Relatório Técnico: Sistema Inteligente de Detecção de Fraudes em Transações com Cartões de Crédito

Trabalho de Conclusão de Curso
Autor: Matheus Mendes Neves
João Pedro
Orientador: Fabio Oliveira Guimaraes
Data: Julho de 2025

Sumário Executivo

Este trabalho apresenta o desenvolvimento de um sistema inteligente de detecção de fraudes em transações com cartão de crédito, utilizando algoritmos de aprendizado de máquina. Diante do cenário atual de crescente volume de transações digitais e do aumento expressivo de fraudes financeiras, o projeto busca atender à necessidade das instituições financeiras de identificar automaticamente transações suspeitas em tempo real.

A solução desenvolvida tem como foco principal a **maximização da detecção de fraudes (recall)**, mantendo os falsos positivos em níveis aceitáveis. O modelo escolhido foi o **XGBoost**, conhecido por sua alta performance e robustez. O sistema foi preparado para implantação em produção por meio de uma **API REST modularizada**, facilitando integração com plataformas financeiras reais.

---

1. Introdução

1.1 Contextualização

O setor de cartões de crédito movimenta trilhões de dólares globalmente. Com a digitalização acelerada pós-pandemia, observou-se um crescimento significativo nas tentativas de fraude eletrônica, impondo um desafio crescente para instituições financeiras. Segundo relatórios recentes, somente em 2024, as perdas com fraudes em cartões ultrapassaram bilhões de dólares.

1.2 Problema

A detecção manual de fraudes tornou-se impraticável diante do elevado volume de transações. Sistemas baseados em regras fixas são obsoletos frente à evolução constante das técnicas de fraude, exigindo abordagens adaptativas baseadas em aprendizado de máquina.

1.3 Justificativa

A proposta justifica-se pelos seguintes pontos:

- Necessidade de **automatização da detecção em tempo real**;
- Redução de **perdas financeiras** e impacto negativo ao consumidor;
- Aumento da **eficiência operacional** e redução de bloqueios injustificados;
- Capacidade do modelo de **adaptar-se a novas fraudes** com base em padrões aprendidos.

1.4 Objetivos

Objetivo Geral:

Desenvolver um sistema inteligente para detecção de fraudes em cartões de crédito utilizando algoritmos de Machine Learning.

Objetivos Específicos:

- Analisar e preparar dados financeiros;
- Testar e comparar algoritmos de classificação;
- Otimizar os modelos para maximizar o recall;
- Desenvolver arquitetura pronta para produção com API RESTful.

---

2. Fundamentação Teórica

2.1 Machine Learning na Detecção de Fraudes

O aprendizado de máquina é adequado para esse tipo de aplicação devido à sua capacidade de:

- Aprender padrões complexos e não lineares;
- Adaptar-se a novos comportamentos fraudulentos;
- Processar grandes volumes de dados em tempo real;
- Reduzir falsos positivos por meio de otimização contínua.

2.2 Desafios Específicos

2.2.1 Desbalanceamento de Classes

- Fraudes representam menos de 1% do total de transações;
- Algoritmos convencionais tendem a ignorar classes minoritárias;
- Requer técnicas como **SMOTE** para balanceamento.

2.2.2 Métricas de Avaliação

Prioriza-se:

- **Recall**: maximizar a detecção de fraudes;

- **Precision**: minimizar alarmes falsos;
- **F1-Score**: equilibrar precisão e sensibilidade;
- **AUC-ROC**: avaliar capacidade discriminativa geral.

2.3 Algoritmos Utilizados

- **Regressão Logística**: modelo base, simples e interpretável;
- **Random Forest**: robustez e resistência a overfitting;
- **XGBoost**: estado da arte em boosting, com excelente performance.

---

3. Metodologia

3.1 Dataset Utilizado

- Fonte: Transações de cartão de crédito na Europa (setembro/2013);
- Total de registros: 284.807
- Fraudes: 492 (0,172%)
- Variáveis: 30 (28 via PCA + Time + Amount + Class)

3.2 Estrutura do Projeto

```
projeto_fraude/
├── data/
│   └── creditcard.csv
├── notebooks/
│   └── 01_modelagem_fraude.ipynb
├── models/
│   ├── modelo_fraude.pkl
│   ├── scaler_amount.pkl
│   └── feature_names.pkl
└── api/
    └── fraud_api.py
```

3.3 Pipeline de Desenvolvimento

- **EDA**: análise de qualidade dos dados, distribuição das classes e correlações;
- **Pré-processamento**: normalização, balanceamento com SMOTE, divisão estratificada;
- **Modelagem**: comparação entre algoritmos, validação cruzada e otimização de hiperparâmetros;
- **Avaliação**: foco em recall e AUC-ROC, análise de matrizes de confusão e curvas ROC.

---

4. Implementação

4.1 Pré-processamento

4.1.1 Normalização do Amount

```
scaler = StandardScaler()
df['Amount_Norm'] = scaler.fit_transform(df[['Amount']])
```

Motivo: as variáveis V1-V28 já estão normalizadas. Amount, não.

4.1.2 Balanceamento com SMOTE

```
smote = SMOTE(random_state=42)
X_train_balanced, y_train_balanced = smote.fit_resample(X_train, y_train)
```

Resultado:

- Antes: 227.445 normais vs. 394 fraudes
- Depois: 227.445 fraudes sintéticas geradas para balanceamento 1:1

4.2 Modelagem

```
# Regressão Logística
log_model = LogisticRegression(...)

# Random Forest
rf_model = RandomForestClassifier(...)

# XGBoost
xgb_model = XGBClassifier(...)
```

4.3 Avaliação

```
def avaliar_modelo(...):
    # retorna accuracy, precision, recall, f1_score e auc_roc
```

5. Resultados e Análise

5.1 Comparativo de Modelos

Modelo	Accuracy	Precision	Recall	F1-Score	AUC-ROC
XGBoost	0.9995	0.9468	0.8265	0.8826	0.9759
Random Forest	0.9994	0.9354	0.8061	0.8662	0.9612
Regressão Logística	0.9992	0.8976	0.7857	0.8382	0.9548

5.2 Análise do Modelo Campeão (XGBoost)

- **Recall:** 82.65%
- **Precision:** 94.68%
- **F1-Score:** 88.26%
- **AUC-ROC:** 97.59%

Matriz de Confusão:

	Predito: Normal	**Predito: Fraude**
Real: Normal	56.844	28

Real: Fraude	17	81

6. Preparação para Produção

6.1 Salvamento de Artefatos

```
joblib.dump(xgb_model, 'modelo_fraude.pkl')
```

6.2 API REST

Entrada:

```
{
 "Time": 0,
 "V1": -1.35,
 "V2": -0.07,
  ...
"Amount": 149.62
}
```

Saída:

```
{
 "probabilidade_fraude": 0.95,
 "classificacao": "FRAUDE",
 "confianca": "ALTA"
}
```

7. Discussão

7.1 Pontos Fortes

- Alta performance (Recall > 82%)
- Baixo número de falsos positivos (0.05%)
- Arquitetura modular e escalável

7.2 Limitações

- Base de dados antiga (2013)
- SMOTE pode gerar padrões artificiais
- Falta de interpretabilidade (PCA)

---

8. Trabalhos Futuros

- Ensemble com diferentes algoritmos
- Feature engineering avançado
- Aprendizado online e monitoramento de concept drift
- Uso de SHAP para explicabilidade

9. Conclusão

9.1 Objetivos Atingidos

- Análise de dados com EDA
- Testes com múltiplos modelos
- Otimização para alta detecção de fraudes
- Sistema preparado para produção

9.2 Contribuições

- Pipeline robusto e replicável
- Redução de perdas financeiras
- Solução aplicável em ambientes reais

9.3 Considerações Finais

O projeto confirma a viabilidade do uso de algoritmos de Machine Learning para detecção de fraudes, demonstrando alta eficácia com métricas robustas. A abordagem proposta é escalável, prática e representa uma solução significativa para o setor financeiro.

---

10. Referências Bibliográficas

Chawla, N. V., et al. (2002). SMOTE: Synthetic Minority Oversampling Technique. Journal of Artificial Intelligence Research.

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference.

Hand, D. J. (2012). Measuring classifier performance: a coherent alternative to the area under the ROC curve. Machine Learning.

Makki, S., et al. (2019). An Experimental Study With Imbalanced Classification Approaches for Credit Card Fraud Detection. IEEE Access.

Pozzolo, A. D., et al. (2014). Credit card fraud detection: a realistic modeling and a novel learning strategy. IEEE Transactions on Neural Networks.

Scikit-learn Documentation (2024). Machine Learning Library for Python. Available at: https://scikit-learn.org/

XGBoost Documentation (2024). Scalable Machine Learning. Available at: https://xgboost.readthedocs.io/