CEFET/RJ
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO

# APRIMORAMENTO DA DETECÇÃO DE CONLUIOS COM APRENDIZADO DE MÁQUINA: EXPLICABILIDADE E ANÁLISE DOS RESULTADOS

Lucas Daflon Scoralick

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Ciência da Computação, Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ, como parte dos requisitos necessários à obtenção do grau de mestre.

Orientadores:
Kele Teixeira Belloze

Rio de Janeiro,

8 de Fevereiro de 2025

**Aprimoramento da Detecção de Conluios com Aprendizado de Máquina: Explicabilidade e Análise dos Resultados**

Dissertação de Mestrado em Ciência da Computação, Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, CEFET/RJ.

Lucas Daflon Scoralick

Aprovada por:

---

Presidente, Profa. Kele Teixeira Belloze, D.Sc. - Cefet/RJ

---

Diego Nunes Brandão, D.Sc. - Cefet/RJ

---

Eduardo Bezerra da Silva, D.Sc. - Cefet/RJ

---

Flavia Cristina Bernardini, D.Sc. - UFF

Rio de Janeiro,

8 de Fevereiro de 2025

Ficha catalográfica elaborada pela Biblioteca Central do CEFET/RJ

S423 Scoralick, Lucas Daflon
Aprimoramento da detecção de conluios com aprendizado de máquina: explicabilidade e análise dos resultados / Lucas Daflon Scoralick. — 2024.
64f. : il. color. , enc.

Dissertação (Mestrado) Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, 2024.
Bibliografia : f. 60-64
Orientador: Kele Teixeira Belloze

1. Aprendizado de máquina. 2. Cartéis. 3. Licitação pública. I. Belloze, Kele Teixeira. (Orient.). II. Título.

CDD 006.31

Elaborada pela bibliotecária Tania Mello – CRB/7 nº 5507/04

# DEDICATÓRIA

Dedico este trabalho especialmente à minha filha, nascida em 2023, fonte de inspiração e alegria, e à minha querida avó, minha segunda mãe, que nos deixou em 2024.

# AGRADECIMENTOS

Agradeço à minha família pelo amor, apoio, incentivo e compreensão pelas horas necessárias para focar neste estudo. Minha gratidão também à minha orientadora, cuja orientação, paciência, tempo para acompanhamento e empatia foram essenciais para superar as dificuldades técnicas e emocionais. Muito obrigado a todos.

# RESUMO


Aprimoramento da Detecção de Conluios com Aprendizado de Máquina: Explicabilidade e Análise dos Resultados

Lucas Daflon Scoralick

Orientadores:
Kele Teixeira Belloze

Resumo da Dissertação submetida ao Programa de Pós-graduação em Ciência da Computação do Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ como parte dos requisitos necessários à obtenção do grau de mestre.

Conluios são acordos ou combinações secretas entre duas ou mais partes, geralmente para ganhar vantagem em algo ou prejudicar terceiros. A prática de conluios em licitações públicas perturba o equilíbrio de preços do mercado, impactando negativamente tanto os custos quanto a qualidade dos serviços públicos. Neste estudo, propõe-se uma metodologia para aprimorar os modelos de classificação de conluio, utilizando variáveis estatísticas combinadas com a análise de modelos explicáveis para explicação dos resultados. Os resultados mostraram uma melhora no desempenho de 0,04 à 0,25, utilizando a estratégia de balanceamento *Synthetic Minority Over-sampling Technique* (Smote), método de validação *holdout* 80% e 20% (treinamento e teste) e *Matthew's correlation coefficient* (MCC) como métrica de desempenho. A utilização dos modelos explicáveis *SHapley Additive exPlanation* (SHAP), importância de atributos e *Local Interpretable Model-agnostic Explanations* (LIME) foram úteis para confirmar a contribuição das variáveis estatísticas para os modelos de aprendizagem de máquina. Além disso, a explicação de amostras com LIME demonstra potencial para explicação dos limites de decisão de classificação para atributos numéricos.



Palavras-chave:
Aprendizado de máquina, conluio, cartel, variáveis estatísticas, modelos explicáveis


Rio de Janeiro,

8 de Fevereiro de 2025

# ABSTRACT

Aprimoramento da Detecção de Conluios com Aprendizado de Máquina: Explicabilidade e Análise dos Resultados

Lucas Daflon Scoralick

Advisors:
Kele Teixeira Belloze

Abstract of dissertation submitted to Programa de Pós-graduação em Ciência da Computação - Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ as partial fulfillment of the requirements for the degree of master.

Collusions are secret agreements or coding between two or more parties, usually to gain an advantage in something or harm others. The practice of collusion in public tenders disturbs the market price balance, impacting both the costs and the quality of public services. This study proposes a methodology to improve completion classification models, using statistics variables associated with the analysis of explainable models to explain the results. The results showed an improvement in performance from 0.04 to 0.25, using the Smote balancing strategy, *holdout* 80% and 20% validation method (training and testing), and MCC as a performance metric. Using the SHAP, feature importance, and LIME explainable models helped confirm the contribution of statistics to the machine learning models. Furthermore, explaining samples with LIME demonstrates the potential for explaining classification decision boundaries for numerical attributes.

Key-words:
Machine learning, collude, cartel, statistics variables, explainable models

Rio de Janeiro,

8 de Fevereiro de 2025

**Sumário**


**I Introdução** **17**

I.1 Contextualização e Problema 17

I.2 Objetivo 20

I.3 Metodologia 21

I.4 Organização do Trabalho 21

**II Fundamentação teórica** **22**

II.1 Estatística descritiva 22

II.2 Descoberta de conhecimento em base de dados 24

II.2.1 Pré-processamento 24

II.2.2 Mineração de dados 30

II.2.3 Pós-processamento 33

II.3 Métricas de avaliação 33

II.3.1 Métricas de classificação 33

II.4 Modelos explicáveis 34

II.4.1 Importância de atributos 35

II.4.2 LIME 36

II.4.3 SHAP 37

**III Trabalhos relacionados** **38**

III.1 Detecção de conluio 39

III.2 Modelos explicáveis 42

III.3 Aprimoramento dos modelos de classificação 44

III.4 Considerações sobre trabalhos relacionados 45

**IV Metodologia** **46**

IV.1 Coleta dos dados 46

IV.1.1 Dados de licitação 46

IV.1.2 Dados de conluio 48

IV.2 Pré-processamento 48

IV.2.1 Seleção de dados 48
IV.2.2 Seleção de atributos 49
IV.2.3 Limpeza 50
IV.2.4 Extração de atributos 51
IV.2.5 Rotulação de conluios 51
IV.2.6 Enriquecimento (variáveis estatísticas) 52
IV.3 Preparação para aprendizagem 52
IV.3.1 Particionamento 52
IV.3.2 Codificação 52
IV.3.3 Normalização 53
IV.3.4 Correção de prevalência 53
IV.4 Classificação 53
IV.5 Avaliação de desempenho 54
IV.6 Explicação do modelo 54
IV.6.1 Importância dos atributos 54
IV.6.2 SHAP 55
IV.7 Explicação amostra com LIME 55
IV.8 Utilização da metodologia 55

**V Experimentos e resultados 57**
V.1 Coleta dos dados 57
V.1.1 Dados de licitação 57
V.1.2 Dados de conluio 58
V.2 Pré-processamento 58
V.2.1 Avaliações sobre conjunto de dados 58
V.2.2 Balanceamento e seleção de atributos 59
V.3 Avaliação do desempenho dos algoritmos 61
V.4 Importância de atributos 62
V.4.1 Experimentos com importância de atributos 63
V.4.2 Experimentos com SHAP 65
V.5 Experimentos com LIME 67
V.6 Discussão dos resultados 68

**VI Conclusões e trabalhos futuros 71**
VI.1 Dificuldades 73
VI.2 Limitações 74

VI.3 Trabalhos futuros 75
Referências 76

## Lista de Figuras


II.1 Etapas operacionais do processo de *Knowledge Discovery in Databases* (KDD) [Goldschmidt and Passos, 2005] 24

II.2 Visão geral das técnicas de seleção de atributos. 27

II.3 Exemplo de esquema de validação cruzada tripla, retirado de Kuhn and Johnson [2013] 30

II.4 Modelo esquemático do *AdaBoost* [Hastie et al., 2013]. 32

II.5 Exemplo da intuição do LIME, apresentada por Ribeiro et al. [2016]. 36

III.1 Pesquisa bibliográfica. (1) Pesquisa inicial sobre aplicações de modelos explicáveis e conluios não retornou artigos. (2) Pesquisa mais abrangente por modelos explicáveis, abrindo a possibilidade de outras aplicações, retornou 185 artigos, mas nenhum relacionado a conluios. (3) Pesquisa mais abrangente sobre métodos de detecção de conluio, retornou diversos artigos, sendo que foram selecionados 7 artigos de interesse. (4) Pesquisa sobre utilização de variáveis estatísticas para aprimorar desempenho, mostra outros cenários que essa configuração foi experimentada. 38

IV.1 Metodologia para aprimoramento de classificação de conluios. 46

IV.2 Atributos existentes em cada tipo de arquivo baixado do portal da transparência. Sublinhado os atributos que são comuns a todos os arquivos. 47

IV.3 Processo para coleta e junção dos arquivos baixados do portal de transparência para os anos de 2013 até 2018. 47

IV.4 Visão geral das atividades do pré-processamento. 49

IV.5 Visão geral do resultado das atividades do pré-processamento sobre o conjunto de dados. 50

IV.6 Processo para rotulação dos conluios. 51

V.1 Totais dos registros de cada tabela por ano. 57

V.2 Importância de cada atributo pelas diversas abordagens de seleção de atributos. 60

V.3 Execução do método de seleção de importância para os algoritmos (01) *K-Nearest Neighbors* (K-NN); (02) *Multi Layer Perceptron* (MLP); (03) *Support Vector Machine* (SVM); (04) Regressão Logística; (05) *eXtreme Gradient Boosting* (XGBoost); (06) *Extra Trees*; (07) *Ridge*; (08) *Stochastic Gradient Descent* (SGD). 63

V.4 Execução do método do SHAP para os algoritmos (01) K-NN; (02) MLP; (03) SVM; (04) Regressão Logística; (05) XGBoost; (06) *Extra Trees*; (07) *Ridge*; (08) SGD. 65

V.5 Execução do método do LIME para quatro observações: classificada como 'Sim' para conluio e o modelo previu 'Sim'; classificada como 'Sim', mas o modelo previu 'Não; classificada como 'Não', mas o modelo previu 'Sim'; classificada como 'Não' e o modelo previu 'Não'. 67

## Lista de Tabelas


II.1 Dimensões mais citadas da qualidade de dados, por [Haug et al., 2011] de [Wang et al., 1995]. 25

II.2 Exemplo de codificação para o *Leave-One-Out Encoder*. 29

II.3 Matriz de confusão. 34

III.1 Telas estatísticas utilizados nos trabalhos relacionados. Para efeito de comparação, foi adicionada uma coluna indicando com um X os algoritmos abordados neste trabalho. 41

III.2 Métricas de avaliação utilizadas nos trabalhos relacionados. 42

III.3 Algoritmos de aprendizado de máquina utilizados nos trabalhos relacionados. A métrica citada é acurácia. Para efeito de comparação, foi adicionada uma coluna indicando com um X os algoritmos abordados neste trabalho. 43

IV.1 Algoritmos executados e suas respectivas funções e bibliotecas do *python*. 54

V.1 Levantamento dos processos de condenação do CADE, a partir de 2013. 58

V.2 Visão geral dos atributos para o conjunto de dados. Os atributos são avaliados sob três variáveis: quantidade de valores únicos; quantidade de valores vazios; variância. Total de registros: **43.796.011** 59

V.3 Abordagens de correção de prevalência sem e com variáveis estatísticas. Resultados com execução dos algoritmos MLP, Regressão Logística (RL) e *Random Forest*. Os resultados em negrito são referentes ao melhor desempenho de cada algoritmo, por estratégia de balanceamento. 60

V.4 Comparação dos resultados com e sem variáveis estatísticas dos algoritmos Árvore de Decisão; *Extra Trees*; *Gradient Boosting*; K-NN; *Lasso*; Regressão Logística; *Naive Bayes (Bernoulli)*; *Naive Bayes (Gaussian)*; MLP; *Quadratic Discriminant Analysis* (QDA); *Random Forest*; *Ridge*; SGD; SVM; OneClassSVM; e XGBoost. Os melhores resultados estão destacados em negrito. As estratégias de balanceamento utilizadas são nenhuma, Smote e SMOTE + ENN (Smoteenn). 62

V.5 Ordem de importância de cada atributo, segundo método de importância de atributos, para os algoritmos (01) K-NN; (02) MLP; (03) SVM; (04) Regressão Logística; (05) XGBoost; (06) *Extra Trees*; (07) *Ridge*; (08) SGD. Os melhores resultados estão destacados em negrito. 64

V.6 Tempos de execução dos métodos de importância de atributos, para os algoritmos (01) K-NN; (02) MLP; (03) SVM; (04) Regressão Logística; (05) XGBoost; (06) *Extra Trees*; (07) *Ridge*; (08) SGD. Os maiores tempos estão destacados em negrito. 64

V.7 Ordem de importância de cada atributo, segundo SHAP, para os algoritmos (01) K-NN; (02) MLP; (03) SVM; (04) Regressão Logística; (05) XGBoost; (06) *Extra Trees*; (07) *Ridge*; (08) SGD. Os melhores resultados estão destacados em negrito. 66

V.8 Tempos de execução dos métodos de SHAP para os algoritmos (01) K-NN; (02) MLP; (03) SVM; (04) Regressão Logística; (05) XGBoost; (06) *Extra Trees*; (07) *Ridge*; (08) SGD. Os maiores tempos estão destacados em negrito. 66

V.9 Matriz de confusão da execução do K-NN com método de balanceamento Smote e variáveis estatísticas. 68

## Lista de Abreviações

| | | |
|---|---|---|
| API | *Application Interface Programming* | 73 |
| AUC | *Area Under the Curve* | 35, 42, 43, 54 |
| CADE | Conselho Administrativo De Defesa Econômico | 17, 18, 19, 21, 46, 48, 58, 68 |
| CGU | Controladoria Geral Da União | 18, 46, 48, 57 |
| CV | Coeficiente De Variação | 22, 23, 40, 41, 45, 52, 53, 60, 63, 64, 65, 66, 67, 69, 72, 73 |
| DALEX | *MoDel Agnostic Language for Exploration and eXplanation* | 42 |
| DIFFA | Diferença Absoluta | 41, 75 |
| DIFFP | Diferença Percentual | 41, 45, 73, 75 |
| DP | Desvio Padrão | 41, 44 |
| ENN | *Edited Nearest Neighbours* | 29 |
| FN | *False Negative* | 34 |
| FP | *False Positive* | 34 |
| IA | Inteligência Artificial | 30, 34, 35, 72 |
| IIA | Instituto Dos Auditores Internos | 19 |
| K-NN | *K-Nearest Neighbors* | 29, 31, 43, 44, 45, 53, 54, 55, 62, 63, 64, 65, 66, 68, 72 |
| KDD | *Knowledge Discovery in Databases* | 21, 22, 24, 26, 46 |
| KS | Kolmogorov-Smirnov | 41, 45, 73, 75 |
| KURT | *Kurtosis* Ou Curtose | 23, 40, 41, 44, 45, 52, 53, 60, 63, 64, 65, 66, 67, 69, 72, 73 |
| LDA | *Linear Discriminant Analysis* | 32, 44 |
| LGBM | *Light Gradient-Boosting Machine* | 44 |
| LIME | *Local Interpretable Model-agnostic Explanations* | 36, 38, 42, 44, 45, 55, 67, 70, 72 |
| MARS | *Multivariate Adaptive Regression Splines* | 35 |
| MCC | *Matthew's correlation coefficient* | 34, 45, 50, 54, 59, 61, 62, 69, 71 |
| MGI | Ministério Da Gestão E Da Inovação Em Serviços Públicos | 73 |
| MLP | *Multi Layer Perceptron* | 50, 54, 59, 60, 61, 62, 63, 64, 65, 66, 72 |
| MSE | *Mean Square Error* | 42 |
| OECD | *Organisation for Economic Cooperation and Development* | 17 |
| PNCP | Portal Nacional De Contratações Públicas | 73 |
| QDA | *Quadratic Discriminant Analysis* | 32, 42, 43, 54, 61, 62 |

| | | |
|---|---|---|
| RD | *Relative Difference* | 41, 45, 73, 75 |
| RDALT | *Relative Difference Alternative* | 41, 75 |
| RDNOR | *Relative Difference Normalized* | 40, 41, 75 |
| RFE | *Recursive Elimination Feature* | 27, 50, 59 |
| RNA | Rede Neural Artificial | 31, 44 |
| ROC | *Receiver Operating Characteristic* | 35, 54 |
| ROS | *Random Over Sampler* | 29, 53, 59, 60 |
| RUS | *Random Under Sampler* | 29, 53, 59, 60, 61 |
| SGD | *Stochastic Gradient Descent* | 32, 43, 45, 53, 54, 55, 62, 63, 64, 65, 66, 72 |
| SHAP | *SHapley Additive exPlanation* | 37, 38, 42, 44, 45, 55, 62, 65, 66, 67, 69, 70, 72, 73 |
| SIASG | Sistema Integrado De Administração De Serviços Gerais | 46 |
| SKEW | *Skewness* Ou Assimetria | 23, 40, 41, 44, 45, 52, 53, 60, 63, 64, 65, 66, 67, 69, 72, 73 |
| SMOTE | *Synthetic Minority Over-sampling Technique* | 29, 53, 55, 59, 60, 61, 62, 63, 68, 71 |
| SMOTEENN | SMOTE + ENN | 29, 53, 59, 60, 61, 62 |
| SPD | *Spread* Ou Dispersão | 40, 41, 45, 52, 73, 75 |
| SVM | *Support Vector Machine* | 29, 31, 40, 42, 43, 44, 45, 54, 62, 63, 64, 65, 66, 67, 72 |
| TCU | Tribunal De Contas Da União | 18 |
| TN | *True Negative* | 34 |
| TP | *True Positive* | 34 |
| VAR | Variância | 44 |
| XAI | *Explainable Artificial Intelligence* | 35, 38, 44 |
| XGBOOST | *EXtreme Gradient Boosting* | 42, 44, 45, 53, 54, 55, 62, 63, 64, 65, 66, 67, 69 |

# Capítulo I Introdução

Na definição do dicionário Michaelis[1], conluio é uma combinação ou acordo estabelecido entre duas pessoas para prejudicar outrem. O Guia CADE [2019] define cartel em licitação como "conluio entre agentes econômicos com o objetivo de eliminar ou restringir a concorrência dos processos de contratação de bens e serviços pela Administração Pública". Em todos os casos, essa é uma conduta a ser combatida, pois afeta a concorrência, condições de preço e qualidade na prestação dos serviços para a Administração Pública.

## I.1 Contextualização e Problema

A Constituição Federal do Brasil de 1988 dispõe no Capítulo 37 sobre a necessidade da Administração Pública obedecer aos princípios da legalidade, impessoalidade, moralidade, publicidade e eficiência [Brasil, 1988]. Considerando os princípios constitucionais, a Lei 14.133 de 2021 estabelece normas gerais de licitação e contratação para as Administrações Públicas [Brasil, 2021]. Esta lei também declara princípios a serem observados, além dos constitucionais, por exemplo, do interesse público, probidade administrativa, igualdade, transparência, eficácia, vinculação ao edital, julgamento objetivo, segurança jurídica, razoabilidade, competitividade, economicidade. O artigo 11 cita como objetivos do processo licitatório assegurar a seleção da proposta mais vantajosa, isonomia entre licitantes, justa competição, evitar preços inexequíveis, superfaturados ou sobrepreços.

Segundo Liu and Serfes [2007], as empresas têm incentivos muito fortes para coordenar suas estratégicas e aumentar sua lucratividade. Liu and Serfes [2007] afirmam existir três esquemas simples de conluio: segmentação de mercado, sendo cada segmento um monopólio; uniformização de preços de monopólio e uniformização de preços competitivos. Essas práticas ferem os princípios constitucionais e o objetivo da Lei 14.133/2021. O Conselho Administrativo de Defesa Econômico (CADE) tem como uma de suas funções o combate a cartéis. Suas responsabilidades, determinadas pela Lei 12.529 de 2011 [Brasil, 2011], são de julgar e punir administrativamente infrações à ordem econômica, analisar atos de concentração de modo a evitar excessiva concentração e promover a competitividade. Num contexto global, a *Organisation for Economic Cooperation and Development* (OECD) dispõe de guia com orientações para combater conluios em licitações públicas [OECD,

[1] https://michaelis.uol.com.br/moderno-portugues/busca/portugues-brasileiro/conluio

2009] e recomendação para mensuração dos impactos de ações anticompetitivas em [OECD, 2014].

O CADE divulga anualmente, desde 2018, um documento sobre os benefícios da atuação do CADE, baseado na recomendação da OECD [2014]. Os benefícios mensurados, relativos a atuação contra cartéis, para os últimos quatro anos são em torno de: 611 milhões em 2020; 3,2 bilhões em 2021; 5,3 bilhões em 2022; e 563 milhões em 2023. Sobre estimativas de impactos financeiros nas licitações, Commission and for Competition [2009] avaliou que, em 70% dos cartéis examinados, o preço é acrescido de 10% até 40%, sendo a média em 20%. Nas estimativas pelo Tribunal de Contas da União (TCU) de casos de cartéis brasileiros, estima-se que o valor mais provável de prejuízo em um ambiente cartelizado é de 17%, em relação as licitações, ou 14,49% sobre o valor do contrato [TCU, 2021].

Ainda sobre a atuação do CADE, o órgão publica em seu site o Anuário do CADE[2], com as informações institucionais do ano anterior. Destacam-se nas últimas quatro publicações as quantidades de julgamentos de cartéis pelo CADE ao ano: 14 em 2020; 22 em 2021; 11 em 2021; e 13 em 2023. Outro indicador interessante é sobre o êxito no judiciário pelo CADE, que inclui julgamento de casos de cartel, condutas unilaterais, tipicamente relacionadas a concentração de mercado por um único fornecedor, e conduta comercial uniforme, que é similar a cartéis, mas tipicamente relacionadas a adoção de tabelas de preços. Em 2020 o indicador foi de 54,91%, 2021 de 57,5%, 2022 de 63% e 2023 de 56%.

Outros números que demonstram o tamanho do mercado de licitações públicas dos órgãos do Governo Federal do Brasil é a quantidade de licitações no ano. Estes números podem ser obtidos no portal de transparência do Controladoria Geral da União (CGU)[3]. Em 2020 foram realizadas 131,7 mil, 2021 foram 19,1 mil, 2022 foram 19,8 mil e 2023 foram 20,3 mil licitações. Esse volume demonstra o desafio de garantir a conformidade de todas as licitações públicas frente as capacidades de atuação dos órgãos de controle. Para estes desafios, a utilização de técnicas de identificação automáticas de conluios podem ser úteis. Além disso, há oportunidades para aumentar o percentual de êxito dos processos de conluio no judiciário. Em um processo judicial, o entendimento e explicação das características que levaram o órgão a concluir sobre a existência de conluios, são relevantes para a fundamentação dos casos.

O dever de evitar, detectar e combater a prática de conluios e cartéis não é exclusiva dos órgãos de controle. Todos os envolvidos no processo licitatório (pregoeiros e membros das comissões de licitação, por exemplo) precisam saber identificar e prevenir tais fraudes. Com este objetivo o CADE dispõe de um guia para o combate de cartéis[4] e um *checklist* para identificação de sinais de alertas[5].

[2]https://www.gov.br/cade/pt-br/centrais-de-conteudo/publicacoes-institucionais/anuarios-do-cade

[3]https://portaldatransparencia.gov.br/licitacoes

[4]https://cdn.cade.gov.br/Portal/centrais-de-conteudo/publicacoes/guias-do-cade/guia-de-combate-a-carteis-em-licitacao-versao-final-1.pdf

[5]https://cdn.cade.gov.br/Portal/assuntos/noticias/2021/Checklist_Cartel-em-licitacoes_sinais-de-

O CADE cita os seguintes sinais de alerta:

- Conjunto de empresas alternando como vencedora no mesmo órgão.
- Propostas vencedoras com padrão de distribuição geográfica.
- Mesmo fornecedor vencendo repetidas licitações de um mesmo órgão ou região.
- Diminuição súbita de participantes da licitação.
- Alinhamento de preços.
- Diminuição dos preços quando participa um novo concorrente.
- Empresas que sempre participam, mas nunca com propostas competitivas.

Os sinais de alertas descritos acima, demonstram padrões que podem ser percebidos nos processos licitatórios para indicar a existência de conluios. Modelos de aprendizado de máquina são tipicamente utilizados para esta tarefa. No entanto, mesmo um modelo com boa acurácia não exclui a necessidade de auditorias para confirmação da fraude. As auditorias devem seguir processos bem definidos. Segundo os padrões do Instituto dos Auditores Internos (IIA), os auditores devem basear suas conclusões e resultados do trabalho em análises e avaliações apropriadas, documentar informações suficientes, confiáveis e úteis para apoiar os resultados e conclusões do trabalho [IIA, 2017]. Logo, a capacidade de explicar as razões que um Auditor concluiu sobre os indícios de uma fraude é muito importante para solicitar que sejam aplicadas medidas admistrativas e legais cabíveis. A falta do cuidado sobre estes aspectos pode invalidar os resultados da Auditoria e investigação em curso.

Quanto aos métodos de detecção de conluios, alguns artigos abordam as diferenças de comportamento dos licitantes em ambiente de conluio e os padrões dos lances nas licitações. Graham and Marshall [1987] estudaram o comportamento de conluio do tipo "anel", onde nem todos os participantes de um leilão participam do cartel. Alguns achados sobre o comportamento indicam que os participantes tendem a cooperar para que o vencedor que seja participante do conluio, ganhe com maior preço possível. Porter and Zona [1993] estudaram o comportamento em licitações de firmas de construção civil em Long Island nos anos 1980 e observaram que o comportamento do menor licitante, que não participa de cartel, não é estatisticamente distinguível das outras empresas não cartéis, já os licitantes de menor oferta, que participam de cartel, diferem estatisticamente dos lances mais altos. Por fim, Bajari and Ye [2003] argumentaram que o problema de classificação de um leilão como competitivo ou influenciado por conluio pode naturalmente ser formulado como um problema de decisão estatística.

alerta_pessoas.pdf

A identificação de conluios a partir de análises estatísticas também é citada na literatura, com o trabalho de Bolotova et al. [2008] que utilizou estatística descritiva e procedimento econométrico e o trabalho de Busu and Busu [2021] que utilizou testes estatísticos paramétricos. Hu et al. [2013] propuseram um novo modelo para detecção de conluio baseado na teoria dos hiper grafos, onde busca encontrar *clusters* fortemente conectados. O trabalho de Amaral [2020] utilizou grafos de mineração sobre redes para identificar conluios. Outra abordagem explorada para detecção de conluios é a análise dos relacionamentos entre os participantes por meio de grafos. Greige et al. [2022] propuseram um sistema que destaca os jogadores que mais provavelmente exibem comportamentos de conluio em jogo *multiplayer*. O método analisa relações pessoais entre os jogadores e utiliza ferramentas de teoria dos grafos.

Embasados na premissa que a existência de conluios tende a perturbar o equilíbrio de preços, que normalmente se enquadraria numa distribuição normal, alguns artigos, como Huber and Imhof [2019] e Dadfarnia et al. [2020], exploraram a utilização de variáveis estatísticas para decidir sobre a existência de conluios. Wallimann et al. [2023] exploraram essa característica e utilizaram as variáveis para decidir pela probabilidade de conluio no seu modelo, mas seu método se propõe a ser mais eficiente em cenários onde nem todos os licitantes participam do conluio. O presente trabalho se baseia na ideia destes artigos, quanto a utilização de variáveis estatísticas e algoritmos de aprendizado de máquina para classificação. No entanto, diferente dos artigos, os modelos foram treinados com alguns atributos não explorados na literatura (*Outlier* e Quantidade de participantes na licitações), para um conjunto mais amplo de algoritmos de aprendizado de máquina e utilizados modelos explicáveis para avaliar a contribuição das variáveis estatísticas.

## I.2 Objetivo

O objetivo deste trabalho é contribuir com estudos a respeito da aplicação de métodos explicáveis no problema de detecção de conluios, demonstrando sua importância para comprovar a contribuição das variáveis estatísticas no aprimoramento do desempenho geral dos modelos de aprendizado de máquina. Logo, os objetivos específicos são:

- Construir um conjunto de dados consistente com os rótulos de quais licitações foram influenciadas por conluios;
- Demonstrar que a inclusão de variáveis estatísticas melhora o desempenho dos modelos de aprendizado de máquina e determinar quais variáveis são mais relevantes para essa finalidade;
- Demonstrar que os modelos explicáveis são úteis na descrição das razões de rotulação de um conluio.

## I.3 Metodologia

A metodologia aplicada foi baseada nas etapas do *Knowledge Discovery in Databases* (KDD), onde a primeira etapa é a obtenção e carga dos dados. Os dados utilizados são licitações públicas de órgãos do Governo Federal do Brasil, provenientes do Portal de Transparência do Governo Federal do Brasil dos anos de 2013 à 2018. A rotulação de existência de conluios nas licitações foi feita a partir de levantamento de casos de condenações do CADE para este mesmo período. A segunda etapa, refere-se às atividades de pré-processamento (seleção de dados, limpeza, codificação para valores numéricos etc.). Foi incluída uma etapa no KDD de enriquecimento, onde as variáveis estatísticas selecionadas são calculadas. A seleção dessas variáveis foi feita conforme levantamento da literatura relacionada. A próxima etapa é de seleção do modelo de aprendizado de máquina, onde diversos modelos foram avaliados quanto às suas métricas de desempenho. As duas últimas etapas envolvem a aplicação dos modelos explicáveis com a avaliação dos resultados de predição do modelo para um conjunto de dados novo.

## I.4 Organização do Trabalho

Este trabalho está organizado em mais cinco capítulos, além desta Introdução. O capítulo II detalha o referencial teórico com os conceitos necessários para desenvolvimento deste trabalho. O capítulo III apresenta os trabalhos relacionados ao tema de detecção de conluio, suas diferentes abordagens, e trabalhos relacionados a aplicação de modelos explicáveis e aprendizado de máquina. O capítulo IV detalha a metodologia, descrevendo as atividades de análise exploratórias, pré-processamento, seleção de atributos e dos algoritmos e métodos utilizados para cada experimento. O capítulo V aborda os resultados e comparações de cada experimento. Por fim, o capítulo VI apresenta as considerações finais e evoluções sobre o estudo.

# Capítulo II Fundamentação teórica

Este capítulo detalha os conceitos importantes para compreensão e discussão deste trabalho. A Seção II.1 descreve as principais variáveis estatísticas utilizadas. A seção II.2 detalha as etapas do KDD, que é o processo utilizado como base da metodologia, além de introduzir conceitos dos principais algoritmos executados nos experimentos. A Seção II.3 explica as métricas de avaliação de desempenho dos algoritmos de aprendizado de máquina, em especial a métrica selecionada para este trabalho. Por fim, a Seção II.4 introduz os conceitos de modelos explicáveis e os métodos selecionados para os experimentos.

## II.1 Estatística descritiva

Morettin and Bussab [2017] definem que uma análise descritiva de dados basicamente são cálculos de algumas medidas de posição e dispersão. As medidas de posição são: moda, que é o valor mais frequente do conjunto de valores observados; mediana, que é o valor que ocupa a posição central da série, quando ordenados crescentemente; média aritmética, que é a soma dos valores divididos pelo número deles. As medidas de dispersão tentam descrever a variabilidade das observações em relação a média. Toledo and Ovalle [1982] dividem esse grupo entre medidas de dispersão absoluta e relativa. Conforme Toledo and Ovalle [1982], todas as medidas citadas juntas ajudam na compreensão completa da distribuição e frequências estudadas.

Entre as medidas de dispersão absoluta, as mais usadas são Variância ($s^2$) e Desvio padrão (s) apresentadas nas equações II.1 e II.2, respectivamente. Estas medidas indicam o quanto as observações de um conjunto se desviam da média. Uma medida alternativa é a distância interquartil, que é a diferença entre o terceiro e primeiro quartis (equação II.3). Um quartil de ordem p é indicado por q(p), onde p é uma proporção entre 0 e 1 de observações abaixo dele. Alguns quartis têm nomes particulares: q(0,25) é primeiro quartil; q(0,50) é a mediana; q(0,75) é terceiro quartil. Os quartis são úteis para verificar se uma distribuição de dados é simétrica. O Coeficiente de Variação (CV) é definido pela razão entre o desvio padrão e média amostral (equação II.4).

$$\text{Variância} = var(X) = s^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n} \tag{II.1}$$

$$\text{Desvio padrão} = s = \sqrt{var(X)} \tag{II.2}$$

onde:

$i$ = a observação

$n$ = total de observações

$x_i$ = valor da observação

$\bar{x}$ = média dos valores das observações.

$$d_q = q_3 - q_1 \tag{II.3}$$

onde:

$q_3$ = valor do terceiro quartil

$q_1$ = valor do primeiro quartil

$$CV = \frac{s}{\bar{x}}.100 \tag{II.4}$$

onde:

$s$ = desvio padrão

$\bar{x}$ = média dos valores das observações

As medidas de dispersão relativas permitem comparar duas ou mais distribuições. O CV (equação II.4) é o resultado do desvio padrão e média (Pearson) ou desvio médio e mediana (Thorndlike) [Morettin and Bussab, 2017]. CV ideal é sempre o menor possível, pois indica que o conjunto de dados é homogêneo, com baixa variabilidade dos dados em relação a mediana. Como o próprio nome sinaliza, *Skewness* ou Assimetria (SKEW) indica o grau de deformação de uma curva de frequência de distribuição (equação II.5). O coeficiente de *Kurtosis* ou Curtose (KURT) indica até que ponto a curva de frequência de distribuição é mais achatada que a curva normal (equação II.6). As duas medidas demonstram numericamente o nível de deformação da curva de frequência de distribuição em relação a curva normal, seja quanto a posição da cauda da curva (SKEW) ou concentração das observações (KURT). Hair et al. [2010] argumentam que os valores de SKEW são considerados aceitáveis para comprovar distribuição normal quando estão entre -2 e +2. Para curtose, quando estão entre -7 a +7.

$$SKEW \text{ ou skewness} = \frac{3(\bar{x} - M_d)}{s} \tag{II.5}$$

$$KURT \text{ ou kurtosis} = \frac{d_q}{q_{90} - q_{10}} \tag{II.6}$$

onde:

$s$ = desvio padrão

$\bar{x}$ = média dos valores das observações

$M_d$ = moda

$d_q$ = distância interquartil

$q_{90}$ = percentil 90 (frequência acumulada relativa a 90% dos dados)

$q_{10}$ = percentil 10 (frequência acumulada relativa a 10% dos dados)

## II.2 Descoberta de conhecimento em base de dados

A Descoberta de Conhecimento em Bases de Dados, ou *Knowledge Discovery in Databases* (KDD), historicamente originou de diversas áreas, tais como Estatística, Inteligência Computacional e Aprendizado de Máquina, Reconhecimento de padrões e Banco de Dados. KDD é um processo composto por etapas operacionais, ilustradas na Figura II.1. A etapa de pré-processamento executa as atividades para captação, organização e tratamento dos dados. O objetivo desta etapa é preparar os dados para os algoritmos da etapa seguinte. A etapa de Mineração de Dados realizada a busca efetiva por conhecimentos. A etapa de pós-processamento executa o tratamento do conhecimento obtido na Mineração de Dados e seu objetivo é viabilizar a avaliação do conhecimento descoberto, nem sempre sendo necessária [Goldschmidt and Passos, 2005].

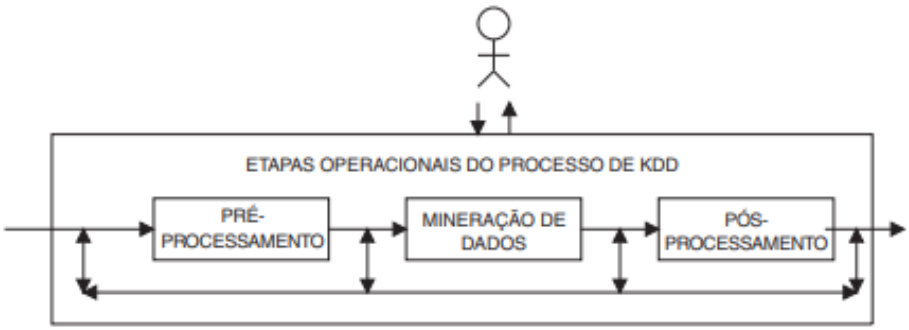



Figura II.1: Etapas operacionais do processo de KDD [Goldschmidt and Passos, 2005]

### II.2.1 Pré-processamento

O pré-processamento compreende diversas funções para carga, correção de dados errados e formatação para os algoritmos de Mineração de Dados. Nesta etapa é muito importante investir um tempo para conhecer os dados que foram coletados. Para cada atributo do modelo é preciso entender seu domínio, tipo do dado e seu significado para o conjunto de dados. Obter informações, tais como dicionário de dados, ou qualquer documentação sobre a fonte de dados que está sendo utilizada, são bons métodos para obter conhecimento sobre os dados. No entanto, a análise exploratória é essencial para percepção da qualidade e relevância de cada atributo. Sumarizar os dados

em formatos gráficos ou pela manipulação do conjunto de dados são importantes ferramentas nesta busca de conhecimento sobre os dados.

A etapa de pré-processamento pode ser dividida em subetapas, listadas como subseções abaixo. A subetapa Seleção de dados remove linhas ou colunas desnecessárias, ou que agreguem pouco para o modelo de aprendizagem. A coleta carrega os dados das fontes selecionadas. A Limpeza resolve problemas sobre as qualidades dos dados. A Extração de atributos cria atributos derivados de outros ou de combinações. Classificação de conluio cria o atributo classificador a partir dos dados unidos dos dois conjuntos de dados (licitação e conluios confirmados). A Codificação transforma os atributos em formato numérico (os modelos de classificação normalmente só aceitam dados numéricos). A Normalização reduz a amplitude dos dados, para evitar vieses de um único atributo. O Particionamento divide o conjunto de dados entre treinamento e teste. Por fim, a Correção de prevalência reduz a diferença entre classificações Sim e Não.

### Coleta de dados

A coleta dos dados de forma correta é uma etapa essencial para descoberta de conhecimento, pois a escolha errada dos dados compromete todo o processo de aprendizagem. Há uma extensa literatura que estuda sobre a importância e os atributos de qualidade requeridos para os dados. Conforme avaliam Haug et al. [2011], há um consenso geral na literatura sobre a importância de ter dados corretos e adequados. A tabela II.1 sumariza as dimensões mais frequentes de qualidade dados Wang et al. [1995]:

Tabela II.1: Dimensões mais citadas da qualidade de dados, por [Haug et al., 2011] de [Wang et al., 1995].

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Acurácia | 25 | Flexibilidade | 5 | Suficiência | 3 | Informatividade | 2 |
| Confiabilidade | 22 | Precisão | 5 | Usabilidade | 3 | Nível de detalhe | 2 |
| Oportunidade | 19 | Formato | 4 | Utilidade | 3 | Quantitatividade | 2 |
| Relevância | 16 | Interpretabilidade | 4 | Clareza | 2 | Escopo | 2 |
| Completude | 15 | Contente | 3 | Comparabilidade | 2 | Compreensibilidade | 2 |
| Atual | 9 | Eficiência | 3 | Concisão | 2 | | |
| Consistência | 8 | Importância | 3 | Livre de viés | 2 | | |

Apesar dos dados serem tratados nas etapas seguintes do pré-processamento, elas não resolvem todos os problemas da má qualidade dos dados, por exemplo, se os dados são verdadeiros e suficientes. Utilizar dados de fontes não confiáveis pode gerar vieses no modelo de aprendizagem ou não resolverem o problema alvo, pois os dados podem ter sido alterados para esconder ou indicar resultados manipulados pela fonte emissora. Quanto a suficiência dos dados, é necessário verificar se todos os dados necessários para execução desta metodologia estão disponíveis. Informações sobre Preço praticado e os participantes de cada licitação são essenciais, uma vez que os atributos

estatísticos serão derivados deles. O volume de dados também é muito importante para o modelo de aprendizagem. Baixo volume de dados pode não gerar modelos suficientemente generalistas.

Por fim, essa metodologia trabalha com dados classificados de casos de conluios em licitações. Logo, pode ser necessário coletar dados de mais de uma fonte e uni-las. A união precisa garantir que os dados que estão sendo relacionados realmente possuem tal relação. Caso contrário, o modelo pode aprender uma classificação errada, ou não convergir para nenhuma classificação, quando os dados são relacionados de forma aleatória e dispersa, não evidenciando para modelo nenhum padrão para aprendizagem.

Logo, a preocupação com a qualidade de dados deve ser observada em todas as etapas do KDD, mas principalmente nas iniciais, tais como a coleta e pré-processamento. Desta forma é importante além de obter dados de fontes confiáveis, verificar se todos os dados de interesse foram carregados para o conjunto de dados (confrontar a totalidade da fonte) e se a qualidade está sendo mantida após cada etapa do processamento. Utilizar métodos simples como calcular e imprimir quantidades de registros, total de valores únicos e nulos de cada atributo e inspecionar os domínios de dados resultantes após cada processamento, ajuda a validar se os processamentos estão produzindo os efeitos esperados.

**Seleção de dados**

Seleção de dados, que pode ser dividida em: redução de dados horizontais, caracterizada pela amostragem, eliminação direta de casos, segmentação ou agregação das informações; e redução de dados verticais, ou redução de dimensionalidade, que busca encontrar um conjunto mínimo de atributos que preserve a informação original do conjunto de dados.

Numa estratégia de seleção de dados horizontais, é preciso entender os dados para ver o que faz sentido para o modelo que será treinado. Linhas com muitos valores nulos ou com valores que não agregam para o objetivo alvo do aprendizado, podem ser eliminadas.

Para estratégia de seleção de dados verticais, a quantidade de valores únicos, faltantes e baixa variabilidade, são análises importantes de serem feitas. Atributos com muitos valores únicos tendem a treinar modelos de aprendizado com muito *overfitting*[1]. Atributos com muitos valores faltantes também afetam diretamente o modelo de aprendizagem, pois algumas operações podem não ser realizadas. Por fim, atributos com baixa variabilidade tendem a ser pouco valiosos para os modelos de aprendizagem. Estes atributos não são bons no reconhecimento de padrões, pois, diferenciam-se pouco os dados entre si.

Além disso, o objetivo da redução da dimensionalidade é determinar um conjunto mínimo de atributos que representam significativamente o conjunto de dados para o objetivo do modelo de

[1] Overfitting é um comportamento indesejado no aprendizado de máquina, uma vez que os dados são excessivamente ajustado ao modelo, tendo pouca generalização e baixa acurácia quando aplicado a novos dados

aprendizado. Essa redução também ajuda na diminuição do custo computacional (CPU e memória) durante as execuções dos algoritmos. A Figura II.2 apresenta uma visão geral dos tipos de técnicas para seleção de atributos, segundo Kuhn and Johnson [2013]. A diferença entre as técnicas de seleção supervisionada e não supervisiona é a utilização do atributo alvo. Um exemplo de seleção não supervisionada é uma matriz de correlação de dados. A seleção intrínseca é quando o próprio modelo já prevê a seleção dos atributos, como é o caso dos modelos baseados em árvores e *Lasso*. Os métodos *wrapper* criam muitos modelos com diferentes subconjuntos de atributos e selecionam os recursos que resultaram no modelo de melhor desempenho, um exemplo é o *Recursive Elimination Feature* (RFE). Já os métodos de filtro, usam técnicas estatísticas para avaliar a relação de cada atributo com o atributo alvo, um exemplo de aplicação é o *Feature importance*.

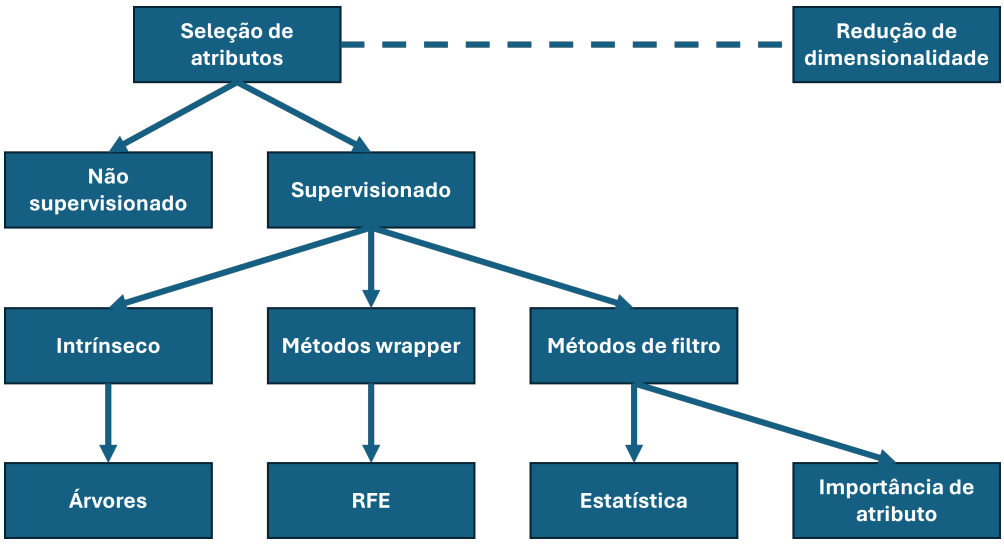



Figura II.2: Visão geral das técnicas de seleção de atributos.

### Limpeza

A subetapa de Limpeza exige muito conhecimento sobre o conjunto de dados. As seguintes abordagens podem ser utilizadas para a construção de um conjunto de dados consistente:

1. Limpeza de informações ausentes. Pode-se adotar diversas estratégias, como por exemplo, a eliminação dos casos com valores de colunas ausentes ou preenchimento dos valores, seja manualmente ou inserindo valores constantes, ou medidas estatísticas;

2. Limpeza de inconsistências, que compreende identificar valores inconsistentes, considerando um único atributo ou conjunto de atributos, por exemplo, idades de pessoas maiores de 1.000 anos, ou cálculo da idade não corresponde à data de nascimento. Entre as estratégias, pode-se optar pela limpeza ou correção dos erros;

3. Limpeza de valores não pertencentes ao domínio.

**Extração de atributos**

Esta subetapa é uma oportunidade de melhorar os dados e atributos existentes no conjunto de dados. Algumas vezes um atributo pode ser relevante, mas quando combinado com outro, produz uma informação equivalente ou melhor. Ou então, ele pode ser dividido ou transformado para produzir atributos mais genéricos e com menos variabilidade, o que tende a ser melhor para o modelo de aprendizagem.

**Enriquecimento**

Enriquecimento é a estratégia que objetiva acrescentar novas informações aos registros existentes, que podem ser obtidas por pesquisas e consultas a base de dados externas, por exemplo.

**Codificação**

Codificação é a operação que formata os dados para serem processados pelos algoritmos de mineração de dados. Os algoritmos possuem necessidades específicas de formato dos dados, por exemplo, as redes neurais exigem dados numéricos. Há diversas estratégias de codificação, como: *Label Encoder*, que cria uma única coluna que mapeia um número para cada categoria; *One-Hot Encoder*, que cria diversas colunas para cada categoria e atribui 1, onde a categoria existe e 0, caso contrário; *Ordinal Encoder*, que atribui um valor numérico para cada categoria mapeada, mas diferente do *Label Encoder*, neste método a ordem importa (por exemplo, atributo severidade com os valores alta, média e baixa).

Outras três abordagens para mapeamento categórico são: *Frequency Encoder*, que substitui as categorias pelas suas frequências; *Target Encoder* que consiste na substituição das classes dos atributos pela média do seu alvo, conforme equação II.7; por fim o *Leave-One-Out Encoder*, que calcula a média do alvo excluindo a observação atual, vide Tabela II.2. Esta última tenta reduzir o fenômeno de fuga de informação, que consiste na utilização inadvertida de dados de teste no treinamento, o que gera uma superestimação do desempenho do modelo durante o treinamento. Esta característica foi considerada para selecionar este método para o *pipeline* da metodologia.

$$E_i = \frac{1}{n_i} \sum_{k=1}^{n_i} Y_k \tag{II.7}$$

onde:

$k$ = a observação

$i$ = a classe

$n_i$ = total de observações da classe

$Y_k$ = valor alvo de cada observação

Tabela II.2: Exemplo de codificação para o *Leave-One-Out Encoder*.

| Divisão | Categoria | Classificador alvo (Y) | Média (Y) |
|---|---|---|---|
| Treino | A1 | 0 | 0,667 |
| Treino | A1 | 0 | 0,667 |
| Treino | A1 | 1 | 0,333 |
| Treino | A1 | 1 | 0,333 |
| Teste | A1 | - | 0,5 |
| Teste | A1 | - | 0,5 |

**Normalização**

Alguns algoritmos de aprendizagem, como SVM e K-NN, utilizam medidas de distância para separar os dados. Logo, se uma variável possui uma escala de valores muito maior que outras, ela tende a dominar o cálculo de distância. A normalização equaciona essa questão, deixando todas as variáveis com a mesma amplitude ou valores próximos. Como exemplos de métodos de escala dos dados, cita-se *Standard Scaler*, que considera a média e o desvio para fazer o escalonamento usando a fórmula II.8 e *Min Max Scaler*, que fixa o maior valor como 1 e o menor como 0 e normaliza os demais valores conforme a amplitude dessa diferença.

$$z = \frac{(x - \bar{x})}{s} \tag{II.8}$$

onde:

$x$ = observação a ser normalizada

$\bar{x}$ = média

$s$ = desvio padrão

**Correção de prevalência**

Esta subetapa deve ser executada quando for notado um desbalanceamento entre as classes. A prevalência de classificações para uma classe tende criar um viés de classificação. Há diversos métodos de correção de prevalência, como *Random Under Sampler* (RUS)[2], que remove registros da classe majoritária, *Random Over Sampler* (ROS)[3], que duplica registros da classe minoritária, Smote[4], que gera novos registros da classe minoritária a partir dos existentes e Smoteenn[5], que é a junção do Smote com *Edited Nearest Neighbours* (ENN) para redução de ruídos de classes. ENN é um método que remove, do conjunto de dados, observações próximas ao limite de decisão. Ou seja,

[2]https://imbalanced-learn.org/stable/references/generated/imblearn.under_sampling.RandomUnderSampler.html
[3]https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.RandomOverSampler.html
[4]https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html
[5]https://imbalanced-learn.org/stable/references/generated/imblearn.combine.SMOTEENN.html

são removidas observações quando a maioria dos seus vizinhos mais próximos são de uma classe diferente.

### Particionamento

O particionamento consiste na divisão aleatória das amostras do conjunto de dados entre um conjunto de treinamento e teste/validação. Uma estratégia de particionamento, nomeada de *Holdout*, consiste na divisão aleatória percentual simples do conjunto dos dados, tal como 60%/40% (treinamento/teste), 70%/30%, 80%/20% e 90%/10%. Existem outras abordagens possíveis, como o *K-Fold-cross-validation*, onde o conjunto de dados é dividido em k conjuntos, um pedaço da amostra desse conjunto utilizada para treinamento em um grupo servirá como teste num outro conjunto, vide Figura II.3, dessa forma, todas as amostras são utilizadas como treinamento e testes. No entanto, pela simplicidade e agilidade, a opção foi pelo *Holdout* 80%/20%.

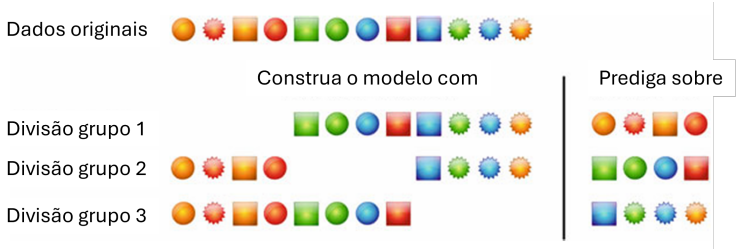



Figura II.3: Exemplo de esquema de validação cruzada tripla, retirado de Kuhn and Johnson [2013]

## II.2.2 Mineração de dados

A mineração de dados consiste na técnica para explorar grandes volumes de dados, em busca de padrões, enquanto o aprendizado de máquina automatiza a construção de modelos analíticos. Essa etapa compreende a execução dos algoritmos sobre os dados. A escolha do algoritmo depende das expectativas do negócio. Por exemplo, suponha que há necessidade do conhecimento sobre o modelo seja transparente, neste caso, deve-se eliminar algoritmos que produzam modelos que não são interpretáveis. Outro fator que influencia na escolha do algoritmo é o tipo das variáveis disponíveis que podem não ser suportadas pelo algoritmo. Um importante conceito nesta etapa é o de aprendizado de máquina, pois muitos algoritmos possuem a capacidade de aprender com exemplos.

Aprendizado de máquina é uma subárea da Inteligência Artificial (IA). A IA é o estudo de agentes que executam ações conforme percepções que recebem do ambiente [Russell and Norvig, 2013]. Um projeto que inclui agentes com aprendizagem, denota-se aprendizado de máquina. Ainda segundo Russell and Norvig [2013], existem três principais tipos de aprendizagem, que se diferenciam pela forma que abordam o *feedback*. Na aprendizagem não supervisionada, o agente aprende padrões

sem que seja recebido *feedbacks* explícitos, a tarefa mais comum neste caso é o agrupamento. Na aprendizagem por reforço, o agente aprende a partir de recompensas ou punições, enquanto na aprendizagem supervisionada, o agente observa entradas e saídas e aprende uma função que mapeia entrada para a saída.

Entre os algoritmos de aprendizado de máquina existentes, a Árvore de Decisão é um dos mais simples. A decisão desse algoritmo é tomada ao executar uma sequência de testes. Baseado no algoritmo de árvore, há um conjunto de algoritmos mais complexos, *random forest*, por exemplo, segundo Kuhn and Johnson [2013], treina diversas árvores a partir de amostras dos dados originais. A predição é dada a partir da média dos diversos preditores gerados.

Os modelos de regressão tentam descrever uma função que melhor se encaixa nos dados de entradas e saídas fornecidos. Conforme descrito por Kuhn and Johnson [2013], as regressões lineares podem ser definidas pela equação II.9 e sempre anunciam previsões confiantes de 1 ou 0, mesmo para casos próximos da fronteira, já as logísticas fazem previsões graduais. Os modelos lineares comuns buscam encontrar parâmetros estimados que minimizam o coeficiente estimado, já regressão *ridge* e *lasso* buscam estimar menor variância.

$$y_i = b_0 + b_1 x_{i1} + b_2 x_{i2} + ... + + b_P x_{iP} + e_i \tag{II.9}$$

onde:

$y_i$ = reposta numérica para o i-ésima amostra

$b_0$ = intercepto estimado

$b_j$ = coeficiente estimado do j-ésimo preditor

$x_{ij}$ = valor do j-ésimo preditor na i-ésima amostra

$e_i$ = erro randômico que não pode ser explicado pelo modelo

*Naive Bayes* é outro modelo comum de aprendizado, que supõe que os atributos são independentes, dada uma classe. Uma vez que o modelo é treinado, obtém-se uma previsão determinística de qual é a classe mais provável [Russell and Norvig, 2013]. A máquina de vetor de suporte ou SVM possui boa flexibilidade para representar funções complexas, pois constrói separador de margem máxima (limite de decisão) e cria separação linear em hiperplano.

Inspirado na ideia das redes de células cerebrais, a Rede Neural Artificial (RNA) implementa um classificador linear, onde diversas unidades são conectadas e elas "disparam" quando uma combinação linear de suas entradas excede algum limiar [Russell and Norvig, 2013]. Uma RNA constrói um modelo matemático inspirado em um sistema neural biológico simplificado, com capacidade de aprendizado, generalização, associação e abstração.

O K-NN é um algoritmo que não precisa de treinamento prévio, e consiste em cálculos de distância a cada novo registro e determinação a qual grupo ele é mais similar. Já o algoritmo

Apriori busca encontrar conjunto de itens que satisfaçam uma condição de suporte mínimo. A partir desse conjunto, o algoritmo gera regras de associação [Goldschmidt and Passos, 2005].

*AdaBoosting* (*adaptive boosting algorithm*) sequencialmente aplica algoritmos de classificação fracos, onde o erro e acurácia são apenas pouco melhores que palpites aleatórios, para modificar versões dos dados. Ao final, a predição de todos são combinadas por uma ponderação de votos majoritários para produzir a predição final, conforme mostra a Figura II.4 [Hastie et al., 2013].

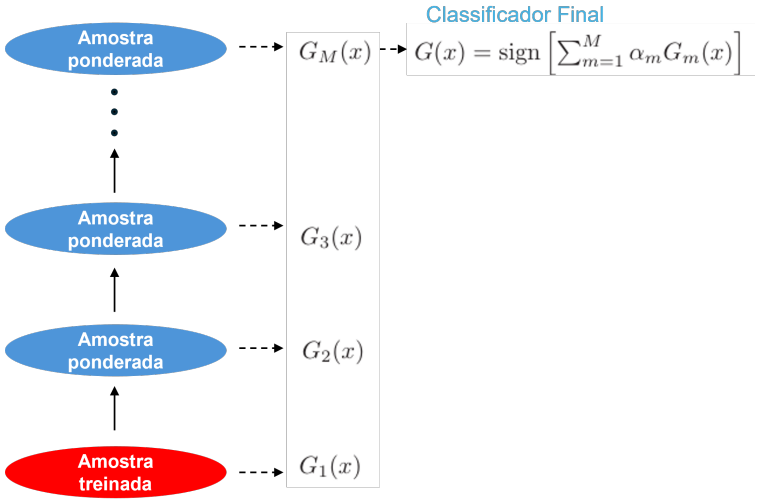



Figura II.4: Modelo esquemático do *AdaBoost* [Hastie et al., 2013].

Segundo Kuhn and Johnson [2013], o *Gradient Boosting* é um algoritmo simples e altamente adaptável para diversos modelos. O princípio básico consiste em definir uma função de perda (erro quadrático, por exemplo), logo, o algoritmo procura um modelo aditivo que minimize essa função. Já o *Stochastic Gradient Descent* (SGD) inclui uma etapa inicial de selecionar aleatoriamente uma fração dos dados de treinamento.

*Linear Discriminant Analysis* (LDA) e *Quadratic Discriminant Analysis* (QDA) podem ser derivadas de modelos probabilísticos simples que modelam a distribuição condicional de classe dos dados para cada classe. As previsões são obtidas usando a regra de Bayes, conforme equação II.10. Os dois modelos usam densidades Gaussiana, conforme a equação II.11 [Hastie et al., 2013]. O processo de cálculo do LDA consiste no cálculo da variância entre classes (distância entre as médias das classes) e variância dentro da classe (distância entre as médias das classes e amostras). Já o QDA é uma extensão do LDA. LDA assume que as variáveis têm distribuições normais e covariâncias entre si iguais para todos os grupos. QDA relaxa essas suposições, permitindo que as covariâncias sejam diferentes em cada grupo e que a distribuições das variáveis sejam não lineares.

$$\delta_k(x) = x^T \Sigma^{-1} \bar{x}_k - \frac{1}{2} \Sigma^{-1} \bar{x}_k^T + \log \pi_k \tag{II.10}$$

$$\delta_k(x) = -\frac{1}{2} \log |\Sigma_k| - \frac{1}{2} (x - \bar{x}_k)^T \Sigma_k^{-1} (x - \bar{x}_k) + \log \pi_k \tag{II.11}$$

onde:

$\delta_k(x)$ = equação em x

$k$ = classe

$\Sigma_k$ = matriz de covariância comum entre as classes k e l

$\bar{x}_k$ = probabilidade média da classe k

$\pi_k$ = probabilidade anterior da classe k, sendo $\sum_{k=1}^{K} \pi_k = 1$

### II.2.3 Pós-processamento

Esta é a etapa para avaliar os resultados obtidos pelo modelo. Entre as operações desta etapa, cita-se: (i) simplificação do modelo de conhecimento, na qual o objetivo é representar o conhecimento por meio de regras; (ii) transformação de modelo de conhecimento, que é um método que tenta converter um formato de representação de modelo em outro, por exemplo, transformar uma árvore de decisão em conjunto de regras; (iii) organização e apresentação dos resultados, por exemplo, em formato gráfico, grafo, árvore ou regras.

## II.3 Métricas de avaliação

A escolha de métricas adequadas para cada modelo é essencial para o sucesso do projeto de análise de dados. Os valores das métricas demonstram a qualidade do modelo escolhido e muitas delas se complementam e devem ser analisadas em conjunto. A precisão, por exemplo, enfatiza o nível de acerto das classificações positivas e o *recall* enfatiza os erros das classificações. Logo, a precisão alta pode indicar um bom nível de acerto, mas o *recall* sendo baixo pode indicar um conjunto de dados desbalanceado, logo um modelo ruim.

### II.3.1 Métricas de classificação

Num modelo de classificação, o objetivo é decidir em qual das classes determinada observação pertence. Para uma classificação binária, pode-se descrever que uma observação é positiva (P) ou negativa (N), sendo a positiva a afirmação verdadeira e negativa a falsa. Esta lógica pode ser expandida para mais de uma classificação. Dentro da lógica da classificação binária, é possível construir uma matriz de confusão, conforme apresentado na Tabela II.3, para indicar quantas observações existem em cada grupo.

A partir dessa matriz são derivadas diversas métricas, como a acurácia, que mede quantas observações foram classificadas corretamente, independente da classificação, conforme equação II.12:

$$\text{Acurácia} = \frac{\text{previsões corretas}}{\text{total de previsões}} = \frac{TP + TN}{TP + TN + FP + FN} \tag{II.12}$$

Tabela II.3: Matriz de confusão.

| | | **Previsto** | |
|---|---|---|---|
| | | **Positivo** | **Negativo** |
| **Real** | **Positivo** | positivo verdadeiro ou *True Positive* (TP) | falso negativo ou *False Negative* (FN) |
| | **Negativo** | falso positivo ou *False Positive* (FP) | negativo verdadeiro ou *True Negative* (TN) |

A precisão mede o quanto as previsões positivas foram previstas corretamente e é avaliada pela equação II.13:

$$\text{Precisão} = \frac{\text{previsões positivas corretas}}{\text{total de previsões positivas}} = \frac{TP}{TP + FP} \tag{II.13}$$

Ao contrário da precisão, a revocação (*recall*), ou taxa de verdadeiro positivo, dá maior ênfase para os erros por falso negativo e é calculada pela equação II.14:

$$\text{Recall} = \frac{\text{previsões positivas corretas}}{\text{total de observações positivas}} = \frac{TP}{TP + FN} \tag{II.14}$$

O *f1-score* é a média harmônica entre precisão e *recall* (equação II.15). Por ser uma média harmônica, é uma métrica muito interessante de ser usada quando se deseja que todas as quantidades da matriz confusão sejam consideradas:

$$\text{F1-score} = 2 * \frac{\text{precisão} * \text{recall}}{\text{precisão} + \text{recall}} \tag{II.15}$$

MCC é um indicador interessante por tentar resumir a matriz de confusão em um único valor, conforme equação II.16. Este retorna valores entre -1 (previsão inversa), 0 (previsão aleatória) e +1 (previsão perfeita). Na avaliação das métricas, considera-se desempenhos do MCC acima de 0,5 como fortes, acima de 0,3, moderados e abaixo como fracos. Chicco and Jurman [2020] avalia que essa métrica é a mais adequada na avaliação de tarefas de classificação binária. Uma vez que, somente produz pontuação alta se a previsão obtiver bons resultados em todas as categorias de matriz de confusão (TP, FN, FP e TN), proporcionalmente ao tamanho dos elementos positivos e ao tamanho dos elementos negativos no conjunto de dados.

$$\text{MCC} = \frac{TN * TP - FN * FP}{\sqrt{(TP + FP) * (TP + FN) * (TN + FP) * (TN + FN)}} \tag{II.16}$$

## II.4 Modelos explicáveis

Há uma distinção conceitual entre a explicabilidade e interpretabilidade de modelos de IA, apesar de que na prática estes conceitos são intercambiáveis. Miller [2019] chama de interpretabilidade a possibilidade de um ser humano entender, num determinado contexto, os critérios levados em consideração pela computação de uma decisão do modelo. Já a explicabilidade é a capacidade

de justificar o comportamento do modelo e as previsões dos classificadores. Em termos gerais, interpretabilidade está ligada à capacidade de explicar a um ser humano os resultados do modelo e explicabilidade está associada à compreensão da lógica interna do algoritmo (etapas e razões para ele chegar ao resultado).

Alguns algoritmos de aprendizado de máquina, como os modelos lineares e árvores de decisão, são possíveis de serem interpretados e explicados pelo próprio modelo gerado. Mas enquanto o volume de dados aumenta e a utilização de modelos mais complexos, como *ensemble methods* e *deep networks* produzem melhores resultados, a explicabilidade se torna extremamente complexa [Lundberg and Lee, 2017]. Segundo Lundberg and Lee [2017], a melhor explicação para um modelo é o próprio modelo. A IA explicável, ou *Explainable Artificial Intelligence* (XAI), refere-se a métodos/processos que tentam fornecer explicações para as decisões dos algoritmos de IA e modelos de aprendizado de máquina. No entanto, vale ressaltar que os resultados desses métodos não representam inteiramente o modelo original a ser explicado.

Neste trabalho, foram priorizadas avaliações e estudos de técnicas que sejam independentes do modelo de aprendizado de máquina utilizado. Mesmo que essas técnicas possuam implementações específicas otimizadas para diferentes características de algoritmos.

### II.4.1 Importância de atributos

Segundo Kuhn and Johnson [2013], a importância de atributos (ou *variable importance* ou *feature importance* em inglês) quantifica globalmente a relação entre o preditor e o resultado final. Ou seja, as técnicas de importância de atributos atribuem uma pontuação para os atributos, com base no quão importantes eles foram na predição do atributo classificador alvo. A maioria dos algoritmos não permite informar ao modelo sobre o quanto aumentar o preditor resulta numa diminuição do resultado. Essa caracterização só é possível em modelos paramétricos precisos, tais como modelos lineares (regressões logísticas, lineares e *Multivariate Adaptive Regression Splines* (MARS), por exemplo) e baseados em árvores (árvores de decisão, *random forest* e *Extra Trees*, por exemplo).

Entre as formas de quantificação de importância dos atributos, Kuhn and Johnson [2013] cita que, para variáveis numéricas tipicamente são utilizadas medidas de correlação estatística, como o coeficiente de correlação de Spearman; enquanto para variáveis categóricas há diversas abordagens, tais como utilizar a *Receiver Operating Characteristic* (ROC) *Area Under the Curve* (AUC) para quantificar a relevância do preditor ou testar se os valores médios de cada classe são diferentes. Outra abordagem para cálculo de importância dos atributos é a permutação de atributos, que consiste em medir a importância de cada um, embaralhando seus valores e observando a queda de desempenho.

### II.4.2 LIME

*Local Interpretable Model-agnostic Explanations* (LIME) foi proposto por Ribeiro et al. [2016] e seu objetivo é identificar um modelo interpretável sobre a representação interpretável que seja localmente fiel ao classificador. Formalmente, define-se $g \in G$, sendo g o modelo explicável e G o conjunto de potenciais modelos interpretáveis. Um modelo interpretável pode ser um modelo linear, árvore de decisão ou uma lista de regras, por exemplo. No entanto, g pode não ser simples o suficiente para interpretação humana, por isso, considera um $\omega(g)$ como uma medida de complexidade (nas árvores de decisão pode ser a profundidade da árvore, por exemplo).

A explicação produzida pelo LIME é dada pela equação II.17. A equação considera duas funções que precisam ser minimizadas. Além da minimização da medida de complexidade $\Omega(g)$, o modelo precisa minimizar a função perda $\mathcal{L}$, que mede o quão perto a explicação está do modelo g proposto, para a função original f, na localidade de $\pi_x$. A Figura II.5 ilustra a intuição do LIME. A cruz vermelha em negrito é a amostra que está sendo analisada. As instâncias da figura são amostras geradas pelo LIME, a partir das previsões do modelo f. Os fundos azuis e rosa representam o modelo caixa preta (função f). Note que não é possível representar f pelo modelo linear g (linha tracejada), mas para a instância que está sendo explicada, o modelo g gerado faz uma representação aproximada.

$$\varepsilon(x) = \arg\min_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g) \tag{II.17}$$

onde:

$\mathcal{L}$ = função para medir proximidade entre duas funções
$f$ = função do modelo construído
$g$ = função do modelo proposto
$\pi_x$ = medida de proximidade de uma amostra gerada e x
$\Omega(g)$ = medida de complexidade do modelo g

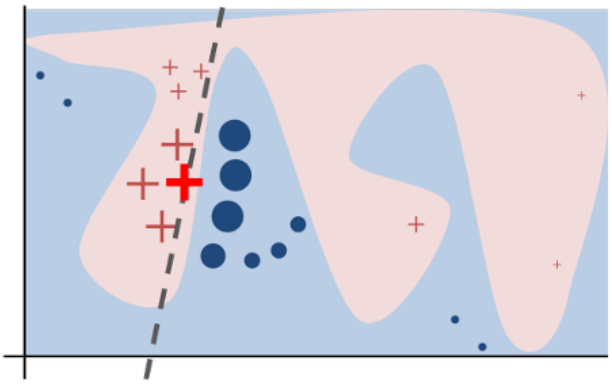


Figura II.5: Exemplo da intuição do LIME, apresentada por Ribeiro et al. [2016].

### II.4.3 SHAP

*SHapley Additive exPlanation* (SHAP) foi proposto por Lundberg and Lee [2017] e sua ideia é atribuir importâncias a cada atributo para cada previsão em particular. Os valores de Shapley foram introduzidos por Shapley [1953] no contexto da teoria dos jogos, onde para um jogo cooperativo, os valores de Shapley são distribuídos para os membros da equipe de forma justa. Mas dentro desta teoria, as equipes são montadas adicionando um jogador por vez e calculando as contribuições. A intuição de Shapley [1953] é de refazer o processo de formação da equipe diversas vezes, em ordens diferentes e fazer a média dos valores atribuídos para cada jogador, formalmente esse processo foi descrito pela equação II.18. A intuição Lundberg and Lee [2017] é de gerar o valor de Shapley para cada atributo.

$$\phi_i = \frac{1}{n} \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f_{S \bigcup \{i\}}(x_{S \bigcup \{i\}}) - f_S(x_S)] \tag{II.18}$$

onde:

| | |
|---|---|
| $\phi_i$ | = função que calcula a contribuição do jogador i |
| $f$ | = função que calcula a contribuição da equipe |
| $S$ | = conjunto de todos os jogadores |
| $F$ | = subconjunto de S |
| $f_{S \bigcup \{i\}}(x_{S \bigcup \{i\}})$ | = treinamento do modelo com o jogados i |
| $f_S(x_S)$ | = treinamento do modelo sem o jogador i |

# Capítulo III Trabalhos relacionados

As pesquisas por trabalhos relacionados foram conduzidas na base Scopus. Um esquema gráfico sobre os resultados de cada pesquisa está representado na Figura III.1. Uma pesquisa inicial por "explain* model" AND ("bidding" OR "collusion" OR "bid rigging") não retornou resultados, dando a percepção que não há aplicações de trabalhos sobre identificação de conluios e modelos explicáveis. Uma pesquisa mais abrangente pela *string*: "explain* model" AND ("bidding" OR "collusion" OR "bid rigging" OR "random forest" OR "neural network" OR "audit"), retornou 185 documentos. Destaca-se um conjunto grande da literatura aplicada na área médica, utilizando modelos de aprendizado de máquina e métodos de XAI como o LIME e o SHAP. No entanto, não há trabalhos sobre identificação de conluios.

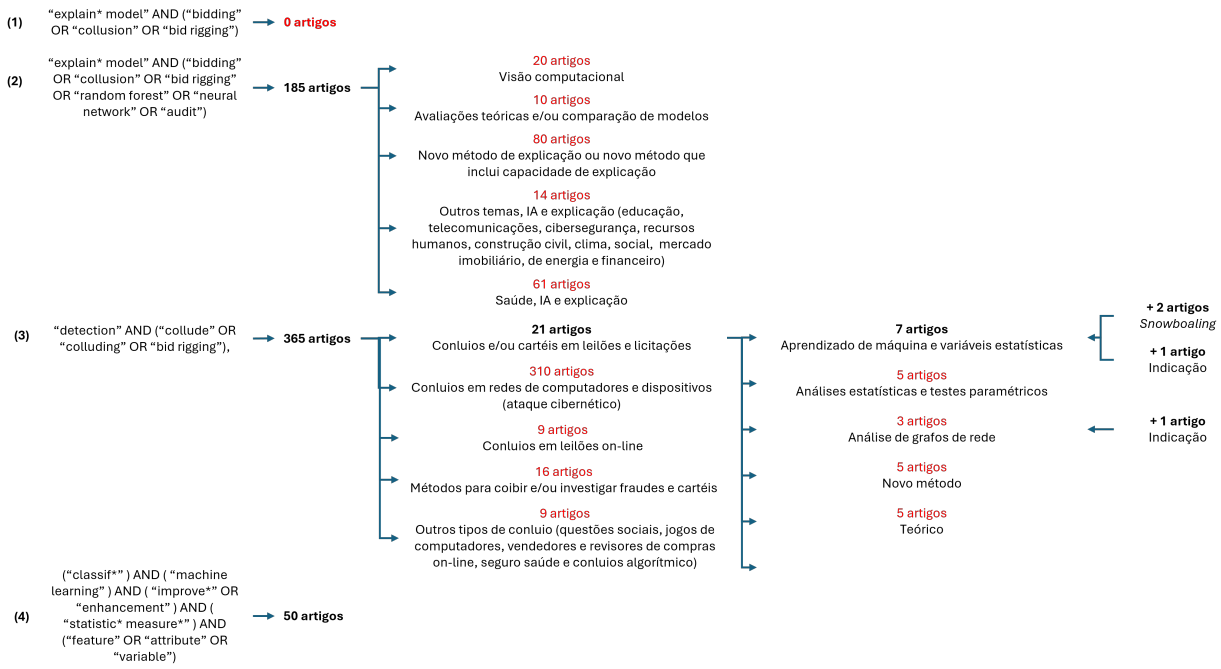



Figura III.1: Pesquisa bibliográfica. (1) Pesquisa inicial sobre aplicações de modelos explicáveis e conluios não retornou artigos. (2) Pesquisa mais abrangente por modelos explicáveis, abrindo a possibilidade de outras aplicações, retornou 185 artigos, mas nenhum relacionado a conluios. (3) Pesquisa mais abrangente sobre métodos de detecção de conluio, retornou diversos artigos, sendo que foram selecionados 7 artigos de interesse. (4) Pesquisa sobre utilização de variáveis estatísticas para aprimorar desempenho, mostra outros cenários que essa configuração foi experimentada.

Para realizar uma pesquisa mais abrangente sobre métodos de detecção de conluio, foi utilizada a*string* de busca "detection" AND ("collude" OR "colluding" OR "bid rigging"), que retornou 365

documentos. Dentre os resultados, há abordagens que utilizam dados classificados e não classificados. Destes últimos, destacam-se a utilização de métodos de *cluster* para separar os dados e enriquecimento de variáveis estatísticas. No entanto, não foi encontrado artigo que referencie o uso de modelos explicáveis.

Por fim, pesquisou-se literatura relacionada a aprimoramentos de métodos de classificação utilizando medidas estatísticas. Para isso, a seguinte *string* foi utilizada: ("classif*" ) AND ( "machine learning" ) AND ( "improve*" OR "enhancement" ) AND ( "statistic* measure*" ) AND ("feature" OR "attribute" OR "variable"). Dos 50 artigos retornados, há algumas aplicações de variáveis estatísticas para melhoramento dos modelos, mas nenhuma aplicada a identificação de conluios. As subseções abaixo citam os artigos mais relevantes relacionados ao presente tema de pesquisa. Também foram levantados artigos relacionados ao tema que são citados pelos artigos encontrados.

### III.1 Detecção de conluio

Uma abordagem encontrada na literatura para detecção de conluios é a análise econométrica. Em Busu and Busu [2021], os dados são separados em dois agrupamentos (com conluio e competitivo) e são aplicados testes não paramétricos para verificar se os *clusters* são significativamente diferentes em termos de distribuição do ponto de vista estatístico. Em seguida são realizados testes de normalidade e simetria, para comparar as distribuições probabilísticas dos *clusters* com a distribuição normal. Por fim, é realizado teste de hipótese estatística para verificar se havia diferenças significativas entre as categorias de dados. Outros artigos que aplicam testes não paramétricos e abordagens econométrica similar, onde o comportamento de licitação de um grupo de empresas é comparado com um grupo de controle, são Porter and Zona [1999], Bajari and Ye [2003], Aryal and Gabrielli [2013] e Lei et al. [2018].

Há artigos que abordam a identificação de conluios por meio de telas e análises estatísticas, como Ballesteros-Pérez et al. [2015], que propuseram o método *"Quickbnormal-Bid-Detection Metho Ad" (QABDM)* e Signor et al. [2023] que propuseram melhorias para este método, que é incapaz de sinalizar padrões incomuns em lances vencedores. Em linha similar, Imhof et al. [2018] utilizaram análises estatísticas de cenário e Abrantes-Metz et al. [2006] analisaram o movimento de preços e indicadores estatísticos que podem caracterizar existência de conluio. Jiménez and Perdiguero [2012] analisaram e confirmaram, estatisticamente, que há uma relação positiva entre monopólio e rigidez de preços, e que por meio de filtro de variância é possível detectar comportamentos não competitivos, ao comparar com *benchmark* apropriado.

A identificação de conluios também é abordada na literatura por meio de análises dos relacionamentos entre os participantes em grafos. Greige et al. [2022] propuseram um sistema que destaca os jogadores que mais provavelmente exibem comportamentos de conluio em jogo *multiplayer*. O mé-

todo analisa relações pessoais entre os jogadores e utiliza ferramentas de teoria dos grafos. Hu et al. [2013] propuseram um novo modelo para detecção de conluio baseado na teoria dos hiper-grafos, onde buscam encontrar *clusters* fortemente conectados. O trabalho de Amaral [2020] utilizou grafos de mineração sobre redes para identificar conluios.

Alguns artigos encontrados abordam métodos de identificação de conluios em leilões *online*. Dadfarnia et al. [2020] apresentaram um algoritmo para detecção dessas fraudes que consiste na extração de valores agregados (número leilões participados, número licitantes distintos no mesmo leilão etc.), cálculo iterativo de um escore de anomalias (que atualiza e propaga reputações a partir de um pequeno conjunto de usuários rotulados conforme os relacionamentos) e SVM para aprender a função de decisão.

Por fim, há um conjunto de artigos que combinam variáveis estatísticas e aprendizado de máquina para investigar conluios. Alguns desses, utilizaram dados rotulados, como Silveira et al. [2022], Wallimann et al. [2023], Huber and Imhof [2019] e García Rodríguez et al. [2022]. A pesquisa de García Rodríguez et al. [2022] faz experimentos muito parecidos com este trabalho, onde são executados diversos algoritmos de aprendizado de máquina para detectar conluios, em diferentes configurações de atributos (todos, somente os comuns entre os conjuntos de dados, todos os atributos + atributos estatísticos ou atributos comuns + atributos estatísticos) e avalia e compara o desempenho das abordagens.

Os artigos que não utilizam dados rotulados, precisam aplicar alguma estratégia de aprendizado não supervisionado, normalmente métodos de agrupamentos, como em Silveira et al. [2023], Júnior et al. [2023] e Imhof and Wallimann [2021]. Silveira et al. [2023] usaram *Gaussian Mixture Clustering Model (GMCM)*, utilizando estatísticas de "Desvio padrão" e "Coeficiente de variação" para separar os conjuntos de dados. Júnior et al. [2023] propuseram agrupar as empresas que participam das mesmas licitações fraudulentas, quantificando a probabilidade de cada grupo ser conluio. Imhof and Wallimann [2021] isolaram grupo suspeito por testes estatísticos, esse isolamento separa subgrupos de três empresas concorrentes em ao menos três licitações.

Um levantamento sobre as telas estatísticas utilizadas, apresentado na Tabela III.1, mostra que SKEW e CV foram utilizadas em todos os artigos. Silveira et al. [2022] afirmaram ter encontrado evidências que a SKEW, KURT e *Spread* ou Dispersão (SPD) podem aumentar o desempenho dos algoritmos. Wallimann et al. [2023] avaliaram que as execuções dos algoritmos com telas estatísticas de resumo aumentaram a taxa de classificação em 3 a 5,7% e diminuiu taxa de erro em 22,2%. Huber and Imhof [2019] concluíram por seus experimentos, que as telas mais valiosas são CV e *Relative Difference Normalized* (RDNOR), enquanto que Silveira et al. [2023] avaliaram CV e SKEW como as características consistentemente observadas nos cartéis de fraudes e licitação.

Sobre a utilização de métricas de avaliação, as métricas mais utilizadas são a Acurácia, citada

Tabela III.1: Telas estatísticas utilizados nos trabalhos relacionados. Para efeito de comparação, foi adicionada uma coluna indicando com um X os algoritmos abordados neste trabalho.

| Telas | Silveira et al. [2022] | Wallimann et al. [2023] | Huber and Imhof [2019] | García Rodríguez et al. [2022] | Silveira et al. [2023] | Júnior et al. [2023] | Imhof and Wallimann [2021] | Este trabalho |
|---|---|---|---|---|---|---|---|---|
| SKEW | X | X | X | X | X | X | X | X |
| Característica da probabilidade | - | - | - | - | - | X | - | - |
| CV | X | X | X | X | X | X | X | X |
| KURT | X | X | X | X | - | X | - | X |
| Desvio Padrão (DP) | X | - | X | - | - | - | - | - |
| Diferença Absoluta (DIFFA) | - | - | X | - | - | - | X | - |
| Diferença Percentual (DIFFP) | - | X | X | X | - | X | X | - |
| SPD | X | X | - | X | - | X | X | - |
| *Relative Difference Alternative* (RDALT) | - | X | - | - | - | - | X | - |
| RDNOR | - | X | X | - | - | - | X | - |
| *Relative Difference* (RD) | - | X | X | X | - | X | X | - |
| Kolmogorov-Smirnov (KS) | - | X | - | X | - | X | X | - |
| Média propostas/lances na licitação | - | - | X | - | X | - | - | |
| Média de todos CV | - | X | - | - | - | - | - | |
| Mediana de todos CV | - | X | - | - | - | - | - | |
| Máximo de todos CV | - | X | - | - | - | - | - | |
| Mínimo de todos CV | - | X | - | - | - | - | - | |
| Número de participantes | - | - | X | - | - | X | - | |
| Número de participantes por proposta (1 lance) | - | - | - | - | X | - | - | |
| Número participantes na mesma contratação | - | - | - | - | X | - | - | |
| Número total de propostas de cada empresa | - | - | - | - | X | - | - | |
| Valor da proposta vencedora | - | - | - | - | - | X | - | |

em cinco dos sete artigos e AUC, citado em três dos sete artigos, vide Tabela III.2.

Tabela III.2: Métricas de avaliação utilizadas nos trabalhos relacionados.

| Métricas | Silveira et al. [2022] | Wallimann et al. [2023] | Huber and Imhof [2019] | García Rodríguez et al. [2022] | Silveira et al. [2023] | Júnior et al. [2023] | Imhof and Wallimann [2021] |
|---|---|---|---|---|---|---|---|
| Acurácia | X | X | X | X | - | - | X |
| Acurácia balanceada | - | - | - | X | - | - | - |
| Acurácia nula | X | - | - | - | - | - | - |
| AUC | X | - | - | - | X | X | - |
| Erro | X | - | - | - | - | | - |
| Especificidade | X | - | - | - | - | - | - |
| F-1 | - | - | - | X | - | - | - |
| *Mean Square Error* (MSE) | - | - | X | - | - | - | - |
| Precisão | X | - | - | X | - | - | - |
| Recall | - | - | - | X | - | - | - |
| Sensibilidade | X | - | - | - | - | - | - |

No levantamento dos algoritmos de aprendizado, vide Tabela III.3, Silveira et al. [2022] obtiveram melhores resultados com o classificador logit, Wallimann et al. [2023] obtiveram bons resultados com seu método, utilizando *Random Forest*, Huber and Imhof [2019] encontraram resultados similares para *Lasso* e *Ensemble method*. Para García Rodríguez et al. [2022], os algoritmos que tiveram melhores resultados foram os *Ensemble methods* (*Extra Trees, Random Forest, Ada Boost e Gradient Boosting*. Os resultados na tabela estão sumarizados com o valor médio de acurácia de cada algoritmo. Para os resultados de García Rodríguez et al. [2022], também foi identificada a melhor configuração de atributos (1: todos os atributos; 2: todos os atributos + telas estatísticas; 3: atributos comuns; 4: atributos comuns + telas estatísticas).

No conjunto de algoritmos que não utilizam dados pré-classificados, Silveira et al. [2023] obtiveram bons resultados com seu método, utilizando QDA na classificação. Júnior et al. [2023] obtiveram bons resultados com SVM, Redes Neurais e *Random Forest* na configuração 1 (características originais + telas estatísticas). No entanto, o melhor resultado foi para Redes Neurais na configuração 3 (características básicas + probabilidade calculada). Já Imhof and Wallimann [2021] obtiveram melhor resultado com *Random Forest*.

Do levantamento apresentado na Tabela III.3, destacam-se os algoritmos *Random Forest*, utilizado em cinco dos sete artigos e Redes Neurais e Lasso utilizados em três artigos.

## III.2 Modelos explicáveis

Quanto à aplicação de modelos explicáveis, Wang et al. [2023] desenvolveram modelos explicáveis para predição de ventilação mecânica nos pacientes de síndrome respiratória aguda, usando abordagem de aprendizagem de máquina. O trabalho utilizou três métodos de aprendizado de máquina descritivo baseado no SHAP, LIME e *moDel Agnostic Language for Exploration and eXplanation* (DALEX).

Du et al. [2023] desenvolveram modelos capazes de prever a probabilidade de nascimento de recém-nascidos grandes para a idade gestacional em mulheres com sobrepeso e obesidade com aproximadamente 21 semanas de gestação. O trabalho utilizou os algoritmos de aprendizado de máquina *Random Forest*, SVM, reforço adaptativo (*AdaBoost*) e XGBoost e aplicação de explicações agnós-

Tabela III.3: Algoritmos de aprendizado de máquina utilizados nos trabalhos relacionados. A métrica citada é acurácia. Para efeito de comparação, foi adicionada uma coluna indicando com um X os algoritmos abordados neste trabalho.

| Algoritmo | Silveira et al. [2022] | Wallimann et al. [2023] | Huber and Imhof [2019] | García Rodríguez et al. [2022] | Silveira et al. [2023] (AUC) | Júnior et al. [2023] (AUC) | Imhof and Wallimann [2021] | Este trabalho |
|---|---|---|---|---|---|---|---|---|
| *AdaBoost* | - | - | - | 86,8% (2) | - | - | - | X |
| Árvore de decisão | - | - | - | - | - | 97,02% (3) | - | X |
| *Ensemble method* | - | - | 84% | - | - | - | - | X |
| *Extra Trees* | - | - | - | 87,77% (2) | - | - | - | X |
| *Gradient Boosting* | - | - | - | 83,28% (2) | - | - | - | X |
| K-NN | - | - | - | 77,77% (3) | - | ? | - | X |
| *Lasso* | 96,71% | - | 84% | - | - | - | 92,3% | X |
| *Logit* | 99,29% | - | - | - | - | - | - | X |
| *Naive Bayes (Bernoulli)* | - | - | - | 78,58% (3) | - | - | - | X |
| *Naive Bayes (Gaussian)* | - | - | - | 77,67% (1) | - | - | - | X |
| Redes Neurais | 95,08% | - | - | 79.38% (3) | - | 99,53% (3) | - | X |
| Processo Gaussiano | - | - | - | 57,03% (4) | - | - | - | - |
| QDA | - | - | - | - | 80,25% | - | - | X |
| *Random forest* | 95,52% | 83% | - | 86,87% (2) | - | 87,62% (1) | 94,9% | X |
| *Ridge Logistic* | - | - | - | - | - | | - | X |
| SGD | - | - | - | 70% (2) | - | - | - | X |
| SVM | - | - | - | 62,58% (3) | - | 97,62% (1) | 93,9% | X |

ticas de modelos interpretáveis locais (LIME).

Nguyen and Byeon [2023] examinaram modelo de conjunto de *stacking ensemble* em LIME para prever a depressão de pacientes com doença de Parkinson. Eles utilizaram *Logistic Regression* como meta modelo e outros cinco modelos *Light Gradient-Boosting Machine* (LGBM), K-NN, *Random Forest*, *Extra Trees*, and *AdaBoost*. Por fim, eles combinaram os modelos *stacking ensemble* com LIME para prover a explicabilidade.

Jain and Jana [2022] propuseram um novo algoritmo *"eXplainable Reasonably Randomized Forest"* (XRRF). O XRRF consiste em quatro etapas: desempenho de aprendizado, interpretabilidade intrínseca, precisão do modelo e explicabilidade.

Iban [2022] analisou como os métodos de XAI podem ser integrados com estudos de avaliação imobiliária em massa. O trabalho utilizou *Random Forest*, XGBoost, LGBM e *Gradient Boosting* para treinar os modelos preditivos, e os modelos treinados com variáveis selecionadas foram interpretados localmente usando o método SHAP.

### III.3 Aprimoramento dos modelos de classificação

No que tange a utilização de medidas estatísticas para aprimoramento de modelos de classificação, Fayaz et al. [2021] propuseram uma nova metodologia baseada em características estatísticas e diferentes algoritmos de aprendizagem. Na etapa de extração de atributos são calculadas as medidas estatísticas média, Variância (VAR), SKEW, KURT, entropia, energia, contraste, homogeneidade e correlação para cada canal RGB (vermelho, verde e azul) da imagem de ressonância magnética. Na etapa de classificação, são utilizados os algoritmos de RNA, K-NN, Árvore de decisão e *Naïve Bayes*.

O estudo de Abbas and Turky Rashid [2024] é sobre melhorias no reconhecimento de gestos utilizando Mecanomiografia (MMG). As medidas estatísticas (média, erro padrão, mediana, moda, DP, variância da amostra, KURT, SKEW, amplitude, mínimo, máximo e soma) são derivadas do sinal MMG. Os algoritmos de classificação Árvore de decisão, LDA, *Naive Bayes*, SVM e K-NN são utilizados no estudo.

Fegade et al. [2023] propuseram um recomendador de cultura agrícola mais adequado baseado nas condições geográficas-geológicas. O recomendador usa medidas estatísticas de média, DP, SKEW, KURT, recursos Peak2Peak e RMS, os algoritmos de aprendizado de máquina *Naïve Bayes*, Árvore de decisão, K-NN e modelo de conjunto híbrido para classificação e recomendação de culturas.

### III.4 Considerações sobre trabalhos relacionados

Os trabalhos relacionados demonstram ser possível caracterizar conluios utilizando variáveis estatísticas, muitas vezes referenciadas como telas estatísticas. Os trabalhos de Imhof et al. [2018], Abrantes-Metz et al. [2006] e Jiménez and Perdiguero [2012] utilizaram análises estatísticas para este fim. Outros trabalhos demonstraram ser possível combinar aprendizado de máquina, com dados rotulados e variáveis estatísticas para detectar conluios, como os Silveira et al. [2022], Wallimann et al. [2023], Huber and Imhof [2019] e García Rodríguez et al. [2022].

O trabalho de García Rodríguez et al. [2022] é o mais similar a este. Assim como este trabalho, ele faz experimentos com diferentes configurações de atributos (todos atributos, atributos comuns e variáveis estatísticas); diferentes modelos de aprendizado (*AdaBoost*, *Extra Trees*, *Gradient Boosting*, K-NN, *Naive Bayes (Bernoulli)*, *Naive Bayes (Gaussian)*, Redes Neurais, Processo Gaussiano, *Random Forest*, SGD e SVM); e variáveis estatísticas (SKEW, CV, KURT, DIFFP, SPD, RD e KS). As métricas utilizadas para comparação do desempenho dos algoritmos e configurações foram acurácia, acurácia balanceada, precisão, *recall*, F1 e uma métrica de erro.

As diferenças deste trabalho e do García Rodríguez et al. [2022] englobam: García Rodríguez et al. [2022] faz experimentos com seis conjuntos de dados contendo ações de conluio e competitivas (Brasil, Itália, Japão, dois conjuntos da Suíça e Estados Unidos), cada conjunto possui um cenário de conluio comprovado, enquanto neste trabalho há apenas um conjunto de dados que reúne nove cenários de conluios comprovados; ele calcula as variáveis estatísticas para cada leilão, neste trabalho o cálculo é feito agrupando por itens e rótulo do conluio; ele utiliza a mais as variáveis estatísticas DIFFP, SPD, RD e KS, enquanto este trabalho utiliza a mais o *outlier* e Quantidade de participantes da licitação; ele experimenta a mais o algoritmo de processo gaussiano, neste trabalho a mais são os algoritmos Árvore de decisão, Regressão Logística, *Ridge*, *Lasso*, XGBoost e *OneClassSVM*; ele avalia o desempenho com diferentes métricas, mas apresenta no artigo um cálculo de erro que considera uma média dessas métricas, neste trabalho é utilizado o MCC.

Por fim, este trabalho explora diferentes abordagens de explicação de modelos, tais como importância de atributos, SHAP e LIME. Este experimento não é feito por nenhum trabalho relacionado à detecção de conluios. Também é explorada diferentes abordagens de seleção de atributos e balanceamento de conjunto de dados, que não são avaliadas em outros artigos relacionados, incluindo García Rodríguez et al. [2022].

# Capítulo IV Metodologia

A metodologia desse trabalho é baseada no método KDD [Goldschmidt and Passos, 2005]. Este método é dividido em três etapas: pré-processamento, mineração de dados e pós-processamento. Para essa aplicação foram estendidas as etapas de enriquecimento, adicionando variáveis estatísticas e explicação dos modelos, conforme mostra a Figura IV.1 e cujas etapas são descritas nas próximas subseções.

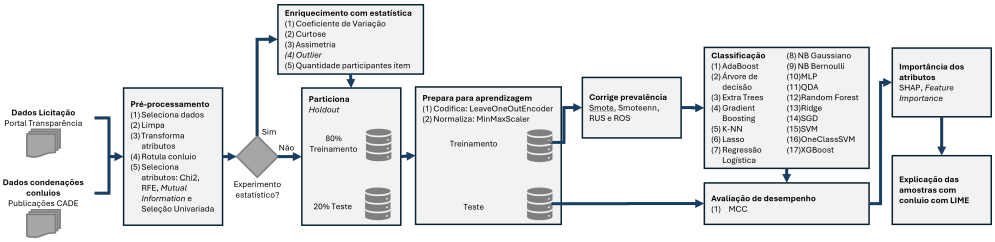



Figura IV.1: Metodologia para aprimoramento de classificação de conluios.

## IV.1 Coleta dos dados

### IV.1.1 Dados de licitação

Os dados de licitação utilizados são referentes a seis anos (2013 a 2018) de licitações do poder executivo da Administração Pública Federal. A escolha desse período foi porque o ano de 2013 é o mais antigo disponível e o ano de 2018 é o ano mais recente com conluios identificados e condenados pelo CADE. Os dados baixados são de domínio público, disponibilizados pela CGU no Portal da Transparência do Governo Federal do Brasil[1], em formato CSV, e cuja origem é o Sistema Integrado de Administração de Serviços Gerais (SIASG)[2]. Arquivos de Licitação, Item de Licitação e Participantes das Licitações são publicados mensalmente no portal.

O Portal da Transparência também disponibiliza dicionário de dados dos arquivos[3]. As análises a seguir são relativas às observações dos dados e leitura e entendimento do dicionário. Na Figura IV.2 estão representados os atributos existentes em cada tipo de arquivo. Os principais pontos do dicionário de dados estão destacados a seguir. O processo de coleta e junção dos dados ocorreu conforme Figura IV.3.

[1]https://portaldatransparencia.gov.br/download-de-dados/licitacoes
[2]https://portaldatransparencia.gov.br/origem-dos-dados
[3]https://portaldatransparencia.gov.br/pagina-interna/603389-dicionario-de-dados-licitacoes

- Algumas informações possuem colunas texto e código, podendo ser utilizada apenas uma delas.
- Não há identificadores únicos para os registros. No entanto, as colunas número licitação, UG (nome e código da unidade gestora), modalidade de compra (nome e código), número do processo e órgão (nome e código órgão) se repetem em todos os arquivos e as colunas item compra (nome e código) e descrição se repetem nos arquivos Item Licitação e Participantes Licitação e podem ser usadas para relacionar os arquivos.
- Todos os arquivos de Licitação, Item Licitação e Participantes Licitação, do período analisado, possuem as colunas pretendidas para junção preenchidas, ou seja, não há dados faltantes para essas colunas.
- O método de junção das tabelas foi a junção interna.

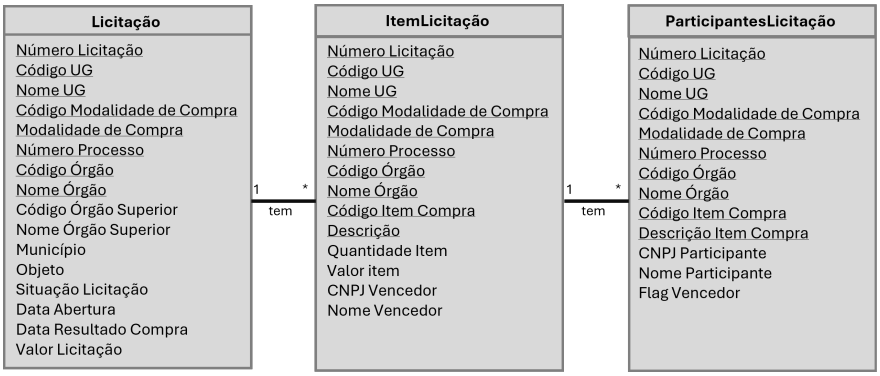



Figura IV.2: Atributos existentes em cada tipo de arquivo baixado do portal da transparência. Sublinhado os atributos que são comuns a todos os arquivos.

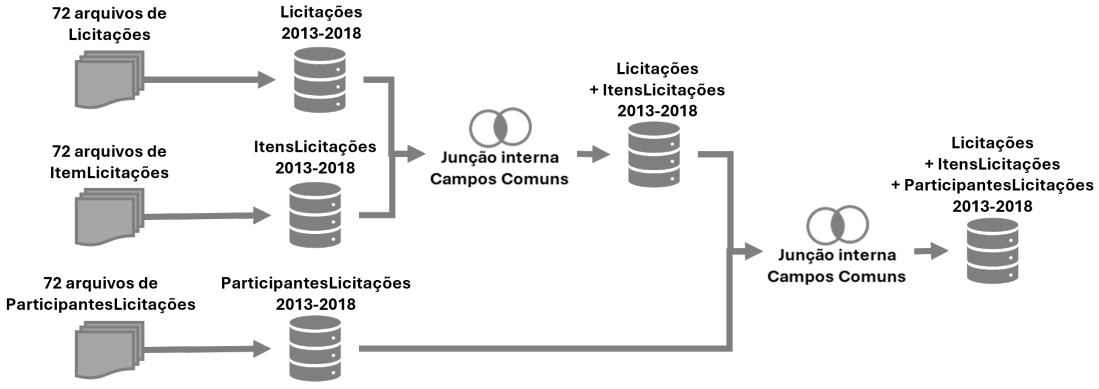



Figura IV.3: Processo para coleta e junção dos arquivos baixados do portal de transparência para os anos de 2013 até 2018.

### IV.1.2 Dados de conluio

O CADE publica[4] notícias relacionadas a investigações e condenações de conluios. As notícias normalmente descrevem o objeto da investigação, período do conluio, empresas investigadas e o *link* para o processo administrativo, onde é possível entender o processo em mais detalhes e complementar alguma informação que esteja ausente na notícia. Para fazer um levantamento de condenações de conluios pelo CADE, os seguintes termos foram pesquisados no Google[5]: "condena", "licitações públicas", site: https://www.gov.br/cade.

Para realizar a correta rotulação dos conluios nos dados de licitação é necessário fazer a compatibilização dos nomes dos participantes dos conluios, indicados no site do CADE, com os nomes dos participantes das licitações, indicados no portal de transparência do CGU. Essa atividade precisa ser feita manualmente, verificando para cada participante do conluio, o seu nome nos dados de licitação.

## IV.2 Pré-processamento

A Figura IV.4 mostra uma visão geral de todo o processo de tratamento dos dados. O resultado desse processo é um conjunto de dados comum para ser utilizado pelos diversos experimentos de melhores abordagens de seleção de atributos, balanceamento e algoritmos de aprendizado de máquina, além de ser base para as execuções das etapas posteriores de explicação do modelo. Uma visão geral dos resultados das execuções das etapas no conjunto de dados é apresentada na Figura IV.5. As subseções a seguir detalham as atividades contidas na Figura IV.4.

### IV.2.1 Seleção de dados

Este estudo está centrado na identificação de conluios entre participantes de uma licitação. Logo, casos de compra na modalidade "Inexigibilidade de Licitação" e "Dispensa de Licitação", onde tipicamente não há concorrência, foram eliminados do conjunto de dados. Um atributo importante para enriquecimento do modelo com variáveis estatísticas é o valor do Item, logo, foram eliminadas linhas que este atributo não está preenchido. Por fim, o conjunto de dados foi filtrado para manter apenas os itens de compra identificados nos levantamentos de conluio. Como as variáveis estatísticas são geradas a partir dos valores de cada item, separando os grupos com e sem conluio, foram mantidas apenas itens com casos de conluios classificados. A finalidade é comparar o comportamento dos dois grupos: itens em uma situação de conluio e sem a situação de conluio.

[4]https://www.gov.br/cade/
[5]https://www.google.com.br/. Pesquisa realizada em 06/12/2023.

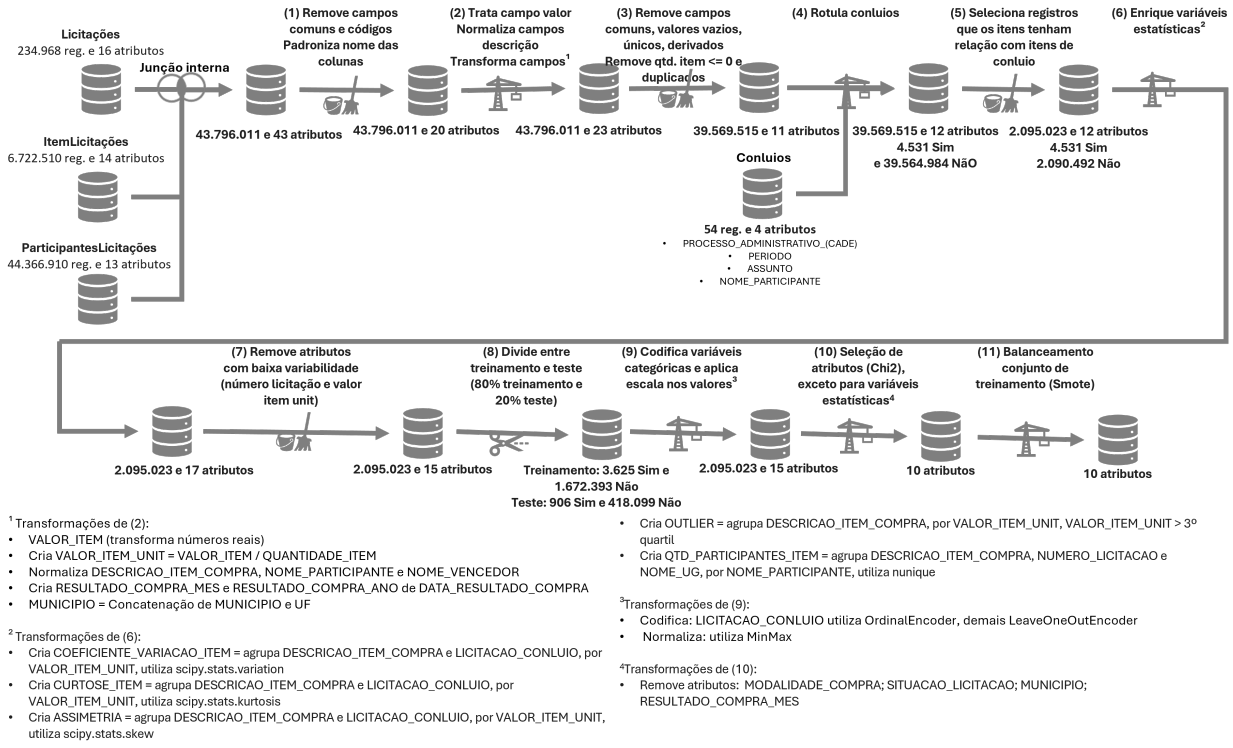


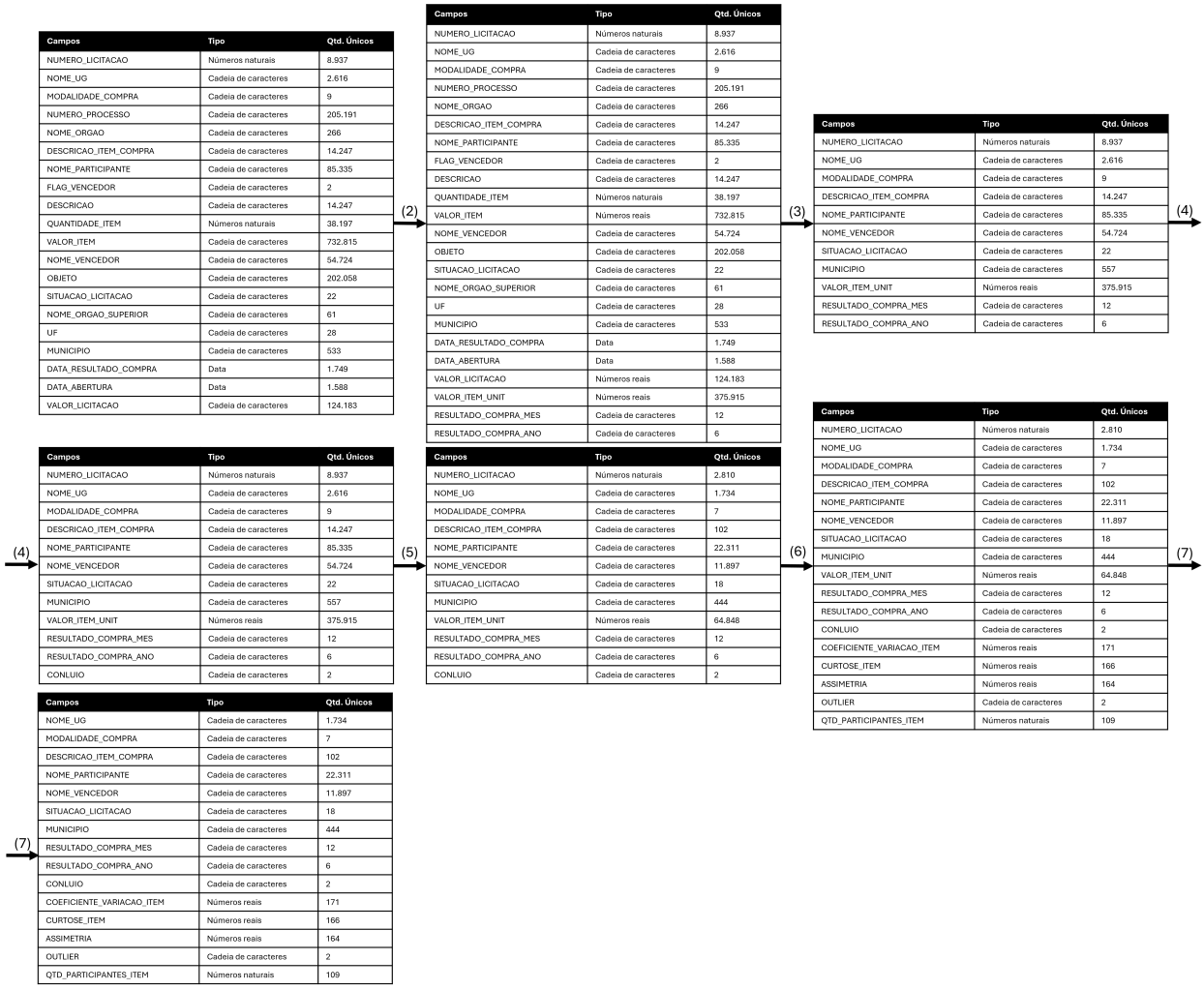

Figura IV.4: Visão geral das atividades do pré-processamento.

### IV.2.2 Seleção de atributos

As primeiras análises que podem ser executadas para decidir pela eliminação de atributos é sobre valores únicos, faltantes e baixa variabilidade. O entendimento do domínio de dados de cada campo também é importante. No modelo, os atributos Descrição, Nome Órgão, Nome Órgão Superior, *Flag* Vencedor e Valor Licitação podem ser eliminados, conforme as seguintes observações:

- Descrição possui os mesmos valores que Item Compra;
- Gestora está vinculado ao Nome Órgão que está vinculado ao Nome Órgão Superior. Unidade Gestora (UG) é a mais granular e mais diretamente envolvida na licitação;
- Flag Vencedor é um atributo que pode ser calculado como 'Sim', quando os campos Participante e Vencedor são iguais e 'Não', caso contrário;
- Valor Licitação traz o valor total da licitação que pode envolver diversos itens e diversas quantidades. Essa composição é particular de cada licitação, logo mais difícil de comparar e derivar conhecimentos. Para análise desse modelo, o mais interessante é avaliar e comparar o preço do item do que o preço global da licitação.

Além disso, para focar os modelos de aprendizado nos atributos de maior relevância, foi aplicado um método de seleção de atributos. Os métodos de seleção de atributos avaliados foram: seleção




| Campos | Tipo | Qtd. Únicos |
|---|---|---|
| NUMERO_LICITACAO | Números naturais | 8.937 |
| NOME_UG | Cadeia de caracteres | 2.616 |
| MODALIDADE_COMPRA | Cadeia de caracteres | 9 |
| NUMERO_PROCESSO | Cadeia de caracteres | 205.191 |
| NOME_ORGAO | Cadeia de caracteres | 266 |
| DESCRICAO_ITEM_COMPRA | Cadeia de caracteres | 14.247 |
| NOME_PARTICIPANTE | Cadeia de caracteres | 85.335 |
| FLAG_VENCEDOR | Cadeia de caracteres | 2 |
| DESCRICAO | Cadeia de caracteres | 14.247 |
| QUANTIDADE_ITEM | Números naturais | 38.197 |
| VALOR_ITEM | Cadeia de caracteres | 732.815 |
| NOME_VENCEDOR | Cadeia de caracteres | 54.724 |
| OBJETO | Cadeia de caracteres | 202.058 |
| SITUACAO_LICITACAO | Cadeia de caracteres | 22 |
| NOME_ORGAO_SUPERIOR | Cadeia de caracteres | 61 |
| UF | Cadeia de caracteres | 28 |
| MUNICIPIO | Cadeia de caracteres | 533 |
| DATA_RESULTADO_COMPRA | Data | 1.749 |
| DATA_ABERTURA | Data | 1.588 |
| VALOR_LICITACAO | Cadeia de caracteres | 124.183 |

| Campos | Tipo | Qtd. Únicos |
|---|---|---|
| NUMERO_LICITACAO | Números naturais | 8.937 |
| NOME_UG | Cadeia de caracteres | 2.616 |
| MODALIDADE_COMPRA | Cadeia de caracteres | 9 |
| NUMERO_PROCESSO | Cadeia de caracteres | 205.191 |
| NOME_ORGAO | Cadeia de caracteres | 266 |
| DESCRICAO_ITEM_COMPRA | Cadeia de caracteres | 14.247 |
| NOME_PARTICIPANTE | Cadeia de caracteres | 85.335 |
| FLAG_VENCEDOR | Cadeia de caracteres | 2 |
| DESCRICAO | Cadeia de caracteres | 14.247 |
| QUANTIDADE_ITEM | Números naturais | 38.197 |
| VALOR_ITEM | Números reais | 732.815 |
| NOME_VENCEDOR | Cadeia de caracteres | 54.724 |
| OBJETO | Cadeia de caracteres | 202.058 |
| SITUACAO_LICITACAO | Cadeia de caracteres | 22 |
| NOME_ORGAO_SUPERIOR | Cadeia de caracteres | 61 |
| UF | Cadeia de caracteres | 28 |
| MUNICIPIO | Cadeia de caracteres | 533 |
| DATA_RESULTADO_COMPRA | Data | 1.749 |
| DATA_ABERTURA | Data | 1.588 |
| VALOR_LICITACAO | Números reais | 124.183 |
| VALOR_ITEM_UNIT | Números reais | 375.915 |
| RESULTADO_COMPRA_MES | Cadeia de caracteres | 12 |
| RESULTADO_COMPRA_ANO | Cadeia de caracteres | 6 |

| Campos | Tipo | Qtd. Únicos |
|---|---|---|
| NUMERO_LICITACAO | Números naturais | 8.937 |
| NOME_UG | Cadeia de caracteres | 2.616 |
| MODALIDADE_COMPRA | Cadeia de caracteres | 9 |
| DESCRICAO_ITEM_COMPRA | Cadeia de caracteres | 14.247 |
| NOME_PARTICIPANTE | Cadeia de caracteres | 85.335 |
| NOME_VENCEDOR | Cadeia de caracteres | 54.724 |
| SITUACAO_LICITACAO | Cadeia de caracteres | 22 |
| MUNICIPIO | Cadeia de caracteres | 557 |
| VALOR_ITEM_UNIT | Números reais | 375.915 |
| RESULTADO_COMPRA_MES | Cadeia de caracteres | 12 |
| RESULTADO_COMPRA_ANO | Cadeia de caracteres | 6 |

| Campos | Tipo | Qtd. Únicos |
|---|---|---|
| NUMERO_LICITACAO | Números naturais | 8.937 |
| NOME_UG | Cadeia de caracteres | 2.616 |
| MODALIDADE_COMPRA | Cadeia de caracteres | 9 |
| DESCRICAO_ITEM_COMPRA | Cadeia de caracteres | 14.247 |
| NOME_PARTICIPANTE | Cadeia de caracteres | 85.335 |
| NOME_VENCEDOR | Cadeia de caracteres | 54.724 |
| SITUACAO_LICITACAO | Cadeia de caracteres | 22 |
| MUNICIPIO | Cadeia de caracteres | 557 |
| VALOR_ITEM_UNIT | Números reais | 375.915 |
| RESULTADO_COMPRA_MES | Cadeia de caracteres | 12 |
| RESULTADO_COMPRA_ANO | Cadeia de caracteres | 6 |
| CONLUIO | Cadeia de caracteres | 2 |

| Campos | Tipo | Qtd. Únicos |
|---|---|---|
| NUMERO_LICITACAO | Números naturais | 2.810 |
| NOME_UG | Cadeia de caracteres | 1.734 |
| MODALIDADE_COMPRA | Cadeia de caracteres | 7 |
| DESCRICAO_ITEM_COMPRA | Cadeia de caracteres | 102 |
| NOME_PARTICIPANTE | Cadeia de caracteres | 22.311 |
| NOME_VENCEDOR | Cadeia de caracteres | 11.897 |
| SITUACAO_LICITACAO | Cadeia de caracteres | 18 |
| MUNICIPIO | Cadeia de caracteres | 444 |
| VALOR_ITEM_UNIT | Números reais | 64.848 |
| RESULTADO_COMPRA_MES | Cadeia de caracteres | 12 |
| RESULTADO_COMPRA_ANO | Cadeia de caracteres | 6 |
| CONLUIO | Cadeia de caracteres | 2 |

| Campos | Tipo | Qtd. Únicos |
|---|---|---|
| NUMERO_LICITACAO | Números naturais | 2.810 |
| NOME_UG | Cadeia de caracteres | 1.734 |
| MODALIDADE_COMPRA | Cadeia de caracteres | 7 |
| DESCRICAO_ITEM_COMPRA | Cadeia de caracteres | 102 |
| NOME_PARTICIPANTE | Cadeia de caracteres | 22.311 |
| NOME_VENCEDOR | Cadeia de caracteres | 11.897 |
| SITUACAO_LICITACAO | Cadeia de caracteres | 18 |
| MUNICIPIO | Cadeia de caracteres | 444 |
| VALOR_ITEM_UNIT | Números reais | 64.848 |
| RESULTADO_COMPRA_MES | Cadeia de caracteres | 12 |
| RESULTADO_COMPRA_ANO | Cadeia de caracteres | 6 |
| CONLUIO | Cadeia de caracteres | 2 |
| COEFICIENTE_VARIACAO_ITEM | Números reais | 171 |
| CURTOSE_ITEM | Números reais | 166 |
| ASSIMETRIA | Números reais | 164 |
| OUTLIER | Cadeia de caracteres | 2 |
| QTD_PARTICIPANTES_ITEM | Números naturais | 109 |

Figura IV.5: Visão geral do resultado das atividades do pré-processamento sobre o conjunto de dados.

univariada, chi2, *mutual information* e RFE. Foi utilizada a implementação desses métodos da biblioteca do *python sklearn.feature_selection*. Em todos eles, quanto maior o valor, mais importante o atributo é para o modelo. Para avaliar o desempenho de cada método foram executados experimentos com os algoritmos: MLP, Regressão Logística e *Random Forest*, vide capítulo V. A métrica utilizada para avaliação foi a MCC. Os experimentos foram executados incluindo e desconsiderando as variáveis estatísticas, a fim de garantir que o método escolhido possua o melhor resultado para ambos os conjuntos de dados. A seleção dos atributos foi feita apenas para os atributos que não são variáveis estatísticas.

### IV.2.3 Limpeza

As atividades de limpeza realizadas foram relativas a: eliminação de caracteres especiais (acentos, cedilhas etc.) dos campos participante, vencedor e item; substituição da vírgula por ponto no campo valor item, seguida de transformação para valor numérico com ponto flutuante.

### IV.2.4 Extração de atributos

Alguns atributos foram transformados (unidos, generalizados ou calculados) para obter novos com o mesmo significado ou com maior potencial de generalização para o aprendizado do modelo. Eventualmente é necessária a criação de atributos temporários para auxiliar nos cálculos ou filtros, mas estes são eliminados ao final do processo. Os atributos transformados foram:

- o Município está sempre relacionado a uma única Unidade da Federação (UF), logo, é possível substituir os dois para um atributo com os valores concatenados;
- Data Resultado Compra é um atributo que pode ser mais generalizado, quebrando em atributos mês (Mês Compra) e ano (Ano Compra);
- mais interessante que Quantidade Item e Valor Item é o Valor Unitário do Item, que é obtido dividindo o Valor Item pela Quantidade Item. No entanto, para evitar divisões por zero, foi necessário eliminar as linhas com Quantidade Item igual ou menor que zero;

### IV.2.5 Rotulação de conluios

Apenas foram rotulados como Sim para conluio as linhas cujo vencedor da licitação é um participante do conluio, durante o período de existência do conluio, em licitações que possuam ao menos um outro participante do conluio, conforme ilustrado na Figura IV.6. Esse método de rotulação considera critérios fortes para indicar a presença de conluios. Contudo, não é possível afirmar com plena certeza que as licitações rotuladas como Sim para conluio são de fato conluio, e como Não, não são. No entanto, avalia-se que os casos que diferem desse critério são residuais e tendem a ser minoria e não atrapalhar a aprendizagem do modelo.

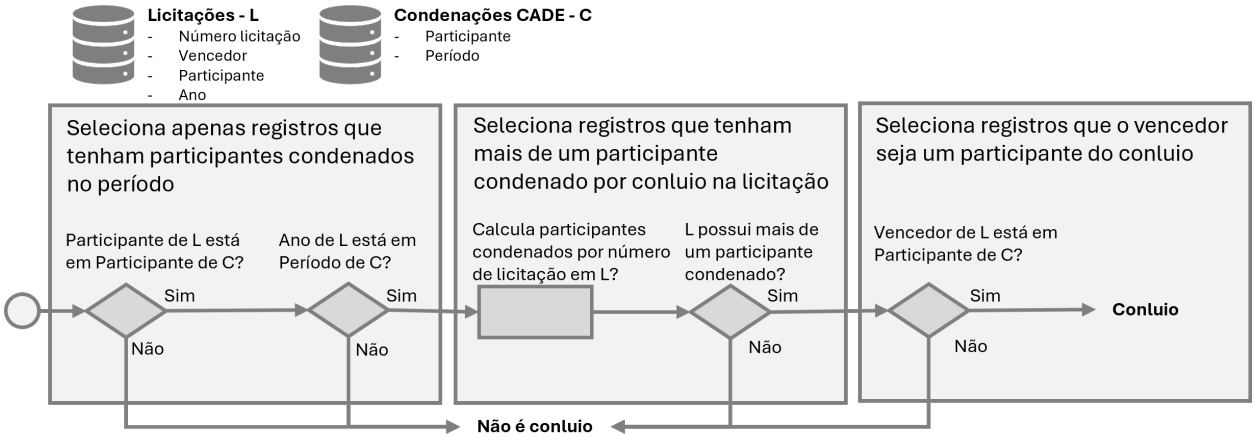



Figura IV.6: Processo para rotulação dos conluios.

### IV.2.6 Enriquecimento (variáveis estatísticas)

Pela análise da literatura, concluiu-se que as variáveis estatísticas mais promissoras para aprimorar o desempenho de classificação dos modelos são SKEW, CV, KURT e SPD. No entanto, este método propõe utilizar as variáveis SKEW, CV, KURT, quantidade de participantes e *outlier*, ao invés da SPD. A SPD não foi utilizada porque para isso seriam necessários os valores de todos os lances das licitações. Alternativamente, esta metodologia propõe a avaliação do atributo *Outlier*, onde são rotuladas como Sim, as licitações que tiverem preços acima do considerado normal.

Os cálculos das variáveis SKEW, CV e KURT utilizaram as funções *skew*, *variation* e *kurtosis*, respectivamente, da biblioteca *scipy.stats* do *python*. Em todos estes, o cálculo é feito para o campo Valor Unitário do Item, agrupando os campos Item de Compra e Licitação Conluio. Já o cálculo Quantidade de Participantes utilizou a função *nunique* da biblioteca pandas do *python*, para contar a quantidade do campo Participantes, agrupando pelos valores Número Licitação, Unidade Gestora e Item da Compra. Por fim, o cálculo do *Outlier* precisou criar um atributo temporário 3º quartil, que utiliza a função *quantile*(0.75) da biblioteca pandas do *python*, agrupando por Item da Compra. O valor do atributo *Outlier* recebe Sim, se o Valor Unitário do Item for maior que o 3º quartil e Não, caso contrário.

## IV.3 Preparação para aprendizagem

### IV.3.1 Particionamento

Como estratégia de particionamento foi utilizado o *holdout*, com a divisão aleatória e estratificada de 80% para treinamento e 20% para teste. Foi utilizada a função *train_test_split* da biblioteca do *python sklearn.model_selection*, com parâmetro *stratify='y'*. Este parâmetro distribui as observações entre treinamento e teste, respeitando as proporções do atributo alvo.

### IV.3.2 Codificação

Para codificação dos atributos categóricos foram executados *Ordinal Encoder* para o atributo alvo (rótulo de conluio) e *Leave One Out Encoder* para os demais. As funções *python* utilizadas foram *OrdinalEncoder* e *LeaveOneOutEncoder*, da biblioteca *python category_encoders*.

Conforme explicações das diferentes abordagens no capítulo II, *Leave One Out Encoder* seria a abordagem mais adequada neste cenário, visto que: o *One-Hot Encoder* explode os atributos categóricos em diversos atributos, aumentando significativamente a dimensionalidade dos dados, além disso, este trabalho objetiva entender a contribuição do atributo de maneira atômica e não pelos seus diferentes valores; o *Ordinal Encoder* requer uma ordem para os dados, não sendo o caso para nenhum dos atributos do conjunto de dados; o *Label Encoder* pode produzir uma ordem

artificial entre as categorias; o *Frequency Encoder* pode não capturar relações complexas entre a categoria e o alvo; por fim o *Target Encoder* pode introduzir fuga de informação.

### IV.3.3 Normalização

O método de normalização (escala) utilizado foi de Mínimos e Máximos. Este método considera o maior valor do atributo como 1 e o menor como 0 e estima uma escala de variação. A função utilizada foi a *MinMaxScaler* da biblioteca *sklearn.preprocessing*.

### IV.3.4 Correção de prevalência

Alguns algoritmos podem tender a favor da classe majoritária, em um conjunto de dados desbalanceados. Problemas de detecção de fraude, de forma geral, tendem a ter um desbalanceamento entre a classificação positiva para fraude e negativa. Por este motivo, foram feitos experimentos, vide capítulo V, com diferentes estratégias de balanceamento para avaliar a mais adequada para compor o *pipeline* da metodologia. As estratégias de balanceamento avaliadas e suas respectivas funções e bibliotecas do *python* são: *Random Under Sampler* (RUS), *RandomUnderSampler* da *imblearn.under_sampling*; *Random Over Sampler* (ROS), *RandomOverSampler* da *imblearn.over_sampling*; *Synthetic Minority Over-sampling Technique* (Smote), *SMOTE* da *imblearn.over_sampling*; SMOTE + ENN (Smoteenn), *SMOTEENN* da *imblearn.combine*; e sem balanceamento. A estratégia que obteve o melhor desempenho geral foi a Smote, no experimento com variáveis estatísticas e com o algoritmo K-NN. As estratégias de balanceamento são aplicadas apenas no conjunto de treinamento.

## IV.4 Classificação

O conjunto de dados utilizado para treinamento da classificação, possui cinco atributos selecionados pelas etapas anteriores e cinco variáveis estatísticas mencionadas no capítulo III. Em linha com os trabalhos relacionados, são utilizadas as características estatísticas CV, KURT e SKEW. A característica *Outlier* foi incluída para testar sua contribuição, uma vez que ela não foi utilizada em nenhum outro trabalho e há uma intuição forte em destacar casos fora do padrão para descoberta de fraudes.

Para os experimentos, foram executados todos os modelos de aprendizado de máquina citados no capítulo II, exceto o processo gaussiano que exige muita memória, o que dificulta a execução para conjunto de dados muito grandes. Também foram executados os algoritmos *OneClassSVM* e XGBoost, não citados nos artigos levantados. Os experimentos utilizaram os parâmetros padrões, com exceção dos algoritmos que possuem o parâmetro *class_weight*. Estes foram atribuídos o valor 'balanced', são eles: Árvore de decisão; *Extra Trees*; Regressão Logística; *Random Forest*; SGD; e

SVM. Os algoritmos executados e suas respectivas funções e bibliotecas do *python* estão listados na Tabela IV.1.

Tabela IV.1: Algoritmos executados e suas respectivas funções e bibliotecas do *python*.

| Algoritmo | Função | Biblioteca |
|---|---|---|
| AdaBoost | AdaBoostClassifier | sklearn.ensemble |
| Árvore de Decisão | DecisionTreeClassifier | sklearn.tree |
| *Extra Trees* | *ExtraTreesClassifier* | *sklearn.ensemble* |
| *Gradient Boosting* | *GradientBoostingClassifier* | *sklearn.ensemble* |
| K-NN | *KNeighborsClassifier* | *sklearn.neighbors* |
| *Lasso* | *Lasso* | *sklearn.linear_model* |
| Regressão Logística | *LogisticRegression* | *sklearn.linear_model* |
| *Naive Bayes (Bernoulli)* | *BernoulliNB* | *sklearn.naive_bayes* |
| *Naive Bayes (Gaussian)* | *GaussianNB* | *sklearn.naive_bayes* |
| MLP | *MLPClassifier* | *sklearn.neural_network* |
| QDA | *QuadraticDiscriminantAnalysis* | *sklearn.discriminant_analysis* |
| *Random Forest* | *RandomForestClassifier* | *sklearn.ensemble* |
| *Ridge* | *RidgeClassifierCV* | *sklearn.linear_model* |
| *SGD* | *SGDClassifier* | *sklearn.linear_model* |
| SVM | *SVC* | *sklearn.svm* |
| *OneClassSVM* | *OneClassSVM* | *sklearn.svm* |
| XGBoost | *XGBClassifier* | *xgboost* |

## IV.5 Avaliação de desempenho

As métricas mais indicadas para avaliação de dados desbalanceados são *F1-score*, MCC e AUC ROC. Neste trabalho, optou-se pela métrica MCC. Para cálculo do MCC foi utilizada a função *matthews_corrcoef* do *sklearn.metrics*. Para efeitos de comparação, também foi treinado um modelo de classificação *Dummy*. Um modelo de classificação *Dummy* é um modelo que faz previsões baseadas em regras simples. A função utilizada foi a *DummyClassifier* da biblioteca *python sklearn.dummy*. A estratégia utilizada foi a constante, que sempre prevê um rótulo constante que é definido (neste caso Sim para Conluio).

## IV.6 Explicação do modelo

### IV.6.1 Importância dos atributos

Para o cálculo de importância dos atributos, alguns algoritmos como SVM (com o *kernel* linear), *Extra Trees*, Regressão Logística, SGD, *Ridge* e XGBoost implementam métodos de cálculos intrínsecos do modelo. Para aqueles que não possuem essa implementação nativa é possível utilizar a abordagem da importância dos atributos por permutação. Esta foi a abordagem adotada para o MLP, K-NN e SVM (*kernel* rbf).

Para obter os valores de importância de cada atributo, para os algoritmos que possuem este cálculo intrínseco, os coeficientes são armazenados em um atributo *coef_* do modelo gerado. Para os demais, foi utilizada a função *permutation_importance* da biblioteca *python* sklearn.inspection.

### IV.6.2 SHAP

O SHAP permite uma visão da importância da variável para o modelo. Foi utilizado o gráfico de resumo para exibição dos resultados (explicado no capítulo V, seção V.4.2), método *KernelExplainer*[6] (aplica regressão linear ponderada especial para calcular a importância de cada recurso) e *k-means* com três conjuntos para sumarizar os dados e otimizar a execução. Para os algoritmos baseados em árvore (*AdaBoost*, Árvore de Decisão, *Extra Trees*, *Gradienet Boosting*, *Random Forest* e XGBoost) e modelos lineares (*Lasso*, *Ridge*, Regressão Logística e SGD) foram utilizadas as implementações mais otimizadas *TreeExplainer*[7] e *LinearExplainer*[8], respectivamente.

## IV.7 Explicação amostra com LIME

A última etapa da metodologia utiliza a melhor combinação de abordagens, aplicando explicações locais para as amostras com o método LIME, e avalia os motivos de uma licitação ter sido rotulada como suspeita de conluio. A melhor combinação de abordagens foi com a inclusão de variáveis estatísticas, método de balanceamento Smote e algoritmo de aprendizado K-NN, vide capítulo V.

## IV.8 Utilização da metodologia

A implementação de todos os passos descritos nessa metodologia está no *script python* disponibilizado em repositório na *internet*[9]. As etapas de pré-processamento e construção do conjunto de dados, para serem reutilizadas, dependem que o novo conjunto de dados tenha as mesmas características (tipo de dados e nome dos atributos) do utilizado neste trabalho, pois os tratamentos são específicos pra cada atributo. O mesmo serve para a metodologia de rotulação dos conluios, que dependem de informações específicas para rotulação: dados de licitações com ano, vencedor e participantes; dados de condenações por conluio com participantes e ano. Logo, a reutilização dessas etapas depende de adaptações para o conjunto de dados a ser trabalhado. No entanto, os mesmos passos podem ser reproduzidos como referência em trabalhos similares.

As etapas finais da mineração de dados, incluindo a seleção de atributos, balanceamento de dados, aprendizado de máquina, avaliação de métricas e explicação dos modelos, podem ser mais facilmente reutilizadas, pois as execuções desses métodos são agnósticas quanto ao conjunto de

[6]https://shap.readthedocs.io/en/latest/generated/shap.KernelExplainer.html
[7]https://shap.readthedocs.io/en/latest/generated/shap.TreeExplainer.html
[8]https://shap.readthedocs.io/en/latest/generated/shap.LinearExplainer.html
[9]Repositório: https://github.com/lucasdaflon/conluio-licitacoes/tree/main

dados utilizado. No entanto, o *script* teria que ser adaptado para carregar outro nome de arquivo e o único atributo que precisaria ser renomeado, ou alterado a referência no *script*, é a variável alvo (Licitação conluio?).

# Capítulo V Experimentos e resultados

Este capítulo apresenta os resultados dos experimentos. Os experimentos foram executados em uma máquina *Windows 11 Home*, processador *11th Gen Intel(R) Core(TM) i7-1165G7*, 16 GB de RAM, em um ambiente *jupyter notebook 6.4.8*[1], *Python 3.9.12*, com as bibliotecas *pandas* 1.4.4[2], *numpy* 1.21.5[3], *matplotlib* 3.5.1[4], *scipy.stats* 1.14.1 [5], *category_encoders* 2.6.4[6], *imblearn* 0.12.4[7], *scikit-learn* 1.0.2[8], *shap* 0.42.1[9] e *lime* 0.2.0.1[10]. Os dados e código utilizados estão disponíveis em repositório na *internet*[11].

## V.1 Coleta dos dados

### V.1.1 Dados de licitação

Os totais de registros coletados, para dados de licitações públicas do portal de transparência do CGU, são apresentados na Figura V.1. Como foi utilizada a estratégia de junção interna para unir os dados em uma única tabela, apenas Licitações que tenham registro de Itens e Participantes são mantidas, por este motivo, os totais de Junção são inferiores ao de Licitações.

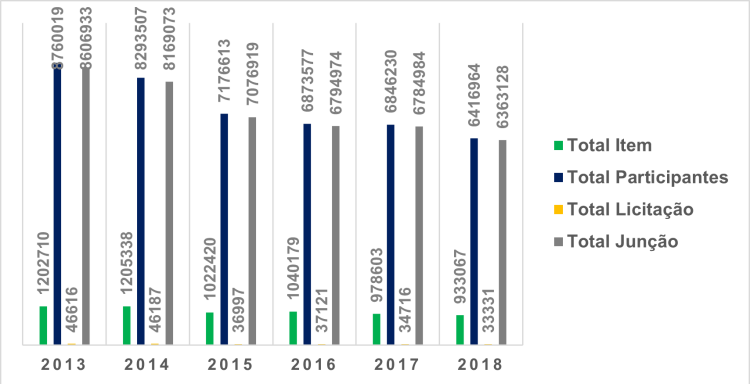



Figura V.1: Totais dos registros de cada tabela por ano.

[1]https://jupyter.org/
[2]https://pandas.pydata.org/
[3]https://numpy.org/
[4]https://matplotlib.org/
[5]https://docs.scipy.org/doc/scipy/index.html
[6]https://contrib.scikit-learn.org/category_encoders/index.html
[7]https://imbalanced-learn.org/stable/
[8]https://scikit-learn.org/stable/
[9]https://shap.readthedocs.io/en/latest/
[10]https://lime-ml.readthedocs.io/en/latest/
[11]Repositório: https://github.com/lucasdaflon/conluio-licitacoes/tree/main

### V.1.2 Dados de conluio

Foram levantados vinte e dois (22) processos administrativos de condenação de conluios vigentes entre 1999 e 2018. No entanto, como só foi possível obter dados de licitações a partir de 2013, aplicando essa restrição, restaram nove processos administrativos, conforme apresentado na Tabela V.1. Em seguida foi feita compatibilização dos nomes dos condenados por conluio com o Participante dos dados de licitação (alguns não foram possíveis compatibilizar), resultando em um arquivo com 80 registros e os seguintes atributos: Processo administrativo (CADE); Período do conluio; Assunto; Condenados; Participante (mesmo dos dados de licitação).

Tabela V.1: Levantamento dos processos de condenação do CADE, a partir de 2013.

| Processo administrativo (CADE) | Período | Assunto |
|---|---|---|
| 08700.003390/2016-60 | 2006-2013 | Tubos e conexões de PVC |
| 08700.003396/2016-37 | 2004-2015 | Tubos e conexões |
| 08700.004095/2020-15 | 2014-2018 | Produtos e serviços de inteligência de negócios (*business intelligence*) |
| 08700.004248/2019-82 | 2011-2014 | Aluguel de equipamentos e veículos |
| 08700.004455/2016-94 | 1999-2014 | Material escolar |
| 08700.004617/2013-41 | 1999-2013 | Licitação de trens e metrôs |
| 08700.005789/2015-02 | 2008-2014 | Aquisição de sacos de lixo |
| 08700.007278/2015-17 | 2014 | Licitações da infraero |
| 08700.003699/2017-31 | 2004-2015 | Aquisição de órteses, próteses e materiais especiais |

## V.2 Pré-processamento

### V.2.1 Avaliações sobre conjunto de dados

A partir da análise sobre valores únicos, faltantes e baixa variabilidade, como mostra a Tabela V.2, decidiu-se eliminar os atributos Número processo, Objeto, Data abertura e Número Licitação:

- Número processo e Objeto possuem muitos valores únicos e são difíceis de serem generalizados. Além disso, Item Compra possui informações similares ao campo Objeto;
- A Data abertura possui muitos valores faltantes e há outro campo de data com nenhum valor faltante (Data Resultado Compra);
- Número Licitação é um identificador que agrega vários itens de licitação e possui baixa variabilidade.

Ao final das etapas de limpeza e eliminação de atributos duplicados ou com muitos valores faltantes, restaram 39.569.515 registros e 12 atributos (Número Licitação, Gestora, Modalidade, Item Compra, Participante, Vencedor, Situação, UF, Qtd. Participantes Item, Mês Compra, Ano Compra, Valor Item Unit.) no conjunto de dados. Na etapa de rotulação, foram classificados 4.531 registros como Sim e 39.564.984 como Não. Ao eliminar as linhas que não possuem itens citados

Tabela V.2: Visão geral dos atributos para o conjunto de dados. Os atributos são avaliados sob três variáveis: quantidade de valores únicos; quantidade de valores vazios; variância. Total de registros: **43.796.011**

| | Número Licitação | Gestora | Modalidade | Número Processo | Órgão |
|---|---|---|---|---|---|
| único | 8.937 | 2.616 | 9 | **205.191** | 266 |
| nulo | 0 | 0 | 0 | **688** | 0 |
| var | **0,000481** | 0,098938 | 0,002088 | 0,081438 | 0,087705 |
| | Item Compra | Participante | Flag Vencedor | Descrição | Quantidade Item |
| único | 14.247 | 85.335 | 2 | 14.247 | 38.197 |
| nulo | 0 | 0 | 0 | 0 | **413** |
| var | 0,088109 | 0,078262 | 0,124067 | 0,088109 | **0,000001** |
| | Valor Item | Vencedor | Objeto | Situação Licitação | Nome Órgão Superior |
| único | **732.815** | 54.724 | **202.058** | 22 | 61 |
| nulo | 0 | 0 | **450** | 0 | 0 |
| var | 0,083677 | 0,07761 | 0,087536 | 0,044829 | 0,0092 |
| | UF | Município | Data Resultado Compra | Data Abertura | Valor Licitação |
| único | 28 | 533 | 1.749 | 1.588 | **124.183** |
| nulo | 0 | 0 | 0 | **22.610** | 0 |
| var | 0,067684 | 0,081406 | 0,083411 | 0,083144 | 0,093716 |

em licitações rotuladas como conluio e remover as duplicatas, o conjunto de dados passou a ter 2.095.023, sendo 4.531 com o rótulo Sim e 2.090.492 com o rótulo Não para conluio.

### V.2.2 Balanceamento e seleção de atributos

Após a divisão do conjunto de treinamento e teste (*holdout*, 80% treinamento e 20% para teste), o conjunto de dados treinado possui 1.672.356 registros, sendo 1.668.747 rotulados como Não e 3.609 como Sim para conluio. Já o conjunto de teste possui 418.090 registros, sendo 417.188 rotulados como Não e 902 como Sim para conluio. Conforme citada na Capítulo II, considera-se desempenhos do MCC acima de 0,5 como fortes.

Para selecionar as melhores abordagens de balanceamento e seleção de atributos foram executados experimentos combinando: com e sem as variáveis estatísticas; sem estratégia de balanceamento e com as estratégias RUS, ROS, Smote e Smoteenn; métodos de seleção de atributos chi2, seleção univariada, *mutual information* e RFE, como mostra a Tabela V.3. Os experimentos foram executados com os algoritmos MLP, Regressão Logística (RL) e *Random Forest* (RF). A escolha desses algoritmos foi para comparar o comportamento de diferentes tipos de algoritmos com as combinações avaliadas (algoritmos lineares, de árvore e de redes neurais).

O método chi2 é o que apresentou resultados superiores mais vezes, como destacado na Tabela V.3. Nos experimentos com variáveis estatísticas, este método foi superior oito vezes, RFE, três vezes e seleção univariada, quatro vezes. Já nos experimentos sem variáveis estatísticas, o chi2 foi superior em sete vezes, RFE, cinco vezes e seleção univariada, três vezes. Os atributos selecionados por cada abordagem de seleção de atributos estão ilustrados na Figura V.2. O método chi2 selecionou os atributos Gestora, Participante, Vencedor, Item compra e Ano Compra. Para o conjunto de dados

Tabela V.3: Abordagens de correção de prevalência sem e com variáveis estatísticas. Resultados com execução dos algoritmos MLP, Regressão Logística (RL) e *Random Forest*. Os resultados em negrito são referentes ao melhor desempenho de cada algoritmo, por estratégia de balanceamento.

| Experimentos com variáveis estatísticas | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Chi2 | | | RFE | | | Seleção Univariada | | | Mutual Information | | |
| | MLP | RL | RF | MLP | RL | RF | MLP | RL | RF | MLP | RL | RF |
| Sem | **0,9622** | 0,7585 | 0,4382 | 0,9501 | **0,771** | 0,1051 | 0,8924 | 0,6081 | **0,5739** | 0,7518 | 0,3593 | 0,047 |
| RUS | **0,6301** | 0,5393 | **0,4553** | 0,6467 | **0,5345** | 0,2052 | 0,5793 | 0,5248 | 0,4468 | 0,3928 | 0,2237 | 0,3172 |
| ROS | **0,928** | **0,7764** | 0,4105 | 0,9141 | 0,7647 | 0,4253 | 0,7637 | 0,6123 | **0,4889** | 0,6758 | 0,3652 | 0,1409 |
| Smote | **0,932** | 0,7767 | 0,4084 | 0,9183 | **0,7782** | 0,2042 | 0,7953 | 0,61 | **0,4837** | 0,6497 | 0,3678 | 0,0332 |
| Smoteenn | **0,9289** | **0,7788** | 0,408 | 0,9166 | 0,7782 | 0,1348 | 0,7817 | 0,6116 | **0,4842** | 0,6795 | 0,3679 | 0,0576 |
| Experimentos sem variáveis estatísticas | | | | | | | | | | | | |
| | Chi2 | | | RFE | | | Seleção Univariada | | | Mutual Information | | |
| | MLP | RL | RF | MLP | RL | RF | MLP | RL | RF | MLP | RL | RF |
| Sem | **0,8943** | **0,7008** | 0,3225 | 0,8584 | 0,7003 | 0,2023 | 0,5241 | 0,5515 | 0,3821 | 0 | 0,0758 | 0 |
| RUS | 0,6099 | 0,5367 | 0,6337 | **0,6114** | **0,5564** | 0,1578 | 0,5493 | 0,5202 | 0,6138 | 0,0914 | 0,0688 | 0,0008 |
| ROS | 0,8234 | **0,7063** | 0,3084 | **0,8511** | 0,697 | 0,1749 | 0,5998 | 0,5516 | 0,4704 | 0,1488 | 0,0758 | 0,0094 |
| Smote | 0,8355 | **0,7036** | 0,5047 | **0,8440** | 0,7014 | 0,182 | 0,598 | 0,5515 | 0,5209 | 0,147 | 0,0762 | 0 |
| Smoteenn | 0,8265 | **0,7071** | 0,4989 | **0,8560** | 0,7014 | 0,2306 | 0,5953 | 0,5516 | 0,4695 | 0,1572 | 0,0762 | 0 |

nos experimentos com variáveis estatísticas são adicionadas as variáveis Coeficiente de Variação (CV), *Kurtosis* ou Curtose (KURT), *Skewness* ou Assimetria (SKEW), *Outlier* e Qtd. participantes conluio.

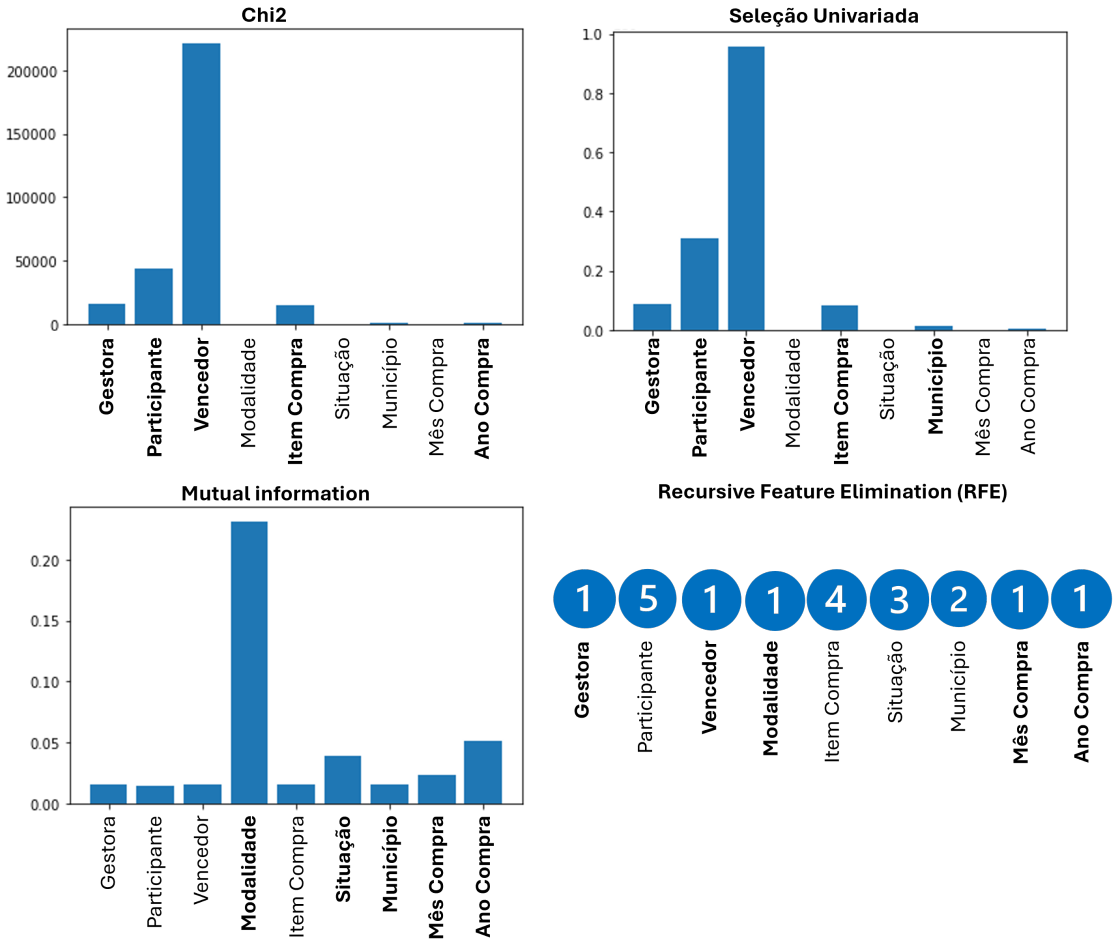



Figura V.2: Importância de cada atributo pelas diversas abordagens de seleção de atributos.

Ao considerar os resultados com chi2, nota-se que para MLP, a melhor abordagem é não balancear os dados, para Regressão Logística é o Smoteenn e para *Random Forest* o RUS. No entanto, o RUS possui desempenho muito inferior para os outros casos. Por outro lado, o Smote, apesar de não ter sido o melhor para nenhum algoritmo, apresenta resultados próximos aos melhores desempenhos, e as diferenças de desempenho entre os experimentos com e sem variáveis estatísticas é bem balanceado (não indica uma majoração para uma estratégia). Logo, os experimentos com todos os algoritmos foram executados para todas as três abordagens (sem estratégia, Smoteenn e Smote).

### V.3 Avaliação do desempenho dos algoritmos

Foram realizados experimentos: com, sem e somente variáveis estatísticas; sem estratégia de balanceamente e com as estratégias de balanceamento Smote e Smoteenn; para todos os algoritmos apresentados no capítulo IV. O treinamento do OneClassSVM utiliza apenas as classes consideradas 'normais'. Este algoritmo classifica como anômalas observações fora do comportamento aprendido como normal. Para tal, foram feitos dois experimentos, um considerando como 'normal' as observações classificadas como Não para conluio e outro experimento considerando como 'normal' as observações classificadas como Sim.

Para efeito de comparação, foi treinado o classificador fictício *DummyClassifier*. Este classificador possui quatro estratégias possíveis, o significado e os resultados do MCC para cada uma delas foi o seguinte: Uniforme, gera previsões randômicas uniformemente, desempenho -0.00108; Estratificada, gera previsões respeitando a proporção de distribuição da classe alvo (conluio), desempenho -0,00047; Constante, sempre prevê um rótulo constante que é definido (neste caso Sim para Conluio), desempenho 0,0; Mais frequente, sempre prevê o rótulo mais frequente (neste caso Não para Conluio), desempenho 0,0.

Os resultados dos experimentos estão apresentados na Tabela V.4. Os algoritmos *AdaBoost*, Árvore de decisão, *Gradient Boosting*, *Lasso*, *Naive Bayes (Bernoulli)*, *Naive Bayes (Gaussian)*, QDA e *OneClassSVM* não obtiveram desempenho superior a 0,5 (considerado bom) para nenhum dos experimentos. *Random Forest* teve desempenho superior a 0,5 apenas no experimento com Smoteenn sem variáveis estatísticas e *Ridge* apenas nos experimentos com Smote e Smoteenn sem variáveis estatísticas.

Os resultados *OneClassSVM* indicam que a melhor estratégia de treinamento é com os dados classificados como Sim para conluio. A metodologia utilizada para rotulação dos conluios dá maior certeza sobre a classificação Sim do que Não. Ou seja, não é possível afirmar com certeza que as observações rotuladas como Não não tenham influência de conluio. Portanto, avalia-se que isso possa ter influenciado mais negativamente o aprendizado do modelo que considera como 'normais' os dados rotulados para conluio como Não do que Sim.

Tabela V.4: Comparação dos resultados com e sem variáveis estatísticas dos algoritmos Árvore de Decisão; *Extra Trees*; *Gradient Boosting*; K-NN; *Lasso*; Regressão Logística; *Naive Bayes (Bernoulli)*; *Naive Bayes (Gaussian)*; MLP; QDA; *Random Forest*; *Ridge*; SGD; SVM; OneClassSVM; e XGBoost. Os melhores resultados estão destacados em negrito. As estratégias de balanceamento utilizadas são nenhuma, Smote e Smoteenn.

| | Com variáveis estatísticas | | | Sem variáveis estatísticas | | | Somente variáveis estatísticas | | |
|---|---|---|---|---|---|---|---|---|---|
| | Sem | Smote | Smoteenn | Sem | Smote | Smoteenn | Sem | Smote | Smoteenn |
| AdaBoost | -0,0275 | -0,0275 | -0,0275 | 0,1116 | 0,3380 | 0,3642 | -0,0275 | -0,0275 | -0,0275 |
| Árvore de decisão | 0,2443 | 0,2449 | 0,2449 | 0,3200 | 0,3205 | 0,3212 | 0,2477 | 0,2477 | 0,2477 |
| **Extra Trees** | 0,6315 | **0,6978** | 0,6634 | 0,4500 | 0,6348 | 0,6688 | 0,1919 | 0,1911 | 0,1922 |
| Gradient Boosting | 0,0016 | 0,4103 | 0,4154 | 0,1787 | 0,4995 | 0,4829 | 0,0197 | 0,1832 | -0,0275 |
| **K-NN** | 0,9733 | **0,9756** | 0,9664 | 0,9294 | 0,9350 | 0,9297 | 0,9392 | 0,5491 | 0,5491 |
| Lasso | 0 | 0,0870 | 0,0870 | 0 | 0 | 0 | 0 | 0,0870 | 0,0870 |
| **Regressão logística** | 0,7585 | **0,8209** | 0,7788 | 0,7008 | 0,7016 | 0,7071 | 0,3096 | 0,3161 | 0,3139 |
| Naive Bayes (Bernoulli) | 0,1887 | 0,3563 | 0,2005 | 0,4668 | 0,2983 | 0,4701 | -0,0249 | -0,0249 | -0,0249 |
| Naive Bayes (Gaussian) | 0,4152 | 0,2005 | 0,3559 | 0,3365 | 0,4701 | 0,2974 | 0,3312 | 0,3110 | 0,3110 |
| **MLP** | **0,9618** | 0,9284 | 0,9262 | 0,8817 | 0,8372 | 0,8434 | 0,4687 | 0,3596 | 0,3625 |
| QDA | 0,4549 | 0,4026 | 0,4028 | 0,3399 | 0,3085 | 0,3084 | 0,3308 | 0,3134 | 0,3134 |
| Random Forest | 0,4580 | 0,2519 | 0,3865 | 0,3195 | 0,3222 | 0,508 | 0,2976 | 0,2774 | 0,2477 |
| **Ridge** | 0,3055 | 0,2841 | 0,2842 | 0,3055 | **0,6425** | 0,6425 | 0 | 0,1642 | 0,1642 |
| **SGD** | **0,6440** | 0,6287 | 0,6291 | 0,5524 | 0,5524 | 0,5524 | 0,2581 | 0,2620 | 0,2620 |
| **SVM** | 0,9078 | 0,9111 | **0,9115** | 0,8019 | 0,8021 | 0,8021 | 0,3210 | 0,2620 | 0,3195 |
| OneClassSVM (treino sem conluio) | 0,1339 | 0,1339 | 0,1338 | 0,0619 | 0,0619 | 0,0619 | 0,0948 | 0,0948 | 0,0948 |
| OneClassSVM (treino com conluio) | 0,2809 | 0,2770 | 0,2778 | 0,3867 | 0,3889 | 0,3898 | 0,1577 | 0,1880 | 0,1880 |
| **XGBoost** | 0,8445 | 0,7110 | **0,9079** | 0,5490 | 0,4659 | 0,4606 | 0,1764 | 0,1919 | 0,1919 |
| *Dummy* | 0,0000 | 0,0000 | 0,0000 | 0,0000 | 0,0000 | 0,0000 | 0,0000 | 0,0000 | 0,0000 |

O melhor desempenho foi obtido pelo K-NN, utilizando Smote e variáveis estatísticas. A ordem de desempenho dos três melhores algoritmos é a mesma, independente da combinação de estratégias de balanceamento e inclusão das variáveis estatísticas: (01) K-NN; (02) MLP; (03) SVM. Os outros algoritmos que merecem destaque quanto ao desempenho (superior a 0,5), mas a ordem varia conforme as combinações são: (04) Regressão Logística; (05) XGBoost; (06) *Extra Trees*; (07) *Ridge*; (08) SGD.

Dos oito algoritmos destacados, os únicos que tiveram desempenho superior sem as variáveis estatísticas foram o *Ridge* (0,36 superior com Smote e Smoteenn) e *Extra Trees* (0,01 superior com Smoteenn). Os maiores incrementos de desempenho com a inclusão das variáveis estatísticas foram obtidos com XGBoost: Smoteenn (0,45); nenhuma (0,30); Smote (0,25). O aumento do desempenho com a inclusão das variáveis estatísticas, utilizando Smote, varia de aproximadamente 0,04 (K-NN) à 0,25 (XGBoost).

## V.4 Importância de atributos

Os experimentos de importância de atributos e SHAP foram executados apenas para os modelos de aprendizado de máquina que obtiveram desempenho superiores a 0,5, segunda a métrica MCC, utilizando o Smote: (01) K-NN; (02) MLP; (03) SVM; (04) Regressão Logística; (05) XGBoost;

(06) *Extra Trees*; (07) *Ridge*; (08) SGD. O Smote foi quem obteve o melhor desempenho geral entre os algoritmos e estratégias de balanceamento com o K-NN e inclusão de variáveis estatísticas. Por isso, foi a estratégia selecionada para compor o *pipeline* da metodologia.

### V.4.1 Experimentos com importância de atributos

A Figura V.3 mostra os resultados das execuções dos métodos de importância de atributos. Estes resultados foram sumarizados na Tabela V.5, com a ordem de importância dos atributos para cada algoritmo de aprendizado. Verifica-se que os atributos Vencedor e *Skewness* ou Assimetria (SKEW) foram o de maior contribuição (*top 1*) para ao menos um algoritmo. Já os atributos que tiveram entre os três de maior contribuição (*top 3*) foram: (01) Vencedor; (02) SKEW; (03) Coeficiente de Variação (CV); (04) *Kurtosis* ou Curtose (KURT); (05) Ano compra; (06) *Outlier*.

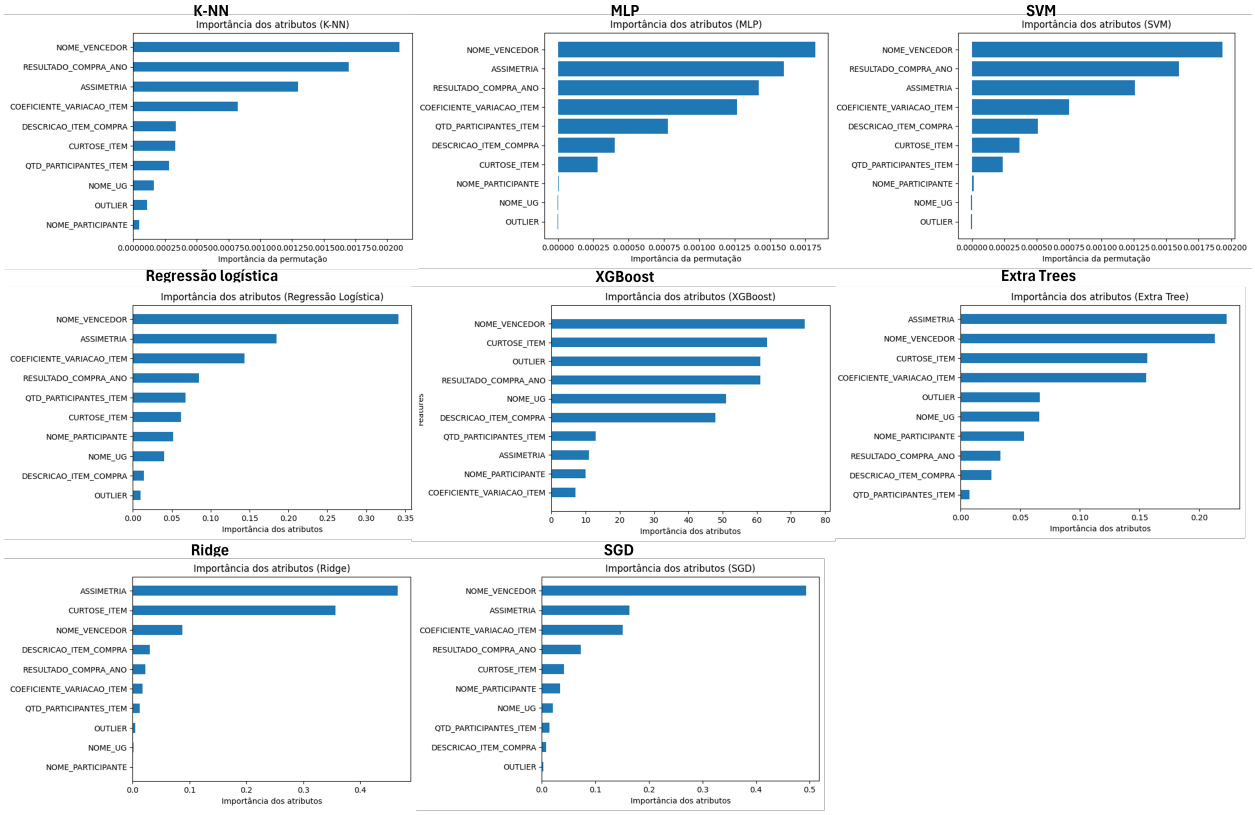



Figura V.3: Execução do método de seleção de importância para os algoritmos (01) K-NN; (02) MLP; (03) SVM; (04) Regressão Logística; (05) XGBoost; (06) *Extra Trees*; (07) *Ridge*; (08) SGD.

Os tempos de execução para os cálculos de importância de atributos para os modelos de aprendizado estão apresentados na Tabela V.6. Os valores descritos na tabela foram arredondados. A Tabela V.6 mostra que os métodos de cálculo de importância por permutação são mais lentos que cálculos de importância intrínsecos dos modelos. A demora do tempo do SVM é causada pelo tempo de treinamento do modelo e não pelo cálculo de importância, enquanto o K-NN é causado pelos treinamentos e pelo cálculo de importância.

Tabela V.5: Ordem de importância de cada atributo, segundo método de importância de atributos, para os algoritmos (01) K-NN; (02) MLP; (03) SVM; (04) Regressão Logística; (05) XGBoost; (06) *Extra Trees*; (07) *Ridge*; (08) SGD. Os melhores resultados estão destacados em negrito.

| | K-NN | MLP | SVM | Regressão Logística | XGBoost | Extra Trees | Ridge | SGD |
|---|---|---|---|---|---|---|---|---|
| Gestora | 8 | 9 | 9 | 8 | 5 | 6 | 9 | 7 |
| Participante | 10 | 8 | 8 | 7 | 9 | 7 | 10 | 6 |
| Vencedor | **1** | **1** | **1** | **1** | **1** | 2 | 3 | **1** |
| Item Compra | 5 | 6 | 5 | 9 | 6 | 9 | 4 | 9 |
| Ano Compra | 2 | 3 | 2 | 4 | 4 | 8 | 5 | 4 |
| Assimetria | 3 | 2 | 3 | 2 | 8 | **1** | **1** | 2 |
| Coeficiente de variação | 4 | 4 | 4 | 3 | 10 | 4 | 6 | 3 |
| Curtose | 6 | 7 | 6 | 6 | 2 | 3 | 2 | 5 |
| *Outlier* | 9 | 10 | 10 | 10 | 3 | 5 | 8 | 10 |
| Qtd. participantes | 7 | 5 | 7 | 5 | 7 | 10 | 7 | 8 |

Tabela V.6: Tempos de execução dos métodos de importância de atributos, para os algoritmos (01) K-NN; (02) MLP; (03) SVM; (04) Regressão Logística; (05) XGBoost; (06) *Extra Trees*; (07) *Ridge*; (08) SGD. Os maiores tempos estão destacados em negrito.

| | Tempo | Abordagem |
|---|---|---|
| K-NN | **6 horas e 18 minutos** | Importância dos atributos por permutação |
| MLP | 4 minutos | Importância dos atributos por permutação |
| SVM | **2 horas e 35 minutos** | Importância dos atributos por permutação |
| Regressão Logística | 0,14 segundos | Importância dos atributos |
| XGBoost | 0,13 segundos | Importância dos atributos |
| Extra Trees | 0,13 segundos | Importância dos atributos |
| Ridge | 0,06 segundos | Importância dos atributos |
| SGD | 0,08 segundos | Importância dos atributos |

Na comparação dos ganhos de desempenhos dos algoritmos com os resultados de importâncias dos atributos, obtêm-se as indicações dos atributos a seguir como os mais relevantes. A presença de variáveis estatísticas ocorre independente do resultado ter piorado, melhorado pouco ou muito:

- *Ridge*: piorou o desempenho com o acréscimo das variáveis estatísticas. Indicou as variáveis SKEW, KURT e Vencedor;
- K-NN: aprimorou o desempenho em 0,04. Indicou as variáveis Vencedor, Ano compra e SKEW;
- *Extra Trees*, aprimorou o desempenho em 0,06. Indicou as variáveis SKEW, Vencedor e KURT;
- SGD: aprimorou o desempenho em 0,08. Indicou as variáveis Vencedor, SKEW e CV;
- MLP: aprimorou o desempenho em 0,09. Indicou as variáveis Vencedor, SKEW e Ano compra;
- SVM: aprimorou o desempenho em 0,11. Indicou as variáveis Vencedor, Ano compra e SKEW;
- Regressão Logística: aprimorou o desempenho em 0,12. Indicou as variáveis Vencedor, SKEW e CV;
- XGBoost: aprimorou o desempenho em 0,25. Indicou as variáveis Vencedor, KURT e *Outlier*.

### V.4.2 Experimentos com SHAP

A Figura V.4 mostra a execução do SHAP, ilustrado pelo gráfico de resumo. Neste gráfico, para cada atributo são impressas as observações do conjunto de dados como pontos. Maiores concentrações de pontos são mostradas com estes pontos crescendo na vertical. O eixo vertical lista as variáveis do modelo por ordem de importância. O eixo horizontal são os valores. Valores positivos indicam suporte para atributo alvo (é conluio), e o negativo para o oposto (não é conluio). Já as cores representam o valor da variável; quanto mais vermelho, maior o valor. Ter essas cores em lados opostos indica que a variável é uma boa preditora, pois só em mudar o valor o modelo já consegue verificar a contribuição para a classe. O intervalo de alcance também é importante (mais à direita, maior a contribuição para o resultado) e variáveis com maior dispersão no eixo tendem a ser mais importantes.

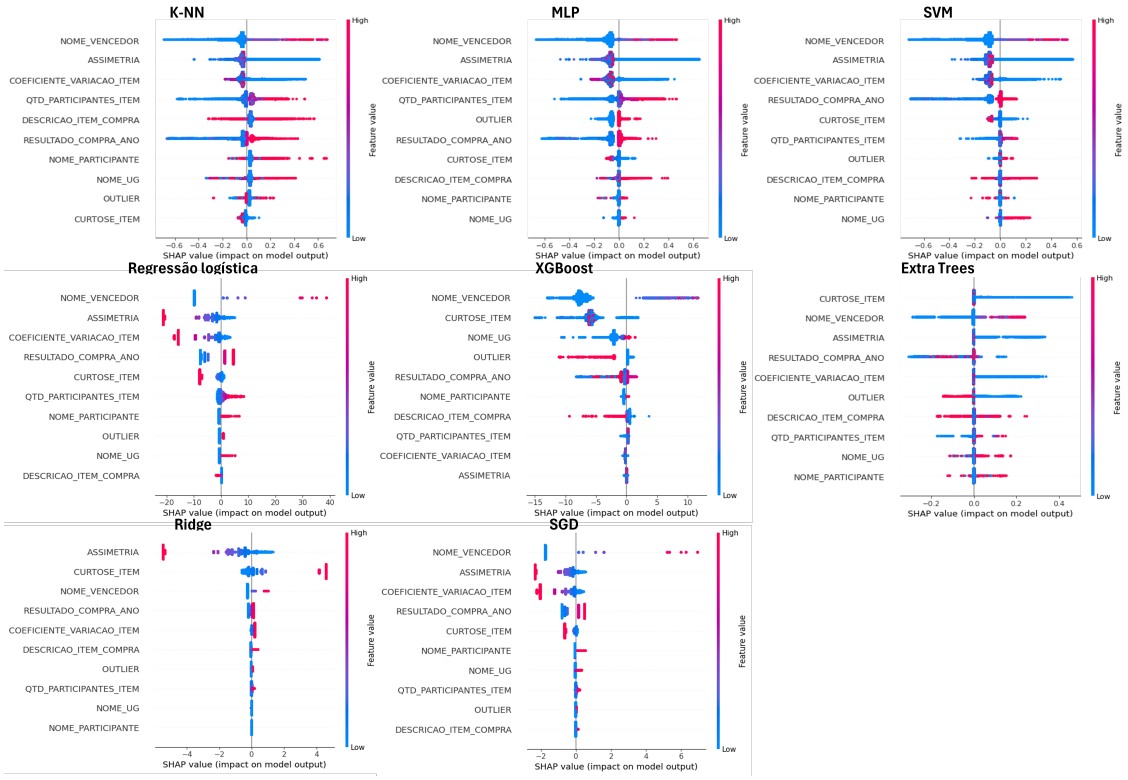



Figura V.4: Execução do método do SHAP para os algoritmos (01) K-NN; (02) MLP; (03) SVM; (04) Regressão Logística; (05) XGBoost; (06) *Extra Trees*; (07) *Ridge*; (08) SGD.

O resultado foi sumarizado, na Tabela V.7, com a ordem de importância dos atributos para cada algoritmo de aprendizado. Verifica-se que os atributos Vencedor, *Skewness* ou Assimetria (SKEW) e *Kurtosis* ou Curtose (KURT) foram a de maior contribuição (*top 1*) para ao menos um algoritmo. Já os atributos que tiveram entre os três de maior contribuição (*top 3*) foram: (01) Vencedor; (02) SKEW; (03) Coeficiente de Variação (CV); (04) KURT; (05) Gestora.

Os tempos de execução do SHAP para os modelos de aprendizado estão apresentados na Tabela V.8. Os valores descritos na tabela foram arredondados. Os tempos dos algoritmos otimizados

Tabela V.7: Ordem de importância de cada atributo, segundo SHAP, para os algoritmos (01) K-NN; (02) MLP; (03) SVM; (04) Regressão Logística; (05) XGBoost; (06) *Extra Trees*; (07) *Ridge*; (08) SGD. Os melhores resultados estão destacados em negrito.

| | K-NN | MLP | SVM | RL | XGBoost | ExtraTrees | Ridge | SGD |
|---|---|---|---|---|---|---|---|---|
| Gestora | 8 | 10 | 10 | 9 | 3 | 9 | 9 | 7 |
| Participante | 7 | 9 | 9 | 7 | 6 | 10 | 10 | 6 |
| Vencedor | **1** | **1** | **1** | **1** | **1** | 2 | 3 | **1** |
| Item Compra | 5 | 8 | 8 | 10 | 7 | 7 | 6 | 10 |
| Ano Compra | 6 | 6 | 4 | 4 | 5 | 4 | 4 | 4 |
| Assimetria | 2 | 2 | 2 | 2 | 10 | 3 | **1** | 2 |
| Coeficiente de variação | 3 | 3 | 3 | 3 | 9 | 5 | 5 | 3 |
| Curtose | 10 | 7 | 5 | 5 | 2 | **1** | 2 | 5 |
| *Outlier* | 9 | 5 | 7 | 8 | 4 | 6 | 7 | 9 |
| Qtd. participantes | 4 | 4 | 6 | 6 | 8 | 8 | 8 | 8 |

para árvore (*TreeExplainer*) e modelos lineares (*LinearExplainer*) são muito inferiores ao tempo do algoritmo genérico (*KernelsExplainer*).

Tabela V.8: Tempos de execução dos métodos de SHAP para os algoritmos (01) K-NN; (02) MLP; (03) SVM; (04) Regressão Logística; (05) XGBoost; (06) *Extra Trees*; (07) *Ridge*; (08) SGD. Os maiores tempos estão destacados em negrito.

| Algoritmo | Tempo | Abordagem |
|---|---|---|
| K-NN | **13 dias e 18 horas** | *KernelsExplainer* |
| MLP | **1 dia e 2 horas** | *KernelsExplainer* |
| SVM | **4 dias e 2 horas** | *KernelsExplainer* |
| Regressão Logística | 1 h e 3 minutos | *LinearExplainer* |
| XGBoost | 22 segundos | *TreeExplainer* |
| Extra Trees | 9 horas e 56 minutos | *TreeExplainer* |
| Ridge | 17 segundos | *LinearExplainer* |
| SGD | 17 segundos | *LinearExplainer* |

Na comparação dos ganhos de desempenhos dos algoritmos com os resultados das contribuições dos atributos pelo SHAP, obtêm-se as indicações dos atributos a seguir como os mais relevantes. Assim como no cálculo de importância de atributos, a presença de variáveis estatísticas ocorre independente do resultado ter piorado, melhorado pouco ou muito:

- *Ridge*: piorou o desempenho com o acréscimo das variáveis estatísticas. Indicou as variáveis SKEW, KURT e Vencedor. Mesma ordem do cálculo de importância de atributos;
- K-NN: aprimorou o desempenho em 0,04. Indicou as variáveis Vencedor, Ano compra e SKEW;
- *Extra Trees*: aprimorou o desempenho em 0,06. Indicou as variáveis KURT, Vencedor e SKEW. Mesmos atributos do cálculo de importância de atributos, mas na ordem inversa;
- SGD: aprimorou o desempenho em 0,08. Indicou as variáveis Vencedor, SKEW e CV. Mesma ordem do cálculo de importância de atributos;
- MLP: aprimorou o desempenho em 0,09. Indicou as variáveis Vencedor, SKEW e CV. Mesma ordem do cálculo de importância, com exceção ao CV, que o cálculo de importância indicou SKEW em terceiro;

- SVM: aprimorou o desempenho em 0,11. Indicou as variáveis Vencedor, SKEW e CV. Diferente do cálculo de importância de atributos, que indicou compra ano em segundo e SKEW em terceiro;

- Regressão Logística: aprimorou o desempenho em 0,12. Indicou as variáveis Vencedor, SKEW e CV. Mesma ordem do cálculo de importância de atributos;

- XGBoost: aprimorou o desempenho em 0,25. Indicou as variáveis Vencedor, KURT e Gestora. Mesma ordem do cálculo de importância de atributos, com exceção de gestora, para o cálculo de importância o terceiro seria *Outlier*.

## V.5 Experimentos com LIME

Para os experimentos com LIME ilustrados na Figura V.5, o método foi executado uma vez para cada combinação da matriz confusão, vide Tabela V.9, assim como na execução dos métodos de importância de atributos e SHAP, das seções V.4.1 e V.4.2, respectivamente. O LIME confirma a relevância dos atributos *Skewness* ou Assimetria (SKEW) e Coeficiente de Variação (CV) nas duas observações preditas corretamente pelo modelo (1099 e 1).

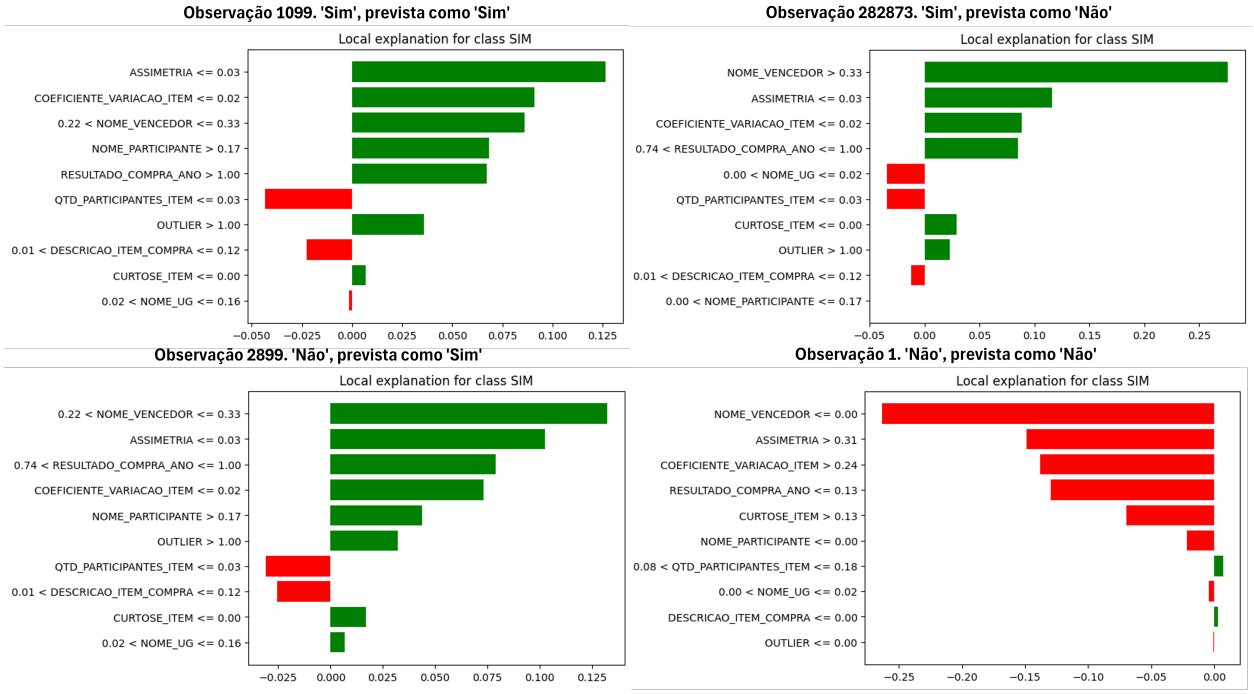



Figura V.5: Execução do método do LIME para quatro observações: classificada como 'Sim' para conluio e o modelo previu 'Sim'; classificada como 'Sim', mas o modelo previu 'Não; classificada como 'Não', mas o modelo previu 'Sim'; classificada como 'Não' e o modelo previu 'Não'.

Conforme explicado no capítulo II, o LIME ajusta um modelo simples e interpretável, tal como uma regressão linear, em torno da previsão de interesse. De forma geral, os gráficos gerados pelo LIME descrevem o quanto um atributo contribuiu para o modelo de aprendizado, por estar dentro de

Tabela V.9: Matriz de confusão da execução do K-NN com método de balanceamento Smote e variáveis estatísticas.

| | | **Previsto** | |
|---|---|---|---|
| | | **Positivo** | **Negativo** |
| **Real** | **Positivo** | 897 | 5 |
| | **Negativo** | 40 | 417.148 |

um limite de valores. Estes resultados são interessantes para explicar os limites de decisão do modelo para variáveis numéricas, tais como as variáveis estatísticas. Por outro lado, as interpretações dos atributos categóricos transformados em numéricos, não trazem muito significado, principalmente por não existir uma ordem entre eles.

## V.6 Discussão dos resultados

Uma grande dificuldade e limitação de trabalhar com conjunto de dados para identificação de conluios é não ter como classificar totalmente o conjunto de dados. Em outras palavras, tarefas de classificação de animais ou plantas, por exemplo, são facilmente confirmadas por especialistas que conhecem suas características e podem garantir um conjunto completo e totalmente classificado. Na classificação de conluios é preciso trabalhar com casos já confirmados, mas isso limita a quantidade de casos que se consegue analisar e classificar. Ou seja, a classificação de 1.000 eventos de conluios em um conjunto de dados de 10.000.000, não significa que não existam outros 1.000 ou 100.000 conluios não rotulados. Essa limitação, pode tendenciar o modelo para entender que determinadas situações, que na realidade são conluios, mas foram rotuladas como não sendo, sejam entendidas como não sendo conluios de fato.

Para tentar mitigar o cenário descrito anteriormente, o conjunto de dados construído rotula apenas itens de compra que tenham sido analisados e condenados pelo CADE. Avalia-se que, quando o CADE faz uma condenação sobre cartel de compras de material hospitalar em nível federal, por exemplo, ele avalia o mercado como um todo. Logo, mesmo não sendo possível afirmar que não exista outro cartel, que não tenha sido condenado para compras de material hospitalar. A chance disso ocorrer é menor do que afirmar que não exista cartel na compra de outros itens que não tenham sido avaliados pelo CADE.

Outra dificuldade relacionada a este tema é a caracterização de falsos positivos e mudanças de características e comportamento de preços e participantes. Padrões de preços sofrem influências de eventos do mundo real, problemas climáticos, crises externas e sanitárias, por exemplo, podem provocar aumentos abruptos de preços e redução abrupta de participantes capazes de atender aos processos licitatórios, características muitas vezes identificadas como sinais de alertas de conluios. Quanto à caracterização de participantes, eventualmente empresas envolvidas em fraudes mudam

de nome, natureza jurídica, ramo de atuação ou localidade para burlar sanções ou para tentar desvincular a imagem negativa ocasionada pela participação em tais malfeitos.

Estas limitações e dificuldades poderiam indicar que a atividade de identificação de conluios por avaliações estatísticas ou por um aprendizado de máquina não supervisionado sejam melhores opções. No entanto este trabalho explorou a utilização da adição de variáveis estatísticas e correção de prevalências, como artifícios de melhoramento do modelo de aprendizado e compensação da limitação de dados classificados. Esta abordagem provou ser viável, obtendo êxito no aumento do desempenho (métrica MCC) de 0,04 a 0,25 com a divisão do conjunto de treinamento e teste com 80% e 20%, respectivamente, exceto para o algoritmo de *Ridge*.

A geração e leitura de métricas são importantíssimas para entender se o modelo criado é útil para aplicação com novos dados. No entanto, por melhor que sejam os resultados obtidos, em uma Auditoria, apenas rotular uma licitação como sendo um caso de conluio ou não, pode não ser suficiente. Compreender as características que levaram o modelo a concluir sobre a classificação são importantes informações para direcionar ações de investigação e fundamentação da escolha dos casos que precisam ser melhor apurados.

Na aplicação deste trabalho, os métodos de cálculo de importância de atributos e SHAP foram úteis para compreender quais atributos são mais relevantes para a classificação do modelo. De forma geral, os atributos Vencedor, SKEW, KURT, CV e Gestora, nessa ordem, são os mais relevantes para explicação da maioria dos algoritmos de aprendizagem. Por outro lado, a adição do atributo *Outlier*, apesar de ser o terceiro atributo em contribuição para o XGBoost (no cálculo de importância de atributo), não provou ter relevância para os demais algoritmos.

Os experimentos com os métodos de cálculo de importância de atributos e SHAP demonstraram resultados muito parecidos. Uma das diferenças desses métodos é o formato de visualização. A importância de atributos permite uma avaliação da magnitude da importância dos atributos, enquanto que no SHAP é possível entender o posicionamento e concentração das amostras conforme o valor de SHAP. A leitura do gráfico SHAP exige maior compreensão sobre o significado das cores, pontos e eixos do gráfico.

Outra diferença está no tempo de processamento. A execução genérica do SHAP (*KernelsExplainer*) é muito mais demorada que o cálculo de importância de atributo. Mas as execuções otimizadas (*LinearExplainer* e *TreeExplainer*), apesar de mais demorados que o cálculo de importância de atributos, reduzem significativamente o tempo de execução. Desta forma, se tempo for um fator relevante, a escolha pelo cálculo de importância e métodos otimizados do SHAP são os mais recomendados.

Conforme explicações do capítulo II, os métodos de importância de atributos tipicamente utilizam como base os modelos gerados e cálculos de correlação entre variáveis. Quanto a isso, de-

pendendo de como o modelo foi criado, pode-se dar importância maior para atributos com muita correlação com outros ou que tenham muitas categorias, como acontece nos algoritmos de árvore, que podem priorizar atributos com muitas categorias, devido a maior possibilidade de divisões e pontos de decisão que esses atributos podem fornecer. Já o método de SHAP possui uma robustez teórica, avaliando diversas combinações dos atributos e modelos, o que fornece um resultado mais preciso e com menos chance de viés.

Por fim, os resultados do LIME demonstraram que este pode ser um método útil para explicação dos limites de decisão de variáveis originalmente numéricas, como as variáveis estatísticas. Combinar diferentes abordagens de modelos explicáveis são úteis para obter diferentes percepções sobre os modelos gerados, como utilizar estes resultados para validar o modelo e verificar possíveis vieses morais ou oportunidades de melhorias. Por exemplo, os modelos explicáveis indicaram Vencedor como o atributo mais relevante. Em um cenário que uma maior generalização seja necessária, esta informação pode ser importante para decidir sobre a exclusão deste atributo.

# Capítulo VI Conclusões e trabalhos futuros

A tarefa de identificação de conluios e cartéis é muito importante na busca de uma economia mais competitiva, eficiente e justa. Tais práticas, além de prejudicar fornecedores idôneos, causam prejuízos e sobrepreços nas licitações públicas. A automatização da tarefa de identificação de conluios, oportuniza aos órgãos de investigação e controle maior abrangência sobre as licitações auditadas. Há sinais de alertas e padrões comportamentais que indicam existência de conluios e cartéis. Logo, este é um problema que pode ser abordado com automatização por algoritmos de aprendizado de máquina, e modelado para incluir variáveis estatísticas, que capturem desvios no comportamento considerado normal, do ponto de vista da estatística.

Sobre a ótica dos processos de Auditoria, obter bons desempenhos com a automatização por algoritmos de aprendizado de máquina são importantes, mas não suficientes. Resultados de Auditorias precisam ser baseados em evidências, descritas quais metodologias e cálculos realizados. Logo, a explicabilidade é um recurso importante para indicar possíveis razões para os desvios de comportamento das amostras identificadas como conluio. Destas explicações, os Auditores podem focar suas inspeções nas observações com maior chance de conluio, segundo indicações do modelo de aprendizagem de máquina, e nas características que mais contribuíram para essa decisão do modelo (por exemplo, localidade, órgão, data, variação do preço).

O primeiro objetivo específico era construir um conjunto de dados consistente com os rótulos de quais licitações foram influenciadas por conluios. O conjunto de dados construído mostrou-se útil para o desenvolvimento de modelos de aprendizado de máquina para identificação de conluios em licitações. Uma das contribuições deste trabalho foi a construção do conjunto de dados rotulados de licitações públicas da esfera federal. A metodologia aplicada e os modelos de aprendizado de máquina gerados demonstraram ser úteis para auxiliar na predição de conluios e na explicação dos resultados.

O segundo objetivo específico deste trabalho é avaliar se a inclusão de variáveis estatísticas influencia a melhora no desempenho dos algoritmos de aprendizagem de máquina. A avaliação da métrica MCC, a partir da estratégia de divisão *holdout* 80% e 20% (divisão aleatória do conjunto de dados de treinamento e teste, respectivamente), balanceando os dados de treinamento com Smote, demonstrou aumentos de desempenho entre 0,04 e 0,25, com a inclusão das variáveis estatísticas, para todos os algoritmos, exceto para *Ridge*.

O terceiro objetivo específico é demonstrar que os modelos explicáveis são úteis na descrição das razões de rotulação de um conluio. Os métodos SHAP, importância de atributo e LIME foram úteis para compreender quais atributos são mais relevantes para o modelo de aprendizado gerado. Segundo esses métodos, os atributos Vencedor, SKEW, KURT, CV foram os mais importantes para os modelos gerados. O LIME é útil para explicar, dada uma observação, a contribuição de uma variável para o modelo, dados os limites de decisão de variáveis originalmente numéricas. Estes resultados comprovam que o comportamento estatístico da ação de grupos em conluio é similar, independente do nicho em que ele opera, e pode ser melhor descrito a partir de componentes estatísticos em relação ao preço contratado dos produtos e serviços.

Como contribuições deste trabalho, destacamos a publicação de uma primeira versão dos experimentos no Simpósio Brasileiro de Banco de Dados (SBBD) 2024, vide Scoralick et al. [2024]. Nesta publicação foi apresentada uma primeira versão do conjunto de dados criado. Também foi demonstrado que as variáveis estatísticas contribuíram para o aprimoramento do desempenho dos modelos de aprendizado de máquina e a utilidade dos métodos de explicação para identificar as variáveis de maior contribuição nestes resultados.

A metodologia utilizada para classificação de conluio é outra contribuição relevante. Apesar de suas limitações, os critérios utilizados demonstraram ser úteis para separar o conjunto de dados, o que refletiu no desempenho dos modelos de aprendizado superiores ao classificador *Dummy*. Outra contribuição foi quanto aos experimentos com modelos explicáveis no contexto do problema de detecção de conluio. Não foram identificados na literatura trabalhos de detecção de conluio que utilizam tais técnicas. Este trabalho pode servir de base para outros que desejam evoluir no estudo de melhores métodos de explicação para este tipo de problema. Esse ponto é especialmente relevante ao considerar Projeto de Lei n° 2338, de 2023[1], que irá regular a IA no Brasil. O projeto foi aprovado no Senado Federal e será apreciado pela Câmara dos Deputados. A versão atual cita o direito de solicitar explicações sobre os resultados do modelo de IA.

Diferente de García Rodríguez et al. [2022], que obtiveram os melhores desempenhos com os *ensemble methods* (*Extra Trees, Random Forest e Ada Boost*), este trabalho obteve melhores desempenhos com K-NN, MLP e SVM. Estes algoritmos estão no grupo que tiveram pior desempenho para García Rodríguez et al. [2022], junto com SGD e *Naive Bayes (Bernoulli e Gaussiano)* (estes dois também com desempenho ruim neste trabalho).

García Rodríguez et al. [2022] obtiveram maior incremento médio da acurácia com os *ensemble methods*, quando treinados com variáveis estatísticas, enquanto este aumento não foi observado nos algoritmos de pior desempenho. Já este trabalho obteve melhora no desempenho com a inclusão de variáveis estatísticas em todos algoritmos, exceto para *Ridge*. Cabe lembrar que há diferenças

[1] https://www25.senado.leg.br/web/atividade/materias/-/materia/157233

entre as variáveis estatísticas utilizadas, os dois trabalhos utilizam em comum CV, KURT e SKEW, enquanto que García Rodríguez et al. [2022] utiliza a mais DIFFP, SPD, RD e KS (medidas relativas a diferenças entre primeiro e segundo lance), e este trabalho utiliza a mais quantidade de participantes e *Outlier*. No entanto, García Rodríguez et al. [2022] não listou a importância individual dos atributos, para avaliar quais das variáveis estatísticas tiveram maior contribuição.

### VI.1 Dificuldades

Durante o desenvolvimento do trabalho foram testadas outras fontes de dados, tais como a *Application Interface Programming* (API) de Compras Governamentais[2]. Essa API possui um conjunto de dados e tabelas mais completo que os disponíveis no Portal da Transparência do Governo Federal[3], mas a API apresentava erro nas consultas de muitas das tabelas. Como parte da investigação dos erros foi feita uma primeira consulta ao Ministério da Gestão e da Inovação em Serviços Públicos (MGI) pelo Fala.BR em 04/11/2023[4]. Em resposta, o MGI informou que a API está passando por mudanças e por isso pode apresentar algumas instabilidades e inconsistências. Ademais, afirmou que Portal Nacional de Contratações Públicas (PNCP)[5], divulgado em 09/10/2023, é o endereço eletrônico oficial destinado à divulgação centralizada e obrigatória dos atos exigidos pela Lei nº 14.133/2021.

Em uma avaliação das tabelas e dados disponíveis no PNCP, não foi possível obter os dados dos lances de cada licitação. Dessa forma, uma segunda manifestação foi aberta para o MGI em 05/05/2024[6]. Em resposta, foi informado que as consultas detalhadas de cada licitação, incluindo o valor dos lances, podem ser feitas pelo comprasnet[7], mas esse caminho não permite consultas em lotes. Também informaram uma nova forma de acessar a API de Compras[8], mas não é possível obter dados dos lances das licitações por essa API.

Por fim, uma grande dificuldade está relacionada ao grande volume de dados utilizados nos experimentos. Muitas execuções demoraram semanas para serem concluídas, em especial as execuções do SHAP. Aguardar tempos tão grandes expõe as execuções a intercorrências não planejadas, tais como reinícios forçados da máquina local (atualização do *windows*), interrupção de conexões, quando executados remotamente nos servidores do Cefet/RJ e desligamentos acidentais. Além disso, houve problemas com interrupções do *kernel* do *jupyter notebook*, devido ao uso excessivo de memória. Estas interrupções exigiram diversas refatorações do código, tanto para incluir funções de limpeza de objetos, quanto para alterar lógicas que evitem processamentos com atributos e valores

[2]https://compras.dados.gov.br/docs/home.html
[3]https://portaldatransparencia.gov.br/download-de-dados/licitacoes
[4]Manifestação de número 18800.067950/2023-69 no endereço eletrônico https://falabr.cgu.gov.br/
[5]https://www.gov.br/pncp/pt-br/acesso-a-informacao/manuais/
[6]Manifestação de número 18002.004266/2024-52 no endereço eletrônico https://falabr.cgu.gov.br/
[7]http://comprasnet.gov.br/livre/pregao/ata0.asp
[8]https://dadosabertos.compras.gov.br/swagger-ui/index.html

excessivos (antecipação de remoção de colunas, durante o carregamento dos dados, avaliadas para exclusão na fase de pré-processamento, por exemplo). Esses eventos atrasaram significativamente o término dos experimentos.

## VI.2 Limitações

Durante o desenvolvimento deste trabalho, foram feitas diversas escolhas, por exemplo, sobre abordagens, artigos relacionados e conjuntos de dados. Estas escolhas permitiram limitar um escopo para estudo mais aprofundado, mas deixam dúvidas se teríamos um melhor resultado ao escolher outros caminhos.

A primeira limitação é sobre a pesquisa bibliográfica. Os termos da pesquisa foram selecionados conforme o objetivo do estudo. Para obter o conjunto mais amplo possível de artigos, foram pesquisados termos diferentes, utilizados na literatura, mas que possuem o mesmo significado. Exemplos disso são as "variáveis estatísticas" e "telas estatísticas". No entanto, pode haver outros termos não capturados, utilizados por artigos de interesse. O uso de uma única base de dados para pesquisa (Scopus) é outra limitação da pesquisa bibliográfica. Essa escolha foi devida à necessidade de utilizar uma base de dados que dê conforto sobre a qualidade dos artigos, mas pode haver artigos de interesse em outras bases que não são indexados pela Scopus.

Sobre o conjunto de dados, cita-se a limitação quanto aos detalhes dos produtos. Alguns produtos podem variar significativamente os preços conforme diferenças de marca e tamanho da embalagem. Por conta da limitação do conjunto de dados e como necessidade de generalizar o conjunto de dados, esses detalhes não foram trabalhados nos experimentos. Outras limitações dizem respeito ao período (2013-2018) e atributos utilizados. Outros atributos, que não estejam disponíveis nos arquivos do portal da transparência, podem ser mais relevantes para o modelo de aprendizado de máquina, do que os que foram utilizados.

Para a abordagem de métodos de IA utilizados para classificar um caso como conluio, há outras opções que poderiam ter sido experimentadas, como métodos de aprendizagem não supervisionada (clusterização, por exemplo) ou análise de grafos. A forma de cálculo do tempo também é uma limitação. Foi utilizada a biblioteca do *python* para realizar esse cálculo. No entanto, algumas das mensurações podem ter sofrido mais influências de processamentos *background* que outras. Apesar disso, a estratégia foi adequada para mensurar uma ordem de grandeza.

Outra limitação foi sobre a forma de medir o desempenho. Foi executada uma única rodada de treinamento e teste para cada modelo. A alternativa seria a execução de diversas rodadas e calcular a média dos desempenhos. Os treinamentos dos modelos de aprendizado de máquina também foram restritos à utilização dos hiperparâmetros padrão. Para melhorar o desempenho individual de cada modelo, o conjunto de dados poderia ter sido dividido em treinamento, teste e validação, para

incorporar uma etapa de otimização dos hiperparâmetros. A simplificação dessas etapas foi devida à quantidade de experimentos, volume de dados e maior foco nas questões de explicabilidade.

Quanto ao método de rotulação de conluios, pela ausência da informação de quais licitações tiveram influência das condenações de conluios levantadas, foram propostos critérios para essa classificação. Os critérios aplicados podem rotular erroneamente um caso de conluio, mas foram pensados critérios mais restritivos possíveis para evitar principalmente o erro na rotulação Sim para conluio. Não é possível afirmar que os casos rotulados como Não, não tiveram a influência de algum conluio não identificado pelo CADE. Por este motivo, outra decisão foi restringir os dados aos mercados de produtos que tiveram alguma análise e condenação de conluio. No entanto, essa decisão limita o treinamento dos modelos de aprendizado a alguns tipos de produtos.

### VI.3 Trabalhos futuros

Como trabalhos futuros, pode-se enriquecer o conjunto de dados com as informações dos lances de cada licitação e maiores detalhes sobre o produto. Esse enriquecimento oportuniza experimentos com outras variáveis estatísticas utilizadas na literatura, como Diferença Absoluta (DIFFA), Diferença Percentual (DIFFP), *Spread* ou Dispersão (SPD), *Relative Difference Alternative* (RDALT), *Relative Difference Normalized* (RDNOR), *Relative Difference* (RD) e Kolmogorov-Smirnov (KS). Outra sugestão seria, a partir dos detalhes dos produtos, confrontar os valores ofertados com os valores de referência do produto homologado pelo TCU e verificar se a magnitude das diferenças indica uma organização de conluio.

Outro ponto a ser explorado é o aperfeiçoamento do desempenho dos algoritmos ajustando os hiperparâmetros dos modelos de aprendizado de máquina. Este trabalho utilizou os parâmetros padrões para a maioria dos modelos. Experimentar configurações diferentes de hiperparâmetros pode incrementar o desempenho dos modelos. Também vale aplicar o mesmo método e passos para execução em outro conjunto de dados, a fim de verificar se as conclusões sobre a importância dos atributos e variáveis estatísticas são as mesmas. Por fim, devido à natureza do problema, avaliar a possibilidade de incluir cálculos para estimar o grau de incerteza acerca de cada rotulação e dos resultados preditos pelos modelos.

Além disso, há oportunidade de explorar a solução do problema com medidas de centralidade e grafos, relacionando empresas que tenham participado de uma mesma licitação. A questão temporal também poderia ser melhor explorada. Foi utilizado o atributo ano da licitação para treinamento dos modelos, que não demonstrou ser relevante para esse experimento. Mas cabe avaliar outros métodos, como séries temporais, para identificar padrões de repetições de conluios por períodos do ano e/ou determinar possíveis recorrências temporais.

## Referências


Abbas, K. A. and Turky Rashid, M. Descriptive statistical features-based improvement of hand gesture identification. *Biomedical Signal Processing and Control*, 92, 2024.

Abrantes-Metz, R. M., Froeb, L. M., Geweke, J., and Taylor, C. T. A variance screen for collusion. *International Journal of Industrial Organization*, 24(3):467 – 486, 2006.

Amaral, W. S. Análise de grafos para apoio em auditoria de licitações públicas. *Dissertação (Mestrado em Ciência da Computação) - Centro Federal de Educação de Educação Tecnológica Celso Suckow da Fonseca (CEFET/RJ)*, 2020.

Aryal, G. and Gabrielli, M. F. Testing for collusion in asymmetric first-price auctions. *International Journal of Industrial Organization*, 31(1):26 – 35, 2013.

Bajari, P. and Ye, L. Deciding between competition and collusion. *Review of Economics and Statistics*, 85(4):971 – 989, 2003.

Ballesteros-Pérez, P., Skitmore, M., Das, R., and Del Campo-Hitschfeld, M. L. Quick abnormal-bid-detection method for construction contract auctions. *Journal of Construction Engineering and Management*, 141(7), 2015.

Bolotova, Y., Connor, J. M., and Miller, D. J. The impact of collusion on price behavior: Empirical results from two recent cases. *International Journal of Industrial Organization*, 26(6):1290 – 1307, 2008.

Brasil. Constituição da república federativa do brasil de 1988. *Constituição da República Federativa do Brasil de 1988.*, 1988.

Brasil. Lei nº 12.529, de 01 de dezembro de 2021. estrutura o sistema brasileiro de defesa da concorrência; dispõe sobre a prevenção e repressão às infrações contra a ordem econômica. *Diário Oficial [da] República Federativa do Brasil*, 2011.

Brasil. Lei nº 14.133, de 01 de abril de 2021. estabelece normas gerais de licitação e contratação para as administrações públicas diretas, autárquicas e fundacionais da união, dos estados, do distrito federal e dos municípios. *Diário Oficial [da] República Federativa do Brasil*, 2021.

Busu, M. and Busu, C. Detecting bid-rigging in public procurement. a cluster analysis approach. *Administrative Sciences*, 11(1), 2021.

CADE. *Guia Combate a cartéis em licitação*. Conselho Administrativo de Defesa Econômica - CADE, 2019.

Chicco, D. and Jurman, G. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1):1–13, 2020.

Commission, E. and for Competition, D.-G. *Quantifying antitrust damages towards non-binding guidance for courts*. Publications Office of the European Union, 2009.

Dadfarnia, M., Adibnia, F., Abadi, M., and Dorri, A. Incremental collusive fraud detection in large-scale online auction networks. *Journal of Supercomputing*, 76(9):7416 – 7437, 2020.

Du, Y., Rafferty, A. R., McAuliffe, F. M., Mehegan, J., and Mooney, C. Towards an explainable clinical decision support system for large-for-gestational-age births. *PLoS ONE*, 18(2 February), 2023.

Fayaz, M., Qureshi, M. S., Kussainova, K., Burkanova, B., Aljarbouh, A., and Qureshi, M. B. An improved brain mri classification methodology based on statistical features and machine learning algorithms. *Computational and Mathematical Methods in Medicine*, 2021, 2021.

Fegade, T. K., Pawar, B., and Bhavsar, R. Crop recommendation using hybrid ensembles model by extracting statistical measures. *Lecture Notes in Networks and Systems*, 662 LNNS:269 – 284, 2023.

García Rodríguez, M. J., Rodríguez-Montequín, V., Ballesteros-Pérez, P., Love, P. E., and Signor, R. Collusion detection in public procurement auctions with machine learning algorithms. *Automation in Construction*, 133:104047, 2022.

Goldschmidt, R. and Passos, E. *Data mining: um guia Prático*. Elsevier Editora, 2005.

Graham, D. A. and Marshall, R. C. Collusive bidder behavior at single-object second-price and english auctions. *Journal of Political Economy*, 95:1217 – 1239, 1987.

Greige, L., De Mesentier Silva, F., Trotter, M., Lawrence, C., Chin, P., and Varadarajan, D. Collusion detection in team-based multiplayer games. In *CEUR Workshop Proceedings*, volume 3121, 2022.

Hair, J., Black, W., Babin, B., and Anderson, R. *Multivariate Data Analysis: A Global Perspective*. Pearson Prentice Hall, 2010.

Hastie, T., Tibshirani, R., and Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer New York, 2013.

Haug, A., Zachariassen, F., and Liempd, D. The costs of poor data quality. *Journal of Industrial Engineering and Management*, 4(2):168–193, 2011.

Hu, J., Fang, D., Wei, X., and Xie, J. Colluder detection based on hypergraph decomposition. In *CIS 2013*, page 630 – 634, 2013.

Huber, M. and Imhof, D. Machine learning with screens for detecting bid-rigging cartels. *International Journal of Industrial Organization*, 65:277 – 301, 2019.

Iban, M. C. An explainable model for the mass appraisal of residences: The application of tree-based machine learning algorithms and interpretation of value determinants. *Habitat International*, 128, 2022.

IIA. Internation standards for the professional practive of internal auditing (standards). *The Institute of Internal Auditors Standards*, 2017.

Imhof, D., Karagök, Y., and Rutz, S. Screening for bid rigging-does it work? *Journal of Competition Law and Economics*, 14(2):235 – 261, 2018.

Imhof, D. and Wallimann, H. Detecting bid-rigging coalitions in different countries and auction formats. *International Review of Law and Economics*, 68, 2021.

Jain, N. and Jana, P. K. Xrrf: An explainable reasonably randomised forest algorithm for classification and regression problems. *Information Sciences*, 613:139 – 160, 2022.

Jiménez, J. L. and Perdiguero, J. Does rigidity of prices hide collusion? *Review of Industrial Organization*, 41(3):223 – 248, 2012.

Júnior, D. P. G., de Sousa Filho, G. F., and dos Anjos Formiga Cabral, L. Classificação de fraudes em licitações públicas através do agrupamento de empresas em conluios. *Anais do XI Workshop de Computação Aplicada em Governo Eletrônico (WCGE 2023)*, 2023.

Kuhn, M. and Johnson, K. *Applied predictive modeling*. Springer New York, 2013.

Lei, M., Yin, Z., Li, S., and Li, H. Detecting the collusive bidding behavior in below average bid auction. In *Fuzzy Systems and Knowledge Discovery*, page 1720 – 1727, 2018.

Liu, Q. and Serfes, K. Market segmentation and collusive behavior. *International Journal of Industrial Organization*, 25(2):355–378, 2007.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. In *NIPS 2017*, volume 2017-December, page 4766 – 4775, 2017.

Miller, T. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019.

Morettin, P. and Bussab, W. *Estatística básica*. Saraiva Educação S.A., 2017.

Nguyen, H. V. and Byeon, H. Prediction of parkinson's disease depression using lime-based stacking ensemble model. *Mathematics*, 11(3), 2023.

OECD. *Guidelines for Fighting Bid Rigging in Public Procurement*. OECD Library, 2009.

OECD. *Guidelines for helping competition authorities assess the expected impact of their activities*. OECD Library, 2014.

Porter, R. H. and Zona, J. D. Detection of bid rigging in procurement auctions. *Journal of Political Economy*, 101(3):518–538, 1993.

Porter, R. H. and Zona, J. D. Ohio school milk markets: An analysis of bidding. *RAND Journal of Economics*, 30(2):263 – 288, 1999.

Ribeiro, M. T., Singh, S., and Guestrin, C. "why should i trust you?"explaining the predictions of any classifier. In *KDD 2016*, volume 13-17-August-2016, page 1135 – 1144, 2016.

Russell, S. and Norvig, P. *Artificial Intelligence*, volume 3nd edition. Elsevier Editora Ltda, 2013.

Scoralick, L., Brandão, D., and Belloze, K. Aprimoramento de modelos para detecção de conluios em licitações públicas brasileiras com variáveis estatísticas e modelos explicáveis. In *Anais do XXXIX Simpósio Brasileiro de Bancos de Dados*, pages 680–686, Porto Alegre, RS, Brasil. SBC, 2024.

Shapley, L. A value for n-person games. *Contributions to the Theory of Games*, pages 307–317, 1953.

Signor, R., Ballesteros-Perez, P., and Love, P. E. Collusion detection in infrastructure procurement: A modified order statistic method for uncapped auctions. *IEEE Transactions on Engineering Management*, 70(2):464 – 477, 2023.

Silveira, D., de Moraes, L. B., Fiuza, E. P., and Cajueiro, D. O. Who are you? cartel detection using unlabeled data. *International Journal of Industrial Organization*, 88, 2023.

Silveira, D., Vasconcelos, S., Resende, M., and Cajueiro, D. O. Won't get fooled again: A supervised machine learning approach for screening gasoline cartels. *Energy Economics*, 105, 2022.

TCU. *Cartéis, superfaturamento e a atuação do TCU*. Secom do TCU, 2021.

Toledo, G. and Ovalle, I. *Estatística básica*. Editora Atlas S.A., 1982.

Wallimann, H., Imhof, D., and Huber, M. A machine learning approach for flagging incomplete bid-rigging cartels. *Computational Economics*, 62(4):1669 – 1720, 2023.

Wang, R., Storey, V., and Firth, C. A framework for analysis of data quality research. *IEEE Transactions on Knowledge and Data Engineering*, 7(4):623–640, 1995.

Wang, Z., Zhang, L., Huang, T., Yang, R., Cheng, H., Wang, H., Yin, H., and Lyu, J. Developing an explainable machine learning model to predict the mechanical ventilation duration of patients with ards in intensive care units. *Heart and Lung*, 58:74 – 81, 2023.