Credit Card Fraud Detection: A Machine Learning Approach

Segneanu Razvan
Masters Student, Faculty of Automation and Computers
Politehnica University of Timioara

August 5, 2025

Abstract

This project implements a comprehensive credit card fraud detection system using Python and machine learning techniques. The system leverages a Random Forest Classifier to identify fraudulent transactions with high accuracy, incorporating data preprocessing, exploratory data analysis (EDA), and interactive visualizations. The codebase is designed to be modular, reusable, and well-documented, making it suitable for both academic and professional applications. This documentation provides an overview of the project, its methodology, results, and instructions for deployment.

Contents

1 Introduction

The Credit Card Fraud Detection project aims to identify fraudulent transactions in a financial dataset using machine learning. The system processes the `creditcard_2023.csv` dataset, sourced from https://www.kaggle.com/datasets/nelgiriyewithana/credit-card-fraud-detection-dat performs exploratory data analysis, and trains a Random Forest Classifier to achieve high accuracy in fraud detection. Key features include interactive visualizations, data preprocessing pipelines, and detailed performance metrics.

The project is implemented in Python using Jupyter Notebooks, leveraging libraries such as `pandas`, `scikit-learn`, `seaborn`, `matplotlib`, and `ipywidgets`. The codebase is designed to be modular and extensible, with a focus on reproducibility and ease of use.

2 Dataset Description

The dataset used is `creditcard_2023.csv`, containing anonymized credit card transaction data. Key characteristics include:

- **Features**: 30 columns, including 28 anonymized features (V1 to V28), `Amount`, `Time`, and `Class`.
- **Target Variable**: `Class` (0 for legitimate transactions, 1 for fraudulent transactions).
- **Size**: Approximately 568,630 transactions, balanced between fraudulent and non-fraudulent cases.
- **Source**: Publicly available dataset from Kaggle (https://www.kaggle.com/datasets/nelgiriyewithana/credit-card-fraud-detection-dataset-2023).

3 Methodology

The project follows a structured approach to data analysis and model development, as outlined below.

3.1 Data Preprocessing

- **Data Loading**: The dataset is loaded using `pandas.read_csv`.
- **Cleaning**: Duplicates and missing values are removed using `drop_duplicates()` and `dropna()`.
- **Feature Scaling**: Numerical features are standardized using `StandardScaler` within a `scikit-learn` pipeline.

3.2 Exploratory Data Analysis (EDA)

The EDA phase includes:

- **Overview**: Displays the first few rows, statistical summary, missing values, and duplicates.
- **Unique Values**: Lists unique values for each column (first 7 shown).

- **Outlier Detection**: Identifies outliers using the Interquartile Range (IQR) method.
- **Frequent Values**: Shows the top 3 most frequent values per column.
- **Memory Usage**: Summarizes memory consumption per column.
- **Visualizations**:
  - Missing values heatmap using `seaborn.heatmap`.
  - Correlation matrix for numerical features.
  - Distribution histograms for the first five numerical columns.
  - Pie charts for categorical variables (e.g., `Class`).

Interactive widgets (`ipywidgets`) allow users to select different EDA functions dynamically.

3.3 Model Development

The machine learning model is a Random Forest Classifier implemented within a `scikit-learn` pipeline:

- **Preprocessing**: Numerical features are scaled using `StandardScaler`.
- **Model**: `RandomForestClassifier` with `random_state=42` for reproducibility.
- **Training**: The dataset is split into 80% training and 20% testing sets using `train_test_split`.
- **Evaluation**: Performance is evaluated using accuracy, precision, recall, and F1-score.

3.4 Visualization Enhancements

- **Pie Charts**: Custom subplots display the distribution of the `Class` variable using `seaborn` and `matplotlib`.
- **Interactivity**: An optional `mpld3` plugin enables hover-to-zoom functionality for pie charts.

4 Results

The Random Forest Classifier achieved the following performance metrics on the test set:

- **Accuracy**: 0.9998
- **Class 0 (Non-Fraudulent)**:
  - Precision: 1.00
  - Recall: 1.00
  - F1-score: 1.00
  - Support: 56,750

- **Class 1 (Fraudulent)**:
  - Precision: 1.00
  - Recall: 1.00
  - F1-score: 1.00
  - Support: 56,976
- **Macro Average**:
  - Precision: 0.9998
  - Recall: 0.9998
  - F1-score: 0.9998
  - Support: 113,726
- **Weighted Average**:
  - Precision: 0.9998
  - Recall: 0.9998
  - F1-score: 0.9998
  - Support: 113,726

These results indicate excellent model performance, with near-perfect classification of both fraudulent and non-fraudulent transactions.

5 Installation and Usage

5.1 Dependencies

The project requires the following Python packages:

- `pandas`
- `numpy`
- `scikit-learn`
- `matplotlib`
- `seaborn`
- `ipywidgets`
- `colorama`
- `mpld3` (optional, for interactive visualizations)

Install dependencies using:

```
pip install pandas numpy scikit-learn matplotlib seaborn ipywidgets
    colorama mpld3
```

5.2 Running the Project

1. Clone the repository:

```
    git clone https://github.com/your-username/credit-card-fraud-
        detection.git
    cd credit-card-fraud-detection
```

2. Download the dataset (`creditcard_2023.csv`) from https://www.kaggle.com/datasets/nelgiriyewithana/credit-card-fraud-detection-dataset-2023 and place it in the project directory.

3. Open the Jupyter Notebook:

```
    jupyter notebook creditCardFraudDetection.ipynb
```

4. Run all cells to perform EDA, train the model, and view results.

5.3 Interactive Usage

The notebook includes an interactive menu powered by `ipywidgets`. Users can select from options such as:

- Dataset Overview
- Unique Values
- Outlier Detection
- Correlation Matrix
- Clean Dataset
- Export Cleaned CSV
- ML Model Suggestion

6 Project Structure

```
credit-card-fraud-detection/
 creditCardFraudDetection.ipynb  % Main Jupyter Notebook
 creditcard_2023.csv             % Input dataset (not included in repo)
 cleaned_dataset.csv             % Exported cleaned dataset
 README.md                       % Project documentation
 LICENSE                         % License file
```

7 Future Improvements

Potential enhancements include:

- Adding cross-validation to improve model robustness.
- Implementing additional models (e.g., XGBoost, Neural Networks) for comparison.
- Enhancing visualizations with more interactive features using `plotly`.
- Optimizing memory usage for larger datasets.
- Deploying the model as a web application using `Flask` or `FastAPI`.

8 Conclusion

This project demonstrates a robust approach to credit card fraud detection using machine learning. The Random Forest Classifier achieves near-perfect accuracy, supported by comprehensive EDA and interactive visualizations. The codebase is well-documented and suitable for deployment on platforms like GitHub and LinkedIn to showcase data science and machine learning expertise.

9 References

- `scikit-learn` Documentation: https://scikit-learn.org/stable/
- `pandas` Documentation: https://pandas.pydata.org/docs/
- `seaborn` Documentation: https://seaborn.pydata.org/
- `matplotlib` Documentation: https://matplotlib.org/stable/
- `mpld3` Documentation: https://mpld3.github.io/
- Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5-32. https://doi.org/10.1023/A:1010933404324 – Overview of the Random Forest algorithm.
- Caruana, R., & Niculescu-Mizil, A. (2006). An Empirical Comparison of Supervised Learning Algorithms. *Proceedings of the 23rd International Conference on Machine Learning*, 161-168. – Comparison of Random Forest with other algorithms like SVM and Logistic Regression.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. Springer. – Detailed discussion on ensemble methods including Random Forest.
- Kaggle Dataset: Credit Card Fraud Detection Dataset 2023, https://www.kaggle.com/datasets/nelgiriyewithana/credit-card-fraud-detection-dataset-2023.

[no-math]fontspec Noto Serif