UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS

Matheus Fortunato Jandiroba Barros

Desenvolvimento de sistema híbrido e adaptativo para identificação de fraudes em transações financeiras eletrônicas

São Carlos

2021

Matheus Fortunato Jandiroba Barros

Desenvolvimento de sistema híbrido e adaptativo para identificação de fraudes em transações financeiras eletrônicas

Monografia apresentada ao Curso de Engenharia Elétrica com Ênfase em Eletrônica, da Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Engenheiro Eletricista.

Orientador: Prof. Dr. Francisco Aparecido Rodrigues

São Carlos

2021

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues Fontes da EESC/USP com os dados inseridos pelo(a) autor(a).

Fortunato Jandiroba Barros, Matheus
F425d Desenvolvimento de sistema híbrido e adaptativo para identificação de fraudes em transações financeiras eletrônicas / Matheus Fortunato Jandiroba Barros; orientador Francisco Aparecido Rodrigues. São Carlos, 2021.

Monografia (Graduação em Engenharia Elétrica com ênfase em Eletrônica) -- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2021.

1. machine learning. 2. classificação. 3. data science. 4. fraude. 5. amostragem. 6. desbalanceamento. 7. decision tree. 8. random forest. I. Título.

Eduardo Graziosi Silva - CRB - 8/8907

FOLHA DE APROVAÇÃO

Nome: Matheus Fortunato Jandiroba Barros

Título: "Desenvolvimento de sistema híbrido e adaptativo para identificação de fraudes em transações financeiras eletrônicas"

Trabalho de Conclusão de Curso defendido e aprovado em 16/07/2021,

com NOTA 8,0(oito,zero), pela Comissão Julgadora:

Prof. Associado Francisco Aparecido Rodrigues - Orientador - SME/ICMC/USP

Prof. Associado Rogério Andrade Flauzino - SEL/EESC/USP

Prof. Associado Carlos Dias Maciel - SEL/EESC/USP

Coordenador da CoC-Engenharia Elétrica - EESC/USP:
Prof. Associado Rogério Andrade Flauzino

Este trabalho é dedicado à minha mãe
que sempre me apoiou e fez de tudo por mim.

AGRADECIMENTOS

Eu gostaria de agradecer à todos estiveram presente na minha vida nos últimos anos e que me apoiaram nessa jornada dentro da Universidade São Paulo. Em especial, agradeço:

- Ao meu orientador de TCC, Prof. Dr. Francisco Aparecido Rodrigues, pelos incentivos, paciência, conselhos e todo conhecimento transmitido que me ajudaram a construir este trabalho.
- À minha mãe, pai, madrinha, irmã e todos os membros da minha familia que estiveram ao meu lado e me apoiaram nos estudos durante esses longos anos, desde o ensino médio do colegial.
- À todos amigos que fiz durante a época de graduação e em especial aos da república espirito de porco que me ensinaram muito sobre amizade, respeito e me proporcionaram experiências para vida.
- À minha namorada, Fernanda, pelo companheirismo durante os anos e por ter me incentivado na etapa final da minha graduação.
- Ao meu orientador de Iniciação Cientifica, professor Prof. Dr. Marco Henrique Terra, por ter me permitido desenvolver um trabalho tão gratificante e ter o primeiro contato com a pesquisa acadêmica.
- Aos meus professores e orientadores de monitoria: Prof. Dr Rogério Flauzino em SEL0302 e Prof. Dra. Juliana Cobre em SME0320 que me permitiram ter um pouco da experiência de ensinar em sala de aula.
- À técnica administrativa Jussara Ramos e auxiliar administrativa Aura Aparecido do departamento da elétrica, por toda paciência, conversas e conselhos que tanto me ajudaram.
- À todos meus professores e professoras responsáveis por compartilhar seus conhecimentos e contribuir para minha formação.

"In God we trust; all others must bring data"
William Edwards

RESUMO

MATHEUS, B. **Desenvolvimento de sistema híbrido e adaptativo para identificação de fraudes em transações financeiras eletrônicas**. 2021. 58p. Monografia (Trabalho de Conclusão de Curso) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2021.

No trabalho em questão foi estudado um conjunto de dados relativos à fraudes financeiras realizadas através de transações eletrônicas. Durante o estudo, o objetivo principal foi encontrar maneiras de identificar possíveis fraudes baseando-se em um conjunto de variáveis disponiveis a respeito de cada transação. Foram discutidos e aplicados diversos métodos de machine learning como *decision tree*, *random forest* e *AdaBoost* para realização da classificação correta da transação em duas classes: fraudulenta ou não fraudulenta. Por se tratar de um conjunto de dados altamente desbalanceado foram discutidas técnicas de amostragem para evitar *overfit*. As técnicas aplicadas foram avaliadas por um conjunto de métricas, incluindo uma métrica financeira baseada nos custos envolvidos em cada transação. Entre as modelagens testadas, foi obtido o melhor retorno financeiro e uma AUC de 0.82 para a combinação de um modelo de *Random Forest* com amostragem do tipo SMOTE.

Palavras-chave: Machine learning. Classificação. Data science. Fraude. Amostragem. Desbalanceamento. Decision tree. Random forest. Adaboost. Undersample. SMOTE.

ABSTRACT

MATHEUS, B. **Development of a hybrid and adaptative system for identification of fraud in financial electronic transactions**. 2021. 58p. Monografia (Trabalho de Conclusão de Curso) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2021.

The object of study on this paper is a dataset of financial frauds relative to electronic transactions. In this paper, the main goal was to find ways of identify potential frauds based on a set of variables available about each transaction. It was discussed and applied several methods of machine learning as decision tree, random forest and adaboost for the correct classification of the transactions in two classes: fraud and not fraud. Since it was a huge imbalanced dataset, it was discussed sampling techniques to avoid overfit. The techniques applied were evaluated based on a set of metrics, including a financial metric that involves financial costs in each transaction. Between all the models that were tested, the best financial return and AUC of 0.82 was obtained with a combination of a random forest model and SMOTE sampling technique.

Keywords: Machine learning. Classification. Data science. Fraud. Sampling. Imbalanced. Decision tree. Random forest. Adaboost. Undersample. SMOTE.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

RF	Random forest
DT	Decision tree
SMOTE	Synthetic Minority Oversampling Technique
ROC	Receiver operating characteristic
AUC	Area under the ROC curve
TP	True positive
FP	False positive
TN	True negative
FN	False negative
KNN	K nearest neighbor

SUMÁRIO

1 INTRODUÇÃO

1.1 Mercado de transações eletrônicas

Cartões de crédito e débito são uma maneira que os bancos têm de permitir com que os clientes acessem os seus recursos de forma eletrônica, seja online ou no mundo real através de uma máquina de leitura, e que os vendedores possam vender e receber recursos através de pagamentos realizados de tal maneira.

No Brasil, o número de transações eletrônicas realizadas vem crescendo substancialmente, como pode ser visto na Figura 1, e está diretamente relacionada à alguns fatores vantajosos para as duas partes da transação. O cliente, no lugar do comprador, possui todo o seu recurso financeiro concentrado em um lugar só e totalmente acessível. O vendedor, no lugar de quem receberá a transação, pode realizar controle das vendas de forma mais fácil e de forma segura, dado que não precisará expor seus recursos recebidos fisicamente. Vale a pena ressaltar que para o ano de 2021 os dados estão incompletos e representam apenas uma fração do valor anualizado esperado.

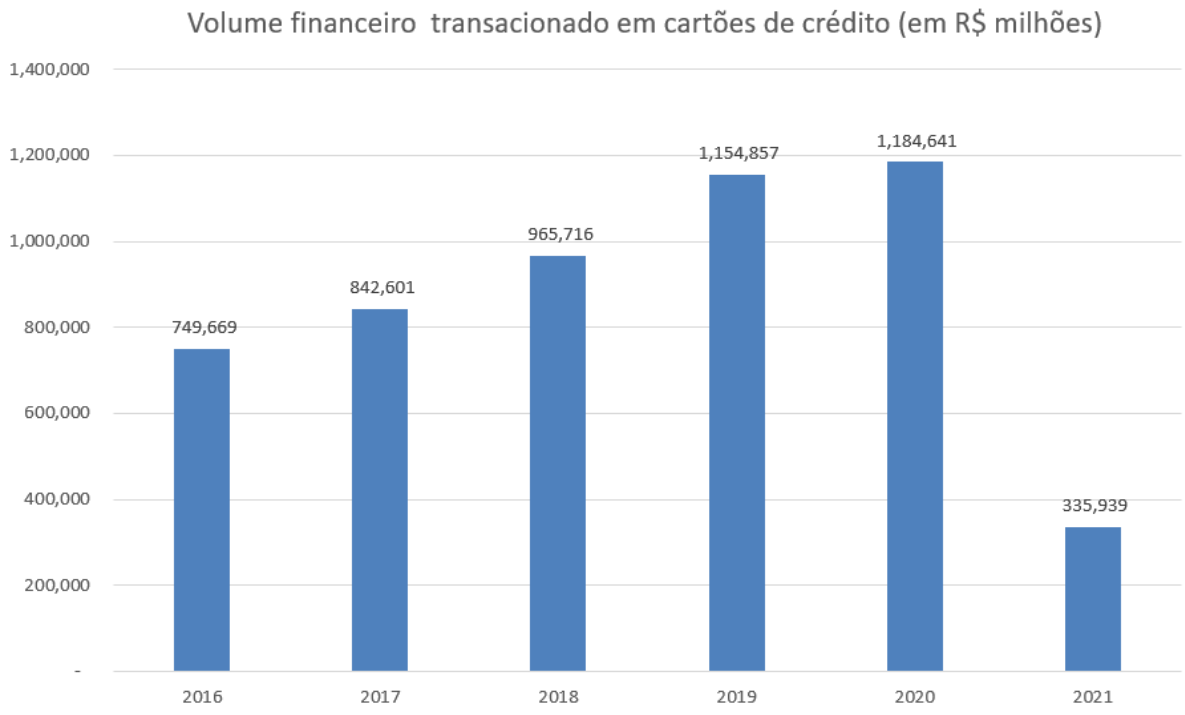

Figura 1: Volume financeiro transacionado no Brasil com cartões de crédito ano após ano. Consultado em maio de 2021. Retirado de: *https://www.abecs.org.br/graficos*

O crescimento em questão toma ainda mais força quando analisado o avanço do *e-commerce*. Um sistema de compras e vendas digital em que toda a transação de pagamento

e recebimento ocorre de forma online e deve ser realizada eletrônicamente.

Riscos em transações financeiras sempre existiram. Quando o uso de cartões não era popular, a maior parte das transações ocorriam através de cédulas de dinheiro e cheques de pagamento. Nos dois cenários, o detentor e o recebedor estavam expostos à roubos e fraudes. Com os cartões não são diferentes.

A organização de uma transação financeira eletrônica acontece em 5 etapas, do cliente que utiliza o cartão até o estabelecimento que pratica a venda. Entre as duas pontas existem o banco que está ligado ao cliente, a bandeira do cartão e a adquirente. A adquirente é responsável por manter relações comerciais com os estabelecimentos e realizar toda a gerência das transações desde a captação até a liquidação da transação. No Brasil pode se citar como exemplos de adquirentes a Cielo e Stone.

1.1.1 Fraudes

As fraudes no universo das transações eletrônicas ocorrem quando um criminoso utiliza o cartão de um terceiro de forma não autorizada em beneficio próprio. É constatado que um crime de fraude ocorreu quando o titular do cartão informa a sua instituição financeira que existe uma transação não reconhecida na sua fatura.

Entre os tipos de fraudes mais comuns no mercado estão: falsificação do cartão, cartão roubado ou perdido, roubo de identidade por vazamento de dados, invasão de conta no estabelecimento. Em todos os casos, como citado anteriormente, um criminoso se apropria do cartão de um terceiro para realizar compras em estabelecimentos.

O procedimento padrão após a constatação da fraude pelo cliente à instituição financeira é a abertura de um processo de *chargeback* ou estorno. Nesse momento existem dois possíveis cenários. No primeiro cenário, a transação ainda não foi finalizada, isso é, o estabelecimento que aceitou a compra do fraudador ainda não finalizou o processo de entrega de produto ou serviço. Dessa maneira, o estorno para o cliente fraudado ocorre sem prejuízo para nenhuma das partes. No cenário seguinte, a transação foi finalizada e o criminoso já está em posse do produto ou serviço. Nessa transação houve um custo para o estabelecimento que aceitou a venda e, a partir desse momento, será iniciado um processo de *chargeback* em que ficará definido quem irá arcar com o custo do crime. Dessa forma, evitar uma fraude significa evitar um possivel prejuízo.

1.2 Problemas de classificação

Existem três tipos distintos de técnicas de aprendizado de máquina: supervisionado, não-supervisionado e aprendizado por reforço. No primeiro tipo, dado N variáveis independente de entrada tenta-se encontrar uma equação que modele a saída para minimizar

uma função de custo. O aprendizado supervisionado podem ser do tipo de regressão ou classificação.

Classificação é o processo de predizer a qual classe um conjunto de variáveis pertence. Dado um vetor x que contém informações sobre uma amostra existe um vetor y de valor discreto que é um atributo especial da amostra e que a categoriza dentro de um conjunto de opções possíveis. O problema mais simples para exemplificar são as classificações binárias, em que o conjunto de dados só pode pertencer a duas classes possíveis, normalmente 0 ou 1. A modelagem de classificação é a atividade de definir uma função f que dado um vetor de variáveis x consiga retornar a classe y em que a amostra pertence (TAN; STEINBACH; KUMAR, 2005).

As técnicas de classificação utilizadas em *machine learning* buscam encontrar um modelo que consiga retornar como resultado a classe que os dados de entrada pertencem. Além disso, a equação também deve ser capaz de predizer corretamente as classes em que dados de entrada nunca vistos pertencem.

Um método conhecido para resolver problemas de classificação em machine learning pode ser observado na Figura 2. Inicialmente, um conjunto de dados de treino que consistem de registros cujas classes pertencentes já são conhecidas previamente deve ser fornecido. Esse conjunto de treino será utilizado para construção do modelo de classificação. Esses coeficientes são escolhidos de forma que uma função de custo seja minimizada. Em seguida, após o treinamento do modelo, ele é aplicado sobre dados não vistos anteriormente para definição da classe y' que será comparada com a classe real y da amostra.

1.2.1 Árvore de decisão

Um classificador baseado em árvore de decisão é aquele que busca modelar a equação que relaciona os dados de entrada e a saída categórica através de uma função com estrutura de árvore.(MITCHELL, 1997) Uma árvore de decisão também pode ser representada como um conjunto de regras “se A, então B” como representado na Figura 2.

Na árvore, cada nó corresponde a uma posição do vetor da amostra de entrada. Sendo que o ramo que se segue após o nó deriva de cada um dos possíveis valores atribuídos a posição do vetor de entrada. Uma amostra é classificada ao começar no topo da árvore e ir “descendo” testando em cada nó os valores que seu vetor de entrada possui em cada posição.

A árvore de decisão junto com a escolha dos atributos que serão testados em cada nó é feita de uma maneira *top-down*. A árvore é construída de cima para baixo, deixando em cima os atributos que trazem o maior ganho de informação para o conjunto de dados apresentado. Isso é, a árvore poderia ser construída como indicado na Figura 3 mas a ordem em que os parâmetros são apresentados em cada nó da árvore da Figura 2 divide de maneira mais eficiente o conjunto e suas informações.

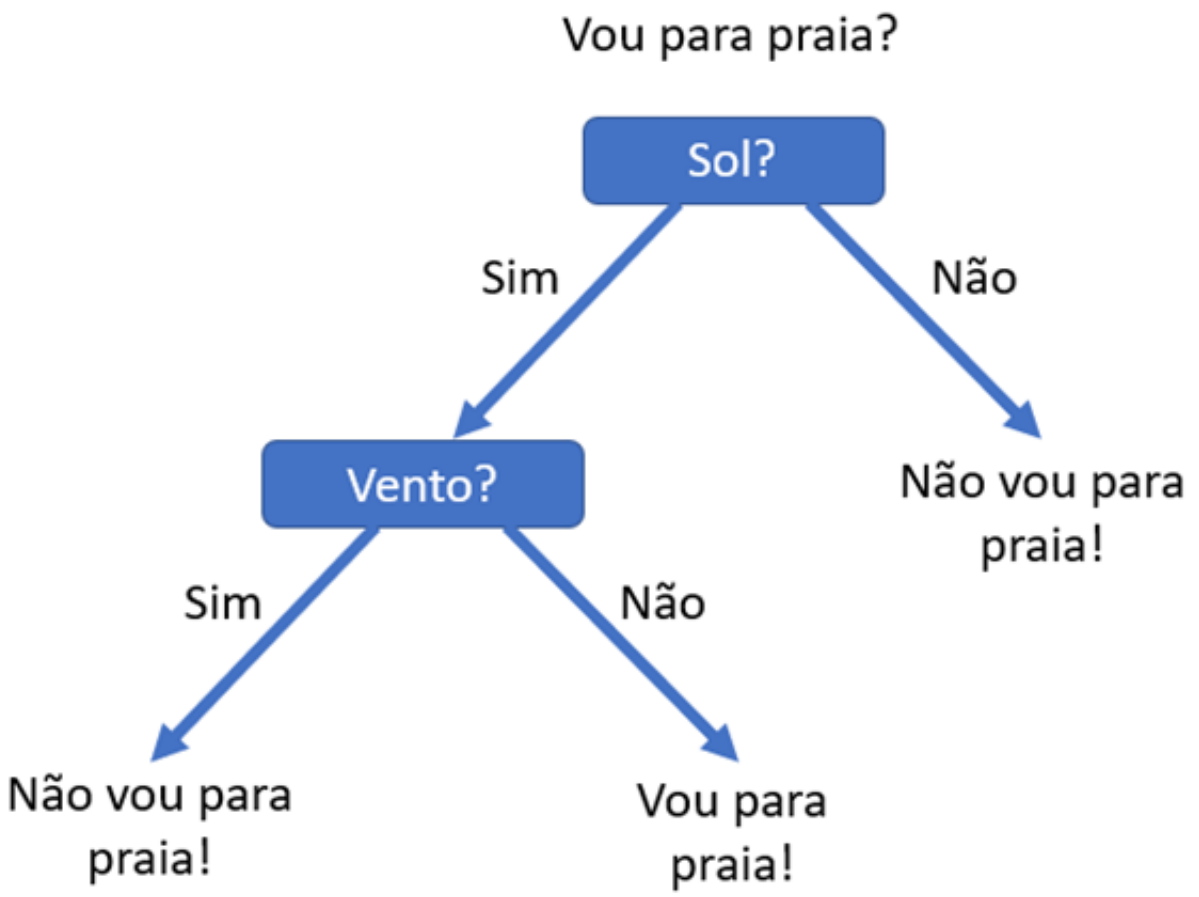

Figura 2: Exemplo de funcionamento do algoritimo de árvore de decisão aplicada para um conjunto de dados que define se a pessoa vai ou não a praia. Obtido em: *https://didatica.tech/wp-content/uploads/2020/07/image-3.png*

A determinação de quais atributos serão utilizados nos nós é feita pelo algoritmo ID3 ou por C4.5 que são famosos por derivarem e serem utilizados por diversos métodos de classificação baseados em árvore de decisão.

Basicamente o ID3 seleciona os melhores atributos para serem utilizados em cada nó. É um processo iterativo e utiliza-se como métrica de escolha o ganho de informação. O ganho de informação calcula a redução de entropia: quão bem um atributo consegue dividir as classes. (NOWOZIN, 2012)

A entropia pode ser calculada como(NOWOZIN, 2012):

$$Entropia(S) = -\sum_{i=1}^{N}[p_i * log_2(p_i)] \tag{1.1}$$

Onde:

n = número total de classes

P_i = probabilidade da classe i

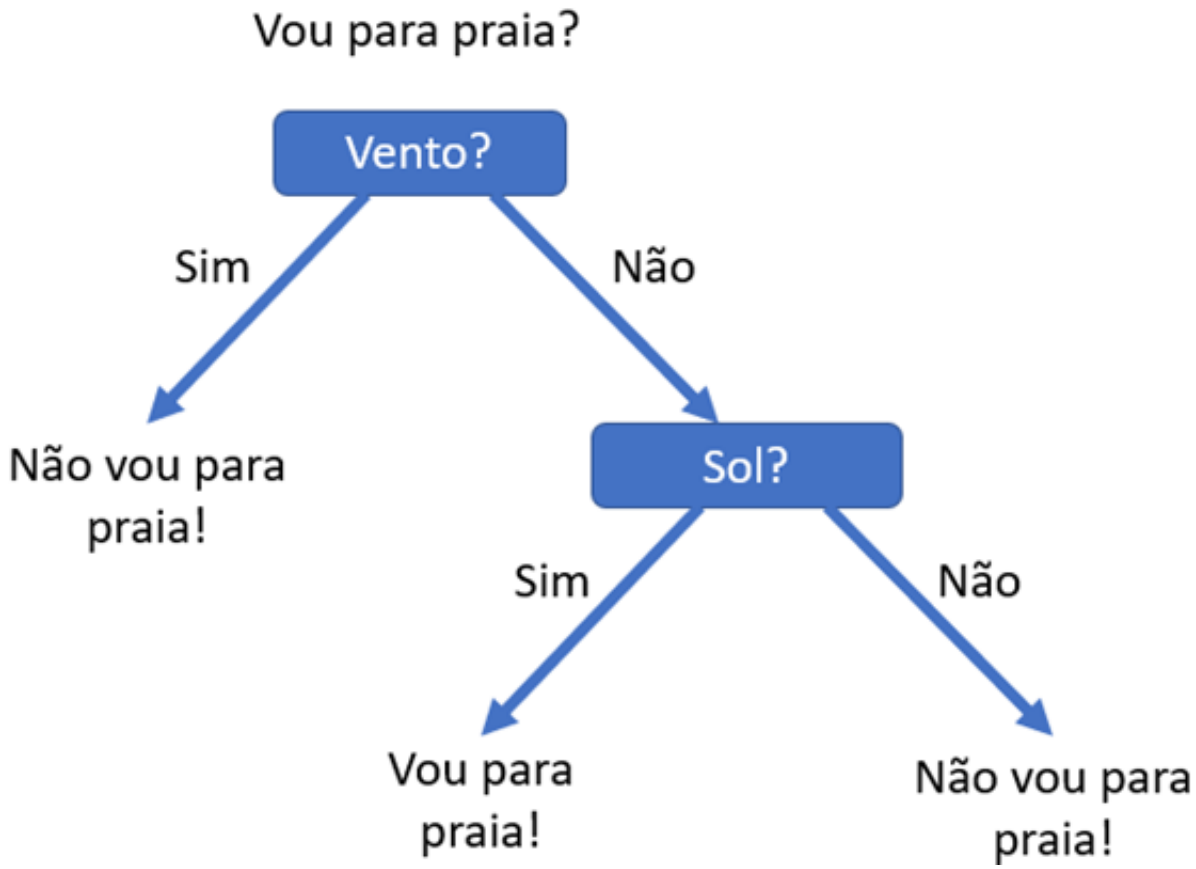

Figura 3: Exemplo de funcionamento do algoritimo de árvore de decisão aplicada para um conjunto de dados que define se a pessoa vai ou não a praia trocando a ordem das *features*. Obtido em: *https://didatica.tech/wp-content/uploads/2020/07/image-5.png*

Já o ganho de informação pode ser calculado através de:

$$IG(S, A) = Entropia(S) - \sum_{v=1}^{K}[(|S_v|/|S|) * Entropia(S_v)] \tag{1.2}$$

Onde:

S_V = conjunto de amostras que da coluna A cujo valor é v.
S = conjunto total de amostras
$|S|$ = número de amostras em S

1.2.2 *Random forest*

Uma modelagem do tipo *ensemble* se baseia em utilizar resultados de diferentes modelos com performance baixa com objetivo de gerar uma classificação melhor. Um exemplo mais simples é a *random forest* que é composta por um conjunto de árvores que dão resultados diferentes e por fim o resultado mais votado por cada uma das árvores é o resultado final.

Existem dois métodos para os modelos ensemble:

- Algoritimos de *boosting*: a saída de cada modelo serve como entrada para o próximo. Onde modelos sucessivos dão pesos extras para amostras que foram preditas incorretamente e que, no fim, uma votação ponderada ocorre.(FREUND; SCHAPIRE, 1996)

- Algoritimos de *bagging*: combina múltiplos modelos de forma independente. O treinamento de cada um dos modelos não é feito com o mesmo conjunto de dados, pelo contrário, ocorre uma amostragem com reposição dos dados para a escolha do conjunto de treinamento, dessa maneira garante-se que cada modelo é diferente. No fim, ocorre uma votação em que o mais votado entre os modelos ganha.

O modelo de *random forests* adota a técnica de *bagging* e utiliza árvores de decisão como o "modelo fraco" na sua construção. Além disso, é adicionada mais uma camada de aleatoriedade na construção do modelo. A *random forest* por natureza já utiliza de um *subset* de dados que foi amostrado com reposição do conjunto de dados original. Além disso, ocorre também uma amostragem das *features*. Na construção de cada árvore do *random forest* para cada um dos nós não será possível escolher o atributo mais significativo do conjunto total de dados e sim de um subconjunto escolhido aleatoriamente. (LIAW; WIENER, 2001)

1.2.3 *AdaBoost*

O algoritmo *AdaBoost* foi desenvolvido no fim da década de 90 e utiliza na sua construção a técnica de boosting. O método combina modelos fracos para formar um classificador final forte. De forma geral, no algoritmo existem duas tarefas principais: treinar de forma iterativa o classificador com um subconjunto de amostras que não foram bem classificadas anteriormente e a segunda tarefa é atribuir diferentes pesos para o voto de cada classificador baseado na performance passada. (FREUND; SCHAPIRE, 1997)

O funcionamento do algoritmo é baseado na chamada iterativa de um modelo fraco de forma repetitiva. Na iteração número t, o algoritmo fornece para o modelo fraco um subconjunto de dados com distribuição Dt sobre o conjunto total S. O modelo fraco gera uma hipótese ht que deve classificar corretamente uma fração das amostras que aparecem com grande probabilidade na distribuição Dt. Esse processo ocorre iterativamente até que uma hipótese final é construída através das hipoteses ht geradas pelos modelos fracos em cima de diferentes frações e distribuições do conjunto total S.(FREUND; SCHAPIRE, 1996)

1.3 Medidas de eficiência da predição

Em atividades desenvolvidas com *machine learning* não basta apenas escolher um algoritmo para modelar os dados já conhecidos e aplicar o modelo para os dados não visto. É necessário fazer um estudo extensivo a respeito de qual modelo melhor se aplica para o problema em questão e fazer uma escolha baseada em dados após medir a performance de cada um. Existem diversas maneiras quantitativas de medir a performance do modelo e o quão boa é a aderência do método com os dados disponíveis.

No problema de identificação de transações fraudulentas de cartão de crédito a classificação de cada transação é alguma entre as duas opções: "é uma transação verdadeira" ou "é uma transação fraudulenta". Portanto, a performance do modelo será medida através de métricas que avaliem o quão corretamente uma transação pode ser classificada. Como se trata de um problema onde a maior parte das transações é verdadeira, não basta simplesmente pedir a acurácia do modelo, isto é, $Acertos/Total$. Um exemplo que ilustra bem a situação é dado por (CHAWLA, 2005), onde ele afirma que um dataset contendo imagens de mamografia contem 98% pixels normais e 2% anormais. Uma estratégia de classificar aleatoriamente a classe majoritária por si só já forneceria uma acurácia de 98% enquanto o modelo não executa nada sofisticado. A natureza do problema em questão necessita que a taxa de acerto na classe minoritária seja alta. Dessa forma, a literatura traz algumas alternativas mais robustas que serão apresentadas a seguir.

1.3.1 Matriz de confusão

Existe uma tabela chamada que relaciona diferentes métricas e é muito utilizada para construção de métricas para problemas de classificação e que pode ser vista abaixo e

	Realmente fraude	Realmente não é fraude
Classificado como fraude	TP	FP
Classificado como não fraude	FN	VN

Tabela 1: Matriz de confusão.

Precisão : proporção de eventos classificados corretamente e todos os eventos positivos

$$\frac{TP}{(TP + FP)} \tag{1.3}$$

Recall : razão entre os eventos positivos classificados corretamente e todas transações realmente positivas

$$\frac{TP}{TP + FN} \tag{1.4}$$

Medida F : precisão e recall apresentam comportamentos opostos, uma boa precisão só é

alcançada em detrimento de um recall pior. Portanto essa métrica se trata de uma média harmônica entre recall e precisão

$$\frac{2 * precisão * recall}{precisão + recall} \tag{1.5}$$

Acurácia : razão entre o número de transações corretamente classificadas e número total de transações

$$\frac{TP + FN}{TP + FP + FN + VN} \tag{1.6}$$

Medida F beta : De forma simples, combina, assim como a medida F, precisão e recall na mesma métrica. No entanto é possível estabelecer um peso maior para o recall em detrimento da precisão.

$$\frac{(1 + \beta^2) * precisão * recall}{\beta^2 * precisão + recall} \tag{1.7}$$

É muito útil quando se deseja estabelecer uma relação para atribuir mais ou menos importância para recall ou precisão.

1.3.2 *Receiver or Operating characteristic curve*

A medida ROC significa *receiver or operating characteristic curve*. A curva ROC relaciona a medida de acerto para ocorrência do evento e a taxa de alarmes falsos. A curva pode ser construída variando-se o *threshold* para o qual uma amostra é classificada como positiva, dessa maneira é possível construir uma relação entre a taxa de sensitividade e especificidade . A curva pode ser deslocada através de uma manipulação do balanceamento de cada classe durante o treino(CHAWLA et al., 2002).

1.3.3 Área sob a curva

A partir do gráfico ROC construído é possível medir a performance do modelo a partir da área sob a curva (AUC). Onde o valor de AUC pode variar de 0.5, quando o modelo acerta e erra os verdadeiros na mesma proporção, até o valor de infinitesimalmente próximo de 1, quando o modelo consegue uma proporção perfeita de verdadeiros positivos para pouquíssimos alarmes falsos. Portanto, a AUC pode ser utilizada para comparar a performance de diferentes classificadores. Sendo que o modelo performa melhor quanto mais próximo de 1 o valor da métrica estiver.

1.3.4 Performance baseada em custos

Em alguns momentos, quando utilizando a área da curva ROC para comparar diferentes classificadores, a métrica pode ser mal interpretada. Isso é, o classificador com a maior área pode não ser o melhor dado para um dado threshold. A curva ROC também

pode ser utilizada para definir o melhor threshold para determinação de uma classe. O melhor threshold é escolhido de forma que o classificador forneça o melhor *trade-off* entre os custos associados à uma falta de detecção da classe positiva ou à um alarme falso.(PROVOST; FAWCETT, 2001) Para isso, é definido uma função de custo de um ponto (x, y) no plano do ROC através da equação

$$C = (1 - p) * \alpha * x + p * \beta * (1 - y) \tag{1.8}$$

Onde:

α = custo de um FP

β = custo de um FN

p = proporção de casos de positivos

Então, dado um conjunto de classificadores distintos, o custo de cada resultado é calculado. Dessa forma, dado um único classificador em questão é possível determinar qual ponto (x, y) e *threshold* a ser utilizado fornece o menor custo possível. Além disso, é possível determinar também dado um conjunto de classificadores qual deve ser escolhido para minimizar os custos envolvidos na classificação.

Outra alternativa para determinação desse valor é escolher o *threshold* associado ao ponto (x, y) de menor distancia até o ponto (0,1) da curva ROC, como sugerido por (SONG et al., 2013).

1.3.5 Linhas de Iso-performance

Um ponto (x_1, y_1) no espaço ROC tem o mesmo custo associado que o ponto (x_2, y_2) quando $\frac{\alpha * p}{\beta * (1-p)} = (y_2 - y_1)$. A equação acima representa a inclinação de um conjunto de curvas que possuem o mesmo custo associado. De maneira que, cada conjunto de distribuição de custo e proporção entre TF e FP fornece uma familia de curvas com o mesmo custo associado. Um classificador ótimo para determinadas condições de α, β e p pode ser escolhido quando uma curva de custos intersecta a envoltória convexa. (PROVOST; FAWCETT, 2001)

1.4 Método ROC Convex Hull

No mundo real, os custos associados a um alarme falso na detecção do evento ou até mesmo a proporção de casos positivos são variáveis. De maneira que não é possível assumir que são constantes ao longo do tempo. Portanto, se em determinado momento certo modelo de classificação ou *threshold* foi escolhido de maneira ótima dadas as condições de custo e proporção das classes, no futuro talvez esse mesmo modelo não continue sendo o melhor. De maneira que seria ideal a construção de um sistema de detecção de fraude que não

estivesse comprometido com um único classificador que, em determinado momento do tempo, se mostrou ideal.

O método do *ROC Convex Hull* foi proposto por (PROVOST; FAWCETT, 2001) e demonstra que pode ser utilizado para construir, a partir dos classificadores disponiveis, um sistema híbrido de classificação de um evento que sempre irá performar o melhor possível dado qualquer custo e proporção entre as classes que esteja associado ao conjunto de dados. Dentro do plano ROC através do método é possivel construir uma envoltória convexa que identifique os subconjuntos de classificadores que são potencialmente ótimos como pode ser observado na Figura 1.4.

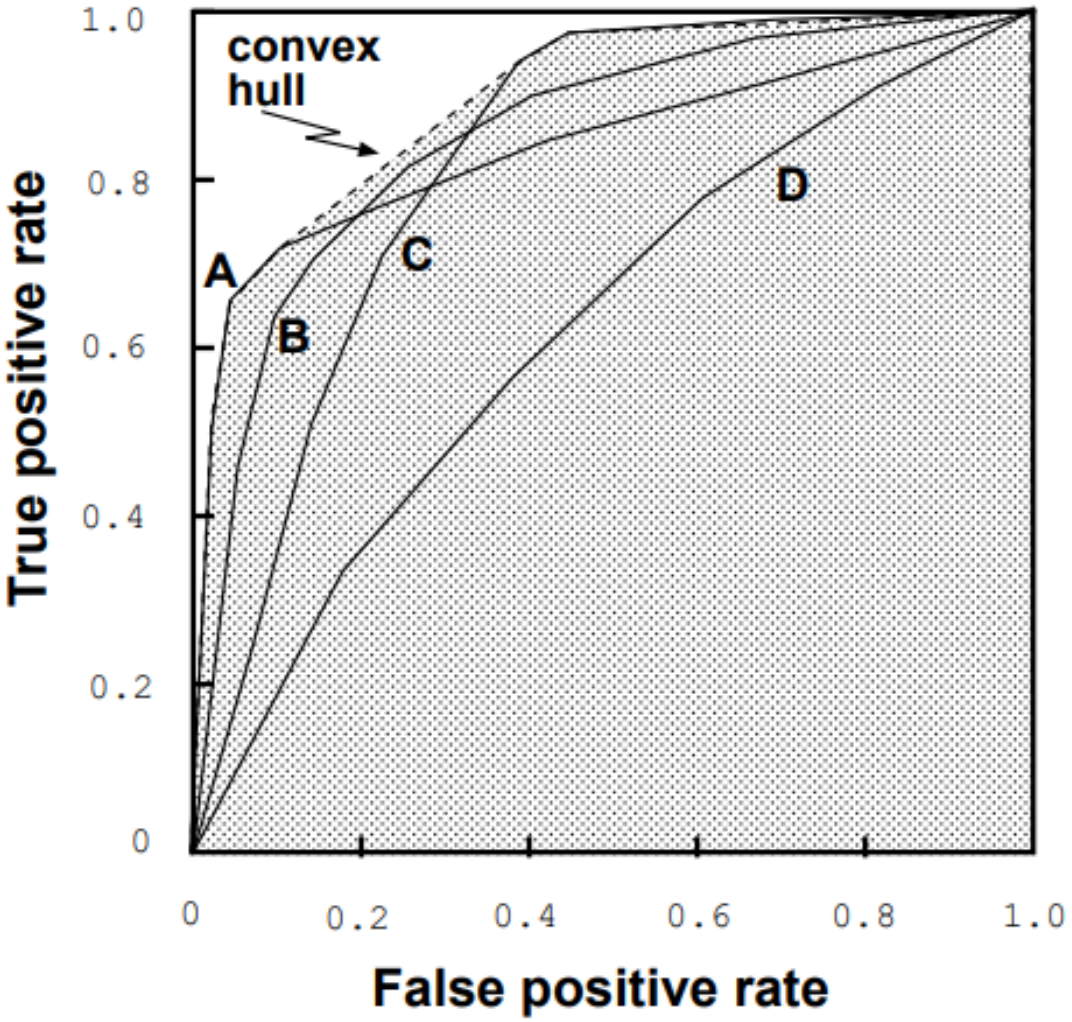

Figura 4: Envoltória convexa para quatro curvas: A, B, C e D.

Para ilustrar, serão considerados dois cenários I e II. Em ambos cenários, a proporção de eventos classificados como falso supera os verdadeiros na proporção de 5:1. Para o primeiro cenário, os custos de falso negativo e falso positivo são os mesmos. Já no segundo

cenário o custo de um falso negativo é 25 vezes maior que o custo de um falso positivo. Para cada um dos dois cenários uma familia de curvas de performance igual é definida. Portanto, para o cenário I a inclinação da curva de custos é 5 e para o cenário II a inclinação é de 0,2. A figura abaixo mostra a envoltória convexa e também as curvas alfa e beta que são as duas curvas com o melhor custo e que intersecta a envoltória no ponto de maneira que esteja o mais próximo possivel do ponto (0,1), como indica (BARBER; DOBKIN; HUHDANPAA, 1996)

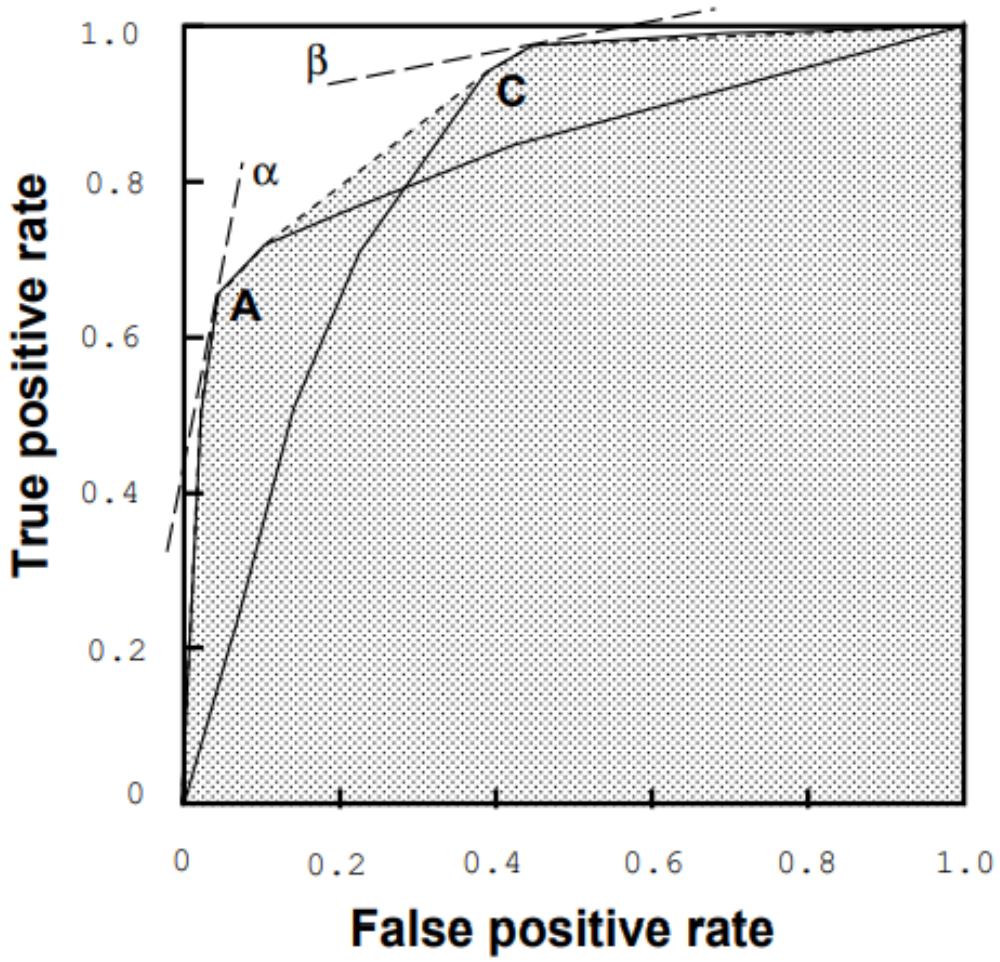

Figura 5: Envoltória convexa para quatro curvas: A, B, C e D com as respectivas α e β que intersectam a envoltória no ponto de melhor custo.

1.4.1 Construção da envoltória convexa

O metodo da envoltória convexa do plano ROC envolve a própria envoltória construida com os resultados dos classificadores e as curvas de custos associadas a cada distribuição de classe.

Portanto, podem-se se seguir os seguintes passos no método (PROVOST; FAWCETT, 2001):

1. Para cada classificador envolvido deve ser plotado no plano ROC os valores de TP e FP encontrados de maneira que o threshold da decisão seja variado

2. Construção da envoltória convexa através do algoritimo de QuickHull e dos resultados dos classificadores previamente adicionados no plano ROC

3. Para cada conjunto de custos e distribuição de classes deve ser encontrada a inclinação da familia de curvas de iso-performance

4. O classificador ótimo será aquele correspondente ao ponto em que a envoltória convexa se intersecta com a curva de iso-performance mais próxima do ponto (0,1) para cada conjunto de custo e distribuição de classes

1.4.2 Variação e imprecisão das distribuições no tempo

Como já foi mencionado, os custos envolvidos e a distribuição de classes normalmente se alteram ao longo do tempo. Dessa forma, um classificador ótimo deve ser definido novamente. Isso deve ser feito calculando-se a inclinação da familia de curvas de iso-performance e buscando a interseção dessas mesmas curvas com o envoltório convexo. Até então, a busca pelo classificador ótimo parte da hipotese que, mesmo que haja variação, as informações de custo e proporção entre as classes são precisas. No entanto, mesmo que a informação da distribuição seja dado em intervalos e não um valor único é possivel encontrar o classificador ótimo.

Quando as informações sobre os custos e distribuições das classes são imprecisas não é possivel determinar uma única inclinação para as familias de curvas de iso-performance. Para esse cenário, é calculado um intervalo em que a inclinação da curva pode estar de modo que é gerada uma família de curvas de iso-performance com inclinações diferentes. Dessa maneira, ao contrário do cenário em que a inclinação é determinística, não é mais possível encontrar um único ponto de interseção entre a curva de iso-performance de melhor custo e a envoltória convexa. A partir disso é encontrado um trecho ótimo da envoltória que é capaz de determinar qual o melhor classificador para a situação, como pode ser visto na Figura 1.4.2.

1.5 Métodos de balanceamento de classes

Conjuntos de dados com classes balanceadas não são vistos naturalmente. (SHEN; XU; XUE, 2020) De tal modo que o tratamento das classes e o rebalanceamento das mesmas é um problema comum e amplamente discutido no campo de *machine learning* (LIU; WU; ZHOU, 2009). O problema acontece quando existe uma classe que possui uma ocorrência rara no conjunto de dados. Para problemas de classificação binário isso significa que existe uma classe que predomina no dataset e o número de registros de uma classe supera fortemente a presença da classe oposta. Dados desbalanceados são comumente

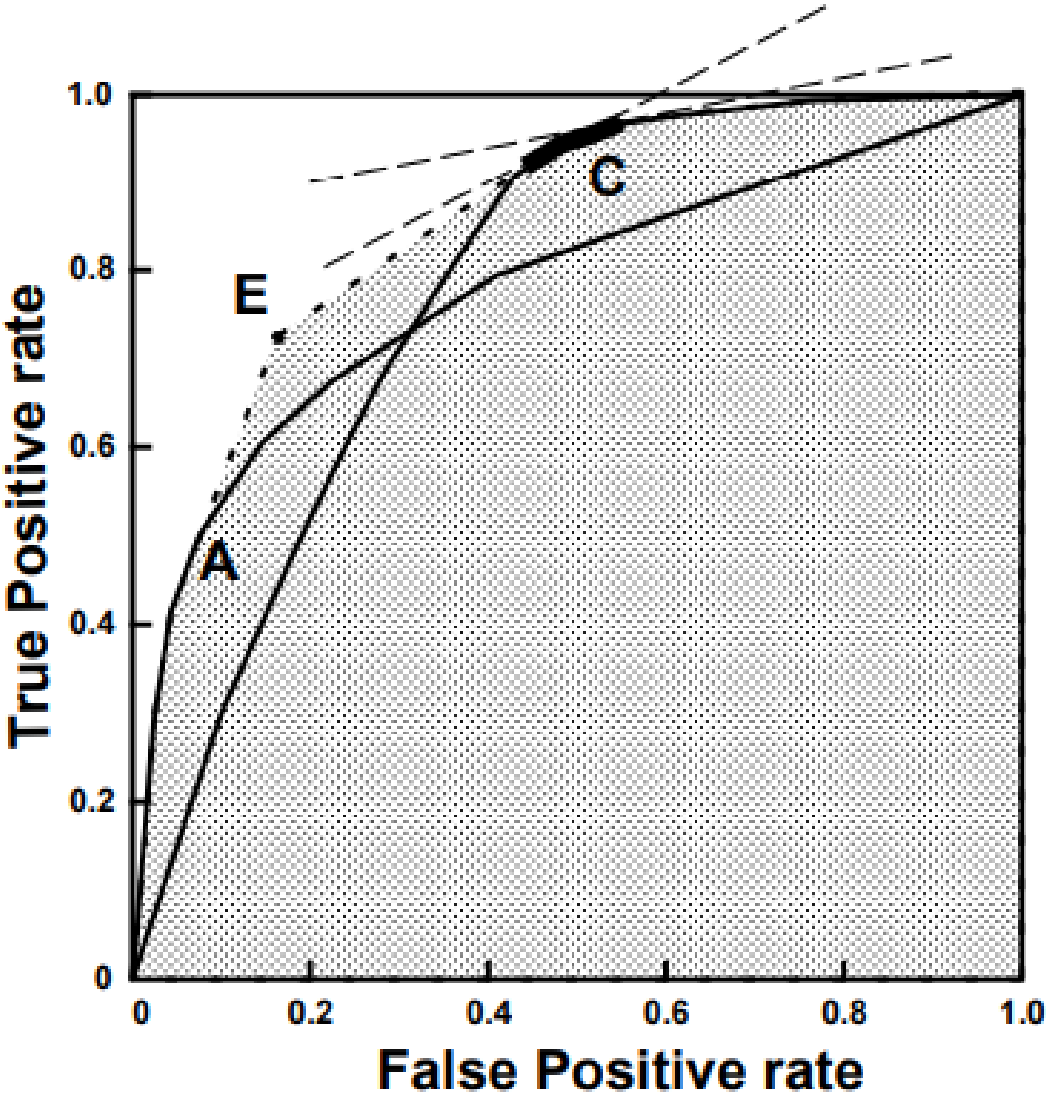

Figura 6: Metodo de construção para envoltória convexa de cinco curvas: A, B, C, D e E.

vistos em classificação de fraude, diagnósticos médicos e classificação de *e-mail* como spam. (LIU; WU; ZHOU, 2009)(WEI et al., 2012)

Grande parte dos algoritmos de *machine learning* não estão preparados para executar a classificação de dados desbalanceados(LUCAS; JURGOVSKY, 2020). Esses algoritmos normalmente visam a melhoria da acurácia geral, de maneira que, os dados quando não recebem o tratamento adequado acabam levando ao *bias* da aplicação(KAMALOV; DENISOV, 2020). Já foi demonstrado que o desbalanceamento de classes de um *dataset* está intrinsecamente ligado a má performance de alguns algoritmos de classificação (CHAWLA, 2005).

As soluções mais conhecidas para o problema em questão se resumem a realizar manualmente a reamostragem *(resample)* dos dados. Sobre-amostrar *(Oversample)* a classe

predominamente ou sub-amostrar *(downsample)* a classe rara são duas das soluções mais simples.

1.5.1 *Undersample*

O método consiste em remover um número de registros aleatórios da classe predominante de modo que a proporção entre as classes se iguale. O método parte da suposição que a informação que está sendo retirada é redundante, visto que ja foi apresentada outras muitas vezes. É comum se deparar com problemas de *underfit* ao se utilizar essa técnica. Dado que com a remoção de dados o dataset final pode conter uma quantidade insuficiente de registros para o algoritmo performar bem. (LUCAS; JURGOVSKY, 2020)(DRUMMOND; HOLTE, 2003)

1.5.2 *Oversample*

Esse metodo de reamostragem consiste em escolher aleatoriamente registros da classificação minoritária e repeti-los no conjunto de dados. O maior problema que o método possui é por não acrescentar novas informações no conjunto de dados, apenas repetir informações já conhecidas. Dessa forma, ao utilizar o método em questão é comum se deparar com problemas de *overfitting* no modelo de aprendizado de máquina. O tempo de execução do algoritimo também é afetado, dado que com o *dataset* maior o tempo de treino e teste aumentam. (LUCAS; JURGOVSKY, 2020)

1.5.3 *SMOTE (Synthetic Minority Oversampling Technique)*

Em resumo, ambos os métodos anteriores são realizados de modo que não haja introdução de novas informações no conjunto de dados, apenas repetição e remoção de informações.(SHEN; XU; XUE, 2020) Logo, para que esses metodos sejam efetivos é necessária a realização de um trabalho manual para definir os melhores hiperparâmetros.

Para atacar o problema da falta de geração de novas informações, em (CHAWLA et al., 2002) é introduzido o método chamado SMOTE. É uma estratégia que visa criar novos registros sintéticos pertencentes à classe minoritária. Esses registros são criados ao realizar uma média dos atributos que os vizinhos dos registros percentences às classes minoritárias possuem. Para atributos categóricos, a categoria mais representativa para determinado atributo entre os vizinhos do registro será a utilizada para criação do novo registro de classe minoritária. (CHAWLA, 2005) (CHAWLA et al., 2002) No entanto, como desvantagem está o fato que o método não considera a que classe pertence os vizinhos de cada registro. Isso pode levar a uma sobreposição entre as classes. (LUCAS; JURGOVSKY, 2020)

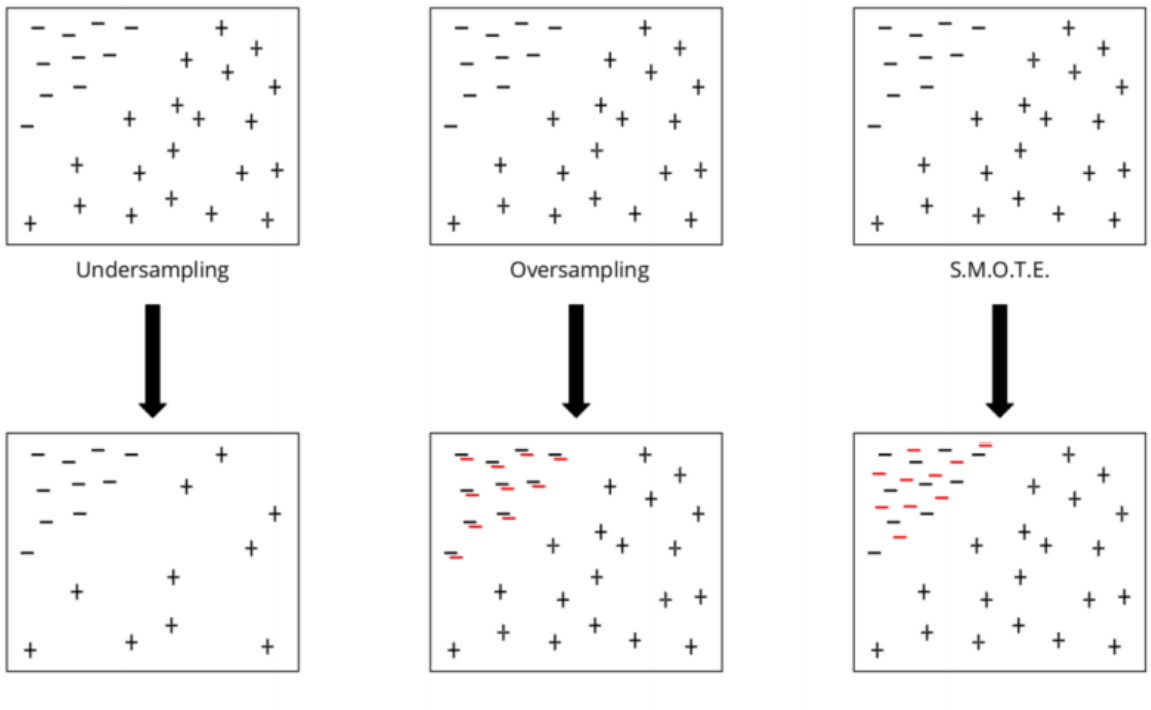

Figura 7: Demonstração e comparação da aplicação dos metodos de amostragem *undersampling*, *oersampling* e SMOTE sobre um conjunto de dados.

2 DESENVOLVIMENTO

2.1 Análise exploratória dos dados

Uma das primeiras etapas no desenvolvimento de um projeto de ciência de dados é análise exploratória dos dados em questão. Se trata de uma investigação que deve ser realizada no conjunto de dados de modo que sejam extraídos informações relevantes para o processo de construção de hipoteses. É crucial, em um projeto de *data science*, entender profundamente os dados que serão utilizados para construção da modelagem posteriormente. Portanto, o objetivo dessa etapa é extrair *insights* que auxiliem no processo de construção de um modelo para atingir o objetivo. Isso será realizado entendendo a qualidade dos dados em questão, como as variáveis estão distribuidas, que tipo de variáveis independentes estão disponíveis e qual a correlação delas com o a variavel *target* final que deseja-se obter.

2.1.1 Organização e tamanho da base de dados

Os dados estão separados em dois datasets identificados como *transaction* e *identity*. O primeiro contém informações relativas as transações de cartão de crédito. Já o segundo diz respeito a identidade de cada um dos clientes. Nem todas as transações disponiveis possuem uma identidade anexada. Muitas das variaveis não possuem nome por conta de questões de privacidade de quem forneceu os dados.

	Linhas	Colunas
Identity	144233	41
Transaction	590540	394

Tabela 2: Número de linhas e colunas para os datasets disponíveis.

2.1.2 Distribuição das variáveis

A variável dependente do problema, e, portanto, o alvo é a coluna *IsFraud* que identifica se uma transação é fraudulenta ou não. A forma mais simples de iniciar o estudo sobre como as variáveis estão distribuídas é justamente pela *target*.

Pelas Figuras 8 e 9 é possível observar a característica do problema ser altamente desbalanceado. Isso é, as transações verdadeiras correspondem a 96.5% do total enquanto as fraudes apenas 3.5%.

Como o dataset possui mais de 400 variáveis e nem todas possuem um rótulo que permite entender do que se tratam, foram escolhidas as *features* mais interessantes para

	Fraude	Transações verdadeiras
Total	20663	569877
Percentual	3.5%	96.5%

Tabela 3: Número absoluto e percentual de fraudes e transações comuns no conjunto de dados.

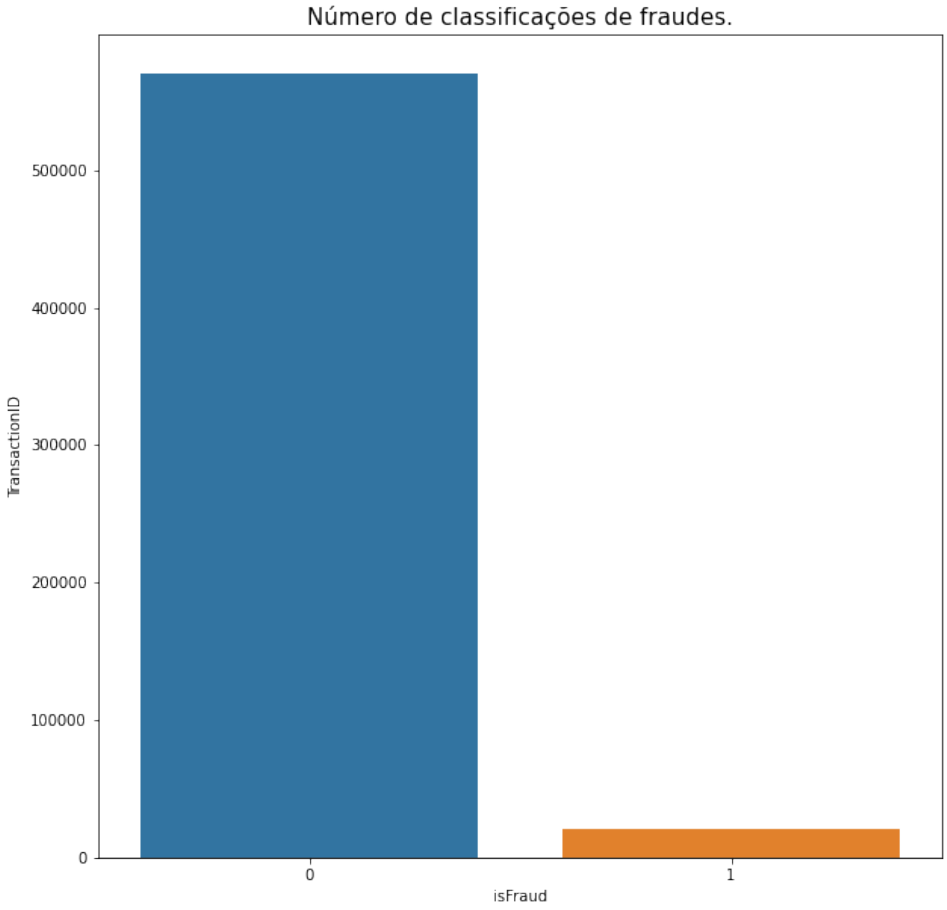

Figura 8: Número absoluto de ocorrência de fraudes e transações veridicas na base de dados.

discussão. Uma delas é a *TransactionAmt* que fornece o valor monetário envolvido na transação financeira e possui a seguinte distribuição:

Fica ainda mais claro ao se deparar com o histograma dos valores que a variável possui uma distribuição com cauda pesada para direita, onde 99% dos valores são inferiores a 1104 como visto na Figura 10.

Como as transações são sempre positivas e sabe-se que distribuições de cauda

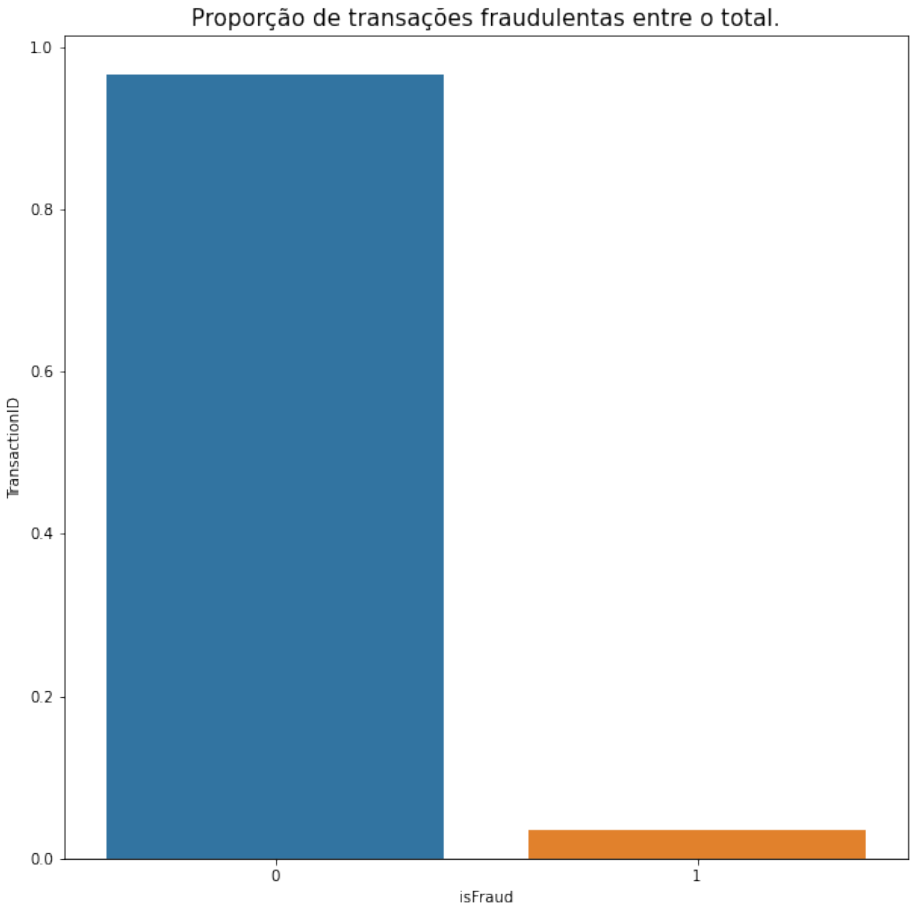

Figura 9: Representatividade percentual de ocorrência de fraudes e transações veridicas na base de dados.

Medida	Valor
Média	135.02
Desvio-padrão	239.16
Minimo	0.25
Máximo	31937.39
1º Quartil	43.32
2º Quartil	68.77
3º Quartil	125.00
Percentil 99	1104.00

Tabela 4: Diversas métricas da variável *TransactionAmt*

pesada estão fortemente ligadas com a curva *Log-normal* foi aplicada a transaformação de *log* na variável e em seguida plotado o histograma que pode ser visto abaixo. A distribuição retornada é muito mais parecida com uma curva normal do que a vista anteriormente.

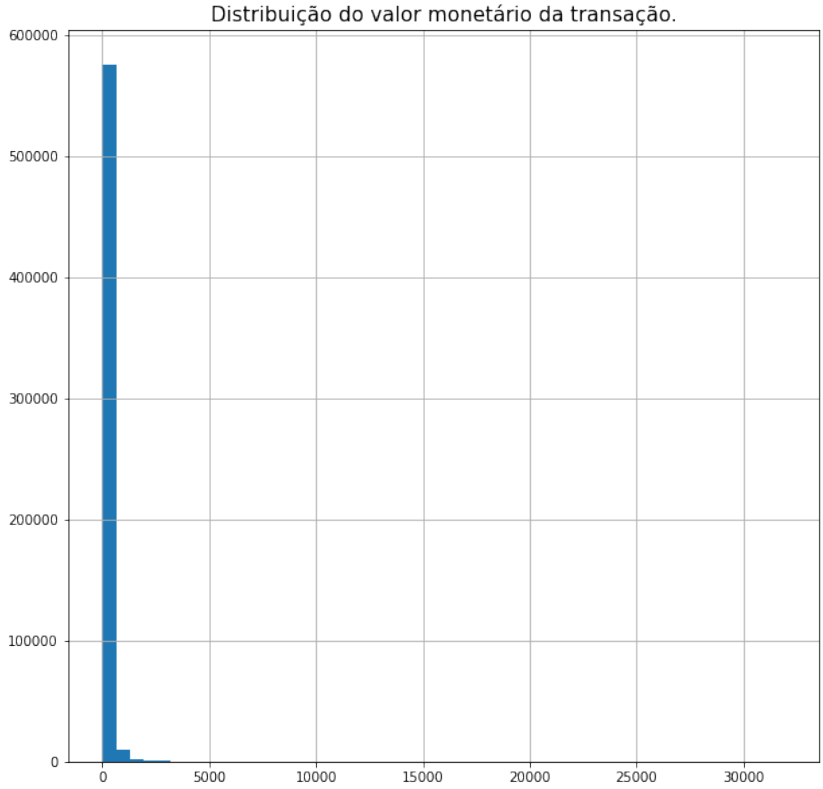

Figura 10: Histograma da distribuição de valor financeiro transacionado.

No mundo das fraudes, os fraudadores estão sempre tentando ganhar o máximo que puderem sem ser descobertos. Por isso surge a hipótese que no universo das transações fraudulentas os valores financeiros transacionados sejam maiores do que no universo de transações verdadeiras.

Medida	Fraudes	Transações verdadeiras
Média	149.24	134.51
Desvio-padrão	232.21	239.39
Minimo	0.29	0.25
Máximo	5191.00	31937.39
1º Quartil	35.04	43.97
2º Quartil	75.00	68.50
3º Quartil	161.00	120.00

Tabela 5: Métricas da variavel *TransactionAmt* separada para casos de transações fraudulentas e transações comuns

Pela tabela 5 observa-se que a média de valores financeiros envolvidos em transações fraudulentas é 11% maior do que a média em uma transação comum. A curva de distribuição

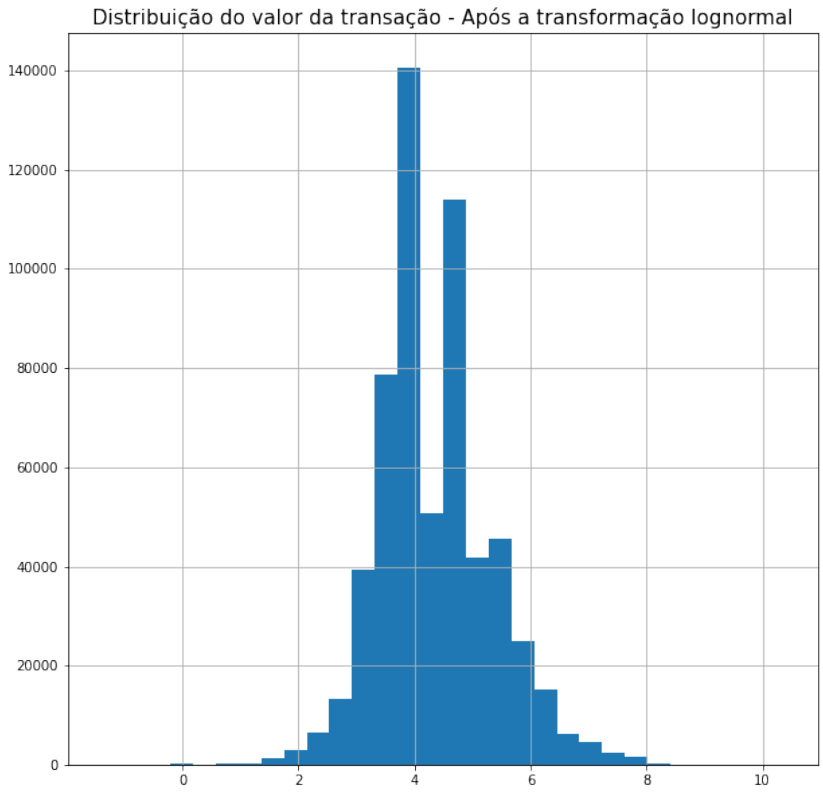

Figura 11: Histograma do *log* do valor financeiro transacionado.

log no universo fraudulento possui valores muito mais esparsos, ou seja, os valores estão distribuidos por uma região maior, se comparado ao universo não-fraudulento.

A próxima *feature* objeto de estudo será a *ProductCD* que indica o tipo de produto envolvido na transação. Como os fraudadores não recebem o dinheiro envolvido na transação mas sim o produto, foi levantada a hipotese que certos tipos de produtos comprados estão mais relacionados a fraudes pela facilidade de revenda ou pelo valor agregado que o produto possui.

Analisando a distribuição do tipo de produto envolvido em cada transação nos dois universos pelos gráficos das Figuras 13 e 14 é possivel através de uma simples análise visual chegar a conclusão que o percentual de produtos do tipo C envolvido em transações fraudulentas é bastante superior do que o percentual de produtos do mesmo tipo no universo de transações comuns. O produto C sai de uma representatividade de 10% do universo de transações comuns para uma representatividade quase quatro vezes superior, de 38%, no universo das fraudes. É uma variável muito importante para construção do modelo e que só seria descoberta fazendo uma análise exploratória profunda dos dados.

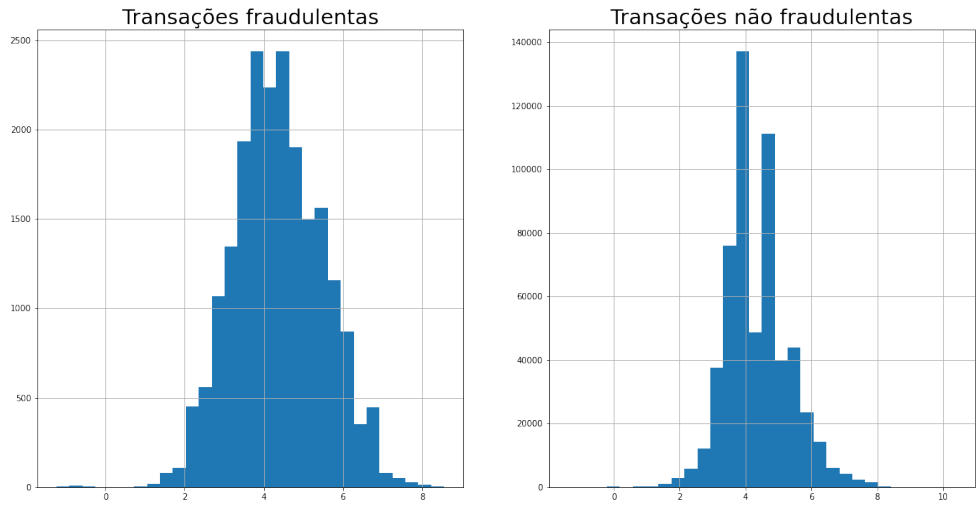

Figura 12: Comparação entre a distribuição do *log* do valor financeiro transacionado para os dois universos.

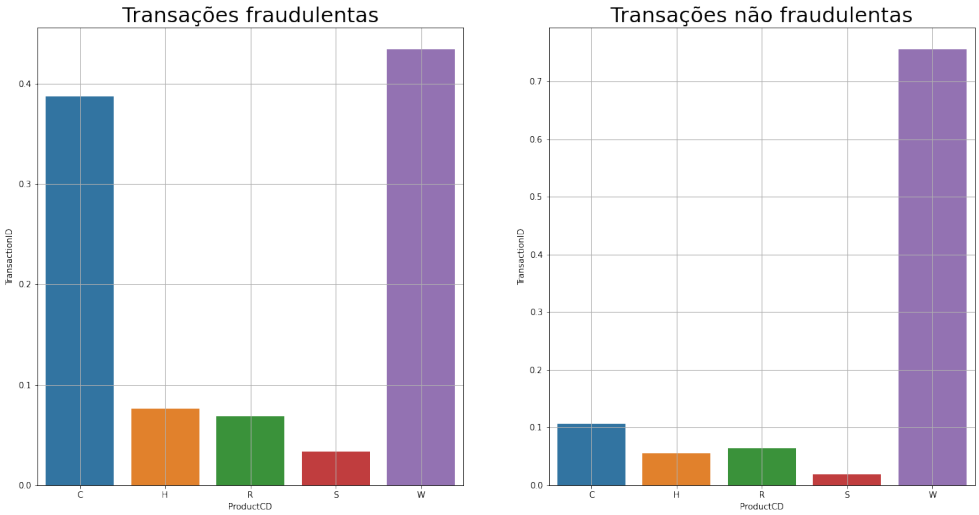

Figura 13: Contagem percentual dos tipos de produto envolvido nas transações para os dois universos com aumento de 300% na ocorrência de produtos do tipo "C"para transações fraudulentas.

As variáveis card4 e card6 representam a bandeira do cartão e o tipo de transação financeira, ou seja, débito ou crédito.

A bandeira do cartão não trouxe nenhuma informação relevante, conforme visto no gráfico da Figura 15. Já na variável que representa o tipo de transação é possivel enxergar

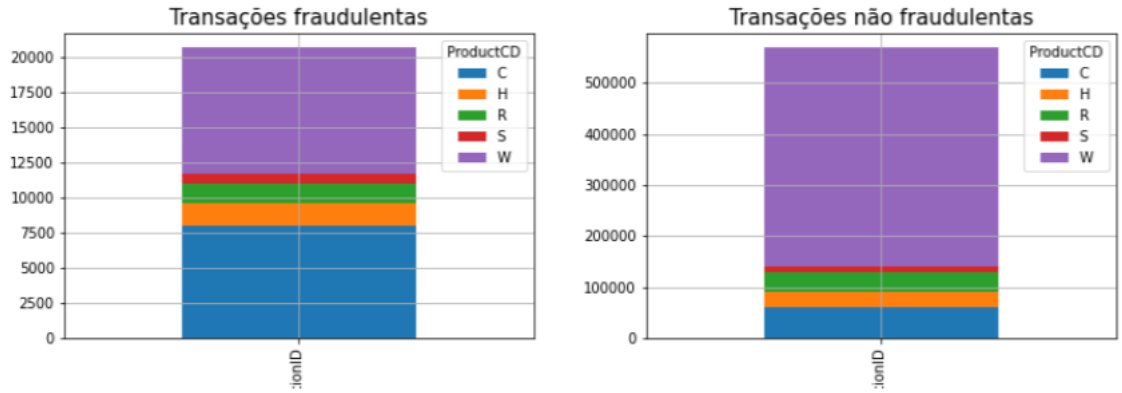

Figura 14: Representatividade de cada um dos tipos de produto nos dois universos analisados.

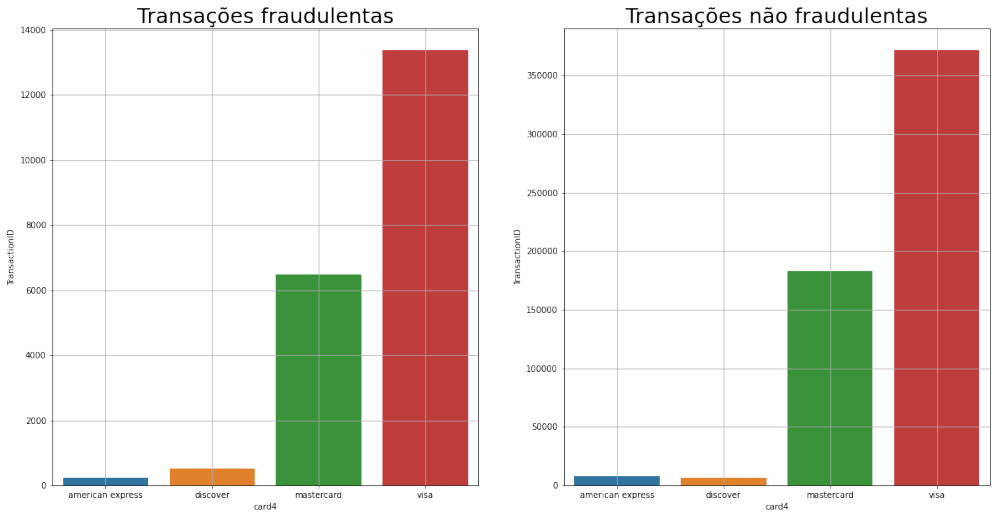

Figura 15: Comparação da distribuição da variável *card4* entre os dois universos. Visivelmente não há diferença entre ocorrência de valores entre os dois universos.

uma alta da representatividade de transações do tipo "crédito"no universo das fraudes. No universo das transações comuns as transações do tipo "crédito"representam 25% enquanto no universo das fraudes esse percentual se aproxima de 50% como pode ser visto na Figura 16.

A feature *DeviceType* representa o tipo de dispositivo que a compra ocorreu. A variável pode assumir dois valores: celular e desktop. Comparando-se a distribuição percentual de cada tipo de variável nos dois universos percebe-se que existe uma leve alta de 10% no percentual de transações em celular no universo das fraudes, visto na Figura 17.

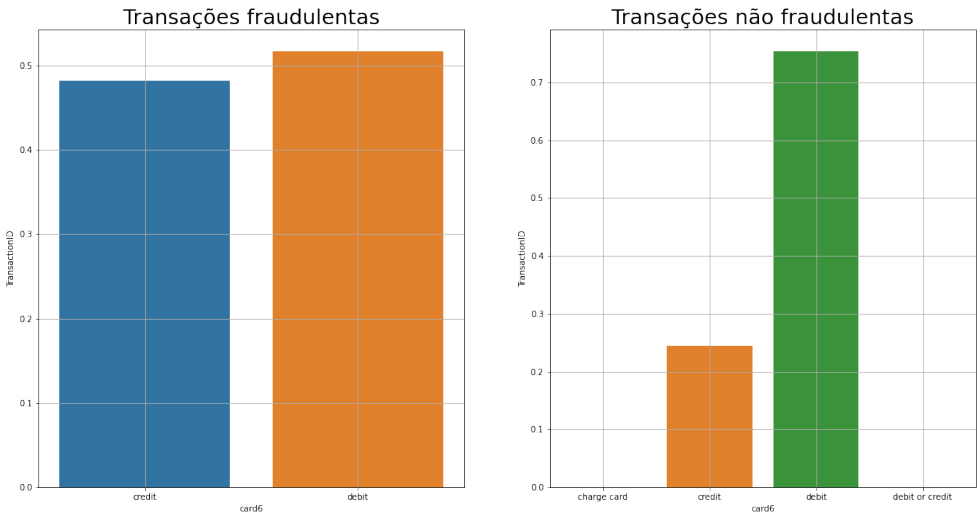

Figura 16: Comparação da distribuição da variável *card6* entre os dois universos. No universo de transações fraudulentas, a ocorrencia do valor "credit"é 100% maior.

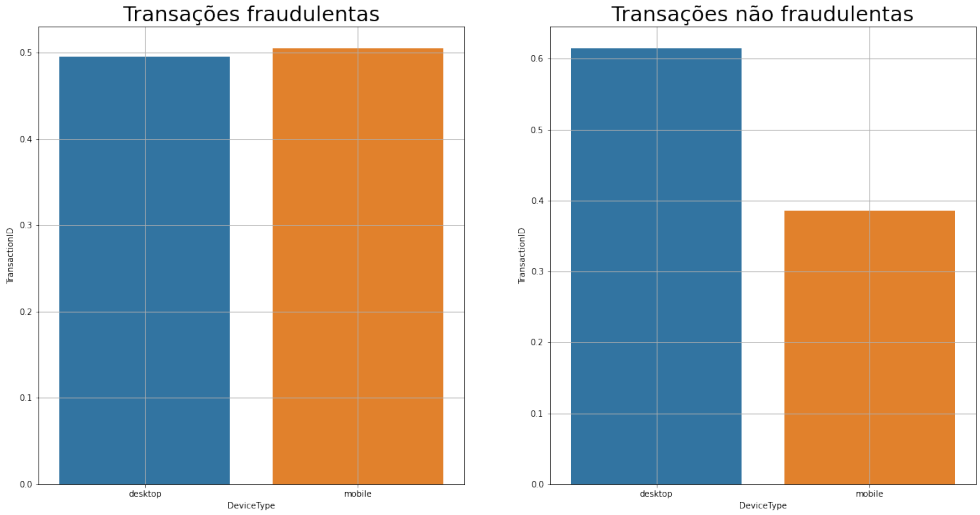

Figura 17: Comparação da distribuição da variável *DeviceType* entre os dois universos. No universo de transações fraudulentas, a ocorrencia do valor *mobile* é 25% maior.

Com os modernos sistemas de confirmação de compras que existem atualmente, os donos de cartão costumam receber mensagens de confirmação em seu telefone celular para que fique ciente da transação. Levanta-se então a hipótese que os fraudadores cometam os crimes em horários menos prováveis que os donos vejam a notificação recebida pela operadora de seu cartão. Portnato, os fraudadores supostamente atuam de maneira mais

intensa tarde da noite e de madrugada.

Investigando os dados fornecidos, não é exatamente o caso, como pode ser visto no gráfico da Figura 18. Certamente há um padrão de mais fraudes conforme o dia passa mas não é necessariamente tarde da noite ou madrugada.

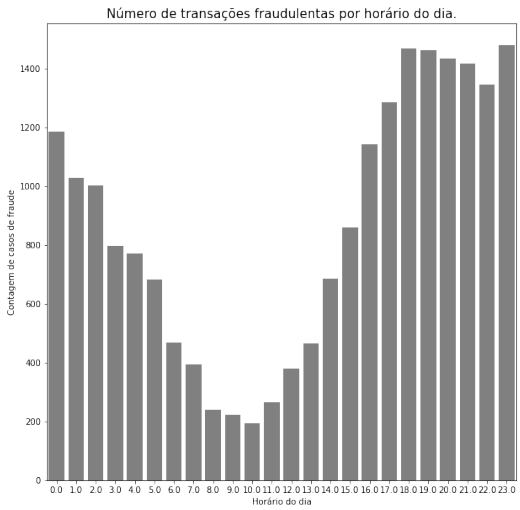

Figura 18: Contagem de transações fraudulentas por horário do dia.

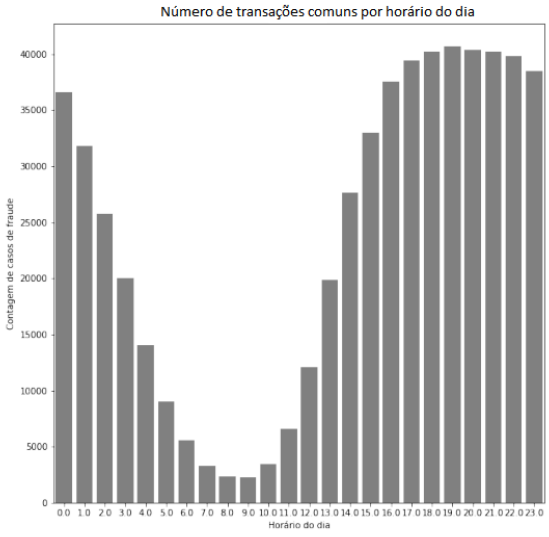

Figura 19: Contagem de transações comuns por horário do dia.

Pode-se notar pelo gráfico da Figura 19 que todas as transações com cartão aumentam ao longo do dia e também atingem seu pico no mesmo horário que o pico de transações fraudulentas atingem.

Por último, há também disponivel no dataset as variáveis dist1 e dist2 que representam, respectivamente, a distância da casa e do trabalho do dono do cartão ao local em que ocorreu a transação. O histograma de distribuição das variáveis dá a entender que as fraudes costumam ocorrer próximo do local de trabalho e de residência do dono do cartão, assim como as transações comuns, indicado na Figura 20

Em resumo, o aumento percentual do valor de cada variável no universo fraudulento pode ser observado na tabela abaixo.

Variavel	Valor	Aumento percentual de presença no universo fraudulento
ProductCD	C	300%
Card6	Crédito	100%
DeviceType	Mobile	25%

Tabela 6: Resumo das variáveis e suas mudanças de representatividade entre universo de transações comuns e transações fraudulentas.

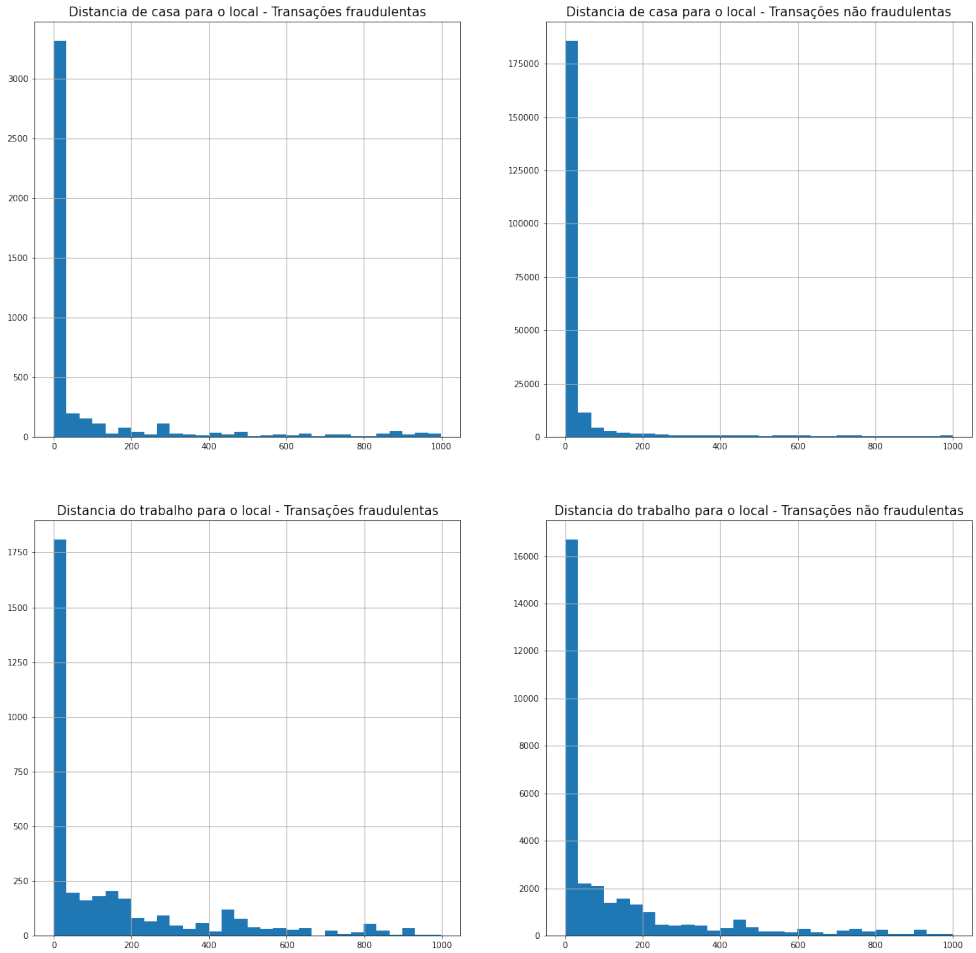

Figura 20: Distâncias entre residência do proprietário do cartão e local da transação; distância entre local de trabalho e local da transação.

Um resumo das correlações entre a variavel dependente (isFraud) e as demais pode ser observado no mapa de calor da Figura 21. As variáveis e seus valores mais correlacionados com a fraude, segundo o mapa, são *ProductCD* com o valor *C* e *card6* com o valor *credit*.

2.2 Resample dos dados

2.2.1 Undersample

A técnica de undersample foi aplicada no conjunto de dados estudado. Foram retiradas amostras do conjunto de transações comuns em uma quantidade igual ao número

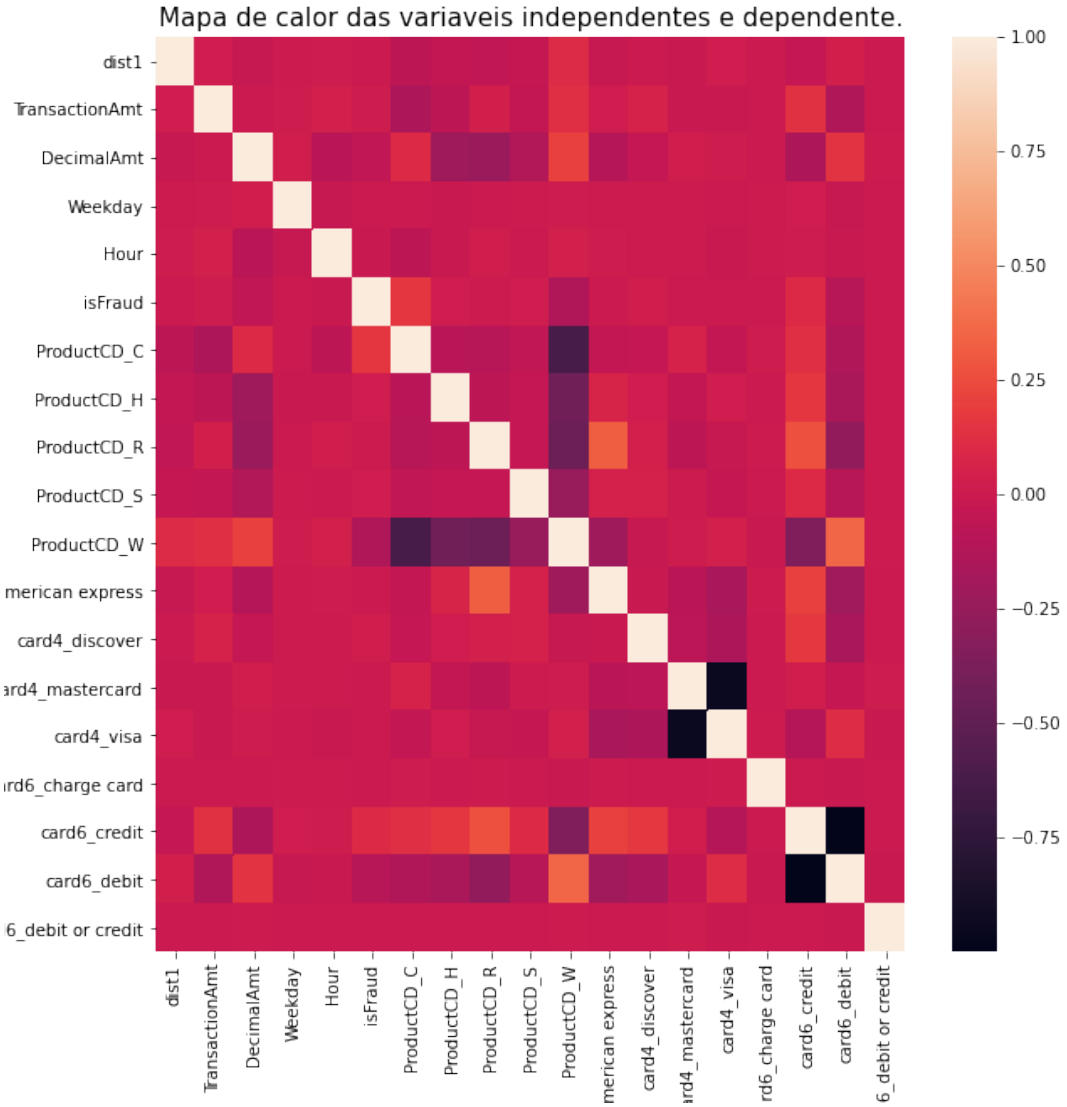

Figura 21: Distâncias entre residência do proprietário do cartão e local da transação; distância entre local de trabalho e local da transação.

de transações fraudulentas no *dataset*. Dessa maneira, era esperado obter um dataset menor que possuísse o mesmo número de transações fraudulentas e comuns. Como pode ser obserdado na contagem de casos fraudulentos e comuns apresentados na Figura 22 a ténica aplicada foi bem sucedida.

2.2.2 SMOTE

Como o undersample é uma técnica de amostragem aleatória não se espera que após aplicá-la os valores das variaveis estejam homogêneos para a classe que foi amostrada. Ou seja, não é esperado que os valores das features da classe de transações comuns estejam bem

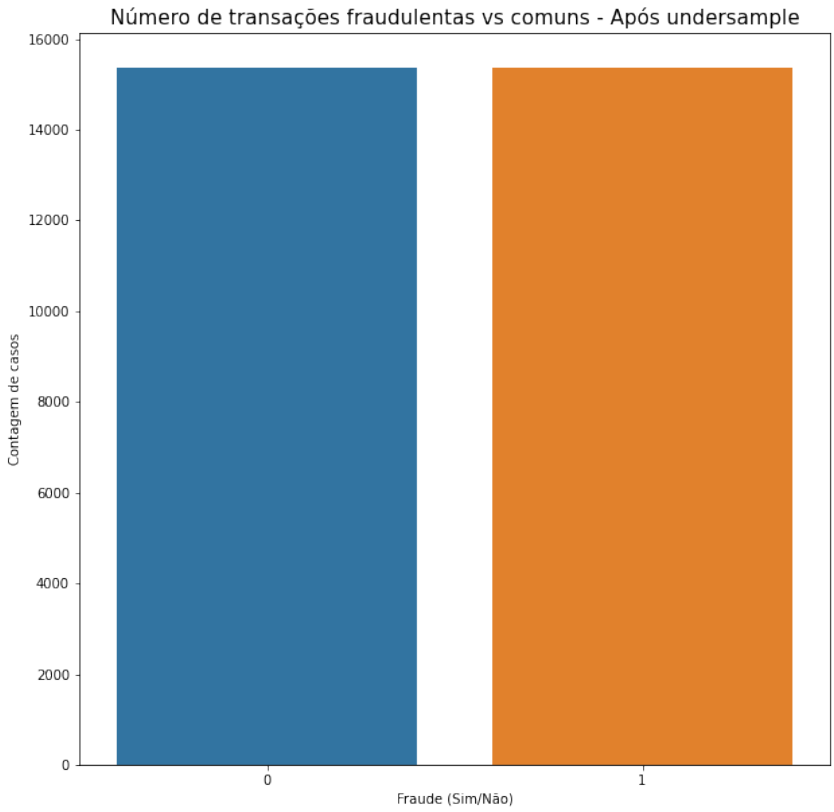

Figura 22: Distâncias entre residência do proprietário do cartão e local da transação; distância entre local de trabalho e local da transação.

representados pelos valores amostrados. Para resolver isso, será aplicada a técnica SMOTE com o objetivo contrário. Ao invés de esperar que o *sample* traga poucas amostras que representem o todo, serão geradas novas amostras cujos valores das features represente sua vizinhança. Após aplicar a técnica esperam-se novas amostras de transações fraudulentas numa quantidade igual a transações comuns e com valores de features inéditos.

Para comparação, pode-se observas nas Figuras 23 e 24 a mudança na distribuição das variáveis *dist1* e *transactionAmt* após a aplicação da técnica. Isso é, antes da aplicação as amostras de fraude eram representadas por poucos dados e valores das variáveis em questão. Após a aplicação é possível observar uma gama maior de valores para as variáveis dos casos de fraude. Foram gerados novos amostras de fraudes com o objetivo de igualar as duas quantidades de ocorrência. É possível observar na Figura 25 que isso foi obtido com sucesso. Para as demais variáveis, toda distribuição dos novos dados sintéticos permanece semelhante ao dos dados originais.

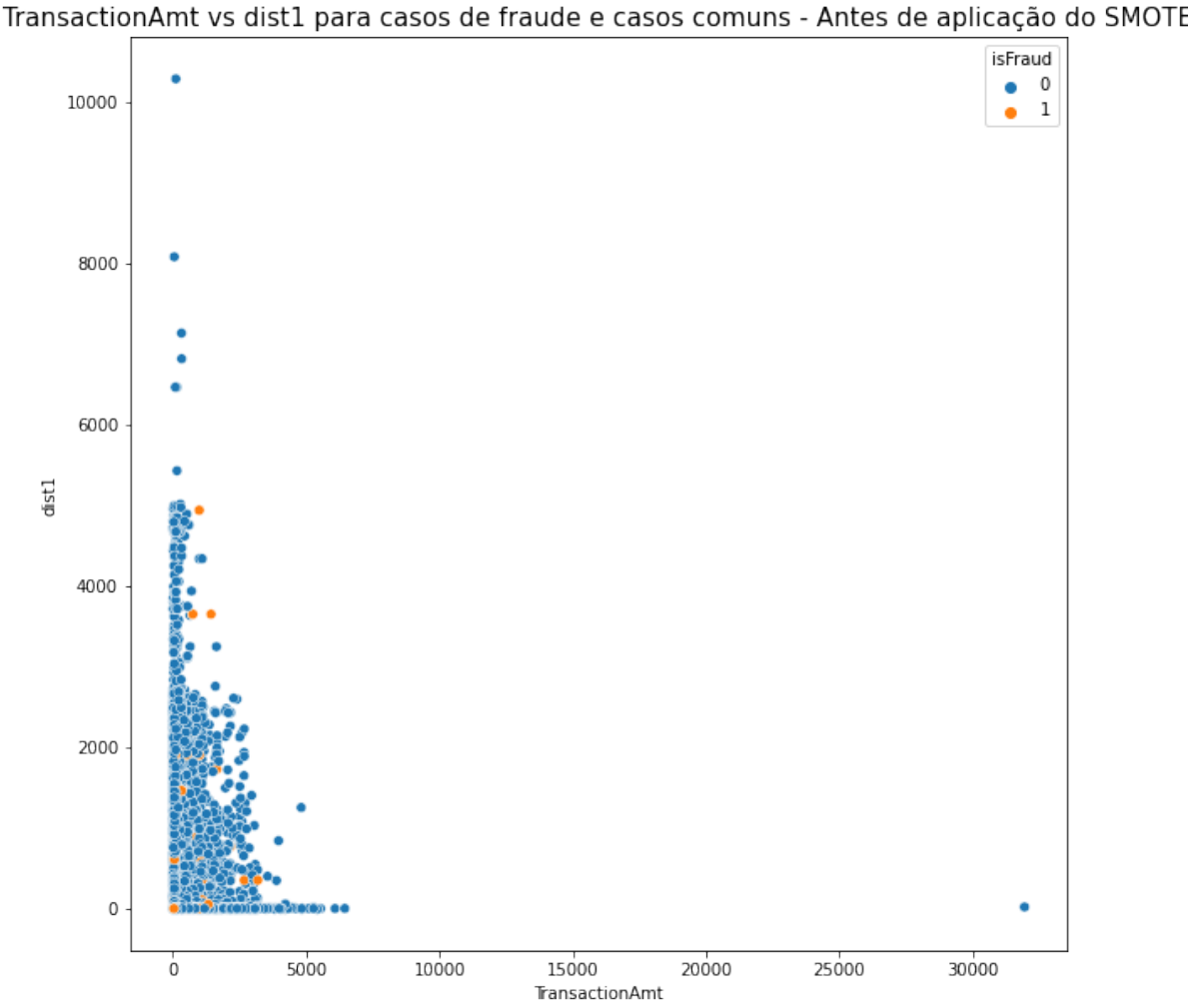

Figura 23: Distribuição dos valores das features *dist1* e *TransactionAmt* em relação a amostras dos dois universos antes da aplicação do SMOTE

2.3 Resultados e discussão

2.3.1 Métricas de avaliação de *machine learning*

Foram utilizados três modelos mencionados na revisão bibliográfica para predição de fraudes: *decision tree*, *adaBoost* e *random forest*. Os modelos foram aplicados em três conjuntos distintos que derivam do mesmo: o original, amostrado com undersample e aplicado a técnica SMOTE. Os resultados serão comparados utilizando-se de diversas métricas já apresentadas na seção de revisão bibliográfica.

Nota-se na tabela 7 que a acurácia dos testes realizados a partir de métodos sem

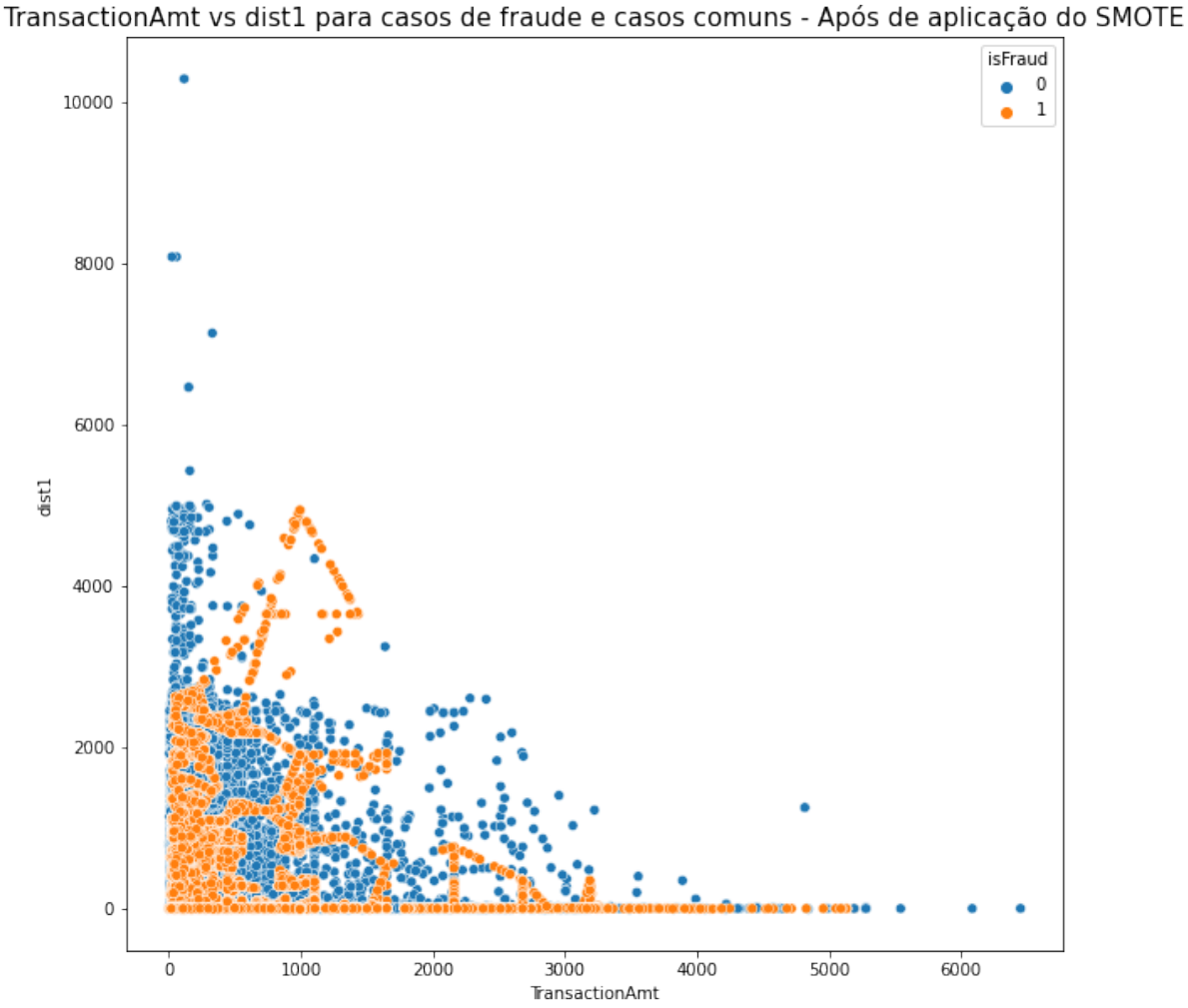

Figura 24: Distribuição dos valores das features *dist1* e *TransactionAmt* em relação a amostras dos dois universos após aplicação do SMOTE

aplicação de técnicas de amostragem possuem valores muito próximos (96.4, 96.6, 96.5) que são justamente a proporção de fraudes e transações comuuns dentro do *dataset*. Esse é um problema comum de uma modelagem que não trata o balanceamento das classes. O modelo se encontra em *overfit* pois repete as mesmas proporções vistas no conjunto de treino.

A métrica de *recall* representa o percentual de casos de fraudes identificados no conjunto de fraudes totais. Ou seja, um valor de 1 para a métrica seria seu valor máximo pois o modelo seria capaz de identificar todas as fraudes. Nota-se que para os testes que não tratam o problema de balanceamento de classes, os modelos testados são incapazes de

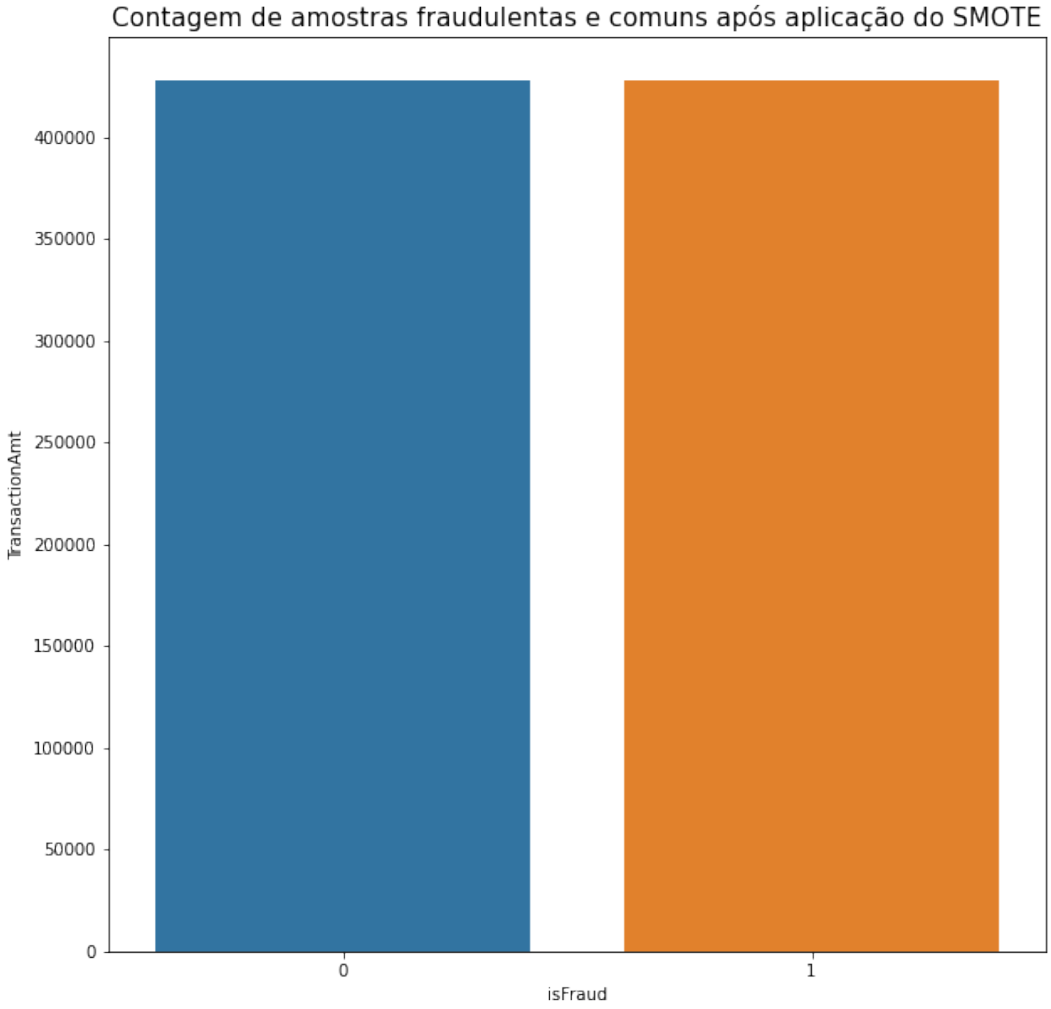

Figura 25: Contagem de amostras fraudulentas e comuns após aplicação do SMOTE.

identificar as fraudes, possuindo valores muito próximos a zero. Enquanto para os testes que fizeram esse tratamento a métrica assume um valor que chega a ser seis vezes superior, o que reforça a percepção de *overfit* para os testes realizados com *datasets* não amostrados.

Através da tabela 7 nota-se que entre todas as métricas apresentadas, os modelos que foram calibrados com os dados sem nenhum tipo de amostragem performaram pior do que os outros. Mostra-se a força e importância da aplicação de amostragem dos dados para problemas com dados desbalanceados e resolução do problema de *overfit*. O SMOTE como técnica de amostragem se mostrou superior na maioria dos casos trazendo um ganho de performance expressivo.

Um caso que chama atenção é o do modelo *adaBoost* calibrado com os dados

Modelo	Amostragem	F1 score	Precisão	Recall	Acurácia	AUC
AdaBoost	Undersample	0.153	0.088	0.579	0.771	0.750
AdaBoost	Smote	0.144	0.081	**0.632**	0.731	0.720
AdaBoost	-	0.000	0.000	0.000	0.964	0.750
RandomForest	Undersample	0.136	0.076	0.631	0.714	0.730
RandomForest	Smote	**0.208**	0.127	0.557	0.848	0.760
RandomForest	-	0.161	**0.621**	0.092	**0.966**	**0.820**
DecisionTree	Undersample	0.103	0.056	**0.625**	0.611	0.680
DecisionTree	Smote	**0.196**	0.118	0.582	0.829	0.760
DecisionTree	-	0.176	**0.557**	0.104	**0.965**	**0.790**

Tabela 7: Métricas de avaliação para os modelos de machine learning e metodos de amostragem aplicados.

originais. Nesse caso, o modelo trouxe o resultado mais indesejado possivel, uma acurácia na mesma proporção entre as classes, precisão, *f1 score* e *recall* nulos. Ocorreu um *overfit* severo. A predição do modelo era sempre que não se tratava de uma fraude e sim uma transação comum. O modelo em questão utiliza técnicas e metodos mais complexos por isso ele requer uma parametrização muito precisa para fornecer uma performance melhor. Enquanto o modelo não é parametrizado com bastante minuciosidade ele tende a *overfitar* os dados. Isso é, o modelo se ajusta e performa muito bem aos dados de treino que lhe foram apresentados e possui dificuldade em extrapolar o conhecimento adquirido para dados que não foram vistos antes. Entre todos os metodos testados, observa-se a performance inferior do modelo em questão. Justamente por se tratar de um método mais complexo que requer um tratamento especial na sua parametrização.

O modelo de árvores de decisão não envolve tantas técnicas e modelagem complexa. A grande vantagem do modelo em questão está na sua explicabilidade. No mundo real, os times de negócios querem entender como os algoritimos funcionam e como se justifica cada uma de suas escolhas e *outputs*. O modelo de árvores, pela sua estrutura mais simples e ausência de metodos matemáticos, possibilita a apresentação de um caminho de tomada de decisão.

Com as métricas apresentadas, o modelo que se saiu melhor foi o *RandomForest*. Por ser uma combinação de árvores de decisão, ele aplica o conceito de Sabedoria das Multidões, então utiliza a decisão de diversos modelos fracos (árvores de decisão) para dar uma resposta final. Isso faz com que o overfit seja diminuido, dado que cada uma das arvores será calibrada com uma parte diferente dos dados e faz com que o underfit também seja diminuido

2.3.2 Análise de custos

Para cada um dos modelos existem as medidas de recall e precision. Para cada uma delas, se aplicado no mundo real, envolvem custos. Isso é, uma fraude que não foi

identificada envolve o custo de transação como *chargeback* para o dono do cartão. Uma transação comum que foi identificada como fraude pode significar um custo de oportunidade para empresa. É necessário balancear as duas coisas para ter um modelo viável e que faça sentido. A empresa não deseja perder dinheiro com *chargeback* nem perder suas vendas.

Para realizar uma análise de custos envolvido, serão realizados os seguintes cálculos:

Custo de oportunidade = $FP * valor * 5\%$ (Baseado na margem média das vendas do varejo)

Custo com chargebacks = $FN * valor * 50\%$

Fraudes evitadas = $VP * valor$

Para um projeto desse porte se justificar dentro de uma empresa, o retorno gerado deve ser positivo. Isso é, o valor das ***fraudes evitadas*** - ***custo de oportunidade*** - ***custo com chargebacks*** deve ser maior do que o valor que seria obtido se nenhuma fraude fosse evitada: $(FN + VP) * valor * 50\%$.

Para isso foram analisados os metodos obtidos e o retorno que cada um deles traz.

Com as predições geradas pelos metodos estudados, foram obtidos os resultados ilustrados na tabela 8:

Modelo	Amostragem	Custos evitados	Custo de oportunidade	Custo de chargeback	Redução percentual do custo de chargeback
AdaBoost	Undersample	421	347	178	-13.3%
AdaBoost	Smote	505	352	136	2.1%
AdaBoost	-	0	0	388	-50.0%
RF	Undersample	500	341	138	2.6%
RF	Smote	448	177	164	**13.6**%
RF	-	108	3.9	334	-29.5%
DecisionTree	Undersample	474	383	151	-7.8%
DecisionTree	Smote	462	198	157	**13.6**%
DecisionTree	-	82	3.2	347	-34.5%

Tabela 8: Resumo de custos evitados e obtidos ao se utilizar cada um dos metódos de inteligência artificial e de amostragem. Valores em milhares de reais.

Nota-se que os modelos que trazem o melhor beneficio financeiro são aqueles que utilizam a técnica de amostragem SMOTE. Entre eles, os que performaram melhor, em empate, foram os modelo de árvore de decisão e random forest.

3 CONCLUSÃO

3.1 Considerações finais

O trabalho em questão propôs o desenvolvimento de um sistema de identificação de transações financeiras fraudulentas no varejo utilizando técnicas de aprendizado de máquina. Foram realizados estudos no âmbito de identificar os melhores classificadores para o problema em questão, assim como o uso de técnicas de amostragem para atacar diretamente o problema de desbalanceamento de classes.

O melhor resultado veio do modelo de *machine learning Random Forest* que foi capaz de generalizar muito bem as observações do conjunto de treino para os dados do *dataset* de teste. Foi obtida uma AUC de 0.82 e provou-se a sua capacidade de generalização para dados não observados anteriormente. A combinação dos modelos de árvore de decisão e *random forest* com a técnica de amostragem de dados SMOTE trouxe retorno na redução de custo em fraude para a empresa, representando 13.6% de redução nesse tipo de gasto.

A aplicação de conceitos e técnicas a respeito de aprendizado de máquina e de amostragem de dados mostrou-se ser um projeto viável no âmbito técnico e financeiro. As fraudes foram identificadas com precisão e uma redução expressiva de custo pôde ser observada.

3.2 Sugestões para pesquisas futuras

No trabalho em questão foram atacadas duas frentes principais no contexto de aplicação de *data science* para identificação de fraudes: modelos de aprendizado de máquina e técnicas de amostragem de dados. Ambos os assuntos são extremamente extensos, como demonstrado na revisão bibliográfica, e podem ser estudados mais a fundo com o objetivo de melhorar a identificação de fraudes.

Modelos de *machine learning* e ajuste fino de parâmetros: podem ser estudada a aplicação de modelos mais complexos que aplicam *boosting* e *bootstrap* como *XGBoost*, *LightGBM* além de reudes neurais com o objetivo de melhorar as métricas de identificação da classe correta. Além disso, a tunagem de parametros é uma etapa crucial para obtenção de ganho expressivo na performance quando utilizado modelos complexos.

Técnicas de amostragem: por se tratar de um contexto com relevante problema de desbalanceamento das classes, estudo de diferentes técnicas de amostragem e geração de novos dados podem ser abordados.

Obtenção de dados adicionais e *feature engineering*: dados públicos de transações financeiras de pessoas fisicas são muito dificeis de serem obtidos por toda questão de

privacidade de dados que existe. Um grande desafio é enriquecer as bases de dados com inserção de novas informações relevantes com que os modelos possam aprender mais e as fraudes fiquem melhor segmentadas. Uma sugestão de trabalho futuro é identificar possiveis *features* relevantes no contexto de fraudes e como obtê-las sem ferir a LGPD.

Velocidade de verificação de fraude: o autor não abordou a questão no presente trabalho mas deixa como sugestão. A velocidade com que os modelos são capazes de definir se uma transação eletrônica se trata de uma fraude é importante no contexto de vendas digitais e sua relevância aumenta conforme o volume de vendas cresce. Um possível trabalho futuro é o estudo da performance da velocidade com que os modelos são capazes de retornar uma resposta ao se apresentar informações.

REFERÊNCIAS

BARBER, C. B.; DOBKIN, D. P.; HUHDANPAA, H. The quickhull algorithm for convex hulls. **ACM Trans. Math. Softw.**, Association for Computing Machinery, New York, NY, USA, v. 22, n. 4, p. 469–483, dez. 1996. ISSN 0098-3500. Disponível em: <https://doi.org/10.1145/235815.235821>.

CHAWLA, N. et al. Smote: Synthetic minority over-sampling technique. **J. Artif. Intell. Res. (JAIR)**, v. 16, p. 321–357, 06 2002.

CHAWLA, N. V. **Data mining and knowledge discovery handbook**: Data mining for imbalance datasets: an overview. [S.l.]: Springer, 2005.

DRUMMOND, C.; HOLTE, R. C4.5, class imbalance, and cost sensitivity: Why under-sampling beats oversampling. **Proceedings of the ICML'03 Workshop on Learning from Imbalanced Datasets**, 01 2003.

FREUND, Y.; SCHAPIRE, R. Experiments with a new boosting algorithm. In: **ICML**. [S.l.: s.n.], 1996.

_____. A decision-theoretic generalization of on-line learning and an application to boosting. In: **COLT 1997**. [S.l.: s.n.], 1997.

KAMALOV, F.; DENISOV, D. Gamma distribution-based sampling for imbalanced data. **ArXiv**, abs/2009.10343, 2020.

LIAW, A.; WIENER, M. Classification and regression by randomforest. **Forest**, v. 23, 11 2001.

LIU, X.-Y.; WU, J.; ZHOU, Z.-H. Exploratory undersampling for class-imbalance learning. **Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on**, v. 39, p. 539 – 550, 05 2009.

LUCAS, Y.; JURGOVSKY, J. Credit card fraud detection using machine learning: A survey. **ArXiv**, abs/2010.06479, 2020.

MITCHELL, T. M. **Machine Learning**. 1. ed. USA: McGraw-Hill, Inc., 1997. ISBN 0070428077.

NOWOZIN, S. Improved information gain estimates for decision tree induction. **ArXiv**, abs/1206.4620, 2012.

PROVOST, F.; FAWCETT, T. Robust classification for imprecise environments. **Machine Learning**, v. 42, p. 203–231, 01 2001.

SHEN, X.; XU, Q.; XUE, X. Nonlinear monte carlo method for imbalanced data learning. 10 2020.

SONG, B. et al. Roc operating point selection for classification of imbalanced data with application to computer-aided polyp detection in ct colonography. **International journal of computer assisted radiology and surgery**, v. 9, 06 2013.

TAN, P.-N.; STEINBACH, M.; KUMAR, V. **Introduction to Data Mining**. Us ed. Addison Wesley, 2005. Hardcover. ISBN 0321321367. Disponível em: <http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0321321367>.

WEI, W. et al. Effective detection of sophisticated online banking fraud on extremely imbalanced data. **World Wide Web**, v. 16, p. 449–475, 2012.