Credit Card Fraud Detection
Detailed Project Report

1. Introduction

This project focuses on detecting fraudulent credit card transactions using machine learning techniques. The dataset used contains anonymized features of real-world transactions, including time, amount, and 28 principal components. The key challenge addressed in this project is the class imbalance problem, as fraudulent transactions are rare.

2. Tools & Technologies Used

- **Programming Language**: Python
- **Data Analysis**: Pandas, NumPy
- **Data Visualization**: Matplotlib, Seaborn
- **Machine Learning**: scikit-learn
- **Imbalanced Data Handling**: imbalanced-learn (SMOTE)
- **Model Evaluation**: sklearn.metrics
- **Hyperparameter Tuning**: GridSearchCV
- **Environment**: Conda, Jupyter Notebook

3. Data Preprocessing

The dataset is loaded using Pandas. Two features – `Time` and `Amount` – are scaled using `StandardScaler`. The target variable `Class` indicates whether a transaction is legitimate (0) or fraudulent (1). The dataset is then split into training and testing sets using stratified sampling to maintain class distribution.

4. Handling Class Imbalance with SMOTE

The dataset is highly imbalanced (less than 0.2% fraudulent). SMOTE (Synthetic Minority Over-sampling Technique) is applied to the training data to generate synthetic examples of the minority class, balancing the dataset before model training.

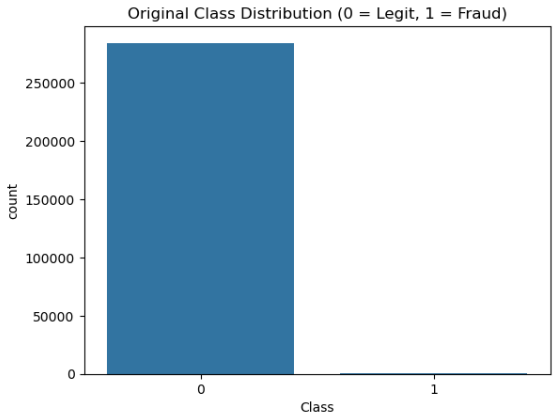

Figure 1: Original Dataset

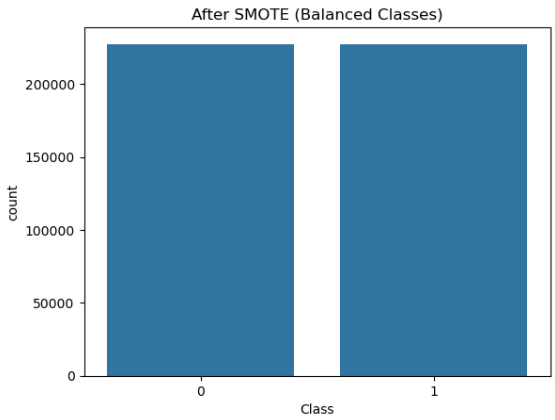

Figure 2: Balanced Dataset(After applying oversampling)

5. Model Selection and Training

Random Forest Classifier is chosen for its robustness and ability to handle imbalanced datasets. It is trained on the resampled dataset. Hyperparameters such as the number of trees, maximum depth, and minimum samples split are tuned using `GridSearchCV` with cross-validation. The best parameters are then used to retrain the final model.

6. Model Evaluation

The model is evaluated on the test set using multiple metrics:

- Accuracy
- Precision
- Recall
- F1-score
- AUC-ROC

7. Model Performance

Below are the performance results on the test data:

- **Accuracy:** 0.9994
- **Precision:** 0.8367
- **Recall:** 0.8367
- **F1-Score:** 0.8367

- **AUC-ROC Score:** 0.9740

Classification Report:

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     56864
           1       0.84      0.84      0.84        98

    accuracy                           1.00     56962
   macro avg       0.92      0.92      0.92     56962
weighted avg       1.00      1.00      1.00     56962
```

Confusion Matrix:

```
[[56848    16]
 [   16    82]]
```

8. Visualization

Class distribution before and after applying SMOTE is visualized using Seaborn's `countplot`. The ROC curve is plotted to assess the trade-off between true positive rate and false positive rate.

9. Conclusion

The project successfully demonstrates a practical implementation of fraud detection using machine learning. By applying SMOTE and tuning Random Forest hyperparameters, the model achieves high recall and good AUC-ROC scores. This approach can be further extended using ensemble methods like XGBoost or deep learning models for production-grade systems.

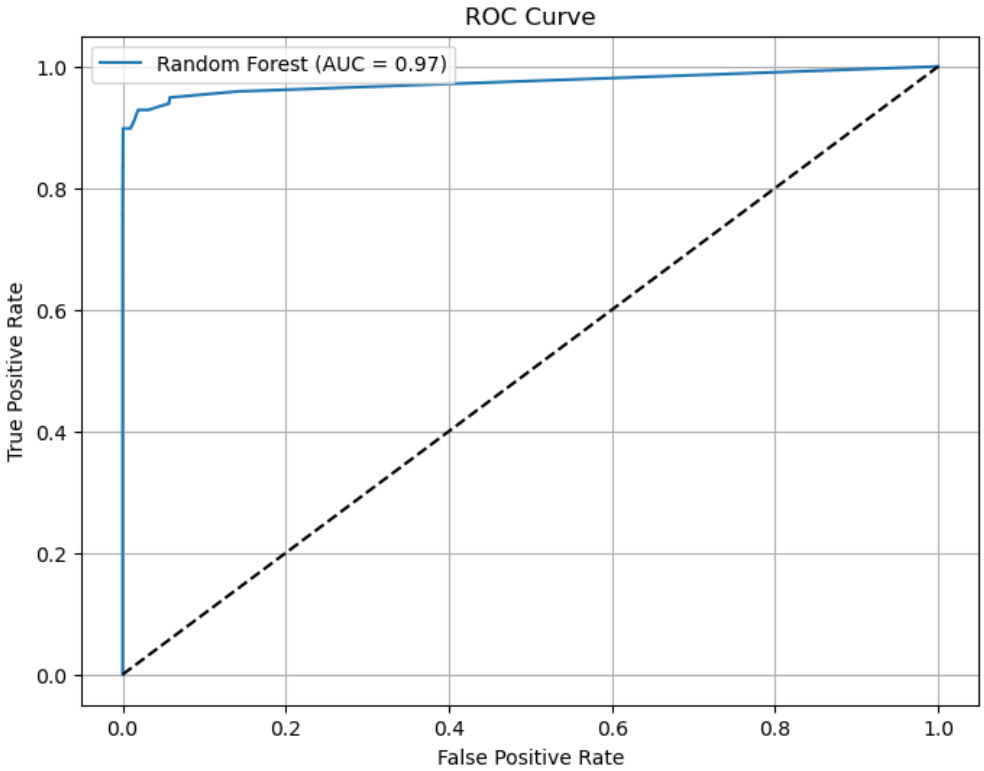

Figure 3: ROC Curve showing the model performance