONVENIOS, AJUSTES E ACORDOS COM ÓRGAOS FEDERAIS.	Transformado através da estratégia OHE gerando 29 novas colunas	29

Campos	**Observações adicionais**	**Decisão da Exploração de Dados**	**Atributos para Geração dos Modelos**
Beneficiário (CNPJ)		Todos esses campos foram removidos e um novo campo gerado "pessoa_juridica", onde o valor 1 significa que a pessoa é jurídica e 0 caso contrário.	1
Beneficiário (CPF)			
Beneficiário (CPF/CNPJ)			
Beneficiário (Nome)		Esse campo foi tratado substituindo o nome das pessoas físicas por "PF"e em seguida tratado utilizando a representação TF-IDF	7445
Período (Dia/ Mes /Ano) (EOF)	1245 valores diferentes 42396; 43915; 43858; 43859; 43816.	O campo teve seus valores convertidos do formato de data do excel para data no formado YYYY/DD/MM. Após isso, apenas o mês foi extraído e usado como valor do campo, pois mês é um valor que se repete.	1

Campos	**Observações adicionais**	**Decisão da Exploração de Dados**	**Atributos para Geração dos Modelos**
Empenho (Número do Processo)	84113 valores diferentes Esse campo não tem uma padronização em seu código pois cada órgão utiliza de uma forma: 201500027000266; 201712404000186; 201700047000659; 2015003485; 202000020001198.	Campo removido devido a não ter uma padronização dos códigos pois cada órgão utiliza de uma forma. Foi criado a partir desse campo um campo "empenhos_por_processo"onde diz para cada numero de processo quantos empenhos ele possui	1
Empenho (Sequencial Empenho)	Um valor diferente para cada empenho. 2020.1704.009.00262; 2019.1762.013.00018; 2017.2701.001.00467.	Como o campo possui um valor diferente para cada empenho este campo não é representativo para o nosso modelo	0
Empenho (Histórico)	297857 valores diferentes (campo de texto livre)	O texto foi processado fazendo a limpeza retirando acentuações, letras maiúsculas, caracteres especiais, stopwords e então transformado utilizado a estratégia TF-IDF gerando uma nova "visão"dos dados	0
Valor Empenhado (EOF)	187018 valores diferentes	Já em formato numérico	1
Valor Anulação Empenho (EOF)	42487 valores diferentes	Já em formato numéricoCampo removido devido a ter apenas o valor 0 para todos os empenhos	1

Campos	**Observações adicionais**	**Decisão da Exploração de Dados**	**Atributos para Geração dos Modelos**
Valor Estorno Anulação Empenho (EOF)	apenas valor 0 para todos os empenhos	Já em formato numérico	0
Valor Cancelamento Empenho (EOF)	21117 valores diferentes	Já em formato numérico	1
Valor Anulação Cancelamento Empenho (EOF)	apenas valor 0 para todos os empenhos	Já em formato numéricoCampo removido devido a ter apenas o valor 0 para todos os empenhos	0
Valor Saldo do Empenho (EOF)	184628 valores diferentes	Já em formato numérico	1
Valor Liquidação Empenho(EOF)	186952 valores diferentes	Já em formato numérico	1
Valor Anulação Liquidacao Empenho(EOF)	25120 valores diferentes	Já em formato numérico	1
Valor Saldo Liquidado(EOF)	183345 valores diferentes	Já em formato numérico	1
Valor Ordem de Pagamento(EOF)	182462 valores diferentes	Já em formato numérico	1

Campos	**Observações adicionais**	**Decisão da Exploração de Dados**	**Atributos para Geração dos Modelos**
Valor Guia Recolhimento (EOF)	8477 valores diferentes	Já em formato numérico	1
Valor Anulação Ordem de Pagamento(EOF)	5971 valores diferentes	Já em formato numérico	1
Valor Estorno Anulação O. Pagamento(EOF)	33 valores diferentes	Já em formato numérico	1
Valor Estorno Guia Recolhimento (EOF)	358 valores diferentes	Já em formato numérico	1
Valor Saldo Pago(EOF)	181119 valores diferentes	Já em formato numérico	1
Valor Saldo a Pagar(EOF)	11401 valores diferentes	Já em formato numérico	1
Valor a Liquidar(EOF)	6452 valores diferentes	Já em formato numérico	1
Valor a Pagar Liquidado(EOF)	6354 valores diferentes	Já em formato numérico	1
		Total:	**8607**