

Credit Card Fraud Detection

Submitted by

Name: Sagar Debnath
Roll No.: 11500120009
Regn. No.: 201150100110115

Name: Ramesh Das
Roll No.: 11500120010
Regn. No.: 201150100110114

Name: Shoham Sen
Roll No.: 11500120075
Regn. No.: 201150100110049

Name: Soudeep Ghosh
Roll No.: 11500120093
Regn. No.: 201150100110031

Under the supervision of

Prof. Mr. Subhasis Mallick

Academic Year: 2023 – 2024

Project report submitted in partial fulfillment of the requirements for
the degree of

Bachelor of technology in Computer Science and Engineering
at

B. P. Poddar Institute of Management & Technology

Affiliated to

Maulana Abul Kalam Azad University of Technology



Department of Computer Science & Engineering
B.P.Poddar Institute of Management & Technology
137, V.I.P Road. Poddar Vihar Kolkata - 700 052.

CERTIFICATE

This is to certify that the project work, entitled “Credit Card Fraud Detection” submitted by group of students is a bona-fide record of project done by

.....Sagar Debnath..... Roll number:11500120009

.....Ramesh Das..... Roll number: 11500120010

.....Shoham Sen..... Roll number: 11500120075

.....Soudeep Ghosh..... Roll number:11500120093 ,

has been prepared according to the regulation of the degree B. Tech in Computer Science & Engineering of the Maulana Abul Kalam Azad University of Technology, West Bengal. The candidates have partially fulfilled the requirements for the submission of the project work (PROJ-CS781).

(Signature of HOD)
Dept. of Computer Science & Engg.

(Signature of the Supervisor)
Dept. of Computer Science & Engg.

(Signature of External Examiner)

ACKNOWLEDGEMENT

It is a great pleasure for us to express our earnest and great appreciation to Mr. Subhasis Mallick sir, our project guide. We are very much grateful to him for his kind guidance, encouragement, valuable suggestions, innovative ideas, and supervision throughout this project work, without which the completion of the project work would have been difficult. We would like to express our thanks to the Head of the Department, Dr. Ananya Kanjilal maam for her active support. We also express our sincere thanks to all the teachers of the department for their precious help, encouragement, kind cooperation and suggestions throughout the development of the project work. We would like to express our gratitude to the library staff and laboratory staff for providing us with a congenial working environment.

(Full Signature of the Student(s))

Date:

Dept. of Computer Science & Engg.

B.P.Poddar Institute of Management & Technology

Table of Content

SL. No	Topic	Page No.
1	Departmental Mission, Vision, PEO, PO, PSO	5
2	Mapping with PO and PSO	8
3	Abstract	9
4	Activity chart	10
5	Introduction	11
6	Literature review	12
7	Theory	13
8	Methodology	14
9	Software requirements	22
10	Used system/ software	22
11	Mathematical Formulation	22
12	Results & Discussions	26
13	Limitations	28
14	Future plan	28
15	References	29

DEPARTMENTAL MISSION

Enrich students with sound knowledge in fundamentals and cutting-edge technologies of Computer Science and Engineering to excel globally in challenging roles in industries and academics.

Emphasize quality teaching, learning and research to encourage creative thoughts through application of professional knowledge and skill.

Inspire leadership and entrepreneurship skills in evolving areas of Computer Science and Engineering with social and environmental awareness.

Instill moral and ethical values to attain the highest level of accomplishment and personal growth.

DEPARTMENTAL VISION

Developing competent professionals in Computer Science and Engineering, who can adapt to constantly evolving technologies for addressing industrial and social needs through continuous learning.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

Graduates of Computer Science and Engineering program will have good knowledge in the core concepts of systems, software and tools for analyzing problems and designing solutions addressing the dynamic requirements of the industry and society, while employed in industries or work as entrepreneurs.

Graduates of Computer Science and Engineering program will opt for higher education and research in emerging fields of Computer Science & Engineering towards building a sustainable world.

Graduates of Computer Science and Engineering will have leadership skills, communication skills, ethical and moral values, team spirit and professionalism.

PROGRAM OUTCOMES (POs)

PO1:Engineering Knowledge: Apply the knowledge of Mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2:Problem Analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3:Design/Development of Solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4:Conduct Investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5:Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6:Engineer and Society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7:Environment and Sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8:Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9:Individual and Team Work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10:Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11:Project Management and Finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12:Life-long Learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSO)

Students will have proficiency in emerging domains like artificial intelligence, data science and distributed computing to develop solutions through innovative projects and research.

Students will have capabilities to work in synergized teams to cater to the dynamic needs of the industry and society.

Program Outcomes (POs):

1.Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2.Problem analysis: Identify, formulate, research literature, and analyses complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3.Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4.Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5.Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6.The engineer and society: Apply to reason informed by the contextual knowledge to health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7.Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and teamwork: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. Lifelong learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PO/PSO mapping:

PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PO 12	PSO1	PSO12
3	3	3	2	3	2	1	2	3	3	-	1	3	3

Abstract:

In recent years, the advancements of e-commerce and e-payment systems have resulted in a rise in financial fraud cases, such as credit card fraud. It is therefore crucial to implement mechanisms that can detect credit card fraud. Here, comes the need for a system that can track the pattern of all the transactions and if any pattern is abnormal then the transaction should be aborted. Today, we have many machine learning algorithms that can help us classify abnormal transactions. But the features of credit card frauds play an important role when machine learning is used for credit card fraud detection, and they must be chosen properly. To validate the performance, we will use a dataset that contains transactions made by credit cards in September 2013 by European cardholders [6].

Activity chart:

Task	16th-31st Jul	1st-15th Aug	16th-31st Aug	1st-30th Sep	1st-31st Oct	1st-10th Nov	11th-30th Nov	1st-20th Dec
Project definition	↔							
Literature Review		↔						
Data Collection		↔						
Familiarization with Machine learning		↔	↔	↔				
Familiarization with Python				↔	↔			
Data cleaning and preprocessing					↔			
Model Development and Training					↔	↔		
Feature Selection and Optimization							↔	
System Integration and Testing								↔
Report writing and project presentation								↔

Introduction:

In today's digital age, where electronic transactions have become the norm, credit card fraud has become a significant concern for both financial institutions and consumers. Detecting fraudulent activities in real-time is crucial to prevent financial losses and protect sensitive information. This has led to the development of advanced techniques and technologies for credit card fraud detection. In this project, we aim to implement a robust and efficient credit card fraud detection system using machine learning algorithms.

The objective of this undertaking is to identify fraudulent transactions conducted through credit cards utilizing machine learning methodologies. The primary goal is to thwart fraudsters from illicitly accessing and exploiting customers' accounts. With the escalating global prevalence of credit card fraud, it becomes imperative to implement measures to curb fraudulent activities. Imposing restrictions on such activities can yield positive outcomes for customers, ensuring the recovery and restoration of their funds. This, in turn, safeguards customers from unwarranted charges for items or services they did not authorize. The core focus of this project is to detect fraudulent transactions through the application of three distinct machine learning techniques: Logistic Regression and Random Forest Classifier. These models will be applied to a dataset comprising credit card transactions for effective fraud detection.

Literature review:

Gupta and his team developed an automated model to detect economically related fraudulent instances, with a focus on credit card transactions. Among the various ML techniques used, Naïve Bayes performed exceptionally well, with an accuracy of 80.4% and an area under the curve of 96.3% (Gupta et al., 2021). [1]

Mailini and Pushpa proposed using KNN and anomaly detection to detect credit card fraud, and the authors, after completing the sample data, found that the method was KNN as the best way to detect and identify flaws in Target. the best. to identify fake memories. Credit card verification requires less computation and memory for suspicious detection and works faster and better on large online databases. However, his studies and results show that KNN is accurate and effective (Malini & Pushpa, 2017). [2]

Varmedja's team uses various machine learning algorithms in their paper, such as logistic regression, multilayer perceptron, random forest and pure Bayesian. Because the data was inconsistent, Varmedja and his team used SMOTE techniques for oversampling, feature selection, and further partitioning of data into training and test datasets. The model with the best score during the test is Random Forest with a score of 99.96%, not much different, with Multilayer Perceptron in second place with a score of 99.93% and Naive Bayes in third place with a score of 99.23% according to Logistic Regression with 97.46% (Varmedja et al., 2019). [3]

Kiran and his team briefly present the K-Neighbor Neighbor credit card fraud detection method (NBKNN) enhanced with Naive Bayesian (NB). Experimental results show the difference in performance of each classifier in the same dataset. Naive Bayes outperforms K Neighbors as it is 95% accurate compared to 90% for KNN (Kiran et al., 2018). [4]

Itoo and his team's work used three different machine learning methods, the first is logistic regression, the second is Naive Bayes, and the last is the K-Best approximation. Itoo and his team documented and compared their work with python. Logistic Regression has an accuracy of 91.2%, Naive Bayes has an accuracy of 85.4%, and the K-Nearest is the closest with an accuracy of 66.9% (Ito et al., 2020). [5]

Theory:

'Fraud' in credit card transactions is unauthorized and unwanted usage of an account by someone other than the owner of that account. Necessary prevention measures can be taken to stop this abuse and the behaviour of such fraudulent practices can be studied to minimize it and protect against similar occurrences in the future. In other words, Credit Card Fraud can be defined as a case where a person uses someone else's credit card for personal reasons while the owner and the card issuing authorities are unaware of the fact that the card is being used. Fraud detection involves monitoring the activities of populations of users in order to estimate, perceive or avoid objectionable behaviour, which consist of fraud, intrusion, and defaulting. This is a very relevant problem that demands the attention of communities such as machine learning and data science where the solution to this problem can be automated. This problem is particularly challenging from the perspective of learning, as it is characterised by various factors such as class imbalance. The number of valid transactions far outnumber fraudulent ones. Also, the transaction patterns often change their statistical properties over the course of time.

These are not the only challenges in the implementation of a real-world fraud detection system, however. In real world examples, the massive stream of payment requests is quickly scanned by automatic tools that determine which transactions to authorize. Machine learning algorithms are employed to analyse all the authorized transactions and report the suspicious ones. These reports are investigated by professionals who contact the cardholders to confirm if the transaction was genuine or fraudulent. The investigators provide feedback to the automated system which is used to train and update the algorithm to eventually improve the fraud-detection performance over time.

Fraud detection methods are continuously developed to defend criminals in adapting to their fraudulent strategies. These frauds are classified as:

- Credit Card Frauds: Online and Offline
- Card Theft
- Account Bankruptcy
- Device Intrusion
- Application Fraud
- Counterfeit Card
- Telecommunication Fraud

Some of the currently used approaches to detection of such fraud are:

- Artificial Neural Network
- Fuzzy Logic
- Genetic Algorithm
- Logistic Regression

Methodology:

Collecting Dataset:

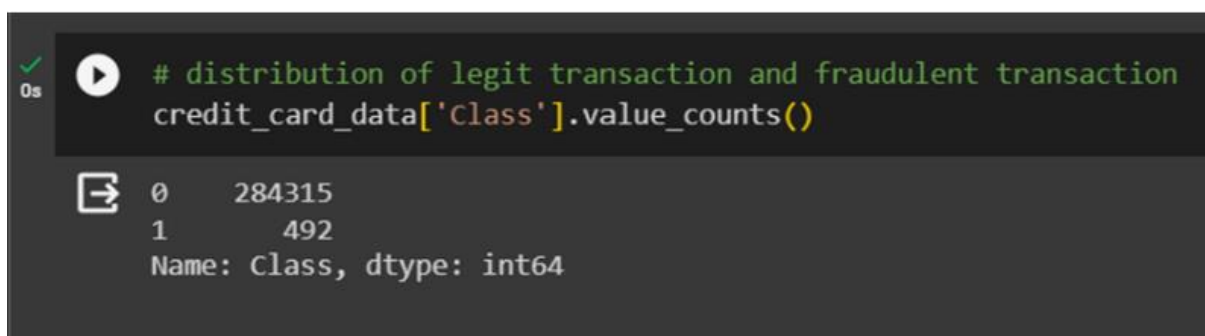
We collected the dataset from Kaggle. The dataset contains transactions made by credit cards in September 2013 by European cardholders.

This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

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. The feature 'Amount' is the transaction Amount, this feature can be used for example-dependant cost-sensitive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise. Given the class imbalance ratio, we recommend measuring the accuracy using the Area Under the Precision-Recall Curve (AUPRC). Confusion matrix accuracy is not meaningful for unbalanced classification.[6]

Data Analysis:

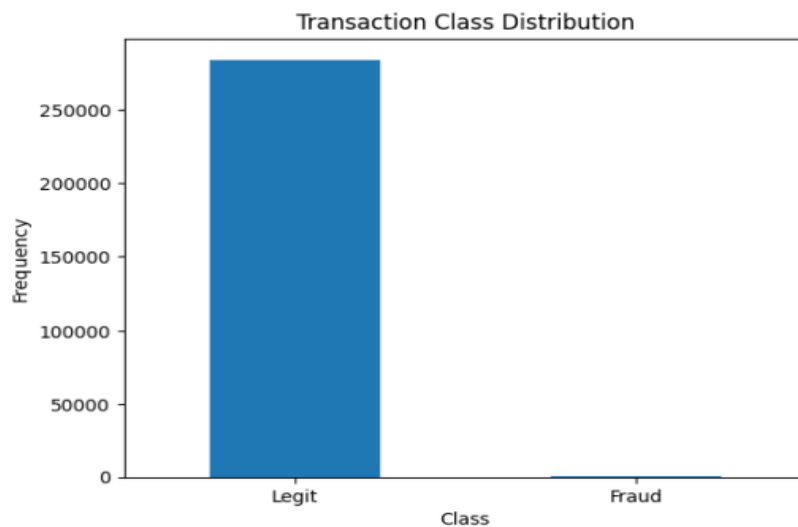
Dealing with highly unbalanced data is a common challenge in machine learning, especially in scenarios where one class (in this case, fraud transactions) is significantly underrepresented compared to the other class (non-fraud transactions). If left unaddressed, machine learning models trained on imbalanced data can be biased towards the majority class, leading to poor performance in detecting the minority class (frauds, in this case).



```
# distribution of legit transaction and fraudulent transaction
credit_card_data['Class'].value_counts()
```

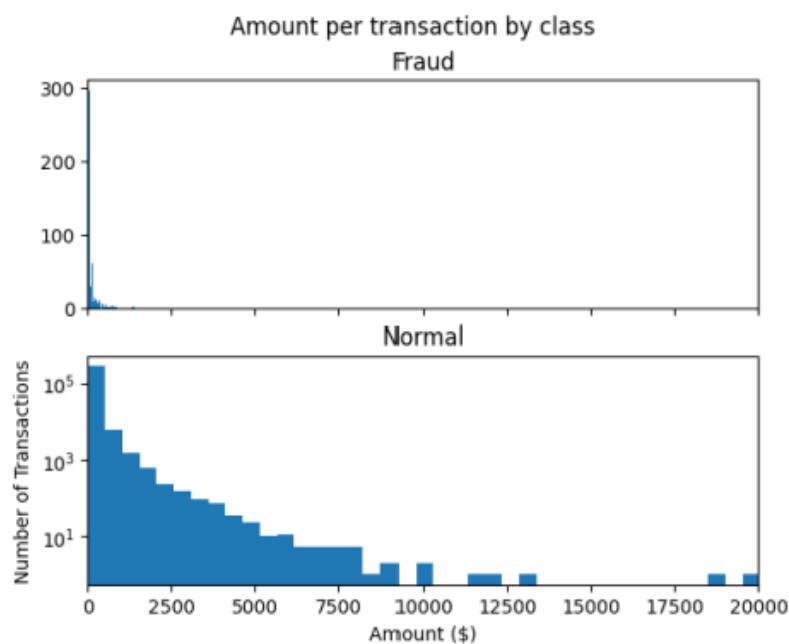
Class	Count
0	284315
1	492

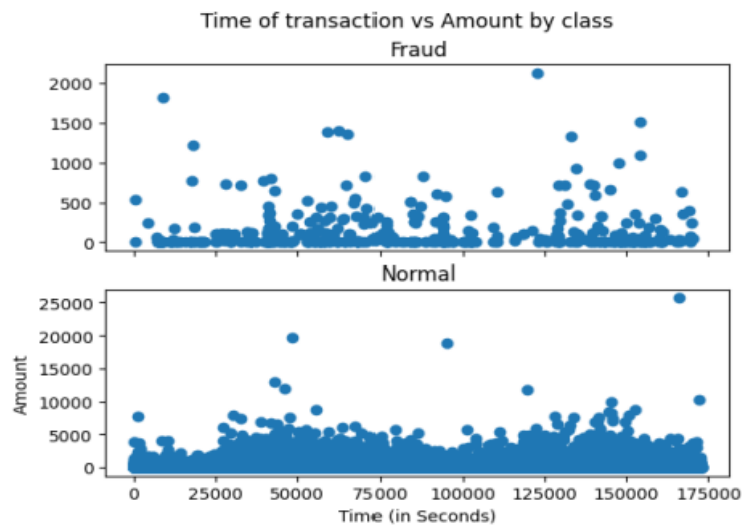
Name: Class, dtype: int64



In our dataset, transactions are classified into two categories: 0 for legitimate transactions and 1 for fraudulent transactions. There are 284,315 instances of legitimate transactions and only 492 instances of fraudulent transactions, highlighting a significant imbalance in the data distribution.

The graph illustrates the distribution of transaction amounts categorized by class, specifically focusing on fraudulent transactions. By visualizing this data, we gain valuable insights into the patterns and disparities between the amounts involved in fraudulent and non-fraudulent activities.





Separating the data and analyse the statistical properties :

```
✓ [12] # separating the data for analysis
0s legit = credit_card_data[credit_card_data.Class == 0]
    fraud = credit_card_data[credit_card_data.Class == 1]
```

```
✓ [13] print(legit.shape)
0s      print(fraud.shape)

      (284315, 31)
      (492, 31)
```

```
✓ [14] # statistical measures of the data
0s legit.Amount.describe()
```

count	284315.000000
mean	88.291022
std	250.105092
min	0.000000
25%	5.650000
50%	22.000000
75%	77.050000
max	25691.160000
Name: Amount, dtype: float64	

```
✓ [15] fraud.Amount.describe()
0s
```

count	492.000000
mean	122.211321
std	256.683288
min	0.000000
25%	1.000000
50%	9.250000
75%	105.890000
max	2125.870000
Name: Amount, dtype: float64	

This graph provides a clear distinction between two classes of transactions: fraudulent and non-fraudulent. By comparing the histograms representing these classes, we can discern how transaction amounts vary significantly between the two categories.

The unbalanced nature of the dataset is evident in this graph, with the majority of transactions falling within the non-fraudulent category.

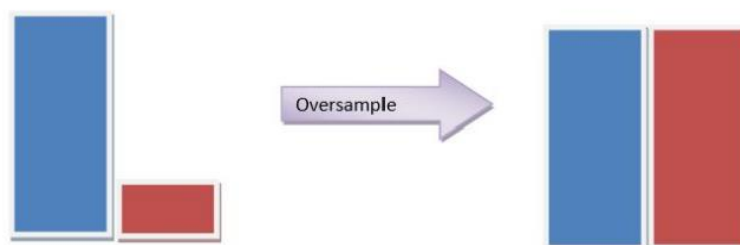
Data Pre-Processing:

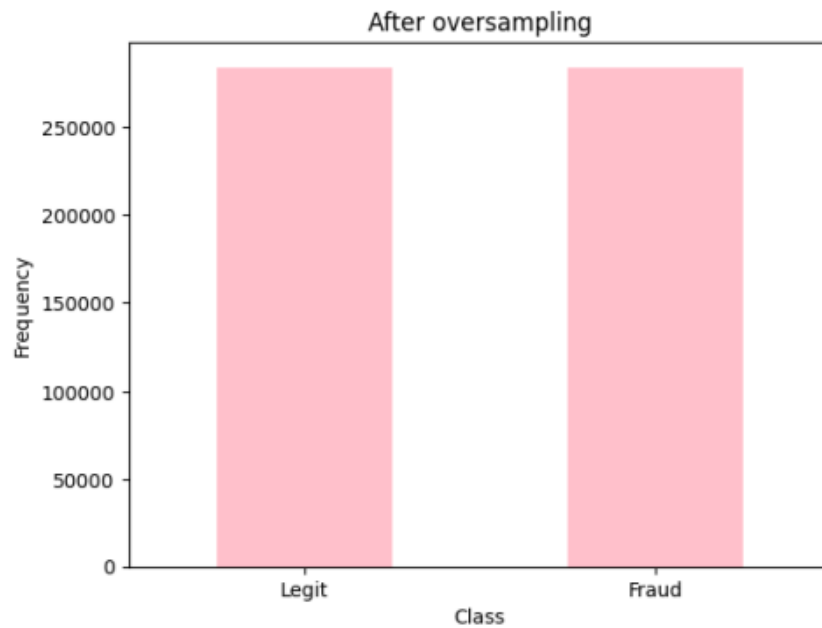
It is vital to identify the minority classes correctly. So model should not be biased to detect only the majority class but should give equal weight or importance towards the minority class too. Here we used Resampling techniques which can deal with this problem. There is no right method or wrong method in this, different techniques work well with different problems.

Resampling to handle Unbalanced datasets: Resampling Method is a statical method that is used to generate new data points in the dataset by randomly picking data points from the existing dataset. It helps in creating new synthetic datasets for training machine learning models and to estimate the properties of a dataset when the dataset is unknown, difficult to estimate, or when the sample size of the dataset is small.

- **Random Over Sampling:** It aims to balance class distribution by randomly increasing minority class examples by replicating them.

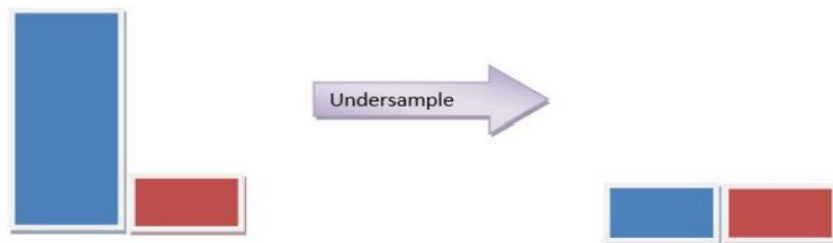
```
df_oversampled.Class.value_counts()
1      284315
0      284315
Name: Class, dtype: int64
```





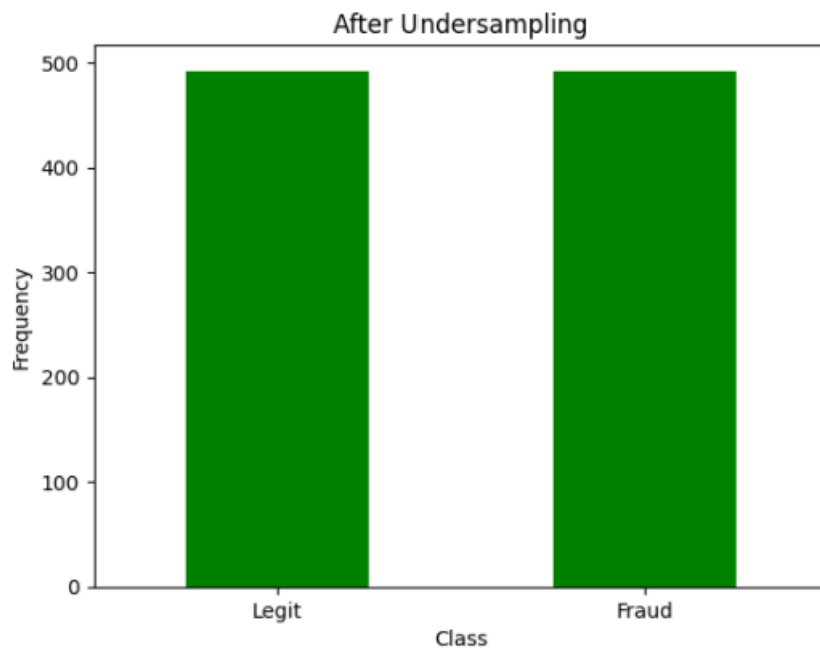
- **Random Under-sampling:**

It aims to balance class distribution by randomly eliminating majority class example.



```
df_undersampled.Class.value_counts()
```

```
0.0    492  
1.0    492  
Name: Class, dtype: int64
```



- **SMOTE (Synthetic Minority Oversampling Technique):**

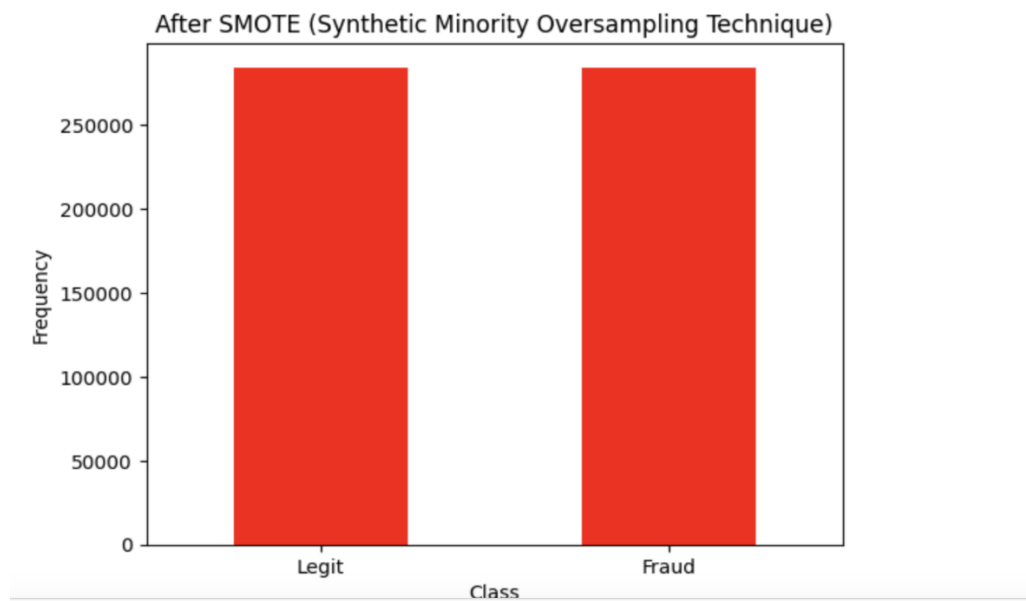
Synthetic Minority Oversampling Technique or SMOTE is another technique to oversample the minority class. Simply adding duplicate records of minority class often don't add any new information to the model. In SMOTE new instances are synthesized from the existing data. If we explain it in simple words, SMOTE looks into minority class instances and use k nearest neighbor to select a random nearest neighbor, and a synthetic instance is created randomly in feature space.

```
# SMOTE(Synthetic Minority Oversampling Technique)
from imblearn.over_sampling import SMOTE
# Resampling the minority class. The strategy can be changed as required.
sm = SMOTE(sampling_strategy='minority', random_state=42)
# Fit the model to generate the data.
X_smote, y_smote = sm.fit_resample(x, y)
X_smote.shape

(568630, 30)

[ ] df_smote = pd.concat([pd.DataFrame(X_smote), pd.DataFrame(y_smote)], axis=1)
df_smote.Class.value_counts()

0    284315
1    284315
Name: Class, dtype: int64
```



Model Training & Testing Using Logistic Regression:

During the model training phase, the logistic regression algorithm is utilized to create a predictive model. Logistic regression is a widely-used binary classification algorithm suitable for this task. The training dataset consists of historical credit card transactions, including both genuine and fraudulent examples. The algorithm learns patterns and relationships within the data, aiming to differentiate between normal and fraudulent activities.

The logistic regression model is trained on the preprocessed dataset, adjusting its parameters to minimize the error in predicting fraudulent transactions.

Cross-validation may be employed to assess the model's generalization performance and mitigate overfitting.

Once the logistic regression model is trained, it is evaluated on a separate testing dataset to assess its performance in real-world scenarios. This testing phase involves:

Performance metrics such as accuracy, precision, recall, and F1 score are calculated to quantify the model's effectiveness in identifying fraudulent transactions while minimizing false positives.

A confusion matrix is generated to provide a detailed breakdown of the model's true positives, true negatives, false positives, and false negatives.

The decision threshold of the logistic regression model may be adjusted to balance sensitivity and specificity, depending on the business requirements and the cost associated with false positives and false negatives.

If the model meets the desired performance criteria, it can be deployed for real-time credit card fraud detection in operational environments.

Further things are in Result & Discussion .

Model Training & Testing Using Random Forest Classifier:

Random Forest is a powerful ensemble learning method that combines multiple decision trees to enhance predictive accuracy and generalization.

Relevant features are selected or engineered to optimize the Random Forest model's ability to distinguish between legitimate and fraudulent transactions.

The algorithm can handle both numerical and categorical features effectively.

Random Forest builds multiple decision trees during the training process. Each tree is constructed on a random subset of the data, and features are randomly selected for each split, introducing diversity and reducing the risk of overfitting.

The model's hyperparameters, such as the number of trees in the forest, maximum depth of trees, and minimum samples per leaf, are tuned to optimize performance and prevent overfitting.

The Random Forest model is evaluated on a separate test dataset using metrics like accuracy, precision, recall, F1 score, and area under the Receiver Operating Characteristic (ROC) curve.

These metrics provide a comprehensive understanding of the model's ability to correctly identify fraud while minimizing false positives.

A confusion matrix is generated to examine the model's true positives, true negatives, false positives, and false negatives, providing insights into its performance across different classes.

Random Forest provides a natural way to assess feature importance. Understanding which features contribute the most to the model's predictions can offer valuable insights into the characteristics of fraudulent transactions.

Similar to logistic regression, the decision threshold of the Random Forest model can be adjusted to achieve the desired balance between sensitivity and specificity, depending on the business requirements.

If the Random Forest model meets the performance criteria, it can be deployed for real-time credit card fraud detection, offering a scalable and effective solution for financial institutions.

Further things are in Result & Discussion

Software/hardware requirements:

Compatible with any OS like:

- Windows
- Linux
- Macintosh
- Android Mobile Operating System
- Solaris Operating System

Used system/ software:

- Google Collabs
- Pycharm
- Sublime Text Editor
- Jupyter
- Spyder

Mathematical Formulation :

a) Logistic Regression: Logistic regression is based on the sigmoid function and its result is from zero to one. The sigmoid for logistic regression is

$$S_i = \sum_{j=1}^p x_{ij}\beta_j = x_i^T \beta$$

The function of the logistic regression is

$$f(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

Logistic regression is similar to linear regression. It is a particular case of the Generalized Linear Model. The main difference between these two regressions is whether the outcome Y is binary. The linear regression allows the outcome to exceed the zero to one range, while the logistic regression is more applicable to the outcome limited in zero to one. Another difference is that logistic regression

calculates the maximum likelihood equations from the probability distribution of the dependent variables.

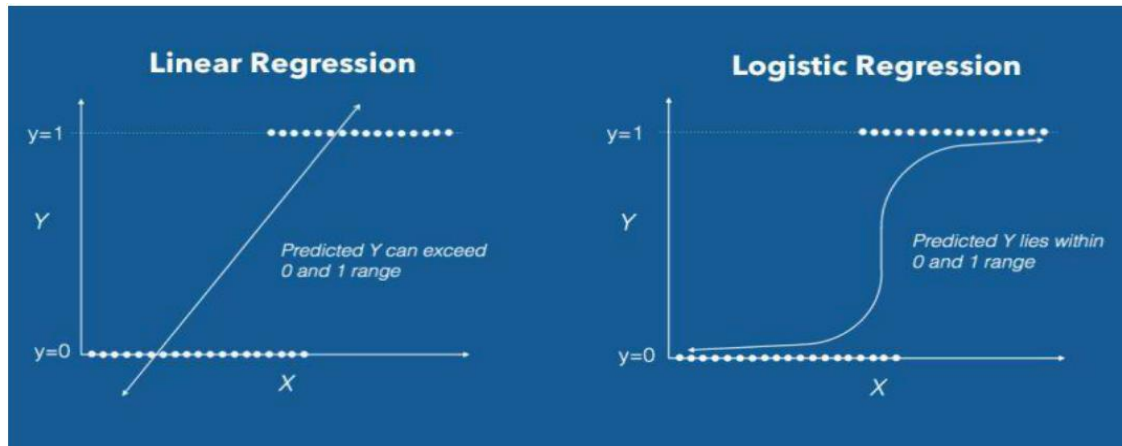


Fig: Regression Comparison

Working:

The logistic regression model transforms the linear regression function continuous value output into categorical value output using a sigmoid function, which maps any real-valued set of independent variables input into a value between 0 and 1. This function is known as the logistic function.

Let the independent input features be

$$X = \begin{bmatrix} x_{11} & \dots & x_{1m} \\ x_{21} & \dots & x_{2m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \dots & x_{nm} \end{bmatrix}$$

and the dependent variable is Y having only binary value i.e. 0 or 1.

$$Y = \begin{cases} 0 & \text{if Class 1} \\ 1 & \text{if Class 2} \end{cases}$$

then apply the multi-linear function to the input variables X

$$z = (\sum_{i=1}^n w_i x_i) + b$$

Here x_i is the i th observation of X, $w_i = [w_1, w_2, w_3, \dots, w_m]$ is the weights or Coefficient, and b is the bias term also known as intercept. simply this can be represented as the dot product of weight and bias.

$$z = w \cdot X + b$$

b) Random Forest Classifiers:

The random forest algorithm is a powerful and popular ensemble learning technique used for classification and regression tasks. Its strength lies in its assembly of multiple decision trees to make predictions. Each decision tree in the forest is trained on a randomly selected subset of the data and features, ensuring diversity and reducing overfitting. During prediction, each tree casts its vote, and the final output is determined by the majority vote. This approach enhances the algorithm's accuracy, stability, and ability to handle large and complex datasets. Moreover, random forests can handle both categorical and continuous data and provide valuable insights into feature importance, making them well-suited for various applications, including credit card fraud detection, medical diagnosis, and more. Random Forest is a popular machine learning algorithm that combines the power of multiple decision trees to make more accurate predictions. It creates a forest of trees during training, where each tree is built using a random subset of the training data and a random set of features. This randomness helps to reduce overfitting and improves generalization. During prediction, each tree in the forest votes on the outcome, and the final prediction is determined by the majority vote. Random Forest is widely used for the tasks. Like classification and regression in various fields due to its robustness and ability to handle large datasets. It explains how the algorithm aggregates multiple decision trees to make more accurate and robust predictions.

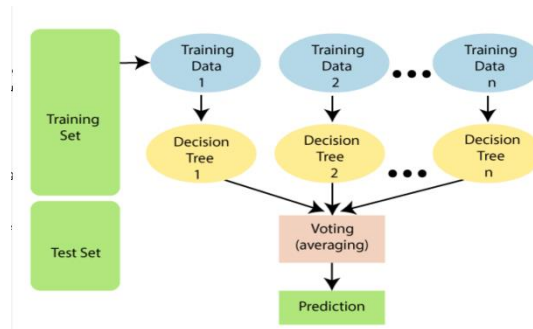


fig: Random Forest Classifier

Challenges and Limitations:

Several challenges of credit card fraud detection using Random forest Algorithm are as follows.

Challenges:

Imbalanced Datasets: Dealing with imbalanced datasets in various domains, where one class is significantly more prevalent than the other, poses challenges for classification models and can lead to biased predictions.

Data Privacy and Security: Maintaining the privacy and security of sensitive data while still allowing effective analysis

can be a challenging task, especially in fields like healthcare and finance.

Interpretable AI: Many modern AI models, such as deep learning neural networks, lack interpretability, making it difficult to understand and explain their decision-making processes, which is crucial for building trust and compliance.

Adversarial Attacks: Sophisticated adversaries can attempt to exploit vulnerabilities in AI systems through adversarial attacks, making the models susceptible to manipulations and leading to incorrect predictions.

Generalization: Ensuring that AI models generalize well to new and unseen data is a challenge, especially when the training data does not fully represent all possible scenarios.

Implementation are as follows :

- Data Pre-processing step
- Fitting the Random forest algorithm to the Training set
- Predicting the test result
- Test accuracy of the result (Creation of Confusion matrix)
- Visualizing the test set result.

Results & Discussions:

In this credit card fraud detection project, we mainly used two machine learning models: Logistic Regression and Random Forest. The results of the model evaluations, based on accuracy, are as follows:

Logistic Regression:

We have performed Logistic regression for two pre-processed data sets. For oversampled dataset we got accuracy on training data 91.8% and testing data we got 91.6%.

```
[ ] print('Accuracy on Training data : ', training_oversampled_data_accuracy)

Accuracy on Training data : 0.9187345022246453
```

```
[ ] print('Accuracy score on Test Data : ', test_oversampled_data_accuracy)

Accuracy score on Test Data : 0.9162900304240016
```

For under sampled dataset, we got accuracy on training data 92.5% and testing data we got 89.8%.

```
[ ] print('Accuracy on Training data : ', training_data_accuracy)

Accuracy on Training data : 0.9250317662007624
```

```
▶ print('Accuracy score on Test Data : ', test_data_accuracy)

⇒ Accuracy score on Test Data : 0.8984771573604061
```

Characteristics:

Interpretability: High, as the model provides coefficients indicating the impact of each feature on the prediction.

Computational Intensity: Low, making it computationally efficient and easy to deploy.

Handling Non-linearity: Limited compared to Random Forest, as it assumes a linear relationship between features and the log-odds of the response variable.

Robustness to Outliers: Moderately sensitive to outliers due to the linear nature of the decision boundary.

Random Forest:

Using Random Forest algorithm we got the accuracy of 93.4%.

```
acc = accuracy_score(yTest, yPred)
print("The accuracy of Test data {}".format(acc))
```



```
The model used is Random Forest classifier
The accuracy of Test data 0.934010152284264
```

Characteristics:

Interpretability: Lower than Logistic Regression due to the ensemble nature, considered as a "black box" model.

Computational Intensity: Higher compared to Logistic Regression, especially for larger datasets, but provides better generalization.

Handling Non-linearity: Better than Logistic Regression, as Random Forest can capture complex non-linear patterns through decision trees.

Robustness to Outliers: More robust to outliers due to the aggregation of predictions from multiple trees.

Model	Accuracy
Logistic Regression	91.6%
Random Forest	93.4%

Comparison Summary:

To compare all models, it is clear that the Random Forest achieved a slightly higher accuracy (93%) compared to Logistic Regression (91.6%).

Logistic Regression is more interpretable and computationally efficient, making it suitable for scenarios where interpretability and deployment ease are crucial.

Random Forest excels in capturing non-linear patterns, is more robust to outliers, and handles imbalanced datasets better.

The choice between the two models depends on factors such as the nature of the dataset, the need for interpretability, computational resources, and the specific goals of the credit card fraud detection application.

In conclusion, both models demonstrated strong performance, but the decision on which model to deploy should be made based on the specific requirements and trade-offs relevant to the credit card fraud detection system.

Limitations:

First of all, although the number of samples is as many as one million, compared to the billions of credit card transactions that occur worldwide every day, this sample only accounts for less than one-thousandth of the global daily number. Thus, the sample size is still not large enough. Also, the background of the dataset needs to be adequate. The data uploader should provide a detailed introduction, including but not limited to which country and region does these records originated from, how they were obtained, etc.

Secondly, the dataset is relatively clean when it is downloaded. For instance, there are no incomplete values such as zero or NA. Moreover, categorical binary variables have been converted into numerical variables to facilitate analysis. Almost all independent variables are particularly significant. Mining or getting source data from real life will definitely be more complicated. In addition, there may have more variables to be considered. For example, the ratio to the mean or mode purchase price may also be significant. And the store type of the transaction may be necessary, like a small store on the street may be more prone to fraud than a large chain store. Meanwhile, different countries may have different credit card fraud rates.

Finally, more models, such as the random forest, can be applied during analysis.

Future plan:

- One of our Future goals is to increase the accuracy score of our respected models.
- We wish to study the other different Machine Learning algorithms and implement the Fraud detection model based on individual Models and compare the accuracy of each model to find the best result
- We aim to create a robust system which can detect and inform the fraud instantly to its customer.

References:

- [1] Gupta, A., Lohani, M. C., & Manchanda, M. (2021). Financial fraud detection using naive Bayes algorithm in highly imbalance data set. *Journal of Discrete Mathematical Sciences and Cryptography*, 24(5), 1559–1572. <https://doi.org/10.1080/09720529.2021.1969733>
- [2] Malini, N., & Pushpa, M. (2017). Analysis on credit card fraud identification techniques based on KNN and outlier detection. 2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB). <https://doi.org/10.1109/aeicb.2017.7972424>
- [3] Varmedja, D., Karanovic, M., Sladojevic, S., Arsenovic, M., & Anderla, A. (2019). Credit Card Fraud Detection - machine learning methods. 2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH). <https://doi.org/10.1109/infoteh.2019.8717766>
- [4] Kiran, S., Guru, J., Kumar, R., Kumar, N., Katariya, D., & Sharma, M. (2018). Credit card fraud detection using Naïve Bayes model based and KNN classifier. *International Journal Of Advance Research, Ideas And Innovations In Technology*, 4(3).
- [5] Itoo, F., Meenakshi, & Singh, S. (2020). Comparison and analysis of logistic regression, Naïve Bayes and Knn Machine Learning Algorithms for credit card fraud detection. *International Journal of Information Technology*, 13(4), 1503–1511. <https://doi.org/10.1007/s41870-020-00430-y>
- [6] <https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud>
- [7] <https://www.geeksforgeeks.org/ml-credit-card-fraud-detection/>
- [8] <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-022-00573-8>

