Credit Card Fraud Detection using Ensemble Learning Algorithm

Gupta Rishi Chandrashekhar

3008898

Submitted in partial fulfillment for the degree of

Master of Science in Big Data Management & Analytics

Griffith College Dublin

September, 2020

Under the supervision of

Dr. Bilal Yousuf

Disclaimer

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the Degree of Master of Science in Big Data Management & Analytics at Griffith College Dublin, is entirely my own work and has not been submitted for assessment for an academic purpose at this or any other academic institution other than in partial fulfilment of the requirements of that stated above.

Signed: __Rishi Gupta_____ Date: ___07/09/2020____

Table of Contents

Acknowledgements

I wish to express my profound gratitude to our Head of faculty/ Programme Director Dr. Waseem Akhtar and project guide Dr. Bilal Yousuf for allowing me to go ahead with this project and giving us the opportunity to explore this domain and for constant encouragement and support towards achieving this goal. We would also like to thank the Review Committee for their invaluable suggestions and feedback without whom our work would have been very difficult. The blessing, help and guidance given by him time to time shall carry me a long way in the journey of life on which I am about to embark. No project is ever complete without the guidelines of these experts who have already established a mark on this path before and have become masters of it, and as a result, our teachers. So I would like to take this opportunity to thank all those who have helped us in implementing this project.

List of Figures

List of Tables

Abstract

According to ShiftProcessing $24.26 billion was lost all over the world in 2018, the countries where the fraud hit the most are: Mexico (34%), Brazil (25%), United States (22%), Australia (21%), India (20%) and the list continues [1]. Each countries embedded chipset in EVM (Europay, Visa & MasterCard) though the progression of increase in credit card fraudality. The main aim of this project is to efficiently evaluate the detection of fraud while transacting credit card, this is achieved by applying two boosting algorithms i.e. Adaptive Boosting (AdaBoost) and Gradient Boosting technique. Both of these boosting technique algorithm are subset of Ensemble Learning which will be explained in next few section. After intensive modelling and training data we'll determine which algorithm have the best outcome out of both.

Chapter 1. Introduction

Optimal prediction of fraud in credit card transaction is critical task and different organization are working in order to reach maximum accuracy level with high efficiency. Initially such types of fraud detection were implemented using supervised learning algorithm and unsupervised which was then advanced with Artificial intelligence and to neural network. Neural network based algorithm has high variance due to the characteristic of nonlinearity. This increase the error rate and would result in improper classification and output. The improvement to the problem we have ensemble learning mechanism which is uses multiple neural network and different models combine them in order to reduce the variance to minimal that would increase the probability of reaching highest accuracy level which is desired. Ensemble learning is divided into 3 different sections: Boosting, Bagging & Stacking. This paper follows boosting technique. Boosting technique types: Adaptive boosting, Gradient Tree boosting and Xtreme Gradient Boosting techniques. Due to an approach of selecting data points/ instances in a dataset under Gradient Boosting technique it has been found that it is not best choice for prediction under big data scenarios. This is due to it creates high variance, more on this will be discussed in the implementation section. Therefore an extension of Gradient Boosting technique i.e. Stochastic Gradient Boosting technique is applied in comparison with Adaptive Boosting technique for the thesis paper. By the end of this particular paper we would be able to determine which algorithm among the aforementioned are the best and most precise to deploy in any real world application. The key aim targeted is to predict fraudulent transaction hence, as per any modelling approach various training and testing is applied on 500000+ instances hence the structure of the application addressed in this paper follows from a single problem domain based to big data and modelling issue.

1.1 Research Question:

To what extent can Adaptive Boosting and Stochastic Gradient Descent Boosting technique predict fraudulent financial transaction?

Chapter 2. Background

2.1 Literature Review

Up till now N-number of application consisting of various different algorithms are deployed in order to resolve the problem of Credit card fraud issue. This includes Naïve Bayes, K-Nearest Neighbour being the basic modern approach following with Logistic regression, similar coefficient, Distance Sum, Decision Tree, Random Forest etc. Few improved algorithms with better performance are Neural Network: Feed Forward NN, Recurrent NN, Convolutional NN etc. In this Section all such deployed algorithm will be discussed.

2.1.1 Single Model:

A single model in learning refers to any particular learning algorithm that supervised or unsupervised, however a complex and hybrid model cannot be derived and computed. Considering algorithms which are highly efficient in supervised learning and problem solving methods such as Linear Regression, Logistic Regression, Naïve Bayes, K-nearest neighbour algorithm etc. Rahul Goyal, Amit Kumar Manjhvar and Vikas Sejwar [1] (Rahul Goyal, Amit Kumar Manjhvar, Vikas Sejwar, May 2020) differentiated between one such algorithm, furthermore a hypothesis is made where these algorithm aren't suitable to solve non-linear problem (Real-World); therefore a performance differentiation is made between a Logistic Regression and Extreme Gradient Boosting technique (Ensemble learning algorithm). Logistic regression works with sigmoidal function which varies from 0-1, hence using threshold values the in between range i.e. (0.1, 0.2,….. n) is converted to 0's and 1's. A model best result when tested under incremental approach i.e. including all generated classifiers evaluated in the best chain and classifier. Therefore for both Logistic Regression and Extreme Gradient Boosting technique 2 different testing is performed where in 1st a static model performance evaluation is derived followed with incremental. As a result XGBOOST technique performed best with recall value @ 73.83 for static setup and 0.99 for incremental setup.

2.1.2 Hybrid Models:

A hybrid model is combination of two or more supervised and unsupervised learning algorithm. Together they share the input and weights and different mechanisms are chosen in order to retrieve the final output. This includes Multi-layer perceptron, Ensemble learning technique, boosting technique, bagging etc. In research paper [2] (KULDEEP RANDHAWA)] Kuldeep Randawa with his co-authors addressed the issue of credit card fraud based on physical and technical challenges. For generalized comparison they have differentiated 12 different modern algorithm on the basis of Accuracy, Fraud, Non-Fraud and Mathews Correlation Coefficient (MCC). Secondly Adaboost boosting technique was separately calculated with the same aforementioned parameters and finally a hybrid model is interpreted which is used to cast vote and get highest accuracy and correlation coefficient in a hybrid environment. As a result an exact MCC score= 1 was evaluated using Adaboost technique. Our research and thesis also follows similar research background, i.e. in our implementation Stochastic Gradient Boosting technique is used along with Adaboost Boosting technique. In Adaboost technique the outputs are combined by using a weighted sum, which represents the combined output of the boosted classifier where as in stochastic gradient boosting technique where each time a random sets of instance are taken rather than the whole dataset, this ensures to get different output after every iteration that could lead to better prediction and faster in processing comparatively.

Multiple experiments are still in progress to evaluate maximum precision and recall value while detecting fraud in a credit card transaction. The following paper [3] by Rahul Goyal, Amit Kumar and Vikas Sejwar published by International Journal of Recent Technology and Engineering is based on such algorithm techniques and extended version, our thesis project will implement using one such algorithm and techniques of same branch of technology. Here Ensemble Learning technique is introduced, it is a learning technique used to classify and predict the model with high efficiency. As per their literature survey logistic regression is deployed in various credit card fraud detection based technology, the hypothesis they formed was logistic regression is best suited for dependent variables which form a linear correlation with each other; however when the possibility of non-linear

problem solving occurs then its scope is reduced. Therefore a different Machine Learning approach has to be implemented for this instance XGBOOST Ensemble learning algorithm. A particular model is trained with multiple algorithms and together combined an optimum output is generated at the front end. Ensemble learning is divided into: Boosting, Bagging, Stacking, Bucketing technique; the technique introduced here [3] Extreme Gradient Boosting (XGBOOST) technique. Continuing, XGBOOST is compared with relative output of Logistic Regression model technique. Logistic Regression works on sigmoidal function i.e. on discrete values. The two analysis metrics used are Recall Value and Average Precision. Also two different approaches are driven that is 'static', 'incremental' setup. This decision will help determining if the Extreme Gradient Boosting technique works better for small scale and large scale or not. Under the static setup logistic regression scored 58.47% for Recall, 77.0% for average precision following XGboost scored 73.83% and 83.0% respectively. On the other hand with Incremental setup XGBoost resulted into 99.00% comparatively 0.3% higher in accuracy.

Considering any analysis on credit card fraud detection 4 parameters are always set as constant, moreover the modelling and workflow of overall project is dependent on this 4 basic pillar which are called as:

True Positive (TP): Classifying a fraudulent transaction as fraudulent.
False Positive (FP): Classifying a normal transaction as fraudulent.
True Negative (TN): Classifying a normal transaction as normal.
False Negative (FN): Classifying a fraudulent transaction as normal.

Fraud detection is operated on TP, FP and FN and a cost is estimated w.r.t. to these parameter. In paper [4] (Fahimeh Ghobadi, Mohsen Rohani, 2016) Fahimeh Ghobadi alog with Mohsen Rohani from Islamic Azad University South Tehran Branch did a detailed work on detecting fraud in real time simultaneously making Hit Rate maximum considering with minimal cost. Overall for implementation Neural Network technique is acquired with the back propagation property. The network begins with 17 input neurons following with 2 hidden layers consisting 60 neurons at 1st hidden layer and 60 neurons for 2nd hidden layer. This makes is highly robust and efficient yet complex algorithm, to train

the prototype with real time data and obtain result back propagation technique is deployed which will train model and reduce the error (Hit Rate) after every iteration. The output of this network is distributed into to section 1.Anaomly detection 2. Fraudulent transaction, anomaly detection will handle the outliers using distance matrix and fraudulent transaction will deal with clustered item set of both fraudulent and non-fraudulent transaction.

The initial approach for credit card fraud detection was based on supervised learning methods which includes complete data associated with the transaction, however in paper [5] (Zhang yongbin, You fucheng, Liu huaqun, 2009)Zhang Yongbin, You Fucheng and Liu Huaqun from Beijing Institute of Graphic Communication designed a model which detect anomaly from behavior of credit card user and past transaction in the database. Critical user data such as Age, Name, and Gender etc. isn't considered while training or testing the model. Hence the work is completely dependent on numeric and transactional information. Following is the workflow of designed algorithm:
Two data store in input, 1. Information from new customer 2. Historical data → Pre-Processing (Data ingestion, data cleaning) → Fraud detection algorithm (Supervised: Fuzzy Logic technique) ➔ Output (1.Legal behavior, 2. Suspected Behavior). Depending upon behavior such as transacting time interval, transaction location fraudulent transaction are classified. However since the model is supervised and doesn't handle large data set (instances) feasibility of deploying is unrealistic.

Large dataset and records are operated with under-sampling in order to deal with high imbalance dataset, however this discards some highly relevant training instances which can me crucial for a classifier to analysis. The following paper [6] (Hongyu Wang, 2018) proposed by Hongyu Wang, Ping Zhu and associates from Beijing University of Posts and Telecommunications deals with such problem where they work on ensemble learning model with the help of training set partitioning and clustering. In order to balance the imbalanced dataset is divided into two parts $S_{minority}$ and $S_{majority}$, as per their EDA number of samples in $S_{majority}$ is evidently high in every block compared to $S_{minority\ hence}$ a machine learning algorithm is acquired to select nearest neighbor from every cluster. At the end of each $S_{minority}$ block of dataset after applying nearest neighbor method we get highly integrated and balanced dataset which is ready to be classified using different classifier. Due to the hypothesis of Random Forest (RF) being the highly rated in terms of

performance the authors decided to C4.5 as base evaluator as an improvement in sampling process of Random Forest. Overall 5 different experiments were conducted with two parameters i.e. Area Under Curve (AUC) and Savings. Savings will help to determine if the proposed model is feasible in terms of time and cost. Random forest based on partitioning and hierarchical clustering (RFPH) and random forest based on under sampling (RFRU) is compared in accordance with ensemble learning framework. As a result after 5 experiments averages of RFPH and RFRU it is evaluated that RFPH has 96.52% and RFRU has 94.69% of accuracy following with 67.97% under savings of RFPH and 62.02% for RFRU, indicating RFPH has better accuracy and is cost efficient.

2.2 Related Work

In this section both single model and hybrid model implemented for fraud detection is discussed in relation to the state of the art along with performance evaluation and future improvement with respect to the field of credit card fraud detection.

In [7] (Xi, April 2008)Wen-Fang Yu and Na Wang have formed a hypothesis based on outliers under cluster or sampling of records/item set. A distance matrix is calculated for each instance which is associated with the neighbour instance. Initially the dataset is clustered into fraud and non-fraud and forecast is computed using distance matrix and coefficient correlation where in a threshold value is passed up to which TRUE POSITIVE(TP), FALSE NEGATIVE(FN), FALSE POSITIVE(FP), TRUE NEGATIVE(TN) are compared. As a result it is predicted that this model has better result than anomaly detection. Data standardization is a complex and crucial pre pre-processing task to address it standard deviation and mean absolute deviation are implemented. This helps to ensure that the value of multiple instances in different attribute ranges in to similar set. Although outlier mining is and has been always a major area to expertise, here the authors have proposed the solution with minimum cost and better performance. Where in a particular model is evaluated 5 times using different threshold levels can be extended further for better sampling

Similarly in research paper [8] (Kosemani Temitayo Hafiz, Dr. Shaun Aghili, Dr. Pavol Zavarsky, 2016) Kosemani Temitayo Hafiz, Dr. Shaun Aghili, Dr. Pavol Zavarsky performed a predication analysis to detect credit card fraud in Canada (specific demographic). This paper particularly focuses on comparing Predictive Analytics Vendors (PAT) in Canada who provide security mechanism to different card holder companies such as Falcon Fraud Manager, IBM SPSS Modeller, SAS Fraud Manager, Cyber Source decision Manager, ACI Proactive Risk Manager etc. Rather applying any deep learning or machine learning algorithm they have compared number of algorithm being accepted into different banks and how efficiently they work. Furthermore, what challenges are faced during or after predictive analysis are listed such as Model & Algorithm issues, Cost misclassification matrices, Human Failure Risk, Compliance with requirements of law and Regulations etc. One major parameter that have major impact in the initial stage of credit card fraud detection is privacy rule and compliance; an analyst or an organisation has to work under

the guidelines amended by regulatory authority, due to this many crucial information related to customer/client are discarded before handling the raw data.
Similar to logistic regression, the following paper explores the features of distance sum matric algorithm. The main objective for this research paper was to maximize the core points in order to randomize into appropriate clusters while performing data mining techniques and multiple models. The algorithm used here is outlier mining, overall recent credit card transaction of past 10 years of 16500 records wherein 1500 were of fraud is used. Secondly the basic data cleansing was performed in order to remove less or no dependent variables/attributes. This increase the precision value and chances of getting accuracy with higher probability. However a simple basic distance matrix was computed using Euclidean distance technique which is widely accepted. Two threshold value of 0.5, 0.6 were considered. After detailed analysis it was found the model could classify the object points better in threshold value =0.6.

Contradict to [9] (Chun-Hua JU, 2009) where in Chun-Hua JU along with Na Wang from Zhejiang Gongshang University worked on Similar Coefficient Sum matrix approach here in [4] Wen-Fang YU and Na Wang did a collaboration and introduced Fraud Detection using Distance Sum. Under Distance Sum Euclidean algorithm was determined. The main approach here is to perform data mining into the outlier instances. The dataset has total 16584 instances wherein 15135 were non-fraudulent and 1449 were categorized as fraudulent. For experiment the threshold value was set to γ=10, 12. Experimenting with threshold help us to determine behavior of instances around the cluster. As per the results the experiment with γ=12 resulted into higher accuracy into the prediction of credit card fraud for forecasted months.

In paper [10] (Anuruddha Thennakoon1, Chee Bhagyani, Sasitha Premadasa, Shalitha Mihiranga, Nuwan Kuruwitaarachchi, 2018) a complete top to bottom solution with credit card fraud detection is illustrated, i.e. storing information, data pre-processing, applying Machine learning strategies, distributing workflow and visualizing the output. This whole architectural approach is gained by accessing Microsoft Azure Services which includes Apache Kafka, Data lake store, Azure DataBricks, Azure ML and Power BI. Lakshya Sahahi and Kemal Gursoy have designed a model that takes real time data (big data) ingest it to the Azure DataBricks via Apache Kafka and apply different Machine Learning and

Ensemble learning algorithm in order to process and predict/detect credit card fraud. A comparative analysis is performed between Stochastic Gradient Descent classifier and Extreme Random Forest algorithm. The parameters obtained are Precision, Recall value, Accuracy, AUC, F1 score for both aforementioned algorithms.
As a result it was determined that SGD algorithm had higher accuracy level and performed better on all parameters compared to Extreme Random Tree.

Sampling a particular item set in order to retrieve highly accurate prediction is benchmark approach in all technical solution. Thus Anuradha Thennakkoon with her partners sampled the dataset into 4 different types and processed further to derive optimum result. In their paper they have classified fraud into 4 different types 1. Transaction with Risky Merchant Category Code (MCC), 2. Transaction larger than $100, 3.Transactions with risky ISO Response code, 4.Transaction with unknown web address. All these 4 types were processed in order to cluster under and over sampling models, this would improve the probability in choosing correct algorithm for training. Hence the end node for these are 4 different algorithms handling 8(over sampled, under sampled data). This is overall a hybrid model which is sophisticated and highly efficient. Finally few performance metrics are evaluated which take all 4 algorithms into consideration and a final result of parameters is shown.

Until now all the research paper discussed had comparative analysis between 2 or 3 algorithm and techniques along with a top to bottom solution pipeline for real time credit card fraud detection using Azure services. Furthermore,[12] (KULDEEP RANDHAWA, CHU KIONG LOO, CHEE PENG LIM, ASOKE K. NANDI , 2018) Kuldeep Randhawa (Senior IEEE Member) with his colleagues proposed a diversified solution which consists over total 12 different algorithm that includes Machine learning, Deep learning, neural network etc. To name these are the algorithms: Naïve Bayes, Decision Tree, Gradient Boosting Technique, Random Tree, Random Forest, Decision Stump, Multi-Layer Perceptron, feed Forward Neural Network, Deep Learning, Linear Regression, Logistic Regression and Support Vector Machine. These are considered as highest rank algorithms and are well accepted by international organization under the deployment of their module. For initial test each individual algorithm resulted their each performance, followed with AdaBoost Boosting Technique accuracy and sensitivity evaluation and finally combination of different algorithm were performed (Hybrid Model). These three analysis were

compared and for each majority voting was taken using predefined techniques. A standardized descriptive analysis resulted into substantial result wherein noise were added to each hybrid algorithm with set of [0%, 10%, 20% and 30%]. Ultimately it was derived that Adaptive Boosting technique performed very well in majority voting than other models with accuracy level up to 100%.

2.3 Analysis of the related work/summary: Critical analysis

In paper[12] (Lakshya Sahai, Kemal Gursoy, 2019), Lakshya Sahai and Kemal Gursoy from Rutgers University performed complete data analysis and accuracy prediction using Azure services which included Kafka, Data lake store, Azure Databricks, Azure ML etc. The approach followed is comparative analysis between Stochastic Gradient Boosting technique and extreme tree boosting technique. As result SGD classifier had higher accuracy level. However using powerful tools such as azure services multiple and hybrid algorithm can be deployed in order to retrieve better and optimum results along with the increasing the possibility of generating random and real time hypothesis that would result into better output.

On contrary considering paper [10], by Kuldeep Randhawa and his associates a total of 12 different models were undertaken for testing and analysis of prediction. This includes both hybrid and singular models. A real world approach is made by combining different algorithms and using voting techniques to get the results. Also as a result the adaboost boosting technique evaluated exact 1.0 score. To reduce the effect of bias on evaluation stage with random sampling 10 cross fold validation is used. Furthermore each model is been trained under different of noise ranging from 0% - 10%. This phenomena will determine the operational and accuracy behavioural change through different phase and how it can be tuned to result in to better prediction.

Dealing with under-sampling an over-sampling is crucial task, in paper [9], Anuruddha Thennakoon and associates have addressed this issue with different category. These categories are based on 4 different hypothesis which were also the null hypothesis. For each iteration/null-hypothesis training is done with different algorithms which are as follows: Support Vector Machine, K-Nearest neighbour and logistic regression. 8 different measure parameters is calculated. A real time model based on banking record and API is deployed which delivers end to end solution i.e. from bank server-> data warehouse -> client; and these all is made feasible and efficient with the on-going algorithm running on the fraud detection model.

2.4 Comparison of top model and application deployed

Sr No.	Title	Approach	Advantages	Disadvantages	Technologies
1	Behavior-Based Credit Card Fraud Detecting Model	Behavior-based fraud detection model based on transactional information	High speed processing due to only transactional pattern data	Due to less feature accessed it is difficult to implement in real world application	Fuzzy logic – unsupervised learning algorithm
2	An Ensemble Learning Framework for Credit Card Fraud Detection based on Training Set Partitioning and Clustering	Ensemble learning framework on training set partitioning and clustering	Produces synthesize and balanced dataset	Time consumption and cost expense is high	Hierarchical clustering and RF based on Random Under sampling
3	Credit Card Fraud Detection Using AdaBoost and Majority Voting	Evaluating multiple models using majority voting technique	Majority voting including hybrid models	Average number of sampling and test	Adaboost, Multilayer perceptron, Neural network etc
4	Credit Card Fraud Detection in Data Mining using XGBoost Classifier	Performance based metrics evaluation	Implementation of both static and incremental setup for all deployed algorithms	Comparing two different learning algorithm techniques, supervised vs unsupervised	Logistic Regression and Extreme Gradient Boosting technique
5	Real-time Credit Card Fraud Detection Using Machine Learning	Focuses on 4 different approach of fraud. Further dividing data into similar types and evaluating metrics	Categorical data sampling which helps to classify evaluation metrics highly accurate	Cost consumption and execution time is high	Support vector machine, K-nearest neighbor, Naïve Bayes

Table 1 Comparative analysis of related work

Chapter 3. Methodology

As per Nilson Report around $30 billion will be counted under loss as to credit card fraud, the total sum of the loss has to be paid by top-level of stakeholder i.e. the owner of the organization. Furthermore the up rise in projection is noted to be consistent in upper layer of bell curve [13, (Navlani, 2020). However dozens of different applications and modern solutions are proposed in order to predict and prevent the ongoing online credit card fraud. Few of these solutions are discussed in the literature review section. Hence to boost the step for improving the detection of fraudulent transaction with less cost and high accuracy was the prior objective which gradually modified as per development cycle. In this era of open source technology data is as equivalent as oil, finding appropriate information and gathering insights has always been challenging.

The dataset selected this thesis consists of multiple crucial information with respect to processing data analysis task. Such as Transaction type, Origin of transaction, destination of transaction, is the transaction fraud or if the transaction has been flagged fraud etc. The initial task of defining the objective and finding an appropriate dataset to work on is done. As discussed earlier different applications are already proposed and being accessed in current market therefore it is easy to comply and distinguish the performance of a particular model in an application with another. This would direct to select and compare an appropriate model/algorithm for the project [14, (Ramzai, 2019)]. The dataset chosen consist of more than half a million records which turns it into big data problem. Hence a normal DBMS and RDBMS data manipulation application would not be the best fit, thus Python language is selected which is capable of handling high volume data manipulation and visualization. By this our next level of objective was completed of choosing best language platform to work on since python also has various packages available for developers and data analyst to work on[15, (Smolyakov, 2019)]. Following to the flow now arrives the pipeline of the model. Creating a skeleton of workflow from start to end is highly essential and required in order to mark any future changes during the development cycle. Initially it was dived into two sections 1. Back End 2. Front End. Backend deals with the data processing and integration whereas front end deals with User Interface and

visualization. Python pyspark is an API that is mainly used to handle large volume of data for data analysis in a distributed environment of both parallel and batch processing.
Bottom-up approach is selected where all small tasks and function is integrated together to form a final working application. Continuing different tests has to be applied at small scale level to ensure high accuracy and prediction of fraud in a transaction hence Top-down level wasn't feasible and could make deployment much complex. Different tests refers to cross validation on both algorithms chosen in this paper which are Adaboost and Stochastic Gradient Descent boosting technique.

Choosing any suitable algorithm for analysis is a big task since it act as backbone to the application. In the process of literature review it was found that adaboost technique showed best results compared to any other hybrid/ supervised algorithm/ unsupervised learning algorithm such as Random forest, Decision Tree, Naïve Bayes etc. Hence this paper is based on following hypothesis testing of two different boosting technique [16, (Yang, n.d.)]. The evaluation parameters chosen in this thesis are: ROC Curve, AUC value, Precision, Cost (execution time). A comparative analysis is conducted to evaluate the best algorithm suited for predicting the credit card fraud detection. Continuing towards designing front end module, Python flask is adopted and implemented. This is due to the scope of project which is predefined and static towards run time execution. The initial model is trained with PS_20174392719_1491204439457_log.csv file and tested under a random sample data i.e. 'validation_data.csv' which runs in to flask application in local host.

The decision of displaying the resultant output is deterministic for users that illustrate real life execution of algorithm with required statistics which are the evaluation parameters as discussed before. Under programmer level perspective major task up to 80% is implemented in the backed under pyspark on jupyter notebook. This helps to encapsulate the detailed implementation from the front end user. Python PyFlask is a robust API by python which helps to develop small scale web application where to scope is predefined and mostly web pages are static in nature [17, (Zhou, 2020)]. Hence it helps our project to work smoothly with less processing usage that reduced the overall cost of time consumption and processing usage. Considering the cost parameter a unique evaluation measure is deployed that is 'cost' it will help us to determine is that particular model/ algorithm is feasible in real world application since the data being generated and accessed

in day to day life is in terabytes or more [18, (Srivastava, 2019)]. Hence to retrieve an optimum result of any algorithm both accuracy level and overall cost should be best. Lower the cost of execution higher the chance of selection of that particular model in deployment phase.

Chapter 4. System Design and Specifications

4.1 Introduction

The application is divided into 3 phases that is training, testing and validating. In testing we are applying 2 respective algorithms and finding are illustrated such as ROC Curve, AUC and accuracy score with a small scale of overall dataset [19, (stochastic gradient boosting, 2018)]. In training phase 70% of the dataset is used to process and evaluate on the model output by testing phase. Finally a front end is designed to validate real world detection of credit card case by end user. More will be discussed on this topic further in this section.

The designing part is divided into 2 phase: Back-End and Front-End.

Technologies used for back end: Jupyter Notebook, PySpark

Technologies used for front end: PyFlask, HTML

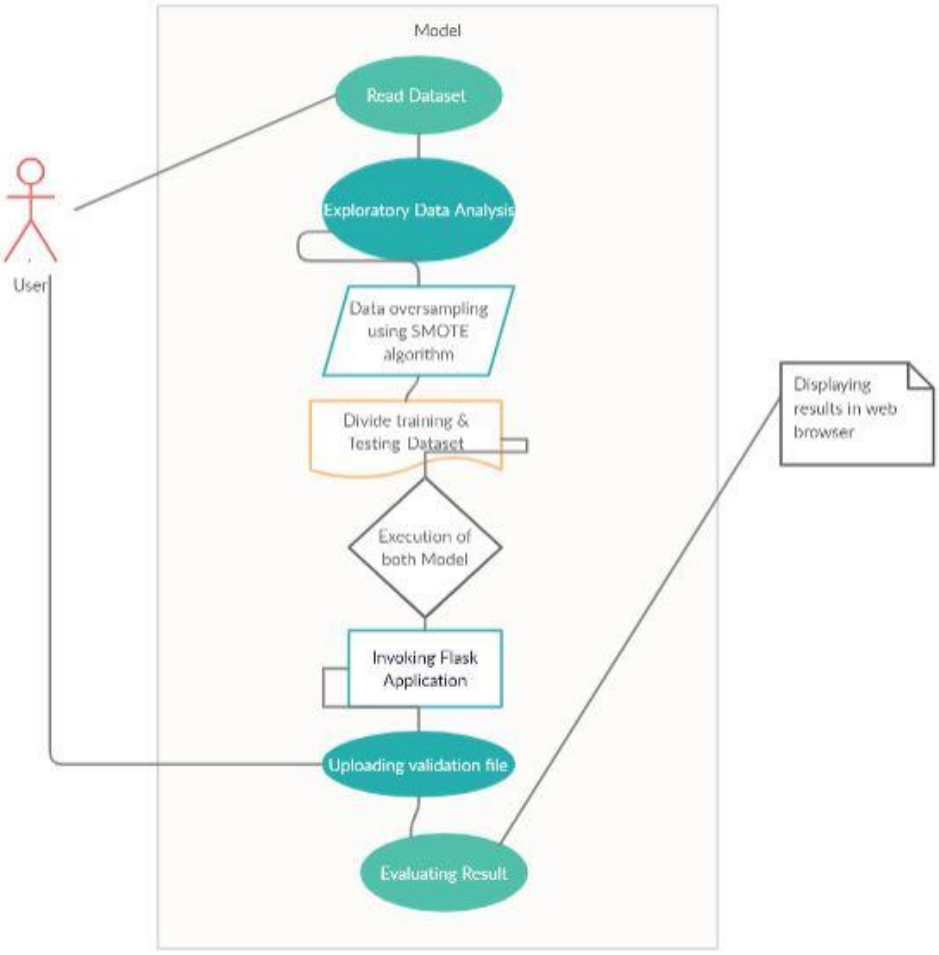


Figure 1.1 UML diagram for Credit card fraud detection using Ensemble learning.

Pyspark is executed under the anaconda 3.64, different packages has to be installed before proceeding the execution of the application such as pyspark, pyFlask, jupyter notebook etc.

Python pyspark is selected for this project, this is due to its compatibility and robustness which helps to integrate various data analysis tasks [20, (Mohamed M. Ahmed, 2019)]. This includes pandas for reading and different file manipulation task, numpy for the arithmetic operation, sklearn which holds multiple learning and training algorithm, seaborn to design and visualize low to high end graphs and matplotlib to plot different graphs. Moreover the huge impact on taking decision on selection of pypark is helps to integrate the back end and front end in a dynamic approach with good processing speed [21, (Business, 2019)].

The whole application code structure begins with:

```
@app.route('/')
def upload_file():
return render_template('index.html')
@app.route('/result', methods = ['GET', 'POST'])
```

This redirects the application to index.html page which holds the request of selecting the validation file.
NOTE: Validation file is set of unique record that is tested in front end. These records are differentiated from the rest of the training, testing dataset combined. This is to predict the optimum result for the real world problem.

app.route(‘/’) will redirect the path to the default location where all the required files are stored. From the result.html is true the ‘GET’ method gets activate and will request for the input file. This particular validation file is uploaded to server and the data processing and modelling begins in the next step of the code which begins with data pre-processing. The task included in pre-processing is mentioned in the implementation section.
To outline here are few task listed:

Checking NULL values

Removing Duplicates

Once successful completion of data cleansing we move forward to exploratory data analysis phase. Here the crucial attributes are considered and the rest are ignored. Since this is a quantitative based thesis it is highly essential to choose attribute with numeric and integers value. This is due to the restriction of data modelling and processing done by different learning algorithm. Also where required few values in a specific attribute is parsed into the numeric value. The attribute transaction type is distinct hence the two types Transfer and Cash_Out are parsed into Boolean values i.e. 0 & 1 respectively. Furthermore the rest of the hypothesis for developing the model is based on this two transaction types.

Once the data is cleaned and segregated next phase is to divide it into test and train dataset. Class balancing is an approach to dynamically fill appropriate and validated data in to the model depending upon the status of low or high dependent values. In this case we have imposed over sampling approach to imply class balancing. This helps to generate the best fit line during the training. SMOTE algorithm is used for class balancing.

```
from imblearn.over_sampling import SMOTE
sm = SMOTE(random_state = 2)
X_train_res, y_train_res = sm.fit_sample(X_train, y_train.ravel())
```

The decision of adding a class balancer was made after several run of training and retrieving the accuracy score of algorithms. The ROC curve was illustrating unbalanced and depreciated type curve. Also the major reason of applying the over sampling is our target attribute has less number of fraud instances i.e ‘1’ compared to ‘0’. Hence over sampling will help to normalize the overall value which in turn result to better line in graph.

4.2 Architectural Diagram of model

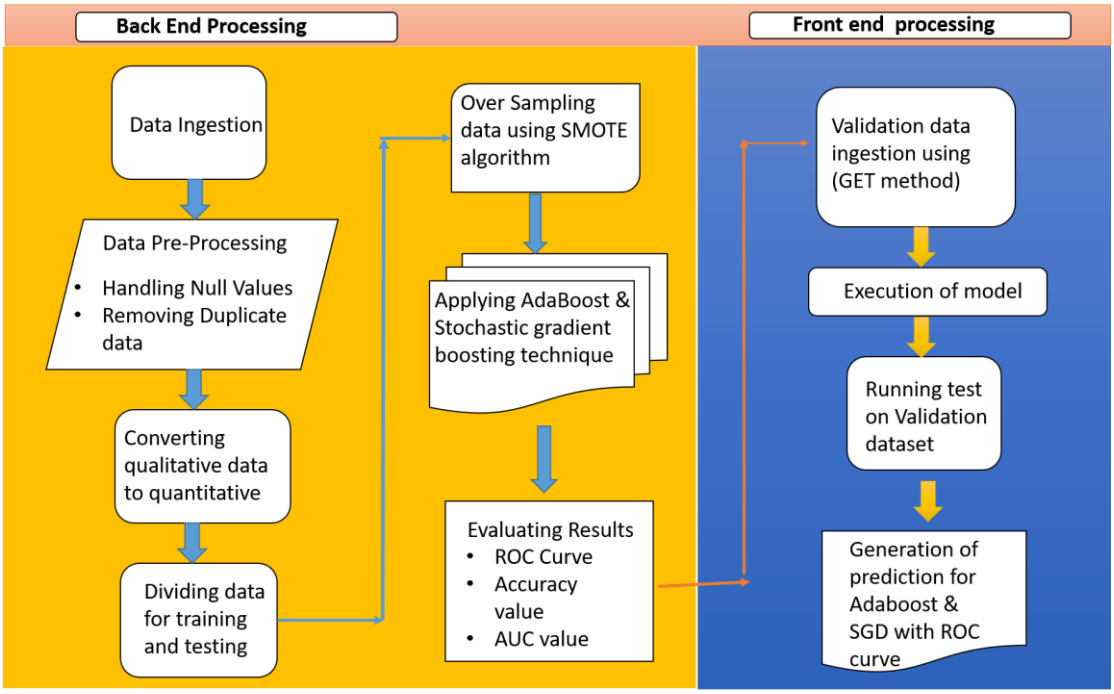


Figure 1.2 Architectural Diagram

Back End Processing

- Data Ingestion : Using pandas package we are reading the actual dataset csv file into the pyspark model. The method used here is 'read_csv()' and printing the initial 10 rows using 'data.head(10)'

- Data Pre-processing : This includes verifying and handling null and duplicate values that would lead to high variance and redundancy problem.
- Qualitative to quantitative: Appropriate attribute selection is highly essential, also in different scenarios a particular type of attribute has to be parsed into Integer or numeric type. This is to conduct data modelling and processing for training and testing. Task defined in this phase are also called as data translation or data transformation.

- Dividing dataset into training and testing: Following a good rule of thumb, the original dataset is divided into 7:3 ratio, where 70% is for training and 30% for testing the model. However the target attribute has fewer fraudulent instances compared to non-fraudulent hence over sampling is applied on it using SMOTE algorithm.

- Evaluating Results: The sklearn package in python avails different machine learning, Neural network, Ensemble learning etc algorithm to be deployed in your application. Furthermore to evaluate this model we have used package sklearn.metric which has different measure related files associated with it, such as accuracy_score, roc_auc_curve, roc_curve, precision recall curve. All the metrics will help us to determine the characteristic of the Adaptive boost algorithm and stochastic gradient boosting algorithm.

Front End Processing

- Request for validation dataset input: PyFlask is accessed and implemented for running tested against validation dataset in front end.
  Using GET/POST method we can retrieve and display the desired output which in this case is predicting the outcome of aforementioned algorithm with respect to validation dataset.

- Running the trained algorithm on validation dataset: In the process of predicting fraudulent transaction validation 'isFraud' attribute is undertaken. The complete training is done under the values of 0 & 1 of 'isFraud' attribute along with transaction type which is also parsed into Boolean value for quantitative analysis.

- Retrieving the output: For the output in the html page, a single dataframe is created which holds 3 different parameters which are 'y_pred', 'adapred', 'sgdpred'.

    - Y_pred: holds the random instances of 'isFraud' with values 0,1 from the validation dataset.
    -
    - Adapred: variable defined to store the trained Adaptive Boosting model.

    - Sgdpred: variable to hold stochastic gradient boosting algorithm.

For every instance present in the validation set both algorithm predicts the output based on the training dataset. Another graphical representation is displayed which illustrates ROC curve and

4.3 Code Snippet:

The following code section initialize both classifiers under training phase, further accuracy is generated using the prediction value for both algorithms. This includes the value derived from training and testing/predicted value. ROC curve deals with 2 values i.e. False Positive ratio and True positive ratio, which determines who a classifier model is actually classifying the targeted value. For anomaly if the target is to classify 0,1 it determines how well does the model classify 0 for 0 and 1 for 1.

PySpark Code snippet:

```
ada= AdaBoostClassifier(n_estimators=10, random_state=42)
# Building a boosted classifier from the training set
ada.fit(X_train,y_train)

# Predicting classes
y_pred= ada.predict(X_test)

adaacc=accuracy_score(y_test, y_pred)*100

# Checking accuracy score of the model
print("Accuracy:",adaacc)

# Predicting class probabilities
probs = ada.predict_proba(X_test)
probs = probs[:, 1]

# Computing Area Under the Curve
adaauc = roc_auc_score(y_test, probs)*100
print('AUC: %.2f' % adaauc)

# Computing the area under the ROC curve
fpr, tpr, thresholds = roc_curve(y_test, probs)
plt.plot(fpr, tpr, color='orange', label='ROC')
plt.plot([0, 1], [0, 1], color='darkblue', linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend()
plt.savefig("static/adaroc.png")
```

```
# ## Stochastic Gradient Descend
# initializing Stochastic Gradient Descend classifier
sgd = SGDClassifier(loss="hinge", penalty="l2", max_iter=10)

# Building a boosted classifier from the training set
sgd.fit(X_train,y_train)

# Predicting classes
y_pred= sgd.predict(X_test)
accsgd= accuracy_score(y_test, y_pred)*100

# Checking accuracy score of the model
print("Accuracy:",accsgd)

# Computing Area Under the Curve
sgdauc = roc_auc_score(y_test, y_pred)*100
print('AUC: %.2f' % sgdauc)
```

Flask application code:

```
data1=pd.DataFrame({'y_pred':Y_val,
'adapred':adapred,'sgdpred':sgdpred})
fig = plt.figure()
    ax = fig.add_axes([0,0,1,1])
    ax.bar(langs,[adaacc,accsgd], color = 'b', width = 0.25,
label="Accuracy Score")
    plt.savefig("static/comp.png")

    fig = plt.figure()
    ax = fig.add_axes([0,0,1,1])
    ax.bar(langs,[adaauc,sgdauc], color = 'g', width = 0.25,
label="Area Under Curve")
    plt.legend()
    plt.savefig("static/comp1.png")

return render_template("result.html",result = data1.to_html(header =
'true'))
```

Html code snippet to parse the dataframe and graph:

{{result|safe}} //displays the dataframe containing y_pred, adapred and sgdped values.
{{url_for('static', filename='adaroc.png')}}
{{url_for('static', filename='sgdroc.png')}}
{{url_for('static', filename='comp.png')}}
{{url_for('static', filename='comp1.png')}}

4.4 Visualisation Diagram

```
Before OverSampling, counts of label '1': 801
Before OverSampling, counts of label '0': 321473

After OverSampling, the shape of train_X: (642946, 8)
After OverSampling, the shape of train_y: (642946,)

After OverSampling, counts of label '1': 321473
After OverSampling, counts of label '0': 321473
Accuracy: 100.0
AUC: 100.00

C:\Users\HP\anaconda3\lib\site-packages\sklearn\linear_model\_stochastic_gradient.py:570: ConvergenceWarning: Maximum number of
iteration reached before convergence. Consider increasing max_iter to improve the fit.
  warnings.warn("Maximum number of iteration reached before "

Accuracy: 94.13617341693335
AUC: 93.82
```

Figure 2 Accuracy score for Adaboost and Stochastic gradient boosting algorithm

Chapter 5. Implementation

This section consists of overall implementation of the application, wherein section 7.1 the working and for Adaboost and Stochastic Gradient Descent Boosting algorithm is defined along with equations. Section 7.2 outlines the overall structure of the application and system design that includes the evaluation measure acquired in order to compare both of the aforementioned algorithms. Furthermore in section 7.3 the output result and analysis is discussed which is based on actual training and testing of the application.

5.1

A] Adaptive Boosting technique:

Consider a bag with full of data points, some are organic data points without any error and few with significance error. Usually in the phase of training all data points or random data points are chosen and trained accordingly. The output of these trained data points is forwarded to next level of training or testing phase. If there is presence of data points with significance of error that will pass on to the next phase and so on; this will lead to poor prediction since the model trained under consist of corrupt data points. Here come the picture of Adaboost boosting technique algorithm. The basic phenomenon is to train the weak learners from the data points until they become strong learners, thus during training phase the priority of getting selected in the pool of dataset for weak learners is high. The algorithm continues to train the weak learner until a desired i.e. no or minimal error rate is recorded. The working depends on the number of different function used in the algorithms, each function will be assigned a stomp (A decision tree with a single depth and two or more leaf node). An entropy value is decided which act as threshold value in order to select the initial stomp. An total error rate is calculated by dividing wrongly classified instances with total number of instances present in the stomp.

Total Error: [1/2(log2(1-1/n)/1/n)] .. n= number of instances in the stomp.

After each iteration error rate is calculated and the actual weight is updated with the new weights. Finally weights for all data points are updated accordingly and then normalized. This happens sequentially until set of data points with less error rate is derived.

1. Initialize the weights $w_i = 1/N$, $i \in \{1, \ldots, N\}$
2. For m=1 to M
   a) Fit the class probability estimate $p_m(x) = \hat{P}_w(y = 1|x) \in [0, 1]$, using weights w_i on the training data
   b) Set $H_m = \frac{1}{2} \log\left(\frac{1-p_m(x)}{p_m(x)}\right) \in \mathcal{R}$
   c) Set $w_i \leftarrow w_i \exp(-y_i H_m(x_i))$ and renormalize to $\sum_i w_i = 1$
3. Output $H(x) = \text{sign}\left(\sum_{m=1}^{M} H_m(x)\right)$

Figure 3 Adaptive Boosting Algorithm

B] Stochastic Gradient Descent boosting technique:

Initially Gradient Descent boosting techniques effectively in a data analysis operation, however while considering big data problem Gradient boosting algorithm consumes a lot of memory and computational power. This is due to selection process of data points in steps while training data. In Gradient boosting all data points are selected and calculated for each step process of function in order to get best fit lie, thus if a dataset having 10000 instances it will calculate the step function of each unique and individual points, this problem is addressed and solved by stochastic gradient boosting technique (SGD). In SGD a random data point/sample for each step and a derivate will be calculated upon that rather choosing all the data points and calculating for them. SGD boosting technique deals efficiently with redundant data where different clusters are randomly formed, in this case a single data point is selected from each cluster known as mini bunch.

1 $F_0(\mathbf{x}) = \arg\min_\gamma \sum_{i=1}^{N} \Psi(y_i, \gamma)$

2 For $m = 1$ to M do:

3 $\{\pi(i)\}_1^N = \text{rand\_perm}\ \{i\}_1^N$

4 $\tilde{y}_{\pi(i)m} = -\left[\frac{\partial \Psi(y_{\pi(i)}, F(\mathbf{x}_{\pi(i)}))}{\partial F(\mathbf{x}_{\pi(i)})}\right]_{F(\mathbf{x})=F_{m-1}(\mathbf{x})}, \ i = 1, \tilde{N}$

5 $\{R_{lm}\}_1^L = L - \text{terminal node } tree(\{\tilde{y}_{\pi(i)m}, \mathbf{x}_{\pi(i)}\}_1^{\tilde{N}})$

6 $\gamma_{lm} = \arg\min_\gamma \sum_{\mathbf{x}_{\pi(i)} \in R_{lm}} \Psi\left(y_{\pi(i)}, F_{m-1}(\mathbf{x}_{\pi(i)}) + \gamma\right)$

7 $F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \nu \cdot \gamma_{lm} 1(\mathbf{x} \in R_{lm})$

8 endFor

Figure 4 Stochastic Gradient Descent Boosting algorithm

5.2

The overall workflow will be discussed in this section, initiating with the dataset description

Sr No.	Attribute Name	Description
1	Step	Unit of time (1 step =1 hour)
2	Type	Type of transaction
3	Amount	Amount of transaction in local currency
4	NameOrig	Customer who initiated the transaction
5	oldbalanceOrg	Initial balance before the transaction
6	NewbalanceOrg	New balance after the transaction
7	nameDest	Customer who is the recipient of the transaction
8	oldbalanceDest	Initial balance recipient before the transaction
9	newbalanceDest	New balance recipient after the transaction
10	isFraud	Transaction marked as fraudulent
11	isFlaggedFraud	Transaction flagged as fraud

Table 2 Attribute description in data set

Initially exploratory data analysis is performed in order to clean the data and prepare it for pre-processing task and further. Since all the task, evaluation measures and algorithm are known hence the required packages are imported and referred. Below are the required packages as per the requirement of functionality:

- Pandas
- Seaborn
- Sklearn.ensemble
- Sklearn.liner_model
- Sklearn.metrics
- Flask
- Numpy
- Matplotlib

5.3 Exploratory Data Analysis Process:

The pre-processing phase begins with handling null value, it is highly essential to deal with null value in the initial phase since a set of null values have potential to manipulate the prediction during the training and testing phase which is not an healthy approach to follow.

```
data.isnull().any()
```

The dataset chosen in this paper doesn't have any null value.

The hypothesis framing is based on two different type of transaction type i.e. Transfer & Cash Our, therefore transaction committed as fraud under the aforementioned transaction types are stored in a variable for future operation.

```
transfer=data[(data.isFraud==1) & (data.type=='TRANSFER')]
cash_out=data[(data.isFraud==1) & (data.type=='CASH_OUT')]
```

Next step is to address redundancy problem, checking for duplicate data points/ instances in the dataset. Redundant data forms multiple cluster which reduced the formation of best fit line that directly reflects into the accuracy level.

```
print(data.loc[data.isFlaggedFraud==1].type.drop_duplicates())
```

Our analysis is based on quantitative analysis hence it is recommended to consider those attributes which reflects integer variables or transform crucial attribute into

numeric/integer value. Thus as discussed before the two transaction type are here transformed here into binary values ‘0’ for Transfer,’1’ for Cash_Out. Furthermore few more attributes are transformed into suitable format and the less relevant are ignored.

```
X.loc[X.type=='TRANSFER','type']=0
X.loc[X.type=='CASH_OUT','type']=1
X.type=X.type.astype(int)
```

Data Preparation for training and testing:

The percentage for training and testing is set to be 70% and 30% respectively.

```
X_train,X_test, y_train, y_test = train_test_split(X, Y, test_size=0.3, random_state=42)
```

Both Transfer & Cash_Out are operated with oversampling mechanism, it is required in order to improve the training output in positive projection.

Algorithm used for oversampling is Synthetic Minority Over-Sampling Technique (SMOTE).

```
sm = SMOTE(random_state = 2)
X_train_res, y_train_res = sm.fit_sample(X_train, y_train.ravel())
```

Now the training model are oversampled the process of testing under the Adaboost and Stochastic gradient boosting algorithm is followed.

Both of the algorithms can predefined in sklearn.ensemble import AdaBoostClassifier, sklearn.liner_model import SGDClassifier package and are ready to use.

For Adaptive Boosting technique:

```
ada= AdaBoostClassifier(n_estimators=10, random_state=42)
```

For Stochastic Gradient Boosting technique:

```
sgd = SGDClassifier(loss="hinge", penalty="l2",
                    max_iter=2)
```

The accuracy level obtained from both of the respective algorithm are as follows:

Adaptive Boosting technique	:	100%
Stochastic Gradient Boosting technique	:	92.76%

According to the analysis conducted it was found that Adaptive boosting technique performed much efficient compared to SGD Algorithm, in next section more detailed will be discuss wherein an actual validation dataset is will we tested and result will be obtained as per the real world application.

Comparative parameter measure undertaken in this paper are as follows:

Accuracy score, Area under the curve, Class Probability, Receiver Operating characteristic curve.

5.4 User Interface and output

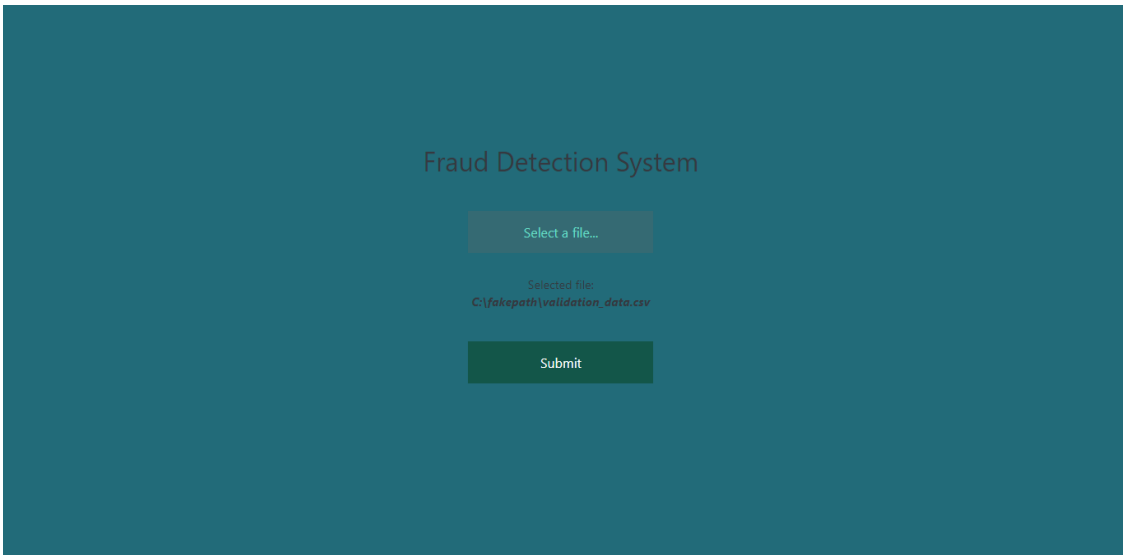
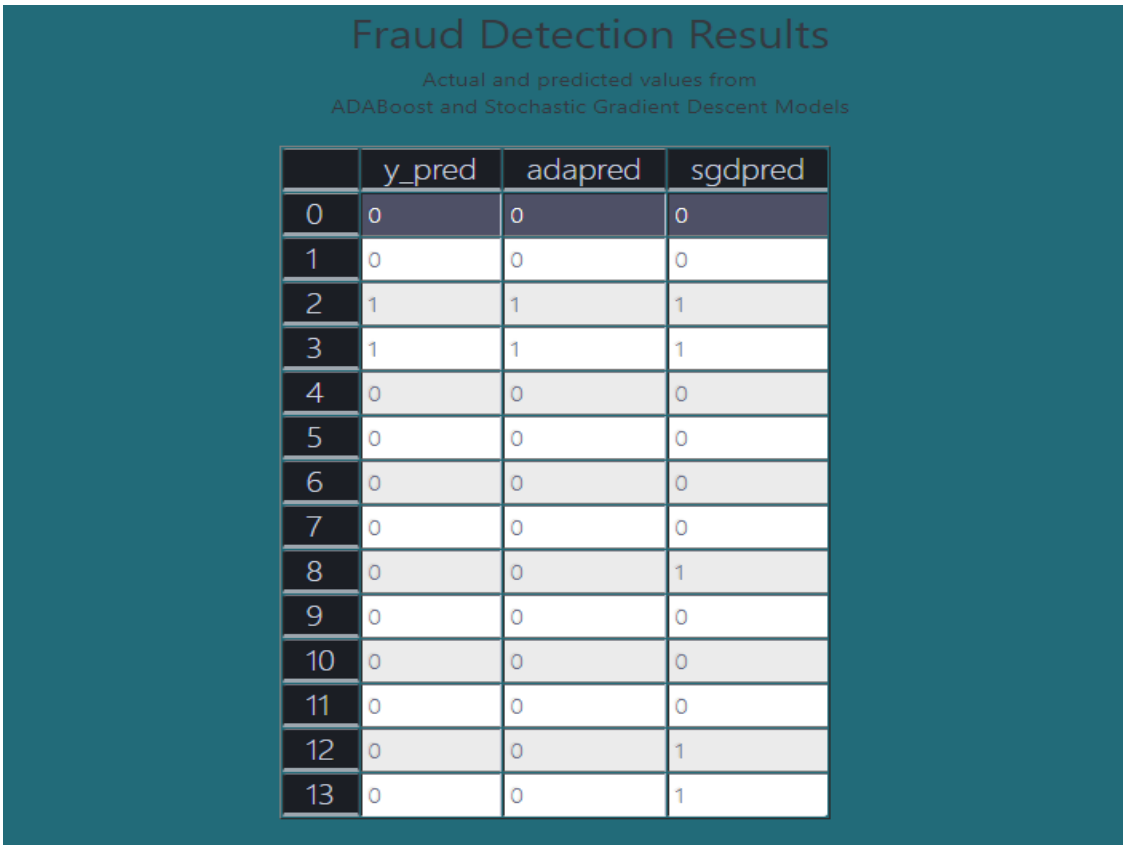


Figure 4 Home page to retrieve validation dataset file



	y_pred	adapred	sgdpred
0	0	0	0
1	0	0	0
2	1	1	1
3	1	1	1
4	0	0	0
5	0	0	0
6	0	0	0
7	0	0	0
8	0	0	1
9	0	0	0
10	0	0	0
11	0	0	0
12	0	0	1
13	0	0	1

Figure 5: Prediction results for Adaboost and SGD algorithms

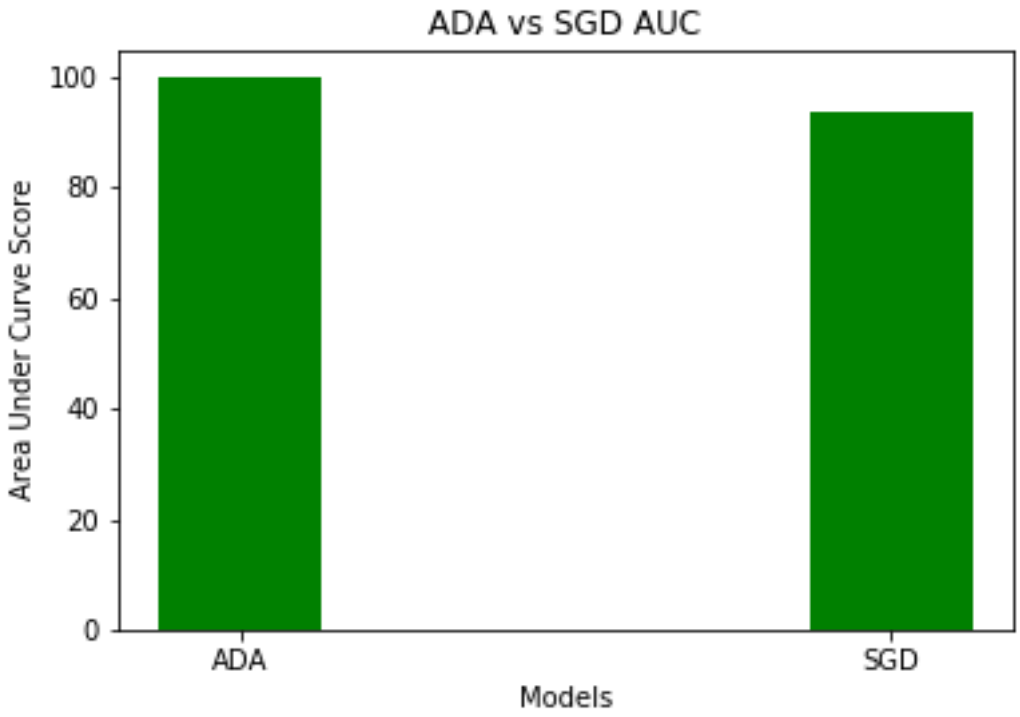


Figure 6 Area Under Curvev score for Adaboost & SGD Algorithm

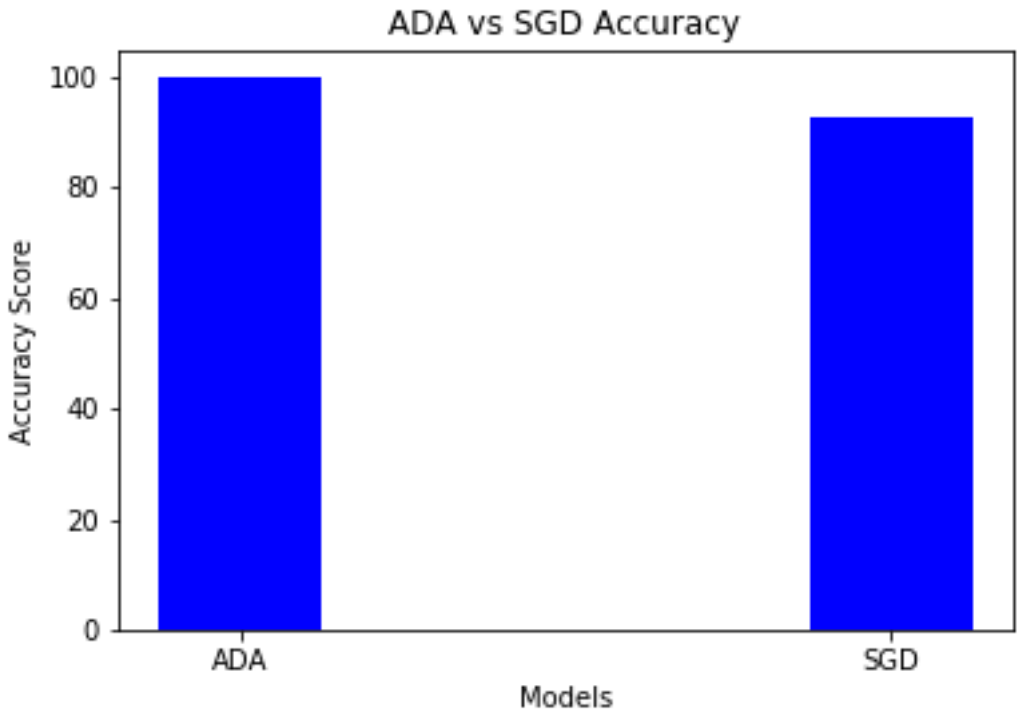


Figure 7 Accuracy score for both Adaboost & SGD algorithm

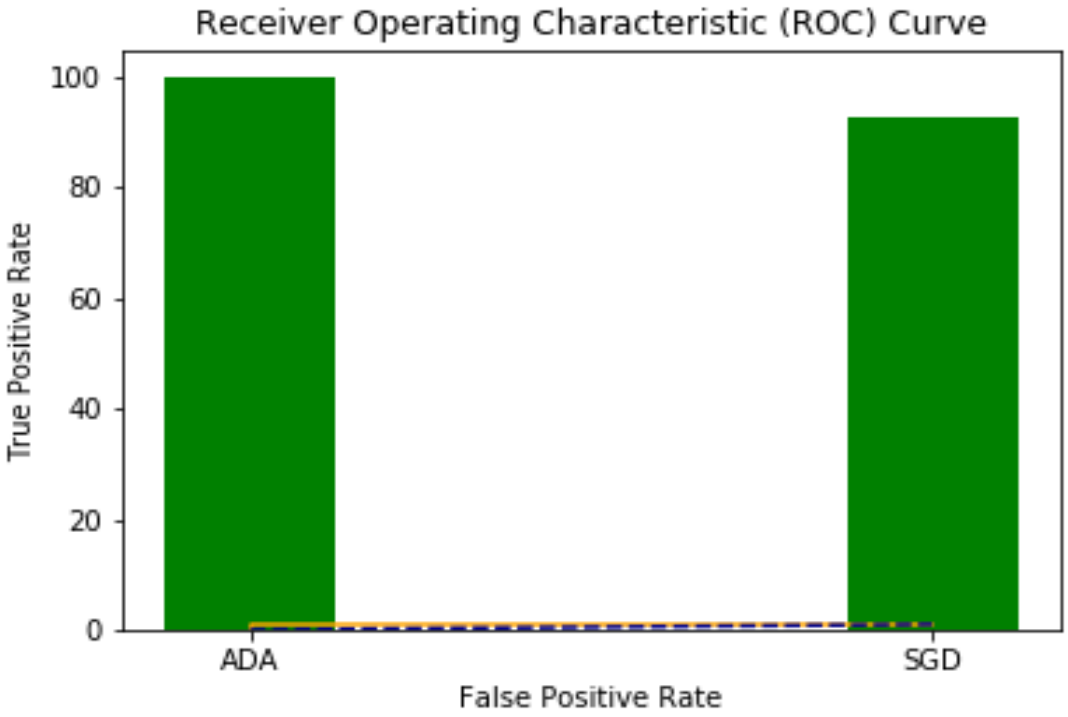


Figure 8 ROC curve score for Asaptive Boosting and SGD models

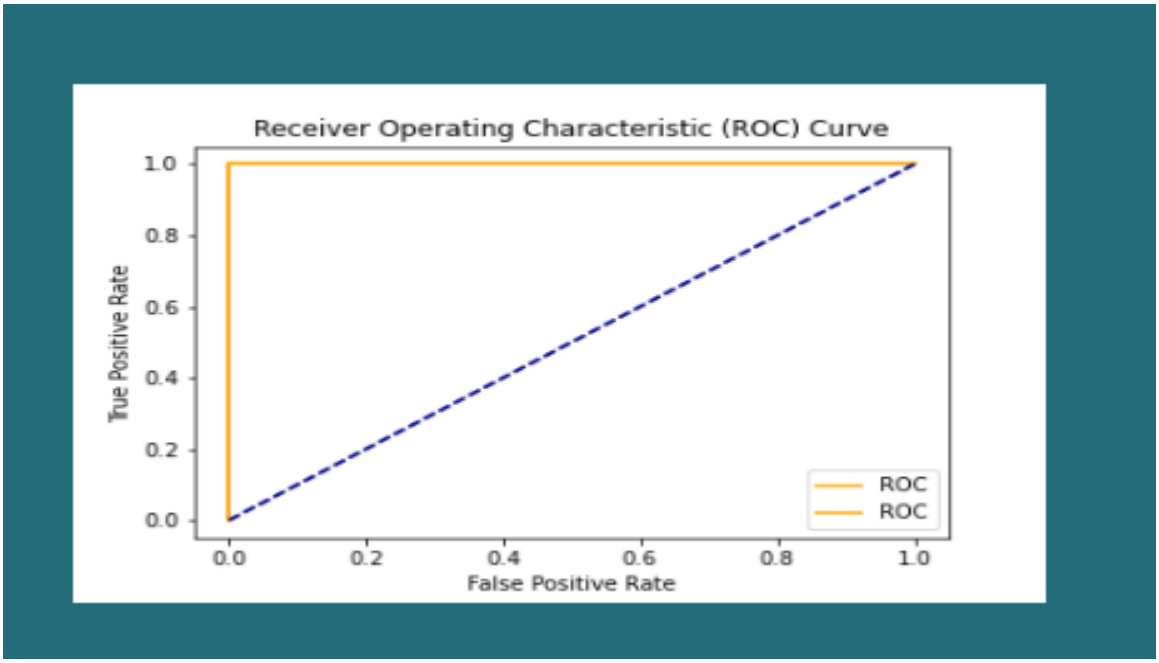


Figure 9: Model curve for Stochastic Gradient Descent model

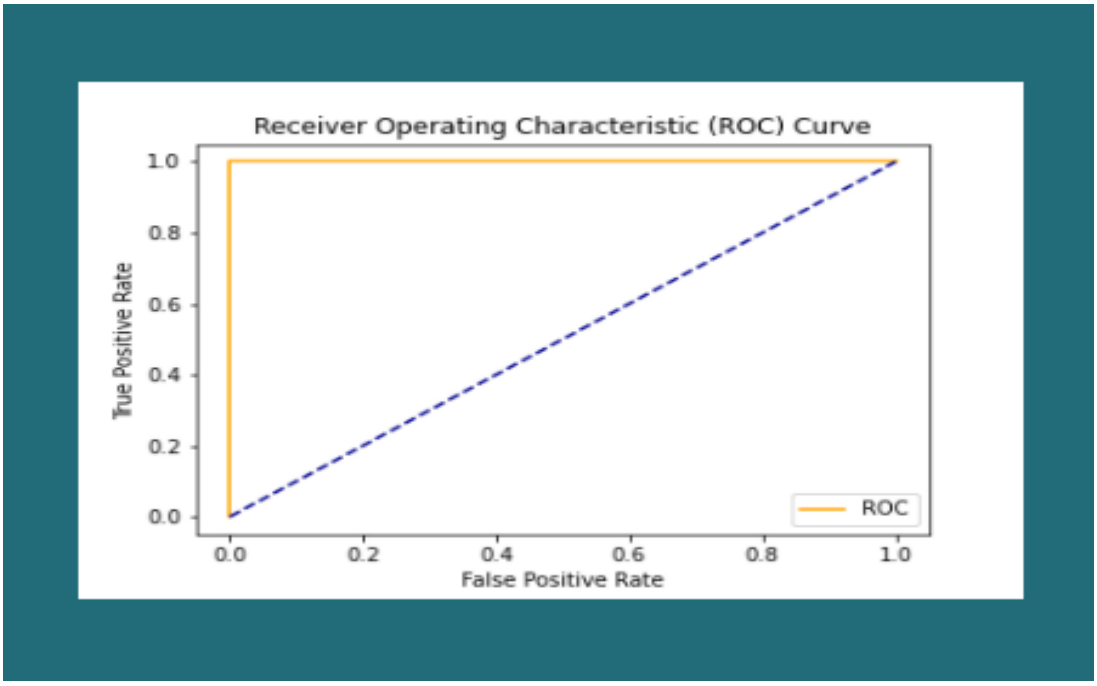


Figure 10 Adaptive Boosting ROC Curve

Chapter 6. Testing and Evaluation

Initially a default value is assigned for maximum iteration while initializing Stochastic Gradient Classifier with value of max_iter=10. During the runtime execution a convergence warning message was thrown as 'Maximum number of iteration reached before..'. Increasing the iteration will reduce the propagated error rate which eventually increase the AUC score and plot best fit ROC curve. Hence 5 different test were conducted, below is the evaluated output with different iteration value.

Test	Maximum iterations	AUC value (%)
1	10	92.76
2	12	93.56
3	15	94.00
4	18	94.15
5	25	96.00

Table 3 Test result with different iteration value for training SGD Model

As per the results the efficiency and accuracy increases with the increasing number of iteration for stochastic gradient descent boosting algorithm.

Furthermore the Adaptive boosting algorithm resulted to 100% of accuracy and AUC score which lead to not conduct further test since the desired value was retrieved. Although the number of estimators was kept as default as compared to SGD algorithm i.e. n_estimators=10

Chapter 7. Conclusions and Future Work

This particular paper addresses the issue of prediction in credit card fraudulent transaction using two different ensemble algorithm which are Adaptive Boosting and Stochastic Gradient Boosting (SGD). The decision of selecting SGD over Gradient Boosting made out a positive result with respect to in comparison among both of them. On overall data modelling and testing the trained data it was found that Adaptive boosting technique has higher accuracy rate and better prediction for problem statement. Further it takes multiple higher number of iteration in order to achieve better result in SGD model whereas Adaptive boosting performed precisely with high through put using lesser number of estimators. The output result supports the decision making of applying Adaboost algorithm in application would result into best outcome compared to any other algorithm discussed or tested in this paper. However the modelling and training consumes plenty amount of time for execution which is reasonable since the data we are dealing here has more than half million records. This problem can be solved by optimizing code and reducing redundancy problem in more efficient manner. Furthermore any machine learning / neural network or hybrid model requires a good processing speed. This implies the fact with higher version of RAM and processing speed the execution of this application is assumed to be quicker and could solve prediction analysis for terabytes and petabytes of data. Finally due to privacy restriction such as GDPR in Europe crucial data of customer are not established to user and developers which could possibly play important role in order to define different hypothesis for solving credit card fraud detection. This phenomena can be addressed in various different ways and optimum result could be retrieved using the best and most optimum attribute from the dataset.

In future work majority voting among all ensemble learning algorithm could be implemented which will ultimately result better output than any other combined neural network or supervised/unsupervised algorithms together.

Bibliography

1. Anuruddha Thennakoon,Chee Bhagyani,Sasitha Premadasa,Shalitha Mihiranga, Nuwan Kuruwitaarachchi. (2019). Real-time Credit Card Fraud Detection Using machine learning. International Conference on Cloud Computing, Data Science & Engineering, 06.

2. Business, U. (2019). Gradient Boosting Machines. Retrieved from uc-r.io: http://uc-r.github.io/gbm_regression

3. Chun-Hua JU, N. W. (2009). Research on Credit Card Fraud Detection Model Based on Similar Coefficient Sum. First International Workshop on Database Technology and Applications, 4.

4. Fahimeh Ghobadi, Mohsen Rohani. (2016). Cost Sensitive Modeling of Credit Card Fraud Using Neural Network Strategy. ICSPIS, 5.

5. Hongyu Wang, P. Z. (2018). An Ensemble Learning Framework for Credit Card Fraud Detection based on Training Set Partitioning & clustering. IEEE SmartWorld, 5.

6. Kosemani Temitayo Hafiz, Dr. Shaun Aghili, Dr. Pavol Zavarsky. (2016). The Use of Predictive Analytics Technology to Detect Credit Card Fraud in Canada. Edmonton, Canada.

7. KULDEEP RANDHAWA, C. K. (n.d.).

8. Anuruddha Thennakoon1, Chee Bhagyani, Sasitha Premadasa, Shalitha Mihiranga, Nuwan Kuruwitaarachchi. (2018). Real-time Credit Card Fraud Detection Using Machine Learning. International Conference on Cloud Computing, Data Science & Engineering, 6.

9. KULDEEP RANDHAWA, CHU KIONG LOO, CHEE PENG LIM, ASOKE K. NANDI . (2018). Credit Card Fraud Detection using Adaboost and amjority voting . IEEE ACCESS, 8.

10. Lakshya Sahai, Kemal Gursoy. (2019). Real-Time Credit Card Fraud Detection. Rutgers University, 7.

11. Mohamed M. Ahmed, M. A.-A. (2019). Application of Stochastic Gradient Boosting Technique to Enhance Reliability of Real-Time Risk Assessment.

12. Navlani, A. (2020). adaboost classifier using python. Retrieved from datacamp.com: https://www.datacamp.com/community/tutorials/adaboost-classifier-python

13. Rahul Goyal, Amit Kumar Manjhvar, Vikas Sejwar. (May 2020). Credit Card Fraud Detection in Data Mining using XGBoost Classifier. IJRTE, 6.

14. Ramzai, J. (2019). Simple guide for ensemble learning methods. Retrieved from towardsdatascience.com: https://towardsdatascience.com/simple-guide-for-ensemble-learning-methods-d87cc68705a2

15. Smolyakov, V. (2019). Ensemble Learning to Improve Machine Learning Results. Retrieved from statsbot.com: https://blog.statsbot.co/ensemble-learning-d1dcd548e936

16. Srivastava, T. (2019). Basics of Ensemble Learning Explained in Simple English. Retrieved from analyticsvidhya.com: https://www.analyticsvidhya.com/blog/2015/08/introduction-ensemble-learning/

17. stochastic gradient boosting. (2018). Retrieved from researchgate.net: https://www.researchgate.net/publication/222573328_Stochastic_Gradient_Boosting

18. Xi, W. (April 2008). Some Ideas about Credit Card Fraud Prediction China Trial.

19. Yang, Y. (n.d.). Ensemble learning. Retrieved from sciencedirect.com: sciencedirect.com/topics/computer-science/ensemble-learning

20. Zhang yongbin, You fucheng, Liu huaqun. (2009). Behavior-Based Credit Card Fraud Detecting Model. Fifth International Joint Conference on INC, IMS and IDC, (p. 4).

21. Zhou, Z. H. (2020). Ensemble learning . Retrieved from springer.com: https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-73003-5_293