SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS

TREINANDO CLASSIFICADORES PARA PREVISÃO DE TRANSAÇÕES FRAUDULENTAS NO CARTÃO DE CRÉDITO

Autores: Andrei Magalhães
Gabriel Negreiros
Pedro Elias
Vitor Gabriel

Professor Orientador: Rogério Martins Gomes

Resumo

É importante que as companhias de crédito sejam capazes de detectar transações de crédito fraudulentas para evitar que clientes sejam cobrados por compras não efetuadas. O intuito deste artigo será treinar e comparar classificadores para previsão de transações fraudulentas.

Palavras-Chave: detecção de fraudes, cartões de crédito, dados desbalanceados, regressão logística, kneighbors classifier, support vector classifier, decision tree classifier.

1 INTRODUÇÃO

Na atualidade as transações financeiras digitais já são protagonistas e estão prontas para substituir quase que por completo o dinheiro físico. Nesse contexto, o cartão de crédito é uma ferramenta importante para a transação concretizar e um alvo em potencial para fraudes. É importante que as empresas de cartão de crédito sejam capazes de reconhecer transações fraudulentas para que os clientes não sejam cobrados por itens que não compraram.
A proposta deste artigo é treinar e comparar diferentes modelos para detecção de fraudes com cartão de crédito, baseados em classificadores de IC aplicados em uma base de dados altamente desbalanceada.

1.1 O conjunto de dados

O conjunto de dados trabalhado contém transações de cartões titulares europeus que ocorreram em dois dias de Setembro de 2013 com um total de 284.807 transações das quais 492 foram classificadas como fraudes.
Com o intuito de garantir a confidencialidade de dados sensíveis e reduzir a complexidade do problema, algumas informações originais foram omitidas nos dados durante a construção do conjunto fazendo com que existam diversas variáveis numéricas de entrada resultantes de um trabalho de redução de dimensionalidade através do algoritmo *Principal Component Analysis* (PCA).
Através do PCA foram obtidos os principais componentes, 28 features (V1, V2,...V28) que compõem a base de dados com as características ‘Tempo’, ‘Valor’ (conservadas do processo de redução de dimensionalidade) e ‘Class’ que determina se uma transação foi classificada como fraudulenta, assumindo valor 1, ou não, assumindo valor 0. A feature 'Tempo' contém os segundos decorridos entre cada transação e a primeira transação no conjunto de dados. A feature 'Valor' é o valor da transação.

2 METODOLOGIA

2.1 Análise do DataSet

Inicialmente foi feita a análise dos dados para identificar as características, os tipos dos valores de cada característica, ou feature, e a distribuição das classes. Através da análise foi possível constatar que os dados estão extremamente desequilibrados, são 492 casos positivos e 284315 casos negativos como pode ser visto na imagem abaixo:

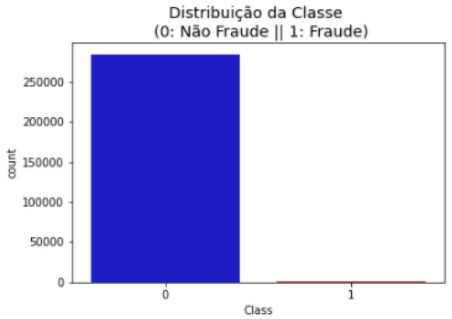


Imagem 1 - Distribuição das Classes

Como grande parte dos atributos não possui características tangíveis para uma primeira análise, o foco inicial recaiu sobre os atributos Amount e Time. Abaixo encontram-se dois gráficos com a distribuição da densidade por valor da transação (Amount) (Imagem 2) e da densidade pelo tempo (Time) em que a transação ocorreu (Imagem 3).

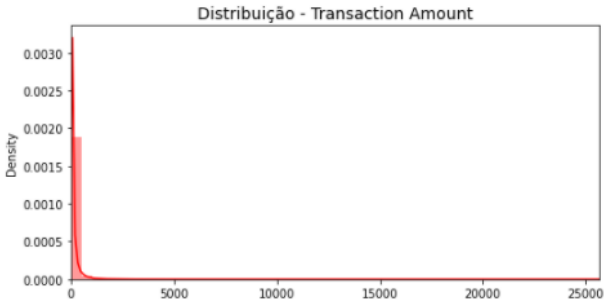


Imagem 2 - Distribuição Densidade x Valor da Transação (Amount)

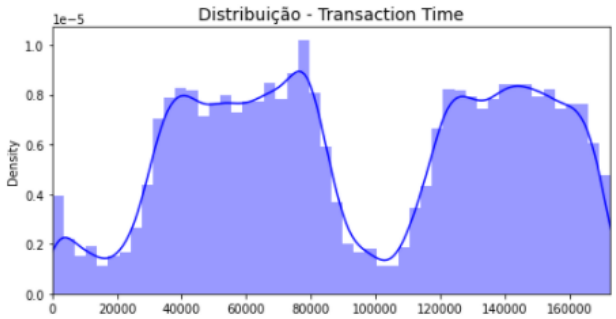


Imagem 3 - Distribuição Densidade x Time

2.2 Normalizando as Features ‘Amount’ e ‘Time’

Como as features de V1 a V28 já foram tratadas previamente, foi preciso normalizar apenas as features 'Amount' e ‘Time’. Foi usado o método RobustScalar da biblioteca scikit-learn. Os valores das features ‘Amount’ e ‘Time’ então foram normalizados de -1,0 (mínimo) a 1,0 (máximo) e atualizados no DataSet.

2.3 Criando um DataFrame com os dados balanceados

A ideia inicial seria utilizar o conjunto de dados original para treinar os modelos. Seria feita a estratificação dos dados para para gerar os conjuntos de treino e teste com a mesma proporção de classe Fraude e Não Fraude. Porém, por se tratar de um DataSet extremamente desbalanceado, qualquer modelo construído com o conjunto de dados original seria viciado e poderia ter uma acurácia grande mesmo que apenas predissesse todas as transações como não fraudulentas.

Assim optamos por criar um Data Frame balanceado (Random Under-Sampling). Como nossas classes são altamente distorcidas, sendo o conjunto Fraude muito maior que o Não Fraude, recolhemos uma amostra Não Fraude com o mesmo tamanho do conjunto Fraude (492). Em seguida criamos o Data Frame Balanceado com uma distribuição normal e aleatória das classes.

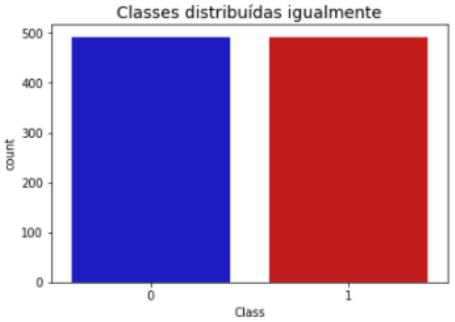


Imagem 4 - Distribuição das Classes DataFrame Balanceado

2.4 Análise Matriz de Correlação

Plotamos a matriz de correlação para verificar a distribuição das classes. Como é possível ver na imagem 5, agora temos uma boa base de dados para treinar os classificadores.

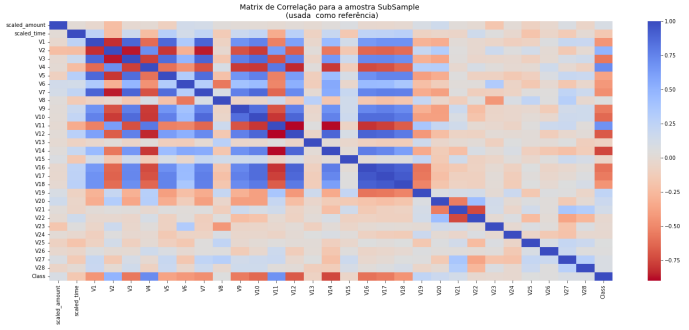


Imagem 5 - Matriz de Correlação DataFrame Balanceado

2.5 Separando os conjuntos de treino e teste

Foram gerados os conjuntos de Treino e Teste sobre o Data Frame Balanceado, sendo 80% dos dados para o conjunto de Treino e 20% para o conjunto de Teste.

2.6 Classificadores

Para treinar os modelos, foram escolhidos quatro tipos de classificadores:

- Logistic Regression
- KNeighbors Classifier
- Support Vector Classifier - SVC
- Decision Tree Classifier

Foi usado o GridSearchCV da biblioteca scikit-learn para encontrar os melhores parâmetros para cada classificador. Em seguida foram definidos os Predict para cada classificador e realizados os treinamentos dos modelos.

3 RESULTADOS

Após o treinamento dos modelos obtivemos os seguintes resultados das métricas de desempenho.

3.1 Métricas de Avaliação

3.1.1 Acurácia

```
LR - acurácia 0.9052631578947369
KN - acurácia 0.9
SVC - acurácia 0.8947368421052632
Tree - acurácia 0.8789473684210526
```

3.1.2 F1 score

```
LR - F1 Score 0.9032258064516129
KN - F1 Score 0.8972972972972972
SVC - F1 Score 0.8936170212765958
Tree - F1 Score 0.877005347593583
```

3.1.3 Recall

```
LR - Recall de 0.84
KN - Recall de 0.83
SVC - Recall de 0.84
Tree - Recall de 0.9425287356321839
```

3.1.4 Precision

```
LR  - Precision: 0.9767441860465116
KN  - Precision: 0.9764705882352941
SVC  - Precision: 0.9545454545454546
Tree  - Precision: 0.9425287356321839
```

3.1.5 Matriz de Confusão

Logistic Regression

	0	1
0	88	2
1	16	84

K Neighbors Classifier

	0	1
0	88	2
1	17	83

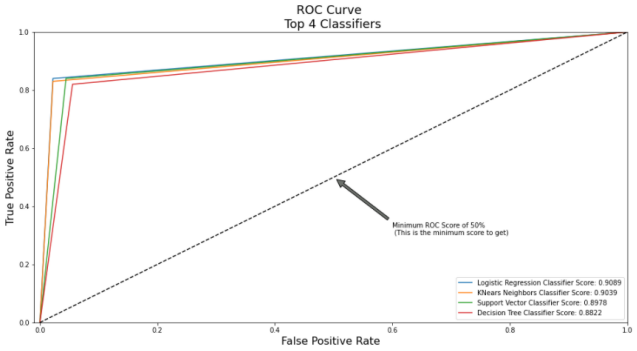
SVC Classifier

	0	1
0	86	4
1	16	84

Decision Tree Classifier

	0	1
0	85	5
1	18	82

3.1.6 Roc Curve



4 CONCLUSÃO

Através da análise das métricas obtidas para os classificadores foi possível perceber que os quatro obtiveram bom desempenho no conjunto de dados analisados. Com o conjunto de dados balanceado, o TradeOff Precision/Recall apresentou valores próximos, bem equilibrados para todos os classificadores, contudo, o classificador Logistic Regression teve um desempenho levemente superior mas com métricas bem próximas às do classificador K Neighbors e do SVC em alguns casos.

É importante salientar que a construção do modelo, em uma situação real e mais factível, deveria contar com um conjunto de dados mais robusto e heterogêneo. Ademais, foi possível constatar a importância da análise prévia do conjunto de dados e do tratamento dos mesmos, essencial para que os classificadores treinem modelos com maior capacidade de generalizar as classes.

5 REFERÊNCIAS

Dataset: https://www.kaggle.com/mlg-ulb/creditcardfraud