Credit Card Fraud Detection - Model Development Report

# 1. Project Title

Credit card fraud detection using classification algorithms and SMOTE

# 2. Brief on the Project

The objective of this project is to develop a predictive model that identifies whether a transaction is fraudulent or legitimate based on operational data. Detecting credit card fraud is crucial for banks as it helps safeguard customer funds, enhance service quality, and improve overall operational efficiency.

The dataset consists solely of numerical features, which are the result of a Principal Component Analysis (PCA) transformation. Due to confidentiality constraints, the original features and detailed background information cannot be disclosed. The features **V1, V2, …, V28** represent the principal components derived from PCA, while **‘Time’** and **‘Amount’** are the only untransformed variables. The **‘Time’** feature indicates the number of seconds elapsed between a transaction and the first transaction in the dataset, whereas **‘Amount’** represents the transaction value and can be utilized for cost-sensitive learning.

To address the class imbalance and improve model performance, machine learning algorithms such as **Random Forest** and oversampling techniques like **SMOTE (Synthetic Minority Oversampling Technique)** are employed.

**3. Dataset Overview:**

Source: <https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud>

The dataset consists of 284807 records and 31 columns. Each row represents a credit card transaction with amount. The task is to use the provided variables to predict the likelihood of a credit card fraud transaction.

**Variable Descriptions:**

Unfortunately, due to confidentiality issues, we cannot provide the original features and more background information about the data. Features V1, V2, … V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'.

# 4. Deliverables of the Project

The deliverables of the project include:  
- A trained Random Forest model saved as a .pkl file.  
- A scaler object for feature normalization.  
- A Streamlit web application for interactive predictions.  
- A detailed report including EDA, model building steps, and evaluation results.

# 5. Questions the Model is Designed to Answer

- Will a model detecting a transaction fraud given the current operational conditions?  
- Which parameters are most influential in predicting fraud?  
- How well can we predict fraud transactions in advance to block the card?

# 6. Dataset Details

The following key features of the dataset contains machine operational data consider for the model building:  
It contains only numerical input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, we cannot provide the original features and more background information about the data. Features V1, V2, … V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

# 7. Exploratory Data Analysis and Preprocessing

EDA was performed to understand the data distribution, detect outliers, skewness, and examine relationships between variables. The dataset was split into training and testing sets using stratified sampling. SMOTE was applied to the training data to handle class imbalance in the 'Class' target variable.

# 8. Feature Engineering

The following steps were performed:  
- Scaling continuous features using Standard scaler.  
- Ensuring feature order consistency for model training and deployment.

# 9. Model Building

To identify the most suitable algorithm for predicting machine failures, multiple classification models were evaluated. The models used for training and testing include:

* Logistic Regression
* Decision Tree Classifier
* Random Forest Classifier
* K-Nearest Neighbors (KNN)
* Gradient Boosting Classifier
* XGBoost Classifier
* AdaBoost Classifier
* Bagging Classifier

The performance of these models was assessed using key evaluation metrics such as Accuracy, Precision, Recall, and F1-Score to ensure a comprehensive understanding of model effectiveness.

Among these, the Random Forest Classifier was selected as the final model because of its robustness, ability to handle complex non-linear relationships, and superior performance during evaluation. Hyperparameter tuning was not applied to optimize the model’s performance further because it is taking lot of time to run.

The final tuned Random Forest model was saved as random\_forest\_credit.pkl for deployment. To make the solution accessible to end-users, a Streamlit web application was developed, allowing real-time predictions based on user-provided parameters.

[Credit Card Fraud Detection · Streamlit](https://credit-card-fraud-detection-appl.streamlit.app/)

# .10. Model Evaluation

The model was evaluated using the following metrics:  
- Accuracy  
- Precision  
- Recall  
- F1-score  
The classification report showed high performance with precision and recall around 90%, indicating excellent predictive capability.

# 11. Findings and Expected Outcomes

The model successfully predicts credit card fraud detection with high accuracy. It highlights the most significant features affecting the fraud, enabling credit card fraud detection planning.

# 12. Deployment

Deployment through a Streamlit application allows users to input parameters and receive real-time predictions.

# [Credit Card Fraud Detection · Streamlit](https://credit-card-fraud-detection-appl.streamlit.app/)

# 13. Software and Tools Used

The following software and Python libraries were used for model development, evaluation, and deployment:

* **Python** – Programming language for data analysis and model development
* **ipykernel** – Kernel for running Python code in Jupyter notebooks
* **Scikit-learn** – Machine learning library for model building and evaluation
* **Imbalanced-learn (SMOTE)** – Handling imbalanced datasets using SMOTE and other techniques
* **Pandas** – Data manipulation and preprocessing
* **NumPy** – Numerical computations and array operations
* **Streamlit** – Building an interactive web application for model deployment
* **Matplotlib** – Data visualization and plotting
* **Seaborn** – Advanced statistical data visualization
* **XGBoost** – Extreme Gradient Boosting library for high-performance models
* **LightGBM** – Gradient boosting framework optimized for speed and performance
* **CatBoost** – Gradient boosting library with efficient handling of categorical features
* **Joblib** – Model serialization for saving and loading trained model

**14. Final Summary Before SMOTE and After SMOTE**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Train Accuracy | Test Accuracy | Recall | Precision | F1 Score |
| Random Forest | 0.999996 | 0.999596 | 0.785714 | 0.974684 | 0.870056 |
| Random Forest + SMOTE | 1.000000 | 0.999561 | 0.836735 | 0.901099 | 0.867725 |
| XGBoost | 1.000000 | 0.999579 | 0.806122 | 0.940476 | 0.868132 |
| XGBoost + SMOTE | 0.999991 | 0.999175 | 0.846939 | 0.721739 | 0.779343 |
| KNN | 0.999570 | 0.999526 | 0.775510 | 0.938272 | 0.849162 |
| KNN + SMOTE | 0.999468 | 0.998174 | 0.867347 | 0.482955 | 0.620438 |
| Decision Tree | 1.000000 | 0.999105 | 0.775510 | 0.723810 | 0.748768 |
| Decision Tree + SMOTE | 1.000000 | 0.997735 | 0.755102 | 0.413408 | 0.534296 |
| AdaBoost | 0.999127 | 0.999140 | 0.683673 | 0.788235 | 0.732240 |
| AdaBoost + SMOTE | 0.948257 | 0.964977 | 0.938776 | 0.044210 | 0.084442 |
| Logistic Regression | 0.999228 | 0.999140 | 0.602041 | 0.855072 | 0.706587 |
| Logistic Regression + SMOTE | 0.949141 | 0.974580 | 0.928571 | 0.059399 | 0.111656 |
| Gradient Boosting | 0.999184 | 0.998947 | 0.602041 | 0.737500 | 0.662921 |
| Gradient Boosting + SMOTE | 0.979083 | 0.988764 | 0.928571 | 0.125691 | 0.221411 |
| Deep Learning ANN + SMOTE | 0.9999 | 0.9994 | 0.8163 | 0.8247 | 0.8205 |

# 15. Project Milestones

1. Define the problem and understand business context  
2. Collect and clean data  
3. Perform EDA  
4. Apply feature engineering  
5. Build baseline models  
6. Apply SMOTE to handle imbalance  
7. Model evaluation and selection  
8. Develop Streamlit app  
9. Final report preparation and submission

# 16. Individual Details

Name: Akanksha Rawat  
Email: meetakanksharawat@gmail.com  
Phone: +91-9820399656