Transaction Anomaly Detection System -- Project Report

---

1. Project Title

Transaction Anomaly Detection System using Python

---

2. Objective

The objective of this project is to **detect anomalous (fraudulent) transactions** in a credit card dataset using both **supervised and unsupervised machine learning techniques**. This helps in preventing financial fraud and ensuring secure transactions.

---

3. Dataset Description

- **Source**: Kaggle – Credit Card Fraud Detection
- **Rows Used**: 5,000 (sampled from original dataset for demo purposes)
- **Columns**:

Column	Description
Time	Seconds elapsed since the first transaction in the dataset
V1–V28	Anonymized features obtained via PCA (for privacy reasons)
Amount	Transaction amount
Class	Target variable: 0 = Normal, 1 = Fraudulent

- **Key Points**:
    - Highly imbalanced dataset: fraud transactions <1% of total.
    - Features are anonymized to protect privacy.

---

4. Exploratory Data Analysis (EDA)

```
# Distribution of target

sns.countplot(x='Class', data=df_small)

plt.title("Fraud vs Normal Transactions")

plt.show()
```

Transaction Amount distribution

sns.histplot(df_small['Amount'], bins=30, kde=True)

plt.title("Transaction Amount Distribution")

plt.show()

Insights from EDA:

- Majority of transactions are normal (Class=0).
- Fraud transactions are rare but can be identified using patterns in PCA features and Amount.
- Amount varies widely; scaling is necessary for modeling.

---

5. Data Preprocessing

- **Scaling**: StandardScaler applied to Time and Amount.
- **Features/Target Split**:
- X = df_small.drop('Class', axis=1)
- y = df_small['Class']
- **Train/Test Split**: 80% training, 20% testing, stratified on Class.

---

6. Machine Learning Models

6.1 Supervised Learning – Random Forest

rf = RandomForestClassifier(n_estimators=100, random_state=42)

rf.fit(X_train, y_train)

y_pred = rf.predict(X_test)

Evaluation Metrics:

- Confusion Matrix
- Classification Report (Precision, Recall, F1-score)
- ROC-AUC Score

Insight: Random Forest can accurately classify most normal transactions and identify some fraudulent transactions despite class imbalance.

6.2 Unsupervised Learning – Isolation Forest

iso = IsolationForest(contamination=df_small['Class'].mean(), random_state=42)

iso.fit(X_train)

y_pred_iso = np.where(iso.predict(X_test)==1, 0, 1)

Insight:

- Useful when labels are unavailable.
- Detects anomalies (fraud) based on isolation score.

---

7. Demo Predictions

Sample 5 transactions:

Amount	Actual Class	Predicted Class
35.68	0	0
12.90	0	0
120.45	0	0
8.50	0	0
50.12	0	0

Note: In a small random sample, all transactions may be normal due to **rare fraud cases**.

- To see a fraud prediction: pick a transaction with Class=1.

---

8. Key Insights

1. Fraudulent transactions are extremely rare (<1%).
2. Supervised models (Random Forest, XGBoost) perform well if trained with enough data.
3. Unsupervised models (Isolation Forest) can detect anomalies without labels.
4. Feature scaling improves model performance for algorithms sensitive to magnitude.
5. PCA-anonymized features (V1–V28) are sufficient for detecting anomalies.

9. Conclusion

- The project demonstrates a **complete pipeline** for transaction anomaly detection:
    - Data sampling and preprocessing
    - EDA and feature scaling
    - Supervised and unsupervised model training
    - Evaluation and demo predictions
- The system can be deployed to **flag suspicious transactions in real-time**, providing a basis for fraud prevention.