Instituto Politécnico da Huíla
Departamento de Informática e Ciência da Computação

# Detecção de Fraudes em Cartões de Crédito Usando Random Forest.

Pedro Calenga - 2022110222

Curso: Ciência da Computação

Orientador: Abel Zacarias

Data: 15 de Maio de 2025


UMN

INSTITUTO POLITÉCNICO DA HUÍLA

Instituto Politécnico da Huíla
Departamento de Informática e Ciência da Computação

Author Note

Pedro Calenga, Departamento de Informática e Ciência da Computação, Universidade Mandume Ya Ndemufayo.

Agradeço ao orientador, Abel Zacarias, por sua orientação técnica, aos colegas da Universidade Mandume Ya Ndemufayo pelo feedback nos testes da aplicação.

Este projeto foi desenvolvido como parte do curso de Ciência da Computação.

Contato: mended2003@gmail.com

Aplicação Streamlit: https://detection-fraude-ap.streamlit.app/

Repositório GitHub: https://github.com/catheke/detection-fraude-ap

Declaração de Originalidade

Declaro que este trabalho é original, realizado por mim, Pedro Calenga, e que todas as fontes utilizadas foram devidamente citadas conforme as normas da APA (7ª edição).

Lubango, 15 de Maio de 2025

_____

Pedro Calenga

Dedicado à minha família e à comunidade angolana,
cujo futuro digital este projeto busca proteger.

“A dor é inevitável, mas o sofrimento é opcional.”

— Pain, Naruto

Abstract

This study developed a credit card fraud detection model using Random Forest, applied to the Kaggle dataset (284,807 transactions, 0.17% fraudulent). The goal was to create an efficient system, minimizing false negatives that cause financial losses. After normalization with StandardScaler and class balancing with SMOTE (199,020 instances per class), the model achieved an accuracy of 1.00, recall of 0.78, 19 false positives, 33 false negatives, and AUC-ROC of 0.93. Cross-validation confirmed robustness (mean AUC 0.93 ± 0.02). Compared to Logistic Regression (AUC 0.9272, 36 false negatives), Random Forest was superior. The project is relevant to Angola, where digital payments are growing. Limitations include anonymized PCA features and lack of local data. The Streamlit application (https://detection-fraude-ap.streamlit.app/) demonstrates practical viability.

Keywords: fraud detection, Random Forest, SMOTE, credit cards, Angola

Resumo

Este estudo desenvolveu um modelo de detecção de fraudes em cartões de crédito usando Random Forest, aplicado ao dataset do Kaggle (284.807 transações, 0,17% fraudulentas). O objetivo foi criar um sistema eficiente, minimizando falsos negativos, que geram perdas financeiras. Após normalização com StandardScaler e balanceamento com SMOTE (199.020 instâncias por classe), o modelo alcançou acurácia de 1,00, recall de 0,78, 19 falsos positivos, 33 falsos negativos e AUC-ROC de 0,93. A validação cruzada confirmou robustez (AUC médio 0,93 ± 0,02). Comparado à Regressão Logística (AUC 0,9272, 36 falsos negativos), o Random Forest foi superior. O projeto é relevante para Angola, onde os pagamentos digitais crescem. Limitações incluem features PCA anonimizadas e falta de dados locais. A aplicação Streamlit (https://detection-fraude-ap.streamlit.app/) demonstra viabilidade prática.

Palavras-chave: detecção de fraudes, Random Forest, SMOTE, cartões de crédito, Angola

# Sumário Executivo

Este projeto desenvolveu um modelo Random Forest para detectar fraudes em cartões de crédito, alcançando recall de 0,78, AUC-ROC de 0,93, 19 falsos positivos e 33 falsos negativos. Usando o dataset do Kaggle (284.807 transações), o modelo foi treinado com SMOTE para balancear classes. A aplicação Streamlit permite previsão em tempo real, sendo relevante para bancos angolanos, onde os pagamentos digitais cresceram 30% entre 2020 e 2023. Comparado à Regressão Logística, o Random Forest reduz falsos negativos, minimizando perdas (ex.: 100.000 AOA por fraude). Recomenda-se integrar o modelo em sistemas bancários e coletar dados locais via parcerias com o Banco Nacional de Angola. O projeto demonstra viabilidade técnica e impacto social, protegendo consumidores em Angola.

# Contents

# List of Tables

# 1 Introdução

Fraudes em cartões de crédito causam perdas globais de $5,9 bilhões anualmente (Statista, 2024). Em Angola, o crescimento de 30% nas transações electrónicas entre 2020 e 2023 (Banco Nacional de Angola, 2024) eleva a necessidade de soluções locais. Este projeto utiliza o dataset Credit Card Fraud Detection do Kaggle, com 284.807 transações (0,17% fraudulentas), para desenvolver um modelo Random Forest, minimizando falsos negativos. Os objetivos gerais são proteger consumidores e instituições financeiras. Os objetivos específicos são:

- Desenvolver um modelo de machine learning para detectar fraudes.
- Minimizar falsos negativos para reduzir perdas financeiras.
- Comparar Random Forest com Regressão Logística.
- Aplicar o modelo ao contexto angolano via aplicação Streamlit.

A relevância reside na segurança financeira em um mercado em expansão.

# 2 Revisão da Literatura

A detecção de fraudes é um campo consolidado. Algoritmos como Regressão Logística e Random Forest são eficazes em dados desbalanceados (Pedregosa et al., 2011). O SMOTE cria amostras sintéticas para balanceamento (Chawla et al., 2002). Dal Pozzolo et al. (2015) destacam a importância de minimizar falsos negativos. Bhattacharyya et al. (2011) notaram a robustez do Random Forest. Adewumi e Akinyelu (2020) sugerem que dados locais melhoram a precisão em contextos africanos, uma lacuna deste projeto. Roy et al. (2018) testaram redes neurais, mas estas exigem mais dados. A escassez de estudos sobre fraudes em Angola reforça a relevância deste trabalho.

# 3 Metodologia

## 3.1 Dataset

O dataset contém 284.807 transações com 31 colunas: Time, Amount, V1 a V28 (features anonimizadas via PCA) e Class (0 para não fraude, 1 para fraude).

## 3.2 Análise Exploratória

Table 1: Estatísticas Descritivas do Dataset

| Variável | Média | Desvio Padrão |
|---|---|---|
| Time | 94813.86 | 47488.15 |
| Amount | 88.35 | 250.12 |

## 3.3 Pré-processamento

Time e Amount foram normalizados com StandardScaler. O SMOTE balanceou as classes, resultando em 199.020 instâncias por classe no conjunto de treino.

## 3.4 Modelo

O Random Forest (100 estimadores) foi treinado e comparado com a Regressão Logística. A validação cruzada (5-fold) usou AUC como métrica principal.

## 3.5 Métricas

Foram calculadas acurácia, precisão, recall, F1-score, falsos positivos (FP), falsos negativos (FN) e AUC-ROC.

## 3.6 Aplicação

Uma aplicação Streamlit foi desenvolvida (https://detection-fraude-ap.streamlit.app/) para previsão em tempo real e em lote.

# 4 Resultados

O Random Forest apresentou os seguintes resultados no conjunto de teste (85.443 instâncias):

- Acurácia: 1.00
- Falsos Positivos (FP): 19
- Falsos Negativos (FN): 33
- Precisão (Fraude): 0.86
- Recall (Fraude): 0.78
- F1-score (Fraude): 0.82
- AUC-ROC: 0.93

Table 2: Métricas de Desempenho do Random Forest

| Métrica | Valor |
|---|---|
| Acurácia | 1.00 |
| Falsos Positivos (FP) | 19 |
| Falsos Negativos (FN) | 33 |
| Precisão (Fraude) | 0.86 |
| Recall (Fraude) | 0.78 |
| F1-score (Fraude) | 0.82 |
| AUC-ROC | 0.93 |

Table 3: Matriz de Confusão do Random Forest

| | Predito: Não Fraude | Predito: Fraude |
|---|---|---|
| Real: Não Fraude | 85276 | 19 |
| Real: Fraude | 33 | 115 |

A Regressão Logística teve AUC de 0.9272, 5 falsos positivos e 36 falsos negativos. Um teste t indicou diferença significativa no AUC ($p < 0.05$). A importância das features destacou V14 e V17 (ver aplicação Streamlit para gráfico).

Table 4: Resultados de Validação Cruzada (5-fold)

| Fold | AUC |
|---|---|
| 1 | 0.94 |
| 2 | 0.92 |
| 3 | 0.93 |
| 4 | 0.93 |
| 5 | 0.92 |
| Média | 0.93 |
| Desvio Padrão | 0.02 |

# 5 Discussão

O Random Forest superou a Regressão Logística, reduzindo falsos negativos de 36 para 33, crucial para minimizar perdas (ex.: 100.000 AOA por fraude não detectada). Falsos positivos (19) causam inconveniência, como bloqueio de compras legítimas (ex.: 5.000 AOA em Luanda). O custo computacional do Random Forest é moderado, viável para bancos angolanos, com implementação estimada em milhares de kwanzas, mas evita perdas maiores. Limitações incluem:

- Features PCA anonimizadas, dificultando interpretação.
- Ruído potencial do SMOTE.
- Dataset global, sem dados angolanos.

A aplicação Streamlit demonstra viabilidade prática para integração em sistemas bancários.

# 6 Considerações Éticas

Falsos positivos podem bloquear transações legítimas, afetando clientes de baixa renda em Angola (ex.: compra de 5.000 AOA em um mercado local). A privacidade dos dados financeiros exige conformidade com regulamentações angolanas e internacionais. Este projeto prioriza a minimização de falsos negativos para proteger consumidores, mas recomenda monitoramento ético na implementação para evitar discriminação ou inconvenientes.

# 7 Conclusão e Recomendações

O projeto alcançou seus objetivos, desenvolvendo um modelo Random Forest com recall de 0.78, AUC-ROC de 0.93, 19 falsos positivos e 33 falsos negativos. A aplicação Streamlit facilita a previsão em tempo real. Para implementação prática, recomenda-se:

- Estabelecer parceria com o Banco Nacional de Angola para coletar dados locais, iniciando com um piloto de 6 meses.
- Testar algoritmos como XGBoost ou redes neurais para melhorar o recall.
- Integrar o modelo em sistemas bancários, com treinamento para equipas técnicas em 3 meses.

O projeto contribui para a segurança financeira em Angola, alinhado ao crescimento dos pagamentos digitais.

# Glossário

- AUC-ROC: Área sob a curva ROC, mede a capacidade de discriminação do modelo.
- SMOTE: Técnica para balancear classes criando amostras sintéticas.
- PCA: Análise de Componentes Principais, usada para anonimizar features.
- Falsos Negativos: Fraudes não detectadas, causam perdas financeiras.
- Falsos Positivos: Transações legítimas marcadas como fraudes.

# 8 Referências

## References

[1] Kaggle. (2016). Credit Card Fraud Detection Dataset. Disponível em: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

[2] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.

[3] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 16, 321-357.

[4] Dal Pozzolo, A., Caelen, O., Le Borgne, Y.-A., Waterschoot, S., & Bontempi, G. (2015). Learned lessons in credit card fraud detection from a practitioner perspective. Expert Systems with Applications, 41(10), 4915-4928.

[5] Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). Data mining for credit card fraud: A comparative study. Decision Support Systems, 50(3), 602-613.

[6] Adewumi, A. O., & Akinyelu, A. A. (2020). A survey of machine-learning and nature-inspired based credit card fraud detection techniques. International Journal of System Assurance Engineering and Management, 11(Suppl 2), 332-346.

[7] Roy, A., Sun, J., Mahoney, R., Alonzi, L., Adams, S., & Beling, P. (2018). Deep learning detecting fraud in credit card transactions. Systems and Information Engineering Design Symposium, 1-6.

[8] Statista. (2024). Financial fraud losses worldwide. Disponível em: https://www.statista.com/statistics/

[9] Banco Nacional de Angola. (2024). Relatório anual de transações electrónicas. Disponível em: https://www.bna.ao/

[10] American Psychological Association. (2020). Publication Manual of the American Psychological Association (7th ed.). APA.

# Apêndices

## Apêndice A: Código de Treinamento (Resumido)

Código completo disponível em: https://github.com/catheke/detection-fraude-ap.

```
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import StandardScaler
from imblearn.over_sampling import SMOTE

# Carregar e normalizar dataset
df = pd.read_csv('creditcard.csv')
X = df.drop('Class', axis=1)
y = df['Class']
scaler = StandardScaler()
X[['Time', 'Amount']] = scaler.fit_transform(X[['Time', 'Amount']])

# Aplicar SMOTE e treinar modelo
smote = SMOTE(random_state=42)
X_bal, y_bal = smote.fit_resample(X, y)
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_bal, y_bal)
```

## Apêndice B: Link da Aplicação

Aplicação Streamlit: https://detection-fraude-ap.streamlit.app/

## Apêndice C: Guia da Aplicação Streamlit

1. Acesse https://detection-fraude-ap.streamlit.app/.

2. Na seção "Prever Transação", insira Time, Amount e V1 a V28 para previsão

manual.

3. Visualize métricas (ex.: matriz de confusão, importância das features) na seção "Desempenho do Modelo".