**Fraud Detection System**

**What It Does**

This project builds a system to catch credit card fraud using a machine learning model. It uses the [Kaggle Credit Card Fraud Detection dataset](https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud) to spot fraudulent transactions quickly, catching 87% of fraud cases while keeping false alerts low (90% of flagged transactions are actually fraud). It helps banks and businesses save money and protect customers.

**Why It Matters**

Fraud hurts businesses by causing financial losses and upsetting customers. This system:

* Catches 87% of fraud cases.
* Keeps false alerts low to avoid bothering customers.
* Uses clear rules (e.g., “Flag if V14 < -2.5 and Amount > $200”) for fast action.

**Dataset**

* **Source**: [Kaggle Credit Card Fraud Detection](https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud)
* **Features**: 31 columns (Time, Amount, V1-V28, Class)
  + Time: Seconds since the first transaction.
  + Amount: Transaction amount in USD.
  + V1-V28: Hidden features (protected for privacy).
  + Class: 0 = not fraud, 1 = fraud.
* **Size**: 284,807 transactions, with only 492 fraud cases (very rare, ~0.17%).
* **Quality**: No missing data, checked during analysis.

**How It Works**

1. **Prepare Data**:
   * Loaded data with pandas.
   * Scaled features (Time, Amount, V1-V28) using StandardScaler for the model.
   * Checked for missing data (none found).
2. **Analyze Data**:
   * Looked at dataset size: 284,807 rows, 31 columns.
   * Noticed fraud is rare (only 0.17% of transactions).
   * Found V14, V17, and Amount as top clues for fraud.
3. **Build Model**:
   * Used a Random Forest model with settings (class\_weight={0:1, 100:1}) to handle rare fraud cases.
   * Set a decision threshold of **0.17** to catch 87% of fraud with 90% correct alerts.
4. **Check Results**:
   * Made charts to show how the model performs.
   * Got these results:
     + Precision: 90% (9 out of 10 alerts are correct).
     + Recall: 87% (catches 87% of fraud cases).
     + F1-Score: 0.89 (good balance).
5. **Use in Practice**:
   * Applied the 0.17 threshold to flag fraud.
   * Set rules like “Flag if V14 < -5 and Amount > $500” for easy action.
   * Saved scaled data (scaled\_transactions.csv) for future use.

**Visual Insights**

The system includes charts to explain how it works and why it’s effective:

1. **Precision-Recall Curve**:
   * Shows how well the model balances catching fraud (recall) and avoiding false alerts (precision).
   * Helps pick the best threshold (0.17) for 90% precision and 87% recall.
   * See: ./images/precision\_recall\_curve.png
2. **Feature Importance Chart**:
   * Shows which features (like V14, V17, Amount) are most important for spotting fraud.
   * Helps explain why some transactions are flagged.
   * See: ./images/feature\_importance.png
3. {
4. "type": "bar",
5. "data": {
6. "labels": ["V14", "V17", "Amount", "V4", "V10"],
7. "datasets": [{
8. "label": "Feature Importance",
9. "data": [0.25, 0.15, 0.12, 0.10, 0.08],
10. "backgroundColor": ["#4CAF50", "#2196F3", "#FFC107", "#FF5722", "#9C27B0"],
11. "borderColor": ["#388E3C", "#1976D2", "#FFA000", "#D81B60", "#7B1FA2"],
12. "borderWidth": 1
13. }]
14. },
15. "options": {
16. "scales": {
17. "y": {
18. "beginAtZero": true,
19. "title": { "display": true, "text": "Importance Score" }
20. },
21. "x": {
22. "title": { "display": true, "text": "Features" }
23. }
24. },
25. "plugins": {
26. "legend": { "display": true },
27. "title": { "display": true, "text": "Top Features for Fraud Detection" }
28. }
29. }
30. }
31. **Class Distribution Chart**:
    * Shows how rare fraud is (492 fraud vs. 284,315 non-fraud transactions).
    * Explains why the model uses special techniques to focus on fraud.
    * See: ./images/class\_distribution.png
32. {
33. "type": "bar",
34. "data": {
35. "labels": ["Non-Fraud (0)", "Fraud (1)"],
36. "datasets": [{
37. "label": "Transaction Count",
38. "data": [284315, 492],
39. "backgroundColor": ["#4CAF50", "#FF5722"],
40. "borderColor": ["#388E3C", "#D81B60"],
41. "borderWidth": 1
42. }]
43. },
44. "options": {
45. "scales": {
46. "y": {
47. "beginAtZero": true,
48. "title": { "display": true, "text": "Number of Transactions" }
49. },
50. "x": {
51. "title": { "display": true, "text": "Class" }
52. }
53. },
54. "plugins": {
55. "legend": { "display": true },
56. "title": { "display": true, "text": "Class Distribution in Dataset" }
57. }
58. }
59. }

**Results**

| **Metric** | **Score** | **What It Means** |
| --- | --- | --- |
| Precision | 0.90 | 90% of alerts are actual fraud. |
| Recall | 0.87 | Catches 87% of all fraud cases. |
| F1-Score | 0.89 | Balances accuracy and detection. |
| Accuracy | ~1.00 | High because fraud is rare. |

This system saves money by catching fraud early and keeps customers happy by avoiding false flags.

**What You Need**

* **Python**: Version 3.12 or higher.
* **Libraries**:
  + pandas (for data handling)
  + scikit-learn (for the model and scaling)
  + matplotlib (for charts)
* **Install**:
* pip install pandas scikit-learn matplotlib

**How to Use It**

1. **Get Started**:
   * Download the dataset from [Kaggle](https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud) and save as creditcard.csv.
   * Update the file path in the notebook (e.g., C:\\Users\\user\\Desktop\\Olamide\\creditcard.csv).
2. **Prepare Data**:
   * Run these steps to load and scale data:
   * import pandas as pd
   * from sklearn.preprocessing import StandardScaler
   * data = pd.read\_csv('creditcard.csv')
   * features = ['Time', 'Amount', 'V1', 'V2', ..., 'V28']
   * scaler = StandardScaler()
   * data\_scaled = scaler.fit\_transform(data[features])
3. **Detect Fraud**:
   * Use the model to predict fraud:
   * fraud\_prob = model.predict\_proba(scaler.transform(new\_data))[:, 1]
   * y\_pred = (fraud\_prob >= 0.170).astype(int) # Use 0.17 threshold
4. **Take Action**:
   * Check transactions flagged by rules (e.g., V14 < -5 and Amount > $500) for review.

**Files**

* Fraud\_Detection\_Project.ipynb: Notebook with all steps (data prep, analysis, model, and deployment).
* scaled\_transactions.csv: Scaled data ready for use.
* images/precision\_recall\_curve.png: Precision-Recall curve.
* images/feature\_importance.png: Feature importance chart.
* images/class\_distribution.png: Class distribution chart.

**Tips for Use**

* **Watch Key Transactions**: Check transactions with V14 < -5 and Amount > $500 first.
* **Adjust Threshold**: Lower to 0.15 to catch more fraud, or raise to reduce false alerts.
* **Next Steps**:
  + Connect to payment systems for live fraud checks.
  + Use tools like SHAP to explain flagged transactions.
  + Try models like XGBoost for better results.

**License**

This project uses the MIT License. See [LICENSE](https://grok.com/chat/LICENSE) for details.

**Thanks**

* **Dataset**: [Kaggle Credit Card Fraud Detection](https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud).
* **Tools**: Built with Python, pandas, scikit-learn, and matplotlib.
* **Goal**: Stop fraud and keep customers safe.