Name: Md. Mushfiqur Rahman

Project: Credit Card Fraud Detection Using Machine Learning

Introduction

Credit card fraud is a significant challenge faced by financial institutions globally. This project aims to develop a machine learning-based system for detecting fraudulent transactions using basic models like Logistic Regression and Random Forest. The system uses a publicly available dataset containing transaction data, where each transaction is labeled as fraudulent or legitimate.

Objectives

1. To preprocess and analyze the given dataset.
2. To build and evaluate machine learning models for fraud detection.
3. To compare the performance of Logistic Regression and Random Forest classifiers.

Dataset

The dataset used in this project, `creditcard.csv`, contains:

- **Features**: Transaction-specific information.
- **Target**: A binary column (`Class`), where:
  - `0`: Legitimate transaction
  - `1`: Fraudulent transaction

Dataset Characteristics

- The dataset is highly imbalanced, with legitimate transactions far outweighing fraudulent ones.
- Features have been standardized for privacy, except for the `Class` column.

Methodology

Step 1: Data Exploration

- Load the dataset using `pandas`.
- Examine the structure and overview of the data.
- Analyze class distribution to understand the imbalance.

Step 2: Data Preprocessing

- Split the data into **features** (`X`) and **target** (`y`).
- Standardize the features using `StandardScaler` to improve model performance.
- Perform a stratified train-test split to maintain class distribution in training and testing sets.

Step 3: Model Development

Logistic Regression

A linear model suitable for binary classification tasks.

- Train the model using the training set.
- Predict labels on the test set.
- Evaluate the model using metrics like confusion matrix and classification report.

Random Forest Classifier

An ensemble-based model that builds multiple decision trees.

- Train the model using the training set.
- Predict labels on the test set.
- Evaluate using the same metrics as above.

Step 4: Model Comparison

- Compare the models based on accuracy, precision, recall, and F1-score.
- Highlight the suitability of Random Forest for imbalanced datasets due to its robustness and ability to handle complex patterns.

Requirements

Libraries

- `pandas`

- `numpy`
- `scikit-learn`

Tools

- Python 3.7+

Code

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix

# Load the dataset
data = pd.read_csv('creditcard.csv')

# Explore the dataset
print("Dataset Overview:")
print(data.head())
print("\nDataset Information:")
print(data.info())
print("\nClass Distribution:")
print(data['Class'].value_counts())

# Data Preprocessing
X = data.drop(columns=['Class'])  # Features
y = data['Class']  # Target

# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42,
stratify=y)

# Model 1: Logistic Regression
```

```
print("\nTraining Logistic Regression Model...")
lr_model = LogisticRegression()
lr_model.fit(X_train, y_train)
lr_predictions = lr_model.predict(X_test)

print("\nLogistic Regression Results:")
print(confusion_matrix(y_test, lr_predictions))
print(classification_report(y_test, lr_predictions))

# Model 2: Random Forest Classifier
print("\nTraining Random Forest Model...")
rf_model = RandomForestClassifier(random_state=42)
rf_model.fit(X_train, y_train)
rf_predictions = rf_model.predict(X_test)

print("\nRandom Forest Results:")
print(confusion_matrix(y_test, rf_predictions))
print(classification_report(y_test, rf_predictions))

# Compare Models
print("\nComparison:")
print("Logistic Regression Accuracy:", lr_model.score(X_test, y_test))
print("Random Forest Accuracy:", rf_model.score(X_test, y_test))

# Conclusion
print("\nFraud detection models successfully trained and evaluated. Random Forest generally performs better for imbalanced datasets.")
```

Results and Evaluation

Logistic Regression

- **Confusion Matrix**: Tabular comparison of true and predicted labels.
- **Classification Report**: Includes precision, recall, F1-score, and accuracy.

```
Training Logistic Regression Model...

Logistic Regression Results:
[[56851    13]
 [   35    63]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     56864
           1       0.83      0.64      0.72        98

    accuracy                           1.00     56962
   macro avg       0.91      0.82      0.86     56962
weighted avg       1.00      1.00      1.00     56962
```

Random Forest Classifier

- Similar metrics as Logistic Regression but generally performs better for imbalanced data due to ensemble learning.

```
Training Random Forest Model...

Random Forest Results:
[[56859     5]
 [   18    80]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     56864
           1       0.94      0.82      0.87        98

    accuracy                           1.00     56962
   macro avg       0.97      0.91      0.94     56962
weighted avg       1.00      1.00      1.00     56962
```

Comparison Summary

- Logistic Regression provides a baseline for performance.
- Random Forest demonstrates improved metrics, particularly in detecting fraudulent transactions.

```
Comparison:
Logistic Regression Accuracy: 0.9991573329588147
Random Forest Accuracy: 0.9995962220427653
```

Conclusion

```
Fraud detection models successfully trained and evaluated. Random Forest generally performs better for imbalanced datasets.
```

This project demonstrates the application of basic machine learning models for fraud detection. While Logistic Regression offers simplicity and interpretability, Random Forest provides better accuracy and robustness for highly imbalanced datasets. Future work could include:

1. Implementing advanced techniques like SMOTE (Synthetic Minority Oversampling Technique) for balancing the dataset.
2. Exploring deep learning methods for improved detection.

References

1. Dataset source: Kaggle Credit Card Fraud Dataset
2. Scikit-learn Documentation: https://scikit-learn.org