CREDIT CARD FRAUD DETECTION

Yuktam Yadav [*1] **, Dr. Pooja Khanna** [*2]

[*1] Student, Amity University, Lucknow, India
yuktam.yadav@s.amity.edu

[*2] Assistant Professor, Amity University, Lucknow, India
pkhanna@lko.amity.edu

ABSTRACT

It is important that credit card companies are able to recognize fraudulent credit card transactions so that customers are not charged for items that they did not purchase. In an era where digital transactions have become ubiquitous, the security of financial transactions is of paramount importance. The advent of machine learning and data science techniques offers promising avenues for enhancing fraud detection mechanisms. Analysing a dataset is a critical step in obtaining relevant insights from massive amounts of data. This research paper delves into the use case of Data science & Machine Learning and its applications for our major project. It goes into the use of Python as a powerful tool for data analysis, emphasizing its importance in dealing with complex datasets. Furthermore, the project investigates the many Python libraries used in data science and machine learning applications, the significance of data visualization with Libraries such as Seaborn and Matplotlib, etc. After which it goes on with the main work. This project of ours delves into the realm of credit card fraud detection, leveraging Python and its powerful libraries such as NumPy, Pandas, Seaborn, TensorFlow, and Matplotlib for comprehensive data analysis and visualization. Employing a host of machine learning algorithms including K-Nearest Neighbors, Logistic Regression, Random Forest, Decision Tree, and LDA, the project aims to tackle the challenge posed by highly unbalanced datasets, specifically transactions made by European cardholders in September 2013. By discerning fraudulent transactions accurately, this endeavour seeks to bolster the integrity of financial systems, safeguarding both customers and institutions from potential losses.

Keywords: Python, pandas, seaborn, data science, machine learning.

I.INTRODUCTION

In today's data driven world, data science and machine learning is crucial for organizations to unearth new insights, make educated decisions, prevent fraudulent activities and gain a competitive advantage. It entails inspecting, cleaning, manipulating, and modelling data in order to extract relevant information while also generating predictions and projections based on machine learning algorithms. Python which is an versatile programming language, has become a popular choice for initiating these operations among data scientists with the help of its libraries. The digital landscape has revolutionized the way financial transactions are conducted, offering unparalleled convenience but also exposing vulnerabilities to fraudulent activities. In this context, the focus on credit card fraud detection emerges as a critical imperative for ensuring the security and trustworthiness of financial systems. This project centres on a dataset encompassing credit card transactions made by European cardholders during September 2013, a corpus of 284,807 transactions, among which a mere 492 are identified as fraudulent. The glaring disparity in class distribution, with frauds accounting for a minuscule 0.172%, accentuates the intricacy of the challenge. Leveraging the prowess of Python and its rich ecosystem of libraries including NumPy, Pandas, Seaborn, TensorFlow, and Matplotlib, alongside a diverse array of machine learning algorithms ranging from K-Nearest Neighbors and Logistic Regression to Random Forest and Decision Tree the project embarks on the journey of anomaly detection. By meticulously analyzing transactional data and arnessing the predictive capabilities of machine learning models, the goal is to discern patterns indicative of fraudulent behaviour, thereby empowering credit card companies to pre-emptively mitigate risks and fortify their defences against financial malfeasance. This endeavour not only underscores the significance of interdisciplinary collaboration between data science and financial domains but also underscores the potential of technology to bolster the resilience of financial ecosystems in the face of evolving threats.

II.PROBLEM STATEMENT

Detecting fraudulent credit card transactions becomes a serious problem particularly in scenarios of skewed datasets where the proportion of the fraud appear small among numerous of legitimate transactions. The goal of this research is to identify machine learning algorithms that would classify the fraudulent transactions differently from the regular transactions, and for this purpose we need to deal with the class imbalances in the dataset. Moreover, the research entails an implementation of algorithms like Logistic Regression, Linear Discriminant Analysis, Decision Tree, Random Forest and KNN using Python and it's different host of libaries. We also produce a Classification Report with the help of Confusion Matrix along with the F1 Score, ROC and AUC Scores for our dataset to check if it can predict fraud transactions with efficient accuracy on different ML based data models or if it needs oversampling to balance the dataset. Through these steps we will ascertain the most fitting approach for fraud detection concerning credit card transactions.

III.UNDERSTANDING DATA SCIENCE

Guarding against fraudulent financial transactions is an ongoing challenge. As technology advances and business becomes further digital, the need for a robust fraud discovery strategy has nowadays been lesser [7]. This is where cross-disciplinary data wisdom comes into play, with robust styles and tools to descry and help fraudulent conditioning with unknown delicacy. [19] At its core, data science is about rooting practicable perceptivity from large datasets. Used to describe fraud, it involves checking sale data for distractions, anomalies, and patterns of fraud. Still, the sheer volume and complexity of fiscal data is a redoubtable handicap. [5] Fortunately, data science wisdom offers numerous results to overcome these challenges. One of the crucial pillars of data wisdom in fraud discovery is data preprocessing. [1] This involves cleaning, transubstantiating, and directly recycling raw data to ensure quality and comity with logical algorithms. styles similar as outlier identification, missing value insinuation and point engineering play an important part in reducing the data, enhancing its discriminative power. Once the data is pre-processed, the coming step is point selection and birth of data. Then, data scientists use sphere knowledge and statistical ways to identify the most applicable factors similar as volume, time, position, and client demographics that distinguish licit deals from fraud. [15]

IV.UNDERSTANDING MACHINE LEARNING

When it comes to financial transactions, there is always the fear of fraud, leaving people worried about the security of their money and trusting the system but there is hope: machine learning, a powerful tool that helps experts provide deceptive activity is detected and released with remarkable accuracy and efficiency [16]. Machine learning is all about finding patterns in big data. In terms of cheating detection, that means looking for unusual behaviors or behaviors that are out of step with the behavior's usual pattern [11]. Experts use smart computer programs like KNearest Neighbors (KNN), Decision Trees, and Random Forests to sift through tons of marketing data to determine what might be suspicious K-Nearest Neighbors (KNN) is an elegant method that checks the proximity of networks to each other. If a rumor comes from too close to others, it can raise red flags about fraud [15]. Decision trees like road maps, classify projects into different categories based on important information such as how much money was spent or where they occurred They are great for indirect spatial communication based on this information [20]. But the real star of the show is Random Forests. They are like a set of decision trees that make sure to catch all bad behavior while ignoring the random noise. But machine learning isn't a magic wand as it takes a lot of work to get it right. As fraudsters come up with new techniques, experts continue to innovate their tools and techniques to stay ahead of the game [9]. But with careful adaptation and lots of experimentation, machine learning can be a powerful ally in keeping our money safe and ensuring financial systems work out for everyone [5].

Categories of Machine Learning:

Machine learning algorithms can be categorized based on their approach, the nature of the input data, and the specific problem they are designed to address. The following are -

Supervised Learning: Supervised learning involves training a model using a labeled dataset. A labeled dataset comprises both input and output parameters [8]. Classification and regression are the two primary subcategories within supervised learning [4]. Classification algorithms come into play when the outcomes are constrained to a finite set of values, while regression algorithms are employed when the outputs can take on any numerical value within a defined range [3].

Unsupervised Learning: Unsupervised learning algorithms work with datasets that only contain input information [7]. Their objective is to uncover underlying structures or patterns within the data, such as grouping or clustering data points [5]. These algorithms learn from unlabeled test data, making them proficient at detecting hidden structures without predefined labels, classifications, or categories [6].

Reinforcement Learning: Reinforcement learning constitutes a subset of machine learning that focuses on the decisionmaking process of software agents in an environment. The aim is to maximize cumulative rewards based on a series of actions taken within the given environment. Feedback, in the form of rewards and penalties, guides the system as it navigates its problem space. This approach is particularly relevant in scenarios where a learning system interacts with its surroundings and adjusts its behavior to optimize the overall outcome [10].

Categories of Machine Learning Algorithms Utilized:

Decision Tree: Decision trees are user-friendly models, which hierarchically display various outcomes of decisions and actions. The nodes inside denote decisions for which functions are the basis and the leaf node is a place to decide an outcome. Decision trees are used across a wide range of industries including finance, medicine, and marketing, among others. In comparison with other algorithms, they are to a greater extent helpful for classification and regression operations due to the factors concerned with their comprehension and ease of understanding. Decision trees are able to recognize the fraud via deepening into the blocks that clearly mark the molehill of fraud transactions from a legit transaction set, such as transaction amount, time and location. An investigation into signal attributes for subsequent tree formation and pattern identifying would allow the system to automatically classify transactions which are suspicious [19].

Random Forest: Random forests is a ML Algorithm that make use of different decision trees in synchronization so as to strengthen the predictive capacity. By training each tree in the set of trees on a randomly chosen subset of the data and allowing the individual forecasts, we can avoid overfitting. The final prediction is the average of the predictions of the individual trees, which is based on how the trees voted, on the basis of which tree is considered more important. Feature Random Forests is adaptable and is can be used for classifying or performing regression tasks. They are in the purely financial applications, medical area and remote sensing. Random forests demonstrate the greatness of the algorithm at fraud detection by taking into an account the outcomes of multiple decision trees trained on diverse data sets. This approach, which mixed models, enhances the accuracy of fraud detection models by recognizing complex fraud schemes and cutting down on overfitting [19].

Logistic Regression: It is a stats-driven machine learning method that is applied in binary classification applications such as operations (binary logistic regression). This algorithm reproduces the probability of a given binary outcome by means of different gradient descent parameters that take the predictor variables into account. The anomaly of logistic regression suggests that it is a linear model and it has several applications such as epidemiology, marketing, and finance for the purpose of binary classifications. Being helpful at recognizing binary outputs, the logistic regression makes it possible to recognize fraudulent actions in transactions. Probabilities of fraud are learned and calculated using the logistic regression model which classifies a transaction into either fraudulent or legitimate category given its characteristics as well as the customer behavior [19].

K-Nearest Neighbors (KNN): K-nearest Neighbours is a simplest and efficient algorithm in the tasks of classification and regression. It works in this way that it gives the nearest K data points to the query point and then, calculates the majority class or the mean value (for classification and regression respectively) of those points. KNN is used in recommender systems, characterized by the automation in processing huge sets of data. It is also applied in pattern recognition and anomaly detection. KNN's detection of fraud is done by means of comparison of fraudulent transactions' parameters with the ones in known cases of fraud to find similarities between them. Through analyzing the KNN neighbors of the transaction, the latter is classified as fraudulent by the deviation of it from standard norm behavior [19].

Linear Discriminant Analysis (LDA): Linear discriminant analysis is a classifier which has been built from the features finding the best linear combination of that separates the classes. In fact, it maps data into a lowerdimensional space but maximizes the inter-mays and the variance within these smaller classes. LDA is good particularly in such applications as image analysis, bioinformatics and financial control for selecting one class out of the others. LDA will be a possible subject used to capture transactions in a much lower dimensional space such that it will be easily distinguishable between fraudulent and legitimate transactions. Through classing the distances between the means of classes and the variance of the within-class as well as of the with features representation, LDA can represents the transactions as fraudulent or legitimate [19].

Naive Bayes: Naive Bayes classifier (also referred to as probabilistic classifier) works with Bayes theorem and an assumption that different features of the data are independently occurring. Ultimately, the Utility of Simple Naive Bayes is ensured not only by its simplicity but by its ability to perform well in many real-world applications, e.g. Text Classification, Spam Filtering, and Medical Diagnostics. Naive Bayes is the preferred detection approach for fraud among others because it is plain and easy in

processing large data sets. It is possible to calculate the probability that the transaction is fraudulent due to its characteristics and discretionary belonging of transactions to one or another class with the use of Naive Bayes which allows transactions separation into legitimate and fake classes [19].

V.PROPOSED METHODOLOGY

Python:

We use Python as the main programming language for analyzing the dataset along with implementing machine learning models or algorithms [16]. Accessing to libraries such NumPy, Pandas, SciPy, TensorFlow, Scikit-learn, Seaborn and Matplotlib enables us to perform all the data & algorithmic processing efficiently including visualization and model building [2]. Using Python, We can model a host of supervised machine learning models, both classification and regression based [1].

Python Libraries:
Pandas: It provides high performance, fast, easy to use data structures and data analysis tools for manipulating numeric data and time series [17]. Pandas is built on the numpy library and written in languages like Python, Cython, and C [13]. In pandas, we can import data from various file formats like JSON, SQL, Microsoft Excel, etc.

Numpy: It's an Python library, It is used for scientific computing in python. It contains a collection of tools and techniques that can be used to solve on computers with mathematical model of problems. It has functionalities such as High performance and multi-dimensional array objects, High level mathematical functions, matrices, etc [3].

TensorFlow: It is an open source library for Artificial Intelligence and Machine Learning applications available with programming languages such as Python. It can be utilized across a range of tasks but is specifically used to focus on deep neural network architecture and training it.

Scikit Learn: Scikit-learn is a popular and free machine learning library in Python which provides easy and efficient tools to process and analyze the data. Built upon NumPy, SciPy, and matplotlib, scikit-learn is equipped with a lot of unsupervised and supervised learning algorithms, such as supervised learning and unsupervised learning, together with model evaluation or hyperparameter tuning tools. The functionalities of Scikit-learn includes Regression, Model Selection, Preprocessing, Clustering, Classification, etc [8].

Matplotlib: It is a Python library used for plotting graphs with the help of other libraries like Numpy and Pandas. It is a powerful tool for visualizing data in Python. It is used for creating statical interferences and plotting 2D graphs of arrays.

Seaborn: It is also a Python library used for plotting graphs with the help of Matplotlib, Pandas, and Numpy. It is built on the roof of Matplotlib and is considered as a superset of the Matplotlib library. It helps in visualizing univariate and bivariate data.

Supervised Machine Learning Algorithms:

Logistic Regression: A widely used linear classification algorithm suitable for binary classification tasks [19].

Linear Discriminant Analysis (LDA): Projects data onto a lower-dimensional space while maximizing class separation [19].

Naive Bayes: A probabilistic classifier based on Bayes' theorem, assuming conditional independence between features [19].

K-Nearest Neighbors (KNN): Classifies data points based on the majority class of their k nearest neighbors [19].

Random Forest: An ensemble learning method that combines multiple decision trees for improved accuracy and robustness [19].

Decision Tree: A tree-like model that classifies data points by traversing a series of decision rules based on feature values [19].

VI.IMPLEMENTATION

Exploratory Data Analysis and Insights :

Before implementing the data models for analyzing data accuracy, we analyze the data to grasp its attributes, dispersion, and associations using EDA. This aids in uncovering potential patterns and outliers in the data. It establishes the foundation for constructing the model later on.

At first, we implemented methods to extract generic insights from our huge dataset such as datatype, mean, median, mode, deviation, etc. related to the variables.

```
# # To know the information of the dataset

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 284807 entries, 0 to 284806
Data columns (total 31 columns):
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   Time    284807 non-null  float64
 1   V1      284807 non-null  float64
 2   V2      284807 non-null  float64
 3   V3      284807 non-null  float64
 4   V4      284807 non-null  float64
 5   V5      284807 non-null  float64
 6   V6      284807 non-null  float64
 7   V7      284807 non-null  float64
 8   V8      284807 non-null  float64
```

Figure 1: Exploratory Analysis on Dataset using Python

```
df.describe()
##describe() is used to view some basic statistical details like per
```

	Time	V1	V2	V3	V4
count	284807.000000	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05
mean	94813.859575	3.919560e-15	5.688174e-16	-8.769071e-15	2.782312e-15
std	47488.145955	1.958696e+00	1.651309e+00	1.516255e+00	1.415869e+00
min	0.000000	-5.640751e+01	-7.271573e+01	-4.832559e+01	-5.683171e+00
25%	54201.500000	-9.203734e-01	-5.985499e-01	-8.903648e-01	-8.486401e-01
50%	84692.000000	1.810880e-02	6.548556e-02	1.798463e-01	-1.984653e-02
75%	139320.500000	1.315642e+00	8.037239e-01	1.027196e+00	7.433413e-01
max	172792.000000	2.454930e+00	2.205773e+01	9.382558e+00	1.687534e+01

8 rows × 31 columns

Figure 2: Exploratory Analysis on Dataset using Python - 2

We generated a bar plot chart using the missingo package of python to check for any null values, and we found none.

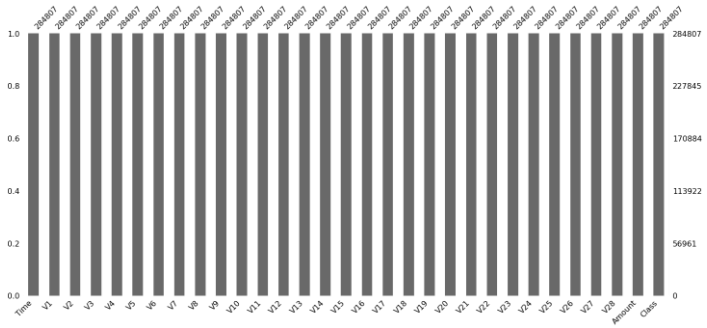

Figure 3: Bar Graph to check Null Values using Missingo Python Package.

Using the matplotlib library's Histograms, we visualized the distribution of data for each variable and gained such overall visual insights:

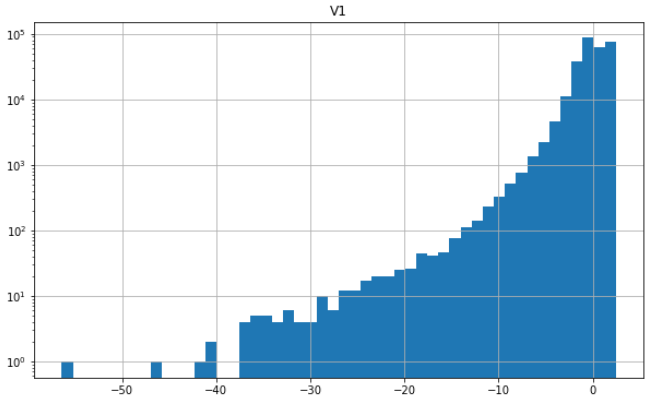

Figure 4: Histogram Visualization of a data variable.

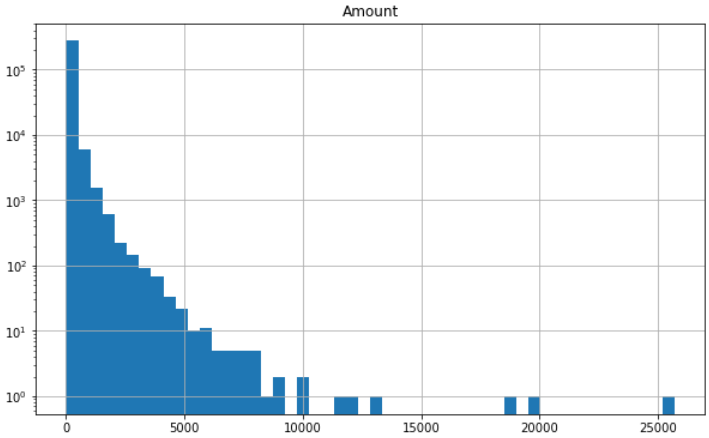

Figure 5: Histogram Visualization of Amount variable.

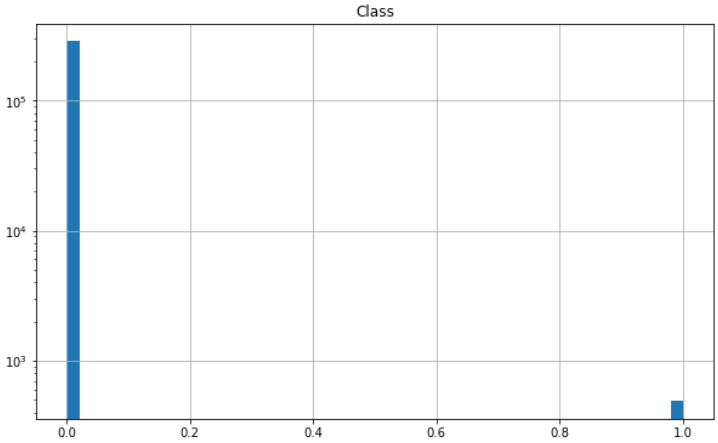

Figure 6: Histogram Visualization of Class variable.

Correlation: We extracted the correlation information between the different variables using a heatmap. A heatmap is a graphical representation of data in which data values are represented as colours. That is, it uses colour in order to communicate a value to the reader. This is a great tool to assist the audience towards the areas that matter the most when you have a large volume of data.

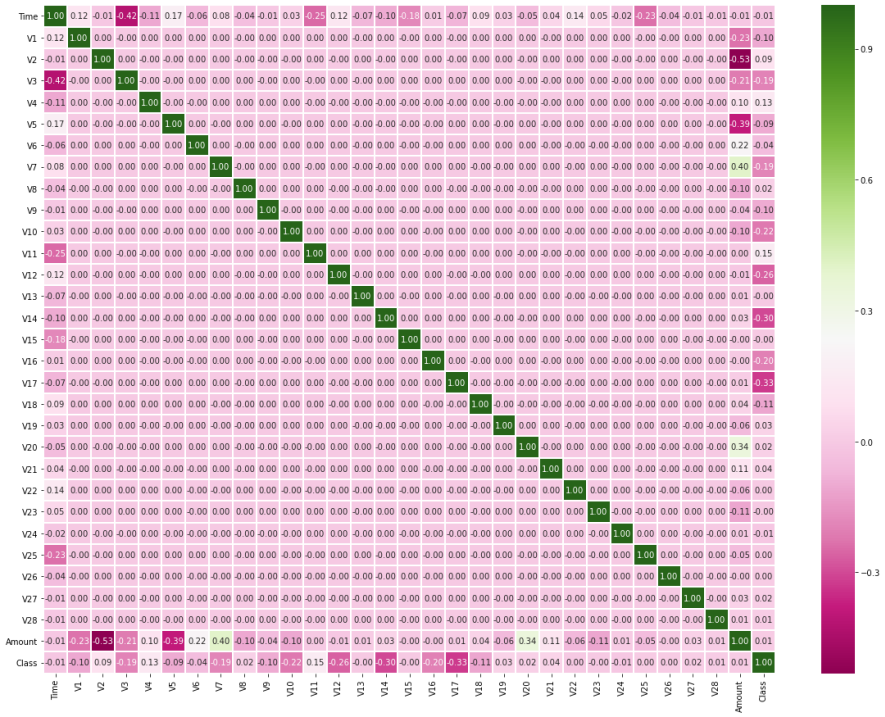

Figure 7: Heatmap to check correlation between data variables.

The outcomes from the heatmaps were as follows:

1. V1, V3, V5, V8, V9, V10, V11, V12, V13, V14, V15, V16, V19, V22, V23, V25, V26 are small and are negatively correlated with amount.

2. V2 is large, negatively correlated with amount.

3. V11 does not correlated with amount i.e 0.0.

4. V4, V6, V7, V14, V17, V18, V20, V21, V27, V28 are Small positively correlated.

Train-Test Split Evaluation:

The Train-Test split is a method for evaluating machine learning models. This method may be applied to classification and regression as a well as any supervised learning approach [8]. The process consists of two parts where firstly, you cut the dataset into two subgroups. The first subset is the training dataset that we will use in fitting the model while the second subset called validation dataset is reserved for other operations. The second subset is not used to train the model but instead, only input data vector is sent into the model for prediction that then makes the comparisons with the predictive values. Following this dataset is test dataset which is the second category. The train dataset is applied to fit a machine learning model [12]. While the test dataset's main responsibility is for evaluating the effectiveness of this model. The objective here is to estimate and analyse the performance of our ML or machine learning model on a new data: A data which is not used to train the model.

Confusion Matrix and Classification Report Metrics:

Confusion Matrix: A confusion matrix is a tabular form that is widely used in evaluating the appropriateness of a classification - based machine learning model. We can call positive cases as true positives when we predict yes (they have the disease) and they are also diagnosed with this disease [10].

True negatives (TN): We predicted no, and no one in the family had that condition, too.

False positives (FP): We supposed yes, but they really did not create an oppressive condition. (Affectionately called a "Type I error" or "false positive.")

False negatives (FN): As shocking as it is, they defy our prediction: "no" but "yes" they actually have the disease. (This is also called "Type II error" or "non-rejection of hypothesis. [14]"

Precision: It's the number of positive identifications that were correct among all of them [9].

Recall: It's about how many real positives did the test system flagged correctly [4].

F1 Score: The F1 Score is the harmonic mean of Precision and Recall. F1 is in most cases more useful than accuracy, especially when there is an imbalance in class distribution [18].

Kappa Score: It is used for the evaluation of the correctness of the classifier model. We already, that Cohen's K is a good evaluation statistic when it comes to working with unbalanced data. Cohen's kappa coefficient (κ) is a statistic used to quantify inter-rater agreement. Cohen's kappa performs this function by taking into account the correct classification a random guess would produce [6].

AUC: Its provides an aggregative measure of performance occurs all possible classification thresholds. It talks about linear dataset. AUC starts from 0 to 1. The higher the AUC, the better the performance of the model at distinguishing between the 51 positive and negative classes [3].

ROC (Receiver Operating Characterstic curve): It shows the performance of the model through all thresholds. It's the curve plot between the two parameters Tpr (Sensivity) and Fpr (specifity) [11].

Now, We've worked and imported specific libraries and their methods so that we can generate visualizations or graphics such as confusion matrix, mathematical outputs with metrics such as Classification Report, Kappa Score, AUC & ROC Curve Score derived from Supervised Machine Learning algorithms applied on our dataset. We have got the following outcomes:

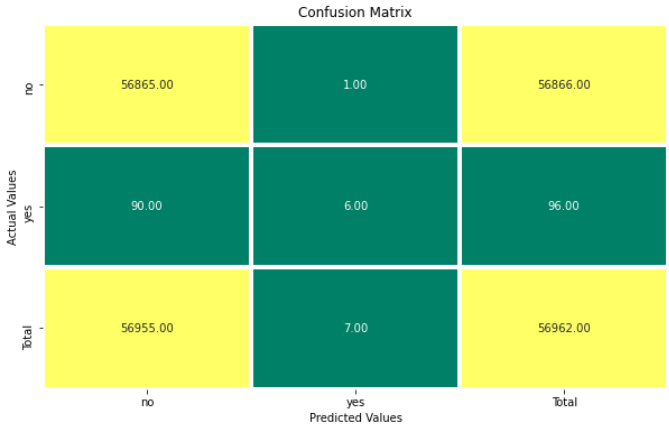

Figure 8: Confusion Matrix to compare predicted and actual values.

Classification Report -

	precision	recall	f1-score	support
0	1.00	1.00	1.00	56866
1	0.86	0.06	0.12	96
accuracy			1.00	56962
macro avg	0.93	0.53	0.56	56962
weighted avg	1.00	1.00	1.00	56962

Figure 9: Classification Report.

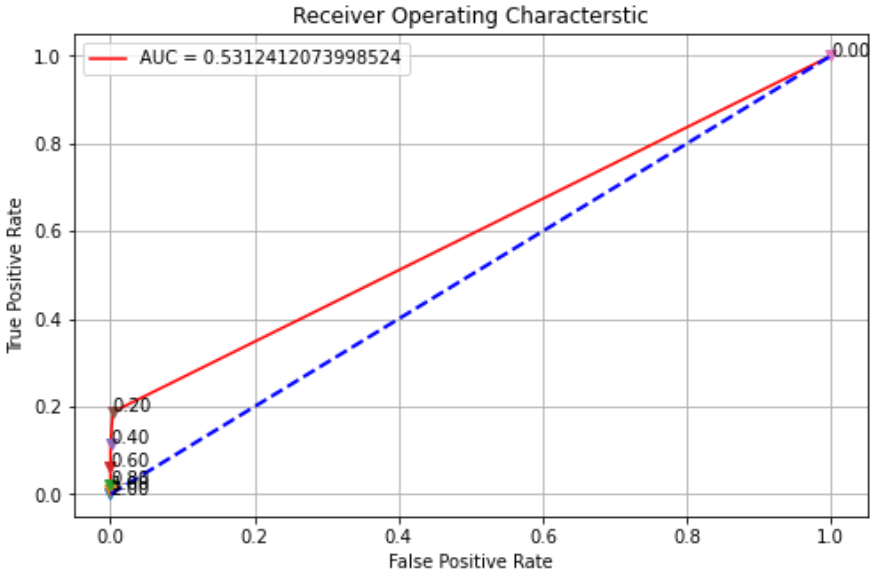

Figure 10: Receiver Operating Characteristic (ROC) Score's Graphical Representation.

The Kappa Score we achieved is 0.1163024217567018, while our AUC Score is 0.5312412073998524. The Number of probabilities to build ROC is 7.

Now, From above outcomes we can understand that Kappa Score which is 0.116 is very less, while the AUC score is moderate i.e 0.53. We'll have to increase it because the confusion matrix provides us an insight that due to a largely imbalanced data, we are not getting efficient outcomes to predict which data model will be effective for analyzing the fraud transactions in our dataset.

Because
the dataset is an imbalance dataset,So the kappa score and AUC score are also low. We can balance it by using technique such as SMOTE (Synthentic Minority Oversample Technique).

SMOTE (Synthentic Minority Oversampling Technique):

SMOTE is an oversampling technique where the synthetic samples are generated for the minority class. This algorithm helps to overcome the overfitting problem posed by random oversampling. It focuses on the feature space to generate new instances with the help of interpolation between the positive instances that lie together. General idea to carry out this technique is to bring the minority class values (either 0 or 1) to a comparable number in terms of the other class. In other words, to match up the length of the other class. With the help of python library imblearn, we successfully applied our SMOTE technique on our unbalanced dataset to fix it. Post our operations, The data gets balanced.

Confusion Matrix and Re-evaluated Classification Report Metrics Post SMOTE:

Now below, We have again performed the Split Train-Testing on our dataset, while re-evaluating our model to generate the confusion matrix and other classification based metrics again.

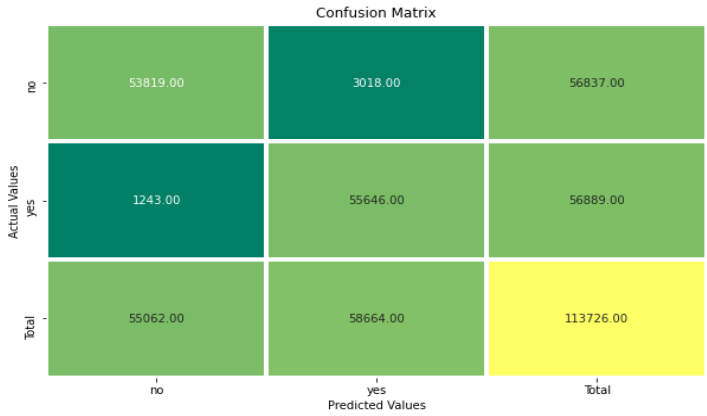

Figure 11: Confusion Matrix post SMOTE application on dataset.

Classification Report (Re-Evaluated)

	precision	recall	f1-score	support
0	0.98	0.95	0.96	56837
1	0.95	0.98	0.96	56889
accuracy			0.96	113726
macro avg	0.96	0.96	0.96	113726
weighted avg	0.96	0.96	0.96	113726

Figure 12: Re-evaluated Classification Report

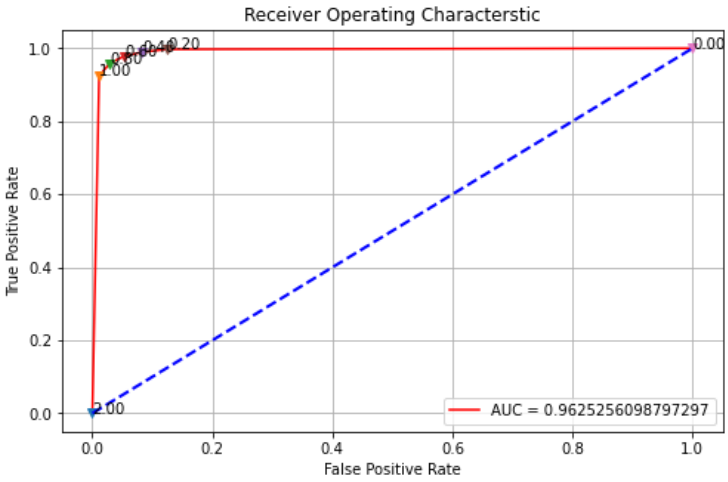

Figure 13: Receiver Operating Characteristic (ROC) Score's Graphical Representation post SMOTE.

The Kappa Score we achieved is 0.925064423112555, while our AUC Score is 0.9625256098797297. The Number of probabilities to build ROC is 7.

We can see an improvement in the ROC AUC score, at k = 7 -> 0.962 we get the highest ROC and AUC score followed by k= 6,5,4 and so on. The F1 score is 96.

Now the dataset is suitable enough for data modelling.

VII.DATA MODELLING

We trained different supervised classification models and tested those models to calculate their accuracy over our dataset containing transactions, So, as to identify which Machine Learning Model processes the data with more efficiency and presents accurate estimation of the fraud and non-fraud transactions. The outcomes are showcased below as per it's machine learning model's category.

Logistic Regression:

Logistic Regression

```
from sklearn.metrics import accuracy_score

#Import Library for Logistic Regression
from sklearn.linear_model import LogisticRegression

#Initialize the Logistic Regression Classifier
logisreg = LogisticRegression()

#Train the model using Training Dataset
logisreg.fit(x_train, y_train)

# Prediction using test data
y_pred = logisreg.predict(x_test)

# Calculate Model accuracy by comparing y_test and y_pred
acc_logisreg = round( accuracy_score(y_test, y_pred) * 100, 2 )
print( 'Accuracy of Logistic Regression model : ', acc_logisreg )
```

```
Accuracy of Logistic Regression model :  97.48
```

Figure 14: Modelling of Logistic Regression (An ML Algorithm) using Python to check it's accuracy in predicting fraudulent transactions efficiently.

Naive Bayes:

Gaussian Naive Bayes

```
#Import Library for Gaussian Naive Bayes
from sklearn.naive_bayes import GaussianNB

#Initialize the Gaussian Naive Bayes Classifier
model = GaussianNB()

#Train the model using Training Dataset
model.fit(x_train, y_train)

# Prediction using test data
y_pred = model.predict(x_test)

# Calculate Model accuracy by comparing y_test and y_pred
acc_ganb = round( accuracy_score(y_test, y_pred) * 100, 2 )
print( 'Accuracy of Gaussian Naive Bayes : ', acc_ganb )
```

```
Accuracy of Gaussian Naive Bayes :  86.83
```

Figure 15: Modelling of Naive Bayes (An ML Algorithm) using Python to check it's accuracy in predicting fraudulent transactions efficiently.

Decision Tree:

```
#Import Library for Decision Tree Classifier
from sklearn.tree import DecisionTreeClassifier

#Initialize the Decision Tree Classifier
model = DecisionTreeClassifier()

#Train the model using Training Dataset
model.fit(x_train, y_train)

# Prediction using test data
y_pred = model.predict(x_test)

# Calculate Model accuracy by comparing y_test and y_pred
acc_dtree = round( accuracy_score(y_test, y_pred) * 100, 2 )
print( 'Accuracy of  Decision Tree Classifier : ', acc_dtree )
```

```
Accuracy of  Decision Tree Classifier :  100.0
```

Figure 16: Modelling of Decision Tree (An ML Algorithm) using Python to check it's accuracy in predicting fraudulent transactions efficiently.

KNN (K-Nearest Neighbors):

```
#Import Library for KNN
from sklearn.neighbors import KNeighborsClassifier

#Initialize the KNN
model = KNeighborsClassifier()

#Train the model using Training Dataset
model.fit(x_train, y_train)

# Prediction using test data
y_pred = model.predict(x_test)

# Calculate Model accuracy by comparing y_test and y_pred
acc_rf = round( accuracy_score(y_test, y_pred) * 100, 2 )
print( 'Accuracy of  KNN : ', acc_rf )
```

Figure 17: Modelling of KNN (An ML Algorithm) using Python to check it's accuracy in predicting fraudulent transactions efficiently.

Linear Discriminant Analysis:

```
#Import Library for Linear Discriminant Analysis
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis

#Initialize the Linear Discriminant Analysis Classifier
model = LinearDiscriminantAnalysis()

#Train the model using Training Dataset
model.fit(x_train, y_train)

# Prediction using test data
y_pred = model.predict(x_test)

# Calculate Model accuracy by comparing y_test and y_pred
acc_lda = round( accuracy_score(y_test, y_pred) * 100, 2 )
print( 'Accuracy of Linear Discriminant Analysis Classifier: ', acc_lda )
```

```
Accuracy of Linear Discriminant Analysis Classifier:  93.29
```

Figure 18: Modelling of Linear Discriminant Analysis (An ML Algorithm) using Python to check it's accuracy in predicting fraudulent transactions efficiently.

Random Forest:

```
#Import Library for Random Forest
from sklearn.ensemble import RandomForestClassifier

#Initialize the Random Forest
model = RandomForestClassifier()

#Train the model using Training Dataset
model.fit(x_train, y_train)

# Prediction using test data
y_pred = model.predict(x_test)

# Calculate Model accuracy by comparing y_test and y_pred
acc_rf = round( accuracy_score(y_test, y_pred) * 100, 2 )
print( 'Accuracy of  Random Forest : ', acc_rf )
```

```
Accuracy of  Random Forest :  100.0
```

Figure 19: Modelling of Random Forest (An ML Algorithm) using Python to check it's accuracy in predicting fraudulent transactions efficiently.

VIII.OUTCOMES AND CONCLUSİON

To solve our issue of proper detection of fraud transaction in an unbalanced dataset, several classification based supervised machine learning models were employed to the dataset, and the main challenge was to provide the most accurate identification of fraud incidents. The results give emphasis on the fact that all the algorithms worked fine, with the Decision Tree and Random Forest achieving a perfect accuracy score of 100%. The second best model was Logistic Regression with an accuracy score of 97.48%, while K-Nearest Neighbors and Linear Discriminant Analysis reached scores of 95.70% and 93.29% respectively. On the other hand, Naïve Bayes displayed 86.83% of being wrong.

This shows that the algorithms of machine learning can be powerful in recognizing credit cards fraud even with highly unbalanced datasets. The high performance of Decision Tree and Random Forest indicate in conclusion that these algorithms are best to use on this job. Yet the accuracy metric may not be sufficient enough on its own to determine the performance of these algorithms while dealing with imbalanced datasets. Thus, there arises the need to explore other performance metrics like precision, recall and F1- score for a more comprehensive evaluation of the algorithms.

In conclusion, this study has shown the capability of machine learning algorithms in detecting credit card fraud, on highly unbalanced datasets. The findings are very important for the credit card firms in developing the more effective fraud detection practices, hence shielding the customers from unauthorized transactions and financial losses.

IX.REFERENCES

[1] Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., & Bontempi, G. (2015). Credit card fraud detection: A realistic modeling and a novel learning strategy. IEEE Transactions on Neural Networks and Learning Systems, 29(8), 3784-3797.
[2]
Bhattacharyya, S., & Jha, S. (2019). A comprehensive review on credit card fraud detection techniques. Expert Systems with Applications, 135, 41-56.
[3]
Li, J., Han, Y., & Zhao, F. (2018). Credit card fraud detection model based on improved random forest algorithm. Mobile Information Systems, 2018, 1-8.
[4]
Phua, C., Lee, V., Smith, K., & Gayler, R. (2010). A comprehensive survey of data mining-based fraud detection research. ArXiv Preprint ArXiv:1009.6119.
[5]
Teixeira, E. R., Freitas, A. A., & Costa, I. G. (2017). A review of anomaly detection in automated financial transactions. Expert Systems with Applications, 83, 128-141.
[6]
Ribeiro, R. A., Zadrozny, B., & Liu, B. (2016). Learning under concept drift: A review. IEEE Transactions on Knowledge and Data Engineering, 28(10), 2349-2368.
[7]
Zhang, X. Y., & Wu, X. M. (2018). A review of online fraud detection techniques. ArXiv Preprint ArXiv:1808.05225.
[8]
Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials, 18(2), 1153-1176.
[9]
Islam, M. S., & Hossain, M. A. (2019). A survey of deep learning techniques for malware detection. IEEE Access, 7, 41512-41533.

[10]
Zhou, V. J., & Wu, Y. (2016). A review of deep learning-based approach for anomaly detection in network traffic. ArXiv Preprint ArXiv:1611.06373.
[11]
Kim, M., Han, J., & Kim, H. (2018). Deep learning in omics: A survey and guideline. Briefings in Bioinformatics, 19(6), 1-11.
[12]
Wang, H., & Oates, T. (2015). Encoding time series as images for visual inspection and classification using tiled convolutional neural networks. ArXiv Preprint ArXiv:1602.02296.

[13] Liao, W., Deshmukh, A. A., & Wang, S. (2019). Deep learning for health informatics. IEEE Journal of Biomedical and Health Informatics, 23(3), 1221-1237.
[14]
Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. (2000). LOF: Identifying density-based local outliers. Proceedings of the ACM SIGMOD International Conference on Management of Data, 93-104.
[15]
Yu, K., & Yu, S. (2015). Adaptive detection of online credit card fraud using temporal difference methods. Journal of Management Analytics, 2(1), 50-64.
[16]
Carcillo, F., Dal Pozzolo, A., Le Borgne, Y. A., Caelen, O., Mazzer, Y., & Bontempi, G. (2019). Scarff: A scalable framework for streaming credit card fraud detection with Spark. Information Fusion, 48, 99-113.
[17]
Lichman, M. (2013). UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences.
[18]
Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 321-357.
[19]
Lee, V. C., Phua, C., & Smith, K. (2012). A novel ensemble approach to credit card fraud detection. International Journal of Machine Learning and Cybernetics, 3(3), 183-197.
[20]
Pustokhina, I. V., Pustokhin, D. A., & Ivanova, T. A. (2019). Machine learning in fraud detection. Pattern Recognition and Image Analysis, 29(1), 172-182.