Universidade Federal do Paraná
Setor de Ciências Exatas
Departamento de Estatística
Programa de Especialização em *Data Science* e *Big Data*

Adriano de Castro Benatto Paul

Avaliação de Métodos de Machine Learning na Detecção de Fraude em Dados Transacionais de Cartão de Crédito

Curitiba
2020

Adriano de Castro Benatto Paul

Avaliação de Métodos de Machine Learning na Detecção de Fraude em Dados Transacionais de Cartão de Crédito

Monografia apresentada ao Programa de Especialização em Data Science e Big Data da Universidade Federal do Paraná como requisito parcial para a obtenção do grau de especialista.

Orientador: Prof. Marco Antonio Zanata Alves
Co-orientador: Prof. Luiz Eduardo Soares de Oliveira

Curitiba
2020

Especialização em Data Science e Big Data
Universidade Federal do Paraná
dsbd.leg.ufpr.br

Avaliação de Métodos de Machine Learning na Detecção de Fraude em Dados Transacionais de Cartão de Crédito

Adriano de Castro Benatto Paul[1]
Marco Antonio Zanata Alves[3]
Luiz Eduardo Soares de Oliveira[3]

Resumo

Apenas no Brasil, estima-se que os prejuízos causados por fraude em transações eletrônicas com cartão de crédito sejam da ordem de 7 bilhões de reais ao ano. Para evitar este tipo de fraude, são utilizados modelos estatísticos e de aprendizagem de máquina para detectar padrões e negar transações fraudulentas antes que sejam finalizadas, evitando assim os prejuízos decorrentes desta prática. Neste contexto, este trabalho apresenta a aplicação de técnicas para visualizar dados, detectar fraudes e avaliar modelos de aprendizado de máquina em uma base aberta e anonimizada de dados transacionais de cartão de crédito. Para visualização dos dados é utilizada a técnica t-SNE (t-Distributed Stochastic Neighbor Embedding), para detecção dos eventos de fraude são utilizados três métodos distintos: KNN (K-Nearest Neighbors), Random Forest e Gradient Boosting. Como métricas de avaliação destes modelos utilizou-se da precisão, revocação e F-score, além de uma breve análise de outros indicadores relevantes no mercado que também devem ser considerados para a escolha de um modelo de detecção de fraudes. Nossos resultados mostram o modelo de Random Forest com o melhor desempenho entre os métodos avaliados, classificando corretamente 99% das transações fraudulentas, incorrendo em falsos positivos em aproximadamente um terço das classificações de fraude.
Palavras-chave: detecção de fraude, visualização multidimensional, precisão e revocação.

Abstract

Keywords: Fraud detection, multidimensional visualization, precision and recall.*In Brazil alone, it is estimated that the losses caused by fraud in electronic credit card transactions are in the order of 7 billion BRL per year. To prevent this type of fraud, statistical and machine learning models are used to detect patterns and deny fraudulent transactions before they are confirmed, avoiding the resulting losses from this practice. In this context, this work presents the application of techniques to visualize data, detect fraud and evaluate machine learning models on an open and anonymous database of transactional credit card data. For data visualization, the t-SNE (t-Distributed Stochastic Neighbor Embedding) technique is used; to detect fraud events, three different methods are used: KNN (K-Nearest Neighbors), Random Forest and Gradient Boosting. As for the metrics to evaluate these models, precision, recall and F-score were used, in addition to a brief analysis in terms of other relevant market indicators, which should also be considered when choosing an assertive fraud detection model. Our results show the Random Forest model with the best performance among the evaluated methods, correctly classifying 99% of fraudulent transactions, while incurring in false positives in approximately one third of the fraud classifications.*
Keywords: Fraud detection, multidimensional visualization, precision and recall.

1 Introdução

O cartão de crédito é um meio de pagamento emitido para usuários (titulares) que permite o pagamento a um comerciante por bens e serviços com base na promessa de que este titular honrará esta transação (bem como outros encargos acordados) ao banco emissor do cartão [4]. Segundo o Banco Central, ao fim de 2019 existiam no Brasil cerca de 123 milhões de cartões de crédito ativos, um crescimento de 33% em relação ao ano anterior [5]. Segundo este mesmo relatório, 24.3% das transações com cartões de crédito - cerca de 74 bilhões de reais - ocorreram sem a presença física do cartão em 2019. Especialistas de mercado estimam que ocorram fraudes em 2.52% deste montante [6].

Esta modalidade de uso do cartão, sem que seja apresentado fisicamente ao comerciante, cria também uma nova modalidade de fraude - mais especificamente quando detalhes do cartão são obtidos de maneiras escusas e utilizados sem a anuência do titular, como descrito por [7]. Esta situação, por fim, ecoa a simples definição de fraude dada por [1] onde o fraudador busca ilegalmente vantagens indevidas para si, em detrimento do comércio e/ou do titular legítimo daquele cartão. A definição de fraude segundo [1] é a seguinte, em tradução

[1]Adriano de C. B. Paul, adr_paul@outlook.com.
[2]Marco A. Z. Alves, mazalves@inf.ufpr.br.
[3]Luiz E. S. de Oliveira, lesoliveira@inf.ufpr.br.

livre: "o crime de enganar alguém de modo a conseguir dinheiro ou bens ilegalmente".

Como transações não presenciais ocorrem principalmente por meio eletrônico com a digitação dos dados por parte do usuário, os mecanismos mais tradicionais de segurança – como a verificação de outros documentos físicos do titular, além da presença do titular em si – não ocorrem. Sendo assim, novos métodos para validação de transações fizeram-se necessários para garantir a segurança desta modalidade de transação, dos quais destacam-se modelos estatísticos e de aprendizado de máquina para a correta identificação e classificação das transações.

Estes modelos são alimentados pelos dados inseridos pelo usuário (como nome completo, endereço, números de documentos); pelas informações intrínsecas a uma transação (tais como valor, data, quantidade e descrição do produtos) e pelos outros dados coletados do usuário no momento da transação (metadados), como por exemplo o modelo e o sistema operacional do aparelho onde foi realizada a transação, endereço de I.P. (*Internet Protocol*), tempo utilizado para inserção das informações, entre inúmeros outros.

Todas estas informações devem ser trabalhadas de forma que possam alimentar este algoritmo, cuja resposta seja uma recomendação de aprovação ou não desta transação, de tal maneira que esta decisão seja favorável à segurança do usuário, do comerciante e do banco emissor do cartão de crédito utilizado, evitando prejuízos futuros a quaisquer uma das partes envolvidas. Portanto, a detecção de padrões distintos aliados à possível similaridade entre transações fraudulentas – bem como sua dissimilaridade entre transações honestas – são os guias para um bom modelo anti-fraude.

Neste trabalho, propomos a utilização de uma técnica para visualização dos dados (t-SNE) e três técnicas para classificação das transações: KNN, *Random Forest* e *Gradient Boosting*.

Uma base de dados pública será adotada (disponível em [8]) – coletada, analisada e anonimizada pela ULB (*Université Libre de Bruxelles*) – onde constam em torno de 284 mil transações com cartão de crédito de usuários europeus em setembro de 2013. Nesta base, todas as variáveis relevantes em uma transação estão anonimizadas por motivos de confidencialidade.

O treinamento dos modelos e seus respectivos testes para detecção de fraude serão realizados na mesma base de dados, devidamente separada em uma base de treinamento e uma base de testes. Paralelamente, serão treinados e testados os mesmos modelos em uma base mais enxuta de atributos, somente com aqueles atributos de maior correlação com a marcação de fraude.

Logo, os resultados obtidos poderão ser comparados entre si para um mesmo modelo em diferentes formatos de base, bem como seu desempenho entre os outros modelos para o mesmo formato de base, utilizando a precisão, revocação e *F-score* como métricas de avaliação de performance.

2 Algoritmos de ML e Métricas

Nesta sessão apresentaremos brevemente cada uma das técnicas utilizadas para visualizar os dados e classificar as transações. A implementação de cada uma delas é feita a partir do pacote *sklearn* para linguagem *Python*, disponível gratuitamente [9].

2.1 t-SNE (*t-Distributed Stochastic Neighbor Embedding*)

t-SNE é um método não-linear de redução de dimensionalidade para visualização em espaços dimensionais de duas ou três dimensões [10]. O algoritmo t-SNE é composto por dois estágios principais: o primeiro constrói uma distribuição de probabilidade sobre pares de objetos de tal maneira que objetos similares recebem uma alta probabilidade. A seguir, o algoritmo define uma distribuição de probabilidades similar sobre os pontos em um mapa de baixa dimensão, minimizando a divergência de Kullback-Leibler entre duas distribuições com relação à localização dos pontos no mapa. A parametrização do algoritmo t-SNE para criação de um bom mapa depende, principalmente, do valor utilizado para perplexidade e do número de iterações para otimização da distribuição de cada ponto no mapa.

2.2 KNN (*K-Nearest Neighbors*)

KNN é um método não-paramétrico utilizado para classificação supervisionada e regressão cuja entrada, em ambos os casos, consiste nos k exemplos de treinamento mais próximos no espaço dos elementos. Este método estima o valor da função densidade de probabilidade ou, diretamente, a probabilidade a posteriori de que um elemento x pertencer a uma classe C a partir da informação proporcionada pelo conjunto dos dados de treinamento [11]. Ou seja, este método supõe que a classificação dos vizinhos mais próximos de um ponto no espaço dos elementos melhor determinam a classificação do elemento em questão.

2.3 *Random Forest*

Random Forest é um método conjunto de aprendizado utilizado para classificação e regressão criado por [12] e melhor desenvolvido por [13] para melhor seleção aleatória de atributos, combinando otimização aleatória de nós e *bagging* (*bootstrap aggregating*).

Uma relação entre *random forests* e KNN foi apontada por [14], onde ambos podem ser vistos como esquemas de vizinhanças ponderadas, onde a forma da vizinhança em uma **random forest** adapta-se à importância local de cada atributo.

2.4 *Gradient Boosting*

É uma técnica utilizada para problemas de classificação e regressão que consiste na produção de um modelo predi-

tivo baseado num conjunto de modelos preditivos fracos, tipicamente árvores de decisão. Esta ideia originou-se da observação que o *boosting* pode ser interpretado como um algoritmo de otimização em uma função custo adequada [15].

A intuição por trás do algoritmo é de fortalecer repetidamente os padrões nos resíduos, chegando ao ponto em que os resíduos não possuam nenhum padrão que possa ser modelado, ou seja, minimizando a função de perda até que alcance seu mínimo.

2.5 Métricas

Como métricas de avaliação e comparação entre modelos, optou-se por utilizar:

2.5.1 Precisão

Precisão ou Confiança denota a proporção de casos positivos daqueles preditos como positivos [16]:

$$\text{Precisão} = \frac{\sum \text{Instâncias positivas}}{\sum \text{Instâncias previstas positivas}} \quad (1)$$

Tratando-se de fraude, pode-se dizer que a precisão do modelo seria a razão das transações fraudulentas em relação ao total de transações que o modelo previu como fraudulentas. Portanto, a melhor precisão de um modelo é aquele sem falsos positivos (casos erroneamente classificados como fraude), ou seja, todas as transações apontadas como fraude são realmente fraudulentas (e portanto, cuja razão é igual a 1).

2.5.2 Revocação

Revocação ou Sensibilidade é a proporção de casos previstos como positivos do total de casos positivos [16]:

$$\text{Revocação} = \frac{\sum \text{Instâncias previstas positivas}}{\sum \text{Total de instâncias positivas}} \quad (2)$$

Para detecção de fraude, a revocação do modelo seria a razão de transações que o modelo classificou como fraudulentas em relação ao total de transações fraudulentas. Portanto, a melhor revocação de um modelo é aquele sem falsos negativos (fraudes não-detectadas), ou seja, que todas as transações fraudulentas foram corretamente classificadas pelo modelo como fraude (e portanto, cuja razão é igual a 1).

2.6 F-score

F-score ou *F-measure* é uma medida da acurácia de um teste, calculado como a média harmônica da precisão e revocação obtidas [16]:

$$\text{F-score} = 2 * \frac{\text{Precisão} * \text{Revocação}}{\text{Precisão} + \text{Revocação}} \quad (3)$$

Na prática, a importância relativa da precisão e revocação devem ser considerados no problema, mesmo que modelos distintos apresentem o mesmo F-score [17].

3 Proposta e Metodologia

Utilizando uma base de dados pública – disponível em [8]) – todas as variáveis relevantes em uma transação estão anonimizadas a partir de Análises de Componentes Principais (PCA). Por razões de confidencialidade, não foram especificados quais são os campos correspondentes a estes dados nem os passos utilizados para anonimizá-los. Os únicos atributos definidos na base são *Time* (intervalo de tempo em segundos entre uma transação da base com a primeira transação listada) e *Amount* (valor da transação). A marcação de fraude é dada pelo campo *Class*, presente em apenas 492 transações (aproximadamente 0.7% do total de transações). Os demais atributos, encontram-se numerados de 01 a 28.

A partir desta base de dados, será buscada uma maneira de mapear e visualizar estas transações em razão de sua similaridade uma com as outras. O intuito desta visualização é verificar de forma rápida a distribuição das transações com marcação de fraude entre si. Caso existam características semelhantes entre elas, espera-se que *clusters* de transações fraudulentas fiquem evidentes, apesar do desbalanceamento da base entre transações com e sem marcação de fraude. O método utilizado para esta visualização será o t-SNE, implementado no pacote *sklearn* para linguagem *Python*.

Com os dados visualizados, procuraremos reduzir a dimensionalidade desta base (que possui 30 atributos distintos) para que sejam usados, paralelamente à base original, uma base reduzida que contenha somente os atributos mais relevantes em relação à marcação de fraude. Para tanto, serão criadas vinte amostras balanceadas com igual número de transações com e sem marcação de fraude, e então calculada a correlação entre cada um dos atributos da amostra com o evento de fraude. Serão descartados todos os atributos cujo valor médio de correlação com fraude esteja entre -0.5 e 0.5.

Então, com duas bases distintas, serão separadas cada uma delas em uma base de treino (composta por 70% das transações, ordenadas de acordo com sua ocorrência) e uma base de testes, com os 30% das transações restantes. Estas novas bases serão utilizadas para o treinamento dos modelos e classificação das transações. No método KNN em particular, a base de treino é balanceada tal como realizado para o cálculo da correlação dos atributos: as transações sem marcação de fraude são escolhidas aleatoriamente, em igual quantidade àquelas marcadas como fraude e a classificação posterior na base de testes é feita para todas as transações.

Para o método KNN, serão treinados modelos distintos usando 1, 3, 5, 7 e 9 vizinhos para classificar as transações na base de testes, enquanto *Random Forest* e *Gradient Boosting* serão implementados diretamente tal como estão no pacote *sklearn*, sem qualquer alteração de hiper-parâmetros.

Como ferramentas de apoio para manipulação dos dados, serão utilizados os pacotes *pandas* e *numpy*, também disponibilizados gratuitamente. Para criação dos gráficos e visualização dos dados, será utilizado o pacote

seaborn, também para a linguagem *Python*.

4 Resultados

4.1 Visualização dos dados

A partir da aplicação do t-SNE em uma amostra balanceada dos dados, com igual número de transações com e sem marcação de fraude e variando apenas os valores de perplexidade do t-SNE, obtém-se os mapas presente nas Figuras 1, 2 e 3.

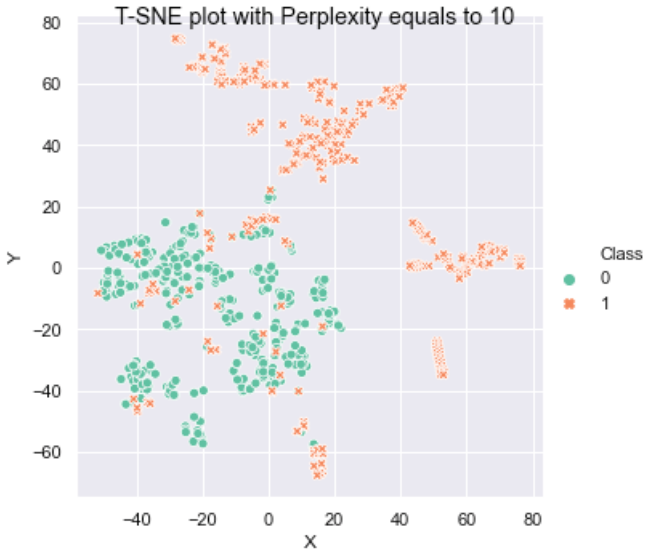

Figura 1: Mapa criado com t-sne em base balanceada com valor de perplexidade igual a 10.

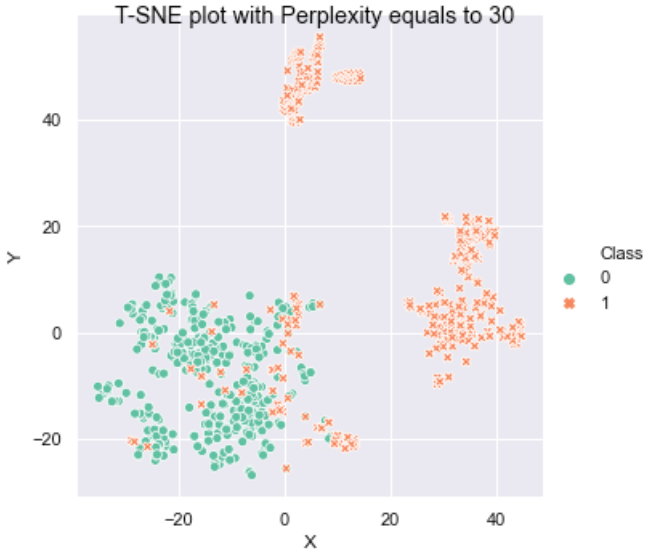

Figura 2: Mapa criado com t-sne em base balanceada com valor de perplexidade igual a 30.

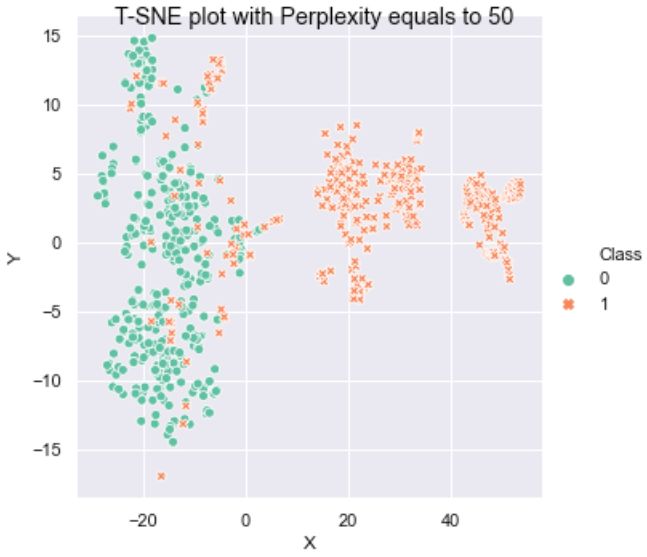

Figura 3: Mapa criado com t-sne em base balanceada com valor de perplexidade igual a 50.

Visualmente, a Figura 1, com valor de perplexidade igual a 10, apresenta dados dispersos, com pouca definição entre os *clusters*. A Figura 3, com valor de perplexidade igual a 50, apresenta *clusters* bem definidos, porém muito próximos uns dos outros. Enfim, a Figura 2, com valor de perplexidade igual a 30, parece agrupar melhor os dados, mantendo uma distância apropriada entre *clusters* distintos. Este valor de perplexidade será então utilizado na visualização de todos os dados da base.

Na Figura 4 é mostrado o mapa de calor da correlação entre cada atributo da base de treino com a marcação de fraude em vinte amostras balanceadas, já ordenadas em ordem crescente do valor médio de correlação. Com o exame destes atributos, pode-se separar a base em duas: uma completa, com todos os atributos; e outra apenas com os atributos cuja correlação média para as amostras balanceadas é maior que 0.5 ou menor que -0.5 (condição atingida por apenas 13 atributos dos 30 originais). Este valor de correlação foi escolhido arbitrariamente, de modo a testar a precisão e revocação dos modelos sem a necessidade de se utilizar todos os atributos da base (e como consequência, acelerar o processo de treinamento dos modelos e da previsão de fraude em si). Para esta base mais enxuta, são então selecionados somente os atributos 2, 3, 4, 6, 7, 9, 10, 11, 12, 14, 16, 17 e 18.

A visualização completa com todos os dados da base é mostrado na Figura 5. De acordo com o que foi visto nas primeiras figuras, um valor de perplexidade igual a 30 parece ser o mais indicado para a visualização destes dados, sendo então utilizado para a criação deste mapa.

Apesar do algoritmo t-SNE ser eficiente para criação de mapas e visualização de dados multidimensionais, este é um método computacionalmente caro pois a similaridade é calculada entre cada ponto em relação a todos os outros pontos do mapa (neste caso, entre to-

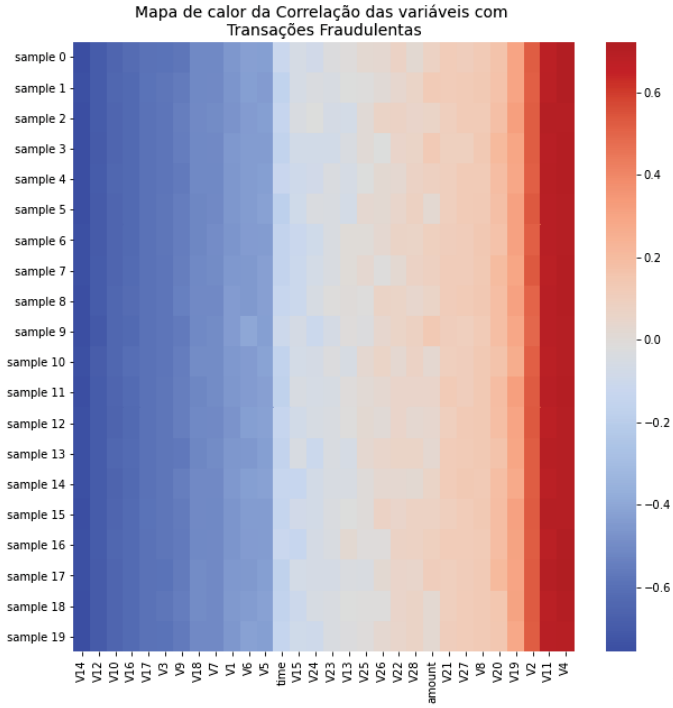

Figura 4: Mapa de calor ordenado da correlação dos atributos da base de treino com a marcação de fraude em 20 amostras balanceadas.

dos os atributos de uma transação em relação a todos os atributos das demais), para então buscar a melhor disposição entre os pontos em uma dimensão mais baixa. Para a base original, com todas as transações com trinta atributos distintos cada, foram calculados, aproximadamente, 40 bilhões de valores de similaridade entre pontos para que finalmente fossem dispostos e iterados sucessivamente em duas dimensões para sua melhor representação. Nessa representação bidimensional ficaram evidentes agrupamentos formados somente por transações fraudulentas e outros *clusters* sem distinção clara entre transações com e sem marcação de fraude.

4.1.1 Treinamento e Predição de Fraude

Na Tabela 1 temos os resultados para o modelo KNN para a base completa. Nesta tabela, temos os resultados de precisão, revocação e *F-score* utilizando como parâmetros os ímpares de um a nove vizinhos próximos. Nota-se que independente do número de vizinhos, o valor de precisão permanece praticamente inalterado, porém sempre com baixo valor de revocação. Isto significa que o modelo não classificou as transações como fraudulentas com frequência, porém, quando as classificou como tal, acertou a classificação de fraude em torno de 86% das vezes. Note que apesar da base de testes ser composta por 30% das transações finais da base de dados (contendo aproximadamente 85 mil transações), apenas 108 (0.12%) apresentam marcação de fraude. Logo, um modelo com esta precisão identifica corretamente 93 transações marcadas como fraude das 108 totais.

Tabela 1: Resultados para o modelo KNN na base de dados com todos os atributos

Nr. vizinhos	Precisão	Revocação	F-score
1	0.87	0.03	0.05
3	0.87	0.05	0.09
5	0.86	0.06	0.10
7	0.86	0.06	0.12
9	0.86	0.07	0.14

Na Tabela 2 temos o mesmo modelo KNN, porém treinado para a base mais enxuta de atributos. Nota-se um aumento sensível, em média, para as três métricas. Entretanto, o modelo continua com baixo valor de revocação e com precisão semelhante ao observado na Tabela 1. Logo, para o formato atual da base e com os atributos utilizados, não recomenda-se a utilização de modelo baseado unicamente em KNN.

Tabela 2: Resultados para o modelo KNN na base de dados com atributos selecionados

Nr. vizinhos	Precisão	Revocação	F-score
1	0.90	0.02	0.04
3	0.87	0.05	0.09
5	0.87	0.07	0.13
7	0.87	0.09	0.09
9	0.86	0.10	0.10

Na Tabela 3 temos todas as métricas para ambas as bases utilizadas no modelo *Random Forest*. Neste modelo, vê-se valores de revocação próximos a 100%. Na prática, isto significa que o modelo classificou corretamente praticamente todas as transações fraudulentas.

Tabela 3: Resultados para o modelo *Random Forest*

Base	Precisão	Revocação	F-score
Completa	0.68	0.98	0.81
Atributos selec.	0.72	0.99	0.83

Tabela 4: Resultados para o modelo *Gradient Boosting*

Base	Precisão	Revocação	F-score
Completa	0.41	0.70	0.52
Atributos selec.	0.58	0.77	0.66

E por fim, na Tabela 4, temos os resultados obtidos com o modelo *Gradient Boosting*. Este modelo apresenta revocação mais baixa que o modelo *Random Forest* e a pior precisão dos três modelos. Isto porém não significa que um modelo baseado nesta técnica seja ruim para classificação de fraude, apenas que exige um trabalho adicional para correta configuração de seus parâmetros de treinamento. Da maneira que está implementado no pacote, este método possui mais de 20 parâmetros inciais que podem ser configurados, e sua correta parametrização pode render excelentes resultados. Entretanto,

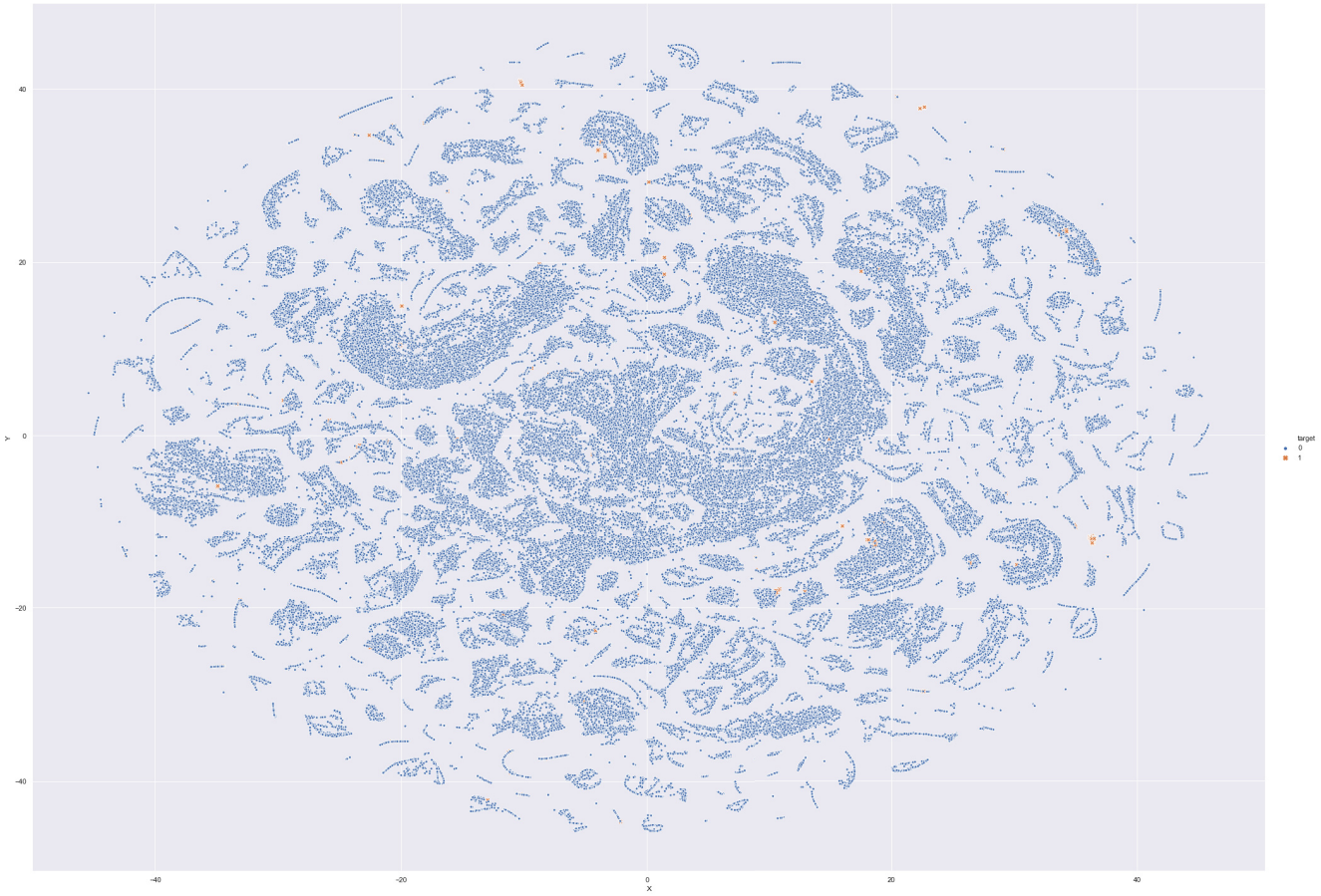

Figura 5: Visualização de toda a base de dados a partir de t-sne com valor de perplexidade igual a 30.

esta parametrização está fora da proposta inicial deste trabalho.

A análise da correlação dos atributos com a marcação de fraude mostrou-se bastante útil para o treinamento dos modelos. Além de reduzir a quantidade de dados em aproximadamente 60% (reduzindo 30 dimensões para 13), nota-se que nos três modelos utilizados obteve-se um aumento na precisão e revocação, principalmente no modelo de *Gradient Boosting*. Outras técnicas para escolha dos atributos podem ser utilizadas, expandindo esta análise em trabalhos futuros.

Com os resultados obtidos, independente do número de vizinhos utilizado, o modelo KNN apresenta os maiores valores de precisão entre os três modelos. Porém, esta precisão acontece ao custo de uma revocação baixíssima (deixando de classificar muitas transações como fraudulentas). Consequentemente, o *F-score* destes modelos também é muito baixo.

O modelo *Random Forest* por sua vez apresenta revocações próximas do valor máximo, ou seja, o modelo previu corretamente quase a totalidade de transações fraudulentas, mesmo que em algumas ocasiões tenha classificado transações honestas erroneamente (em torno de 30% das vezes). Entretando, como as transações fraudulentas correspondem a apenas 0.05% da base de testes, estes falsos positivos correspondem a apenas 0.2% do total de transações nesta base. Na prática, isto ocorre em situações de compra atípicas, como por exemplo, quando um cliente está realizando sua primeira compra (e portanto, sem qualquer histórico pregresso) com um valor muito acima do valor médio das transações usuais dos demais clientes deste estabelecimento. Normalmente desconfia-se deste tipo de comportamento e, por precaução, é até preferível que transações como esta sejam de fato negadas.

Os excelentes resultados obtidos com o método de *random forest* para a base mais enxuta de atributos levantam, a partir de agora, outras perguntas cujas respostas são necessárias para a criação de modelos mais assertivos. Estas perguntas estão diretamente relacionadas à maneira com que esta base de dados pública foi criada – pois existem diversos fatores externos que afetam diretamente tanto a qualidade dos dados em si – para que um modelo estatístico criado a partir destes dados históricos funcione na classificação de novas transações, recebendo-as em tempo real. Estas perguntas incluem, mas não se limitam a:

- Quais são as dificuldades impostas ao usuário para cancelar/reportar transações quando ocorre algum problema?
- É mais fácil/rápido/conveniente reportar ao banco casos de desacordo comercial como fraude para rea-

ver o valor de uma transação ou diretamente com o local onde foi realizada a transação?

- Quantos dias/meses o usuário possui para reportar uma situação de fraude em seu nome?
- A extração da base de dados foi realizada após este período máximo de reporte de fraude (garantindo então que não existem transações fraudulentas que ainda não receberam a devida classificação, ou seja, estão devidamente maturados)?
- Todas as variáveis anonimizadas dispostas na base de dados estão à disposição no momento da transação para que sejam utilizadas por um modelo semelhante ao apresentado, para uma eventual tomada de decisão para aprovar ou negar a transação em tempo real?
- Mesmo que existam fatores humanos alheios que impeçam a correta classificação dos dados, estes fatores poderiam também ser modelados e previstos estatisticamente (tais como situações de auto-fraude ou perda do prazo de reporte de fraude)?

A escolha de um modelo estatístico para detecção de fraude também está intimamente ligada ao modelo de negócio onde será aplicado, e não necessariamente a um alto valor de precisão e/ou revocação. Em transações financeiras, frequentemente, o melhor modelo é aquele que detecta e evita o maior montante em fraudes, e não necessariamente a maior quantidade de transações fraudulentas.

Outro ponto a ser considerado é o custo de uma transação fraudulenta: supondo que uma empresa opere com uma margem de lucro de 10% sobre suas vendas, para cada ocorrência de fraude esta empresa precisará realizar outras 10 vendas honestas apenas para cobrir o prejuízo de uma fraude. Portanto, para ela o melhor modelo a ser utilizado é aquele com o maior valor de revocação possível, evitando ao máximo a ocorrência de uma fraude. Em compensação, para uma outra empresa que, por exemplo, trabalhe com serviços *online*, tais como jogos, *streaming* de áudio/vídeo, acesso remoto a mídias de entretenimento, etc, a escolha de um modelo para detecção de fraudes fica mais voltada à precisão deste, pois evita o bloqueio de transações honestas, e consequentemente, permite o crescimento da base de usuários desta plataforma. Neste caso, dado o modelo de negócio desta empresa, sua tolerância à ocorrência de fraudes é maior em relação à primeira. Ou seja, mesmo que existam dois modelos com exatamente o mesmo *F-score*, as diferenças na precisão e revocação devem ser avaliadas para que o modelo seja consonante ao modelo de negócio onde será utilizado, tal como mencionado por [17].

5 Conclusões e Trabalhos Futuros

Neste trabalho vimos a aplicação de técnicas de classificação para detecção de fraude em uma base de dados com apenas 2 dias de transações de usuários. A aplicação de um modelo qualquer para detecção de fraudes envolve não apenas o trabalho analítico de um (ou vários) cientista(s) de dados, mas de toda uma infra-estrutura criada por diferentes áreas técnicas dedicadas à esta finalidade.

Na base de dados utilizada, comodamente, os atributos já encontram-se tratados e transformados numericamente. Na prática, todos os dados e metadados que compõem uma transação devem ser trabalhados em tempo real, seja em transformações numéricas ou buscando e agregando outras informações a partir destas, em outros bancos de dados. Toda a informação deve fluir do usuário ao comerciante, que redireciona estes dados ao banco, ao sistema de anti-fraude e a todos os *players* envolvidos nesta operação. A confirmação ou não da transação deve fazer todo o caminho de volta ao usuário em tempo hábil de forma que não prejudique a experiência dele neste processo – que pode não mais utilizar o meio eletrônico para suas compras, caso este processo não flua a contento.

Assim sendo, outros tópicos que podem ser estudados e que fazem parte de uma transação em meio eletrônico são: a eficiência e integridade em diferentes *pipelines* de dados; tempos de reposta entre diferentes modelos; criação de modelos a partir de outras técnicas estatísticas e/ou de aprendizagem de máquina; precisão e revocação da agregação de diferentes modelos para a tomada de decisão; estudos mais aprofundados para a eficiente parametrização de cada um dos modelos. Também, se possível, a utilização de uma base de dados conhecida e bem consolidada, evitando assim quaisquer divagações sobre os atributos que a compõem ou a qualidade dos dados em si.

6 Agradecimentos

Agradecemos a todos os colegas e professores do Programa de Especialização em Data Science & Big Data da Universidade Federal do Paraná, em especial aos coordenadores deste excelente curso, Wagner Bonat e Walmes Zeviani.

Referências

[1] https://www.oxfordlearnersdictionaries.com/us/definition/english/fraud *Oxford Advanced Learner's Dictionary*. Oxford University Press, acesso em 10/06/2020.

[2] Van Vlasselaer, Véronique and Eliassi-Rad, et al., *Gotcha! Network-based fraud detection for social security fraud*. Journal of Management Science, INFORMS (2017).

[3] Baesens, B., Van Vlasselaer, V., & Verbeke, W. *Fraud analytics using descriptive, predictive, and social network techniques: a guide to data science for fraud detection*. John Wiley & Sons (2015).

[4] Arthur, Sullivan; Sheffrin, Steven M. *Economics:*

Principles in action. Upper Saddle River, New Jersey, v. 7458, p. 173 (2003).

[5] https://www.bcb.gov.br/estatisticas/spbadendos. Banco Central do Brasil, *Estatísticas de Pagamentos de Varejo e de Cartões no Brasil - 2019*. Acesso em 11/09/2020.

[6] https://blog.konduto.com/pt/2020/06/censo-da-fraude-2020/. Konduto, *Censo da Fraude 2020*. Acesso em 13/09/2020.

[7] Bolton, Richard J and Hand, David J, *Statistical fraud detection: A review*. Journal of Statistical Science, JSTOR (2002)

[8] https://www.kaggle.com/mlg-ulb/creditcardfraud, Machine Learning Group - ULB, versão 3, acesso em 02/06/2020.

[9] https://scikit-learn.org/stable/index.html, Scikit Learn, acesso em 02/06/2020.

[10] Maaten, L. V. D., & Hinton, G. *Visualizing data using t-SNE*. Journal of Machine Learning Research, no. 9 (2008): 2579-2605.

[11] Altman, Naomi S. *An introduction to kernel and nearest-neighbor nonparametric regression*. The American Statistician 46, no. 3 (1992): 175-185.

[12] Ho, Tin Kam. *Random decision forests*. In Proceedings of 3rd international conference on document analysis and recognition, vol. 1, IEEE (1995): 278-282.

[13] Breiman, Leo. *Random forests*. Machine learning 45, no. 1 (2001): 5-32.

[14] Lin, Y., and Y. Jeon. *Random forests and adaptive nearest neighbors (Technical Report No. 1055)*. University of Wisconsin (2002).

[15] Brownlee, Jason. *XGBoost With Python: Gradient Boosted Trees with XGBoost and scikit-learn*. Machine Learning Mastery, 2016.

[16] Powers, David Martin. *Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation*. (2011).

[17] Hand, David, and Peter Christen. *A note on using the F-measure for evaluating record linkage algorithms*. Statistics and Computing 28, no. 3 (2018): 539-547.