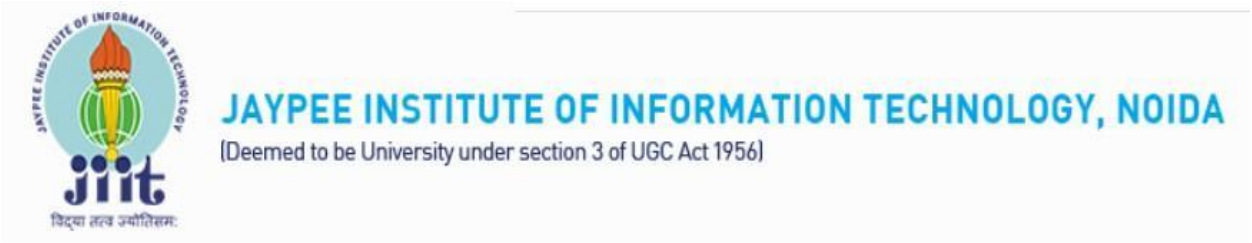


Performance Analysis Of Different Machine Learning Algorithms On Credit Card Fraud Detection

Submitted By

Abhijot Singh 19103180

Vansh Sachdeva 19103194

Ayush Jaiswal 19103185

Under Supervision Of

Dr Amanpreet Kaur

Assistant Professor (Senior Grade)

Department Of Computer Science And Information Technology,

Jaypee Institute of Information Technology, Noida, India

Email:19103180@mail.jiit.ac.in, 19103194@mail.jiit.ac.in

19103185@mail.jiit.ac.in

Acknowledgement

The completion of any inter-disciplinary project depends upon cooperation, coordination, and combined efforts of several sources of knowledge. We are grateful to **Dr Amanpreet Kaur** for her willingness to give us valuable advice and direction whenever we approached her with a problem.

We are thankful to her for providing us with immense guidance for this project. We would also like to thank our College authorities for allowing us to pursue our project in this field.

Supervisor: **Dr Amanpreet Kaur.**

Names:

Abhijot Singh(19103180)

Ayush Jaiswal(19103185)

Vansh Sachdeva(19103194)

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and beliefs, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma from a university or other institute of higher learning, except where due acknowledgement has been made in the text.

Place: Noida
Date: May 18, 2022

Name: Vansh Sachdeva
Enrolment No.:19103194
Name: Abhijot Singh
Enrolment No:1910381
Name: Ayush Jaiswal
Enrolment No.: 19103185

CERTIFICATE

This is to certify that the work titled “Performance Analysis Of Machine Learning Algorithms On Credit Card Fraud Detection” submitted by Vansh Sachdeva, Abhijot Singh And Ayush Jaiswal of B.Tech of Jaypee Institute of Information Technology, Noida has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of any other degree or diploma.

Digital Signature of Supervisor

Dr Amanpreet Kaur

ASSISTANT PROFESSOR (SR. GRADE)
Date

Abstract

Machine learning (ML) is a scientific study of algorithms and statistical models that computer systems use to perform specific tasks that are not explicitly programmed. Learn algorithms in many of the applications we use every day. Every time a web search engine like Google is used to crawl the internet, one of the reasons it works so well is because of the learning algorithms that have learned how to rank web pages. These algorithms are used for a variety of purposes such as data mining, image processing, and predictive analytics. The main advantage of machine learning is that once an algorithm learns how to process data, it can do its job automatically. This white paper provided an overview and future outlook for a wide range of applications for machine learning algorithms. This project aims to statistically analyze different machine learning and deep learning algorithms, and compare and contrast their performance in credit card fraud detection.

The algorithms used are -

Artificial Neural Networks(ANN), Support Vector Machine (SVM), Kth Nearest Neighbour (KNN), Decision Tree, Logistic Regression, Random Forest., and Adaboost.

The usage of various imbalance learning techniques has been proposed for getting unbiased results without any key details being missed out or skipped because of the highly unbalanced datasets being available and present in real-world applications.

The results of these algorithms are based on accuracy, precision, recall, and F1-score. Then, the ROC curve is plotted based on the confusion matrix. Finally, these algorithms are compared and the algorithm that has the greatest accuracy, precision, recall, and F1 score is considered the best algorithm that is used to detect fraud.

Keywords: Imbalance Learning, Neural Networks, Support Vector Machine, KNN, Decision tree, Random Forest, Logistic Regression.

Link Of The Project's Github Repository:
https://github.com/MrArthor/Statistical-Analysis-Of-Machine-Learning-Algorithms

Contents Of Report

Problem Statement:-

Solving the problem of imbalance data by using various imbalance learning techniques like Undersampling and Oversampling And Hybrid data sampling and further performing Comparison of performance of different machine learning algorithms on credit card fraud detection.
Performance metrics for comparison used are Accuracy, Recall, Precision and F1-Score of results

Objective Of The Project: -

The objective of the project is to solve the imbalance data problems and perform a Comparison between the supervised learning and deep learning algorithm outperformed based on accuracy for data of credit card frauds of duration between 1st Jan 2019 and 31st Dec 2020 which was generated via using Sparkov.

List Of Abbreviations

Abbreviation	Full-Form
KNN	Kth Nearest Neighbour
ANN	Artificial Neural Networks
SVM	Support Vector Machine
DT	Decision tree
ELM	Extreme learning machine
MLP	Multilayer perceptron
SOAP	Simple object access protocol
REST	Representational state transfer
NRBE	Non-overlapped risk-based bagging ensemble
RFA	Random forest algorithm
KSH	Kernel-based supervised hashing
TBC	To Be Calculated

Related Work

Some of the related study made by various researchers is described in this section.

- Mohamad Zamini and Golamali montazar proposed an independent charge card Fraud area structure using autoencoders based gathering. They used three mystery layers and k means is used for packing and took a stab at the European dataset which played out all around and appeared differently in relation to other existing structures [1]

- Ishan Sohony, Rameshwar Pratap, and Ullas Nambiar proposed a gathering learning approach for Credit Card blackmail distinguishing proof as the extent of deception to standard trade is somewhat appropriate. They saw that the Random forest area is generally suitable to give higher accuracy and neural associations for recognizing the coercion events. They moreover attempted various things with the tremendous authentic Visa trades. Outfit learning is mixes of Random forest area and neural associations[4]
- Pawan Kumar and Fahad Iqbal made a review on all methods used to recognize Credit Card coercion area using AI computations and survey the display with the estimations. Enormous heaps of assessments occurred over this space. They say that there is a need to use a more successful structure that performs well in every situation. [5].
- Altab Altar Taha and Sareef Jameel Mulberry depicted that up-gradation in internet-based business and correspondence development has made charge card use an all the more notable strategy for portion and the deception related with trades is similarly extending. They have used the further developed light point helping machine, where Bayesian-based hyper-limit progressions are merged to tune with the limits of the light tendency supporting machine (LightGBM). In this procedure, they used two plans of genuine open datasets comprising of phoney and non-counterfeit trades. Taking into account the assessment with various strategies, their proposed system beat similar to precision. The proposed structure conveys a precision of 98.40%, a district under authority working characteristics twist (AUC) of 92.88%, a Precision of 97.34% and an F1 score of 56.95%. [6]
- Debachudamani Prusti and Santhnu Kumar Rath arranged an application with applied AI draw near, for instance, Decision tree (DT), k-nearest estimation (kNN), Extreme learning machine (ELM), Multilayer perceptron (MLP) and support vector machine (SVM) to recognize the precision in coercion ID. They proposed a model by hybridizing the DT, SVM and kNN strategies. They used two web-based shows, for instance, essential thing access show (SOAP) and Representational state move (REST) for the powerful exchange of data across various heterogeneous stages. They contemplated five AI computation results reliant upon the precision metric. SVM performed better contrasted with various estimations by 81.63% yet the hybrid system proposed by them had higher precision of 82.58%. [7]

- Phuong Hanh Tran, Kim Phuc Tran, Truong Thu Huong, Cédric Heuchenne, Phuong HienTran, and Thi Minh Huong Le research depicts that in past a few years' charge card distortion extended little by little. Various techniques were performed using AI estimations to recognize the distortion trades and square them. They introduced two new data-driven strategies, which use the best peculiarity methodology for distortion trade-in Credit Card trades. The two unique ways are segment limit decision and T2 control chart. [10]
- E. Esenogho, I. D. Mienye, T. G. Swart, K. Aruleba and G. Obaido, "A Neural Network Ensemble With Feature Engineering for Improved Credit Card Fraud Detection," proposed a The performance of the proposed approach is benchmarked against the following algorithms: support vector machine (SVM), multilayer perceptron (MLP), decision tree, traditional AdaBoost, and LSTM. The experimental results show that the classifiers performed better when trained with the resampled data, and the proposed LSTM ensemble outperformed the other algorithms by obtaining a sensitivity and specificity of 0.996 and 0.998, respectively
- Konduri Praveen Mahesh and Shaik Ashar Afrouz and Anu Shaju Areeckal in Detection of fraudulent credit card transactions: A comparative analysis of data sampling and classification techniques proposed usage of various imbalance learning techniques and data preprocessing before using them as training sets in machine learning algorithms for getting better-unbiased results.
- Praveen Mahesh, K., Ashar Afrouz, S., & Shaju Areeckal, A. (2022). Detection of fraudulent credit card transactions: A comparative analysis of data sampling and classification techniques developed a behaviour analysis model based on SVM and random forest for detecting and preventing credit card frauds based on the behaviour analysis of the users and training model sets

Proposed Methodology

Data Preprocessing

What is Data Preprocessing

It is the process of transforming raw data into a usable format. This is also an important step in data mining because you cannot work with raw data. Before applying machine learning or data mining algorithms, you need to check the data quality.

Why is Data preprocessing important?

Data preprocessing is primarily used to check data quality. You can check the quality as follows

- Accuracy: Check if the entered data is correct.
- Integrity: Check if the data is available.
- Consistency: Checks if the same data is stored in all matching or unmatched locations.
- Timeliness: The data needs to be updated correctly. Reliability: The data must be reliable.
- Interpretability: Data comprehension.

Major Tasks in Data Preprocessing:

1. Data cleaning
2. Data integration
3. Data reduction

Data cleaning:

Data cleaning is the process of removing incorrect, incomplete, and inaccurate data from records and replacing missing values.

There are several techniques for data cleansing

- **Handling missing values:**

the default values such as "Unavailable" and "NA" are used to replace the missing value. Or the missing values are inserted manually, but this is not recommended in case datasets are large. If data is normally distributed, the average of the attributes is used to replace the missing values. For nonnormal distributions, use the median of the attribute. If regression or decision tree algorithms are used then replace the missing values with the most likely values.

- **Noisy**: Noisy usually means a random error or contains unwanted data points. Here are some ways to handle noisy data:

- **Binning**: This method is used to smooth or process noisy data. The data is sorted first, then the sorted values are separated and stored in the form of bins. There are three ways to smooth the data in the bin. Smoothing with the bin average method: This method

replaces the values in the bin with the mean of the bin. Smoothing by median bin: This method replaces the value in the bin with the median. Smoothing with bin limit: This method gets the minimum and maximum bin values and replaces the value with the nearest limit.

- **Regression**: It is used to smooth the data and is useful for handling data when there is unnecessary data. For analysis, objective regression helps determine the appropriate variables for the analysis.
- **Clustering**: It is used to find outliers and group data. Clustering is commonly used in unsupervised learning.

Data reduction: This procedure facilitates withinside the discount of the extent of the information, making the evaluation less complicated but producing identical or nearly identical results. This discount additionally reduces lessened garage space.

There are some of the techniques in data reduction are

Dimensionality reduction

Numerosity reduction

Data compression.

Dimensionality reduction: Due to the large data size, real applications require this process. Random variables or attributes are reduced to reduce the dimensions of the dataset. Combines and merges data attributes without losing the original properties. This also contributes to the reduction of storage space and reduces calculation time. If the data is high-dimensional, a problem called the "curse of dimensionality" arises.

Numerosity Reduction: In this method, the representation of the data is made smaller by reducing the volume. There will not be any loss of data in this reduction.

Data compression: The compressed form of data is called data compression. This compression can be lossless or lossy. When there is no loss of information during compression it is called lossless compression. Whereas lossy compression reduces information but it removes only the unnecessary information.

Data Transformation:

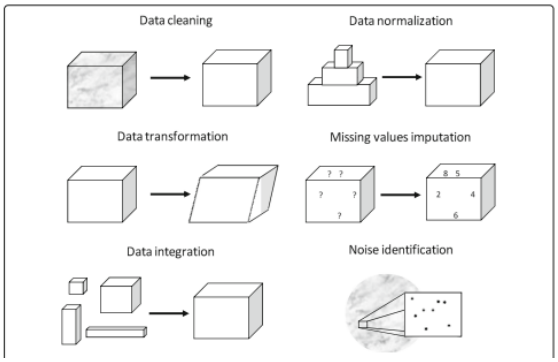
Changing the format or structure of data is called data conversion. This procedure can be simple or complex depending on your requirements. There are several ways to convert data.

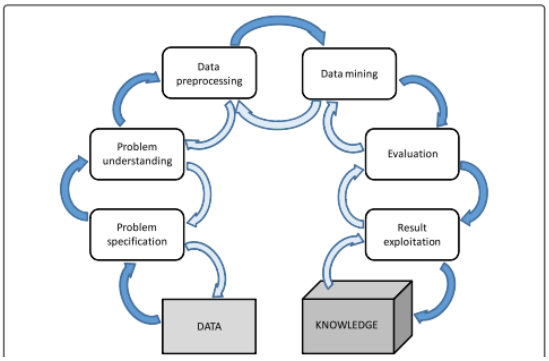
Smoothing: With the help of algorithms, we can remove noise from the dataset and helps in knowing the important features of the dataset. By smoothing we can find even a simple change that helps in prediction.

Aggregation: In this method, the data is stored and presented in the form of a summary. The data set which is from multiple sources is integrated into with data analysis description. This is an important step since the accuracy of the data depends on the quantity and quality of the data. When the quality and the quantity of the data are good the results are more relevant.

Discretization: The continuous data here is split into intervals. Discretization reduces the data size. For example, rather than specifying the class time, we can set an interval like (3 pm-5 pm, 6 pm-8 pm).

Normalization: It is the method of scaling the data so that it can be represented in a smaller range. Example range from -1.0 to 1.0.





Imbalanced Classification Of Data

What is imbalanced Data Handling?

Today, all machine learning practitioners dealing with binary classification problems must have encountered this typical unbalanced dataset situation. This is a typical scenario for many useful business problems such as fraud detection, spam filtering, rare disease detection, hardware failure detection, and so on. The number of data points in the negative class (majority class) is much higher than in the positive class (minority class).

In general, minority / positive classes are of interest and we tend to aim for the best results in that class. Failure to preprocess imbalanced data will reduce the performance of the classifier model. Most predictions are based on the majority class, treating minority class features as data noise and ignoring them. This will increase the distortion of the model.

The Accuracy Paradox

Suppose you are working on the issue of fraud detection for credit card fraud detection. Due to these issues, it is generally found that 99 out of 100 claims are not fraudulent and 1 is fraudulent. Therefore, the binary classification model does not have to be a complex model to predict all

results as 0. This means achieving 99% accuracy, not fraud. If the class distribution is distorted, the accuracy metric is clearly biased and unfavourable.

Dealing with Imbalanced Data Resampling data is one of the most commonly preferred approaches to deal with an imbalanced dataset.
There are broadly two types of methods for this
Oversampling — Duplicating samples from the minority class
Undersampling — Deleting samples from the majority class.

- **Oversampling: -** For unbalanced classes, randomly increase the number of observations that are just copies of existing samples. This will ideally give you enough samples to play with. Oversampling can lead to the overfitting of training data.
  for example, randomly duplicates a minority class sample and adds it to the training dataset. Samples from the training dataset are randomly selected using surrogates. This means that you can select multiple minority class examples to add to a new "more balanced" training set. They are selected from the original training dataset, added to the new training dataset, and then reverted or "replaced" to the original dataset so that they can be reselected. This technique is useful for machine learning algorithms that are plagued by biased distributions where multiple overlapping examples of a particular class can affect the fitting of the model. This may include algorithms that iteratively learn the coefficients, such as artificial neural networks that use stochastic gradient descent. It can also affect the model for finding the right partition for your data such as the Support vector machine and decision tree.
- Undersampling is a technique to balance unequal data sets by keeping all data in the minority class and reducing the size of the majority class. This is one of many techniques that data scientists can use to extract more accurate insights from initially unbalanced data sets. Undersampling is a technique to balance unequal data sets by keeping all data in the minority class and reducing the size of the majority class. This is one of many techniques that data scientists can use to extract more accurate insights from initially unbalanced data sets. While it has limitations, such as potentially important information loss, it is still a common and important skill for data scientists.

In most cases, oversampling takes precedence over the undersampling technique. The reason is that undersampling tends to remove instances from data that may contain important information.
In this project we have used one method of each type i.e. undersampling, oversampling and hybrid data sampling methods.

SMOTE: Synthetic Minority Oversampling Technique

SMOTE (Synthetic Minority Oversampling Technique) is one of the most commonly used oversampling methods to solve the problem of imbalance. The purpose is to equalize the class distribution by replicating the minority class example and randomly increasing it. SMOTE synthesizes new minority instances between existing minority instances. SMOTE is an oversampling technique that produces synthetic samples of minority classes. This algorithm helps overcome the problem of overfitting caused by random oversampling. The focus is on the functional space that creates new instances using interpolation between adjacent positive instances.

Pseudo Code

Step 1: Setting the minority class set A, the k-nearest neighbours of x are obtained by calculating the Euclidean distance between x and every other sample in set A.

Step 2: The sampling rate N is set according to the imbalanced proportion. N examples (i.e x1, x2, …xn) are randomly selected from its k-nearest neighbours.

Step 3: For (k=1, 2, 3…N), the following formula is used to generate a new example in which rand(0, 1) represents the random number between 0 and 1.

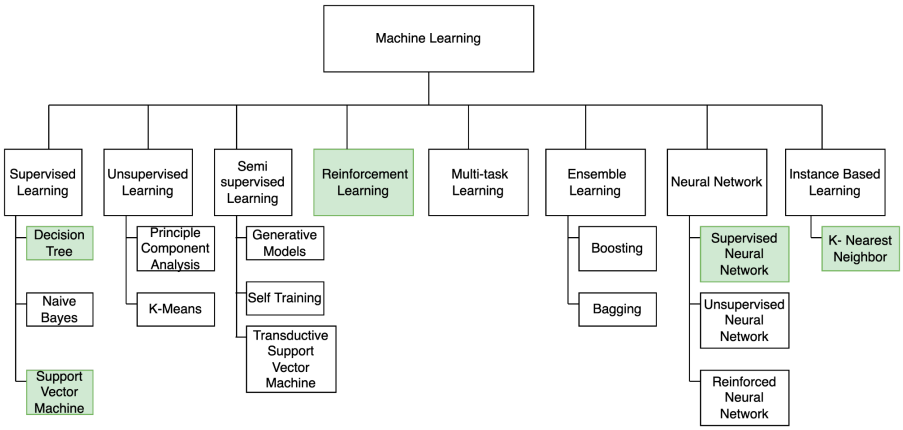
SMOTE + Tomek Links

- Hybridization techniques include a combination of undersampling and oversampling techniques. This is done to optimize the performance of the sample classifier model created as part of these techniques. SMOTE + TOMEK is one such hybrid technique aimed at cleaning up overlapping data points for each class distributed in the sample space. After being oversampled by SMOTE, class clusters can invade each other's spaces. As a result, the classifier model is overfitted. This method combines the functionality of SMOTE to generate synthetic data for a minority class with the functionality of a Tomek link to remove data identified as a Tomek link from the majority class.

Steps:-

- (**Start of SMOTE**) Choose random data from the minority class.
- Calculate the distance between the random data and its k nearest neighbours.
- Multiply the difference with a random number between 0 and 1, then add the result to the minority class as a synthetic sample.
- Repeat step number 2–3 until the desired proportion of minority class is met. (**End of SMOTE**)
- (**Start of Tomek Links**) Choose random data from the majority class.
- If the random data’s nearest neighbour is the data from the minority class (i.e. create the Tomek Link), then remove the Tomek Link.

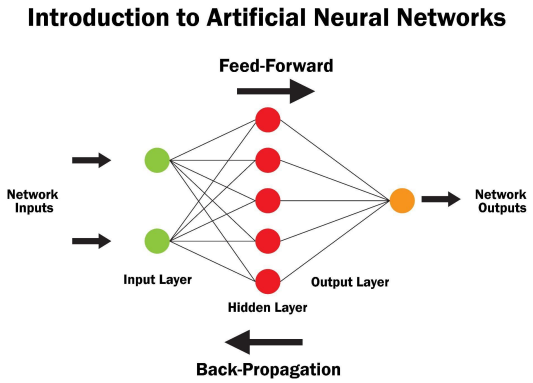
Algorithms Used



Artificial Neural Networks(ANN)

The multi-layer perceptron is a neural structure with several levels, each of which plays out a distinct limit. As the model's multidimensional design develops, the number of layers expands in lockstep.

Because many relationships among information and yield are not direct, ANN has the ability to learn and show non-immediate and complex associations. Following planning, ANN can generate covered relationships from subtle data, which is then summarised. Unlike many AI models, ANN has no dataset restrictions, such as the requirement that data be distributed in a Gaussian manner.



Support Vector Machine (SVM)(SVM) is a synchronised AI estimating technique that can be used to solve both gathering and backslide problems. SVM plots all data as a point in n-dimensional space, with the value of each portion representing the value of a single facilitator. The hyper-plane that isolates the two classes is then obtained and a gathering is

performed. The benefits of SVM include its great robustness in light of its reliance on assist vectors and the fact that it is unaffected by outliers. SVM can handle numerical assumption difficulties. The problem with SVM is that it is a black box technique that is prone to overfitting and requires extensive calculations.

Kth Nearest Neighbour (KNN) KNN can be used for both requests and backslide farsighted issues. In any case, it is even more commonly used in gathering issues in the business. KNN works on a standard expecting every datum directly falling in close to each other is falling in a comparative class. All things considered, it arranges another data point subject to resemblance.

KNN Pseudocode

1. Load the training and test data
2. Choose the value of K

For each point in test data:

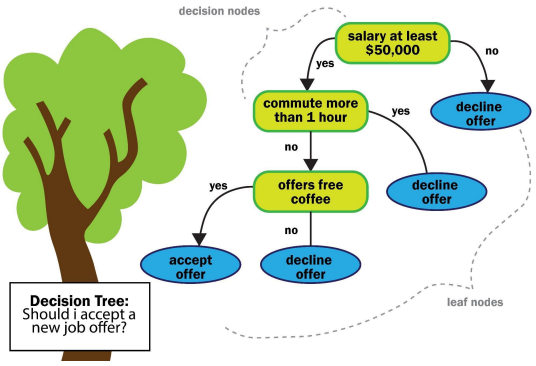
- find the Euclidean distance to all training data points
- store the Euclidean distances in a list and sort it
- choose the first k points
- assign a class to the test point based on the majority of classes present in the chosen points

3. End

- **Decision Tree: -**Decision Trees (DT) are a non-parametric administered learning technique for relapse detection and characterisation. The goal is to create a model that predicts the value of an objective variable using simple choice rules derived from the data. It starts from the root of the tree, determining the quality decided by this hub, and then drops down the tree limb corresponding to the value of the feature. It checks to see if the condition is genuine, and if it is, it proceeds to the next hub related to that option. This cycle is then repeated for the next hub's subtree.

Decision Tree Pseudocode: -

1. Begin the tree with the root node, says S, which contains the complete dataset.
2. Find the best attribute in the dataset using the Attribute Selection Measure (ASM).
3. Divide the S into subsets that contain possible values for the best attributes.
4. Generate the decision tree node, which contains the best attribute.
5. Recursively make new decision trees using the subsets of the dataset created in step 3 Continue this process until a stage is reached where you cannot further classify the nodes and call the final node as a leaf node



- **Logistic Regression** Calculated relapse is a method of illustrating the probability of a discrete result given a set of data. Multinomial strategic relapse can reveal circumstances when numerous discrete outcomes are possible. Calculated relapse is a useful investigative strategy for characterization concerns, such as deciding whether another sample would fit better into a class. Instead of fitting a straight line or hyperplane, the calculated relapse model uses the strategic capacity to press the yield of a direct condition between 0 and 1. For parallel and straight grouping problems, strategic relapse is a simple and effective strategy. The computed relapse model is a measurable two-factor characterisation technique that can be summarised as a multiclass grouping.

- **Random Forest** A Random Forest is an AI technique for dealing with relapse and order difficulties. It employs collecting realising, a method of combining multiple classifiers to provide answers to complex questions. Packing is a meta-calculation that works with AI systems' precision. It makes predictions by calculating the normal or mean of the yield from various trees. The accuracy of the outcome improves as the number of trees grows. The constraints of a choice tree computation are destroyed in irregular woods. It reduces dataset overfitting and increases accuracy. Because of its simplicity and versatility, it is likely the most often used algorithm. It could be used in ML for both classification and regression problems.

Random Forest Pseudocode: -

1. Randomly select K features from total m features where k << m
2. Among the K features, calculate the node d using the best split point
3. Split the node into daughter nodes using the best split
4. Repeat the a to c steps until l number of nodes has been reached
5. Build forest by repeating steps a to d for “n” number times to create n number of

- **AdaBoost Classification Algorithm** AdaBoost is an ensemble machine learning approach. A single-level decision tree, or one-part decision tree, is the most prevalent algorithm employed in AdaBoost. Decision stumps is another name for these trees. This

technique creates a model by equally weighting all data points. Then give the misclassified points a larger weight. All points with high weights will be heavier in the next model. Train your model until a Lowe mistake appears. Any machine learning algorithm can benefit from AdaBoost. These are categorization models that outperform chance. During the data training period, make "n" decision trees. When the first decision tree/model is constructed, the first model's misclassified records are used. This process continues until you specify the number of basic learners to create. Remember that all boosting techniques allow you to play records

- **Gradient Boosting** One of the most powerful methods in machine learning is the gradient boosting technique. Gradient boosting is one of the boosting strategies that is used to reduce the model's bias error. We are unable to discuss the gradient boosting algorithm's basic estimator. The Gradient Boost algorithm's base estimator is fixed, namely Decision Stump. We can modify the n estimator of the gradient boosting algorithm, just like AdaBoost. The default value of n estimator for this algorithm is 100 if the value of n estimator is not specified. The gradient boosting approach can be used to forecast not just continuous but also categorical target variables (as a Regressor) (as a Classifier). The gradient boosting approach can be used to forecast not just continuous but also categorical target variables (as a Regressor) (as a Classifier). Mean Square Error (MSE) is the cost function when used as a regressor, and Log loss is the cost function when used as a classifier.

- **XGB** XGBoost was created to boost speed and performance while also offering regularisation options to avoid overfitting. Gradient boosted trees use regression trees as weak learners in a sequential learning process. These regression trees are similar to decision trees, but instead of using a single score for each leaf, they use a continuous score for each leaf, which is totalled to offer the final prediction. Scores w that predicts a specific outcome y are calculated for each iteration I that builds a tree t. The goal of the learning process is to reduce the total score, which is made up of the loss function at i-1 and the new t tree structure. This allows the algorithm to develop the trees in a sequential manner and learn from prior iterations. The key advantages of XGBoost over other algorithms are its lightning speed and its regularisation parameter, which successfully minimises variance. However, in addition to the regularisation parameter, this algorithm takes advantage of a learning rate from the features, which improves its capacity to generalise even more. XGBoost, on the other hand, is more difficult to comprehend, visualise, and tune than AdaBoost and random forests.

- **Gaussian Naive Bayes** Gaussian Naive Bayes is a Naive Bayes variation that allows continuous data and follows the Gaussian normal distribution. When working with continuous

data, one common assumption is that the continuous values associated with each class follow a normal (or Gaussian) distribution. Gaussian Naive Bayes accepts continuous-valued features and models them all as Gaussian (normal) distributions.To build a simple model, assume the data is characterised by a Gaussian distribution with no covariance (independent dimensions) between the parameters. This model can be fitted by simply calculating the mean and standard deviation of the points within each label, which is all that is required to construct a distribution of this type.

Software Requirements:

1) Platform: Windows 10 or above.
2) Language: Python
3) Text Editor Our whole code execution is finished utilizing Jupyter Notebook programming. Which depends on Anaconda Distribution which is utilized for Programs dependent on data science and related Advanced Python projects It is an open-source IDE extraordinarily intended for the python language.

Some of its features used in our project are –

1) Its editor is used for code completion, editing and highlighting syntax Editing of variables and exploring them using GUI.
2) Its file explorer, variable explorer and help features were of great use. Linkage with various libraries is helpful in writing code easily.
3) In some parts during implementation, Codebooks was also used to run the code in C++ to find errors and display the output.
4) Microsoft excel has also been used to maintain the dataset used in the entire project.

Hardware Requirements

1) 16 GB RAM required
2) Processor with base clock speed 2 GHz Intel Core i9 processor(11th Generation)

System Type: 64-bit Operating System

Evaluation Measures

The final product is assessed depending on the disarray grid and accuracy, review and exactness are determined. It contains two classes: real class and anticipated class. The disarray lattice relies upon these elements: True Positive: in which both the qualities positive that is

1. True Positive: in which both the values are positive that is 1.
2. True Negative: it is a case where both values are negative that is 0.
3. False Positive: this is the case where the true class is 0 and the non-true class is 1.
4. False Negative: It is the case when the actual class is 1 and the non-true class is 0.

- **Precision is defined as follows:**

Precision is the ratio between the True Positives and all the Positives. For our problem statement, that would be the measure of patients that we correctly identify as having a heart disease out of all the patients actually having it. Mathematically:

Precision = true positive / Actual result

Precision = true positive/(true positive + false positive)

Precision also gives us a measure of the relevant data points. It is important that we don't start treating a patient who actually doesn't have a heart ailment, but our model predicted having it.

- Having a high precision is crucial because it means that we can be confident that our model is detecting fraudulent activity and that we are not marking genuine transactions as fraudulent.
- **The recall is defined as follows:**

The recall is the measure of our model correctly identifying True Positives. Thus, for all the patients who actually have heart disease, recall tells us how many we correctly identified as having a heart disease. Mathematically:

Recall = true positive / predicted result

Recall = true positive/(true positive + false negative)

- High recall is also crucial because it means that we are catching all of the fraudulent transactions.
- **Accuracy defined as**

Accuracy is the ratio of the total number of correct predictions and the total number of predictions. Mathematically

Accuracy = (true positive + true negative)/ total

- **F-1 Score defined as**

Precision and Recall are the two building blocks of the F1 score. The goal of the F1 score is to combine the precision and recall metrics into a single metric. At the same time, the F1 score has been designed to work well on imbalanced data.F1 score formula The F1 score is defined as the harmonic mean of precision and recall.

Mathematically:

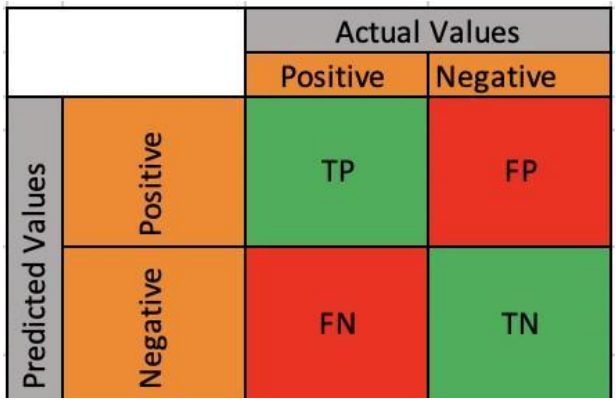
F_1= 2. (Precision . recall) / (precision +recall) = TP / (TP + ½(FP+FN)

We want to achieve high precision and recall, but there is a tradeoff. As precision goes up, recall goes down, since models tuned to prioritize recall will not perform as well with regards to precision and vice versa. As we consider the tradeoff, we can use the F1 score as a metric, which is the harmonic mean of precision and recall. This acts as sort of a proxy for overall accuracy and is sensitive to extreme numbers (i.e. if either recall or precision is poor, F1 will be poor as well).

- **Confusion Metrics are defined as:**

A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making.

For a binary classification problem like our project, we would have a 2 x 2 matrix as shown below with 4 values:



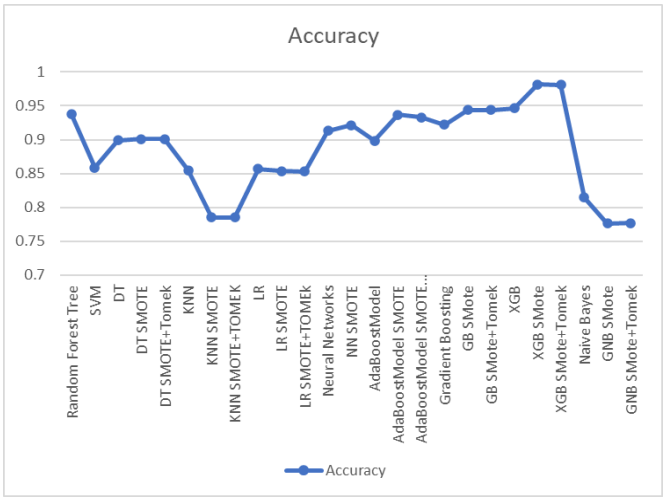
- **ROC Curve:**

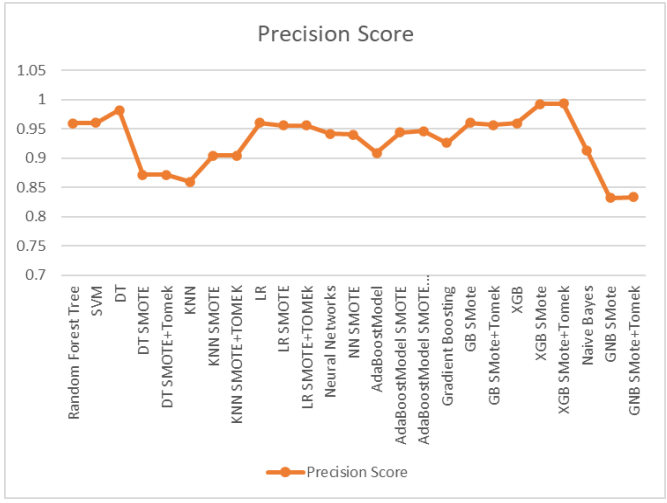
A receiver operating characteristic curve (ROC curve) is a graph that shows how well a classification model performs across all categorization levels. Two parameters are plotted on this curve: True Positive Rate is a measure of how often something is true. The rate of false positives.

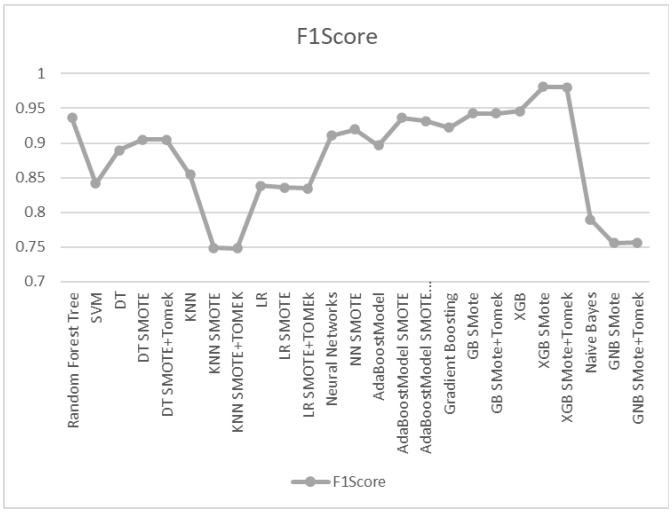
Area Under The ROC Curve: The area under a ROC curve is used to compare the usefulness of tests because it is a measure of the usefulness of a test in general, with a larger area indicating a more valuable test.

Result And Observations

For New DataSet

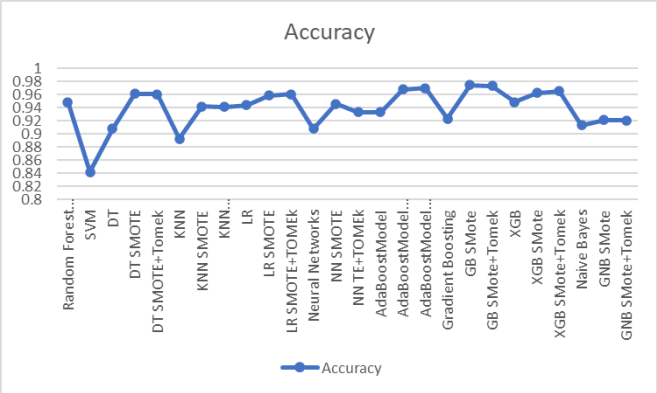


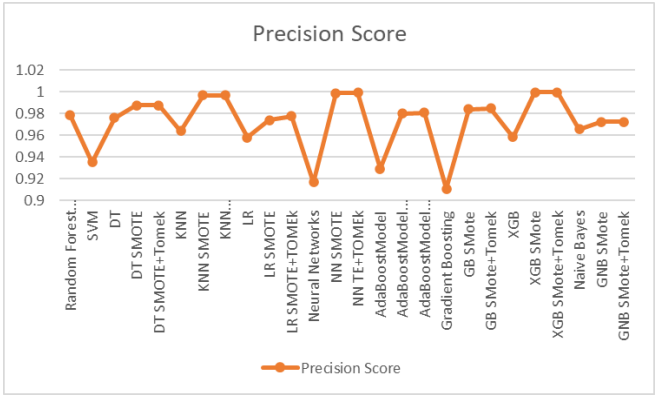


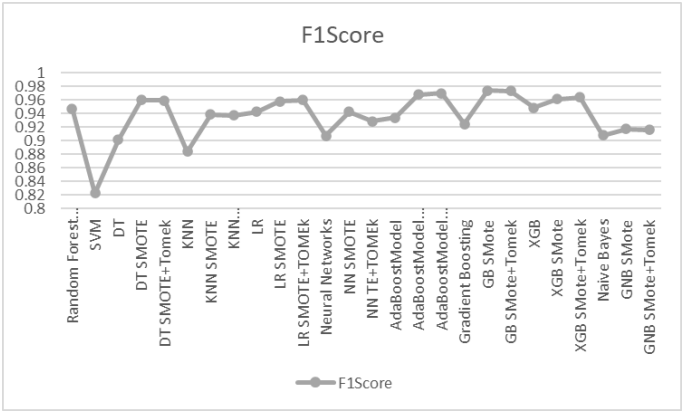


Serial Number	Model Name	Accuracy	Precision Score	F1Score
0	Random Forest Tree	0.9378	0.9598	0.9362
1	SVM	0.8590	0.9605	0.8415
2	DT	0.8988	0.9825	0.8892
3	DT SMOTE	0.9011	0.8718	0.9048
4	DT SMOTE+Tomek	0.9008	0.8715	0.9046
5	KNN	0.8552	0.8598	0.8543
6	KNN SMOTE	0.7855	0.9044	0.7485
7	KNN SMOTE+TOMEK	0.7853	0.9045	0.7482
8	LR	0.8566	0.9603	0.8385
9	LR SMOTE	0.8539	0.9563	0.8354
10	LR SMOTE+TOMEk	0.8533	0.9563	0.8346
11	Neural Networks	0.9133	0.9417	0.9104
12	NN SMOTE	0.9214	0.9405	0.9196
13	AdaBoostModel	0.8981	0.9090	0.8968
14	AdaBoostModel SMOTE	0.9370	0.9444	0.9364
15	AdaBoostModel SMOTE +Tomek	0.9327	0.9462	0.9317
16	Gradient Boosting	0.9224	0.9266	0.9220
17	GB SMote	0.9438	0.9606	0.9427
18	GB SMote+Tomek	0.9436	0.9572	0.9428
19	XGB	0.9466	0.9601	0.9458
20	XGB SMote	0.9814	0.9927	0.9812
21	XGB SMote+Tomek	0.9807	0.9931	0.9805
22	Naive Bayes	0.8149	0.9131	0.7899
23	GNB SMote	0.7766	0.8326	0.7560
24	GNB SMote+Tomek	0.7772	0.8340	0.7565

For Old Dataset







Serial Number	Model Name	Accuracy	Precision Score	F1Score
0	Random Forest Tree	0.9490	0.9783	0.9474
1	SVM	0.8418	0.9351	0.8229
2	DT	0.9082	0.9762	0.9011
3	DT SMOTE	0.9612	0.9878	0.9601
4	DT SMOTE+Tomek	0.9606	0.9878	0.9595
5	KNN	0.8929	0.9639	0.8840
6	KNN SMOTE	0.9420	0.9969	0.9386
7	KNN SMOTE+TOMEK	0.9412	0.9968	0.9377
8	LR	0.9439	0.9579	0.9430
9	LR SMOTE	0.9587	0.9739	0.9581
10	LR SMOTE+TOMEk	0.9610	0.9778	0.9603
11	Neural Networks	0.9082	0.9167	0.9072
12	NN SMOTE	0.9460	0.9990	0.9430
13	NN TE+TOMEk	0.9335	0.9991	0.9288
14	AdaBoostModel	0.9337	0.9293	0.9340
15	AdaBoostModel SMOTE	0.9684	0.9801	0.9680
16	AdaBoostModel SMOTE +Tomek	0.9702	0.9810	0.9698
17	Gradient Boosting	0.9235	0.9109	0.9246
18	GB SMote	0.9746	0.9841	0.9744
19	GB SMote+Tomek	0.9735	0.9849	0.9731
20	XGB	0.9490	0.9583	0.9485
21	XGB SMote	0.9628	0.9997	0.9613
22	XGB SMote+Tomek	0.9655	0.9996	0.9643
23	Naive Bayes	0.9133	0.9655	0.9081
24	GNB SMote	0.9217	0.9725	0.9173
25	GNB SMote+Tomek	0.9208	0.9724	0.9162

Project Novelty

As already mentioned in our study of related work that previously made similar projects and papers had quite a few drawbacks which were:-

1. Using an old dataset from the year 2013 which is not only degraded but was also encoded hence providing no extra information about the users,
2. The second drawback is the dataset used in the training and testing of the model is highly unbalanced and has some missing values and was uncleaned.
3. The major drawback of the previously made project is that they used only one algorithm for predicting the results.

Our Project is unique and better than those which are available on public Cyberspace in that for this specific performance analysis project we used a dataset of the year 2020 having 1296675 rows and 23 columns of around two years spanning from January 2019 to December 2020

- The data being latest along with various details about the users whose data have been collected giving us proper insight and more content for better training and testing of the models,
- This vast information creates a new problem which is that the libraries used for various algorithms take only numeric inputs, not strings which are also solved in our project by using encoding mechanisms
- The dataset had some missing values as well so we cleaned our data our project also solves the problems of the unbalanced dataset. We have balanced our data for training purposes to a level where we get 50% of True (1) cases and 50% of False Cases(0)

Lastly, to get a better result we run our model in 6 different algorithms and then compare their metrics for example precision, accuracy and recall. We represented our results in graphs to have a big picture of our findings.

Conclusion

In this project, we have proposed a method for Performance Analysis Of different Machine Learning algorithms on credit card fraud detection, using various methods of imbalanced data handling. We first compare it with supervised machine learning and deep learning algorithms such as k-Nearest Neighbor, Support vector machine Decision Trees And Random Forest. Finally, we have used neural networks, even though it is tough to train the model which would fit fine to the model for detecting True(1) results in our test dataset. In our model, using a boosting algorithm like gradient boosting and eXtreme Gradient boosting which gives accuracy approximately equal to 97-98% accuracy score on different datasets which was the highest accuracy so far for our test models followed by Neural Networks whose accuracy might increase depending on the number of the hidden layers and number of iterations between them. It gives accuracy more than that of the unsupervised learning algorithms. In this project work, we have done data pre-processing, imbalanced data handling, normalization and under-sampling were carried out to overcome the problems faced by using an imbalanced dataset.

Future Aspects of The Project

The possible future aspects of this project can be that we can use various other types of data sets and predict their desired results, we can try some new machine learning or deep learning algorithms like other types of regressions and we can also use hybrid models for making the predictions more accurate, other than trying out new algorithms and datasets we can try to optimize Neural Network algorithm by changing the values of the number of hidden layers of the neural network along with the maximum number of iterations

References

Research Paper And Sites Used To Understand Various Algorithms

1. M. Zamini and G. Montazer, "Credit Card Fraud Detection using autoencoder based clustering," 2018 9th International Symposium on Telecommunications (IST), 2018, pp. 486-491, doi: 10.1109/ISTEL.2018.8661129.
2. Sadgali, I., Sael, N., & Benabbou, F. (2020). Adaptive Model for Credit Card Fraud Detection. *International Journal of Interactive Mobile Technologies (iJIM)*, *14*(03), pp. 54–65. https://doi.org/10.3991/ijim.v14i03.11763
3. S. Makki, Z. Assaghir, Y. Taher, R. Haque, M. Hacid, H. Zeineddine, A trial review with imbalanced characterization approaches for charge card misrepresentation location, IEEE Access 7 (2019) 93010–93022, doi:10.1109/ACCESS.2019.2927266.
4. Sohony, Ishan & Pratap, Rameshwar & Nambiar, Ullas. (2018). Ensemble learning for credit card fraud detection. 289-294. 10.1145/3152494.3156815.
5. P. Kumar, F. Iqbal, Credit card extortion recognizable proof utilizing AI draws near, in:
6. Proceedings of the first International Conference on Innovations in Information and Communication Technology (ICIICT), CHENNAI, India, 2019, pp. 1–4, doi:10.1109/ICIICT1.2019.8741490.
7. A.A. Taha, S.J. Malebary, An insightful way to deal with charge card misrepresentation identification utilizing an upgraded light slope helping machine, IEEE Access 8 (2020) 25579–25587, doi:10.1109/ACCESS.2020.2971354.
8. Kumar, Abhishek & Prusti, Debachudamani & Ingole, Shubham & Rath, Santanu. (2021). Real time SOA based credit card fraud detection system using machine learning techniques. 1-6. 10.1109/ICCCNT51525.2021.9579598.
9. C. Jiang, J. Melody, G. Liu, L. Zheng, W. Luan, Credit card misrepresentation discovery: a clever methodology utilizing total procedure and criticism component, IEEE Internet Things J. (5) (Oct. 2018) 3637–3647, doi:10.1109/JIOT.2018.2816007.
10. Z. Li, G. Liu, S. Wang, S. Xuan, C. Jiang, Credit card misrepresentation identification through kernelbased directed hashing, in: Proceedings of the IEEE SmartWorld, Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), Guangzhou, 2018, pp. 1249–1254, doi:10.1109/SmartWorld.2018.00217.
11. 10.Tran, Phuong Hanh & TRAN, Kim Phuc & Huong, Truong & Heuchenne, Cédric & Tran, Phuong Hien & Le, Huong. (2018). Real Time Data-Driven Approaches for Credit Card Fraud Detection. 10.1145/3194188.3194196.

12. 11.S. Akila, U.S. Reddy, Credit card misrepresentation discovery utilizing non-covered danger based sacking outfit (NRBE), in: Proceedings of the IEEE International Conference on Computational Intelligence and Computing Research (ICCIC),
13. Coimbatore, 2017, pp. 1–4, doi:10.1109/ICCIC.2017.8524418.
14. 12.M.S. Kumar, V. Soundarya, S. Kavitha, E.S. Keerthika, E. Aswini, Credit card extortion recognition utilizing irregular woods calculation, in: Proceedings of the third International Conference on Computing and Communications Technologies (ICCCT),
15. Chennai, India, 2019, pp. 149–153, doi:10.1109/ICCCT2.2019.8824930. 40 A. RB and
16. S.K. KR Global Transitions Proceedings 2 (2021) 35–41
17. Y. Jain, N. Tiwari, S. Dubey, S. Jain, A similar investigation of different Visa extortion discovery methods, Int. J. Late Technol. Eng. 7 (2019) 402–407.
18. Kumar, Abhishek & Prusti, Debachudamani & Ingole, Shubham & Rath, Santanu. (2021). Real time SOA based credit card fraud detection system using machine learning techniques. 1-6. 10.1109/ICCCNT51525.2021.9579598.
19. E. Esenogho, I. D. Mienye, T. G. Swart, K. Aruleba and G. Obaido, "A Neural Network Ensemble With Feature Engineering for Improved Credit Card Fraud Detection," in IEEE Access, vol. 10, pp. 16400-16407, 2022, doi: 10.1109/ACCESS.2022.3148298.
20. Praveen Mahesh, K., Ashar Afrouz, S., & Shaju Areeckal, A. (2022). Detection of fraudulent credit card transactions: A comparative analysis of data sampling and classification techniques. *Journal of Physics. Conference Series*, *2161*(1), 012072. https://doi.org/10.1088/1742-6596/2161/1/012072
21. Praveen Mahesh, K., Ashar Afrouz, S., & Shaju Areeckal, A. (2022). Detection of fraudulent credit card transactions: A comparative analysis of data sampling and classification techniques. Journal of Physics. Conference Series, 2161(1), 012072. https://doi.org/10.1088/1742-6596/2161/1/012072.
22. Simulated Credit Card Transactions generated using Sparkov,Credit Card Transactions Fraud Detection Datas link https://www.kaggle.com/kartik2112/fraud-detection