UMA ABORDAGEM PARA DETECÇÃO DE FRAUDES NO CONTEXTO DE TRANSAÇÕES BANCÁRIAS

Dunfrey P. Aragão
Ítalo de Pontes Oliveira
Fernando Felix

Sumário

Introdução

De acordo com *The World Payments Report 2019*, feito pela Capgemini Financial Services Analysis (2019)[1], o montante operacional movimentado por transações financeiras em 2019 corresponde à 680 Bi de dólares ao redor do mundo, isso sem contabilizar operações que envolvem o uso de dinheiro físico. Para fins de comparação, esse valor chega a superar o Produto Interno Bruto (PIB) da Argentina contabilizado para o ano de 2019, cerca de 477 Bi de dólares[2], como também é maior que o dobro do PIB de países como Colômbia, Finlândia, Egito, Chile e Portugal. Em um único dia, empresas como a Visa chegam a processar mais que 100 milhões de transações financeiras, correspondendo à cerca de 24 mil novas operações a cada segundo[3].

Durante o processo de uma transação financeira, ocorrem uma série de procedimentos[4]: 1) o estabelecimento comercial que está vendendo o produto, ou prestando um serviço ao cliente, coleta os dados do cartão de crédito, tais como o número do cartão, data de validade, código de segurança, entre outros, por meio de uma maquineta ou por fornecimento manual, como ocorrem em sistemas de e-commerce, e repassa esses dados para a rede de cartão de crédito; 2) a rede de cartão de crédito deverá autenticar as informações disponibilizadas e solicitar que a instituição bancária aprove a transação; 3) o órgão bancário realiza o pagamento ao estabelecimento comercial que realizou a venda e aprova a transação.

Desta forma, cabe às instituições de rede de cartão de crédito a responsabilidade de autenticar as informações do cliente fornecidas pelo estabelecimento comercial, como também, validar a transação foi realizada pelo cliente. Dessa maneira, é imprescindível que o sistema de validação funcione com baixo tempo de resposta e que, ao mesmo tempo, seja confiável, impedindo que transações fraudulentas ocorram.

Este documento, propõe uma abordagem capaz de identificar automaticamente o tipo de uma transação com acurácia de 74.4%**,** utilizando o método F1-score para obtenção do resultado. A abordagem foi validada em uma base de dados real e publicamente disponível. A solução proposta está entre os Top-70 no desafio Zindi[5]. A Seção Metodologia apresenta uma descrição detalhada sobre como os experimentos foram conduzidos.

Objetivos

Objetivo Geral

- Identificar de maneira automática se uma transação foi fraudulenta ou genuína.

[1] https://worldpaymentsreport.com/resources/world-payments-report-2019/ (acesso em 07/12/2019).
[2] http://worldpopulationreview.com/countries/countries-by-gdp/ (acesso em 07/12/2019).
[3] https://usa.visa.com/run-your-business/small-business-tools/retail.html (acesso em 07/12/2019).
[4] https://www.mastercard.com.br/pt-br/estabelecimentos/comece-aceitar/processo-pagamento.html (acesso em 07/12/2019).
[5] https://zindi.africa/competitions/xente-fraud-detection-challenge (acesso em 07/12/19).

Objetivos Específicos

- Estimar o impacto financeiro causado pelo sistema de detecção de fraude.

Metodologia

Para identificação de transações fraudulentas foi adotada a base de dados fornecida no desafio organizado pela Zindi[6]. Os dados são oriundos da plataforma Xente[7], empresa voltada a prestação de serviços financeiros e comércio eletrônico que atende a mais de 10 mil clientes em Uganda. O conjunto de dados inclui uma amostra aproximada de 140 mil transações, que ocorreram entre os dias 15 de novembro de 2018 e 14 de março de 2019. Os dados são divididos em treinamento (conjunto de transações que ocorreram entre 15 de novembro de 2018 e 13 de fevereiro de 2019 com prévia identificação do tipo de transação - fraude ou genuíno) e teste (conjunto de transações que correspondem ao período de 13 de fevereiro a 14 de março de 2019, sem identificação do tipo de transação).

Os principais desafios envolvendo detecção de fraude estão relacionados ao alto nível de desbalanceamento dos dados, visto que cerca de 0.002% das transações correspondem a fraudes. Além disso, transações fraudulentas nem sempre se comportam de maneira suspeita, muitas vezes se passam despercebidas.

Considerando tais dificuldades, a abordagem adotada nesse projeto é apresentada na Figura 1, que mostra o pipeline completo. Cada etapa é descrita em detalhes nas seções seguintes. Em resumo, são elas:

- **Pré-processamento dos dados:** Nessa etapa buscou-se identificar existência de dados faltantes, verificação de repetição de dados no conjunto de treinamento, realizada análises descritivas e tendências nas transações fraudulentas e, por fim, correlação dos atributos no conjunto de treinamento;
- **Balanceamento dos dados:** Observado o alto nível de desbalanceamento nos dados, utilizou-se diferentes técnicas de balanceamento de dados usando *oversampling*[8], visto que *undersampling*[9] resultaria em um descarte significativo no montante de dados disponíveis (cerca de 99% das instâncias são da classe não-fraudulenta). Com base nisso, observou-se um melhor desempenho do classificador ao usar técnicas de balanceamento baseadas em *oversampling*;
- **Identificação de *outliers*:** Aplicação de técnicas ao experimento que consistem em identificar instâncias que se diferencie das demais, e usar essa informação como uma novo atributo para o modelo;
- **Treinamento dos dados:** Por fim, treinou-se um modelo de aprendizado supervisionado chamado Catboost[10], para identificar se uma instância é ou não fraudulenta. Os resultados foram submetidos na plataforma do Zindi e a

[6] https://zindi.africa/competitions/xente-fraud-detection-challenge (acessado em 07/12/2019).
[7] https://zindi.africa/competitions/xente-fraud-detection-challenge/data (acessado em 07/12/2019).
[8] Réplica de dados com menor índice até que os níveis do tipo de transação sejam iguais.
[9] Descarte de dados com maior índice até que os níveis do tipo de transação sejam iguais.
[10] https://catboost.ai/ "CatBoost is a high-performance open source library for gradient boosting on decision trees" (acessado em 07/12/2019).

metodologia proposta está entre os top-70 para os mais de mil cientistas de dados participantes da competição.

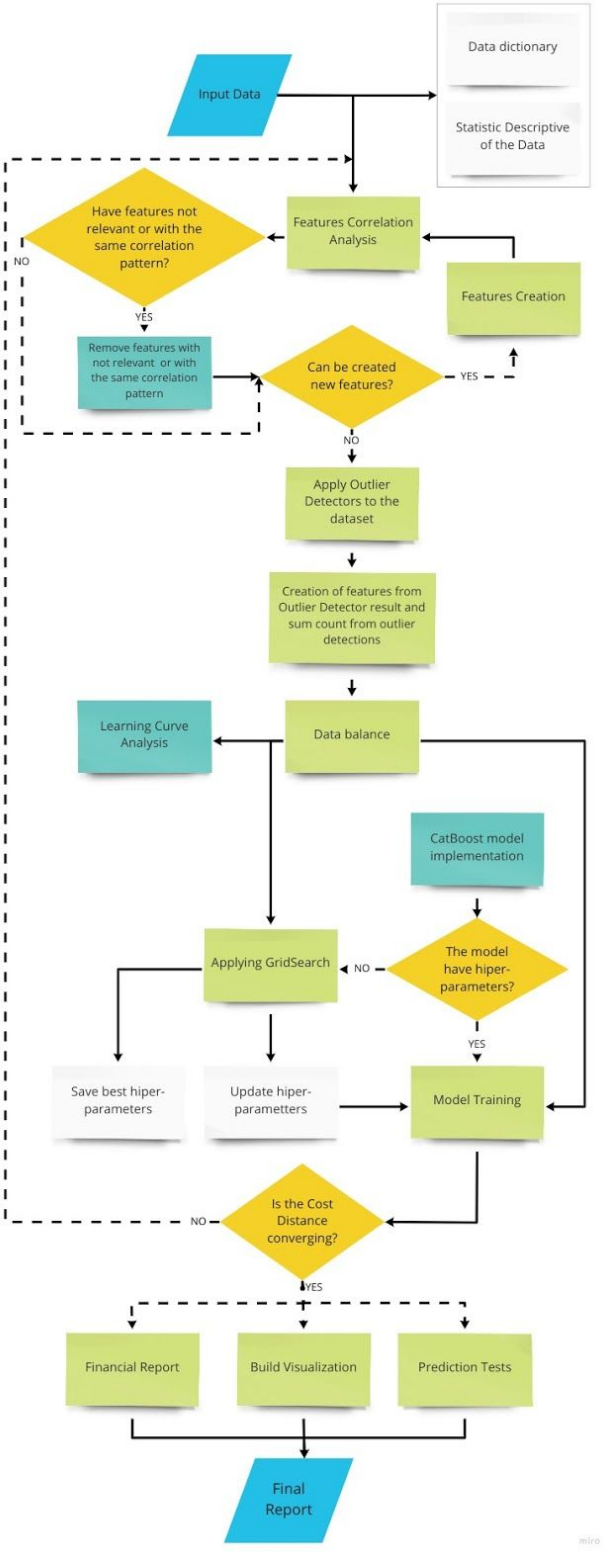


Figura 1: Fluxo metodológico de ações aplicadas ao conjunto de dados para classificação do tipo de transação.

Compreensão dos Dados

A primeira etapa da *pipeline* (fluxo de trabalho) consiste na compreensão dos dados fornecidos pelo desafio[11]. A Figura 2 apresenta o fluxo de ações realizadas na primeira etapa. O *pipeline* consiste nas seguintes sub-etapas: 1) seguindo a metodologia CRISP-DM, neste momento o foco é em criar dicionários de dados e analisá-los de maneira descritiva, isso nos forneceu melhor entendimento sobre os dados e o contexto de negócio envolvido; 2) análise de correlação entre os atributos para identificação e remoção dos atributos altamente correlacionadas; 3) criação de novos atributos, volta ao passo (2) até que não haja mais atributos correlacionados entre si e que devam ser removidos.

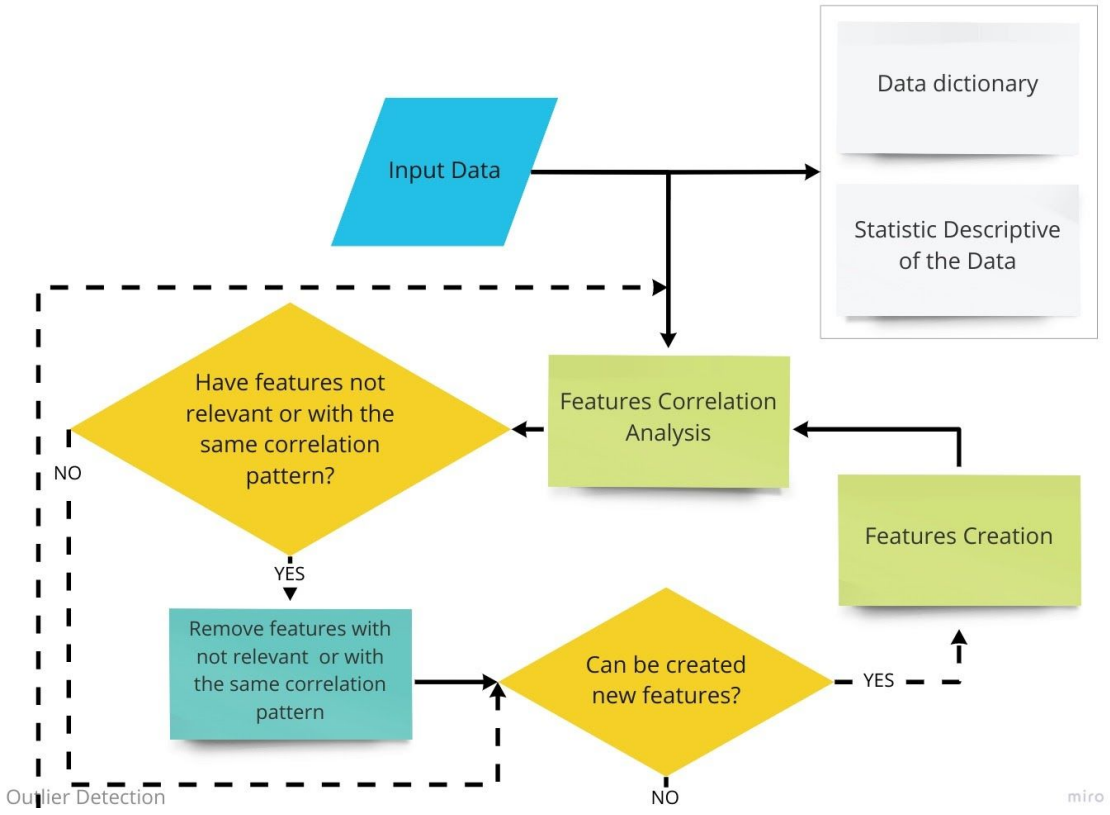


Figura 2: Fluxo metodológico de ações aplicadas ao conjunto de dados na etapa de obtenção e análise dos dados.

Dicionário de Dados

O conjunto de dados engloba um transações realizadas e, para cada uma destas, existe um grupo de atributos, que são:

- **TransactionId:** Identificador único da transação na plataforma.
- **BatchId:** Número único do conjunto de transações enviadas por processamento.
- **AccountId:** Identificador único do usuário na plataforma.
- **SubscriptionId:** Identificador único do usuário subscrito.
- **CustomerId:** Identificador único anexo a AccountId.

[11] https://zindi.africa/competitions/xente-fraud-detection-challenge/data (acessado em 08/12/2019).

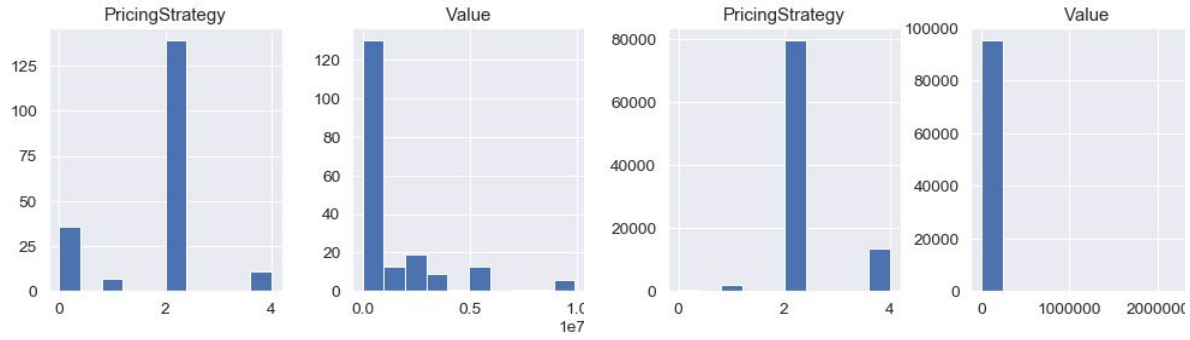
- **CurrencyCode:** Moeda do país.
- **CountryCode:** Código do país.
- **ProviderId:** Fornecedor de origem do item comprado.
- **ProductId:** Identificador do item comprado.
- **ProductCategory:** Os ProductIds são organizados em categorias de acordo com os produtos.
- **ChannelId:** Identifica a plataforma usada pelo usuário, como *web*, sistema Android ou IOS, *pay later* ou *checkout*.
- **Amount:** Valor da transação. Se o valor é positivo, significa que o usuário utilizou em sua conta a opção de débito, se negativo, o usuário utilizou crédito.
- **Value:** Valor absoluto do valor da transação.
- **TransactionStartTime:** Dia e hora da transação.
- **PricingStrategy:** Categoria de precificação para venda disponibilizado pelo Xente.
- **FraudResult:** Status da transação: 1) fraude; ou 0) não-fraudulenta.

Observação**:** A nomenclatura usada para identificar os rótulos foram fraudulentas e genuínas. Para maior clareza, optamos por identificar as operações não-fraudulentas como operações genuínas.

Análise descritiva dos dados

Análise dos atributos para as instâncias genuínas e fraudulentas

Para identificar se algum atributo reflete em padrões típico das transações fraudulentas, utilizamos histogramas observar cada distribuição. A Figura 3 apresenta a forma que as transações financeiras genuínas e fraudulentas se comportam em relação ao valor empregado na transação (*Value*) e ao tipo de estratégia usado (*PrincingStrategy*).



Dados Fraudulentos Dados Genuínos

Figura 3: Histograma para os atributos *PricingStrategy* e *Value.*

Desta maneira, pode-se observar que as transações genuínas e fraudulentas possuem um padrão muito semelhante para o atributo *PrincingStrategy*, com a diferença que quase não houve transações fraudulentas para o *PrincingStrategy* igual a um e nenhuma do tipo zero. Além disso, percebe-se que as operações fraudulentas possui transações com valores bem concentrados em uma certa região.

Como os dados estão distribuídos considerando os diferentes atributos?

Nesse ponto da análise descritiva, observamos quais evidências temos acerca dos dados fraudulentos em relação a cada atributo. A seguir, listamos os aspectos mais importantes encontrados para cada atributo:

Operation

Foi identificado um viés nesse atributo, devido proporção muito maior de transações de débito terem sido feitas para os dados fraudulentos. A Tabela 1 mostra os valores encontrados nessa análise.

Operações	**Fraudulentas**	**Genuínas**
Débito	97%	60%
Crédito	3%	40%

Tabela 1: Proporção de como as transações fraudulentas e genuínas estão distribuídas para cada tipo de operação (débito ou crédito).

PrincingStrategy

Foi identificado um maior desvio padrão nas transações fraudulentas. Que significa que a estratégia usada são mais dispersas, entretanto em valores mais baixos.

Value

O valor médio das transações genuínas foi de $672, com desvio padrão de $3.995, já nas transações fraudulentas esse valor foi de $1.560.156, com desvio padrão de $2.082.015. Ademais, as seguintes observações foram feitas:

- 98.9% das Fraudes: valor da transação > média de todas as transações;
- 27% das Fraudes: valor da transação > média de todas as transações fraudulentas;
- 80.6% das Genuínas: valor da transacoes < média de todas as transações;
- 20% das Genuínas: valor da transação > média de todas as transações genuínas.

Em relação aos demais atributos relacionados ao valor da transação, podemos destacar as seguintes observações:

- CHANNEL=**3**: Canal onde aconteceu o maior número de transações fraudulentas;
- OPERATION=**1**: Essa operação é a que possui maior probabilidade de haver fraude;
- PRODUCT=**15**: Produtos com maior índice de fraude;
- PROVIDER_ID=**[1, 3, 5]**: São os fornecedores onde ocorrem o maior número de fraudes;
- PRODUCTCATEGORY=**9** (*financial services*): A categoria de serviços financeiros foi aquele com o maior índice de fraude. Para melhor observar esse viés, foi contabilizado o número de transações por categoria, normalizados e visualizados no histograma apresentado na Figura 4 . Percebe-se que as transações fraudulentas (representadas em azul), incidem principalmente em operações do tipo *Financial Services*, como também *Airtime* e *Utility Bill*.

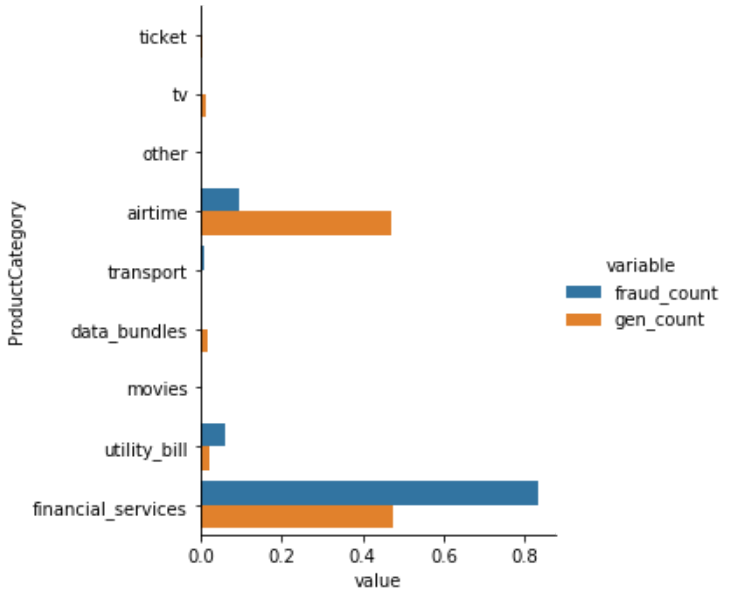


Figura 4: Histograma para as diferentes categorias de produto considerando as transações fraudulentas e genuínas.

Manipulação dos Dados

Realizada a compressão geral dos dados, foi observado a oportunidade de criar novos atributos, como também remover atributos que não possuíam relação com o atributo alvo (*FraudResult*) ou que possuem alta correlação com outros atributos.

Remoção de Atributos

Os atributos referentes a valores de identificadores únicos e relativos ao usuário são removidos pois seus conteúdos não possuem influência na classificação do tipo de transação como, por exemplo, o identificador do cliente. Além destas, os atributos que possuem valores constantes também foram removidos, como por exemplo, *CountryCode* que identifica o código do país onde a transação ocorreu, como todos os dados foram coletados em Uganda, esse valor é constante para todas as instâncias e pode ser descartado pois não trouxe qualquer contribuição para o modelo. Ao todo, os atributos removidos nessa etapa foram: *AccountId, SubscriptionId, CustomerId, CurrencyCode e CountryCode.*

Análise de Correlação de *features*

Quando duas variáveis aleatórias são correlacionadas entre si significa que ambas obedecem uma taxa de crescimento proporcional. Essa ambiguidade se não tratada irá requerer um desnecessário e maior processamento e alocação de memória e, portanto, o mais indicado é que seja feito o descarte de uma das variáveis. A Figura 5 mostra a coeficiente de correlação entre os atributos numéricos, perceba que: 1) os atributos *Amount* e *Value* são altamente correlacionados, nesse caso, optou-se por remover o atributo *Amount*; 2) Existe uma correlação moderada entre o tipo de transação (fraudulenta ou

genuína) e o valor da transação (representado por *Amount* e *Value*); 3) O atributo *PrincingStrategy* não possui correlação significativa com nenhum outro atributo.

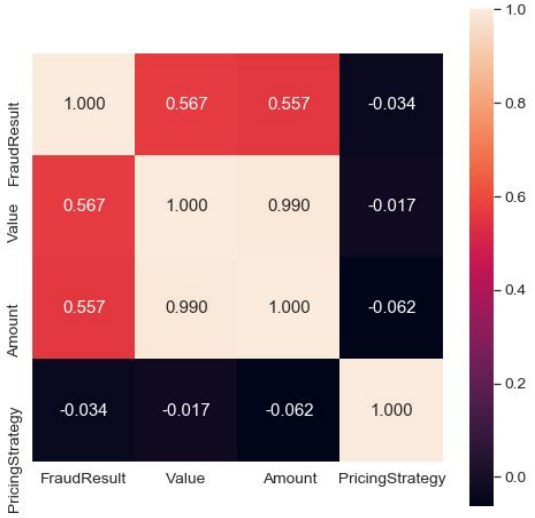


Figura 5: Correlação de Pearson entre os atributos do tipo numérico.

Criação de novos atributos com base em outros atributos

Foram geradas novos atributos com base nos atributos já existentes, são elas:

- **Operation:** Identifica se a transação é crédito (1) ou débito (-1).
- **Hour:** Hora do dia que foi realizada a transação.
- **DayOfWeek:** Dia da semana que foi realizada a transação.
- **WeekOfYear:** Semana do ano que foi realizada a transação.
- **Vl_per_weekYr:** Razão entre o valor da transação e a posição dessa semana no ano.
- **Vl_per_dayWk:** Razão entre o valor da transação e a posição desse dia na semana.
- **Vl_per_dayYr:** Razão entre o valor da transação e a posição desse dia no ano.
- **ValueStrategy:** Classificação de valores das transações. A verificação é valor de uma única transação ser superior, ou não, a média de todas as transações multiplicado por um valor n. De acordo com esta análise, a transação entra em uma faixa classificatória (n : 0 a 5).

Criação de novos atributos usando detectores de *outliers*

Em estatística, um *outlier* é aquela instância que se comporta de maneira diferente da maioria das instâncias de um mesmo conjunto de dados. Assim, um questionamento que pode surgir é “há possibilidades de incluir novos atributos no conjunto de dados usando detectores de *outlier* para melhorar o desempenho do modelo?”. Com base nisso, foram adotados três detectores de *outliers* diferentes, que são: o Isolation Forest, o Locally Selective Combination in Parallel Outlier Ensembles (LSCP) e o KNN[12], todos esses descritos a seguir.

[12]Os algoritmos citados estão implementados em: https://scikit-learn.org/ (acessado em 08/12/2019).

Isolation Forest

O *Isolation Forest*[13] é um algoritmo supervisionado que detecta pontos fora da curva padrão no conjunto de dados, ou anomalias. O algoritmo é baseado na estrutura de árvores de decisão, em que os atributos são utilizados para criar nós ao longo da árvore e, quanto mais atributos em comum, menos camadas são necessárias para diferenciar as instâncias. A partir da árvore de decisões gerada pelo algoritmo, é possível identificar anomalias que não compartilham o mesmo padrão de atributos (nesse contexto, as transações fraudulentas).

Locally Selective Combination in Parallel Outlier Ensembles (LSCP)

O LSCP[14] é um algoritmo não-supervisionado que combina, em uma única aplicação, diversas metodologias base de outros detectores, adotando aquela cuja ênfase nos dados locais melhor se adapta (abordagem conhecida como *ensemble*). Essa variedade de aplicação de metodologias em detectar *outlier* considera o comportamento dos dados para aquele método que melhor se encaixe ao problema.

K-Nearest Neighbors (KNN)

O KNN trabalha com a ideia de distância de um determinado ponto para o(s) seu(s) k-ésimo vizinho(s). Assim, a ideia é que, se um ponto for *outlier*, ele estará mais distante das demais instâncias.

Geração de Novos Atributos

Após a finalização do processo de identificar *outliers* usando os três detectores descritos previamente, gerou-se novos atributos indicando se a instância se comporta como *outlier* ou não conforme mostrado na Figura 6. Essa abordagem teve como objetivo identificar as transações fraudulentas, partimos do pressuposto que as instâncias fraudulentas deveriam se comportar como *outlier*.

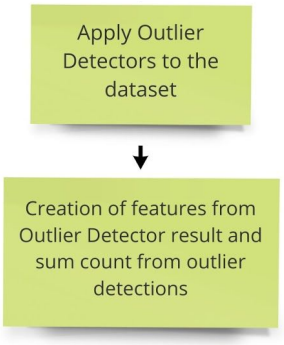


Figura 6: Fluxo metodológico de ações aplicadas ao conjunto de dados para detecção de outliers e criação de novas *features* relacionado a este tipo de detecção.

[13] https://towardsdatascience.com/outlier-detection-with-isolation-forest-3d190448d45e (acesso em 08/12/2019).
[14] Código-fonte publicamente disponível: https://pyod.readthedocs.io/en/latest/ (acesso em 08/12/2019).

Os novos atributos criados foram:

- **IsolationForest:** Classe estimada pelo *IsolationForest*, 1 se é *outlier* e 0 se for normal.
- **LSCP:** Classe estimada pelo LSCP, 1 se é *outlier* e 0 se for normal.
- **KNN:** Classe estimada pelo KNN, 1 se é *outlier* e 0 se for normal.
- **CountDetection:** Soma de total das classes previstas pelos detectores de *outliers*. Por exemplo, se nenhum dos detectores classificou uma instância como *outlier*, esse valor será zero, se apenas o KNN detectou uma certa instância como *outlier,* então esse valor será 1, mas se todos os classificadores detectaram uma instância como *outlier,* então o valor desse atributo será 3.

A Figura 7 mostra um novo gráfico contendo o coeficiente de correlação entre os atributos após a criação dos atributos descritos nessa seção.

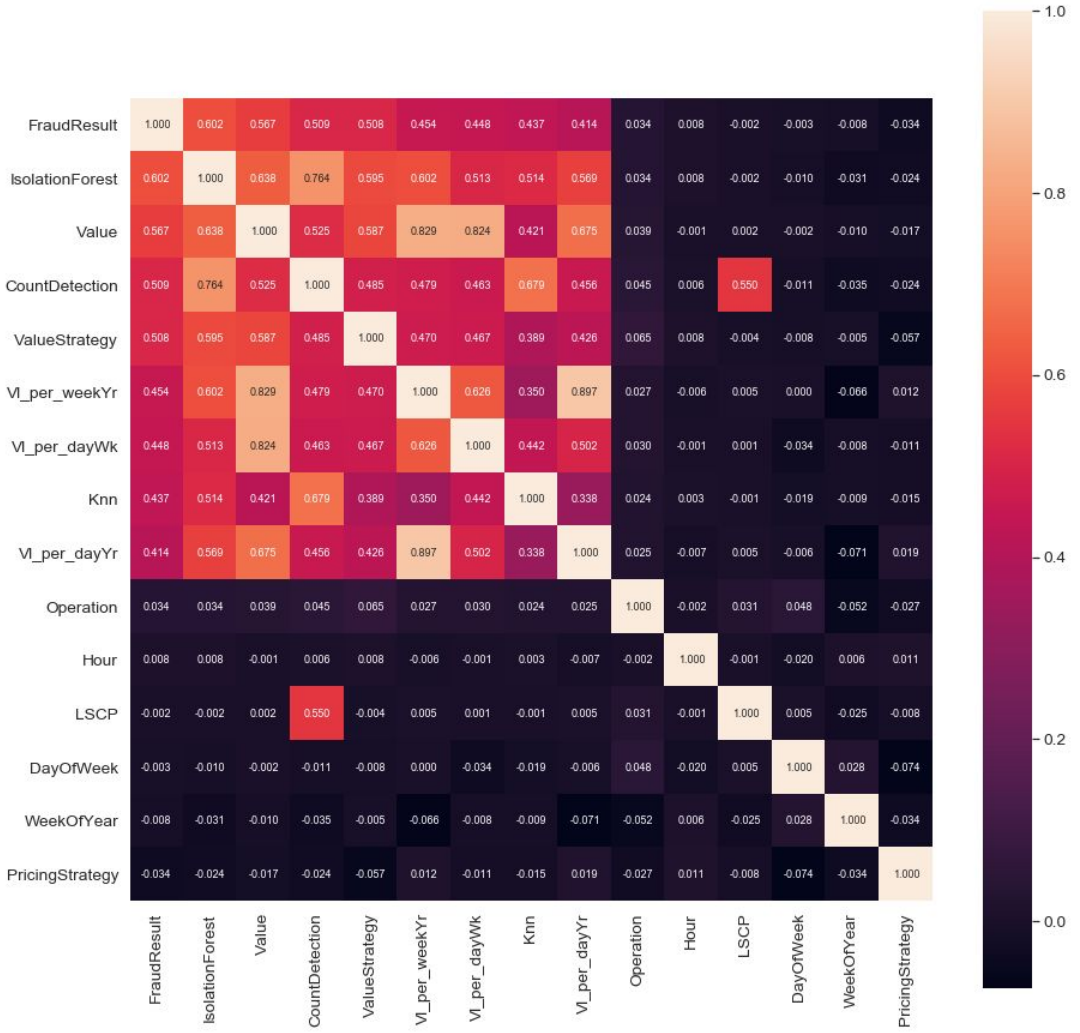


Figura 7: Gráfico de correlação de *features* do conjunto de dados com implementação de remoção e adição de novas *features*.

Desenvolvimento da solução

Esta seção trata da manipulação e implementação de algoritmos computacionais destinados a resolução do problema em detectar o tipo de transação. Esta seção é subdividida em balanceamento do conjunto de dados, implementação do modelo e avaliação do modelo conforme exibido na Figura 8.

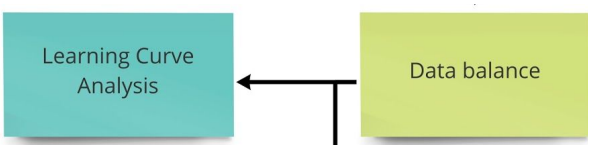


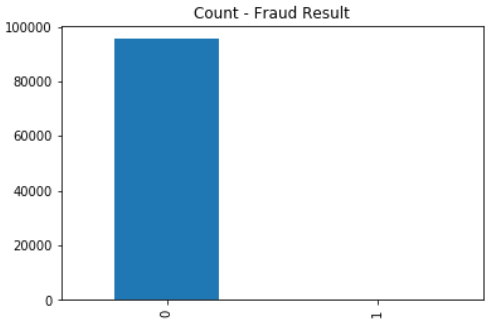
Figura 8: Etapas de balanceamento do conjunto de dados e geração de curva aprendizado.

Balanceamento do Conjunto de Dados

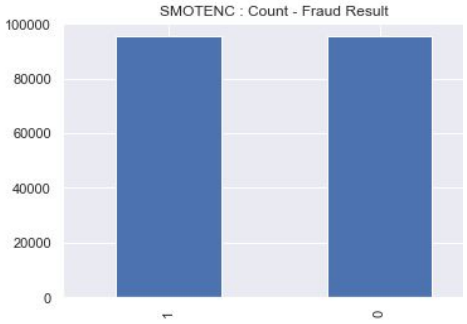
A etapa de balanceamento é importante devido alto desnivelamento no conjunto de dados referente ao tipo de classificação de cada transação. A realização da atividade de balanceamento de dados é importante uma vez que permitirá que o modelo classificatório tenha disponível para seu processo aprender a classificar o tipo da transação uma quantidade razoável de exemplos disponíveis e de iguais quantidades para ambos os tipos de tipo de transação.

Para isso, a abordagem de balanceamento do conjunto de dados foi realizada usando o método de *Oversampling*, através do uso do método SMOTENC[15], que é capaz de realizar a criação de novos elementos vizinhos (semelhantes) aos elementos que se deseja multiplicar, ou seja, é observado as características dos elementos fraudulentos, estes sofrem pequenas alterações e são multiplicados considerando. Os novos dados não possuem características distantes de uma transação real fraudulenta.

A representação quantitativa dos dados desbalanceados são apresentados na Figura 9a, enquanto a representação quantitativa dos dados balanceados são apresentados na Figura 9b.



a) Conjunto de dados desbalanceados



b) Conjunto de dados balanceados

15 https://imbalanced-learn.readthedocs.io/en/stable/generated/imblearn.over_sampling.SMOTENC.html (acesso em 08/12/2019).

Figura 9: Balanceamento dos dados fraudulentos. O SMOTENC foi aplicado ao conjunto de dados cujo tipo de transação cuja classificação é fraudulenta, gerando novos elementos similares.

Modelagem

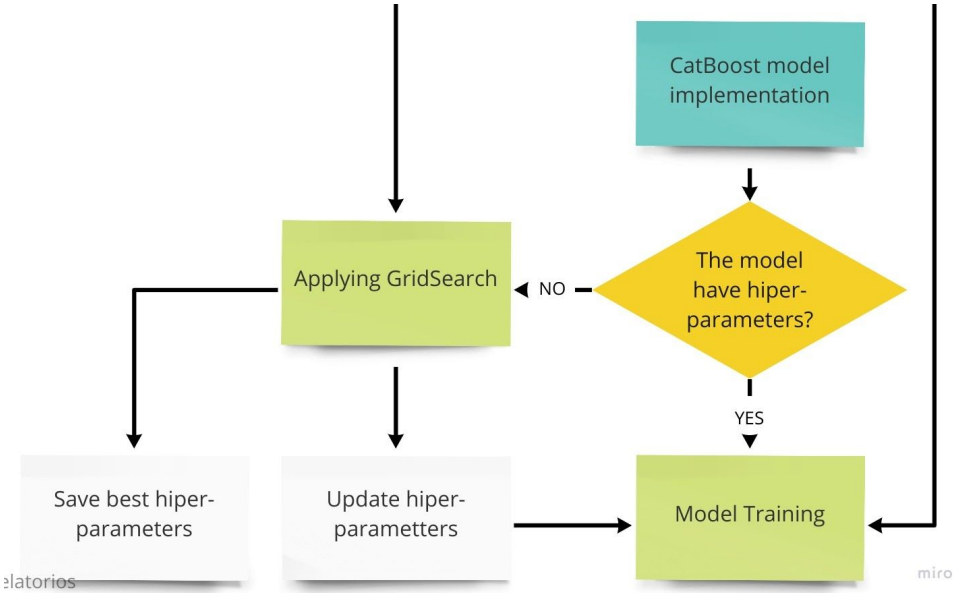


Figura 10: Fluxo metodologia de ações realizadas na criação e treinamento do modelo classificador.

CatBoost

CatBoost é uma biblioteca gradiente *boosting*, isso significa que o modelo tenta melhorar os modelos anteriores combinando-os entre si. O gradiente *boosting* trabalha com o algoritmo de gradiente sobre uma função objetiva, sendo portanto um treinamento supervisionado, utilizando um conjunto de instâncias de treinamento rotuladas como entrada e criando um modelo que tenta prever corretamente o rótulo de novos exemplos não apresentadas no treinamento pelo conjunto de dados. Além disso, o Catboost quando comparado com outros algoritmos como LightGBM, XGBoost e H2O apresentou desempenho superior para diferentes bases de dados, como também o custo computacional foi bastante competitivo[16].

Definindo os parâmetros do modelo

Para definir os melhores hiperparâmetros do modelo, adotou-se a técnica conhecida como *Grid Search*. Neste procedimento é feita a validação cruzada sob o conjunto de dados (ou, idealmente, usar um conjunto de validação separado), sem observar o conjunto de testes até o cálculo de precisão final. O resultado gerado é apresentado na Figura 11, que apresenta o F1-score[17] para o todo conjunto de dados testado com a diversidade de parâmetros submetidos.

[16] https://catboost.ai/ (acesso em 08/12/2019).
[17] https://en.wikipedia.org/wiki/Receiver_operating_characteristic (acesso em 08/12/2019).

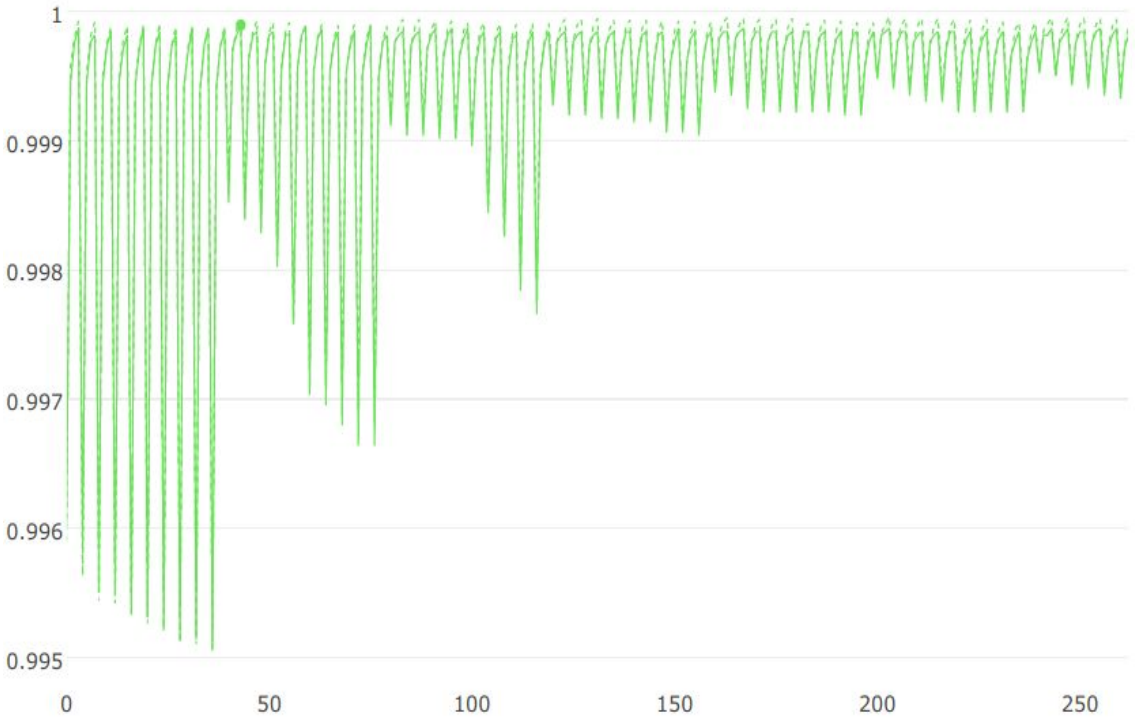


Figura 11: Curva criada a partir do GridSearch feito para diversos parâmetros do modelo.

Observando atentamente para o ponto inserido no gráfico, aproximadamente na quinquagésima época, podemos observar que foi o ponto de maior acurácia com o CatBoost nos dados de validação. Esse processo pode retornar os seguintes parâmetros de configuração a serem utilizados:

- *Learning rate* = 0.1
- *Depth* = 5
- *L2 regularizer* = 1

Avaliação

Dicionário de Classificação de Tipo de Transação Financeira

O problema de classificação do tipo de transação financeira, que pode ser realizada por clientes de instituições financeiras, serão aprovadas ou negadas pelo órgão empresarial financeiro. A instituição pode deparar quatro possibilidades de detecções possíveis de transações são (apresentadas na Tabela 2):

Resultados		**Classificação**	
		Genuína	**Fraude**
Resposta	**Genuína**	Transação genuína Classificação genuína	Transação genuína Classificação fraude

	Fraude	Transação fraude Classificação genuína	Transação fraude Classificação fraude

Tabela 2: Tabela de tipos de classificações possíveis acerca dos tipos de transações realizadas. O objetivo é uma maior acurácia em acertar as classificações em destaque.

1. **Verdadeiro Positivo**: Transação genuína classificada corretamente;
2. **Falso Positivo**: Transação fraudulenta que não deveria ser aprovada, mas é classificada como genuína pelo modelo;
3. **Verdadeiro Negativo**: Transação fraudulenta classificada corretamente.
4. **Falso Negativo**: Transação genuína que deveria ser aprovada, mas é classificada como fraudulenta pelo modelo;

Com os dados de validação, o modelo foi submetido a testes avaliativos, averiguando-se a acurácia do modelo proposto, seguindo o fluxo apresentado na Figura 12, que averiguava a convergência de valores em classificar o tipo de uma transação. O treinamento do modelo de classificação consiste na submissão do conjunto de dados balanceados ao modelo CatBoost com o parâmetros definidos na etapa de *GridSearch*. Este procedimento nos permitiu além de realizar o treinamento do modelo, extrair informações sobre os atributos mais relevantes ao processo de aprendizagem utilizando o conjunto de dados específico.

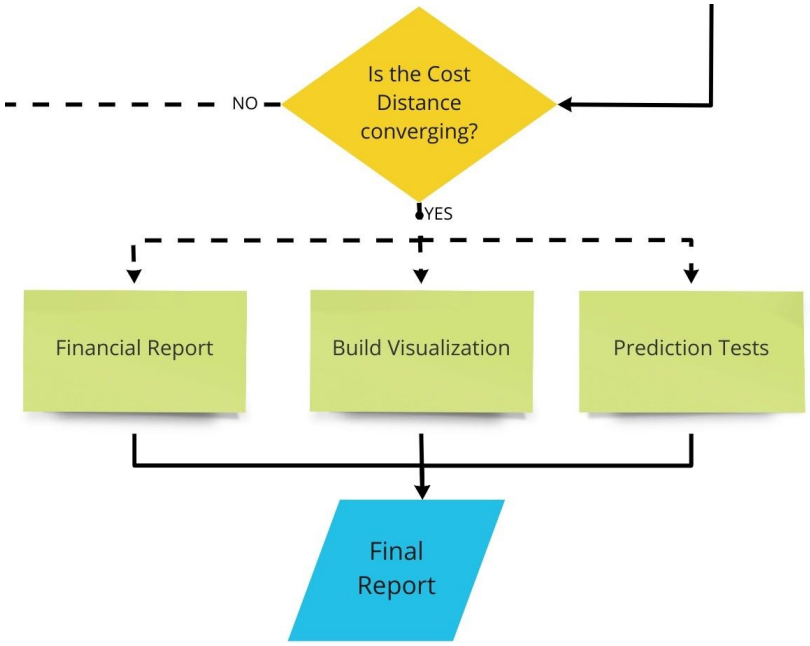


Figura 12: Processo de classificação feito pelo modelo e a saída final do sistema.

Na Figura 13 pode ser observado o gráfico de *Feature Importance* gerado pelo CatBoost no seu processo de aprendizado utilizando a técnica de interpretabilidade SHAP[18]. Observe que os atributos coloridos são aqueles com maior contribuição para a predição,

[18] https://github.com/slundberg/shap (acesso em 08/12/2019).

perceba que diversos atributos gerados no *pipeline* descrito neste documento aparecem entre os mais importantes.

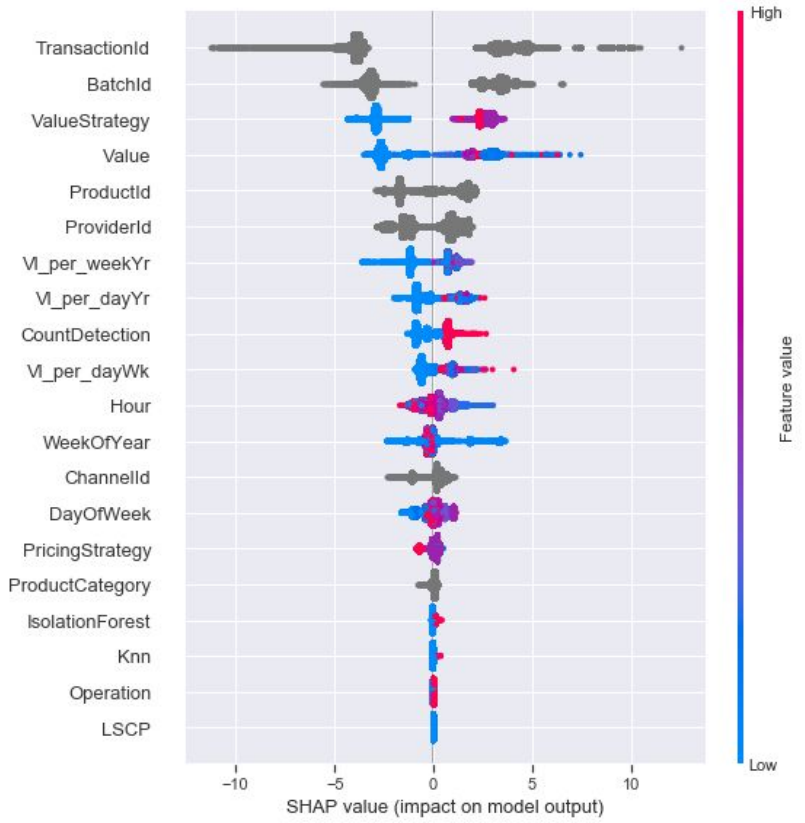


Figura 13: Importância de cada atributo para o modelo.

O modelo atual foi submetido na plataforma do Xente, embora o desafio já tenha se encerrado, ainda aceita novas submissões. Assim, o *score* obtido no modelo descrito neste documento foi de 0.75, conforme exibido na Figura 14. Este resultado posiciona-se entre os Top 70 melhores resultados obtidos[19]. As métricas de avaliação utilizadas foram: F1-score; Precision; e Recall[20].

Figura 14: Submissão e o resultado atingido na plataforma utilizando o modelo proposto.

Ademais, o resultado obtido posiciona-se entre os Top 70 melhores resultados obtidos, e que podem ser acessados pelo link Rank. O *score* adotado pelo idealizadores da competição é com base em três métricas:

- F1 Score
- Precision
- Recall/Sensitivity/True Positive Rate (TPR)

[19] https://zindi.africa/competitions/xente-fraud-detection-challenge/leaderboard (acesso em 08/12/2019).
[20] Uma descrição completa sobre cada métrica citada pode ser encontrada em: https://en.wikipedia.org/wiki/Receiver_operating_characteristic (acesso em 08/12/2019).

Análise Financeira

Quando acontece um Falso Positivo, isto é, uma transação genuína é classificada como fraude, o consumidor não consegue efetuar a compra e portanto, o banco ao negar a transação deixa de receber a margem operacional que é calculada com base no valor da transação que seria realizada. Logo, para calcular esse custo deve-se conhecer o percentual destinado ao banco a cada transação, baseado no valor da compra. No Brasil, esse valor gira em torno de 1% à 5% nas maquinetas[21] [22] de cartão de crédito.

Sendo assim, a equação para calcular o Custo dos Falsos Positivos (CFP) pode ser descrito como:

$$Custo\ dos\ Falsos\ Positivos\ (CFP) = \sum_{i=1}^{N} V_i$$

Em que, *V* indica o valor das transações classificadas como Falsos Positivos, dado que *N* é o número de Falsos Positivos do classificador.

Por outro lado, quando ocorre algum Falso Negativo, isto é, o banco classifica como genuína uma transação fraudulenta, o banco deve arcar com o prejuízo da operação, sendo 100% do valor da transação, logo o Custo dos Falsos Negativos (CFN) pode ser descrito como:

$$Custo\ dos\ Falsos\ Negativos\ (CFN) = alpha \times \sum_{i=1}^{N} V_i$$

Em que, *alpha* é o percentual de lucro do banco em cima de uma transação de débito/crédito, *V* indica o valor das transações que o classificador fez do tipo Falso Negativo. Logo, o objetivo dessa função de otimização é minimizar a equação:

$$Custos\ Financeiro = CFP + CFN$$

Essa ideia pode ser empregada no contexto de detecção de fraudes, visto que, um Falso Positivo irá acarretar em um certo desconforto pelo cliente que, ao passar o cartão terá a sua transação negada, além da rede de crédito perder uma margem do seu lucro (participação na transação). Enquanto um Verdadeiro Negativo condiz em um prejuízo completo, uma vez que uma transação fraudulenta foi aprovada pela rede de crédito e o cliente irá exigir o estorno da fatura.

Transações fraudulentas correspondem a menos que 1% de todo o volume financeiro transacional. Considerando que um falso negativo indica que uma transação fraudulenta aconteceu mas o algoritmo não detectou e portanto, o banco assumirá o prejuízo integralmente, e que um falso positivo indica que uma transação genuína aconteceu mas o algoritmo não detectou e portanto, o banco perderá a participação dele na transação caso ela tivesse sido aprovada. Portanto, caso o algoritmo não existisse, o prejuízo seria de quase 45 BILHÕES de dólares.

---

[21] https://www.visa.com.br/sobre-a-visa/geral/taxas-intercambio.html
[22] https://www.hnb.net/images/bank-downloads/card-center/agreement-english.pdf

Considerando o modelo treinado com o CatBoost, o banco conseguiu economizar o montante de prejuízo estimado de R$ 1.441.922,20. Desta forma, o modelo permitiu evitar mais de 99.9% do prejuízo total aos cofres da empresa.

Conclusão

As principais conclusões dos experimentos desenvolvidos aqui mostraram que:

- Observamos que o descarte de atributos altamente correlacionados permitiu que o modelo continuasse competitivo com os demais competidores.
- Os novos atributos criados ajudaram o modelo a alcançar o desempenho esperado, identificado usando o interpretador de modelos SHAP implementado no CatBoost.
- A inserção da técnica de detecção de fraudes permitiu uma economia bilionária para a rede de cartão de crédito.