UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO

Anthony Carlleston de Lima

One-Class Classification Algorithms Aplicado a Modelos de Propensão

São Carlos

2022

Anthony Carlleston de Lima

One-Class Classification Algorithms Aplicado a Modelos de Propensão

Trabalho de conclusão de curso apresentado ao Centro de Ciências Matemáticas Aplicadas à Indústria do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, como parte dos requisitos para conclusão do MBA em Ciências de Dados.

Área de concentração: Ciências de Dados

Orientador: Prof. Dr. Vicente Garibay Cancho

São Carlos

2022

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

Lima, Anthony Carlleston
L732o One-Class Classification Algorithms Aplicado a Modelos de Propensão / Anthony Carlleston Lima; orientador Vicente Garibay Cancho. -- São Carlos, 2022.
52 p.

Trabalho de conclusão de curso (MBA em Ciência de Dados) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2022.

1. one-class classification. 2. propensity. 3. score model. 4. anomaly score. 5. linear time complexity. I. Garibay Cancho, Vicente, orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Anthony Carlleston de Lima

One-Class Classification Algorithms Aplicado a Modelos de Propensão

Trabalho de conclusão de curso apresentado ao Centro de Ciências Matemáticas Aplicadas à Indústria do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, como parte dos requisitos para conclusão do MBA em Ciências de Dados.

Data de defesa: 05 de março de 2022

Comissão Julgadora:

Prof. Dr. Vicente Garibay Cancho
Orientador

Professor
Convidado1

Professor
Convidado2

São Carlos
2022

AGRADECIMENTOS

Agradeço a Deus por me proporcionar a oportunidade de concluir essa especialização, agradeço a minha mãe pela compreensão de minha ausência nesse ano corrido, agradeço aos amigos da especialização em ciência de dados da USP pela troca de conhecimento e por tornar esse curso ainda mais rico de informações. Agradeço especialmente aos amigos do Banco Santander do time de ciência de dados, que me acolheram neste ano de 2021 como uma família, me motivando a cada dia no desenvolvimento deste trabalho.

RESUMO

Lima, A. C. ***One-Class Classification Algorithms* Aplicado a Modelos de Propensão**. 2022. 52p. Monografia (MBA em Ciências de Dados) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2022.

O uso de dados tem se tornando amplamente utilizado para realizar tomadas de decisões em diferentes áreas de negócio, onde vem se popularizando a utilização dos dados para as estratégias de negócio e com isso aumentando as captações e armazenamento. As empresas estão cada vez mais *Data-driven* e um dos problemas que acarretam o uso massivo dos dados é a utilização dos recursos computacionais seja para armazenar, manter ou utilizar diferentes tipos de análises de dados e algoritmos de aprendizado de máquina. Com isso está crescendo a preocupação com a complexidade computacional dos algoritmos utilizados, assim como o potencial de escalabilidade e sustentação, onde não apenas o desempenho desses algoritmos vêm sendo levado em conta na escolha desses modelos, mas também a quantidade de recurso computacionais utilizados e o tempo de processamento dos mesmos. Algoritmos com complexidade computacional linear O(n) vem sendo testados para simplificar todo o *pipeline* de dados melhorando os modelos implementados. Além disso, estudos de técnicas de redução de dimensionalidades, conhecidos como seleção de variáveis, vêm sendo testadas em paralelo. Com tudo, o trabalho abordou diferentes tipos de algoritmos de aprendizado de máquina em problemas de propensão de públicos-alvos com diferentes complexidades como: *Random Forest*, *Isolation Forest* e *OCSVM*. Foi testado junto aos algoritmos a aplicação de 4 pipelines de seleções de variáveis e a utilização dos scores de anomalias como variáveis de entrada em outros algoritmos com o objetivo de melhora na performance, onde a seleção de variáveis baseada em borda aggregation rank teve resultados promissores em relação às demais técnicas utilizadas. Os algoritmos de *One-class classification* não conseguiu vencer a performance dos modelos baseados em *Random Forest*, porém apresentou um potencial em agregar informação a esse tipo de algoritmo servindo como variável de entrada. O lucro hipotético aumentou cerca de 9,4% utilizando o score de anomalia no modelo de *Random Forest* trazendo uma possível melhoria utilizando esses scores para substituir as variáveis com elevado nível de curtoses.

Palavras-chave: one-class classification, modelos de propensão, isolation forest, complexidade de tempo linear, scores de anomalias.

ABSTRACT

Lima, A. C. **One-Class Classification Algorithms Applied to Propensity Models**. 2022. 52p. Monografia (MBA em Ciências de Dados) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2022.

The use of data has become widely used to make decisions in different areas of business, where the use of data for business strategies has become popular and with this increasing capture and storage. Companies are increasingly data-driven and one of the problems that lead to the massive use of data is the use of computing resources, whether to store, maintain or use different types of data analysis and machine learning algorithms. As a result, there is a growing concern about the computational complexity of the algorithms used, as well as the potential for scalability and support, where not only the performance of these algorithms has been taken into account when choosing these models, but also the amount of computational resources used and the processing time. Algorithms with linear computational complexity O(n) have been tested to simplify the entire data pipeline by improving the implemented models. In addition, studies of dimensionality reduction techniques, known as variable selection, have been tested in parallel. however, the work addressed different types of machine learning algorithms in target audience propensity problems with different computational complexities such as: *Random Forest*, *Isolation Forest* and *OCSVM*. The application of 4 features selection pipelines and the use of anomaly scores as input variables in other algorithms were tested with the algorithms with the objective of improving performance, where the selection of variables based on borda aggregation rank had promising results in compared to the other techniques used. The One-class classification algorithms could not beat the performance of models based on Random Forest, but it presented a potential to aggregate information to this type of algorithm, serving as an input variable. The hypothetical profit improved about 9.4% using the anomaly score in the Random Forest model, bringing a possible improvement using these scores to replace the variables with a high level of kurtosis.

Keywords: one-class classification, propensity models, isolation forest, linear time complexity, anomaly scores

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

RF	Random Forest
IForest	Isolation Forest
Itree	Isolation Tree
IV	Information Value
FS	Feature Selection
OCC	One-Class Classification
KS	Kolmogorov-Smirnov
OCSVM	One-Class Support Vector Machine
RBF	Radial Basis Function
Smote	Synthetic Minority Over-sampling Technique
ANOVA	Análise de variância
AUC	Area Under the Curve

SUMÁRIO

1 INTRODUÇÃO

Com o aumento da quantidade de informações e acúmulos de dados a preocupação no desempenho de algoritmos de *Machine Learning* para ambientes massivos e a complexidade computacional desses algoritmos vem se tornando fatores decisivos na escolha dos modelos implementados (HENSMAN; FUSI; LAWRENCE, 2013)

Estudos de redução de complexidades dos algoritmos têm sido realizados em busca do santo graal entre o desempenho e redução de recursos computacionais utilizados pelos algoritmos, aplicado a diferentes áreas como negócios e aplicações de visão computacional (GRELLERT, 2018).

Nesse cenário surge estudos relacionados a pipelines de pré-processamento com o objetivo de redução de dimensionalidade, comparações de algoritmos, paralelismo de diferentes técnicas, assim como abordagens que criam diferentes amostras dos conjuntos de dados originais para realizar a criação de vários modelos em paralelo utilizando técnicas de *ensembles* para uni-los (GARCÍA-OSORIO; HARO-GARCÍA; GARCÍA-PEDRAJAS, 2010).

Todavia a complexidade dos modelos está atrelada a dificuldade dos problemas, quanto mais complexos o conjunto de dados, mais quebras no espaço de variáveis são necessárias para as predições, e assim mais recursos computacionais são utilizados para otimizar os modelos, porém torna-se complexo estimar essa complexidades dos problemas sem antes testar os diferentes tipos de algoritmos.

O objetivo principal desse trabalho é aplicar modelos baseados em *One-Class Classification* no cenários de seleção de público-alvos suscetíveis a convergência de uma determinada ação estabelecida pelo negócio, será abordado modelo com complexidade linear O(n) de *One-class* chamado Isolation Forest desafiando o modelo de *Random Forest* com complexidade O(n*log(n)*d*k) com o objetivo de substituí-lo. Será abordado também o impacto desses scores de anomalias criados pelo *IForest* e *OCSVM* como variáveis de entrada no modelo de RF com o objetivo de agregar informações e facilitar a quebra no espaço de variáveis.

Com o intuito de simplificar o pré-processamento dos dados será testado 4 tipos de *Feature Selection* para reduzir a dimensionalidade, o que seria bem útil em ambientes *big data*, devido ao fato de que a quantidade de variáveis é fator crucial na utilização da memória ram utilizada, reduzindo a quantidade de variáveis de entrada nos algoritmos estaríamos reduzindo a complexidade dos mesmos.

2 REVISÃO BIBLIOGRAFICA

Nessa seção abordaremos a revisão bibliográficas sobre modelos de propensão, suas definições, importância, aplicações, impactos e falhas junto com os propósitos, vantagens e principais algoritmos baseados em *One-Class Classification* e seus usos para a discriminação de classes e grupos.

2.1 Modelos de Propensão

Um modelo de propensão pode ser referenciado como uma tabela de score de propensão estatística utilizado para calcular a probabilidade que visa prever o comportamento ou clientes em potenciais, ou seja, aqueles com maior probabilidade de aderir a uma oferta. (ROSENBAUM P. R., 1983) define o score de propensão como a probabilidade condicional de atribuir um tratamento particular referentes aos dados obtidos de cada clientes.

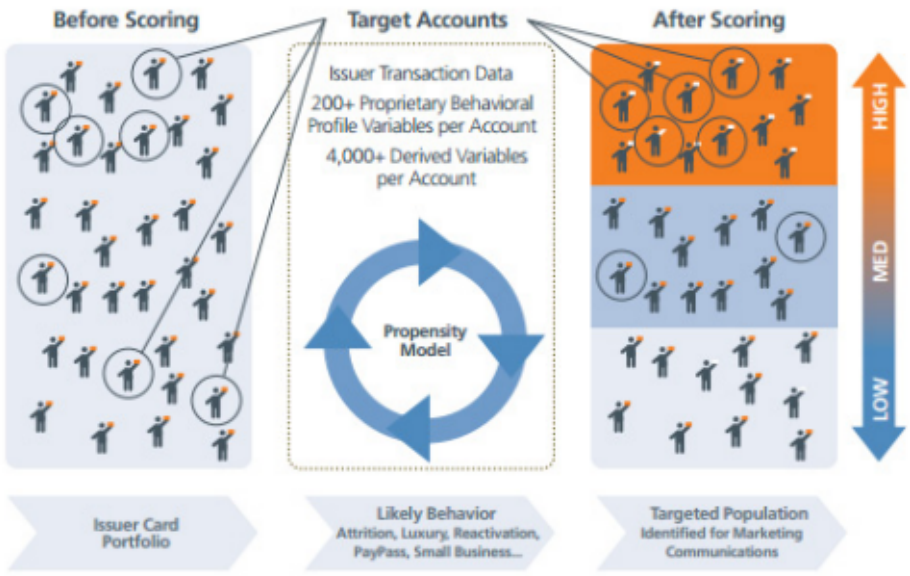
O uso de modelos de classificação utilizados para mensurar a probabilidade de potenciais clientes têm sido amplamente utilizado em diversos setores com o objetivo de auxiliar na detecção de perfis semelhantes aos que já aderiram a algum tipo de produto ou campanha, de forma a evitar ações aleatórias ou utilizando regras de negócio, o que acarretaria em vieses devido a intuições de especialistas da área (CHORIANOPOULOS, 2016).

(MASTERCARDADVISORS, 2017) explica de maneira simplificada como um escore de propensão funciona, Na figura 1 podemos identificar a classificação de uma população de clientes em 3 classes: Alta, média e baixa propensão representando a probabilidade daquela indivíduos escorados a terem uma resposta positiva em relação a campanha oferecida, identificando os clientes mais propensos podemos direcionar nossos produtos.

2.2 Desafios do modelo de Propensão

(SOLEDAD RAY BOSTON, 2003) relata alguns comportamentos interessantes em comparação aos métodos de escore de propensão e de regressão logística em um estudo experimental. No modelo de propensão foi relatado uma queda no viés inversamente proporcional às forças de associação de exposição conforme os resultados aumentavam, assim constou um modelo menos enviesado, mais robusto e mais preciso em comparação ao modelo de regressão logística. Em contrapartida, o erro médio padrão dos modelos de probabilidade da regressão logística tendem aos erros do modelo de propensão quando há pelo menos 8 variáveis de confusão por evento conforme o número de variáveis independentes aumentam, o autor conclui que a técnica de regressão logística é a técnica escolhida quando

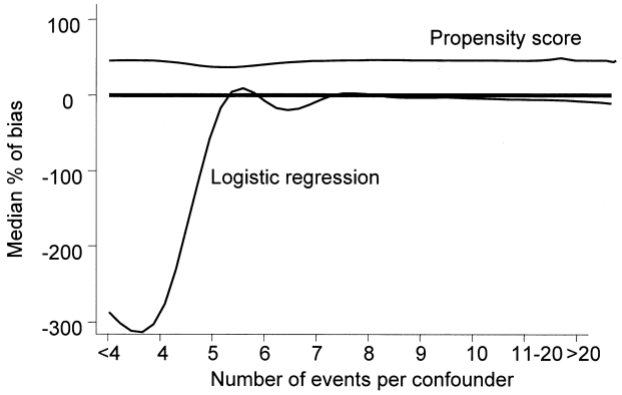
Figura 1 – Como modelos de propensão funciona.



Fonte: MasterCardAdvisors (2017)

temos pelo menos 6 ou mais eventos por variáveis de confusão.

Figura 2 – Viés por Número de eventos por Confounder.

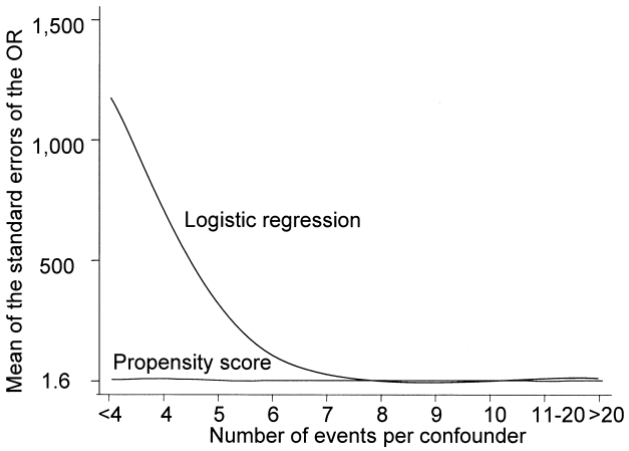


Fonte: Soledad Ray Boston (2003)

Entretanto, modelos baseado em escores de propensão possui algumas limitações, as mais comuns são erros de especificações, modelos de classificação com mais de duas classes, inflexibilidade com valores faltantes e problemas com relações não-lineares, conforme cita em seu estudo (ZHAO; FAN, 2016).

Modelos baseados em RF pode superar as desvantagens descritas acima devido a média, de suas saídas, sendo o conjunto de *Decision Tree* não-paramétrica podemos contornar o problema da não-linearidade, além de se manter estável conforme aumenta o número de dimensões e facilmente introduzida para métodos de processamento em paralelo,

Figura 3 – Erro médio padrão por Número de eventos por confounder.



Fonte: Soledad Ray Boston (2003)

relata (R. KARAMPATZIAKIS N., 2008). Porém, apesar das vantagens da RF, como a alta interpretabilidade de seus resultados, robustez a outlier e a diferentes magnitudes das variáveis, obter a árvore perfeita pode ser uma tarefa computacionalmente difícil, devido aos custos algoritmos de otimização como otimização Bayesiana, *Grid Search* e *Random Search* (GRUS, 2016).

2.3 One-Class Classification

Para que possamos otimizar modelos de classificação é necessário realizar algum nível de esforço computacional, onde diferentes métodos foram propostos com uma abordagem de algoritmos com o proposito de diminuir o erro gerado pelos algoritmos. O algoritmo de *one-class classifications* se difere dos modelos de classificação convencionais devido a utilização somente das classes positivas e majoritárias no conjunto de treinamento, de forma que ensinamos os modelos a identificar somente a classe predominante, logo tudo que não se encaixa na classe comum majoritária é classificado como negativa ou *outlier* (TAX, 2001). O objetivo do *One-class* é identificar as fronteiras em torno da classe majoritária de forma a discriminar entre classe positiva e classe negativa (TAX, 2001). A fronteira de decisão pode ser alocada em qualquer lugar do espaço das variáveis independentes, quanto maior a complexabilidade dessas fronteiras maior a quantidade de informação necessárias para melhorar na discriminação, logo para que o modelo possa discriminar as classes dos objetos, devemos utilizar variáveis características com um poder de discriminação suficiente de acordo com a complexidade do problema (TAX, 2001).

A principal vantagem dos métodos de *One-Class Classification* é evitar a estimativa de densidade de probabilidade completa, de forma a não ser requerida uma probabilidade a priori ou sendo suficiente apenas uma amostra de nossa população, sendo possível aprender com a estrutura dos dados quando a densidade da *target* é desconhecida, característica

dos modelos de aprendizado de máquina não-supervisionados (TAX, 2001).

Uma das motivações para a utilização de modelos baseados em *One-Class Classification* é a dificuldade de encontrar um *dataset* com classes balanceadas, evitando assim o emprego de técnicas de geração ou exclusão de dados na etapa de pré-processamento para realizar o treinamento do modelo, assim como os vieses devido a *dataset* desbalanceados (KEMMLER ERIK RODNER, 2013).

Os algoritmos de *one-class classification* pode utilizar de alguns métodos conhecidos, de forma a modificar a criação das fronteiras de decisão de acordo com o método, segue alguns exemplos nas sub-seções.

2.3.1 Isolation Forest (IForest)

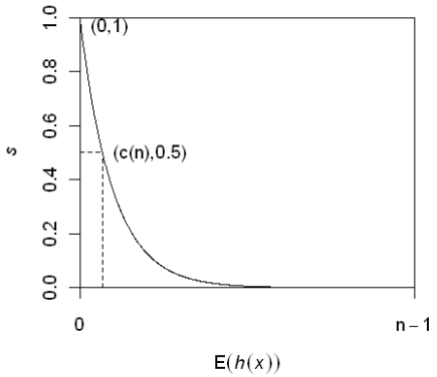
O objetivo dos métodos de IForest é criar *Isolation Trees* no conjunto de dados em questão, criando várias árvores simplificadas de forma a isolar possíveis *outliers*, construindo um perfil de normalidade e excluindo os que não se encaixam. No artigo principal da IForest são apresentadas apenas duas variáveis para esse algoritmo, o número de árvores e o tamanho do subconjunto de dados. O ideal para esse tipo de abordagem é processar pequenos subconjuntos de dados de forma a evitar o efeito de *Swamping* e mascaramento das *targets* negativas, onde o *Swamping* é o ato do modelo considerar um dado normal como anomalias, o que pode ser evitado utilizando um *sample size* pequeno, é sugerido um valor de 256 observações na amostra, (LIU, 2008).

O Iforest utiliza o conceito de comprimento do caminho para calcular o score de anomalias, onde quanto menor o comprimento do caminho maiores as chances de ser uma anomalia, representando pontos isolados no conjunto de dados não sendo necessário muitas divisões no espaço de *features*. Na imagem 4 é mostrado a relação do comprimento do caminho esperado *E(h(x))* e o score de anomalia *s*, onde quando o E(h(x)) é igual a média do comprimento do caminho de todas as *Itree* o score de anomalia é igual a 0.5, (LIU, 2008).

(LIU, 2008) sugere que para encontrar os pontos mais anômalos devemos ordenar os dados usando o score de anomalias, no qual quanto maior o score maior a probabilidade de ser um *outlier*.

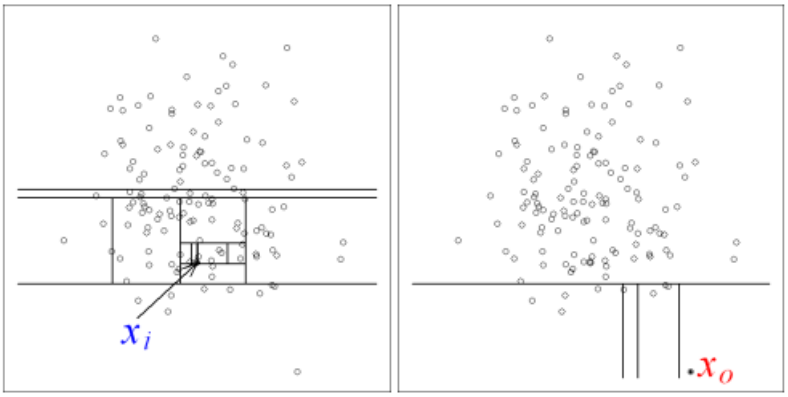
Uma das vantagens do algoritmo *iForest* está em seu processamento não sendo preciso computar distâncias ou medidas de densidade para detectar essas anomalias, e com isso sua complexidade do modelo é aproximadamente linear *big O notation*, O(ntlog).(LIU, 2008). A figura 5 mostra como identificar um outlier devido às repartições criadas pelas ramificações das árvores.

Figura 4 – Relação comprimento do caminho x score de anomalia.



Fonte: Liu (2008)

Figura 5 – Identificação de anomalias por Isolation forests.



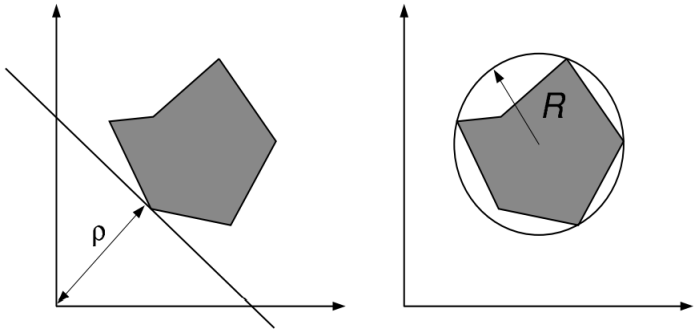
Fonte: Liu (2008)

2.3.2 One-Class Support Vector Machine (OCSVM)

(SCHOLKOPF JOHN C. PLATT, 2001) introduziu o *one-class classification Support Vector Machine*, também conhecido como OCSVM, como um classificador que separa os dados de treino utilizando a maior margem, ou seja, distância entre hiperplano e os vetores de suporte, no qual foi utilizado o *kernel Gaussian RBF* que centraliza os dados na origem do espaço, projetando-os em uma superfície de metade de uma hiperesfera de raio unitário, conforme a figura 6. Assim não podemos considerar a origem como membro de uma classe, logo um modelo com um bom fator discriminante não irá utilizar a origem do espaço das *features*, no entanto utilizará o isolamento dos dados de entrada do restante dos dados (BOUNSIAR, 2014). Na figura 6 mostra as 2 maneiras apartadas que o OCSVM utiliza em seu algoritmo.

A combinação de ambas técnicas nos dará a ilustração geométrica da figura 7, onde h1(x), h2(x) e h3(x) são os hiperplanos, sendo $\rho 1$, $\rho 2$, $\rho 3$ os raios das hiperesfera

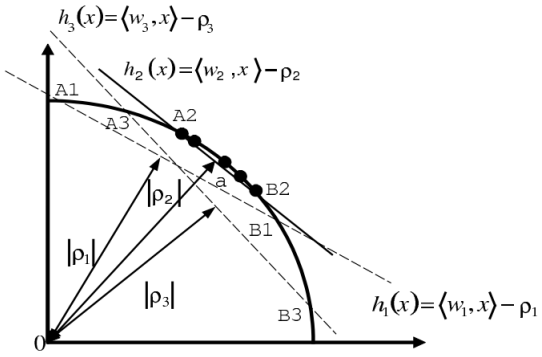
Figura 6 – O OCSVM com separação de hiperplano (esquerda) e SVDD com separação de hiperesfera (direita).



Fonte: TAX (2001)

(BOUNSIAR, 2014).

Figura 7 – Ilustração Geométrica do algoritmo OCSVM



Fonte: Bounsiar (2014)

2.4 Random Forest

A Random Forest (RF) é um dos algoritmos amplamente utilizados para classificação binária no setor financeiro. Devido a sua complexidade computacional linear, e sua abordagem que identifica relações não-lineares entre as variáveis independentes e target (JAMES et al., 2013) acaba se sobressaindo em relação à algoritmo mais clássico que captura apenas relações lineares.

É um algoritmo baseado em árvore de decisões na qual utiliza técnicas de *ensembles* e *bagging*. O RF cria aleatoriedade entre as observações e nas *features* do conjunto de dados, ao invés de procurar a melhor variável para o modelo, ele irá procurar a melhor variável dentro de uma amostra de variáveis escolhidas aleatoriamente, o que originaliza o nome, evitando o domínio de uma variável no modelo resultando em árvores distintas reduzindo a variância geral do modelo (GERON, 2017).

Além de conseguir predizer o algoritmo de RF consegue calcular a probabilidade de uma observações de pertencer a uma classe, na biblioteca *scikit-learn* há o método predict-proba, que é utilizado para construir um modelo de score.

2.5 Kernel Trick

Kernel é uma maneira de quantificar a similaridade de 2 observações através de uma função, as funções mais populares são lineares, polinomiais e radiais. As funções *kernel* são utilizadas para criar novas *features* utilizando as originais de forma a ajudar qualquer tipo de modelos com o objetivo de quebrar a linearidade que limita alguns algoritmos (JAMES et al., 2013).

As transformações Kernel são utilizadas no estágio de pré-processamento dos dados e é classificada como uma transformação, além da quebra da linearidade é possível capturar aspectos não-lineares, ou seja, problemas não-lineares são transformadas em lineares, beneficiando algoritmos baseados nesse comportamento, (AHMAD, 2014).

Kernels também podem ser utilizadas em árvores de decisões e métodos *ensembles* com o objetivo de tornar os modelos mais preditivos. Utilizando as transformações kernel como variáveis adicionais do modelo, onde as *kernel features* serão selecionadas somente no níveis mais altos da arvore com um poder de discriminação mais elevado que as variáveis originais, caso contrario ficaram em níveis mais baixos das arvores (AHMAD, 2014).

No caso de kernel lineares, que vem da linearidade entre duas *features*, também conhecida como sinergia, que pode ser escrito como:

$$K\left(x_i x_{i'j}\right) = \sum_{j=1}^{p} x_i x_{i'j} (2.1)$$

Kernel Polinomiais geram maiores flexibilidade em relação a linearidade dos modelos, podendo ser descrita como:

$$K\left(x_i x_{i'j}\right) = \left(1 + \sum_{j=1}^{p} x_i x_{i'j}\right)^d \tag{2.2}$$

onde d é o grau do polinômio, sempre sendo um inteiro positivo.

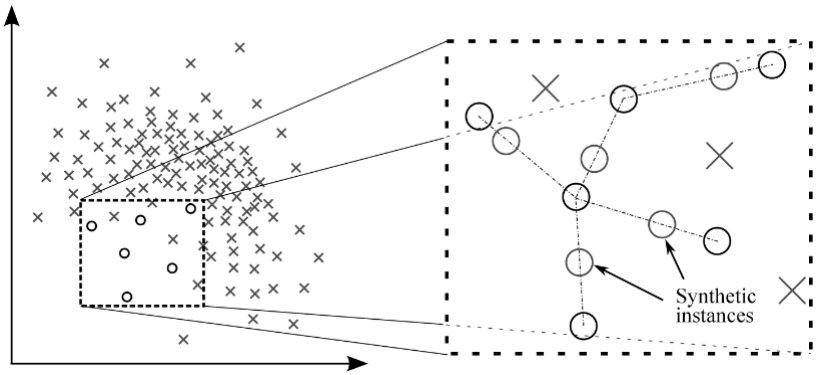
Também utilizaremos nesse trabalho kernels logaritmos com o objetivo de realizar a quebra da linearidade das *features* que serão testadas em alguns modelos.

2.6 Synthetic Minority Over-sampling Technique - Smote

A técnica *Synthetic Minority Over-sampling Technique*, mais conhecida como Smote, é uma abordagem de *over-sampling* em relação a classe minoritária de um *dataset* desbalanceado, através da criação de exemplos sintéticos ao invés da substituição dos

espaços. Os dados sintéticos são criados através do espaço de *features*, onde os dados minoritário sintéticos são criados através da linha que une dois exemplos da classe minoritária existentes, criando dados sintéticos através da interpolação utilizando o conceito de vizinhos mais próximos, de forma que se existir um empate será aleatoriamente decidido a criação ou não da observação sintética (CHAWLA et al., 2002).

Figura 8 – Smote



Fonte: https://www.datasciencecentral.com/profiles/blogs/handling-imbalanced-data-sets-in-supervised-learning-using-family

2.7 Seleção de Características

A seleção de característica é o estágio onde selecionamos quais sub-conjuntos das variáveis independentes originais entraram em nosso modelo. Basicamente é uma redução de dimensionalidade do conjunto de dados capturado, onde filtramos variáveis irrelevantes, redundantes ou variáveis ruídos nas quais não aumentam o poder de discriminação de nossos modelos, sem o deterioramento da performance preditiva (LIU et al., 2010).

Alguns métodos mais comuns utilizados para a remoção das variáveis são: variância nula, variáveis altamente correlacionadas ou associadas, ganho de informação por Gini ou entropia, teste estatístico ANOVA, Distância Interquartílica, IV entre outras dependendo da abordagem linear ou não-linear do modelo que receberá as variáveis.

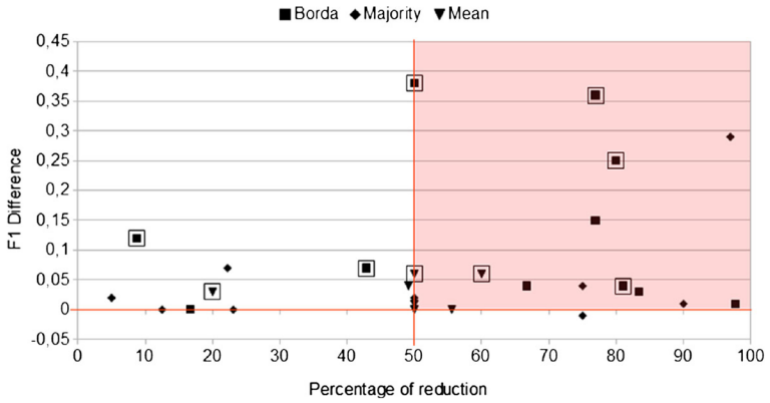
2.7.1 Rank Aggregation

A agregação por ranqueamento é utilizado para combinar diferentes técnicas de FS obtendo as melhores variáveis de acordo com o consenso, produzindo um novo ranqueamento geral. Uma das maneiras mais comum de criar esse novo ranqueamento de característica são por métodos de comitê, onde os mais comuns são: média, votos majoritários, medianas, borda (LORENA; CARVALHO; LORENA, 2015).

O método de agregação borda é calculado pela soma dos *rank* de cada técnica para cada *feature*, onde a menor soma será a melhor *feature*. O método borda obteve

melhores resultados para a maioria dos conjuntos de dados testados para algoritmos de OC, mantendo ou até mesmo melhorando a performance (LORENA; CARVALHO; LORENA, 2015).

Figura 9 – Porcentagem de seleção de característica versus ganho de F1 após diferentes agregações



Fonte: Lorena, Carvalho e Lorena (2015)

2.8 Métricas de Classificação Binária

Para avaliarmos como nossos modelos estão performando ou qual é o poder de predição devemos utilizar métricas específicas para cada tipo de modelo, assim como considerar as características do conjunto de dados. Em conjuntos de dados desbalanceados devemos ter ainda mais cuidado para não considerar uma métrica que distorce a performance real do modelo em questão (BRANCO; TORGO; RIBEIRO, 2016).

2.8.1 Receiver Operating Characteristics (ROC)

Na teoria da comunicação a métrica ROC nos mostra os dois tipos de erro para todos os possíveis *thresholds*, onde conforme variamos o threshold acabamos influenciando a taxa positiva (sensitivity) e a taxa de falso positivo (1-specificity) (JAMES et al., 2013).

Onde os tipos de previsões são definidos com a figura 10.

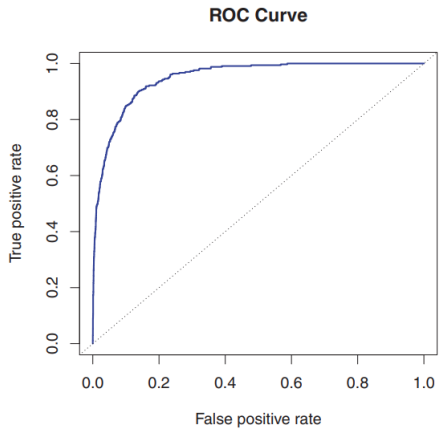
Figura 10 – Tipos de previsões.

Name	Definition	Synonyms
False Pos. rate	FP/N	Type I error, 1−Specificity
True Pos. rate	TP/P	1−Type II error, power, sensitivity, recall
Pos. Pred. value	TP/P*	Precision, 1−false discovery proportion
Neg. Pred. value	TN/N*	

Fonte: James et al. (2013)

A performance geral de um classificador, ao longo de todos os *thresholds* é calculado pela área sob a curva ROC, mais conhecida como AUC, onde um AUC igual a 1 significa um classificador perfeito e um AUC igual ou inferior a 0,5 é considerado um classificador aleatório (JAMES et al., 2013)

Figura 11 – ROC



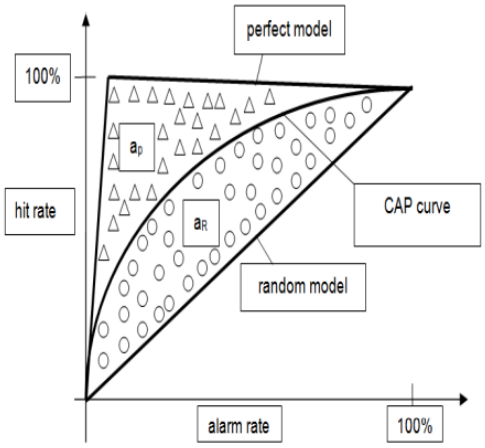
Fonte: James et al. (2013)

2.8.2 Coeficiente Gini

O coeficiente Gini é uma método estatístico que indica o poder de discriminação de um modelo, na qual é usada para comparar diferentes modelos de propensão.

A curva CAP é criada utilizando a distribuição da porcentagem acumulada de todo o público-alvo de nosso modelo (eixo X) e a porcentagem cumulativa da target positiva (eixo Y) (BELÁS; CIPOVOVÁ, 2013).

Figura 12 – Curva CAP



Fonte: Belás e Cipovová (2013)

onde a_r é a área localizada entre o *rating* atual e o *rating* da aleatoriedade e a_p é a

área entre *rating* aleatório e o *rating* perfeito. Dessa forma o Gini pode ser calculado pelas seguintes equações:

$$Gini = \frac{a_r}{a_p} \tag{2.3}$$

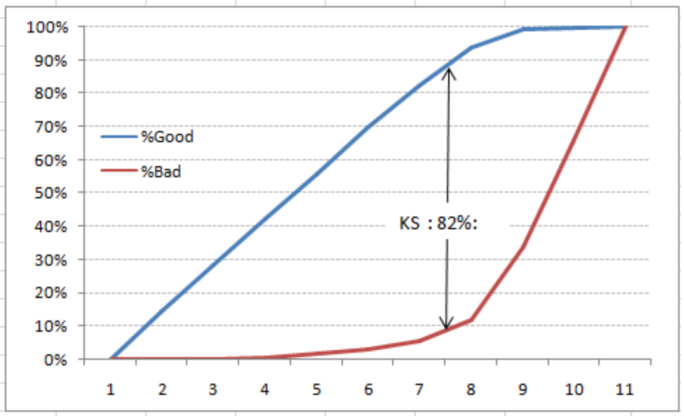
$$Gini = 2 * AUC - 1 \tag{2.4}$$

O coeficiente de Gini possui um *range* de -1 a 1, e quanto mais próximo de 1 maior o poder discriminativo do modelo.

2.8.3 Kolmogorov–Smirnov

A métrica de Kolmogorov–Smirnov , mais conhecida como KS, é usada em classificação binaria como uma métrica de dissimilaridade para que possamos identificar o poder de discriminação dos modelos de scores, na qual é usada o score das classes 0 e 1 para criar duas funções de distribuição acumulada (CDFs), onde teremos o valor do KS como a maior distancia vertical entre essas duas CDFs (ADEODATO; MELO, 2016).

Figura 13 – Kolmogorov–Smirnov



Fonte: https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/

3 DESENVOLVIMENTO

3.1 Considerações Iniciais

Neste capítulo, iremos discorrer sobre a aplicação e comparação dos algoritmos RF, IForest e OCSVM em relação a modelagem de propensão. Nesse tópico será descrito o problema junto com o conjunto de dados utilizado, o pipeline de dados e as análises de desempenho dos modelos.

3.2 Descrição do Problema

Após entendido como os modelos de propensão são aplicados, iremos buscar nesse trabalho maneiras alternativas e eficientes de calcular a propensão dos clientes a contratarem um tipo de serviço ou solução, ou seja, dado um conjunto populacional de clientes ou *prospect*, buscaremos identificar quem são os clientes que constituem o grupo com maior probabilidade de aderência a um serviço, de forma que essa amostra possa ser alvo de campanhas e ofertas, de maneira a otimizar nossos retornos e não incomodar aqueles que não possuem nenhum tipo características a utilizar tais serviços.

A seleção dos clientes que serão alvos de campanhas será abordada como métrica de comparação entre os modelos, no qual modelos baseados na probabilidade de conversão serão ordenados de forma a considerar quintis de separação entre as faixas abordadas, de modo que clientes pertencentes ao primeiro quintil serão altamente propensos a contratação e conforme o quintil aumenta diminuímos essa probabilidade.

Modelos baseados em *One-Class Classification* não possui a probabilidade de predição de cada classe, como são baseados na detecção de *outliers* esse tipo de modelagem nos trás o score da amostra, onde cada algoritmo adota um sentido para o score de anomalia, ou seja, o score de anomalia pode ser crescente ou decrescente de acordo com a probabilidade de ser um outlier, adotando para o nosso caso, quanto mais propenso em ser uma anomalia maior a probabilidade de aderir a campanha. Com base nisso, os quintis mais baixos serão construídos com os scores de anomalias mais baixos.

A métrica de comparação entre esses modelos será no resultado trazido até o segundo quintil, de forma a focarmos nos clientes realmente propensos em nosso conjunto de teste, onde temos 30 % do *dataset* no qual representa uma volumetria de aproximadamente 23 mil clientes, sendo apenas 900 propensos a contratação.

Nosso conjunto de dados aborda clientes que aderiram a um churn, mostrando insatisfação com os serviços prestados e aqueles que não tiveram churn, trata-se de uma base de dados com *target* binária, onde os valores são 1's representam aqueles que aderiram

ao churn.

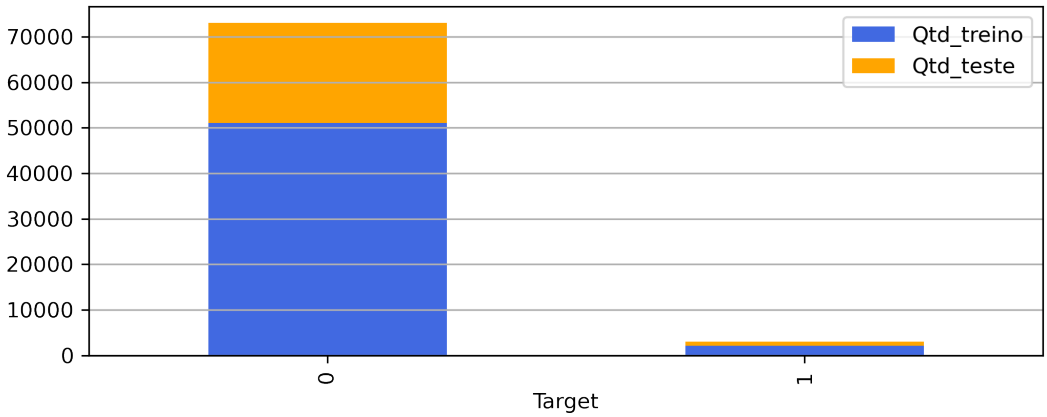
A estrutura do conjunto de dados que será utilizada aborda 76 mil clientes, com 369 colunas contendo informações pessoais e bancárias. Por motivos de sigilo a etapa de exploração e análise de dados da base utilizada não será divulgada, tendo as *features*, assim como a importância e nível de discriminação das mesmas, não citadas.

A base de dados será dividida em treino e teste, com a proporção 70/30 respectivamente, em valores teremos 53214 para treinamento e otimização dos modelos e 22786 para teste e validação, assim como mensurar os ganhos reais.

Temos um conjunto de dados desbalanceado,ou seja, há uma classe majoritariamente dominante, muito comum em conjuntos de dados reais, apesar disso não é um impeditivo para trabalharmos com conjuntos de dados com essa característica, porém devemos tomar as devidas atenções em relação às técnicas, modelagem e métricas empregadas nas quais discorreremos de maneira mais detalhadas.

A proporção entre as classes e a visão treino e teste é apresentada na figura abaixo.

Figura 14 – Proporção treino-teste classes desbalanceadas.



Fonte: Elaboração do autor(2021).

Onde temos na figura 14 uma proporção de 96 % sendo classe majoritária de clientes não-contratantes (classe 0) e apenas 4 % como classe 1 de clientes contratantes.

3.3 Atividades Realizadas

Para que possamos mensurar o impacto dos modelos de *One-Class Classification*, em relação a modelos mais utilizados para o proposito de propensão de clientes, utilizamos o algoritmo RF como *Baseline* e os algoritmos de OCC como desafiantes. Foram criados 4 tipos de *feature selection* e 10 modelos baseados nos 3 algoritmos RF, IForest e OCSVM.

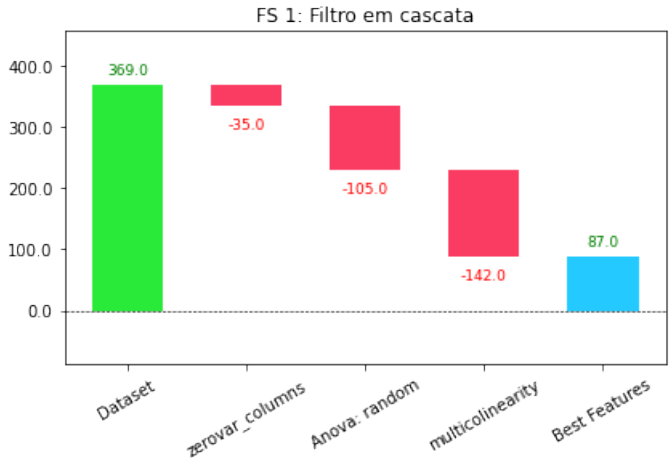
3.3.1 Feature Selection

Inicialmente foi desenvolvimentos 4 *FS*, nos quais foram utilizados obedecendo a forma de funcionamento de cada algoritmos, foram eles: (1) Filtro cascata, (2) Kernel trick, (3) Curtoses e (4) Borda Aggregation Ranking. O objetivo dos *FS* foi tentar melhorar o poder de discriminação do modelo, assim como a ordenação dos scores de propensão reduzindo as dimensões do conjunto de dados original e melhorando a explicabilidade dos modelos.

3.3.1.1 Feature Selection 1: Filtro em cascata

O *FS 1* foi projetado para reduzir o número de variáveis principais do conjunto de dados originais, reduzindo de 369 para 87 variáveis. Foram utilizadas 3 filtros sequenciais para a redução da dimensionalidade, o primeiro filtro foi realizado removendo as variáveis com variância nula, ou seja, valores constantes ou com baixa variação. O segundo filtro foi feito calculando o teste estatístico ANOVA entre a variável e a *Target* onde o *threshold* foi o Anova de uma variável aleatória. O terceiro filtro foi a remoção das variáveis com multicolinearidade entre si, ou seja, removemos as *features* com um correlação Pearson igual ou superior a 0.75, as quedas de cada filtro está representada na figura 15.

Figura 15 – FS 1: Cascade Feature Selection



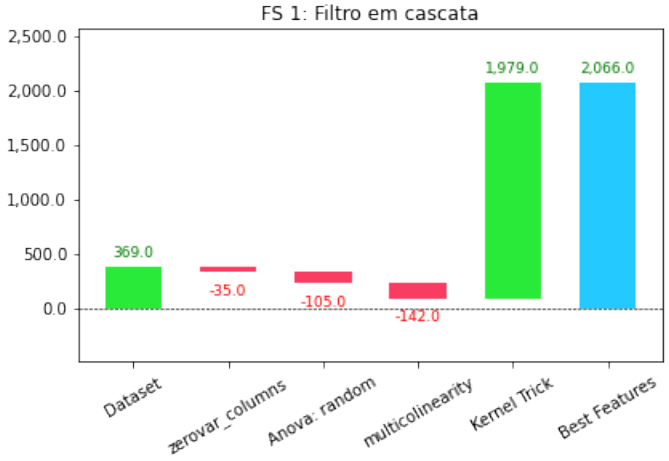
Fonte: Elaboração do autor(2021).

3.3.1.2 Feature Selection 2: Filtro em cascata + Kernel Trick

O *FS 2* foi criado não apenas com foco em redução da dimensão, mas sim agregando novas variáveis com o Kernel Trick. Aplicando o kernel trick sequencialmente ao *FS 1*, de forma a aumentarmos o número de variáveis de entrada no modelo. Essa estrutura tem a ideia de selecionarmos as melhores variáveis e a criação de nova variáveis lineares e não-lineares em relação a *Target*. Foi utilizado para a criação das novas variáveis Kernel

lineares, potência de 2 e o logaritmo das variáveis com o objetivo da transformação não-linear.

Figura 16 – FS 2: Cascade Feature Selection + Kernel Trick

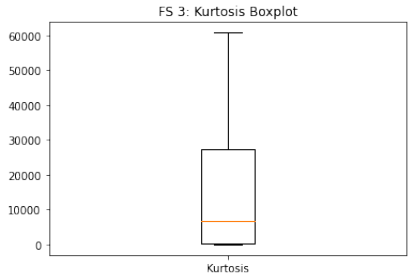


Fonte: Elaboração do autor(2021).

3.3.1.3 Feature Selection 3: Filtro Curtose

Para que possamos identificar as variáveis que possuem um determinado grau de quantidade de *outliers* podemos utilizar o teste estatístico de Curtose. Conforme cita (LIU, 2008), podemos utilizar as variáveis com elevado nível de Curtose nos algoritmos de OCC. Considerando todas as variáveis criadas pelo *FS 2*, calculamos o Curtose de cada variável e ranqueamos em ordem decrescente de forma a selecionar 25 % dos Curtose mais altos.

Figura 17 – Distribuição da Curtoses em relação a todas as variáveis da FS 2.

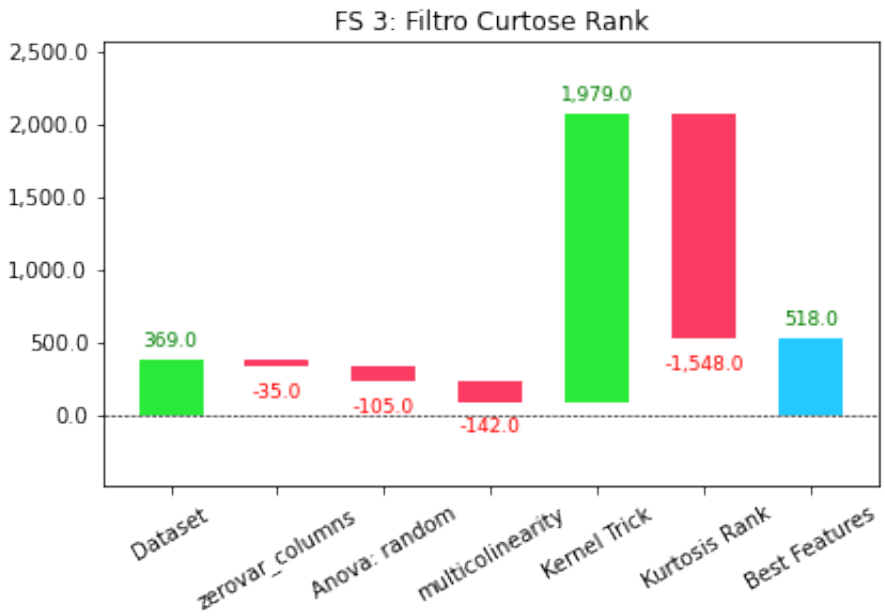


Fonte: Elaboração do autor(2021).

3.3.1.4 Feature Selection 4: Borda Aggregation Rank

Com a finalidade de termos um FS consensual foi utilizado a técnica *borda aggregation*, como citado na revisão bibliográfica. Em nosso FS 4 utilizamos 5 tipos de *Feature Ranking* de forma que ranqueamos as melhores variáveis de acordo com todas as técnicas utilizadas. Foram aplicadas as seguintes FS para a agregação: (1)Spectral Rank, (2)Anova

Figura 18 – FS 3: Filtro Curtose Rank



Fonte: Elaboração do autor(2021).

Rank, (3)Pearson Rank, (4)Gini Feature Importance Rank e (5) Curtoses Rank. Borda Aggregation é a ação de criarmos fronteiras utilizando a soma dos ranks, dessa forma conseguimos organizar de forma justa a importância daquela variável para diferentes tipos de abordagem, se uma variável foi muito boa em um dos ranqueamento, tenderá a ter um bom ranqueamento geral, mas não é garantido, dependendo fortemente das outras técnicas.

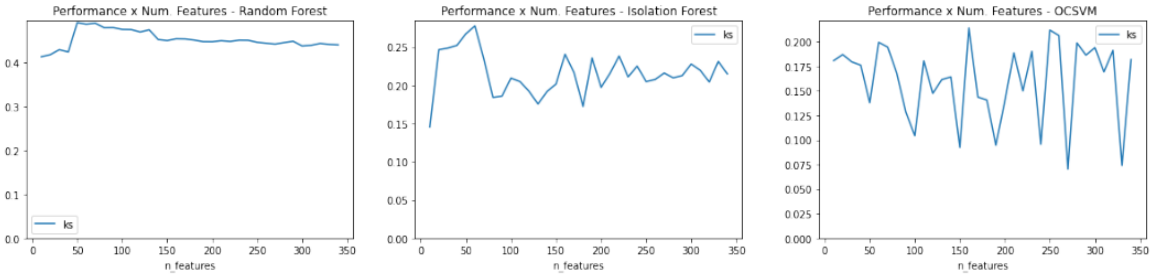
Figura 19 – FS 4: Borda Aggregation Rank

Features	spectral_ranking	anova_ranking	pearson_ranking	FI_ranking	kurtosis_ranking	Borda_agg	reordered_ranks
var1	96	76	6	1	125	304	1
var2	98	100	69	18	175	460	2
var3	99	158	1	28	198	484	3
var4	74	92	209	3	107	485	4
var5	97	93	146	15	176	527	5
...	...	...	...	...	...	...	...

Fonte: Elaboração do autor(2021).

Uma das desvantagens do método acima é não ter um *threshold* especificado para que possamos adotar e filtrar o ranqueamento, assim foi testado para cada algoritmo com hiper-parâmetros *default* qual é a quantidade promissora de variáveis seguindo o ranqueamento de forma ascendente, rank 1 para rank n. Como resultado, conseguimos otimizar o KS para os modelos de RF, IF, OCSVM com 50, 60 e 160 *features* respectivamente.

Figura 20 – FS 4: Ks para diferentes quantidade de variáveis utilizando ranqueamento borda agg.



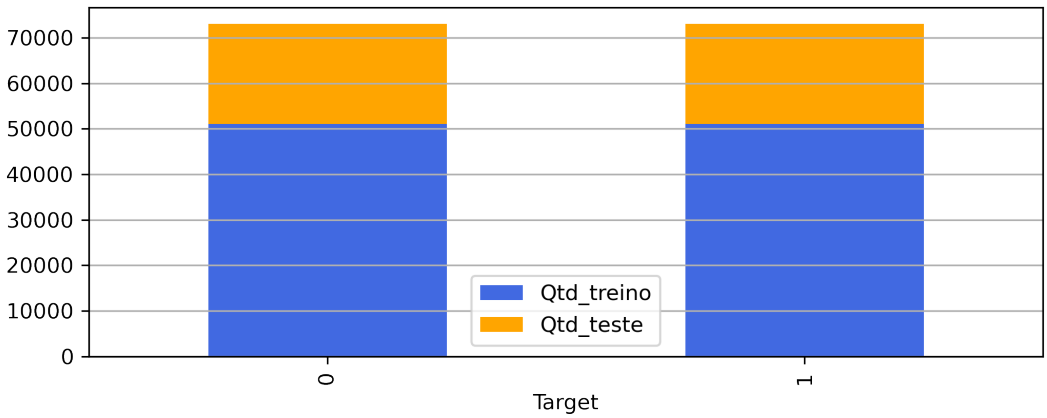
Fonte: Elaboração do autor(2021).

3.3.2 Pipeline de Dados dos Modelos

3.3.2.1 Pipeline RF

A primeira etapa para a construção dos pipelines é a FS, na qual seleciona as variáveis mais importantes com abordagens diferentes para cada tipo de FS. Na segunda etapa para a construção do modelo de RF foi utilizada a técnica Smote para que possamos equilibrar o conjunto de dados desbalanceado para que a target majoritária não domine o algoritmo. O Smote consiste em criar dados sintéticos utilizando interpolação dos dados da classe minoritária, de forma a igualar as proporções de ambas as classes. A figura 21 abaixo retrata o resultado deste pré-processamento dos dados resultando no *dataset* contendo as classes balanceadas.

Figura 21 – Proporção treino-teste classes balanceadas.

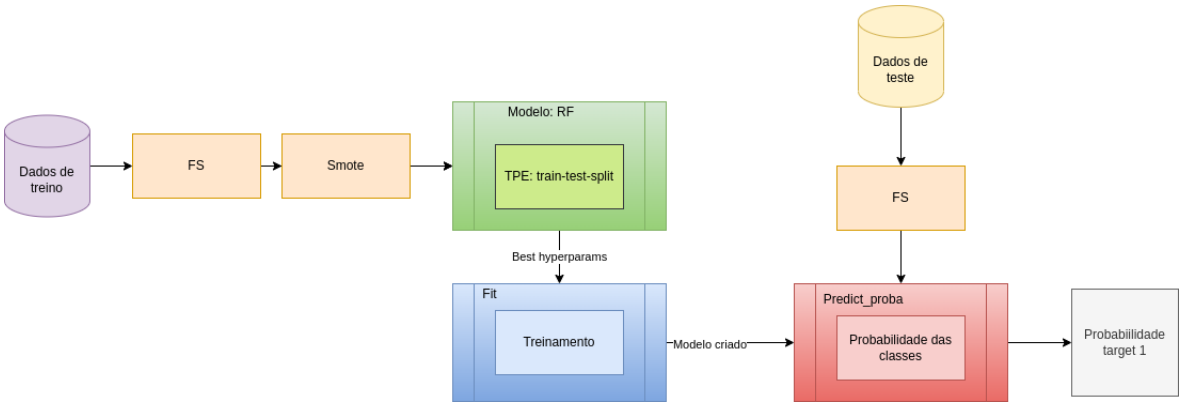


Fonte: Elaboração do autor(2021).

Posteriormente foi realizado a otimização do algoritmo de RF utilizando otimizador TPE Bayesiano que consiste em criar distribuições de probabilidade para cada hiperparâmetro construindo um modelo Bayesiano com as distribuições de forma a utilizar somente aqueles com alta probabilidade de otimização na função de custo (BERGSTRA et

al., 2011). Seguido do treinamento do modelo com os hiper-parâmetros otimizados. Com o modelo ajustado, utilizamos a mesma FS para selecionar as variáveis do conjunto de dados de treinamento, seguida do método de predição da probabilidade da classe 1, que será usado para realizar escoragem das observações.

Figura 22 – Diagrama Pipeline RF



Fonte: Elaboração do autor(2021).

Foram criados 4 modelos seguindo esse pipeline: (1)RF+FS1, (2)RF+FS1+score(IForest e OCSVM), (3)RF+FS4 e (4)RF+FS4+score(IForest). Onde os modelos 2 e 4 foram utilizado score dos modelos IForest e OCSVM como variáveis do modelo RF.

O Isolation Forest foi escolhido para realizarmos o Stacking devido a sua simplicidade computacional linear O(n), sendo muito mais simples que o modelo de *Baseline Random Forest* com complexidade O(nlog(n)dk) e ao Benchmark dos desempenhos dos algoritmo para diferentes algoritmo de detecção de *outliers* realizado pelos criadores do pacote de detecção de anomalias *Python Outlier Detection package (PyOD)*, conforme a tabela de desempenho abaixo.

Figura 23 – Comparação de performance ROC Algoritmos de Outlier Detection

*conditional formatting is a row-wise comparison

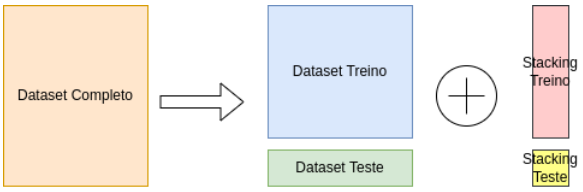
ROC Performances (average of 10 independent trials)

Data	#Samples	# Dimensions	Outlier Perc	ABOD	CBLOF	FB	HBOS	IForest	KNN	LOF	MCD	OCSVM	PCA
arrhythmia	452	274	14.60	0.7688	0.7835	0.7781	0.8219	0.8005	0.7861	0.7787	0.779	0.7812	0.7815
cardio	1,831	21	9.61	0.5692	0.9276	0.5867	0.8351	0.9213	0.7236	0.5736	0.8135	0.9348	0.9504
glass	214	9	4.21	0.7951	0.8504	0.8726	0.7389	0.7569	0.8508	0.8644	0.7901	0.6324	0.6747
ionosphere	351	33	35.90	0.9248	0.8134	0.873	0.5614	0.8499	0.9267	0.8753	0.9557	0.8419	0.7962
letter	1,600	32	6.25	0.8783	0.507	0.866	0.5927	0.642	0.8766	0.8594	0.8074	0.6118	0.5283
lympho	148	18	4.05	0.911	0.9728	0.9753	0.9957	0.9941	0.9745	0.9771	0.9	0.9759	0.9847
mnist	7,603	100	9.21	0.7815	0.8009	0.7205	0.5742	0.8159	0.8481	0.7161	0.8666	0.8529	0.8527
musk	3,062	166	3.17	0.1844	0.9879	0.5263	1	0.9999	0.7986	0.5287	0.9998	1	1
optdigits	5,216	64	2.88	0.4667	0.5089	0.4434	0.8732	0.7253	0.3708	0.45	0.3979	0.4997	0.5086
pendigits	6,870	16	2.27	0.6878	0.9486	0.4595	0.9238	0.9435	0.7486	0.4698	0.8344	0.9303	0.9352
pima	768	8	34.90	0.6794	0.7348	0.6235	0.7	0.6806	0.7078	0.6271	0.6753	0.6215	0.6481
satellite	6,435	36	31.64	0.5714	0.6693	0.5572	0.7581	0.7022	0.6836	0.5573	0.803	0.6622	0.5988
satimage-2	5,803	36	1.22	0.819	0.9917	0.457	0.9804	0.9947	0.9536	0.4577	0.9959	0.9978	0.9822
shuttle	49,097	9	7.15	0.6234	0.6272	0.4724	0.9855	0.9971	0.6537	0.5264	0.9903	0.9917	0.9898
vertebral	240	6	12.50	0.4262	0.3486	0.4166	0.3263	0.3905	0.3817	0.4081	0.3906	0.4431	0.4027
vowels	1,456	12	3.43	0.9606	0.5856	0.9425	0.6727	0.7585	0.968	0.941	0.8076	0.7802	0.6027
wbc	378	30	5.56	0.9047	0.9227	0.9325	0.9516	0.931	0.9366	0.9349	0.921	0.9319	0.9159
			mean	0.7031	0.7636	0.6767	0.7819	0.8179	0.7758	0.6792	0.8075	0.7935	0.7737
			median	0.7688	0.8009	0.6235	0.8219	0.8159	0.7986	0.6271	0.8135	0.8419	0.7962
			sd	0.2038	0.1890	0.1966	0.1866	0.1590	0.1764	0.1929	0.1738	0.1775	0.1916

Fonte: https://towardsdatascience.com/isolation-forest-is-the-best-anomaly-detection-algorithm-for-big-data-right-now-e1a18ec0f94f

O conjunto de dados e suas divisões de treino e teste ficaram conforme a figura 24 abaixo:

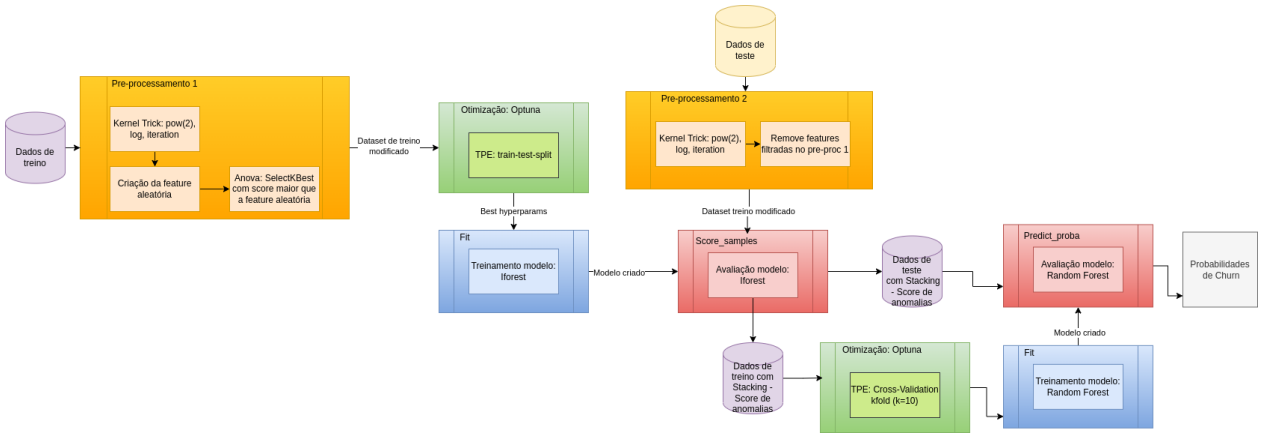
Figura 24 – Dataset com Stacking Isolation Forest Aplicado



Fonte: Elaboração do autor(2021).

Para um melhor entendimento de como ficou todo o *pipeline* dos dados vemos a figura 25, podemos perceber que o conjunto de dados com o *Kernel Trick* não está sendo utilizado diretamente no modelo de *Random Forest*, e com isso ganhamos as informações das interações das variáveis sem utilizá-las no treinamento do algoritmo principal. Tendo como vantagem em relação a outras técnicas de redução de dimensionalidade a inexistência de perda de explicabilidade.

Figura 25 – Pipeline do modelo final: Random Forest utilizando o score de anomalias

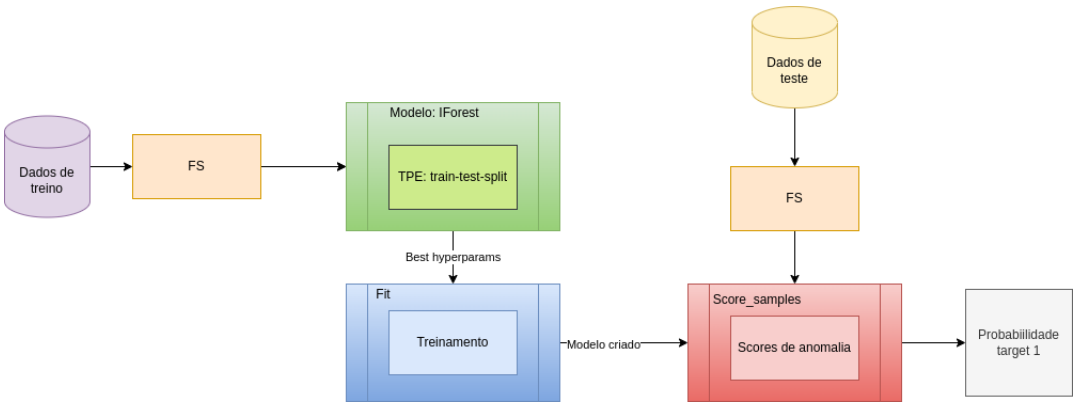


Fonte: Elaboração do autor(2021).

3.3.2.2 Pipeline IForest

O pipeline do *IForest* segue a mesma linha de raciocínio do pipeline RF, porém por ser um algoritmo *OCC* não é necessário utilizar a técnica *SMOTE* e com isso o modelo é treinado com as proporções do conjunto de dados originais, reduzindo os recursos computacionais utilizado. O *IForest* possui como saída o score de anomalia e com isso toda suas análises são baseadas nesse score.

Figura 26 – Diagrama Pipeline IForest



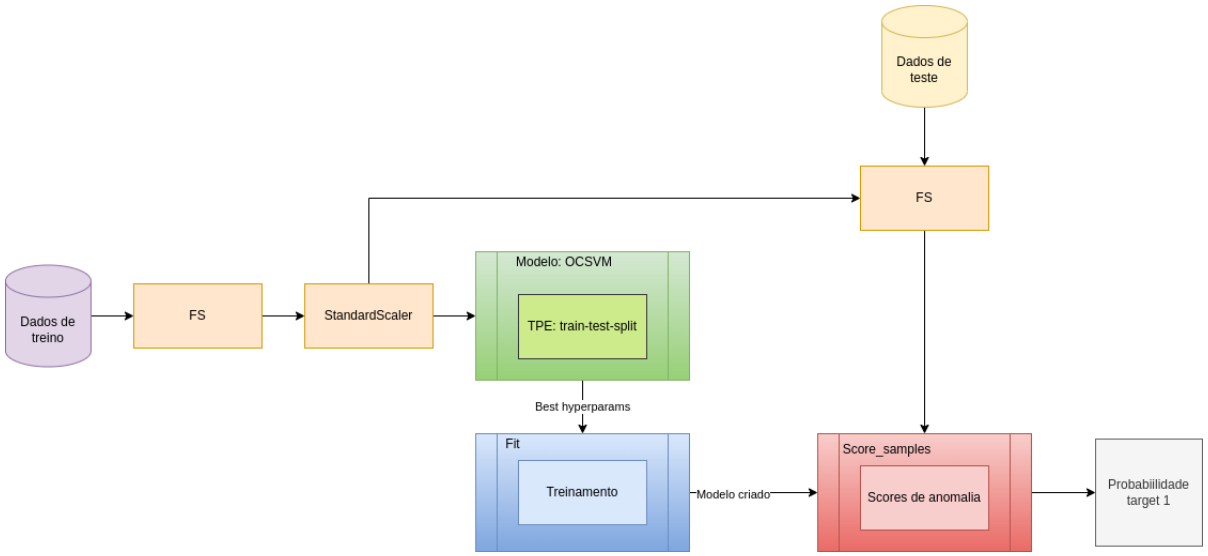
Fonte: Elaboração do autor(2021).

Foram criados 4 modelos seguindo esse pipeline: (1)IF+FS1, (2)IF+FS2, (3)RF+FS3 e (4)IF+FS4, na qual foi modificado somente o FS para cada um deles.

3.3.2.3 Pipeline OCSVM

Assim como IForest tratasse de um algoritmo de OCC, não necessitando de equilibrar o conjunto de dados de treinamento. Porém o OCSVM é um algoritmo baseado em distância e requer escalonamento dos dados de treino e teste para que não seja influenciado pela diferença de escalas entre as variáveis.

Figura 27 – Diagrama Pipeline OCSVM



Fonte: Elaboração do autor(2021).

Foram criados 2 modelos seguindo esse pipeline: (1)OCSVM+FS1 e (2)OCSVM+FS4, o FS2 não foi utilizado, pois um dos hyper-parâmetro de otimização do algoritmo de OCSVM foi diferentes Kernels, assim o processamento desses dados foram feitos na otimização.

3.3.3 Resultados Obtidos

Nesta seção, serão abordados os resultados, análises dos métodos e combinações de técnicas citadas na seção anterior, lembrando que nosso foco está no ganho de desempenho de modelos de propensões, classificação binárias, não deixando de lado os recursos e tempos de processamento cruciais em empresas *Big Data*.

A primeira abordagem foi utilizar nossos modelos RF como *baseline* utilizando como métrica de avaliação a KS. Em seguida será abordada os modelos baseados em algoritmo de OCC IForest com a mesma modelagem e avaliações do modelo anterior. Na sequência será apresentado os resultados dos modelos de OCSVM, através da análise de resultados iremos comparar a performance dos melhores modelos de cada tipo de algoritmo.

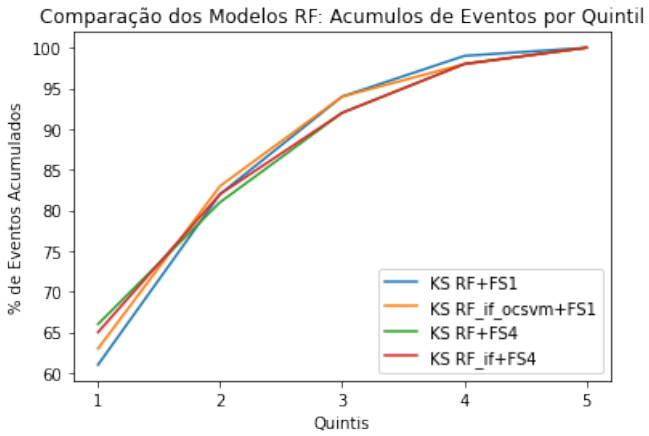
A comparação dos modelos será feita baseando em algoritmos de KS, ordenação dos scores, a taxa de *target* acumulada até o segundo quintil.

3.3.4 Modelos RF

Os modelos de RF otimizado com método TPE apresentou performance significativas com os objetivos propostos, focando em clientes com possíveis potenciais de churn até o segundo quintil. Os 4 modelos, RF+FS1, RF+if+ocsvm+FS1,RF+FS4, RF+if+FS4 obtiveram performance e ordenações bastante semelhante entre si, porém o Modelo 2 que

utiliza os 2 scores de anomalias, Iforest e OCSVM, se sobressai aos demais considerando até o segundo quintil.

Figura 28 – Comparação dos Modelos RF: Acúmulos de Eventos por Quintil



Fonte: Elaboração do autor(2021).

Obteve um acúmulo de 83 % dos churns até o segundo quintil, o que funcionaria muito bem para possíveis ações, limitando satisfatoriamente o público-alvo.

Figura 29 – Resultados Modelo 2

Quintis	min_prob	max_prob	Qtd de eventos (Target)	Qtd de não-eventos (Target)	% de eventos	% de não-eventos	% de evento acumulados	% de não-evento acumulados	KS
1	0.432001	0.871788	385	2656	63.0	18.0	63.0	18.0	45.2
2	0.294885	0.431840	119	2922	20.0	20.0	83.0	38.0	44.8
3	0.194320	0.294848	64	2976	11.0	20.0	94.0	59.0	35.0
4	0.102900	0.194288	29	3012	4.8	21.0	98.0	79.0	19.1
5	0.005709	0.102871	10	3031	1.6	21.0	100.0	100.0	0.0

Fonte: Elaboração do autor(2021).

Porém conforme visto o Modelo 2 utiliza os dois modelos de OCC, o que tornaria seu processamento inviável devido a dependência da otimização e treinamento de 3 modelos, com o consciência de viabilidade de implantação, em ambientes *Big Data*, esse modelo seria considerado um pouco mais trabalhoso para sustentá-lo, porém tudo depende do retorno resultante ao negócio. Em contrapartida, temos o modelo 4 RF+FS4 que apresenta uma simplicidade muito significativa em relação ao modelo 2, sem perdas significativas de acúmulo de *target* até o segundo quintil, tendo ainda um acúmulo superior aos demais modelos de RF considerando apenas o primeiro quintil, o que trata de um modelo bastante viável.

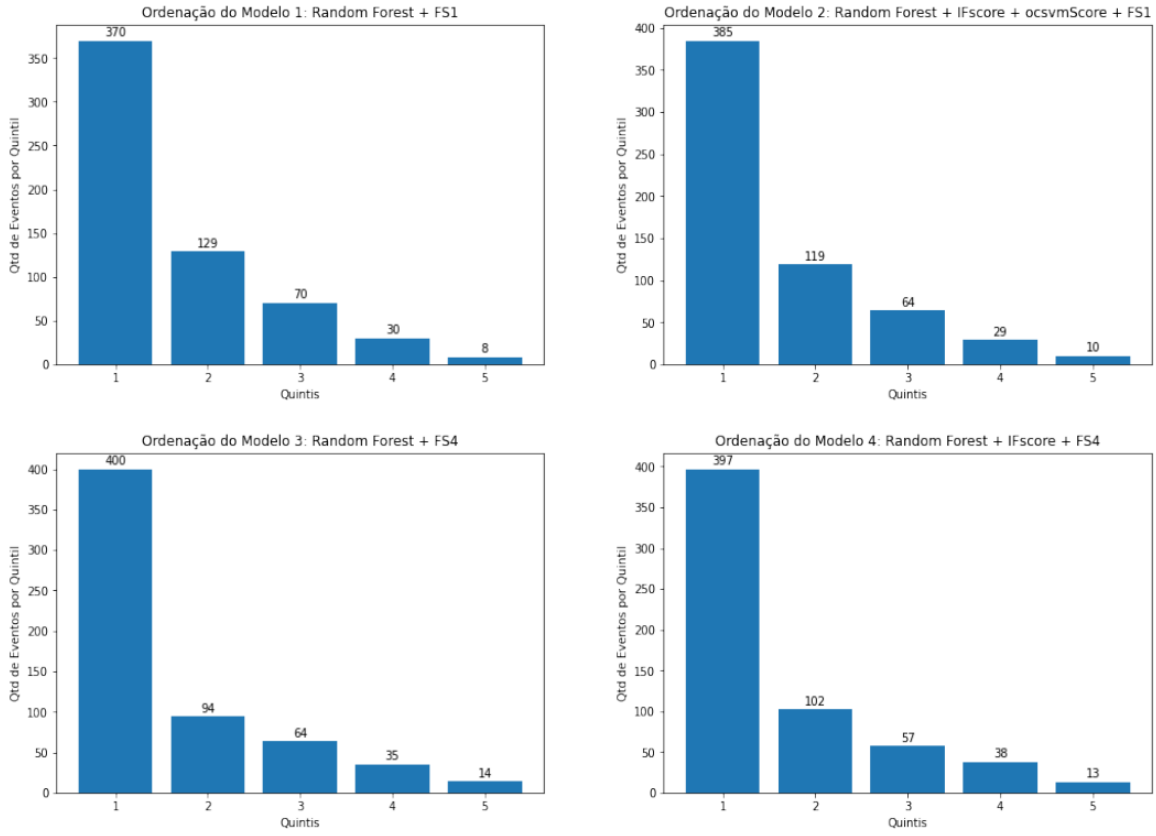
Concentrando a maioria da *target* acumulada entre os primeiros quintis, todos os modelos poderiam ser utilizados para diferentes estratégias de negócios, tomando uma postura mais agressiva, considerando quintil adjacentes ao segundo, ou mais conservadora considerando apenas o primeiro quintil.

Figura 30 – Resultados Modelo 3

Quintis	min_prob	max_prob	Qtd de eventos (Target)	Qtd de não-eventos (Target)	% de eventos	% de não-eventos	% de evento acumulados	% de não-evento acumulados	KS
1	0.410579	0.872330	400	2628	66.0	18.0	66.0	18.0	47.9
2	0.254734	0.410455	94	2960	15.0	20.0	81.0	38.0	43.1
3	0.134750	0.254727	64	2976	11.0	20.0	92.0	59.0	33.3
4	0.079343	0.134724	35	3003	5.8	21.0	98.0	79.0	18.5
5	0.033459	0.079332	14	3030	2.3	21.0	100.0	100.0	0.0

Fonte: Elaboração do autor(2021).

Figura 31 – Ordenação dos Modelos RF



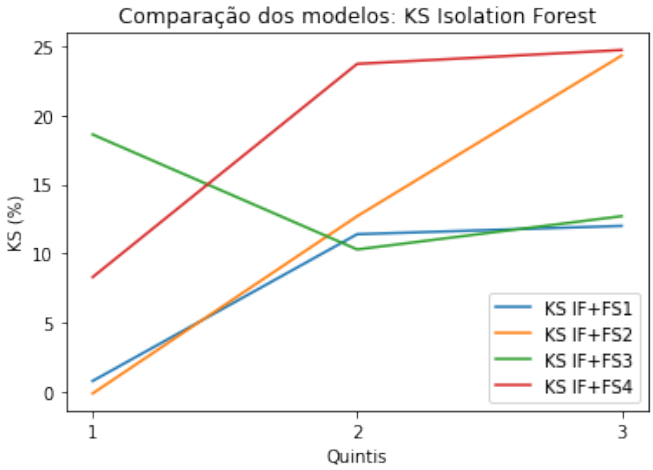
Fonte: Elaboração do autor(2021).

3.3.5 Modelos IForest

Em geral, os modelos de IForest não apresentaram resultados significantes para substituir os modelos de RF considerando a abordagem de público-alvo acumulado por quintil e utilizando a métrica KS. Por se tratar de um modelo com complexidade linear, seria um algoritmo em potencial para a substituição da RF em ambientes futuros de *Big Data*. Não apresentou uma ordenação que possa se adequar às estratégias de negócios.

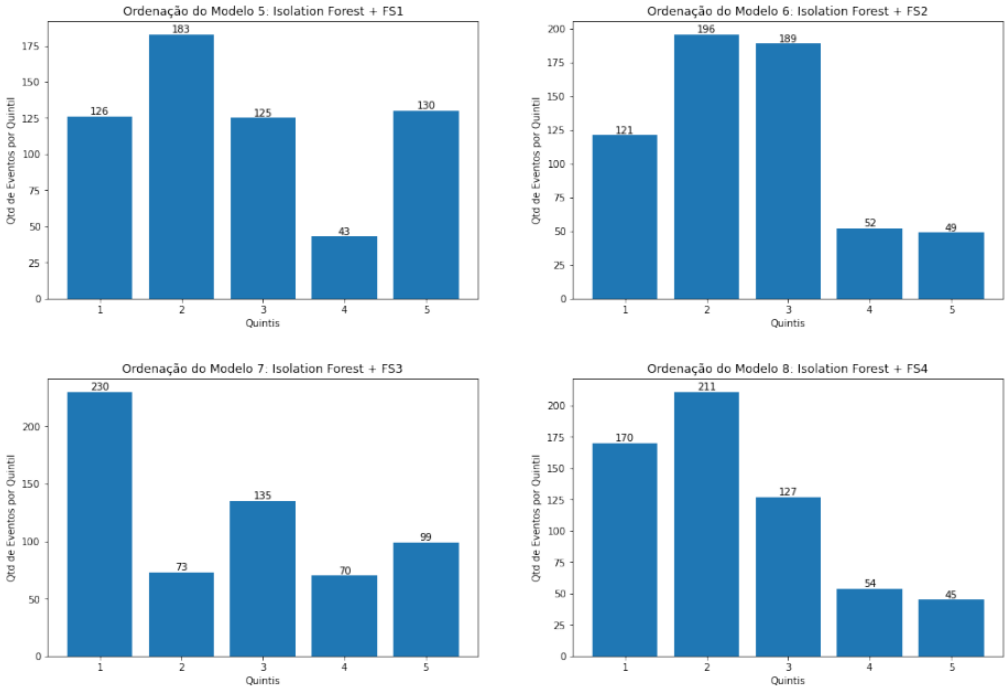
O melhor modelo de IForest foi onde utilizamos o FS4, que possui um acumulo maior que os demais modelo de Iforest, porém assim como os demais não conseguiu ordenar.

Figura 32 – Comparação dos Modelos IForest: Acúmulos de Eventos por Quintil



Fonte: Elaboração do autor(2021).

Figura 33 – Ordenação dos Modelos IForest



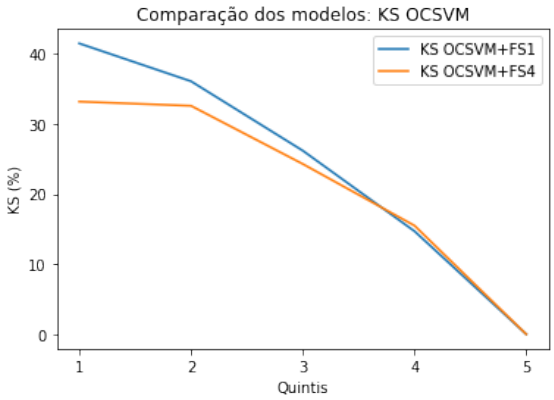
Fonte: Elaboração do autor(2021).

3.3.6 Modelos OCSVM

Os modelos de *OCSVM* apresentaram níveis satisfatórios de KS, acúmulo por quintil e ordenação, superiores ao *IForest* e inferior ao RF. O melhor modelo de *OCSVM* foi o modelo 9 utilizando apenas o FS1, não tendo resultados superiores utilizando o FS4.

A ordenação dos modelos podem acompanhar as estratégias de negócios devido a sua ordenação ao longo dos quintis, podendo ser um potencial substituto para o RF ou

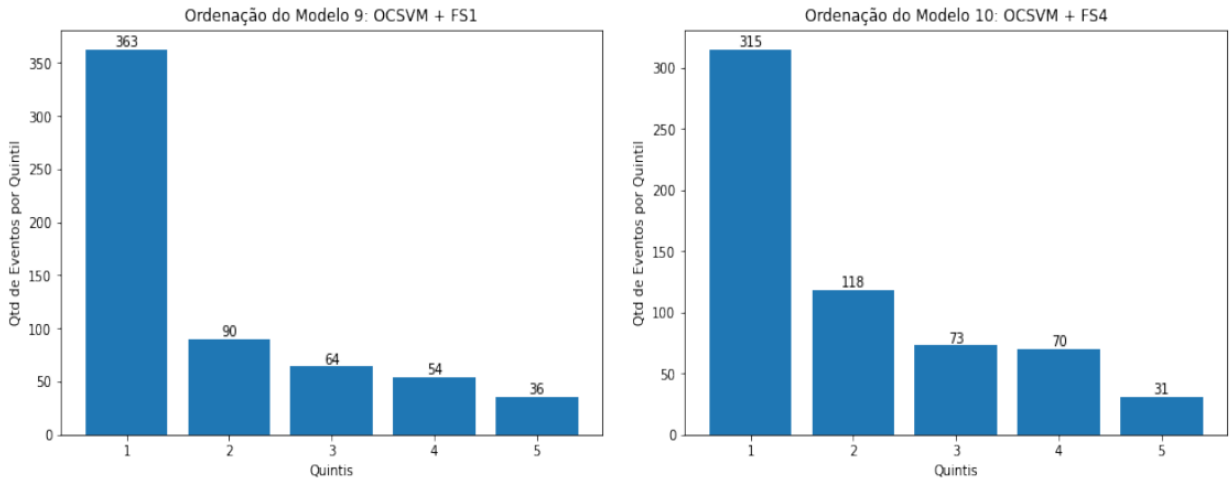
Figura 34 – Comparação dos modelos: KS OCSVM



Fonte: Elaboração do autor(2021).

até mesmo criar *ensembles* entre eles dependendo somente dos recursos computacionais disponíveis, devido sua complexidade exponencial $O(N^3)$ pode ser mais complexo ou mais simples dependendo da quantidade de árvores da RF que possui complexidade O(n*log(n)*d*k).

Figura 35 – Ordenação dos Modelos OCSVM



Fonte: Elaboração do autor(2021).

3.3.6.1 Comparação Entre os Modelos

Desconsiderando o potencial de flexibilidade de acordo com as estratégias de negocio que estão atreladas a ordenação criada pelos modelos, podemos compara-los utilizando métricas estatísticas de poder de discriminação onde o melhor Auc, e por consequência maior Gini, temos os modelos baseados em RF como os melhores algoritmos onde o modelo 4 (RF+IF+FS4) foi o melhor entre eles. Em relação aos modelos baseados em IForest, obtivemos ganhos significativos ao longo dos tipos de FS, o modelo 8 (IF+FS4)se

destacou dos demais com 29 pontos de KS. Em contrapartida, o modelo de *OCSVM* não se beneficiou do FS4, diferente dos outros algoritmos experimentados.

Figura 36 – Resultados dos modelos com as métricas AUC, Gini e KS

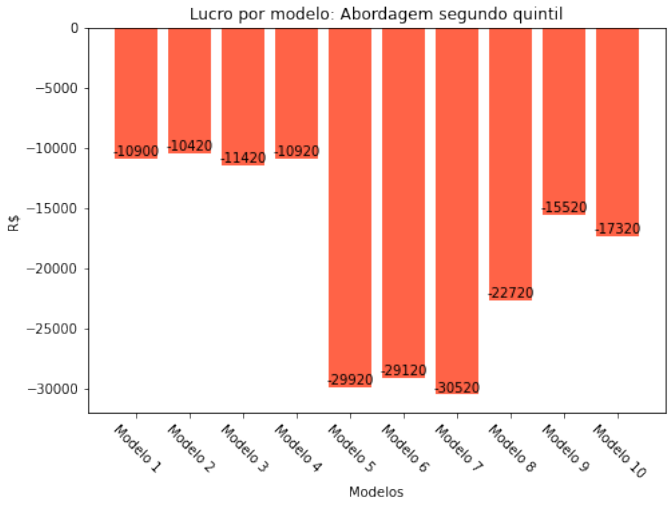
Modelo	Pipeline	Auc	Gini	KS(%)
Modelo 1	RF+FS1	0.8052	0.6104	45.87
Modelo 2	RF+IF+OCSVM+FS1	0.8083	0.6166	48.8
Modelo 3	RF+FS4	0.8022	0.6044	48.08
Modelo 4	RF+IF+FS4	0.8086	0.6171	49.73
Modelo 5	IF+FS1	0.5920	0.1840	22.75
Modelo 6	IF+FS2	0.5991	0.1981	26.86
Modelo 7	IF+FS3	0.6018	0.2032	20.26
Modelo 8	IF+FS4	0.6417	0.2834	29.24
Modelo 9	OCSVM+FS1	0.7375	0.4743	41.65
Modelo 10	OCSVM+FS4	0.7144	0.4289	38.14

Fonte: Elaboração do autor(2021).

3.3.6.2 Implementação dos Modelos

Para estimar o lucro obtido pelos modelos devemos considerar algumas premissas de gastos e lucros com as ações realizadas baseadas nos modelos. Considerando uma estratégia de negócios fixa, onde apenas o público-alvo de até o segundo quintil será abordado, o custo suposto por ação por cliente será de R$ 10,00 reais onde o ganho de um cliente mantido é de 100,00 reais, assim o lucro hipotético de um cliente mantido seria de R$ 90,00 reais. Com esses parâmetros preestabelecidos podemos fazer a análise de lucratividade dos modelos.

Figura 37 – Lucro dos modelos: Abordagem de 2 quintis.



Fonte: Elaboração do autor(2021).

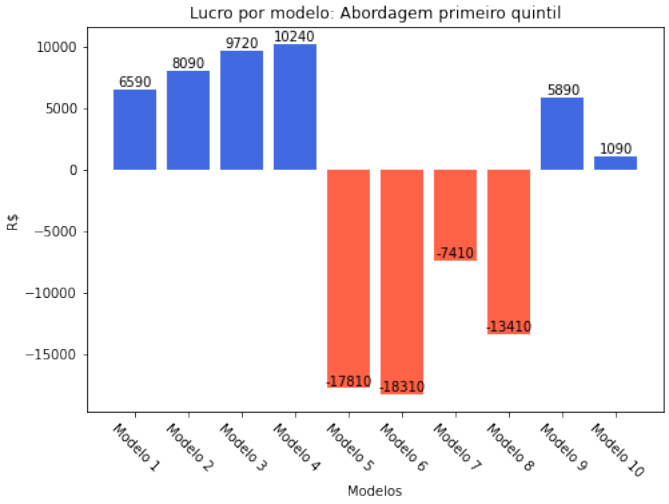
conforme a figura 38 percebemos que não foi possível obter lucro em nenhum dos modelos abordando até segundo quintil, isso se da pelo fato de que o conjunto de dados de teste não permite essa abordagem até o segundo quintil, devido a distribuição das proporções onde apenas 4% seria target 1, e mesmo considerando todos os potenciais

churns, 607 target 1's, até o segundo quintil temos um total de 5475 não-target, isso permitiria um "lucro máximo"negativo de todas as formas, -120 R$. Assim, além dos modelos, é importante entender as nuances dos negócios e proporções estabelecidas pelo conjunto de dados.

Tomando uma abordagem somente do primeiro quintil, seria possível obter um lucro máximo positivo de R$ 30290, concentrando os 607 *target* 1's em um quintil de 3041 público-alvo, onde teríamos 2434 *target* 0's.

Na abordagem de até o primeiro quintil temos que os modelos baseados em algoritmos de RF e *OCSVM* obtiveram lucros significativos, porém os modelos baseados em algoritmos de *IForest* não foi possível devido sua ordenação, onde a maior quantidade de *target* 1's se encontra no segundo quintil.

Figura 38 – Lucro dos modelos: Abordagem de 1 quintil.



Fonte: Elaboração do autor(2021).

O lucro máximo foi obtido no modelo 4, RF+IF+FS4, onde foi utilizado o algoritmo de RF com os scores do algoritmo de IForest.

4 CONCLUSÃO

Neste trabalho foi abordado diferentes tipos de aprendizado de máquina com diferentes algoritmos com foco em estimar a probabilidade de um cliente pertencer a uma classe, muito utilizado em marketing, concentrações de público-alvos e até mesmo scores de crédito.

Os algoritmos apresentados foram de aprendizado supervisionado e aprendizado de uma classe, onde para aprendizado supervisionado foi escolhido o modelo de *Random Forest* como *baseline* e os modelos desafiantes de *One-Class Classification* chamados *Isolation Forest* e *One-class SVM*. De forma que substituíssemos o algoritmo de RF por um cuja complexidade computacional fosse menor, porém os resultados desse trabalho mostram que os algoritmos de RF foram superior em todos os quesitos abordados nesse trabalho, tais como: métricas de AUC, GINI e KS, assim como diferentes estratégias de negócios utilizando a ordenação dos scores de probabilidade.

Foi abordado como pré-processamento 4 tipos de *Feature Selection*, onde os resultados mais promissores para o RF e IForest foram utilizando o *Borda Aggregation Rank*, constituída *Spectral rank*, *Anova Rank*, *Pearson Rank*, *Feature Importance: Gini Rank* e Curtoses Rank. Todavia o algoritmo de *OCSVM* apresentou melhores desempenho para FS de filtro em cascata, com as seguintes sequências: remoção das variáveis com variância nula, Anova menor que uma variável aleatória e correlação de Pearson.

Os algoritmos de OCC apresentaram um potencial de agregar informações aos modelos de RF, utilizando os scores criados por esses modelos como variáveis de entrada do algoritmo principal de RF, onde agregando o score do IForest utilizando o *FS* borda aggregation rank foi possível aumentar o lucro do RF em 9,4%.

Os modelos de *OCSVM* tiveram lucros inferiores ao RF, além da complexidade computacional do mesmo ser maior por se tratar de um algoritmo baseado em distâncias, no qual não obtivemos resultados promissores em utilizá-lo como variáveis de entrada para o RF.

Finalmente, podemos concluir com esse trabalho que para um conjunto de dados onde possuímos 4% de *target* 1's é mais viável utilizamos RF para os fins de ordenação e desempenho estatístico visando o poder de discriminação de duas classes, e caso haja recurso computacional disponível é viável testar o score criado no *IForest* como variável de entrada para os modelos baseados em RF.

4.1 Trabalhos Futuros

Sabendo que os algoritmos de OCC não obtiveram resultados superiores ao modelo *baseline* RF para um conjunto de dados desbalanceados com proporção de 4% de *target* 1's seria válido testá-los com um conjunto onde algoritmos de RF falham, ou seja, conjunto de dados com *target* 1's igual ou inferior a 1%, onde ficaria um problema semelhante a detecção de fraudes, no qual foi a motivação da criação dos algoritmos de *One-class classification* na detecção de anomalias e *outliers*.

Tendo em vista um potencial de agregar os modelos de RF com o scores gerados por modelos de OCC, como trabalho futuro é sugerido a troca de variáveis com alto Curtosis, ou seja um elevado nível de *outliers*, pelo score do modelo de OCC criado somente com essas variáveis. Assim removeremos as *features* consideradas ruídos para alguns tipos de algoritmo sensíveis a *outliers* por esses scores dos modelos de OCC.

Com o objetivo de agrupar perfis de clientes utilizando *clusterização* seria plausível utilizar os scores de modelos de OCC para reduzir o número de variáveis com elevado nível de *outliers* que influenciaram esse tipo de agrupamento, com o objetivo de reduzir o custo computacional dos algoritmos de agrupamento baseados em distâncias, como k-means, k-medoids, k-modes assim como outros algoritmos baseados em partições hierárquicas.

REFERÊNCIAS

ADEODATO, P. J.; MELO, S. B. On the equivalence between kolmogorov-smirnov and roc curve metrics for binary classification. **arXiv preprint arXiv:1606.00496**, 2016.

AHMAD, A. Decision tree ensembles based on kernel features. **Applied intelligence**, Springer, v. 41, n. 3, p. 855–869, 2014.

BELÁS, J.; CIPOVOVÁ, E. The quality and accuracy of bank internal rating model. a case study from czech republic. **International journal of mathematics and computers in simulation**, North Atlantic University Union (NAUN), v. 7, n. 2, p. 206–214, 2013.

BERGSTRA, J. et al. Algorithms for hyper-parameter optimization. **Advances in neural information processing systems**, v. 24, 2011.

BOUNSIAR, M. G. M. A. One-class support vector machines revisited. **International Conference on Information Science Applications (ICISA)**, IEEE, n. 14431822, p. 413–422, 2014.

BRANCO, P.; TORGO, L.; RIBEIRO, R. P. A survey of predictive modeling on imbalanced domains. **ACM Computing Surveys (CSUR)**, ACM New York, NY, USA, v. 49, n. 2, p. 1–50, 2016.

CHAWLA, N. V. et al. Smote: synthetic minority over-sampling technique. **Journal of artificial intelligence research**, v. 16, p. 321–357, 2002.

CHORIANOPOULOS. Effective crm using predictive analytics. **Chichester**, John Wiley Sons, 2016.

GARCÍA-OSORIO, C.; HARO-GARCÍA, A. de; GARCÍA-PEDRAJAS, N. Democratic instance selection: a linear complexity instance selection algorithm based on classifier ensemble concepts. **Artificial Intelligence**, Elsevier, v. 174, n. 5-6, p. 410–441, 2010.

GERON, A. **Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems**. Sebastopol, CA: O'Reilly Media, 2017. ISBN 978-1491962299.

GRELLERT, M. Machine learning mode decision for complexity reduction and scaling in video applications. 2018.

GRUS, J. **Data Science do Zero**. [S.l.: s.n.], 2016.

HENSMAN, J.; FUSI, N.; LAWRENCE, N. D. Gaussian processes for big data. **arXiv preprint arXiv:1309.6835**, 2013.

JAMES, G. et al. **An introduction to statistical learning**. [S.l.]: Springer, 2013. v. 112.

KEMMLER ERIK RODNER, E.-S. W.-J. D. M. One-class classification with gaussian processes. **Elsevier**, Pattern Recognition, n. 46, p. 3508–3518, 2013.

LIU, H. et al. Feature selection: An ever evolving frontier in data mining. In: PMLR. **Feature selection in data mining**. [S.l.], 2010. p. 4–13.

LIU, K. M. T. F. T. Isolation forest. **International Conference on Data Mining**, IEEE, n. 8, p. 413–422, 2008.

LORENA, L. H.; CARVALHO, A. C.; LORENA, A. C. Filter feature selection for one-class classification. **Journal of Intelligent & Robotic Systems**, Springer, v. 80, n. 1, p. 227–243, 2015.

MASTERCARDADVISORS. **Propensity Models**. [S.l.], 2017.

R. KARAMPATZIAKIS N., Y. A. C. An empirical evaluation of supervised learning in high dimensions. **American Journal of Epidemiology**, Contemp Clin Trials, n. 47, p. 96–102, 2008.

ROSENBAUM P. R., . R. D. B. **The central role of the propensity score in observational studies for causal effects**. [S.l.], 1983. v. 70, 41–55 p.

SCHOLKOPF JOHN C. PLATT, J. S.-T. A. J. S. B. Estimating the support of a high-dimensional distribution. **Neural Computation**, Massachusetts Institute of Technology, n. 13, p. 1443–1471, 2001.

SOLEDAD RAY BOSTON, J. T. F. B. L. S. M. C. Comparison of logistic regression versus propensity score when the number of events is low and there are multiple confounders. **American Journal of Epidemiology**, Johns Hopkins Bloomberg School of Public Health, v. 158, n. 3, p. 280–287, 2003.

TAX, D. M. J. **One-class classification**. 2001. Tese (Tese de doutorado) — Delft University of Technology, 2001.

ZHAO, X. S. P.; FAN, J. Propensity score and proximity matching using random forest. **American Journal of Epidemiology**, Contemp Clin Trials, n. 47, p. 85–92, 2016.