**CREDIT CARD FRUAD DETECTION**

**ABSTRACT**

Credit card fraud detection is presently the most frequently occurring problem in the present world. This is due to the rise in both online transactions and e-commerce platforms. Credit card fraud generally happens when the card was stolen for any of the unauthorized purposes or even when the fraudster uses the credit card information for his use. In the present world, we are facing a lot of credit card problems. To detect the fraudulent activities the credit card fraud detection system was introduced. This project aims to focus mainly on Data Mining Models. The Data Mining models used are Decision Tree, SVM, Random Forest, KNN, XGBClassifier, and Naive Bayes. The models are evaluated using accuracy, classification report, and AUC score, both before and after data balancing. Models are compared and the model that has the greatest accuracy, precision, recall, F1-score, AUC score is considered as the best algorithm that is used to detect the fraud.

**1. INTRODUCTION:**

Detecting fraud is a challenge for many businesses, as they often only discover fraudulent activities after experiencing significant financial losses. Such activities cost companies billions of dollars globally each year. In addition, maintaining customer trust is crucial for any business, even more important than securing a competitive position in the market. Failing to identify fraudulent activities can result in a loss of customer trust and ultimately lead to customer churn. To mitigate this risk, credit card fraud classification algorithms will be utilized to identify fraudulent activities before they escalate into a major issue for credit card companies.

**1.1 Goal:**

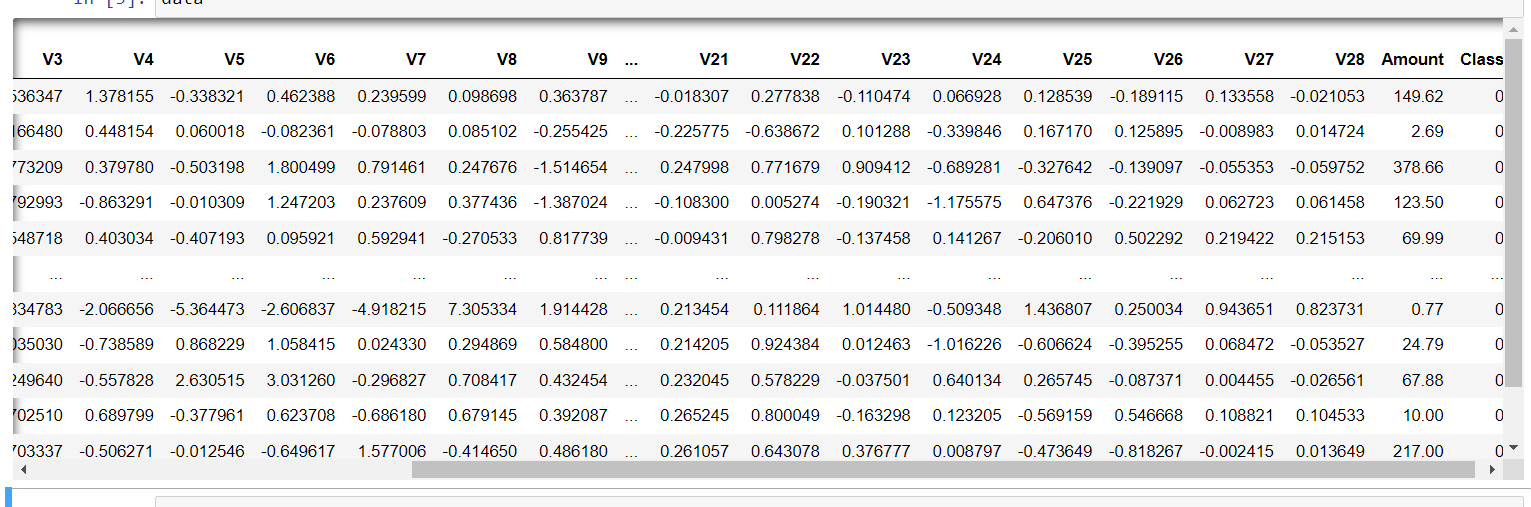
* Application is Credit card fraud detection using data mining models

.

**1.2 DATASET:**

* Data set we used is Credit Card Transactions Fraud Detection. This is found in Kaggle and can be accessed from below link.
  + <https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud>
* 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.
* It has 30 features; Class is the target variable with 0’s and 1’s representing fraud and non - fraudulent transactions. Most of the features are preserved and presented as V1, V2..etc. These are transformed using PCA (Principal Component Analysis)
* From our initial analysis of data, we see there is 0.17% of Fraudulent data of the total transactions, which shows that data is unbalanced. Therefore, we will have a considerable amount of work to do when balancing the data.

Data representation:



1.3 **Challenging Part**:

* Credit card fraud detection is a challenging problem due to the large volume of transactions that occur daily and the sophisticated methods used by fraudsters to evade detection. Fraudulent transactions can occur in a variety of ways, including stolen credit card information, identity theft, and account takeover, making it difficult to identify patterns of fraud.

**2. Solution**

**2.1 Existing system**:

Currently we have several data mining techniques being used for Credit card fraud de detection. Few of them are Random Forest Classifier, Logistic Regression, MLP Neural Network Classifier.

**2.2 Proposed System**:

We have implemented 6 data mining models such as Decision Tree Classifier, Random Forest Classifier, Logistic Regression, Support Vector Machine, Naive Bayes Classifier and XGBClassifier. These models are evaluated before and after data balancing. Models are compared and the model that has the greatest accuracy, precision, recall, F1-score, AUC score is considered as the best algorithm that is used to detect the fraud.

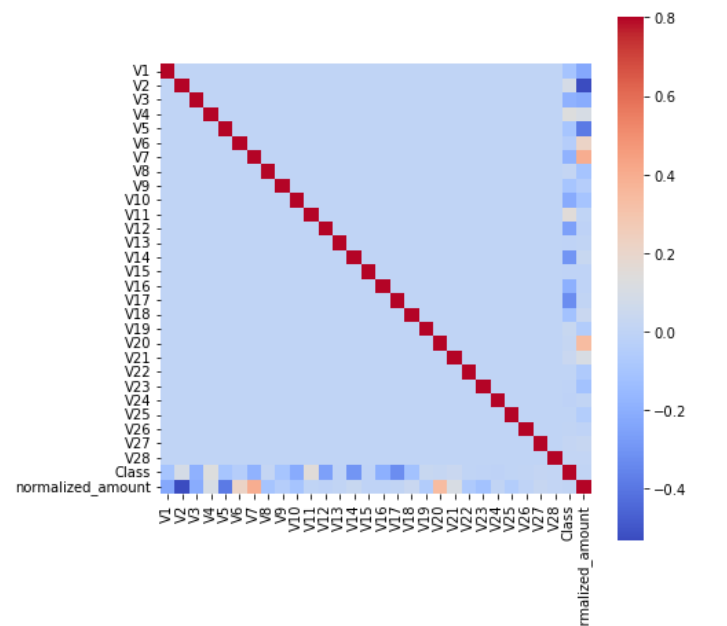
**2.3. Steps Involved:**

* Data pre-processing
* Build the Models

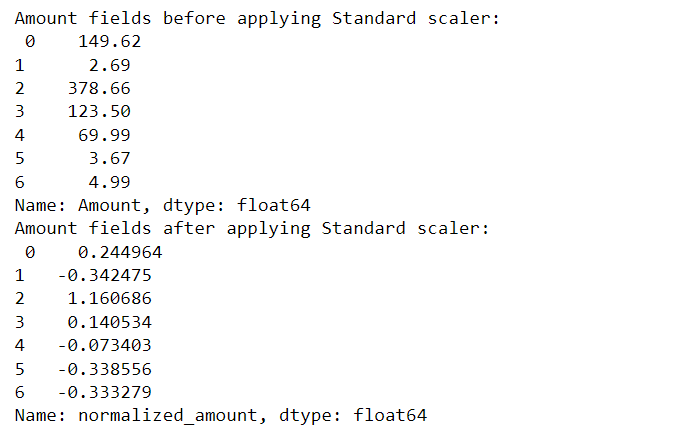
**2.3.1: Data pre-processing:**

* First step as part of data pre-processing is to check for Null values. Checking for null values is an important step in data pre-processing before applying data mining algorithms. Null values, also known as missing values, can cause issues with data analysis, accuracy, and model performance. We have used pandas library to check Null values.
* Checked for any duplicate data and removed the data.
* **Perform Correlation**:

Correlation measures the strength and direction of the relationship between variables. A positive correlation means that as one variable increases, the other variable also tends to increase. A negative correlation means that as one variable increases, the other variable tends to decrease. A zero correlation means that there is no relationship between the variables. Removed Time feature as it has no real impact.



* **Apply Scalability**:
* Standardization is a common preprocessing technique in machine learning that involves transforming the data and also to ensure that all features are on the same scale.
* We have performed scaling of variables Amount as it the only original variable from dataset. Also, while checking the amount fields, we have identified the difference is huge which requires scaling.



* **Split the data**:
  + We have defined the dependent variable(y) with all values of target variable class and other variables in to independent variable X.
  + We have Split the data into test and train in the ratio of 0.25 and random state as ‘0’. We observed a slight change in the area under curvature score with change in split ratio and random state.
  + Training data is used to build the model and testing data is used to evaluate the model
* **Perform sampling to balance data**:
  + As it is already mentioned the data is imbalanced with more towards on class. We will need to balance the classes of dataset. There are two main sampling techniques, Under sampling and Oversampling. We have used Under Sampling (RUS).
  + Under sampling involves randomly selecting a subset of the majority class (legitimate transactions) to match the size of the minority class (fraudulent transactions).

**2.3.2: Build the Models**:

**Decision Tree Classifier**:

* The decision tree is the simplest and most popular classification algorithm. For building the model the decision tree algorithm considers all the provided features of the data and comes up with the important features. Once the important features identified then the model trains with the training data to come up with a set of rules. These rules used in predicting future cases or for the test dataset.
* We have the fit the model on X\_train and y\_train values, no other parameters used for training.

**Random Forest**:

* The Random Forest falls under ensemble learning algorithm category. Here, we create N decision tree models using the random forest algorithm. The target value is predicted by all the models. The final target value will be predicted using the majority voting method. The random forest algorithm generates the sample dataset at random to build the individual decision tree. The bootstrap samples are these sample datasets. If we want to build these N decision trees to make the forest, the algorithm will first generate N bootstrap samples. Later, one decision tree model will be built for each bootstrap sample.
* We have the fit the model on X\_train and y\_train values, no other parameters used for training, also used 50 n-estimators. Algorithm internally makes use of 50 Decision trees. We tried modifying n-estimators.

**Support Vector Machine**:

* Support Vector Machine (SVM) is a supervised machine learning algorithm used for classification and regression tasks. It is based on the concept of decision planes that define decision boundaries. SVM is used to find an optimal hyperplane that maximizes the margin between the two classes of data points. The data points nearest to the hyperplane are called the support vectors. SVM is one of the most popular machine learning algorithms used in various classification and regression tasks.
* We have the fit the model on X\_train and y\_train values, no other parameters used for training.

**Naive Bayes Classifier**:

* Naive Bayes classifier is a probabilistic algorithm that is used for classification tasks. It is based on Bayes' theorem, which is a way of calculating the probability of a hypothesis given some observed evidence. The "naive" part of the algorithm's name refers to the simplifying assumption that the features being used for classification are independent of one another, which is often not true in real-world applications.
* We have the fit the model on X\_train and y\_train values, no other parameters used for training.

**KNN Classifier**:

* KNN (K-Nearest Neighbors) is a classification algorithm used in machine learning that works by finding the K closest data points to a new observation (in our case, a credit card transactions) and assigning it to the class that is most common among its nearest neighbors.
* We have fit the model on X\_train and y\_train values and we have taken n\_neighbors as 7. We performed our evaluation with different values of neighbors.

**XGBClassifier**:

* Extreme Gradient Boosting Classifier is a powerful machine learning algorithm used for classification tasks, including credit card fraud detection. It is an ensemble method that combines multiple weak learners (decision trees) to form a strong learner that can make accurate predictions on new data.
* We have fit the model on X\_train and y\_train values and have provided the max\_depth as 4.

3: **Empirical Results:**

All the models are evaluated with imbalanced data set and the balanced dataset. Since the data is highly imbalanced, for model evaluation we considered Area Under Curvature score as one primary metric along with Accuracy, recall and F1 score.

When dealing with imbalanced datasets, accuracy may not always be the most informative metric. This is because in imbalanced datasets, the classifier may perform well on the majority class but not on the minority class, which is often the class of interest. As a result, AUC-ROC measures the performance of the classifier across all possible threshold values, and is insensitive to the class distribution.

**3.1 Model Performance**:

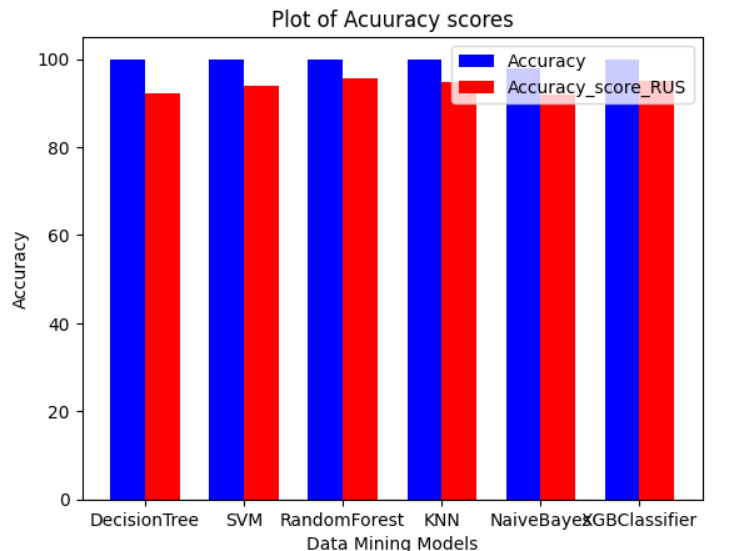
Model performance before applying Under Sampling:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Accuracy | F1- Score | AUC score | Recall |
| Decision Tree | 99.91 | 76 | 87.89 | 76.66 |
| SVM | 99.93 | 77 | 83.32 | 66.66 |
| Random Forest | 99.95 | 84 | 88.74 | 77.5 |
| KNN | 99.94 | 82.5 | 87.49 | 75 |
| Naïve Bayes | 97.83 | 11.5 | 90.5 | 83.33 |
| XGBClassifier | 99.95 | 86.60 | 90.5 | 80 |

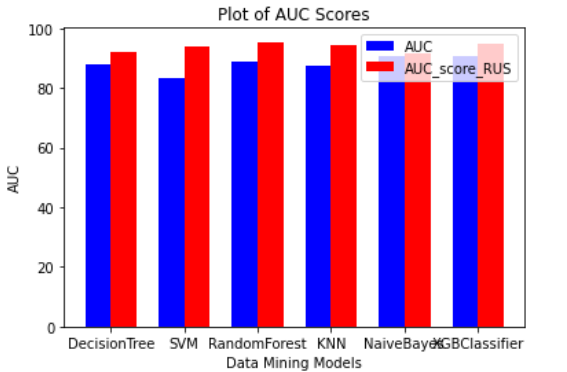
Model performance after applying Under Sampling:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Accuracy | F1- Score | AUC score | Recall |
| Decision Tree | 92.27 | 92 | 92.28 | 92.43 |
| SVM | 93.90 | 93.39 | 93.75 | 89 |
| Random Forest | 95.52 | 95.23 | 95.43 | 92.43 |
| KNN | 94.71 | 94.32 | 94.59 | 90 |
| Naïve Bayes | 91.86 | 91 | 91.6 | 84 |
| XGBClassifier | 95.12 | 94.82 | 95 | 92 |

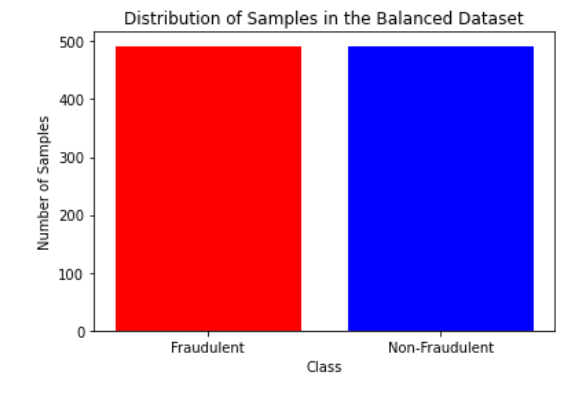
Plot of Accuracy with and without sampling:



Plot of AUC score with and without sampling:



Distribution of Samples in Balanced Dataset:



**3.2 Observations and** **Computational Efficiency**

* Initially when we look into models trained on imbalanced dataset, all most all the models gave similar accuracy of 99% which is overfitting, but when we consider Area Under Curvature score Naïve Bayes and XGB classifier performed really better with AUC score of 90%.
* After applying the Under Sampling, Random Forest model performed really better in detecting or predicting the fraud transactions. It gave an accuracy and AUC score of 95%.
* Models performed well after Under Sampling with good AUC score and accuracy.

As part of future work, we can also include spark streaming and sparkML to deal with credit card fraud detection using real time processing of data.

4. **DELIVERABLES:**

We have uploaded our code to GitHub. Instructions are provided in the README file on how to run the code. Feel free to download and work on our project.

5. **References:**

* [**https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud**](https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud)
* [**https://ijarcce.com/wp-content/uploads/2020/06/IJARCCE.2020.9542.pdf**](https://ijarcce.com/wp-content/uploads/2020/06/IJARCCE.2020.9542.pdf)
* <https://towardsdatascience.com/credit-card-fraud-detection-using-machine-learning-python-5b098d4a8edc>
* <https://www.geeksforgeeks.org/ml-credit-card-fraud-detection/>