Nanodegree Engenheiro de Machine Learning

Detectar transações fraudulentas de cartão de crédito

Talita Shiguemoto
30 de outubro de 2018

Proposta

Histórico do assunto

Segundo a *Konduto*[1], a cada cinco segundos uma tentativa de fraude por cartão de crédito clonado ocorre em *e-commerces* brasileiros. De acordo mesma empresa, no ano passado o Brasil sofreu cerca de 6 milhões de compras fraudulentas. Comparado a outros, um estudo realizado em 2016 pela *Business Insider Intelligence*[2] mostrou o Brasil em segundo lugar no ranking de maior porcentagem de consumidores expostos a transações fraudulentas via cartão de crédito nos últimos cinco anos.

Este estudo iniciou-se para entender como combater este cenário nacional, tendo como objetivo criar um algoritmo de detecção das transações com fraudes baseadas no conjunto de dados retirado do *Kaggle*[3].

Descrição do problema

Muitas transações por cartão de crédito são fraudulentas, o objetivo deste projeto é detectá-las baseado em um conjunto de dados histórico que já possui tal classificação. Cada transação possui algumas variáveis e com base nelas poderemos identificar quais atributos possuem maior influência para desvendar quais são fraudes ou não. O algoritmo criado será baseado em aprendizagem supervisionada e poderá ser utilizado para futuras detecções de fraudes.

[1] https://www.konduto.com/
[2] https://www.businessinsider.com/the-us-has-the-third-highest-card-fraud-rate-in-the-world-2016-7
[3] https://www.kaggle.com/mlg-ulb/creditcardfraud/home

Conjuntos de dados e entradas

O conjunto de dados é referente a transações de usuários de cartões europeus em setembro de 2013, obtidos no *Kaggle*. Possuindo 284.807 linhas e 31 *features*, das quais 28 delas são variáveis dependentes que devido a confidencialidade são resultados de transformações de PCA. Os dados estão desbalanceados, sendo as transações fraudulentas 0,172% dos dados.

A tabela abaixo demonstra a descrição de cada *feature*.

Nome	Descrição	Tipo	Valores
Time	Segundos transcorridos entre cada transação	Quantitativo	-
V1-V28	Variáveis dependentes anônimas devido confidencialidade	Quantitativo	-
Amount	Montante da Transação	Quantitativo	-
Class	Variável resposta. Se a transação foi fraudulenta	Qualitativo	0 (falso), 1 (verdadeiro)

Descrição da solução

Será necessário pré-processar os dados para lidar com o desbalanceamento, utilizado a técnica de subamostragem[4], feito isso, a solução proposta para este problema será baseada em algoritmos de aprendizagem supervisionada de classificação. Será testado os modelos de Métodos de Ensemble (AdaBoost, Random Forest, Gradient Boosting, XGBoost), Regressão Logística, Naive Bayes, Support Vector Machines, K-Nearest Neighbors e Árvores de decisão, verificando qual deles terá melhor performance e resultados. O modelo final será escolhido embasado por algumas métricas (veja em **Métricas de avaliação**) e terá seus hiperparâmetros otimizados pelo *GridSearch*[5].

Modelo de referência (benchmark)

Utilizando o mesmo conjunto de dados no estudo[6] mais votado no *Kaggle,* o autor utilizou uma Regressão Logística para criar o modelo que obteve uma acurácia e recall próximo de 93% e a Area Under the ROC Curve (AUC) em aproximadamente 0,95.

[4] http://www.teses.usp.br/teses/disponiveis/55/55134/tde-06012016-145045/publico/VictorHugoBarella_dissertacao_revisada.pdf
[5] http://scikit-learn.org/stable/modules/grid_search.html
[6] https://www.kaggle.com/joparga3/in-depth-skewed-data-classif-93-recall-acc-now

Métricas de avaliação

Devido nosso conjunto ser desbalanceado a acurácia e o *Confusion Matrix* não são as melhores métricas para serem usadas na avaliação do modelo. Para este estudo acredito que o *recall, precision* e *AUC* serão as melhores métricas para verificar a qualidade do algoritmo.

O *precision*[7] é a proporção entre os Verdadeiros Positivos sobre todas os positivos, sendo eles classificados corretamente ou não. Matematicamente representado por:

$$Precision = \frac{TP}{TP + FP}$$

O *recall,* também chamado de ***True Positive Rate (TPR)***, é a proporção entre os Verdadeiros Positivos sobre todas os positivos classificados corretamente e os falsos negativos. Matematicamente representado por:

$$Recall = \frac{TP}{TP + FN}$$

Para entender o AUC é necessário compreender primeiro a curva do ***Receiver Operating Characteristic (ROC)***[8], que é um gráfico que mostra o desempenho de um modelo de classificação em todos os limites de classificação. Esta curva mostra dois parâmetros: o TPR e o False Positive Rate (FPR).

$$FPR = \frac{FP}{TP + TN}$$

Uma curva ROC traça TPR vs. FPR em diferentes limiares de classificação. Diminuir o limiar de classificação classifica mais itens como positivos, aumentando assim tanto os falsos positivos quanto os verdadeiros positivos. A curva AUC mete toda a área bidimensional por baixo de toda curva ROC.

AUC fornece uma medida agregada de desempenho em todos os possíveis limites de classificação. Uma maneira de interpretar AUC é como a probabilidade de o modelo classificar um exemplo positivo aleatório mais do que um exemplo negativo aleatório. Um modelo cujas previsões são 100% erradas tem uma AUC de 0,0; enquanto um cujas previsões são 100% corretas tem uma AUC de 1,0.

[7] https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall
[8] https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

Design do projeto

O fluxo deste projeto seguirá a seguinte ordem:

- Exploração dos dados
    - Carregar bibliotecas e dados
    - Verificar as distribuições
    - Sumários estatísticos
    - Visualização de dados
- Pré-Processamento
    - Normalização e transformação dos dados
    - Limpeza de dados
    - Métodos de subamostragem (*undersampling*)
    - Separar os dados em treinamento e teste
- Implementação do Modelo
    - Construir os modelos
    - Avaliar performance dos modelos e selecionar o melhor
    - Otimizar modelo final
    - Selecionar *features* com maior importância
    - Avaliar performance do modelo final
- Interpretação dos resultados e conclusão