Universidade de Brasília
Instituto de Ciências Exatas - IE
Departamento de Estatística - EST

Predição e Detecção de Anomalias em Pregões Eletrônicos do GDF

Rodrigo Fernando Murça Barroso

Projeto apresentado para obtenção do título
de Bacharel em Estatística

Rodrigo Fernando Murça Barroso

Predição e Detecção de Anomalias em Pregões Eletrônicos do GDF

Orientador:
Prof. Guilherme Souza Rodrigues

Projeto apresentado para obtenção do título
de Bacharel em Estatística

Brasília
2020

Agradecimentos

Aos meus pais, Márcia e Mário, que se esforçaram a todo o momento para que eu tivesse uma vida feliz no futuro, com orientações, cuidados com a saúde, carinho, etc. E ao meu irmão, Vítor, que ajudou a formar minha personalidade.

Aos meus tios, tias, avô, avós, primas, primos e bisavó, por todo o carinho e apoio que sempre me deram.

Aos meu colegas de faculdade, que de alguma forma me ajudaram em momentos complicados – eu não esqueci nenhum deles. Em especial, João, Letícia, Gauthier, Gabriel, Douglas e Brenda, que, além de me darem conselhos, fizeram com que a faculdade fosse um lugar que gostava de ir todos os dias. Ao Fernando, por sempre ajudar nas tarefas quando eu precisei sem pedir nada em troca.

À minha namorada, Letícia, que me deixa feliz e motivado o tempo inteiro, me dá conselhos, me diverte, me cuida, e foi importante para a produção desse TCC.

A todos os professores, que me proporcionaram diversos momentos diferentes, de admiração, sono, vontade, proatividade, momentos descontraídos, etc. Em especial os professores Leandro, Maria Tereza e Gilardoni, pela compreensão e empatia pelos alunos.

Aos esportes que pratico regularmente, que, além de fazerem eu me sentir forte e feliz, esvaziam a minha mente, deixando-a preparada para uma maratona na UNB.

Aos funcionários da UNB, envolvidos por deixarem essa universidade um ambiente muito agradável, sendo a minha segunda casa a maior parte do curso.

Ao Restaurante Universitário por proporcionar momentos inesquecíveis com amigos, e uma comida que me deixava feliz o dia todo.

Ao meu orientador Prof. Guilherme, que sempre buscou ter uma boa relação comigo, depositou esperanças em mim e deu conselhos fundamentais para a produção desse TCC.

Resumo

Este trabalho apresenta um conjunto de ferramentas desenvolvidas para facilitar o monitoramento dos pregões eletrônicos feitos pelo Governo do Distrito Federal (GDF). Para tanto, formulamos uma função de mineração de dados para coletar e validar dados. A partir desses dados, foi feita a análise da diferença entre o lance vencedor e um preço de referência estimado por um modelo de regressão *boosted* beta. Para encontrar anomalias, foi criado um meta-aprendizado entre o modelo estatístico citado e um algoritmo de aprendizado de máquina chamado Floresta de Isolamento.

Palavras-Chave: Modelos de regressão beta. Anomalias em pregões. Floresta de isolamento. Agrupamento. Mineração de dados. Métricas de qualidade. Árvore de decisão. Validação cruzada.

Abstract

In this work we present a set of techniques that aims to improve the monitoring process of electronics preaching in the Distrito Federal's Government (GDF). We extracted information about the winner bidding and the reference price using Text Mining algorithms. Then we analyzed these variables using the Regression Boosted Beta model. Also, we detected anomalous observations using a meta-learning algorithm based on the Regression Boosted Beta model and the Isolation Forest algorithm.

Key Words: Beta regression models. Preaching anomalies. Isolation forest. Clustering. Text mining. Quality metrics. Decision tree. Cross validation.

Conteúdo

Lista de Figuras

Lista de Tabelas

1 Introdução

O governo brasileiro, com a missão de sustentar diversos órgãos e setores públicos, sempre comprou materiais e serviços. No entanto, a realidade do país impõe cada vez mais uma necessidade: otimizar gastos. Nesse contexto, a lei das licitações (Lei nº 8.666), de 1993, ampara a necessidade dos órgãos públicos de contratar serviços ou comprar produtos com a melhor qualidade e menor preço. Vantajosas para o governo, as licitações devem respeitar princípios como ética, justiça, igualdade para os participantes, celeridade e objetividade nas decisões. A igualdade, por exemplo, permite uma disputa justa entre empresas de pequeno e grande porte pela aquisição e fornecimento de bens e serviços.

Em 2002 foi criada a lei federal 10.520/2002 (Lei do Pregão), na qual foi introduzida a modalidade mais recente de licitação, o pregão, que faz parte do tipo de licitação menor preço (TCU, Licitações e Contratos, 2010)[16, p. 111–115], no qual a entidade pública tem interesse pelo menor preço. Essa modalidade é amplamente utilizada para a realização de compras governamentais, permitindo uma otimização de gastos por parte do governo.

Pregão eletrônico é uma modalidade de licitação que o governo usa para contratar bens e serviços. Nessa modalidade, empresas concorrem entre si com o lançamento de ofertas, chamados de lances, cujos preços não podem ultrapassar o preço de referência estipulado em uma pesquisa de mercado. O lance homologado, ou vencedor, é o mais vantajoso para administração pública.

Haja vista o dado informado pelo Ministério do Planejamento, Orçamento e Gestão, que indica uma economia de 48 bilhões de reais ao governo entre 2010 e 2015, essa forma de compra poupa aos cofres públicos uma considerável quantia. Uma das vantagens do pregão eletrônico é a transparência, pois é público e de fácil acesso a todos. O sistema responsável por melhorar os processos de compra e aquisições do Governo Federal é o Sistema integrado de Administração de Serviços Gerais (SIASG). O SIASG é crucial para a facilidade organizacional dos pregões eletrônicos. Entre seus benefícios, tem-se o prático cadastro dos fornecedores e a celeridade dos processos.

Apesar das vantagens, o sitema de pregões eletrônicos é passível de fraudes de variados tipos, além disso, a possível falta de padronização pode resultar em números irreais de preços os quais impactariam negativamente os gastos do GDF. Assim, é de grande valia estudar o certame licitatório em busca de melhorar o processo de compras.

Sabendo da importância da inteligência nos gastos públicos, é de interesse do projeto apurar eventuais falhas do processo licitatório, bem como demarcar possíveis melhorias no sistema de preços praticados. Em cada compra disponível, por exemplo, serão analisados o lance vencedor, o menor lance, o valor de referência e o valor final após sua negociação.

Existem algumas técnicas recentes na estatística e no aprendizado de máquina que são capazes de contribuir significativamente com a investigação; além disso, algumas estratégias de extração de dados são capazes de coletar qualquer conteúdo da internet, fortalecendo a pesquisa com robustez e confiabilidade.

A ideia é que este trabalho consiga facilitar o processo de investigação de atas de pregões eletrônicos ao formular uma ferramenta que colete dados, e uma que capte anomalias.

Assim, esse trabalho tem os seguintes objetivos:

- identificar possíveis fraudes nos pregões, como lances homologados inesperados, negociações exageradas etc. Retornar uma lista dos itens mais discrepantes;

- obter uma análise da diferença entre o preço estimado e o preço do lance vencedor, e fazer previsões a respeito dessa diferença;

- automatizar o processo de coleta de dados para realizar a investigação em qualquer certame licitatório de interesse.

2 Conceitos Importantes

2.1 Pregões Eletrônicos

Nessa modalidade, assim como no leilão, a qualidade do produto não contribui para a vitória do fornecedor, uma vez que esta já é especificada no edital do pregão. A disputa é feita apenas com base no preço lançado. No leilão, o vencedor é o que faz o lance com maior preço para o produto. Já no pregão, o item homologado é o de menor preço [2], com o objetivo de o governo poupar gastos. Assim, para o governo, o leilão é importante para realizar vendas, e o pregão, para compras. Este trabalho é focado inteiramente nos pregões.

Apesar de a modalidade ter sido criada em 2002, foi em 2005, com o decreto 5.450/2005, que o pregão na forma eletrônica foi estabelecido para as compras do GDF e, posteriormente, tornado obrigatório, salvo excessões de inviabilidade, em 2019 pelo Decreto 10024/2019 (Art. 2º; §1º).

Decreto 5.450/2005

Art. 1º A modalidade de licitação pregão, na forma eletrônica, de acordo com o disposto no § 1º do art. 2º da Lei nº 10.520, de 17 de julho de 2002, destina-se à aquisição de bens e serviços comuns, no âmbito da União, e submete-se ao regulamento estabelecido neste Decreto.

Art. 2º O pregão, na forma eletrônica, como modalidade de licitação do tipo menor preço, realizar-se-á quando a disputa pelo fornecimento de bens ou serviços comuns for feita à distância em sessão pública, por meio de sistema que promova a comunicação pela internet.

§ 1º Consideram-se bens e serviços comuns, aqueles cujos padrões de desempenho e qualidade possam ser objetivamente definidos pelo edital, por meio de especificações usuais do mercado.

Os pregões eletrônicos funcionam da seguinte forma: tudo começa com uma necessidade de uma UASG (Unidade Administrativa de Serviços Gerais) para algum bem ou serviço, o qual deve ser comum[1]. Essa necessidade deve ser justificada e detalhada, e é julgada pelo Secretário-Geral da administração do Tribunal de Contas da União. A partir disso, cria-se, por meio de um processo, um termo de referência para o pregão (TCU, Licitações e Contratos, 2010)[16, p. 80–85], onde deve estar esclarecida a demanda, o preço estimado, a quantidade demandada, qual o objetivo da compra, especificações do produto a ser comprado, prazo de entrega, etc.

Esses editais contêm diversos itens cada, que representam cada compra a ser realizada. Logo antes da etapa dos lances, há a fase de abertura do pregão. Nesse momento, diferentes empresas soltam lances para o item, com o intuito de afirmar interesse ao item, além disso, é verificada a conformidade da proposta. Nessa etapa não são analisados os valores de cada lance e, por esse motivo, tais lances podem apresentar valores muito diferentes do mercado. A proposta só avança para a fase de lances após confirmarem sua adequabilidade, assim um fornecedor que não apareceu na fase de abertura, não pode aparecer na fase de lances, pois contraria parte do Acórdão 539/2007 das normas do TCU sob licitações e contratos (TCU, Licitações e Contratos, 2010)[16, p. 76]. A ocorrência dessa norma será analisada neste trabalho.

[2] A partir de 20 de setembro de 2019, com o Decreto 10024/2019 (Art.7º), o item homologado pode ser também o item com melhor negociação, não sendo necessariamente o menor lance da fase de lances.

[1] Bens e serviços comuns: São possíveis de estabelecer padrões de desempenho e qualidade no edital de forma objetiva (TCU, Licitações e Contratos, 2010)[16, p.64]. Assim, podem ser escolhidos apenas com base em seus preços ofertados. Um exemplo de serviços considerados não comuns são, em geral, os de engenharia.

Fonte: Portal Conexão Mineral

Iniciada a fase de lances, os fornecedores aprovados competem entre si com sucessivas propostas, e o item é adjudicado ao licitante com a proposta de menor preço. Caso não haja propostas após a abertura do certame, verifica-se o melhor lance feito na fase anterior. Em geral isso ocorre na falta total de concorrência, quando apenas um licitante é aprovado na fase de abertura do certame. Essa falta de concorrência não é interessante para o comprador sob o ponto de vista econômico. Ainda sobre o certame, o pregoeiro determina o período de 30 min em que se encerrará a fase de lances aleatoriamente pelo sistema [3].

A partir do encerramento, não ocorrem mais lances, salvo em excepcionalidades onde há a reabertura do pregão

Em geral, os pregões eletrônicos seguem um documento chamado Ata de Registro de Preços – ARP. Nele, encontram-se registrados os preços vantajosos para a administração, os fornecedores, os órgãos participantes e as condições processuais na licitação.

Nesse processo de compras também há a possibilidade de se realizarem **negociações**, as quais devem ser baseadas em dois critérios:

1. Caso possível, o acordo deve ser sensato e eficiente;
2. Não se deve deteriorar a relação entre as partes.

Portanto, o pregoeiro deve negociar com muita responsabilidade, buscando assegurar o interesse da administração, ao mesmo tempo em que é justo com o licitante.

Apesar das vantagens, o sitema de pregões eletrônicos, **comprasnet**, é passível de fraudes de variados tipos (Oliveira, 2016)[12], entre elas temos as chamadas "coelhos":

Coelhos: Fornecedores que ofertam preços muito abaixo do valor de referência, o que garantiria vitória. Porém, reprovam de propósito na fase de documentação, com o intuito de que o segundo colocado, agindo em conluio com o coelho, vença o pregão, prejudicando assim, a admininstração pública e os outros licitantes.

[3] A partir de 20 de setembro de 2019, com o Decreto 10024/2019 (Art.7º), mesmo após o término do prazo, o prazo é estendido até que não haja nenhum lance dentro de 2 minutos.

2.2 Métodos de *Machine Learning*

Em 1997, Tom M. Mitchell definiu aprendizado de máquina dessa forma:

> "Diz-se que um programa de computador aprende pela experiência E, com respeito a algum tipo de tarefa T e performance P, se sua performance P nas tarefas em T, na forma medida por P, melhoram com a experiência E."[10]

O *Machine Learning*, ou aprendizado de máquina, é um campo totalmente focado em resolver problemas práticos, como classificação e previsão, e tem a característica de se preocupar apenas com o que o dado oferece, e não com a formalização e pressupostos matemáticos, exercendo o raciocínio indutivo em um método puramente operacional. No aprendizado de máquina, o algoritmo consegue melhorar sua própria performance com base na experiência, utilizando técnicas de algoritmos supervisionados e não supervisionados.

- Supervisionado: quando já se tem um conjunto de dados com as relações e classificações feitas. O algoritmo supervisionado buscará um bom ajuste aos dados aprendendo com os erros, e generalizará os resultados para conjuntos de dados desconhecidos ou futuros;
- Não supervisionado: quando o algoritmo tem o dever de descobrir padrões desconhecidos em dados não rotulados;
- Por reforço: não tem o intuito de fazer uma predição ou uma classificação a partir de um conjunto de dados. Ele é guiado em um ambiente de estudo por tentativa e erro, maximizando uma recompensa final numérica, dada como orientação ao objetivo.

Esta seção abordará os principais métodos de *machine learning* de interesse neste trabalho, começando com mineração de dados e concluindo com o *Isolation Forest*

2.2.1 Mineração de Dados

Foi usada a Mineração de Dados para extrair informações de PDFs para este trabalho. Este recurso tem a capacidade de filtrar, de um conjunto de informações grande (e às vezes disperso), os conhecimentos de interesse do produtor, seja de um estudo, seja do mercado de trabalho. É um subterfúgio muito usado no meio profissional, com uso muito comum em diversas áreas, como o marketing (ver site **rockcontent**).

O *software* R faz a leitura do PDF completo, o qual contém várias informações, relevantes ou não. O algoritmo de Mineração de Dados nesse caso será implementado primeiramente para conseguir extrair os dados prioritários organizadamente.

Após isso, os códigos são adaptados para buscar outros padrões de outros arquivos até que este consiga coletar os dados de atas governamentais desconhecidas com sucesso. Buscam-se então padrões comportamentais comuns entre os arquivos, com o intuito de fazer a coleta de forma automatizada. Ao coletar dados direto da fonte, problemas como redundância extrapolação de dados são resolvidos. A limpeza dos dados também é implementada no algoritmo.

2.2.2 k-*means*

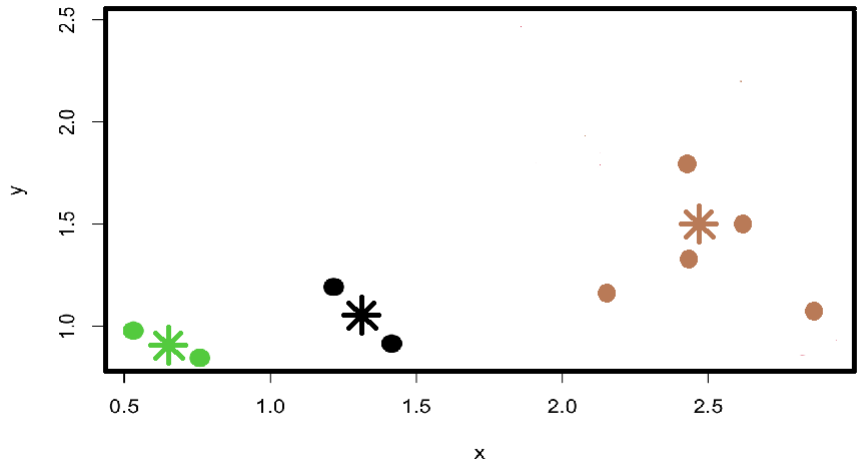
No período do *Big data*, é cada vez mais importante buscar alternativas para reduzir a dimensionalidade, agrupar, entre outras ações. Para agrupar, é bastante comum o uso do algoritmo não supervisionado k-médias (*k-means*), que busca agrupar dados considerados similares por algum critério de distância.

Para implementar o algoritmo, escolhem-se as n variáveis de análise e o número de classes de interesse. Assim, cada observação pertence ao hiperplano formado por essas variáveis. No espaço onde concentra a maior massa de dados de uma certa característica, encontra-se o centroide de uma das k classes, que é calculado pelo algoritmo.

O cálculo dos centroides pede a escolha inicial de k centroides arbitrariamente. Assim, calculam-se as distâncias dos pontos para cada centroide. Os centroides são recalculados pela média das observações mais próximas. O processo iterativo consiste em recalcular outros k centroides até que minimizem a soma dos quadrados dessas distâncias.

Com os k centroides calculados, é medida a distância n-dimencional de um ponto para cada um destes centroides (Drummond, 2013)[3], assim, o ponto é associado ao cluster de centroide mais próximo. A Figura 1 indica como fica a clusterização feita com a utilização de duas variáveis.

Figura 1: k-*means* com os Centróides Finais



2.2.3 Árvore de Decisão

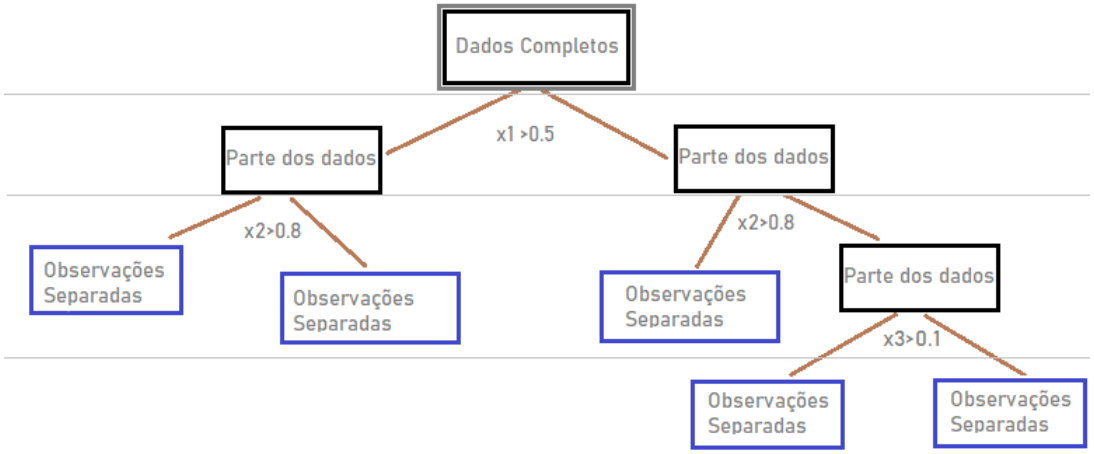
A Figura 2 indica como funciona uma árvore de decisão. Cada quadrado é um nó da árvore, e cada ramo, ou caminho de uma observação, é um conjunto de regras de decisão (Gama, 2002)[6].

Os dados completos pertencem ao nó raiz da árvore, no qual os dados são divididos pela primeira regra de decisão, nesse caso $x_1 > 0.5$, estabelecida pelo parâmetro **crescimento**. Assim, os ramos levam os dados para os nós filhos. Sobre a direção, caso o valor da regra de decisão seja verdadeiro, os dados caminham pelo ramo da direita até o nó descendente, ao passo que valores falsos os levam para a esquerda. Após todas as regras de decisão serem testadas, temos os nós "folha", representados pelos quadrados azuis. Note que algumas observações necessitam de mais testes para serem separadas, isso significa que são dados muito semelhantes em relação as variáveis (*features*) do banco de dados, precisando de muitas cisões para fazer a separação desejada.

Já algumas observações são isoladas muito facilmente em relação a outras, pois estão mais distantes da massa de dados.

O algoritmo do exemplo apresenta a característica de ser binário, pois cada regra de decisão divide os dados em apenas duas classes.

Figura 2: Árvore de Decisão binária



Outro importante parâmetro é a **poda**, que controla o momento de parada do algoritmo, representado pelas linhas cinzas da Figura 3. Nesse exemplo, caso a função pare na primeira iteração, os dados estarão agrupados em duas classes; caso pare na segunda, serão quatro classes; e caso pare na ultima iteração, o número de classes será igual ao número de observações. Essa escolha pode ser feita utilizando cálculos complexos que estimam erros, ou apenas com a análise visual do gráfico.

2.2.4 Floresta de Isolamento

A existência de outliers[2] implica, tipicamente, em prejuízos para algoritmos de previsão e classificação.

Assim, Liu et al., 2008 [8] criaram um algoritmo não supervisionado de grande valor para detecção de *outliers* em um conjunto de dados multivariados, o *Isolation Forest*.

Esse algoritmo retorna uma métrica do quão anômala é a observação baseando-se em duas características bastante evidentes em pontos discrepantes: *few and different*.

1. *Few*: refere-se ao tipo de observação de baixa frequência.

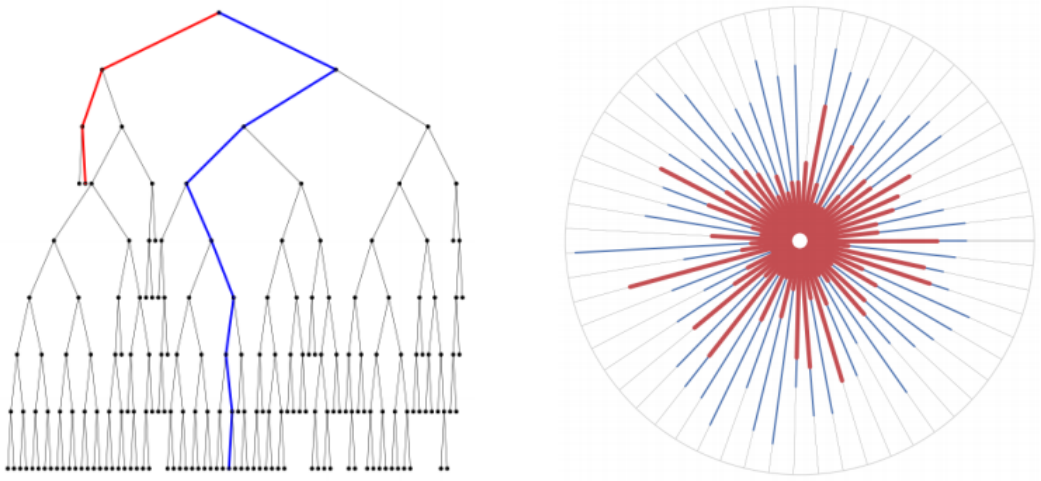
2. *Different*: refere-se a ao tipo de observação que apresenta valores muito distantes dos dados normais.

O funcionamento desse algoritmo consiste em criar diversas árvores de decisão binárias e combiná-las. Cada árvore recebe de entrada uma amostra do conjunto de dados e é executada individualmente. Em cada árvore, dados isolados próximo à raiz são provavelmetente discrepantes, ao passo que dados que necessitam de mais cisões para o separo, ou seja, mais longe da raiz, são provavelmente acordantes. Esse comportamento fica claro na Figura 3a, onde a trajetoria vermelha é de um *outlier* e a azul é de uma observação (instância) normal.

[2]**Outlier:** Refere-se a um valor atípico, anormal. Essas observações são naturalmente afastadas da massa de dados, seja por um motivo especial, seja por uma inconsistência, seja pela qualidade das variáveis.

Figura 3: Ilustração dos Diferentes Pontos em um *Isolation Forest*

(a) Representação de uma Árvore (b) Representação de um *Isolation Forest*



Fonte: towardsdatascience, **Isolation Forest from Scratch**

A Figura 3b representa uma árvore do *isolation forest*, em que o raio é seu número máximo de nós. As linhas azuis indicam o número de nós de uma observação normal nessa árvore, e as linhas vermelhas, de um *outlier*. O número de nós em que uma observação x passa em cada árvore é computada em $h(x)$, e é usada para obter uma medida que indica o quão anômala é cada observação em relação à massa de dados: o *score*. Antes de calcular o *score*, precisa-se normalizar o número de nós para facilitar a comparação entre diferentes amostras. Para isso, utiliza-se um fator normalizante $c(n)$, que indica o número médio de nós que todas as observações passam, estimado pela Fórmula (1):

$$c(n) = 2\left(ln(n-1) + \gamma\right) - 2\frac{(n-1)}{n}, \tag{1}$$

em que γ é a constante de Euler-Mascheroni, de valor 0.5772156649, e n é o número de observações.

Assim, temos tudo para executar a fórmula do *score* [2, p. 15](Ding et al., 2013) a seguir:

$$s(x,n) = 2^{-\frac{E(h(x))}{c(n)}}, \tag{2}$$

onde $E(h(x))$ é o número médio de nós que uma observação x precisa para ser isolada, considerando todas as árvores, e $c(n)$ é a média global de nós considerando todas as observações do banco de dados e todas as árvores.

Note que a comparação entre $E(h(x))$ e $c(n)$ é o ponto principal para a detecção de anomalias:

- em bancos de dados com anomalias, valores normais tem a tendência de serem isolados com um número maior de nós do que a média, ou seja, $E(h(x)) > c(n)$. Assim, a divisão $\frac{E(h(x))}{c(n)}$ é um valor acima de 1, tornando o valor do *score*, calculado pela Fórmula (2), um número menor ou igual a 0.5. Assim, sendo n grande,

$$\lim_{E(h(x))\to n} S(x,n) = 0;$$

- as anomalias têm a tendência de serem isoladas com um número menor de nós em relação à média global, ou seja, $E(h(x)) < c(n)$. Assim, a divisão $\frac{E(h(x))}{c(n)}$ é um valor entre 0 e 1, tornando o valor do *score* um número acima de 0.5. Quanto mais anormal a observação, mais próximo de 1 é seu *score*:

$$\lim_{E(h(x))\to 0} S(x,n) = 1;$$

- caso todas as observações apresentem $E(h(x)) = c(n)$, então o banco de dados não apresenta anomalias, e o valor do *score* será $S(x,n) = 2^{-1} = 0.5$ para cada observação.

Logo, o *score* assume valores no conjunto $(0,1)$, sendo valores próximos de 1 considerados *outliers* e valores próximos de 0, pontos normais.

Um parâmetro muito importante na implementação do algoritmo é o número de árvores da floresta. Utilizando apenas uma árvore, quase todas as observações terão o mesmo valor de *score*, pois $E(h(x))$ é uma média calculada com base em apenas uma árvore. Já um número grande de árvores com diferentes regras de decisão tornará $E(h(x))$ um valor muito mais variável para cada observação, permitindo dar um valor de *score* diferente para cada observação. Assim, o número de árvores escolhido é diretamente proporcional ao grau de separação entre as observações.

Como o objetivo é separar apenas os *outliers*, não há necessidade de um número muito grande de árvores. O ideal é fazer uma análise do número de árvores desejadas, e uma das formas de fazer isso é aumentar iterativamente o número de árvores para calcular a correlação entre os conjuntos de *scores* obtidos em cada iteração. Por exemplo, se a correlação entre uma iteração com 10 árvores e uma com 15 árvores é de 0.6, significa que o acrécimo de 5 árvores trouxe uma mudança significativa nos *scores*. Sob outra perspectiva, caso a correlação entre os *scores* de uma *isolation forest* com 50 árvores e uma de 75 árvores seja 1, significa que o acréscimo de 25 árvores não resultou em uma mudança significativa na separação das variáveis, ou seja, utilizar um número de árvores maior do que 50 apenas torna o algoritmo mais lento.

2.3 Modelos e Conceitos Estatísticos

A estatística tem foco principal em entender os dados, medir relações entre variáveis, estimar valores, estimar erros e intervalos de confiança. Diferentemente do *Machine Learning*, ela não deixa de lado nenhum requisito matemático para formulação de modelos, utilizando o raciocínio dedutivo ao invés do indutivo. Esta seção abordará alguns temas da estatística que serão levados em conta na produção desde trabalho.

2.3.1 Modelo de Regressão Linear Simples

Assim, um modelo de regressão linear simples é uma função (Fórmula 3) que gera valores de Y a partir de uma combinação linear de variáveis explicativas, obtendo a estimativa do valor esperado de Y.

$$Y = \beta_0 + X_1\beta_1 + X_2\beta_2 + \cdots + X_p\beta_p + \epsilon, \tag{3}$$

sendo $\beta_0, \cdots, \beta_p$ desconhecidos, chamados de coeficientes. Além disso, ϵ representa o erro aleatório, proveniente de fatores que são desconhecidos e não relacionados com as covariáveis $X_1, \cdots, X_p$.

Esse modelo tem como pré-requisitos :

- distribuição de probabilidade gaussiana para Y;
- $\epsilon \overset{i.i.d}{\sim} N(0, \sigma^2)$, com σ^2 constante. Ou seja, os erros são independentes, homocedásticos, gaussianos e com média zero.

Para aproximar o valor real Y, utilizam-se os coeficientes $\hat{\beta}_1, \cdots, \hat{\beta}_n$, e obtém-se $\hat{Y}$:

$$\hat{Y} = \hat{\beta}_0 + X_1\hat{\beta}_1 + X_2\hat{\beta}_2 + \cdots + X_p\hat{\beta}_p. \tag{4}$$

A fim de medir a qualidade de ajuste do modelo criado, estima-se o erro aleatório ϵ. Essa estimativa é o resíduo, $\hat{\epsilon}$, que mede a diferença entre $\hat{Y}$ e Y, ou seja, o quanto que o modelo errou para uma certa observação:

$$\hat{\epsilon} = Y - \hat{Y}. \tag{5}$$

O ajuste do modelo é feito de forma que as observações preditas se encontrem próximas dos valores observados. Uma forma de fazer isso é minimizando a soma dos quadrados dos resíduos, visto na Fórmula (6), conhecido como **método dos mínimos quadrados**. Encontram-se analiticamente os coeficientes $\hat{\beta}_0, \cdots, \hat{\beta}_n$ que minimizam (6) dado o conjunto de covariáveis.

$$min\left(\sum^{n} \hat{\epsilon^2}\right) = min\left(\sum^{n} \left(Y - (\hat{\beta}_0 + X_1\hat{\beta}_1 + X_2\hat{\beta}_2 + \cdots + X_p\hat{\beta}_p)\right)^2\right). \tag{6}$$

Outra forma de obter estimadores para os coeficientes é pelo método de **Máxima Verossimilhança**, que consiste em maximizar a função de verossimilhança.

Definição:
Seja uma amostra aleatória $\widetilde{y} = y_1, \cdots, y_n$ e um conjunto de coeficientes θ de uma variável aleatória com densidade $f(y, \theta)$, a função de verossimilhança é dada por:

$$L(\theta, \widetilde{y}) = \prod_{i=1}^{n} f(y_i; \theta). \tag{7}$$

Assim, o objetivo desse método é escolher um conjunto de parâmetros θ tal que maximize a função de verossimilhança, obtendo-se assim os estimadores de máxima verossimilhança.

Com o modelo feito, é necessário fazer um diagnóstico para saber se ele cumpre os pré-requisitos. Com as suposições de distribuição Gaussiana para Y e para ϵ então $\hat{Y}$ e, consequentemente, o resíduo devem seguir também essa distribuição. Logo, é verificado se os resíduos são gaussianos, homocedásticos e independentes entre si.

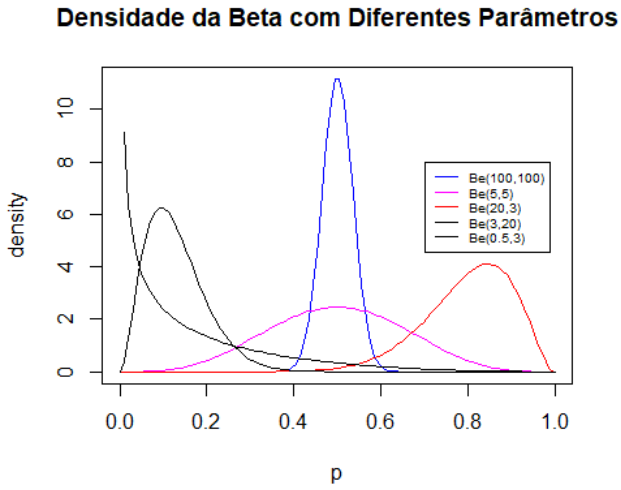
Apesar de ser um modelo bastante útil, ele não é recomendado para todos os tipos de dados, pois apresenta dificuldade em dados com características específicas.

2.3.2 Modelo de Regressão Beta

Modelos com distribuição normal não adequam-se bem a dados limitados no intervalo (0,1), como proporções e taxas. Esses tipos de dados são usualmente heterocedásticos e assimétricos, logo assumir que a variável dependente é gaussiana não é recomendado, principalmente para amostras pequenas. A distribuição beta, por outro lado, além de ser bastante maleável, aceitando heterocedasticidade e assimetria, apresenta suporte no intervalo (0,1). Assim, (Ferrari et al., 2004) [14] propuzeram um modelo de regressão assumindo que a variável resposta é beta-distribuida, diferente do modelo de regressão linear simples, que assumia distribuição gaussiana.

A Figura 4 mostra como a distribuição beta (Be(α,β)) pode apresentar diferentes formatos a medida que os parâmetros α e β são modificados:

Figura 4: Densidade da Beta para Diferentes Parâmetros



O Modelo de Regressão Beta, ou MRB, é composto de três componentes (Cordeiro , 2008)[1]:

1. componente aleatório, representado pela variável resposta beta-distribuida;
2. componente sistemático, representado pelo preditor linear, no qual entram as variáveis explicativas e seus respectivos coeficientes por meio de uma combinação linear;
3. uma função de ligação, que faz a associação entre os componentes aleatório e sistemático do modelo.

Para formular o modelo de regressão beta, foi feita uma reparametrização da densidade, a fim de que os parâmetros de posição e de precisão sejam, respectivamente, $\mu = \alpha/(\alpha+\beta)$ e $\phi = \alpha + \beta$. Para isso, substitui-se, na Equação (8), α por $\mu\phi$ e β por $(1-\mu)\phi$, obtendo-se a Equação (9).

$$f(y;\alpha;\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} y^{\alpha-1}(1-y)^{\beta-1}; \tag{8}$$

$$f(y;\mu;\phi) = \frac{\Gamma(\phi)}{\Gamma(\mu\phi)\Gamma((1-\mu)\phi)} y^{\mu\phi-1}(1-y)^{(1-\mu)\phi-1}, \tag{9}$$

sendo $0 < \mu < 1$ e $\phi > 0$.

Assim, $E(Y) = \frac{\mu\phi}{\mu\phi+(1-\mu)\phi} = \mu$ e $var(Y) = \frac{\mu\phi(1-\mu)\phi}{(\mu\phi+(1-\mu)\phi+1)(\mu\phi+(1-\mu)\phi)^2} = \frac{\mu(1-\mu)}{\phi+1}$.

Definição:

Seja uma amostra aleatória $y_1, \cdots, y_n$, sendo $Y_i \sim Be(\mu_i, \phi), i = 1, \cdots, n$, o modelo de regressão beta é definido como:

$$g(\mu_i) = x_i^T\beta = \eta_i,$$

onde μ_i é a média da variável Y_i, β é o vetor de parâmetros, um para cada covariável x_j, e η_i é o preditor linear, resultado da função de ligação g aplicada a μ_i.

Logo, ao multiplicar as covariáveis pelos seus parâmetros em uma dada observação e aplicar o resultado à transformação inversa de $g()$, é possível encontrar μ, que é uma estimativa da variável resposta.

A tabela a seguir mostra os exemplos de função de ligação mais comuns para modelos de regressão beta:

Tabela 1: Funções de Ligação

Nome	$g(\mu)$
Logit	$log(\frac{\mu}{1-\mu})$
Probit	$\Phi^{-1}(\mu)$
Complemento log-log	$log(-log(1-\mu))$
cauchit	$tan(\pi(\mu-0.5))$
log-log	$-log(-log(\mu))$

Nota-se também que, como a variância de Y_i é uma função da média, ela também é função das covariáveis. Logo, esse modelo naturalmente comporta variáveis dependentes heterocedásticas. Para estimar os parâmetros, é utilizado o método de máxima verossimilhança. No pacote *betareg* do R (Zeileis et al., 2016)[17], a forma utilizada para maximizar a função de log-verossimilhança é pela função *optim*. Nessa abordagem, estima-se o ϕ como constante e estima-se o μ para cada observação.

Outra opção de modelagem é estimar o μ e o ϕ, ambos como função linear das covariáveis, modelando assim não só a média da variável, mas a precisão, assim terá um preditor linear calculado para cada parâmetro a cada observação.

$$g(\mu) = x_i^T\beta_\mu = \eta_\mu; \tag{10}$$

$$g(\phi) = x_j^T\beta_\phi = \eta_\phi. \tag{11}$$

2.3.3 Seleção de Variáveis por Regularização *Boosted* Beta

Outra forma de estimação de parâmetros é pelo GamboostLSS (Hofner et al., 2014)[7], que tem como ideia central ajustar sucessivos modelos de regressão beta, atualizando o preditor linear a cada iteração.

Nele, utiliza-se a chamada **regularização** *boosted*[3], em que a cada iteração são calculadas as derivadas parciais da função de log-verossimilhança para um conjunto de parâmetros, que é a seguir atualizado.

Explicando brevemente como funciona uma iteração do boosting para os parâmetros $\hat{\mu}$ e $\hat{\phi}$ do modelo beta, tem-se a seguinte representação:

Iteração $n+1$:

$$\frac{\partial l(y, \hat{\mu}^{[n]}, \hat{\phi}^{[n]})}{\partial \eta_\mu} = 0 \overset{atualizar}{\rightarrow} \hat{\eta}_\mu^{[n+1]} \Rightarrow \hat{\mu}^{[n+1]};$$

$$\frac{\partial l(y, \hat{\mu}^{[n+1]}, \hat{\phi}^{[n]})}{\partial \eta_\phi} = 0 \overset{atualizar}{\rightarrow} \hat{\eta}_\phi^{[n+1]} \Rightarrow \hat{\phi}^{[n+1]}.$$

A cada iteração estimam-se os parâmetros que maximizam a função log-verossimilhança, os quais são incluídos na nova função de log-verossimilhança, a ser maximizada na próxima iteração.

A forma como o algoritmo escolhe os parâmetros iniciais é crucial para se obter uma seleção de variáveis robusta. Na primeira iteração, é selecionada apenas uma pequena parte das covariáveis, considerada de maior importância para explicar a variável resposta. Assim, utilizar apenas uma iteração geralmente provoca o *underfitting*. A cada iteração, uma maior parte das informações é utilizada, tornando o modelo cada vez melhor ajustado aos dados. Assim, utilizar muitas iterações pode provocar o *overfitting* (Figura 5). A ideia é parar no momento certo para reduzir a variância o suficiente para tornar as previsões menos instáveis e mais acertivas.

O próprio pacote *GamboostLSS* fornece uma função que faz o critério de parada conforme a validação cruzada de K-grupos, que será explicada mais a frente. Assim, a regressão regularizada boosting é um poderoso método de **seleção de variáveis**.

A métrica recomendada por (Mayr et al., 2012) [9], utilizada para comparar modelos *boosted*, é o risco preditivo, que é a função predita da log-verossimilhança negativa.

2.3.4 Métricas de Ajuste de um Modelo

Um modelo estatístico busca minimizar uma determinada métrica de erro para o conjunto de dados analisado. Esta seção abordará quatro métricas importantes, que serão utilizadas no decorrer no trabalho.

Erro Médio Quadrático:

A métrica erro médio quadrático, de sigla MSE, do inglês *Mean Squared Error*, é calculada da seguinte forma:

$$MSE = \frac{\sum_{i=1}^{n} \hat{\epsilon}_i^2}{n}, \tag{12}$$

Sendo $\hat{\epsilon_1}, \cdots, \hat{\epsilon_n}$, os resíduos das n predições de um modelo.

[3] *Boosting*: funciona pela combinação sequencial de algoritmos. Nesse método, os modelos têm enfoque nas observações menos acertadas pelos modelos anteriores, assim, o problema do viés é substancialmente reduzido.

Uma característica dessa métrica é que, ao elevar ao quadrado um resíduo alto, ele aumenta de valor. Logo, ela penaliza de forma cada vez maior os maiores erros do modelo. Por esse motivo, apesar do MSE ser bastante adequado a dados com alta variância, a presença de anomalias a deixa distorcida.

Erro Mediano Absoluto:

A métrica erro mediano absoluto é basicamente a mediana dos resíduos absolutos. Essa fórmula de cálculo é robusta a anomalias, no entanto apresenta dificuldade em dados com alta variância.

Erro Médio Absoluto:

A métrica erro médio absoluto, de sigla MAE, do inglês *Mean Absolute Error*, assim como o MSE, utiliza os resíduos ϵ para o cálculo, porém, cada termo do somatório, ao invés de elevar ao quadrado, aplica-se o módulo. Essa métrica indica o quanto que o modelo erra, em média, para cada observação. O cálculo é feito pela Fórmula (13).

$$MAE = \frac{\sum_{i=1}^{n} |\hat{\epsilon}_i|}{n}. \tag{13}$$

Uma característica dessa métrica é que, à medida em que a distância entre o valor real e a predição aumenta, o erro aumenta de forma linear. Assim, ela não é distorcida por anomalias como o MSE, e não é fraca para dados com elevada variância, como a mediana dos erros absolutos.

Critério de Akaike:
O critério de akaike, AIC, descrito em (Emiliano, 2010)[5], é calculado pela seguinte forma:

$$AIC = -2L(\hat{\theta}) + 2p. \tag{14}$$

Esse critério estima a quantidade de informação real perdida pelo modelo. Assim, quanto menor seu valor, melhor é o modelo.

2.3.5 Validação Cruzada

Ao ajustar um modelo para um determinado conjunto de dados, sabemos que quanto mais parâmetros, melhor o modelo se ajusta aos dados. Porém, isso não significa que esse modelo vai se ajustar bem a um outro conjunto de dados, executando uma boa generalização.

Ainda sob a capacidade preditiva de um modelo, devem-se evitar duas situações extremas: o *underfitting*: quando o modelo tem alta tendência e baixa variância, apresentando dificuldade de generalizar dados desconhecidos; e o *overfitting*: quando o modelo não apresenta tendência, mas apresenta elevada variância, ajustando perfeitamente ao conjunto de treino, mas apresenta baixo desempenho em dados desconhecidos.

Figura 5: *Overfitting* e *underfitting*

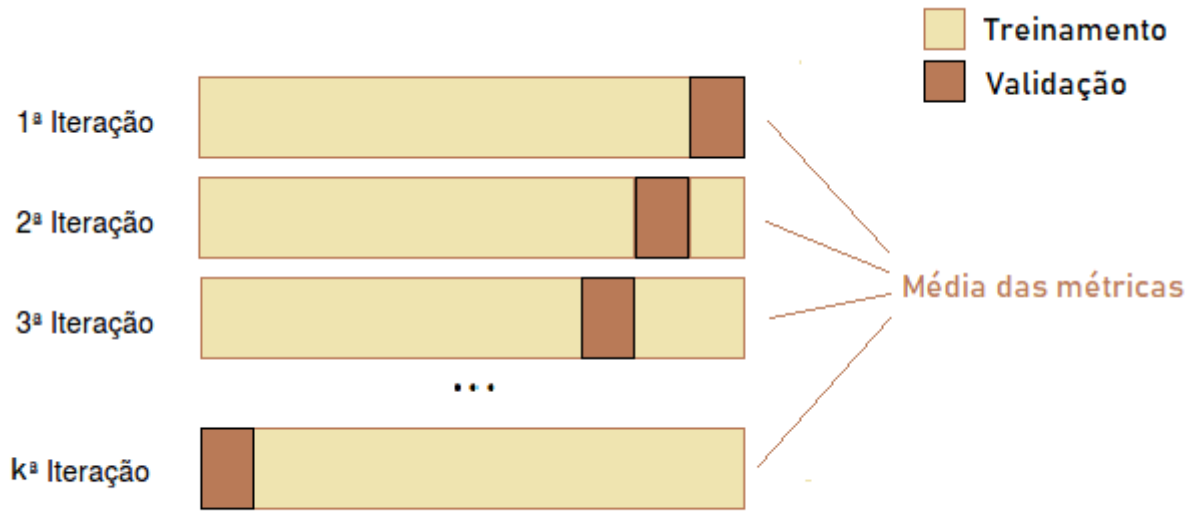
Repare, na Figura 5, que modelos com *underfitting* e *overfitting* têm dificuldade de prever dados desconhecidos, em azul.

A validação cruzada é uma forma de medir se o modelo feito para um conjunto de observações pode ser acertivo em um outro conjunto de observações desconhecido (Santana, 2020)[13].

Entre as técnicas de validação cruzada (ver **wiki/Cross-validation**), a validação cruzada em k-grupos, ou ***k-fold cross validation***, é bastante utilizada, e seus passos, representados na Figura 6, estão listados abaixo:

1. dividir os dados em K-grupos;
2. executar um modelo, excluindo um dos grupos, e depois utilizar o grupo que sobrou para validar;
3. repetir o passo anterior K vezes, sendo que, a cada iteração, usar outro grupo como sobra;
4. dado uma métrica escolhida, tirar a média aritmética dos resultados adquiridos nas iterações.

Figura 6: Representação da Validação Cruzada K-grupos



Com essa técnica, pode-se fazer uma comparação preditiva de modelos, e é muito útil para a decisão da complexidade ótima do modelo. Assim, fazer uma seleção de variáveis com base no *k-Fold Validation* é uma estratégia bastante utilizada, a exemplo do modelo de regressão *boosted* beta.

3 Metodologia

3.1 O Banco de Dados Inicial

O banco de dados proposto para este estudo, fornecido pela secretaria do orçamento e finanças do GDF em 10 de setembro de 2019, tem número de observações de 63787, e cada uma delas representa o menor lance de cada fornecedor para cada item. Esse banco de dados apresenta o mês do término do pregão, informações sobre o comprador e o fornecedor, qual edital do pregão, qual item, como é o produto, a quantidade demandada e o preço vencedor. Os produtos comprados, por exemplo, são subdivididos de forma hierárquica, sendo que a divisão inicial informa se o produto é de material ou de serviço, e é seguida de subdivisões em classes de objetos e serviços, daí por diante. Seguem abaixo as variáveis em estudo:

Tabela 2: Exemplos para Contextualizar as Variáveis

Variável	Exemplo
Mês_Resultado_Compra	Mai 2019
Número_Processo_Compra	05000006354/2018
Identif_Item_Compra	4501070500001201900001
Objeto_Compra	"Explica o item e o objetivo dele, textual"
Forma_Compra	SISRP
Tipo_Material_Serviço	Serviço
Grupo_Material_Serviço	SUPRIMENTOS AGRÍCOLAS
Classe	FORRAGENS E ALIMENTOS
Padrão_Desc_Material	ração animal
Mat_Serv_Mais_Específico	Ração animal, alimento industrializado de consumo animal
Nome_Fornecedor	AUTARQUIA COMERCIO E SAUDE ANIMAL LTDA
CPF/CNPJ_Fornecedor	07764000000107
Porte_Empresa_Fornec	Pequena Empresa
Valor_Unit_Homologado	220.00

Agora, cabe fazer uma explicação de algumas das variáveis.

- Mês_Resultado_Compra: mostra a data de conclusão do pregão, quando os itens são adjudicados ao licitantes vencedores;
- Modal_Compra_Grupo: indica se na compra foi usada a licitação pregão ou se não utilizou licitação. O último, considerado dispensa de licitação, acontece apenas em casos específicos, listados no art.24 da Lei 8.666. Nesse caso, apesar da desburocratização, deve seguir os mesmos princípios licitatórios, como igualdade e moralidade. Para este estudo será utilizada apenas a modalidade pregão. Na dispensa de licitação, a competição não é obrigatória;
- Forma_Compra é uma variável que retorna esses dois possíveis valores:
  - Sistema de Preços Praticados (SISPP), onde a administração escolhe um licitante pelo menor preço, mas deve seguir um contrato específico que, além de ser obrigado a fazer a compra, é obrigado a comprar a quantidade pré-estabelecida nele.
  - Sistema de Registro de preços (SISRP), onde a administração escolhe um licitante pelo menor preço, e decide se realiza a compra ou não. Além disso, a administração pode realizar outras compras do mesmo item, desde que seja com o vencedor do pregão [16, p. 242–253];

- Identif_Item_Compra

  Essa variável retorna todas as informações necessárias para encontrar o item de um determinado pregão. O valor da variável é uma concatenação de várias informações, por exemplo, se a observação for "45010705000022019000015", extraem-se as seguintes informações:

  - "450107": código relacionado à UASG responsável pela compra, que nesse caso é a Secretaria de Estadio de Segurança Pública;
  - "05": pregão (caso fosse "06", seria dispensa de licitação);
  - "000022019": indica que é o segundo pregão iniciado em 2019;
  - "00015": indica o item da compra.

Uma outra variável existente nesse conjunto de dados é "qtd_ofertada", mas uma análise dela mostrou diversas inconsistências, e por isso foi preciso coletar tal variável de outra forma.

Além da variável "qtd_ofertada", buscou-se extrair outros tipos de informação para agregar ao estudo e possibilitar atingir os objetivos listados anteriormente, para isso foi necessário baixar dezenas de atas em PDF, no site **comprasnet**. Delas, será feita a mineração de dados, abordada na subseção 2.2.1.

3.2 Mineração para o Banco de Dados

Primeiramente, foi utilizado o comando *pdf_text* do *R* para ler uma ata qualquer, por exemplo a **22019**, para treinar o algoritmo de coleta de dados. Essas atas são baixadas no site **comprasnet**.

A partir dessa ata, buscou-se padrões na organização dos dados para extrair todas as variáveis de interesse. Abaixo, na Figura 7, tem-se uma captura de tela de uma parte de bastante interesse para este trabalho:

Figura 7: Exemplificação do PDF

Valor do Lance	CNPJ/CPF	Data/Hora Registro
R$ 1.000,0000	21.822.463/0001-09	28/01/2019 09:33:07:493
R$ 500,0000	28.128.604/0001-37	28/01/2019 09:33:07:493
R$ 452,8000	04.896.000/0001-72	28/01/2019 09:33:07:493
R$ 300,0000	13.464.349/0001-26	28/01/2019 09:33:07:493
R$ 261,0700	17.451.234/0001-58	28/01/2019 09:33:07:493
R$ 260,0000	25.041.538/0001-75	28/01/2019 09:33:07:493
R$ 259,7900	28.128.604/0001-37	28/01/2019 09:49:10:193
R$ 250,0000	25.041.538/0001-75	28/01/2019 09:51:14:090
R$ 240,0000	13.464.349/0001-26	28/01/2019 09:52:54:427
R$ 180,0000	25.041.538/0001-75	28/01/2019 10:03:57:107
R$ 179,9000	13.464.349/0001-26	28/01/2019 10:05:08:270
R$ 259,7800	21.822.463/0001-09	28/01/2019 10:09:34:473
R$ 160,0000	25.041.538/0001-75	28/01/2019 10:09:47:800
R$ 150,0000	13.464.349/0001-26	28/01/2019 10:10:48:727
R$ 184,4500	17.451.234/0001-58	28/01/2019 10:14:52:413
R$ 184,0000	28.128.604/0001-37	28/01/2019 10:25:18:947

Não existem lances de desempate ME/EPP para o item

Eventos do Item

Evento	Data	Observações
Aberto	28/01/2019 09:48:00	Item aberto.
Iminência de Encerramento	28/01/2019 10:06:47	Batida iminente. Data/hora iminência: 28/01/2019 10:08:47.
Encerrado	28/01/2019 10:28:11	Item encerrado
Aceite	28/01/2019 15:42:15	Aceite individual da proposta. Fornecedor: N3 COMERCIO E SERVICOS EIRELI, CNPJ/CPF: 13.464.349/0001-26, pelo melhor lance de R$ 150,0000.
Habilitado	28/01/2019 15:59:53	Habilitação em grupo de propostas. Fornecedor: N3 COMERCIO E SERVICOS EIRELI - CNPJ/CPF: 13.464.349/0001-26

Por essa parte, extraem-se diversas informações tanto sobre os lances, quanto sobre o item em questão. Além de capturar esses valores, também é feita a limpeza dos dados, por exemplo, o valor do lance original como "$R\$1.000,0000$" está salvo com precisão demasiadamente alta e contém caracteres especiais, então transforma-se para "1000". Já para o CNPJ, excluem-se todos os símbolos que não ajudam a identificar o fornecedor, como barras, pontos e hífens. Para fazer a limpeza e a localização dos dados, utilizam-se técnicas de expressões regulares, e para tal utilizou-se o pacote *stringr* do *software* R.

Além disso, é possível, na Figura 7, identificar também o lance vencedor e os horários de abertura e fechamento.

Para a coleta de dados referentes aos lances, por exemplo, necessitam-se de duas *strings* comuns a todos os PDFs e relacionadas aos momentos de início e término de lances. Pela Figura 7 dá para vizualizar que as *strings* "Valor do Lance" e "Eventos do item" são indicadas para fixar a leitura do arquivo. Porém, algumas particularidades de cada PDF faz com que nem sempre seja confiável tal escolha, assim, é necessário fazer uma limpeza nas atas antes de rodar o algoritmo.

Entre as sujeiras que prejudicam o desempenho da automação, tem-se itens cancelados, itens sem lances, sequência de grupos de itens, itens com existência de empates, e particularidades de cada PDF, como a troca de página.

Em outras regiões do PDF encontram-se também várias outras informações passíveis de serem extraídas.

Para melhorar o processo de automação, utilizou-se outras atas para agregar novos padrões aos códigos, a fim de generalizar para qualquer outra ata a ser colocada de entrada. Assim, a função criada extrai diversas informações de cada arquivo, e as junta em uma tabela só, que é posteriormente agregada ao banco de dados inicial.

Além de obter diversas novas variáveis, a mineração também retorna todos os lances de cada fornecedor, lembrando que o banco inicial informava apenas o menor lance deles. Porém, apenas os lances vencedores serão usados, pois são suficientes para a conclusão dos objetivos encontrados na subseção 1.

Caso o leitor queira analisar detalhadamente como o algoritmo funciona, os códigos em R estão disponibilizados no apêndice 7.1.

3.3 Banco de Dados Final

As variáveis mineradas, e incluídas no banco de dados original (seção 3.1), foram:

- valor estimado – preço de referência previsto para cada item;
- valor dos lances – valor de cada um dos lances propostos pelos fornecedores;
- menor lance – o menor lance recebido pelo item;
- n_lances – número de lances que cada item teve;
- p_lances_min – razão entre o lance vencedor do item pelo lance mínimo do respectivo item. Valores maiores do que 1 significam que houve algum lance menor que o vencedor, mas por algum motivo não foi homologado. Podem-se detectar eventuais coelhos com essa variável;
- p_lances_est – razão entre o lance da linha pelo valor estimado do respectivo item;
- inicio – horário do primeiro lance dado ao respectivo item;
- hora do lance – horário em que o lance vencedor foi lançado;

- horário de abertura – horário em que o pregão foi aberto para iniciar a fase de lances;
- horário de fechamento – horário em que a fase de lances foi encerrada;
- qte_ofertada – quantidade de um mesmo item que a UASG comprou. Caso seja SISPP, essa quantidade é pré-estabelecida no edital, e o "valor do lance" é o preço total da compra da quantidade ofertada. Porém, caso seja SISRP, o algoritmo de mineração de dados deve ficar atento, pois o "valor do lance" nesse caso é unitário. Sabendo dessas particularidades, pode-se saber o gasto total do governo para qualquer item.

O número total de atas do banco de dados original é 758, e foram baixadas arbitrariamente 29 delas para o exercício deste trabalho.

Assim, o número de observações do banco de dados utilizado para esse trabalho é de 1288, e corresponde ao total lances homologados na amostra não probabilística de 29 pregões coletados.

3.4 Planejamento do Projeto

Inicialmente, é oportuno conduzir um estudo exploratório dos dados minerados, com a apresentação de gráficos e tabelas que ajudam a entender o comportamento da base de dados. É dessa análise que surgirá a base de conhecimento para a melhor tomada de decisões na parte de modelagem, e também será essencial para saber quais são as questões mais importantes para este estudo. Além disso, uma análise descritiva bem feita permite que a motivação para o uso das técnicas estatísticas seja clara e coesa.

Analisar os dados também indicará boa parte dos outliers, assim a análise exploratória é essencial para o objetivo de detecção de anomalias e, consequentemente, o cumprimento dos objetivos listados na introdução deste trabalho.

Após a análise exploratória, serão aplicadas técnicas avançadas de estatística, como modelos de regressão beta, para prever a distância do lance para o preço estimado dado o item e o tipo de licitante. Também será usado o algoritmo de classificação *k-means* para agrupar dados e melhorar o desempenho do modelo. Com o intuito de detectar anomalias no processo licitatório, será utilizada uma técnica chamada *Isolation Forest* em conjunto com um modelo de regressão beta. Essa técnica isolará os pontos mais discrepantes, e fornecerá um valor indicativo de *outlier*.

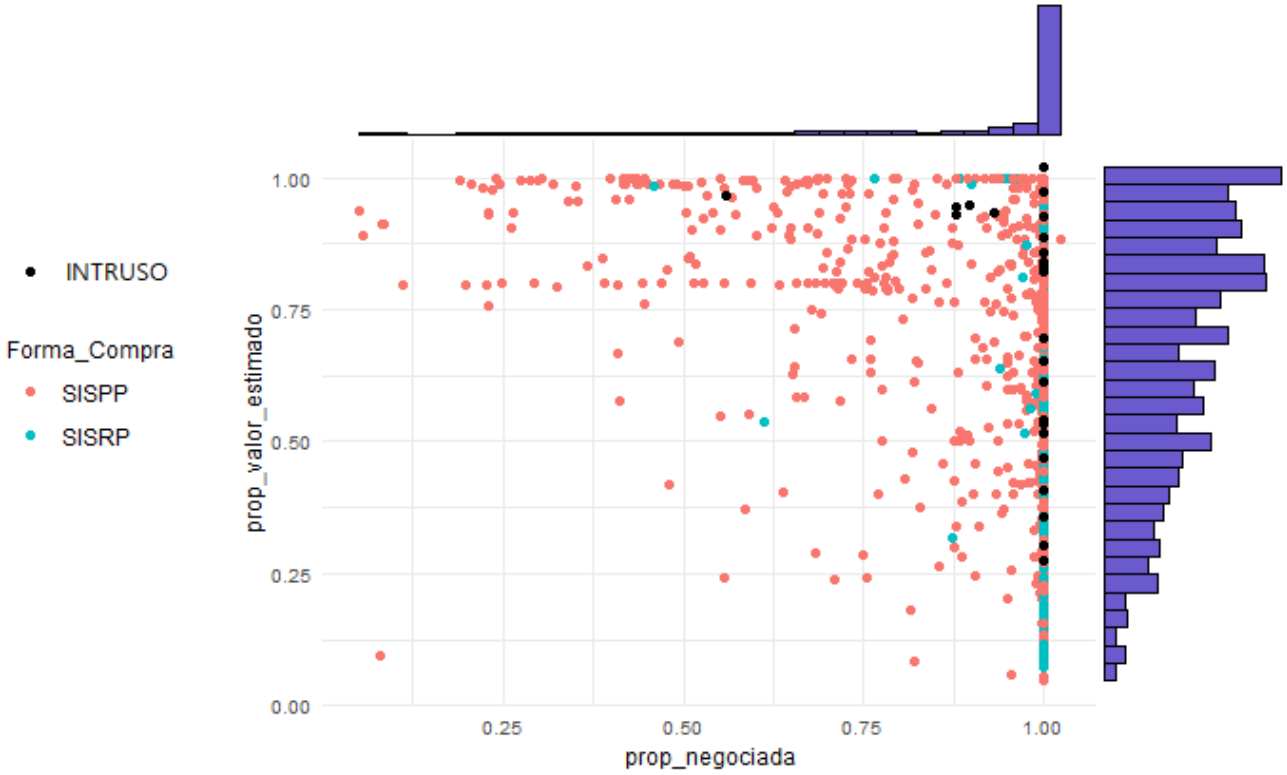
4 Análise Exploratória

Esta seção apresentará as relações entre variáveis que serão úteis na etapa da modelagem preditiva. Espera-se, também, que sejam encontradas algumas anomalias evidentes.

Agora, será feita a análise da variável prop_valor_estimado, que representa o resultado da divisão do lance vencedor pelo valor estimado e, portanto, indica o quão próximo o lance vencedor é do valor estimado de cada item.

Também analisou-se a variável prop_negociada – razão entre os valores antes e após as negociações.

Figura 8: Variáveis de negociação e Valor Estimado



É possível extrair bastante informação da Figura 8. Cabe dizer, por exemplo, que a maior parte dos dados apresenta prop_negociada de valor 1, ou seja, não houve negociação após o item ser aceito. Ainda sobre essa variável, notam-se alguns pontos com valores muito baixos, o que indica uma negociação exageradamente elevada, possível de ser algum engano. Sobre a variável prop_valor_estimado, ela é distribuída com mais variância, pois vemos muito mais pontos abaixo de 1 do que na outra variável, sendo os valores mais próximos de 1 os mais comuns.

Outra informação que se tira da Figura 8, representada pelos pontos pretos no gráfico de dispersão, são os lances que, apesar de serem vitoriosos, não poderiam ser aprovados, pois não estiveram na fase de abertura, logo não deveriam ter avançado para a fase de lances. Esses lances, chamados de intrusos, serão removidos do banco de dados para a etapa de modelagem.

O gráfico também indica uma possível relação entre a forma da compra e prop_valor_estimado. Sobre isso, compras utilizando SISRP parecem ter sido feitas, em geral, com preço abaixo do estimado, pois sua maioria está abaixo da metade do seu valor estimado para o item, ao passo que em compras utilizando o SISPP isso não parece ocorrer. O gráfico também revela a existência de

pontos que tiveram valor acima de 1 para as variáveis prop_negociada ou prop_valor_estimado.

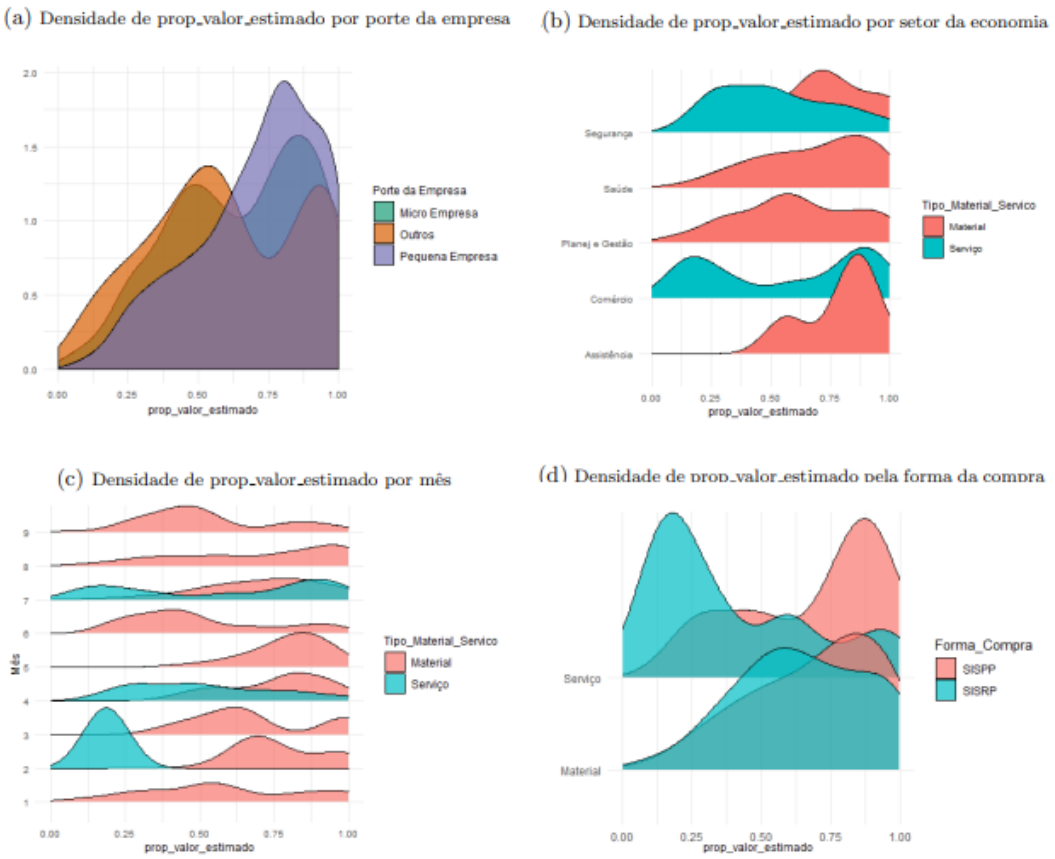
Assim, a Figura 8 mostra três situações inesperadas:

- Vinte e cinco itens, que venceram o pregão mas não deveriam ser autorizados, pois não foram aprovados na fase de abertura, chamaremos esses itens vencedores de intrusos;
- Um lance que foi aceito acima do valor estimado com valor acima de 1 de prop_valor_estimado, contrariando as normas gerais dos pregões. Coincidentemente, este item era intruso;
- Um item que aumentou de valor após a negociação com o governo. O que não faz sentido, pois o governo quando negocia tem como objetivo diminuir o preço.

A Figura 9 mostra a relação entre a variável prop_valor_estimado, que está no eixo x de todos os quatro gráficos, e algumas outras variáveis. O eixo Y é o valor da densidade, ou seja, deve integrar um. Assim, esses gráficos mostram como as variáveis se comportam, ignorando o tamanho da classe. A Figura 9a mostra a diferença dos picos de cada tipo de empresa, sendo que pequenas empresas parecem fechar negocio com preços próximos ao de referência.

A Figura 9c mostra que não parece haver tendência temporal nos dados, apesar de claramente uns meses se comportarem de forma diferente de outros. A Figura 9b evidencia como o comportamento de prop_valor_estimado é diferente para os diversos setores da economia. A Figura 9d mostra que a forma da compra tem comportamento diferente quando separada entre material e serviço, sendo que, em ambos os casos, SISPP apresenta tendência de possuir menor valor de prop_valor_estimado do que SISRP.

Figura 9: Variáveis de negociação e Valor Estimado



5 Modelos e técnicas para os Dados

5.1 Técnicas para melhoria dos dados

Uma das ações que foram usadas para melhorar a qualidade do banco é o agrupamento de algumas variáveis, algumas de forma manual e outras utilizando algoritmos automatizados. Por exemplo, a variável "Grupo_Material_Servico", que indica características do item, originalmente tinha 46 classes, um número pouco interpretativo, sendo pouco útil para um modelo de previsão ou classificação dos dados. Além disso, problemas de *overfitting* também podem surgir ao incluir variáveis tão específicas. Dessa forma, o algoritmo *K-means*, explicado na subseção 2.2.2, foi utilizado para reduzir o número de classes dessa variável para oito. Isso permitiu um aumento no número de observações em cada classe, tornando o estimador do modelo Beta possível de ser calculado. Também agrupamos as Unidades Administrativas de Serviços Gerais, passando de sete para cinco classes, relacionadas ao o setor econômico que cada classe pertencia: planejamento e gestão, segurança, saúde, comércio e assistência social.

5.2 Modelo Beta para Ajustar aos Dados

Nesta seção, será formulado um modelo de regressão para a variável prop_valor_estimado, chamada nesta seção de Y, com base em informações anteriores à fase de abertura do pregão, ou seja, informações sobre o item, a empresa fornecedora e a forma da compra. A expectativa é que o modelo entenda o comportamento dessa variável e possivelmente faça previsões. Sabendo que a variável resposta escolhida mede o quociente entre o valor homologado e o valor estimado, conclui-se que ela pertence ao intervalo $[0, 1]$, a exceção de um exemplo em que o valor vencedor é acima do estimado, que foi retirado para fins de modelagem. Assim, considera-se plausível utilizar o modelo de regressão beta. Nesse modelo, assume-se que a variável resposta pertença a ao intervalo unitário, porém não aceitam-se valores iguais a um nem a zero. Por isso, precisou-se fazer a seguinte transformação em prop_valor_estimado – Y, encontrada em (Smithson et al., 2006)[15]:

$$Y^* = \frac{Y(n-1) + 0.5}{n}.$$

As covariáveis possíveis de serem usadas são: a forma da compra, o setor da economia em que o gasto é feito, o porte da empresa fornecedora, e o tipo do item.

Primeiro, será utilizada a função *betareg* para uma regressão beta, que utiliza conceitos clássicos de verossimilhança para a estimação dos parâmetros μ_i e ϕ. E será testado o resultado para diferentes funções de ligação $g(\mu)$.

A fórmula usada é tratada a seguir:

$$\hat{\eta} = g(\mu) = \hat{\boldsymbol{\beta}}_{\mathbf{0}} + (\text{Porte\_empresa})\hat{\boldsymbol{\beta}}_{\mathbf{1}} + (\text{Tipo\_Objeto})\hat{\boldsymbol{\beta}}_{\mathbf{2}} + (\text{Setor})\hat{\boldsymbol{\beta}}_{\mathbf{3}} + (\text{Forma\_Compra})\hat{\boldsymbol{\beta}}_{\mathbf{4}}, \tag{15}$$

sendo $\hat{\boldsymbol{\beta}}_{\boldsymbol{i}}$, com $i = 1, \cdots, n$, o conjunto de parâmetros da covariável i. A Fórmula (15) então retorna o preditor linear η para cada observação e, ao aplicá-la à função de ligação inversa, obtém-se a aproximação para a variável resposta, denotada por $\hat{Y}^*$.

Embora a função de ligação da função *betareg* seja, por *default*, a canônica *logit*, é prudente analisar o modelo com outras, a fim de se perceber se vale a pena trocar a função de ligação.

Tabela 3: Comparação das Diferentes Funções de Ligação

Métricas	*logit*	*probit*	*cloglog*	*cauchit*	*loglog*
AIC	-679.694	-679.377	-678.607	-681.573	-680.209
MSE	0.052	0.052	0.052	0.052	0.052
$RMSE$	0.229	0.229	0.229	0.229	0.229
MAE	0.190	0.190	0.190	0.190	0.190
md_erros	0.177	0.177	0.177	0.176	0.176
log_verossimilhanca	355.847	355.688	355.303	356.787	356.104
ϕ	2.560	2.559	2.556	2.566	2.561
iterações	26	25	25	449	26
Pseudo R^2	0.065	0.072	0.089	0.047	0.057

Sabe-se que, além de apresentar muitos pontos anômalos, a variável prop_valor_estimado apresenta alta variância. Dessa forma, tendo como base a explicação da subseção 2.3.4, a métrica mais adequada para medir o quanto que o modelo erra é o MAE, e os modelos apresentaram um baixo desempenho para essa métrica (0.19,Tabela 3), já que esse valor indica uma distância média de 19% entre o valor real e o predito.

Outra métrica que indica um baixo desempenho no ajuste do modelo é o Pseudo R^2 próximo de 0, indicando que as covariáveis utilizadas não foram suficientes para explicar a variação de "prop_valor_estimado".

Note que o valor da log-verossimilhança, otimizado pela função *optim* do R, também é semelhante em todos os modelos e, consequentemente, o AIC também não distingue os modelos. Um fator que chamou atenção foi o número de iterações que o modelo utilizando *cauchit* precisou para encontrar o ponto de máximo da sua função de log-verossimilhança.

Conclui-se que alterar as funções de ligação não implica uma mudança significativa na qualidade do ajuste do modelo, por esta razão a função *logit* não será trocada.

Porém, é relevante entender se esses modelos apresentariam desempenhos parecidos caso fossem executados em dados desconhecidos. Para isso, em conjunto com técnicas de validação, utiliza-se uma seleção de variáveis, a fim de diminuir a variância das estimativas e estabilizar as predições, sem prejudicar a acertividade. Dessa forma, é possível encontrar uma boa previsão com um número de parâmetros menor, melhorando o valor do AIC.

Assim, será colocada em prática uma outra forma de modelar, que, diferentemente das anteriores, modela o parâmetro ϕ conjuntamente com o parâmetro μ. Além disso, a estimação dos parâmetros é feita por **regularização *boosted***, resultando em uma seleção de variáveis. A função utilizada é o *glmboostLSS*, do pacote *gamboostLSS*. Para a implementação, foi utilizada a mesma fórmula anterior (Equação 15), tanto para μ quanto para ϕ, com funções de ligação *logit* e *log*, respectivamente.

Com o modelo criado, precisa-se escolher o critério de parada, o qual retorna a melhor capacidade preditiva do modelo possível. Para isso, o pacote gamboostLSS fornece uma função que faz validação cruzada com 25 partições de igual tamanho (25-*Fold Cross Validation*) da função predita log-verossimilhança para mensurar a complexidade ótima do modelo, e retorna o número ótimo de iterações para cada parâmetro, minimizando o risco preditivo.

As cores mais escuras das linhas na figura 10b indicam os maiores riscos preditivos calculados na validação cruzada. Assim, o número ótimo de iterações é 94 para μ e 36 para ϕ, retornando um risco preditivo de -356. Note que, a partir da iteração 36, o parâmetro ϕ é deixado como constante, e apenas o μ continua iterando, como mostra a figura 10a, que também mostra o desenvolvimento de quase todos os parâmetros ao passar das iterações.

Figura 10: As Iterações *Boosted*

(a) Iterações para os Parâmetros μ e ϕ (b) Número Ótimo de Iterações

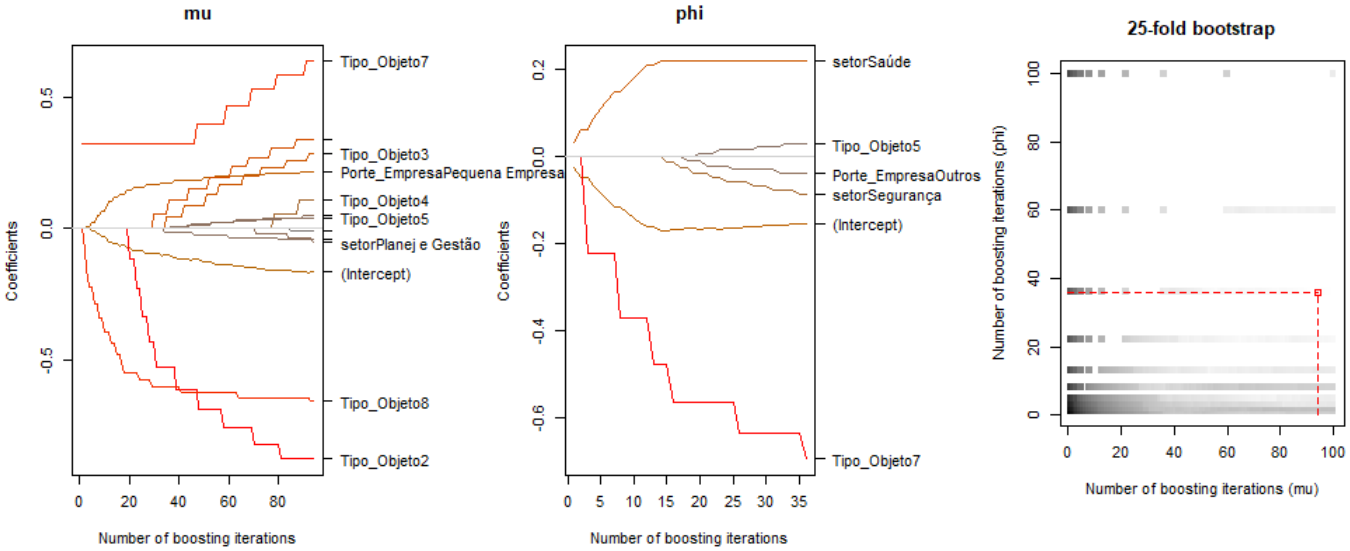


Tabela 4: Coeficientes Estimados pela Regressão *Boosted*

Fatores para μ	Coeficientes μ	Fatores para ϕ	Coeficientes ϕ
(Intercepto)	-0.166	(Intercepto)	-0.155
Porte-EmpresaPequena Empresa	0.216	Porte_EmpresaOutros	-0.039
Tipo_Objeto2	-0.877	Tipo_Objeto5	0.028
Tipo_Objeto3	0.287	Tipo_Objeto7	-0.695
Tipo_Objeto5	0.042	setorSaúde	0.220
Tipo_Objeto6	0.336	setorSegurança	-0.087
Tipo_Objeto7	0.640		
Tipo_Objeto4	0.110		
Tipo_Objeto8	-0.657		
setorPlanej_e_Gestão	-0.052		
setorSegurança	0.048		
Forma_CompraSISRP	-0.043		
Porte_EmpresaOutroas	-0.007		

Repare, na Tabela 4, que alguns fatores são relevantes não só para estimar a média μ mas também para estimar a precisão ϕ, a exemplo do "setorSegurança".

Já a Tabela 5 mostra como o *AIC* do modelo *boosted* melhorou levemente em relação ao modelo usual, pois houve uma redução de parâmetros (p) para estimar μ e um aumento no valor máximo da log-verossimilhança.

Tabela 5: Comparação do Modelos Sem e Com Regurarização

Modelo	p	log-verossimilhança	*AIC*
Modelo beta sem regularização	15	355.848	-679
Modelo *Boosted* Beta	12	356.2432	-687

Agora, serão analisados os resíduos do modelo *boosted*, a fim de se observar os maiores erros e entender o porquê dos mesmos, contribuindo com o objetivo de detecção de pontos anômalos.

Tabela 6: Os Dez Maiores Resíduos

Identif_Item_Compra	prop_negociada	prop_valor_estimado	preditos	residuo
9742000500068201900036	0.079	0.096	0.676	0.580
9742000500071201900062	0.999	0.141	0.722	0.581
9266370500009201900036	0.998	0.134	0.727	0.593
9742000500071201900063	0.999	0.128	0.722	0.594
9742000500186201900007	0.954	0.058	0.665	0.607
9742000500071201900013	1.000	0.054	0.665	0.611
9742000500113201900061	0.821	0.085	0.712	0.627
9742000500113201900062	0.999	0.085	0.712	0.627
9742000500071201900061	1.000	0.047	0.676	0.630
9742000500186201900008	1.000	0.053	0.712	0.659

Note, na Tabela 6, que há uma característica evidente nesses itens: eles foram adjudicados com valores muito abaixo do estimado. Uma possível causa para isso é o valor de referência estabelecido no edital para o determinado item ser muito alto. Outras possibilidades se referem a erro humano ou de sistema, e é de suma relevância investigar esses casos individualmente nas atas para entender o ocorrido.

5.3 *Isolation Forest* Para Detectar Anomalias

Partindo para a vertente de detecção de anomalias, apesar do resíduo do modelo de regressão *boosted* beta ajudar um pouco na detecção de alguns casos anormais, sabe-se que há outras variáveis que, quando atingem determinado valor, apresentam indícios de anomalia, como o tamanho da negociação. O objetivo é selecionar essas variáveis para um modelo de detecção de anomalias multivariado.

Como vimos na análise exploratória, temos inúmeras formas de um determinado item estar fora do esperado. É estranho, por exemplo, uma negociação que exceda 80%. Um valor negociado superior ao valor anterior também não é intuitivo, pois espera-se que o governo negocie um valor para baixo. Além disso, um valor homologado muito abaixo do estimado pode ser causado por uma superestimação do preço do item, e pode indicar também que a empresa tenha fornecido um produto com padrão de qualidade abaixo do aceitável.

Outro acontecimento inesperado é a presença de lances não vencedores muito menores que o lance vencedor em um determinado item. É importante analisar cuidadosamente estes casos, pois representam um dos indícios de coelho.

Todas essas possibilidades foram extraídas automaticamente das atas, resultando em variáveis importantes na detecção de anomalias as quais serão incluídas no algoritmo. Seguem abaixo as variáveis utilizadas:

- "autorizado": indica se o fornecedor vencedor realmente poderia participar da fase de lances, ou seja, se ele foi aprovado na fase de abertura do pregão;
- "prop_negociada": indica o quanto que o valor negociado é menor que o valor original;
- "prop_valor_estimado": indica o quanto que o valor vencedor é menor que o valor de referência;
- "acima_do_estimado": indica se o valor vencedor é maior do que o valor de refêrencia;
- "negociado_acima": indica se o valor negociado é maior que o valor original;

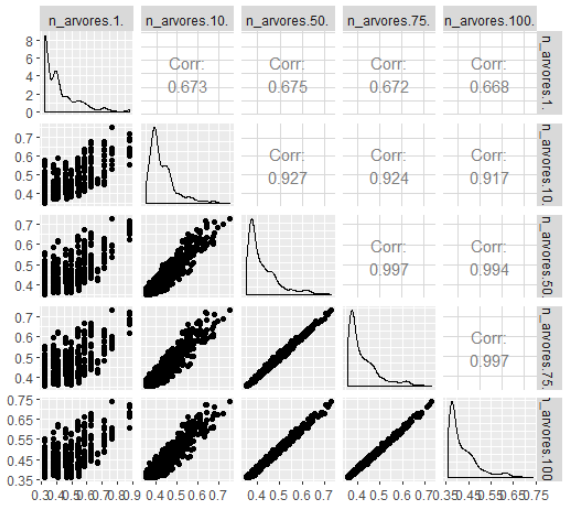
- "p_lances_min": indica se houveram lances muito menores que o vencedor, mas por algum motivo não foram aceitos.

Com o intuito de fazer um **meta-aprendizado** do tipo *boosted*, outra variável utilizada para o *Isolation Forest* representa o conjunto de resíduos do modelo beta anterior. O motivo disso é que tal modelo, para estimar prop_valor_estimado, utilizou algumas variáveis capazes de agregar a procura de anomalias. Assim, o resíduo do modelo beta contribuirá com a detecção de outros tipos de anomalias. Como sabemos que os resíduos mais altos estão fora da grande massa de resíduos, eles serão reconhecidos como valores atípicos.

Com as variáveis selecionadas, é hora de implementar a Floresta de Isolamento em si. Como tratado na seção de literatura, sabemos que quanto menor o caminho, ou seja, menor o número de passos para terminar o isolamento de uma determinada observação, mais atípica ela é. Além disso, também é claro que quanto mais árvores, mais preciso e robusto é o modelo. Assim, o usual é procurar um número de árvores suficientemente grande para produzir um bom modelo, mas não tão grande a ponto de tornar o algoritmo computacionalmente caro.

Como tratado na seção 2.2.4, a função utilidada para rodar o algoritmo retorna uma métrica para cada observação denominada *score*. A Figura 11 explicita a correlação dos *scores* calculados por Florestas de Isolamento de diferentes números de árvores. Nota-se uma correlação altíssima entre o caso de 50 árvores e o de 75 casos, isso indica que o algoritmo com 50 árvores já havia convergido suficientemente bem e, portanto, será o número usado para a Floresta de Isolamento.

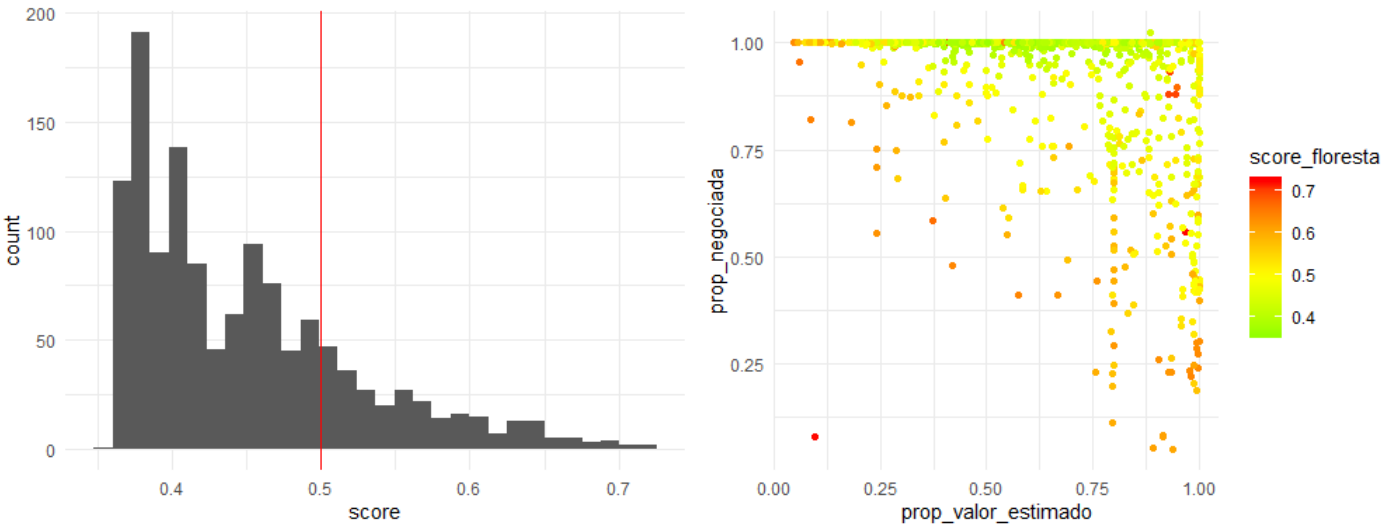
Figura 11: Convergência na Alteração do Número de Árvores



Determinados o número de árvores e as variáveis, o *score* de cada observação é incluído no banco de dados. O maior valor foi de 0.76, considerado o maior *outlier*, ao passo que o menor valor foi de 0.36.

Figura 12: *Score* da *Isolation Forest*

(a) Histograma do *Score* (b) *Score* Segundo Variáveis



A linha vermelha, na Figura 12a, é onde separa as observações com *scores* acima de 0.5, que, como visto na seção 2.2.4, são mais anômalas do que o normal, e indicam itens com algo de estranho no valor de alguma variável. Essa análise pode ser feita na Tabela 7, onde mostra os dez itens mais anormais considerados pelo algoritmo.

A Figura 12b mostra como valores baixos das duas variáveis tendem a ter um *score* alto e ser tratado como *outlier*. Além disso, os pontos que, mesmo com valores altos de prop_valor_estimado e prop_negociada, são vermelhos, indicam outras anomalias no processo, como a presença de um lance bem mais baixo que o vencedor que por algum motivo não foi aceito. Esse gráfico também mostra a enorme quantidade de itens com sobrepreço, valor baixo de prop_valor_estimado, visto que o preço estimado nestes casos é muito acima do preço vencedor.

Tabela 7: Os Dez Maiores *scores*

Identif_Item_Compra	p_lances_min	prop_valor_estimado	prop_negociada	residuo	*score*
9742000500071201900061	1.066	0.047	1.000	0.630	0.637
9742000500232201800019	6.176	0.999	0.240	0.327	0.638
9742000500232201800020	5.095	0.997	0.275	0.321	0.638
9742000500036201900021	1.044	0.419	0.479	0.293	0.643
9742000500232201800022	4.708	0.980	0.221	0.304	0.643
9742000500040201900033	2.436	0.576	0.410	0.136	0.644
9742000500113201900061	1.000	0.085	0.821	0.627	0.648
9742000500186201900007	1.000	0.058	0.954	0.607	0.662
9742000500071201900057	4.325	0.372	0.585	0.350	0.666
9742000500068201900036	1.000	0.096	0.079	0.580	0.724

A Tabela 7 mostra para o investigador os dez lances vencedores mais inesperados junto às variáveis que apontaram suas anomalias. Por exemplo, o item "9742000500071201900057" tem o lance vencedor com o segundo maior *score*, e apresenta valor 4.325 de p_lances_min, o que indica, portanto, a existência de um lance quatro vezes menor que o lance vencedor. Esse lance

é provavelmente um coelho, que, na hora de vencer, não entregou os documentos de forma proposital. Para confirmar isso o investigador deve olhar a ata e verificar o motivo da não aceitação.

O item considerado mais anormal é o "97420005000682019000036", no qual encontram-se valores absurdos de prop_valor_estimado e prop_negociada. A análise desse item no PDF permite mostrar que houve um erro de digitação que fez o pregoeiro pagar dez vezes menos do que pagaria inicialmente. O motivo pelo qual o fornecedor aceitou isso também é do trabalho do investigador.

6 Conclusão

Este trabalho mostra que é possível detectar anomalias em pregões eletrônicos de forma prática, utilizando não só técnicas robustas como regressão e *isolation forest*, mas também a análise gráfica. É um trabalho que se mostra útil para os órgãos investigadores, pois com ele foi possível encontrar lances homologados inesperados, negociações exageradas, erros de digitação etc.

A mineração de dados se manifestou como uma técnica bastante eficiente de coleta de informações, além de se mostrar importante na realização da validação e da reconstrução de dados.

Foi também apresentado um modelo de previsão, porém, este apresentou baixo desempenho na predição justamente pela característica de alta variância dos dados, a qual as covariáveis usadas não foram suficientes para explicar. Além disso, como a amostra utilizada não é probabilística, não representa a população de atas de pregões eletrônicos, e deve ser usada com cuidado.

Como visto, a mineração de dados permite coletar qualquer informação das atas. Porém, no momento há a limitação no *download* de PDFs do **comprasnet**, pois eles possuem um esquema de segurança, o CAPTCHA, tornando lento o processo de obtenção de PDFs, motivo pelo qual não foram coletadas todas as atas para a realização desse projeto. Para resolver esse problema, existe um pacote no R chamado *decryptr*, capaz de quebrar CAPTCHA, tornando possível uma leitura automática no **comprasnet**, facilitando assim a coleta de todas as atas existentes.

Outras frentes que podem ser estudadas são:

- Coletar todos os lances, e a partir deles fazer uma análise de comportamento temporal. Com esses lances, é possível também analisar a diferença entre os horários dos lances para detectar robôs, que tem a característica de ofertar lances com intervalos de tempo ínfimos, na casa dos milisegundos [4](Silva et al.);

- Encontrar e identificar todos os coelhos, que são recusados por não apresentarem a documentação exigida. Como o motivo da recusa está na ata, uma atualização da ferramenta de mineração localizaria esse motivo;

- Apurar se os lances que foram feitos após o fechamento, poderiam ser homologados ou não. Para isso, a ferramenta de mineração de dados deve ser atualizada para detectar se houve uma convocação especial ou reabertura justificada da fase de lances;

- Contabilizar os itens cancelados, e investigar os motivos desses cancelamentos. Segundo [11](Neves et al., 2020), entre outros fatores, os cancelamentos podem ocorrer por falta de propostas na fase de abertura, ausência de lances dentro dos conformes previstos no edital, recusa do licitante em negociar e desistência do licitante;

- Automatizar a formulação dos modelos de forma que, dado como entrada um conjunto de atas qualquer, forneça automaticamente uma saída com os lances mais anômalos. Esse processo pode ser feito com a utilização do *R markdown*.

Referências

[1] Gauss Moutinho Cordeiro and Clarice GB Demétrio. Modelos lineares generalizados e extensões. *Piracicaba: USP*, page 31 e 124, 2008.

[2] Zhiguo Ding and Minrui Fei. An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window. *IFAC Proceedings Volumes*, 46(20):12–17, 2013.

[3] Isabela Neves Drummond. Implementação do método de classificação contínua fuzzy k-médias no ambiente terralib. *Monografia final do curso de Introdução ao Geoprocessamento. Inpe. São José dos Campos*, page 10, 2003.

[4] SILVA e PERENHA. Fraudes no pregão eletrônico: Como combater? *Trabalho de Conclusão de Curso*, page 5.

[5] Paulo C EMILIANO, Elayne P Veiga, Mário JF Vivanco, and Fortunato S Menezes. Critérios de informaçao de akaike versus bayesiano: análise comparativa. *19º Simpósio Nacional de Probabilidade e Estatística*, page 2, 2010.

[6] João Gama. Árvores de decisão. *Palestra ministrada no Núcleo da Ciência de Computação da Universidade do Porto, Porto*, 2002.

[7] Benjamin Hofner, Andreas Mayr, and Matthias Schmid. gamboostlss: An r package for model building and variable selection in the gamlss framework. *arXiv preprint arXiv:1407.1774*, 2014.

[8] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422. IEEE, 2008.

[9] Andreas Mayr, Nora Fenske, Benjamin Hofner, Thomas Kneib, and Matthias Schmid. Generalized additive models for location, scale and shape for high dimensional data—a flexible approach based on boosting. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 61(3):403–427, 2012.

[10] Tom M Mitchell and Machine Learning. Mcgraw-hill science. *Engineering/Math*, 1:27, 1997.

[11] Meryellem Yokoyama Neves and Rafael Pereira Ocampo Moré. Pregão eletrônico: um estudo das causas de cancelamento de itens no âmbito de uma universidade federal. *Revista do Serviço Público*, 71(1), 2020.

[12] OLIVEIRA. Pregões eletrônicos : Suas aplicações, vantagens e temas polêmicos. *Monografia- Faculdade de Direito da Universidade de Brasília (UnB). 2016.*, page 39.

[13] Santana. Validação cruzada: Aprenda de forma simples como usar essa técnica. 2020.

[14] Ferrari SLP and Cribari-Neto F. Beta regression for modelling rates and proportions. *Journal of Statistical Software*, pages 1–6, 2004.

[15] Michael Smithson and Jay Verkuilen. A better lemon squeezer? maximum-likelihood regression with beta-distributed dependent variables. *Psychological methods*, 11(1):54–71, 2006.

[16] TCU. *TCU, Licitações Contratos – Orientações e Jurisprudências do TCU*, 2010.

[17] Achim Zeileis, Francisco Cribari-Neto, Bettina Gruen, Ioannis Kosmidis, Alexandre B Simas, Andrea V Rocha, and Maintainer Achim Zeileis. Package 'betareg'. *R package*, 2016.

7 Apêndice

Pacotes utilizados para o trabalho:

```
library(pacman)
p_load(dplyr, tidyr, maditr, DT, tidyverse, magrittr, ggplot2, readxl,
       lubridate, stringr, knitr, pdftools, GGally, betareg, isofor,
       viridis, metR, viridisLite, gghighlight, ggridges, FactoMineR,
       cluster, factoextra, xlsx, gridExtra, RColorBrewer, gamboostLSS,
       boot, ggExtra)
```

7.1 Mineração de Dados

Ler os pdf do site do governo:

```

@ Dados do governo com a limpeza feita

dados <- readRDS(file = "dados.rds")


@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Leitura das atas
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

@ funcao para coletar variaveis importantes:
mining_est_lan <- function(arquivos)
  {

  tabelao <<- list()
  lances_por_item <<- list()
  tabi <<- list()

  for(k in 1: length(arquivos)){
    arquivo <- pdf_text(arquivos[k])%>%  @Ler o i-esimo arquivo do conjunto
        escolhido
      readr :: read_lines()


    @ Pode haver itens sem propostas (lances), removeremos do banco:

    sem_lance <- grep("Situacao: Cancelado por inexistencia de prop", arquivo)
        @exclui o valor estimado
    sem_lances <- c(sem_lance, sem_lance - 1)

    a <- grep("Valor estimado" , arquivo)
    arquivo[309]
    itens_sem_proposta <- match(sem_lance,a)

    if(length(sem_lance) != 0)
      {
            arquivo <- arquivo[-sem_lances] @ para excluir esses itens na
                primeira parte do pdf
      }




    @lixo1 ("comprasnet.gov.br")
```

```
42
43      linhas_tabela_limpeza2 <- grep("comprasnet.gov.br", arquivo)
44      arquivo[linhas_tabela_limpeza2]
45
46      if(length(linhas_tabela_limpeza2) != 0)
47        {
48              arquivo <- arquivo[-linhas_tabela_limpeza2]
49        }
50
51      linhas_tabela_limpeza3 <- grep("COMPRASNET - O SITE DE COMPRAS DO GOVERNO",
            arquivo)
52      arquivo[linhas_tabela_limpeza3]
53
54      if(length(linhas_tabela_limpeza3) != 0)
55        {
56              arquivo <- arquivo[- linhas_tabela_limpeza3]
57        }
58
59
60
61      sem_proposta_inicio <- grep("Nao existem propostas",arquivo)
62      sem_proposta_fim <- sem_proposta_inicio + 5
63      linhas_sem_proposta <- numeric(0)
64      if(length(sem_proposta_fim != 0))
65        {
66              for( i in 1:length(sem_proposta_inicio))
67                {
68                        linhas_sem_proposta <- c(linhas_sem_proposta, sem_proposta
                              _inicio[i] : sem_proposta_fim[i])
69                }
70        }
71
72      if(length(linhas_sem_proposta) != 0)
73        {
74              arquivo <- arquivo[-linhas_sem_proposta] @para excluir esses itens
                   na segunda parte do pdf
75        }
76
77      item_vazio <- arquivo[c( grep("Nao existem propostas", arquivo) - 1)]
78      item_vazio <- str_sub(item_vazio, 1, 11)
79      item_vazio <- str_trim(item_vazio)
80      item_vazio <- str_remove_all(item_vazio, " -")
81      item_vazio <- as.numeric(str_trim(str_sub(item_vazio, start = - 3)))
82
83
84
85      @lixo2: grupos
86      grupos_inicio <- grep("Relacao de Grupos", arquivo)
87      grupos_fim <<- grep("Historico", arquivo)
88      if(length(grupos_inicio) != 0){arquivo <- arquivo[-c(grupos_inicio:(grupos_
            fim - 1))]}
89
90
91      @lixo3: desempate
92      desempate_inicio <- grep("Desempate de", arquivo)
93      desempate_fim <- desempate_inicio + 2
94
95
96
97
```

```
98
99
100
101     linhas_desempate <- numeric(0)
102     if(length(desempate_fim != 0))
103       {
104             for( i in 1:length(desempate_inicio))
105               {
106                       linhas_desempate <- c(linhas_desempate, desempate_inicio[i
                            ] : desempate_fim[i])
107               }
108       }
109     arquivo[linhas_desempate]
110     i <- 1
111     if(length(linhas_desempate) != 0){
112       arquivo <- arquivo[- linhas_desempate]
113     }
114
115
116     @ Para compor a variavel Identif_Item_Compra, precisaremos da uasg, do
          pregao, do item, e da modalidade de licitacao (05-pregao)
117
118     linha_uasg <- grep("\\d{5}", arquivo)[1]
119     historico <- grep("Historico", arquivo)
120     linhas_item <- grep("Item:", arquivo[1 : historico]) @para buscar ate a
          primeira parte do relatorio, vai at  "historico"
121     item <- str_sub(arquivo[linhas_item], 8, 11)
122     item <- str_trim(item)
123     item <-str_remove_all(item, " -")
124
125     for(i in 1 : length(item))
126       {
127             if(item[i] %in% itens_sem_proposta)
128             {
129               item[i] <- NA
130             }
131       }
132
133     item <- na.omit(item)
134
135     item <- str_c("0000", item)
136     item <-str_sub(item, star = -5)
137     if(length(item_vazio) != 0)
138       {
139             item <- item[-c(item_vazio)]        @item
140       }
141
142
143
144
145
146     @Unidade Administrativa de Servicos Gerais
147     uasg_pregao <- data.frame(uasgm = str_sub(arquivo[linha_uasg]))
148     uasg_pregao <- uasg_pregao %>% separate(uasgm, c("uasg", "pregao"), sep="\\.
          ", remove = F)
149
150     uasg <- uasg_pregao[, 2]
151     uasg <- str_trim(uasg)
152
153     @Numero do pregao
```

```
154     pregao <- uasg_pregao[, 3]
155     pregao <- str_trim(pregao)
156     pregao <- str_c("00000000000", pregao)
157     pregao <- str_sub(pregao, start = -9)
158
159     @Modalidade de licitacao, sempre 5
160     mod <- "05"
161
162
163     Identif_Item_Compra <- str_c(uasg, mod , pregao, item)
164
165
166
167     a <- numeric(0)
168     for(i in 1 : length(arquivo))
169       {
170             if(arquivo[i] == "")
171             {
172               a <- c(a,i)
173             }
174     }
175
176     table(arquivo == "")
177     arquivo <- arquivo[-a]
178
179
180
181     @ Coletar o Valor Estimado de cada item
182
183     linhas_est <- grep("Valor estimado:", arquivo)
184
185     a <- str_locate(arquivo[linhas_est], "R\\$ ")
186     j <- 1
187     est <- numeric(0)
188
189     for(i in linhas_est)
190       {
191             est[j] <- substr(arquivo[i], a[1, 2] + 1, a[1, 2] + 1 + 10)
192             j <- j + 1
193       }
194
195     est <- str_remove_all(est, "\\.")
196     est <- str_replace(est, ",", "\\.")
197     est <- str_trim(est)
198     est <- as.numeric(est)
199
200     @Coletar a quantidade ofertada do item
201
202
203     linhas_qtd <- linhas_est - 1
204
205     a <- str_locate(arquivo[linhas_qtd], ":")
206     j <- 1
207     qtd <- numeric(0)
208
209     for(i in linhas_qtd)
210       {
211             qtd[j] <- substr(arquivo[i], a[1,2] + 1, a[1,2] + 1 + 8)
212             j <- j + 1
213     }
```

```
214
215     qtd <- str_remove_all(qtd, "\\.")
216     qtd <- str_trim(qtd)
217     qtd <- as.numeric(qtd)
218     qtd
219
220
221
222     @ Coletar os lances de cada item, bem como o cnpj de cada fornecedor
223
224
225     linhas_lan_inicio <- grep("Valor do Lance" , arquivo) + 1     @ Aonde comeca
            os lances em todos os itens
226     linhas_lan_fim <- grep("Eventos do Item", arquivo) - 1       @ Aonde termina
            os lances em todos os itens
227
228     n_lances <<- linhas_lan_fim - linhas_lan_inicio + 1  @ O numero de lances em
             cada item
229
230     lances_por_item[[k]] <<- n_lances
231
232     linhas_lances <- c() @ para listar todas as linhas em que ha lances
233
234     for (i in 1 : length(linhas_lan_fim))
235       {
236             linhas_lances <- c(linhas_lances, linhas_lan_inicio[i] : linhas_lan_
                  fim[i])
237       }
238
239
240     j <- 1
241     lan <- numeric(0)
242     cnpj <- numeric(0)
243     for(i in linhas_lances)
244       {
245             b <- str_locate(arquivo[linhas_lances], "R\\$ ")
246             l <- str_locate(arquivo[linhas_lances], "-") - 17
247             lan[j] <- substr(arquivo[i], b[j,1] + 2, b[j,2] + 10)
248             cnpj[j] <- substr(arquivo[i], l[j,1] + 2, l[j,2] + 19)
249             str_sub(arquivo[linhas_lances], start = -12)
250             j <- j + 1
251     }
252     options(scipen = 999)
253
254     @ Cnpj do fornecedor de cada lance
255     cnpj <- str_remove_all(cnpj, "\\.")
256     cnpj <- str_remove_all(cnpj, "/")
257     cnpj <- str_remove_all(cnpj, "-")
258     cnpj <- as.numeric(cnpj)
259     cnpj <- as.character(cnpj)
260     cnpj <- str_c("0", cnpj)
261     cnpj <- str_sub(cnpj,start = -14)
262
263
264     @Valor de cada lance
265     lan <- str_remove_all(lan, "\\.")
266     lan <- str_replace(lan, ",", "\\.")
267     lan <- str_trim(lan)
268     lan <- as.numeric(lan)
269
```

```
270
271     @pegar a hora de abertura do pregao
272
273     linha_abertura <- grep("      Item aberto.", arquivo) + 1
274     hora_abertura <- arquivo[linha_abertura]
275
276     hora_abertura <- hora_abertura[1 : length(n_lances)]
277
278     @ repetir o horario para adequar o numero de linhas ao numero de lances
279     hora_abertura_rep <- numeric(0)
280     for(i in 1 : length(hora_abertura))
281       {
282             hora_abertura_rep <- c(hora_abertura_rep, rep(hora_abertura[i], n_
                  lances[i]))
283       }
284
285
286     linha_fechamento <- grep("      Item encerrado", arquivo) + 1
287     hora_fechamento <- arquivo[linha_fechamento]
288     hora_fechamento <- str_trim(hora_fechamento)
289
290     hora_fechamento <- hora_fechamento[1 : length(n_lances)]
291     hora_fechamento_rep <- numeric(0) @ para adequar o n mero de linhas ao
          n mero de lances
292     for(i in 1 : length(hora_fechamento))
293       {
294             hora_fechamento_rep <- c(hora_fechamento_rep, rep(hora_fechamento[i
                  ], n_lances[i]))
295       }
296
297
298
299     @ Ler a data e a hora do Lance
300     data_hora <- str_sub(arquivo[linhas_lances], start = -23)
301     data_hora_frame <- data.frame(datahora = data_hora)
302     data_hora_sep <- data_hora_frame %>%
303             separate(datahora, c("data", "hora"), sep = " ", remove = F)
304
305
306     data <- data_hora_sep[, 2]
307     hora <- data_hora_sep[, 3]
308
309
310     @Repetir Identif_Item_Compra para adequar o numero de linhas ao numero de
          lances
311     Identif_Item_Compra <- Identif_Item_Compra[1 : length(n_lances)]
312
313     id_item_rep <- numeric(0)
314     for(i in 1 : length(Identif_Item_Compra))
315       {
316             id_item_rep <- c(id_item_rep, rep(Identif_Item_Compra[i], n_lances[i
                  ]))
317       }
318
319     @ Repetir o valor estimado para adequar o numero de linhas ao numero de
          lances
320     est <- est[1 : length(n_lances)]
321     est_rep <- numeric(0)
322     for(i in 1 : length(est))
323       {
```

```
324             est_rep<-c(est_rep, rep(est[i], n_lances[i]))
325       }
326
327     @ Repetir a qtd ofertada para adequar o numero de linhas ao numero de lances
328     qtd <- qtd[1 : length(n_lances)]
329     qtd_rep <- numeric(0)
330     for(i in 1 : length(qtd))
331       {
332             qtd_rep <- c(qtd_rep, rep(qtd[i], n_lances[i]))
333       }
334
335
336
337     l <- data.frame(id_item_rep, hora_abertura_rep, hora_fechamento_rep, cnpj,
338                     lan, data,hora, qtd_rep)
339
340     names(l) <- c("Identif_Item_Compra", "Hora de Abertura", "Hora de Fechamento
            ",
341                   "CPF/CNPJ_Fornecedor", "Valor dos Lances", "Data do Lance","
                        Hora do Lance",
342                   "Qtd ofertada")
343
344     tab <<- data.frame(id_item_rep, hora_abertura_rep, hora_fechamento_rep, cnpj
            ,
345                        lan, est_rep, qtd_rep, data, hora)
346
347
348     names(tab) <<- c("Identif_Item_Compra", "Hora de Abertura", "Hora de
            Fechamento",
349                      "CPF/CNPJ_Fornecedor", "Valor dos Lances", "Valor Estimado"
                             ,
350                      "Qtd ofertada", "Data do Lance", "Hora do Lance")
351
352     tabi[[i]] <<- tab
353
354     dad <- tab %>%
355       group_by(Identif_Item_Compra, `CPF/CNPJ_Fornecedor`) %>%
356       summarise(`Valor dos Lances` = min(`Valor dos Lances`),
357                 `Valor Estimado` = min(`Valor Estimado`)) %>%
358       na.omit() %>%
359       as.data.frame()
360
361
362     @Toda a informacao extraida de uma ata i e armazenada na lista como i-esimo
            elemento
363
364     tabelao[[i]] <<- left_join(dad, l,
365                                by = c("Identif_Item_Compra", "Valor dos Lances",
366                                       "CPF/CNPJ_Fornecedor"))
367
368
369   }
370   return(tabelao)
371 }
372
373
374
375
376
377
```

```
378
379
380
381
382
383
384 @ Funcao para selecionar os arquivos, chamar a funcao de mineracao de atas,
      juntar o minerado de cada ata e acrescentar o
385 @ minerado no banco de dados fornecido pela secretaria da fazenda do DF
386
387 Juntar_bancos_dados <- function()
388   {
389     cat("selecione as atas desejadas, desde que elas estejam na mesma pasta")
390
391     @ Vetor com o endereco dos arquivos
392     arquivos.aux <- choose.files(default = getwd(),
393                                  caption = "Selecione os arquivos com os bancos de
                                         dados")
394
395
396     tabelas <- mining_est_lan(arquivos.aux) @ retorna uma tabela minerada para
          cada ata
397     minerado <<- do.call(rbind, tabelas) @ Junta em uma tabela so
398
399
400     tab2 <<- do.call(rbind, tabi)
401
402     x <- left_join(tab2, dados) %>% na.omit()
403     y <-left_join(dados, x)
404
405     @banco do governo mais o minerado com todos os lances, nao apenas o minimo
          de cada fornecedor
406     y <<- -y
407
408
409     @ Retorna o banco de dados atualizado
410     tabela <<- left_join(dados, minerado)
411
412     return(minerado)
413 }
414
415 Juntar_bancos_dados()
416
417
418
419 tabela <- na.omit(tabela) @ Os dados considerando apenas o minimo de cada
      fornecedor
420 y <- na.omit(y) @ Os dados considerando todos os lances de cada ata
421
422
423
424
425
426 criar_banco <- function(base)
427   {
428     @Criar variavel menor lance de cada item
429
430     dada1<- base %>%
431           group_by(Identif_Item_Compra) %>%
432           na.omit() %>%
```

```
433         summarise( "Menor Lance" = min(`Valor dos Lances`)) %>%
434         as.data.frame()
435
436     m <- as.data.frame(table(base$Identif_Item_Compra))
437     names(m) <- c("Identif_Item_Compra", "n_lances")
438     dada1 <- left_join(dada1, m)
439
440
441     tabel <- left_join(base, dada1) %>% na.omit()
442
443     @Variaveis de taxa
444
445     tabel$p_lances_min <- tabel$`Valor dos Lances` / tabel$`Menor Lance`
446     tabel$p_lances_est <- tabel$`Valor dos Lances` / tabel$`Valor Estimado`
447
448     @Converter variaveis de Tempo
449
450     tabel$`Hora do Lance` <- str_sub(string = tabel$`Hora do Lance`, end = -5)
451     tabel$`Hora do Lance`<- hms(tabel$`Hora do Lance`) %>% as.numeric()
452     tabel$`Hora de Abertura`<- hms(tabel$`Hora de Abertura`) %>% as.numeric()
453
454
455
456     tabel <- tabel %>% as.data.frame() %>%
457         group_by(Identif_Item_Compra) %>%
458         summarise(inicio = min(`Hora do Lance`)) %>%
459         left_join(tabel, .) %>%
460         mutate(tempo=`Hora do Lance` - inicio)
461
462
463
464     a <- tabel[order(tabel$`Data do Lance`, tabel$Identif_Item_Compra, tabel$`
        Hora do Lance`,
465                      decreasing = c(TRUE, FALSE, F)), ]
466     names(a)
467     @a <- a[,c(1,6,10,19:33)]
468     a <- rev(a[nrow(a) : 1,])
469     a <- a[ , c(ncol(a) : 1)]
470
471
472     banco <<- a
473
474     @Criar variavel que indica se o lance foi feito apos o periodo de abertura
475     valido <- numeric(0)
476     for( i in 1 : length(banco$`Hora do Lance`))
477       {
478             ifelse(banco$`Hora do Lance`[i] >= banco$`Hora de Abertura`[i],
479                    valido[i] <- 1,
480                    valido[i] <- 0)
481       }
482     banco$valido <<- valido
483
484     apos_termino <- numeric(0)
485     for( i in 1 : length(banco$`Hora do Lance`))
486       {
487             ifelse(banco$`Hora do Lance`[i] >= banco$`Hora de Fechamento`[i],
488                    apos_termino[i] <- 1,
489                    apos_termino[i] <- 0)
490       }
491     banco$apos_termino <<- apos_termino
```

```
492
493
494     @Criar variavel que indica o numero do lance
495     id <- c(table(banco$Identif_Item_Compra))
496     bancocontagem <- numeric(0)
497     for(i in 1 : length(id)){
498
499       bancocontagem <- c(bancocontagem, 1 : id[i])
500     }
501     banco$contagem <- bancocontagem
502
503
504
505     return(banco)
506 }
507
508
509
510 criar_banco(y)
511
512
513 for(i in 1 : length(banco$cancelado))
514   {
515     ifelse(banco$cancelado[i] == 0, banco$cancelado[i] <- 1, banco$cancelado[i]
          <- 0)
516   }
517
518
519
520 @ Salvar o objeto em rds
521
522 saveRDS(banco, file = "banco.rds")
523
524
525 @ Ler o objeto em rds
526 banco1 <- readRDS(file = "banco.rds")
527
528 apos_termino <- numeric(0)
529 for( i in 1 : length(banco1$`Hora do Lance`))
530   {
531     ifelse(banco1$`Hora do Lance`[i] >= banco1$`Hora de Fechamento`[i],
532            apos_termino[i] <- 1,
533            apos_termino[i] <- 0)
534   }
535
536 banco1$apos_termino <<- apos_termino
537 table(apos_termino)
538
539 apos_termino <- numeric(0)
540 for( i in 1 : length(banco3$`Hora do Lance`))
541   {
542     ifelse(banco3$`Hora do Lance`[i] >= banco3$`Hora de Fechamento`[i],
543            apos_termino[i] <- 1,
544            apos_termino[i] <- 0)
545
546   }
547 banco3$apos_termino <- apos_termino
548
549
550
```

```
@banco com todos os lances
banco1
nomes <- names(banco1)
nomes <- str_replace_all(nomes, "\\.", " ")
names(banco1) <- nomes

banco1$Gasto_Compra <- banco1$Valor_Unitario_Homologado * banco1$`Qtd ofertada`


banco1 <- banco1 %>%
    group_by(Identif_Item_Compra) %>%
    summarise(a = max(Gasto_Compra)) %>%
    left_join(banco1, .)


SRP <- c("45010705000022019", "45010705000072019", "92504105000092019",
         "92504105001412018", "92601605000052019", "92611905000952019")

@Concertar erros de padronizacao em algumas atas

for( i in 1 : length(banco1$`Valor Estimado`))
  {
    if(banco1$Identif_Compra[i] %in% SRP)
      {
            banco1$`Valor Estimado`[i] <- banco1$`Valor Estimado`[i] * banco1$`
                Qtd ofertada`[i]
            banco1$`Valor dos Lances`[i] <- banco1$`Valor dos Lances`[i] *
                banco1$`Qtd ofertada`[i]
      }

}



@banco com apenas o minimo de cada fornecedor
dad <- banco1 %>%
    group_by(Identif_Item_Compra, `CPF/CNPJ_Fornecedor`) %>%
    summarise(`Valor dos Lances` = min(`Valor dos Lances`),
              `Valor Estimado` = min(`Valor Estimado`)) %>%
    na.omit() %>%
    as.data.frame()


banco2 <- left_join(dad, banco1)


@banco com apenas os lances homologados



banco3 <- banco2 %>%
    filter(Valor_Unitario_Homologado>0)



b <- banco3[, c("Identif_Item_Compra", "Valor dos Lances")]
names(b) <- c("Identif_Item_Compra", "Lance vencedor")
banco1 <- right_join(b, banco1)


```

```
609 banco1$prop_vencedor <- banco1$`Valor dos Lances` / banco1$`Lance vencedor`
610
611
612 for( i in 1 : length(banco1$`Lance vencedor`))
613   {
614     if(banco1$Identif_Compra[i] %in% SRP)
615       {
616         banco1$`Lance vencedor`[i] <- banco1$`Lance vencedor`[i] * banco1$`Qtd
              ofertada`[i]
617       }
618   }
619
620
621
622
623 est_unitario <- names(table(banco3$Identif_Compra))[c(1, 2, 3, 5, 6, 7)]
624
625
626
627
628
629 @Criar variaveis importantes para a modelagem
630
631 banco3$prop_valor_estimado <- banco3$Gasto_Compra / banco3$`Valor Estimado`
632 banco3$prop_negociada <- banco3$Gasto_Compra / banco3$`Valor dos Lances`
633
634
635
636
637 @transformacao meses
638
639 meses <- data.frame(Mes_Resultado_Compra =
640                       c("Abr 2019", "Ago 2019", "Fev 2019", "Jan 2019", "Jul
                            2019",
641                         "Jun 2019", "Mai 2019", "Mar 2019", "Set 2019"),
642                     Mes=c(4,8,2,1,7,6,5,3,9))
643
644 banco3 <-left_join(banco3, meses)
645
646
647 @cluster com grupo material servico
648
649 length(table(banco3$Grupo_Material_Servico))
650
651 ad <- tapply(banco3$Gasto_Compra, banco3$Grupo_Material_Servico, mean)
652 ad1 <- tapply(banco3$Qtde_Ofertada, banco3$Grupo_Material_Servico, mean)
653 asd <- data.frame(ad, ad1)
654 asd <- scale(asd)
655
656 fviz_nbclust(asd, kmeans, method = "gap_stat")
657 dados_kmeans <- kmeans(asd, 8)
658
659 lista <- dados_kmeans$cluster
660 asdf <- cbind(asd,lista)[, 3]
661 l <- names(asdf)
662
663 Grupo_M_S <- data.frame(Grupo_Material_Servico=l,Tipo_Objeto=asdf)
664 Grupo_M_S$Grupo_Material_Servico <- as.character(Grupo_M_S$Grupo_Material_
      Servico)
665 Grupo_M_S$Tipo_Objeto <- as.factor(Grupo_M_S$Tipo_Objeto)
```

```
666 table(Grupo_M_S$Tipo_Objeto)
667
668 banco3 <- left_join(Grupo_M_S, banco3)
669
670 fviz_cluster(dados_kmeans, data = asd)
671
672
673 @cluster com grupo material servico
674
675 dados_kmeans <- kmeans(asd, 8)
676
677 lista <- dados_kmeans$cluster
678 asdf <- cbind(asd, lista)[, 3]
679 l <- names(asdf)
680
681 Grupo_M_S_G <- data.frame(Grupo_Material_Servico = l, Tipo_Objeto_G = asdf)
682 Grupo_M_S_G$Grupo_Material_Servico <- as.character(Grupo_M_S$Grupo_Material_
        Servico)
683 Grupo_M_S_G$Tipo_Objeto_G <- as.factor(Grupo_M_S_G$Tipo_Objeto_G)
684 table(Grupo_M_S_G$Tipo_Objeto)
685 banco3 <- left_join(Grupo_M_S_G, banco3)
686
687 fviz_cluster(dados_kmeans,data = asd)
688
689
690
691
692
693 @cluster manual para uasg resp compra
694
695 uasgs <-sort(names(table(banco3$UASG_Resp_Compra)))
696 cluster <- c("Saude", "Seguranca", "Saude", "Comercio",
697              "Assistencia", "Planej e Gestao", "Seguranca")
698 Grupo_Uasg <- data.frame(UASG_Resp_Compra = uasgs, setor = cluster)
699 banco3 <-left_join(Grupo_Uasg, banco3)
700
701
702
703
704 names(banco3)[25] <-"Porte_Empresa"
705
706
707
708 @Encontrar itens que venceram e nao apareceram antes da fase de abertura
709
710 testar1 <- banco1 %>%
711     filter(valido == 0) %>%
712     group_by(Identif_Item_Compra) %>%
713     distinct(`CPF/CNPJ_Fornecedor`)
714
715 testar1$valido <- rep(0, length(testar1$Identif_Item_Compra))
716
717
718 testar2 <- banco1 %>%
719     filter(valido == 1) %>%
720     group_by(Identif_Item_Compra) %>%
721     distinct(`CPF/CNPJ_Fornecedor`)
722
723 testar2$valido <- rep(1, length(testar2$Identif_Item_Compra))
724
```

```
ff <- rbind(testar1, testar2)


test <- ff %>%
    group_by(Identif_Item_Compra) %>%
    filter ( !duplicated('CPF/CNPJ_Fornecedor')) %>%
    filter(valido == 1)
test$Identif_Item_Compra @itens que venceram e nao apareceram na fase de
   abertura


for( i in 1 : length(banco3$Identif_Item_Compra))
  {
    ifelse(banco3$Identif_Item_Compra[i] % in % test$Identif_Item_Compra,
           banco3$autorizado[i] <- 0,
           banco3$autorizado[i] <- 1)
  }
table(banco3$autorizado)

@Duas variaveis que nao deixam passar de 1
banco3$acima_do_estimado <- as.numeric(banco3$prop_valor_estimado - 1 > 0)
banco3$negociado_acima <- as.numeric(round(banco3$prop_negociada, 10) > 1)




hist(banco3$prop_negociada)
round(quantile(banco3$prop_negociada, seq(0.0, 0.5, 0.05)), 1)



@Classes para prop_valor_estimado
hist(banco3$prop_valor_estimado)
round(quantile(banco3$prop_valor_estimado, seq(0.1, 1, 0.1)) * 100, 1)
teste <- banco3$prop_valor_estimado - 0.0001
brk <- seq(0, 1, .1)
classes <- c(1 : 10) / 10 @ nomes das classes
teste <- cut(teste, breaks = brk, right = FALSE, labels = classes)
banco3$prop_valor_estimado_classes <- teste
table(banco3$prop_valor_estimado_classes)

@Classes para prop_negociada

round(quantile(banco3$prop_negociada, seq(0, 0.7, 0.05)), 3)
negociacao_alta <- numeric(0)
for(i in 1 : length(banco3$UASG_Resp_Compra))
  {
    if(banco3$prop_negociada[i] <= 0.469)
      negociacao_alta[i] <- 1
    if( banco3$prop_negociada[i] > 0.469 && banco3$prop_negociada[i] < 0.999)
      negociacao_alta[i] <- 2
    if( banco3$prop_negociada[i] > 0.999)
      negociacao_alta[i] <- 3
  }


banco3$negociacao_alta <- negociacao_alta
table(banco3$negociacao_alta)
```

```
784
785 @Classes para prop_valor_estimado
786 hist(banco3$p_lances_min)
787 round(quantile(banco3$p_lances_min, seq(0.1, 1, 0.025)), 3)
788
789 lance_menor <- numeric(0)
790 for( i in 1 : length(banco3$UASG_Resp_Compra))
791   {
792     if(banco3$p_lances_min[i]<= 1)
793       lance_menor[i] <- "Nao ha Lance menor"
794     if(banco3$p_lances_min[i] > 1 && banco3$p_lances_min[i] <= 1.382)
795       lance_menor[i] <- " um pouco menor"
796     if(banco3$p_lances_min[i] > 1.382 && banco3$p_lances_min[i] <= 2.714)
797       lance_menor[i] <- "menor"
798     if( banco3$p_lances_min[i] > 2.714)
799       lance_menor[i] <- "Muito menor"
800   }
801
802 banco3$lance_menor <- as.factor(lance_menor)
803
804
805
806
807
808
809 table(banco3$lance_menor)
810
811 @(y * ( n 1 ) + 0.5) / n
812 @transformacao na variavel resposta:
813 prop_valor_estimado <- banco3$prop_valor_estimado
814 saveRDS(prop_valor_estimado, file = "variavelantesdetransformar.rds")
815
816
817
818 Y <- numeric(0)
819 for( i in 1 : length(banco3$prop_valor_estimado))
820   {
821     ifelse(banco3$prop_valor_estimado[i] > 1,
822            Y[i] <- NA ,
823            Y[i] <-  banco3$prop_valor_estimado[i])
824 }
825 Y1 <- (Y * (length(Y) - 1) + 0.5) / length(Y)
826
827 table(is.na(Y1))
828
829 max(Y1)
830 tail(sort(Y1), 100)
831 banco3$Y <- Y1
832
833
834 @minerando<-banco3
835
836
837
838 saveRDS(minerando, file = "minerando.rds")
839
840
841 banco3 <- readRDS(file = "minerando.rds")
842 names(banco3)
843
```

```
844
845
846
847 banco1$`Hora de Fechamento` <- hms(banco1$`Hora de Fechamento`) %>% as.numeric()
```

Listing 1: Código fonte em R da mineração

7.2 Códigos dos modelos e gráficos

```
1
2 @BANCO3- tabela com os lances homologados
3 getwd()
4 banco3 <- minerando
5
6
7 @distribuicao beta
8 p = seq(0, 1, length = 100)
9 plot(p, dbeta(p, 100, 100), ylab = "",xlab = "", type = "l", col = 4)
10 lines(p, dbeta(p, 5, 5), type = "l", col = 6)
11 lines(p, dbeta(p, 12, 3), col = 2)
12 lines(p, dbeta(p, 3, 20), col = 9)
13 lines(p, dbeta(p, 0.5, 3), col = 1)
14 legend(0.7, 8, c("Be(100,100)","Be(5,5)","Be(20,3)", "Be(3,20)","Be(0.5,3)"),
15        lty = c(1, 1, 1, 1, 1),
16        col = c(4, 6, 2, 9, 1), cex =.6)
17
18
19
20 @lances na fase de lances
21 a<-banco3 %>%
22      filter(valido == 1) %>%
23      filter(apos_termino == 0)
24
25 a <- a %>% filter(apos_termino == 0)
26 dim(a)[1]
27
28
29
30 @lances na fase de abertura
31 a <- banco3 %>% filter(valido == 0) @226
32
33 @lances apos o termino do pregao
34 a <- banco3 %>% filter(apos_termino == 1) @195
35
36
37 sum(banco3$Gasto_Compra)
38
39 sort(tapply(banco3$Gasto_Compra, banco3$Grupo_Material_Servico, sum))
40 tapply(banco3$Gasto_Compra, banco3$setor, sum)
41 tapply(dados$Gasto_Compra, dados$Tipo_Material_Servico, mean)
42
43
44 ver_coelho <- banco1 %>% filter(valido == 1)
45
46
47
48 g3 <- ggplot(data = banco3, aes(x = prop_negociada, y = prop_valor_estimado)) +
49      geom_point(aes(colour = Forma_Compra)) +
50      geom_point(data = aut) +
51      theme_minimal() +
52      theme(text = element_text(size = 9)) +
53      theme(legend.position = "left") +
54      theme(legend.text = element_text(size = 9))
55
56
57 p1 <- ggMarginal(g3, type = "histogram", size = 4, fill = "slateblue")
58
```

```
59
60
61
62 ggplot(data = banco3, aes(x = prop_valor_estimado)) +
63     geom_density(aes(fill = Porte_Empresa), alpha = 0.7) +
64     ylab("") +
65     theme_minimal() +
66     scale_x_continuous(breaks = c(0, 0.25, 0.5, 0.75, 1), limits = c(0,1)) +
67     xlab("") +
68     theme(legend.text = element_text(size = 9)) +
69     scale_fill_manual("Porte da Empresa", values = brewer.pal(3, "Dark2")) +
70     theme(text = element_text(size = 9))
71
72
73
74 @g2<-ggplot(data=banco3,aes(x = prop_valor_estimado)) +
75 @  geom_density(aes(fill = Forma_Compra),alpha = 0.5)+
76 @  ylab("")+
77 @  xlab("")+
78 @  theme_minimal()+
79 @  theme(legend.text=element_text(size=7))
80
81 ggplot(data = banco3, aes(y = Tipo_Material_Servico, x = prop_valor_estimado)) +
82     ylab("") +
83     xlab("") +
84     scale_x_continuous(breaks = c(0, 0.25, 0.5, 0.75, 1), limits = c(0,1)) +
85     theme(legend.text = element_text(size = 9)) +
86     geom_density_ridges(na.rm = TRUE, aes(fill = Forma_Compra), alpha = 0.7) +
87     theme(text = element_text(size = 9)) +
88     theme_minimal()
89
90
91 teste <- banco3 %>% filter(Tipo_Material_Servico == "Servico")
92 table(teste$Forma_Compra)
93
94 @ggplot(data=banco3,aes(x = prop_valor_estimado)) +
95 @  geom_density(aes(fill = Tipo_Material_Servico),alpha = 0.5)+
96 @  ylab("")+
97 @  xlab("")+
98 @  theme_minimal()+
99 @  theme(legend.text=element_text(size=7))
100
101
102 ggplot(data=banco3, aes(y = factor(Mes), x = prop_valor_estimado)) +
103     ylab("Mes") +
104     theme_minimal() +
105     scale_x_continuous(breaks = c(0, 0.25, 0.5, 0.75, 1), limits = c(0,1)) +
106     theme(legend.text = element_text(size = 9)) +
107     geom_density_ridges(na.rm = TRUE, aes(fill = Tipo_Material_Servico), alpha =
          0.7) +
108     theme(text = element_text(size = 9))
109 table(banco3$Mes)
110
111
112 ggplot(data=banco3, aes(y = setor, x = prop_valor_estimado)) +
113     geom_density_ridges(na.rm = TRUE, aes(fill=Tipo_Material_Servico))+
114     theme(legend.text = element_text(size = 9)) +
115     ylab("") +
116     scale_x_continuous(breaks = c(0, 0.25, 0.5, 0.75, 1), limits = c(0,1)) +
117     theme_minimal() +
```

```
    xlab("") +
    theme(text = element_text(size = 9))
table(banco3$setor)

@grid.arrange(g1, g3, g5, g6, ncol = 2, nrow = 2)

table(banco3$setor)


@prop_valor estimado


names(banco3)


variavel <- banco3$setor
min <- tapply(banco3$prop_valor_estimado, variavel, min)
max <- tapply(banco3$prop_valor_estimado, variavel, max)
media <- tapply(banco3$prop_valor_estimado, variavel, mean)
desvio <- tapply(banco3$prop_valor_estimado, variavel, sd)
mediana <- tapply(banco3$prop_valor_estimado, variavel, median)
data.frame(media, desvio, CV = desvio / media, mediana, min, max)




@modelo glm para ajustar aos dados


banco_mod <- banco3
banco_mod <- banco_mod %>% filter(autorizado == 1)

summary(banco_mod$prop_valor_estimado)

@banco_mod<- banco_mod %>% filter(prop_valor_estimado<1)@ tem um valor
    estranhamente acima de 1




mod <- numeric(0)
met <- numeric(0)
beta <- function(ligacao)
  {
    mod <<- betareg(Y ~
                    Porte_Empresa + Tipo_Objeto + setor +
                     Forma_Compra, link = ligacao,
                    data=banco_mod)

    mse <- mean((mod$residuals) ^ 2)
    mae <- mean(abs(mod$residuals))
    md_erros <- median(abs(mod$residuals))
    n_interacoes <- tail(summary(mod))$iterations[1]
    r2 <- mod$pseudo.r.squared
    loglik <- logLik(mod)[1]
    met <- data.frame(AIC = AIC(mod), MSE = mse, RMSE = sqrt(mse), mae, md_erros
        ,
                      Verossimilhanca = loglik, phi= mod$coefficients[[2]], n_
                          interacoes, r2)
```

```
175
176     return(met)
177   }
178 beta("logit")
179
180 cof <- mod$coefficients
181 length(cof$mean)
182
183
184 data.frame(sapply(c("logit", "probit", "cloglog", "cauchit", "loglog"),
185                   beta))
186
187 summary(mod)
188
189 cor(predict(mod), banco_mod$prop_valor_estimado)
190
191 par(mfrow=c(2, 2))
192 plot(mod)
193
194
195 coef(mod)
196
197
198 @GAMBOOSTLSS
199 mod<-glmboostLSS(Y ~
200                    Porte_Empresa + Tipo_Objeto + setor +
201                    Forma_Compra,  data = banco_mod, families = BetaLSS(),
                       weights = NULL)
202
203 set.seed(123)@ semente 2 deu 15 fold
204 @opt_int<-cvrisk(mod)@Optimal number of boosting iterations: 94 36
205 mstop(mod) <- mstop(opt_int) @ Retorna o numero otimo de interacoes para cada
      parametro
206 a <- coef(mod)
207 p <- length(a$mu)
208 emp_risk <- risk(mod, merge = TRUE)
209 risco <- tail(emp_risk, n = 1)
210 risco
211 names(risco) <- "AIC"
212
213 AIC = -2 * (-risco) + 2 * p
214 AIC
215
216
217 preditos_lig <- fitted(mod, parameter = "mu")
218 preditos <- inv.logit(preditos_lig)
219
220 residuo <- preditos - banco_mod$prop_valor_estimado
221
222 mse <- mean(residuo ^ 2)
223 mae <- mean(abs(residuo))
224
225
226
227
228 par(mfrow=c(1, 3),mai = c(1, 0.8, 0.5, 1.0))
229 plot(mod)
230 plot(opt_int)
231
232 mstop(mod) <- c(1000, 1000)
```

```
233
234
235
236
237 summary(mod)
238
239
240 min(preditos)
241
242
243 summary(mod)
244
245 summary(mod)
246 median(abs(residuo))
247
248
249
250 residuos_por_item <- data.frame(Identif_Item_Compra = banco_mod$Identif_Item_
    Compra,
251                                 banco_mod$prop_negociada, prop_valor_estimado =
                                        banco_mod$prop_valor_estimado,
252                                 preditos, residuo = abs(residuo))
253
254
255 s <- residuos_por_item[, c("Identif_Item_Compra", "residuo", "preditos")]
256 banco3 <- right_join(s, banco3)
257
258 residuos_por_item <- arrange(residuos_por_item, residuo)
259 lista_beta <- tail(residuos_por_item, 10)
260
261
262
263 quantis <- round(quantile(residuos_por_item$residuo, seq(0.1, 1, 0.1)), 2)
264 quantis
265 hist(residuos_por_item$residuo)
266
267 plot(banco3$residuo)
268
269
270 @ISOLATION FOREST
271
272
273 names(banco3)
274
275 variaveis_floresta <- c("prop_valor_estimado", "prop_negociada", "residuo",
276                         "lance_menor", "negociado_acima")
277
278
279 banco3_floresta <- banco3[, variaveis_floresta]
280 head(banco3_floresta)
281
282 n_arvores <- function(x)
283   {
284     floresta <- iForest(banco3_floresta, nt = x, seed = 0)
285     predict(floresta, banco3_floresta)
286   }
287
288
289 a <- data.frame(n_arvores(1), n_arvores(10), n_arvores(50), n_arvores(75), n_
    arvores(100))
```

```
290
291 ggpairs(a)
292 @usar n=50
293
294
295
296 floresta <- iForest(banco3_floresta, nt = 50, seed = 0)
297 score <- predict(floresta, banco3_floresta)
298
299
300 ggplot(banco3_floresta,
301        aes(x = score)) +
302        geom_histogram() +
303        ggtitle("Histograma do Score com 50 arvores") +
304        geom_vline(xintercept =  .559, colour = "red")
305
306 round(quantile(score, seq(0.866, 0.950, 0.475)), 3)
307 round(quantile(score, seq(0.0, 1, 0.05)), 3)
308 round(quantile(banco3$p_lances_min, seq(0.1, 1, 0.05)), 3)
309
310 banco3$score_floresta <- score
311
312 min(score)
313 max(score)
314 @View(banco3[,c(variaveis_floresta,"score_floresta")])
315
316
317 banco_final <- banco3
318
319
320 saveRDS(banco_final, file = "banco_final.rds")
321
322 banco3 <- readRDS(file = "banco_final.rds")
323
324
325
326
327  ggplot(data = banco3, aes(x = Y, y = factor(Mes))) +
328      geom_jitter(aes(colour=score_floresta),show.legend = T)+ggtitle("")+
329      theme(legend.position = "bottom") +
330      xlab("prop_valor_estimado") +
331      scale_colour_gradient2(midpoint = 0.5, low = "green", mid = "yellow",
332                             high = "red", space ="Lab") +
333      theme_minimal()
334
335
336
337   ggplot(banco3_floresta, aes(x = score)) +
338      geom_histogram() + ggtitle("") +
339      geom_vline(xintercept =  0.5, colour = "red") +
340      theme(panel.background = element_rect(fill = "azure")) +
341      theme_minimal()
342
343
344
345
346 max(banco3$prop_valor_estimado)
347
348
349 @gghighlight(valido == 1, label_key = pais) +
```

```
@gghighlight(negociado_acima == 0,unhighlighted_colour = "black") +
@gghighlight(acima_do_estimado == 0, unhighlighted_colour = "black")


@ Fazer uma lista de itens com scores mais altos do que o aceitavel



length(score)

residuos_por_item <- data.frame(Identif_Item_Compra = banco_mod$Identif_Item_
    Compra,
                                preditos, residuo = abs(residuo))
names(residuos_por_item)
residuos_por_item <- arrange(residuos_por_item, residuo)


names(banco3)

score_por_item <- banco3[, c("Identif_Item_Compra", "p_lances_min",
                             variaveis_floresta, "score_floresta")]

score_por_item <- arrange(score_por_item, score_floresta) %>% na.omit()
lista_floresta <- tail(score_por_item, 10)
lista_floresta <-lista_floresta[, -c(6, 7)]
lista_floresta



res_score <- residuos_por_item %>% filter(residuo <= 0.7)
```

Listing 2: Código fonte em R dos modelos e graficos