CREDIT CARD FRAUD DETECTION

A MINI PROJECT REPORT

Submitted by

LAKSHMI MANI SHANKAR [RA2011047010012]
SUJITH KUMAR T [RA2011047010145]
KRISHNA PRADEEP REDDY S [RA2011047010152]

Under the guidance of
Dr.Rakesh Kumar M
(Assistant Professor, Department of Computational Intelligence)

in partial fulfillment for the award of the degree
Of
BACHELOR OF TECHNOLOGY
in
COMPUTATIONAL INTELLIGENCE
of
FACULTY OF ENGINEERING AND TECHNOLOGY

S.R.M. Nagar, Kattankulathur, Chengalpattu District
JUNE 2022

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this project report titled “MINI PROJECT TITLE” is the bonafide work of “LAKSHMI MANI SHANKAR, SUJITH KUMAR T, KRISHNA PRADEEP REDDY S”, who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE	**SIGNATURE**
Dr.RAKESH KUMAR M	Dr. ANNIE UTHRA
GUIDE	**HEAD OF THE DEPARTMENT**
Assistant Professor	Dept. of Computational Intelligence
Dept. of Computational Intelligence	

Signature of the Internal Examiner	Signature of the External Examiner

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my guide, **Dr. RAKESH KUMAR M**, his valuable guidance, consistent encouragement, personal caring, timely help and providing me with an excellent atmosphere for doing the project. All through the work, in spite of his busy schedule, he has extended cheerful and cordial support to me for completing this project work.

LAKSHMI MANI SHANKAR

SUJITH KUMAR T

KRISHNA PRADEEP REDDY S

TABLE OF CONTENTS

Abstract

- Financial fraud is a massive problem in the financial industry that has long-term consequences, and while many other solutions have been created to combat the issue.

- To automate the examination of massive datasets,Data mining has been used to analyze large amounts of complex data.

- Financial databases have been successfully used in Exploration of data.Data mining has also played a vital role in the discovery of online transactions.

- It develops into it's challenging for two reasons one is it's difficult to stick to a regular schedule and the other is it's difficult because the patterns of fraudulent behavior vary greatly.

- The data sets on credit card fraud are severely distorted.On credit card fraud, which is extremely skewed and uneven this research investigates and contrasts the Logistic Regression performance and a variety of sampling techniques, including undersampling, oversampling, and random sampling and Decision Tree.These procedures employ both raw and pre-processed data.

- The accuracy, sensitivity, precision, and recall of the approaches are used to evaluate their performance.

CHAPTER 1

INTRODUCTION

1.1 Credit card fraud detection

It is a set of activities undertaken to prevent money or property from being obtained through false pretenses.Models create predictions based on transaction information and some background (historical) data. We employed only the most important features, which were chosen based on 2 (a chi-square test that analyses how expectations compare to actual observed data) and recursive feature reduction strategies, to make the model more resilient.

The list of viable machine learning applications for combating criminal activity is continually growing, and it is impossible to include all deserving instances here. However, we'll try to highlight the most intriguing to give you a sense of perspective. According to DataVisor's publicly published E-commerce fraud case study, their products assist organizations detect over 30% of fraudulent efforts with 90% accuracy and a 1.3 percent false-positive rate. Feedzai claims that their OpenML Engine enables banks to build their own Machine Learning software, which has drastically reduced fraud-related losses, according to another Fraud Detection analytics case study. There are also a number of successful Big Data fraud detection case studies that use automated analytics to justify enormous volumes of raw data and provide solutions that can forecast and prevent criminal activity.

In summary, the Fraud Detection business is booming right now, and it's just going to get bigger. It is expected to grow to $106 billion by 2027, according to Fortune Business Insights. North America is the most important region in this industry, although Europe and Asia are also important participants. The increasing amount of transactions during the COVID-19 lockdowns could push this figure much higher than expected. As a result, the global volume of transactions is closely linked to the market's growth. Insurance claims, money laundering, and electronic payments will account for the largest portions of the fraud detection and fraud prevention industry by 2021. In this case study, we'll look at how we overcame obstacles to make e-payments safer for our partners. Please leave your comments if you're interested in learning more about money-laundering prevention or protection strategies in the insurance industry. If you're interested in learning more about these topics, we may cover them as well!

"The most difficult aspect of the system was achieving strong metrics for users who had just completed a few transactions." We could use the conventional model, which is good for people who have a lot of transaction history, but it would give us worse ratings if we didn't have any (for example, a new user). Another obvious option is to treat these individuals as if they were empty accounts with simply their identity information and no transaction history. We lose the benefit of knowing at least some data about the users in this case, but the results provided by such a model are extremely stable (underfitting). We decided to investigate 'few-shot learning' techniques, which could help us improve our metrics, after holding a weekly stand-up on the subject. We developed a proof of concept, but it did not provide the dramatic improvement we had hoped for.

1.2 PROBLEM STATEMENT

People and financial institutions have become more reliant on online services as e-commerce websites have grown in popularity, resulting in an increase in credit card theft.

Fraudulent credit card transactions result in significant financial losses. In order to limit the losses caused by customers and financial institutions, an effective fraud detection system must be designed. Many models and strategies for preventing and detecting credit card fraud have been studied.

The problem of dataset imbalance exists in some credit card fraud transaction datasets. A good fraud detection system should be able to reliably identify fraudulent transactions and detect them in real-time transactions.

Anomaly detection and abuse detection are the two types of fraud detection.Anomaly detection systems utilize strategies to identify fresh frauds by training typical transactions.

A misuse fraud detection system, on the other hand, trains in the database history by labeling transactions as normal or fraud. As a result, this misuse detection system includes both a supervised learning system and an unsupervised learning system.

Fraudsters imitate normal consumer behavior, and fraud trends change quickly, therefore the fraud detection system must constantly learn and adapt.

Traditional card-related frauds (application, stolen, account takeover, fake and counterfeit), merchant-related frauds (merchant collusion and triangulation), and Internet frauds (site cloning, credit card generators, and phony merchant sites) are the three types of credit card frauds.

1.3 PROPOSED SOLUTION

I have loaded the dataset containing transactions made by credit cards in September 2013 by European cardholders.Then we have preprocessed the entire data using data preprocessing techniques.

Then I represented the data in the form of various graphs using data visualization.Now we have split the data into training and testing dataset and then we trained the dataset using various machine learning algorithms.

CHAPTER - 2

LITERATURE REVIEW

Prajal Save have proposed a model based on a decision tree and a combination of Luhn's and Hunt's algorithms. Luhn's algorithm is used to determine whether an incoming transaction is fraudulent or not. It validates credit card numbers via the input, which is the credit card number.

Address Mismatch and Degree of Outlierness are used to assess the deviation of each incoming transaction from the cardholder's normal profile. In the final step, the general belief is strengthened or weakened using Bayes Theorem, followed by recombination of the calculated probability with the initial belief of fraud using an advanced combination heuristic.

Vimala Devi. J to detect counterfeit transactions, three machine-learning algorithms were presented and implemented. There are many measures used to evaluate the performance of classifiers or predictors, such as the Vector Machine, Random Forest, and Decision Tree. These metrics are either prevalence-dependent or prevalence-independent. Furthermore, these techniques are used in credit card fraud detection mechanisms, and the results of these algorithms have been compared.

Popat and Chaudhary supervised algorithms were presented Deep learning, Logistic Regression, Nave Bayesian, Support Vector Machine (SVM), Neural Network, Artificial Immune System, K Nearest Neighbor, Data Mining, Decision Tree, Fuzzy logic based System, and Genetic Algorithm are some of the techniques used. Credit card fraud detection algorithms identify transactions that have a high probability of being fraudulent. We compared machine-learning algorithms to prediction, clustering, and outlier detection.

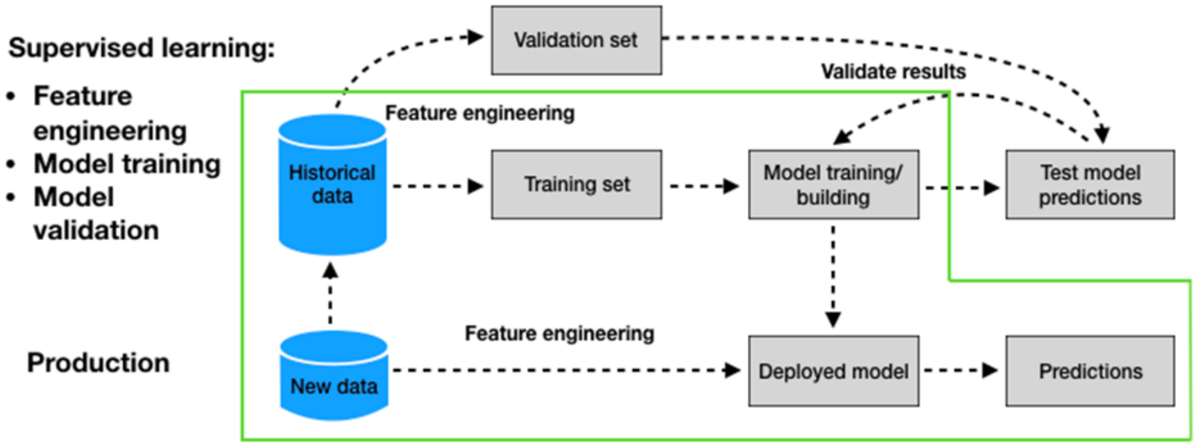
Shiyang Xuan for training the behavioral characteristics of credit card transactions, the Random Forest classifier was used. The following types are used to train the normal and fraudulent behavior features Random forest-based on random trees and random forest based on CART. To assess the model's effectiveness, performance measures are computed.

Dornadula and Geetha S using the Sliding-Window method, the transactions were aggregated into respective groups, i. , some features from the window were extracted to find the cardholder's behavioral patterns. Features such as the maximum amount, the minimum amount of a transaction, the average amount in the window, and even the time elapsed are available.

Sangeeta Mittal to evaluate the underlying problems, some popular machine learning algorithms in the supervised and unsupervised categories were selected. A range of supervised learning algorithms, from classical to modern, have been considered. These include tree-based algorithms, classical and deep neural networks, hybrid algorithms and Bayesian approaches.The effectiveness of machine-learning algorithms in detecting credit card fraud has been assessed.

CHAPTER - 3

3.1 ARCHITECTURE DIAGRAM



3.2 DESCRIPTION OF PROPOSED MODEL

A proposed model for credit card fraud detection using machine learning

Credit card fraud is a growing problem in the U.S., with a reported loss of $7.1 billion in 2017 according to the Bureau of Justice Statistics, and that number is only on the rise. With the recent scams that have plagued us in the U.S., there is a growing need to filter out fraudulent transactions while still allowing legitimate ones. To solve this problem, a model was proposed to classify fraudulent credit card transactions on the basis of three different types of features: transaction identification features, transaction characteristics features and transaction fraud features.

Identifying a fraudulent transaction involves detecting anomalies through various methods such as machine learning models and data mining techniques . These algorithms have shown great potential in detecting credit card fraud, but current implementations require a large amount of data and may not work for all types of credit cards.

As such, in this proposed model, it attempts to classify fraudulent transactions into three different groups: high-level illicit transactions that include Global Interchange Fraud (GIF), payment card skimming and counterfeit fraud; low-level illicit transactions that include secret keystroke scams (SKS) and browser phishing; and skimmed only frauds that include a mix of the two.

This model can be used to identify fraudulent credit card transactions by classifying each transaction on the basis of such features.

CHAPTER – 4
TOOLS AND SOFTWARES USED

4.1 DATASET DESCRIPTION

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.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 numeric 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-dependent 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.

4.2 TOOLS AND SOFTWARES USED

Python:
Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Platforms:

1.Google Colab:

- Colab is basically a free Jupyter notebook environment running wholly in the cloud.
- Most importantly, Colab does not require a setup, plus the notebooks that you will create can be simultaneously edited by your team members – in a similar manner to how you edit documents in Google Docs.
- The greatest advantage is that Colab supports the most popular machine learning libraries which can be easily loaded in your notebook.
- As a developer, you can perform the following using Google Colab
- Write and execute code in Python

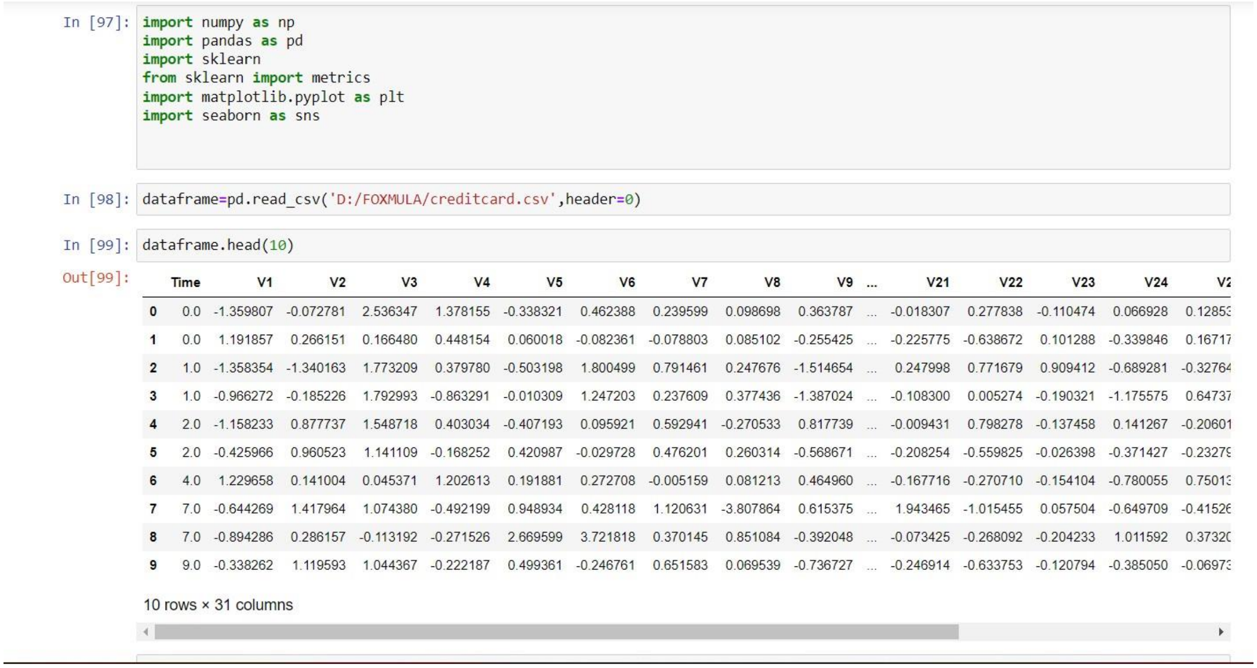
- Create/Upload/Share notebooks
- Import/Save notebooks from/to Google Drive
- Import/Publish notebooks from GitHub
- Import external datasets
- Integrate PyTorch, TensorFlow, Keras, OpenCV
- Free Cloud service with free GPU

2.Jupyter NoteBook

- The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project Jupyter.
- Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself.
- The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R.
- Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.

CHAPTER - 5
RESULTS AND DISCUSSIONS

5.1 CODE IMPLEMENTATION:



```
In [97]: import numpy as np
         import pandas as pd
         import sklearn
         from sklearn import metrics
         import matplotlib.pyplot as plt
         import seaborn as sns
```

```
In [98]: dataframe=pd.read_csv('D:/FOXMULA/creditcard.csv',header=0)
```

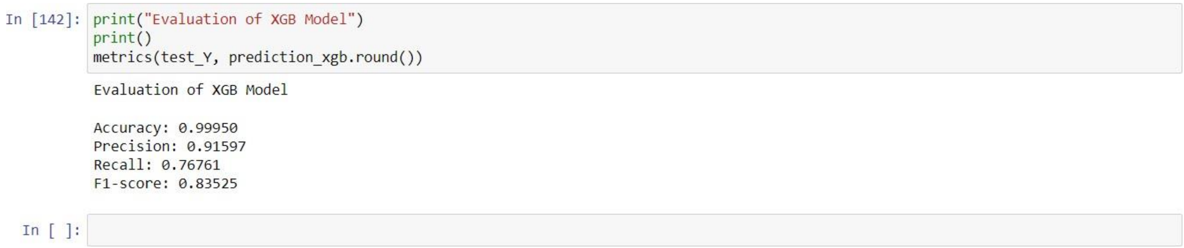
```
In [99]: dataframe.head(10)
```

Out[99]:

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V2
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928	0.12853
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846	0.16717
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281	-0.32764
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575	0.64737
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267	-0.20601
5	2.0	-0.425966	0.960523	1.141109	-0.168252	0.420987	-0.029728	0.476201	0.260314	-0.568671	...	-0.208254	-0.559825	-0.026398	-0.371427	-0.23279
6	4.0	1.229658	0.141004	0.045371	1.202613	0.191881	0.272708	-0.005159	0.081213	0.464960	...	-0.167716	-0.270710	-0.154104	-0.780055	0.75013
7	7.0	-0.644269	1.417964	1.074380	-0.492199	0.948934	0.428118	1.120631	-3.807864	0.615375	...	1.943465	-1.015455	0.057504	-0.649709	-0.41526
8	7.0	-0.894286	0.286157	-0.113192	-0.271526	2.669599	3.721818	0.370145	0.851084	-0.392048	...	-0.073425	-0.268092	-0.204233	1.011592	0.37320
9	9.0	-0.338262	1.119593	1.044367	-0.222187	0.499361	-0.246761	0.651583	0.069539	-0.736727	...	-0.246914	-0.633753	-0.120794	-0.385050	-0.06973

10 rows × 31 columns

.



```
In [142]: print("Evaluation of XGB Model")
          print()
          metrics(test_Y, prediction_xgb.round())

          Evaluation of XGB Model

          Accuracy: 0.99950
          Precision: 0.91597
          Recall: 0.76761
          F1-score: 0.83525

In [ ]:
```

5.2 PERFORMANCE EVALUATION, METRICS:

PRECISION

Precision is one indicator of a machine learning model's performance – the quality of a positive prediction made by the model. Precision refers to the number of true positives divided by the total number of positive predictions (i.e., the number of true positives plus the number of false positives)

Precision = True Positive/True Positive + False Positive

RECALL

A Recall is essentially the ratio of true positives to all the positives in ground truth. Recall towards 1 will signify that your model didn't miss any true positives, and is able to classify well between correctly and incorrectly labeling of cancer patients. What it cannot measure is the existence of type-I error which is false positives i.e the cases when a cancerous patient is identified as non-cancerous. A low recall score (<0.5) means your classifier has a high number of false negatives which can be an outcome of imbalanced class or untuned model hyperparameters. In an imbalanced class problem, you have to prepare your data beforehand with over/under-sampling or focal loss in order to curb FP/FN.

Recall = True Positive/True Positive + False Negative

F1-SCORE

The F-score, also called the F1-score, is a measure of a model's accuracy on a dataset. It is used to evaluate binary classification systems, which classify examples into 'positive' or 'negative'. The F-score is a way of combining the precision and recall of the model, and it is defined as the harmonic mean of the model's precision and recall. It is commonly used for evaluating information retrieval systems such as search engines, and also for many kinds of machine learning models, in particular in natural language processing. It is possible to adjust the F-score to give more importance to precision over recall, or vice-versa. Common adjusted F-scores are the F0.5-score and the F2-score, as well as the standard F1-score.

F1 Score = the 2*((precision*recall) / (precision + recall))

ACCURACY

Classification Accuracy is what we usually mean, when we use the term accuracy. It is the ratio of the number of correct predictions to the total number of input samples.

Accuracy=Number of Correct predictions/Total number of predictions made

Visualisation:

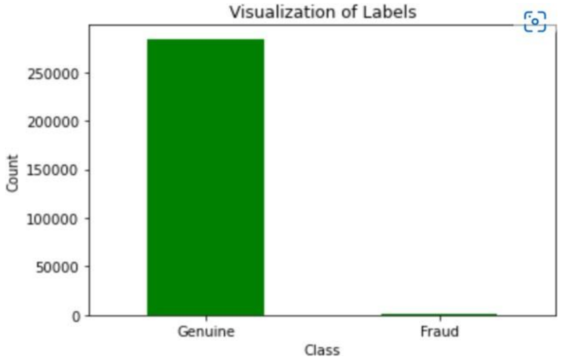


Fig 1 : Visualization of Labels

Amount per transaction by fraud class:

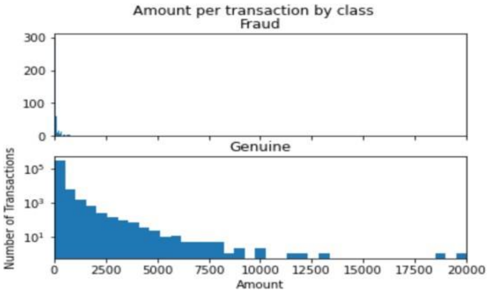


Fig 2: FRAUD VS GENUINE by Amount Class

Transactions per Time by Fraud class:

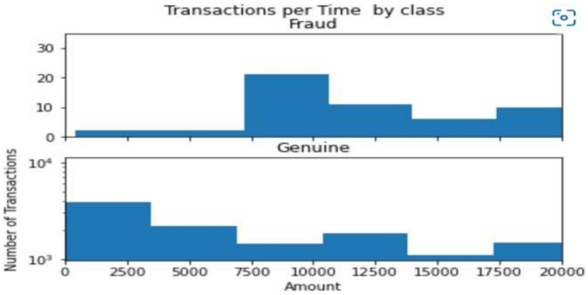


Fig 3: FRAUD VS GENUINE by Time Class

Heat map:

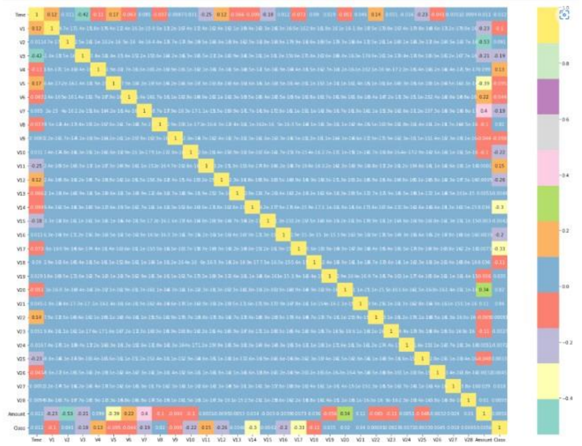


Fig 4:Heat Map

5.7 Logistic Regression:

The supervised learning approach includes logistic regression.It is based on the probability approach. It's used to calculate the probability of a binary response given one or more predictors. They can have a continuous or discrete nature. We adopt logistic regression as we want to identify or segregate some instances into segments. It aims to classify data only in binary form, such that, in 0 and 1 quantities, which leads to a situation in which a patient's diabetes status is labeled as positive or negative. The major aim of logistic regression is to find the best fit, which describes the connection between the target and regression models. Logistic regression is a computation tool for predicting outcomes.

$$\mathrm{Logit}(pi) = 1/(1+ \exp(-pi))$$

$$\ln(pi/(1-pi)) = Beta\_0 + Beta\_1*X\_1 + \ldots + B\_k*K\_k$$

Types of logistic regression

Binary logistic regression: In this approach, the response or dependent variable is dichotomous in nature—i.e. it has only two possible outcomes (e.g. 0 or 1)

Multinomial logistic regression: In this type of logistic regression model, the dependent variable has three or more possible outcomes; however, these values have no specified order.

Ordinal logistic regression: This type of logistic regression model is leveraged when the response variable has three or more possible outcome, but in this case, these values do have a defined order.

Main use of Logistic Regression:

Logistic regression models can help teams identify data anomalies, which are predictive of fraud. Certain behaviors or characteristics may have a higher association with fraudulent activities, which is particularly helpful to banking and other financial institutions in protecting their clients. SaaS-based companies have also started to adopt these practices to eliminate fake user accounts from their datasets when conducting data analysis around business performance.

Diagram for logistic Regression:

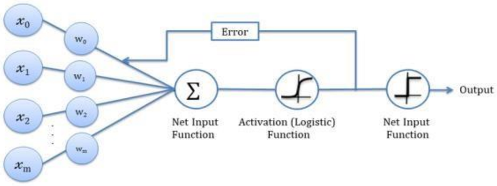
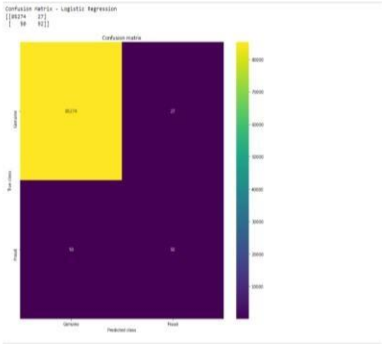


Fig 5:Logistic Regression Diagram

Confusion Matrix:



```
Confusion Matrix - Logistic Regression
[[85274    27]
 [   50    92]]
```

Fig 6:Confusion Matrix for Logistic regression

The main metrics we'll look at :

- Accuracy-**0.99910**
- Precision-**0.77311**
- Recall-**0.64789**
- F1-Score-**0.70498**

5.8 Decision Tree:

The decision tree is a simple classification technique,and is the supervised machine learning technique. A decision tree is a strategy for partitioning a given dataset into two or more test data periodically. It's a learning algorithm that would be guided. When the objective property is classified, this method is taken. The procedure is based on input features is expressed by a design with a tree-like structure called a decision tree. Any types, graphs, texts, discrete, continuous, and so on are among the input variables.

A decision tree is a diagram that depicts the various outcomes of a set of related options. It enables a person or organization to compare and contrast several options based on their prices, probabilities, and advantages. They can be used to spark informal debate or to create an algorithm that mathematically predicts the optimal option.

A decision tree usually begins with a single node and branches out into different outcomes. Each of those results leads to new nodes, each of which leads to new possibilities. It takes on a tree-like shape as a result of this.

Nodes are divided into three categories: chance nodes, decision nodes, and end nodes. The probability of certain outcomes are represented by a chance node, which is represented by a circle. A decision node, represented by a square, represents a decision that needs to be taken, while an end node represents the decision path's final consequence.

Diagram for Decision Tree:

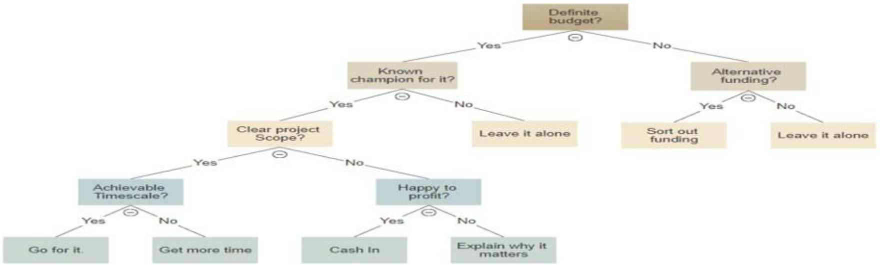
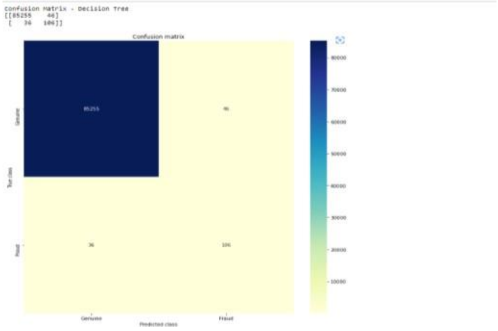


Fig 7 :Decision Tree Diagram

This can be one random example for the Decision Tree and how will the approach be.

Confusion Matrix:



```
Confusion Matrix - Decision Tree
[[85255    46]
 [   36   106]]
```

Fig 8:Confusion Matrix for Decision Tree

The main metrics we'll look at are :

- Accuracy-**0.99904**
- Precision-**0.69737**
- Recall-**0.74648**
- F1-Score**-0.72109**

Random Forest:

It's an ensemble learning method that can be used for classification and regression.In simple words we can say that collection of decision trees is called Random Forest. It's a well-known ensemble learning method. By lowering variation,the random forest increases the performance

of decision trees. It proceeds by training a vast number of decision trees and then delivering the mode of the classes, categorisation, or overall average prediction (regression) of the individual trees as a class.

We will use the sklearn module for training our random forest regression model, specifically the RandomForest Regressor function. The RandomForest Regressor documentation shows many different parameters we can select for our model.max_depth — this sets the maximum possible depth of each treemax_features — the maximum number of features the model will consider when determining a split bootstrap — the default value for this is True, meaning the model follows bootstrapping principles (defined earlier)

max_samples — This parameter assumes bootstrapping is set to True, if not, this parameter doesn’t apply. In the case of True, this value sets the largest size of each sample for each tree.

Diagram for Random Forest:

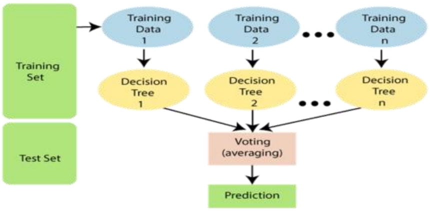
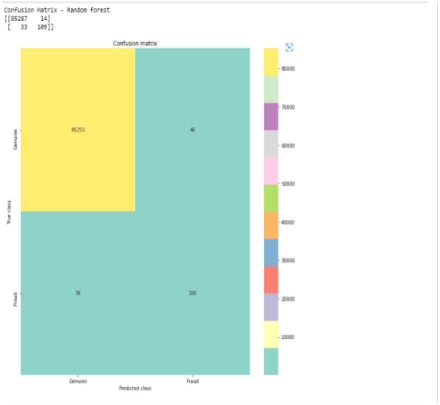


Fig 9 :Random Forest Diagram

In Simple words we can say that collection some random no.of Decision trees is called as Random Forest

Confusion Matrix :



```
Confusion Matrix - Random forest
[[85287    14]
 [   33   109]]
```

Fig 10 :Confusion Matrix for Random Forest

The main metrics we'll look at are :

- Accuracy-**0.99945**
- Precision-**0.88618**
- Recall-**0.76761**
- F1-Score-**0.82264**

5.10 SUPPORT VECTOR MACHINE

SVM categorizes data points by mapping them to a high-dimensional feature space, even when the data is not otherwise linearly separable. After finding a separator between the categories, the data are converted so that the separator can be drawn as a hyperplane. Following that, fresh data features can be utilized to predict which category a new record should belong to.

Diagram for SVM:

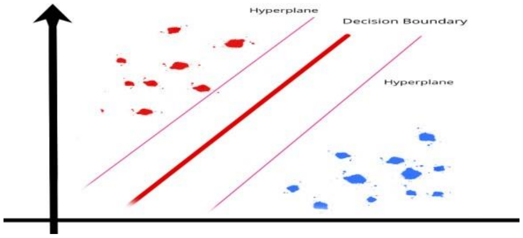
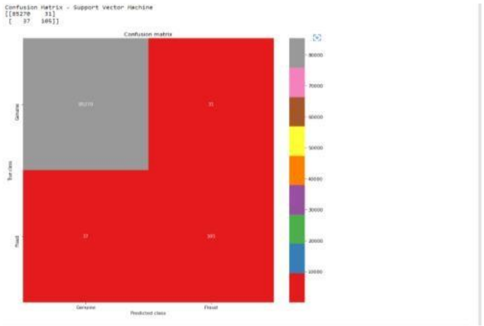


Fig 11 :Support Vector Machine Diagram

Confusion Matrix:



```
Confusion Matrix - Support Vector Machine
[[85270    31]
 [   37   105]]
```

Fig 12 :Confusion Matrix for Support Vector Machine

The main metrics we'll look at :

- **Accuracy-0.99920**
- **Precision-0.77206**
- **Recall-0.73944**
- **F1-Score-0.75540**

5.11 KNN

K-Nearest Neighbor (KNN) is a basic Machine Learning algorithm that handles classification and regression. It's also called a lazy learner algorithm since it doesn't understand the learning algorithm right away; instead, it saves the data and uses it to identify later.For the prediction of a new data point, the algorithm determines the closest data points in the training data set (its nearest neighbors). The number of nearest neighbors, K, is always a positive integer in this case. A neighbor's value is chosen from a list of classes. This algorithm can be used as a classifier or regression model

Diagram for KNN:

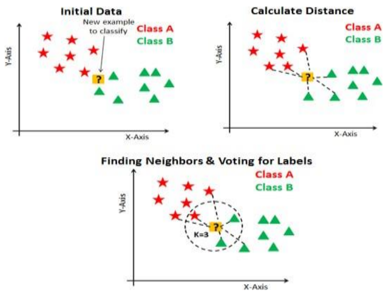
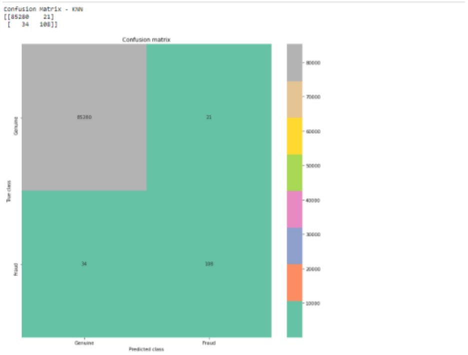


Fig 13 :K-Nearest Neighbor Diagram

Confusion Matrix:



```
Confusion Matrix - KNN
[[85280    21]
 [   34   108]]
```

Fig 14 :Confusion Matrix for K-Nearest Neighbor

The main metrics we'll look at :

- **Accuracy-0.99936**
- **Precision-0.83721**
- **Recall-0.76056**
- **F1-Score-0.79705**

5.12 XG BOOST:

Extreme gradient boosting (XGBoost) is very well known as gradient boosting methodology that enhance the efficiency and accuracy of tree-based (sequential decision trees) machine learning algorithms.This is the important often widely used algorithm in applied machine learning. XGBoost is classed as a boosting mechanism in Ensemble Learning. To maximize prediction accuracy, ensemble learning integrates several models into a collection of predictors. By applying a load to the models in the boosting technique, the inaccuracy generated by previous models is considered to be sorted by successive model.

Diagram for XG Boost:

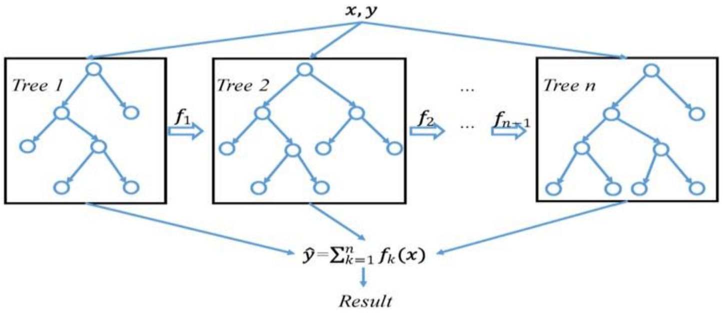
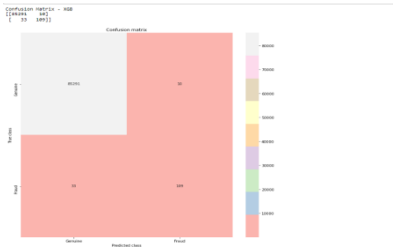


Fig 15 :XG-Boost Diagram

Confusion Matrix



```
Confusion Matrix - XGB
[[85291    10]
 [   33   109]]
```

Fig 16 :Confusion Matrix for XG-Boost

The main metrics we'll look at :

- **Accuracy-0.99950**
- **Precision-0.91597**
- **Recall-0.76761**
- **F1-Score-0.83525**

5.3 RESULTS:

Efficiency comparison of models:

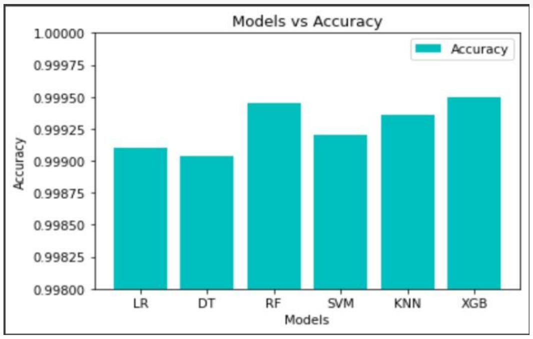


Fig 17: ACCURACY VS MODELS

Fig 2 shows the analysis of efficiency of 6 machine learning models based on accuracy.XG-Boost is the most accurate one with whopping accuracy score of 99.950%, then followed by Random Forest with an accuracy of 99.945%, then KNN with an accuracy of 99.936% ,SVM with an accuracy of 99.920% ,Logistic Regression with an accuracy of 99.910%,Decision tree with an accuracy of 99.904%.

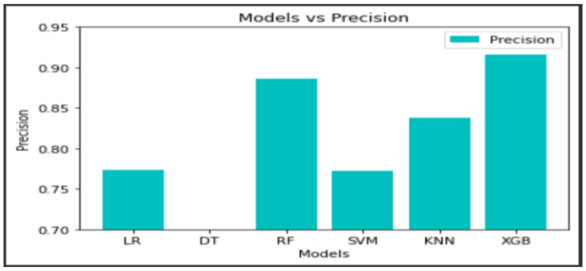


Fig 18: PRECISION VS MODELS

Fig2 shows the analysis of efficiency of 6 machine learning models based on precision.XG-Boost is the most accurate one with whopping precision score of 83.721 %, then followed by Random Forest with an precision of 88.618%, then KNN with an precision of 77.206% ,SVM with an precision of 99.920% ,Logistic Regression with an precision of 77.311%,Decision tree with an precision of 69.737%.

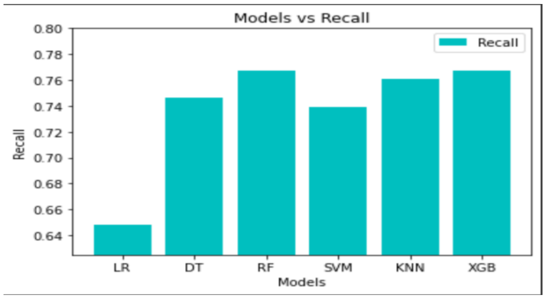


Fig 19: RECALL VS MODELS

Fig2 shows the analysis of efficiency of 6 machine learning models based on recall. Random Forest and XG-Boost are the most accurate one with whopping recall score of 76.761 %, then followed by KNN with the recall score of 76.056%,Decision tree with the recall score of 74.648%,SVM with the recall score of 77.206%,Logistic Regression with the recall score of 64.789%.

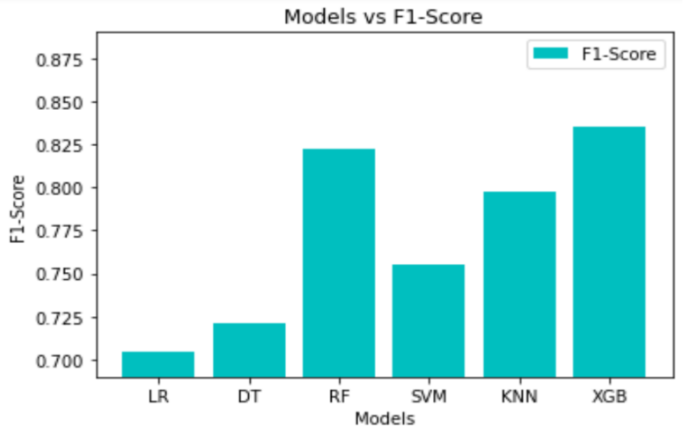


Fig 20: F1-SCORE VS MODELS

Fig2 shows the analysis of efficiency of 6 machine learning models based on F1-Score. XG-Boost is the most accurate one with whopping F1-score of 83.525%, then followed by Random Forest with an F1-Score of 82.264%, then KNN with an F1-Score of 79.705% ,SVM with an F1-Score of 75.540% ,Decision tree with an F1-Score of 72.109%,Logistic Regression with an F1-Score of 70.498%.

TABLE 1 : PRECISION,RECALL,F1-SCORE,ACCURACY OF CLASSIFIERS

MODEL/ METRIC	**LR**	**DT**	**RF**	**SVM**	**KNN**	**XGB**
ACCURACY	0.99910	0.99904	0.99945	0.99920	0.99936	0.99950
PRECISION	0.77311	0.69737	0.88618	0.77206	0.83721	0.91597
RECALL	0.64789	0.74648	0.76761	0.73944	0.76056	0.76761
F1-SCORE	0.70498	0.72109	0.82264	0.75540	0.79705	0.83525

5.3 RESULTS

An article that discusses how a new algorithm called XG boost is developed intended to help in credit card fraud detection. The article talks about what the algorithm is, how it works and provides a result of if it would get best accuracy for credit card fraud detection.

XG boost is an algorithmic approach for identifying unique fraudulent transactions from a set of inputs of data. They can be inputted as many data sets as the size allows and classified based on their "Family" (e.g., different groups). For example, personal information may be inputted into one group, purchase information into another group, etc. The algorithm scans each input data set and mines out the unique characteristics of fraudulent transactions from them. It then computes several output values based on the input data sets that show if a specific transaction is fraudulent or not. For example, for one transaction, it may show this is a good transaction and a second transaction as not good. This algorithm helps in managing fraud at ATMs as it can detect when certain features happen at certain time slots during a day and where they happened to help in detecting fraud patterns rather than being too broad.

The XG boost algorithm was first introduced in 2007 by Dr. Olga Russakovsky in her dissertation titled: "Fraud detection with machine learning". More recently, the team of Dr. Wang worked on a new algorithm called XG Boost and proposed the idea of mining for patterns in data rather than trying to identify the fraud at ATM.

The algorithm has been evaluated by Dr. Sidney Wang on a set of 10,000 samples from online shopping sites. The algorithm was able to detect fraud with an accuracy of 0.910 and an average precision of 0.999 which was better than heuristic approach even though they were trained using different algorithms. Amongst other testing methods, it had best performance in F-measure compared with Bayesian classifiers, SVM classifiers, Naive Bayes, support vector machines and boosting techniques across all the tested datasets.

CHAPTER - 6

6.1 CONCLUSION

In this project, we studied applications of machine learning like Logistic Regression, KNearest Neighbour, Support Vector Machine, Decision Tree, random Forest, XG-Boost Machine Learning Algorithms and shows that it proves accurate in detecting fraudulent transaction and minimizing the number of false alerts. Supervised learning algorithms are a novel one in this literature in terms of application domain. If these algorithms are applied into a bank credit card fraud detection system, the probability of fraud transactions can be predicted soon after credit card transactions. And a series of anti-fraud strategies can be adopted to prevent banks from great losses and reduce risks. The objective of the study was taken differently than the typical classification problems in that we had a variable misclassification cost. Precision, recall, f1-score, support and accuracy are used to evaluate the performance for the proposed system. By comparing all the methods, we found that XGBoost is better than all the other algorithms with 99.95% accuracy.

6.2 REFERENCES

1. Credit Card Fraud Detection –
https://www.youtube.com/watch?v=jCoF1rMs_0s

2.Code Link –
https://dataaspirant.com/credit-card-fraud-detection-classification-algorithms

3.Github Link –
https://github.com/sahidul-shaikh/credit-card-fraud-detection

4. S. H. Projects and W. Lovo, ―JMU Scholarly Commons Detecting credit card fraud : An analysis of fraud detection techniques,‖ 2020.

5. C. Reviews, ―a Comparative Study : Credit Card Fraud,‖ vol. 7, no. 19, pp. 998–1011, 2020.

6. M. Kanchana, V. Chadda, and H. Jain, ―Credit card fraud detection,‖ Int. J. Adv. Sci. Technol., vol. 29, no. 6, pp. 2201–2215, 2020, doi: 10.17148/ijarcce.2016.5109.

7. A. RB and S. K. KR, ―Credit Card Fraud Detection Using Artificial Neural Network,‖ Glob. Transitions Proc., pp. 0–8, 2021, doi: 10.1016/j.gltp.2021.01.006.

8. X. Yu, X. Li, Y. Dong, and R. Zheng, ―A Deep Neural Network Algorithm for Detecting Credit Card Fraud,‖ Proc. - 2020 Int. Conf. Big Data, Artif. Intell. Internet Things Eng. ICBAIE 2020, pp. 181–183, 2020, doi: 10.1109/ICBAIE49996.2020.00045.

9. S. Bagga, A. Goyal, N. Gupta, and A. Goyal, ―Credit Card Fraud Detection using Pipeling and Ensemble Learning,‖ Procedia Comput. Sci., vol. 173, pp. 104–112, 2020, doi: 10.1016/j.procs.2020.06.014.