

DSpace Institution

DSpace Repository

<http://dspace.org>

Information System

thesis

2020-02-27

Bank Customer Classification and Prediction Using Ensemble Machine Learning Approaches

Ayichew, Ewnetu

<http://hdl.handle.net/123456789/10884>

Downloaded from DSpace Repository, DSpace Institution's institutional repository



BAHIR DAR UNIVERSITY
INSTITUTE OF TECHNOLOGY SCHOOL OF RESEARCH AND POSTGRADUATE
STUDIES FACULTY OF COMPUTING

BANK CUSTOMER CLASSIFICATION AND PREDICTION USING ENSEMBLE
MACHINE LEARNING APPROACHES

EUNETU AYICHEWKASSAW

A THESIS SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES OF THE BAHIR
DAR UNIVERSITY PARTIAL FULFILLMENT FOR THE DEGREE OF MASTER OF
SCIENCE IN INFORMATION TECHNOLOGY.

BAHIR DAR, ETHIOPIA

February 27, 2020

Bahir Dar University
Bahir Dar Institute of Technology-
School of Research and Graduate Studies
Faculty of Computing
THESIS APPROVAL SHEET

Student:

Ewnetu Ayichew Kassaw

Name



Signature

16/06/2012 Ge

Date

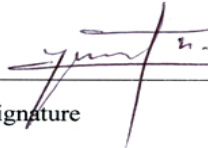
The following graduate faculty members certify that this student has successfully presented the necessary written final thesis and oral presentation for partial fulfillment of the thesis requirements for the Degree of Master of Science in *Information Technology*

Approved By:

Advisor:

Gebevehu Belay (Dr. of Eng.)

Name



Signature

16/06/2012 G.C.

Date

External Examiner:

Dr. Adane Letta

Name



Signature

Date

Internal Examiner:

Abreham Debasu (Ast Prof)

Name



Signature

16/06/2012 Ge

Date

Chair Holder:

Derejow
Dr. Tesfa Adugna

Name



Signature

16/06/2012 G.C.

Date

Faculty Dean:

Selete B.

Name



Signature

16/06/2012 G.C.

Date



DECLARATION

I, the undersigned, declare that the thesis comprises my own work. In compliance with internationally accepted practices, I have acknowledged and referred all materials used in this work. I understand that non-adherence to the principles of academic honesty and integrity, misrepresentation, fabrication of any data/information will constitute sufficient ground for disciplinary action by the University and can also invoke penal action from the sources which have not been properly cited or acknowledged.

Name of the student: Runeel Michael

Date of submission: 18/01/2016 G.C.

Place: Habit One

This thesis has been submitted for examination with my approval as a university Advisor.

Advisor Name: Gehajulu S. (20)

Advisor's Signature: [Signature]

Acknowledgment

I would like to praise God and thank full for all his grace, mercy and strength that has sustained me throughout this time of my life and for his help to me to realize this work. Then, I was glad for the support I have received from many people and without them the completion of this study would have been very difficult.

Sincerely, I am glad to express my deep thanks and gratitude to Dr Gebeyehu for his dedicated support, energetic guidance, valuable advice and abundant experience throughout this work. I am indebted to Mr. Asegahen for his commitment and follow up to accomplish our tasks by alerting and pre-informing about the schedules.

Finally, I would also like to express special thanks to my dearly loved family for their love and support while doing this work (I have done this work on their time).

ABSTRACT

In the competitive banking industry, knowing the customer status and their interest creates an important aspect in business continuity to provide appropriate service for customers as per the demand and develop strategies for classified selected group customers. Currently there are various classification methods used for prediction of bank customers with different prediction accuracy levels. To compare the accuracy of classification and Prediction of the algorithms for bank customers ensemble prediction methods and to identify the preferable method. To determine bank customer classification and prediction bank customer data collected from UCI and we explore the data first to improve the quality of data set using various data exploration methods. After doing so using XGB ensemble methods we perform a comparative study against other existing methods. In our study Support Vector Machine (SVM), Ensemble Machine Learning (EML), Logistic regression (LR), XGB classifier, Randomforest (RF) have been compared. Our study proved that the use of the XGBoost ensemble method improves the accuracy increased from 74.94% by 5% with XGBboost when tested using python 3.6.5.

Key Words and Phrases: *Support Vector Machine (SVM), Ensemble Machine Learning (EML), Logistic regression (LR), XGB classifier, Randomforest (RF)*

TABLE OF CONTENT

ABSTRACT	v
TABLE OF CONTENT	vi
LIST OF ACRONYMS	viii
LIST OF TABLES	ix
LIST OF FIGURES	x
CHAPTER ONE	1
INTRODUCTION	1
1.1 BACKGROUND	1
1.2 STATEMENT OF THE PROBLEM	2
1.3 OBJECTIVE	3
1.4 SCOPE AND LIMITATION	3
1.5 METHODOLOGY	3
1.6 SIGNIFICANCE OF THE STUDY	7
1.7 THESIS ORGANIZATION	8
CHAPTER TWO	9
LITERATURE REVIEW	9
2.1 LITERATURE REVIEW DISCUSSION	9
2.2 ENSEMBLE LEARNING METHODS	12
2.2.1 TYPES OF ENSEMBLE ALGORITHMS	13
2.2.1.1 BAGGING	13
2.2.1.2 BOOSTING	14
2.2.2 CUSTOMER CLASSIFICATION	15
2.2.3 APPLIED METHODOLOGIES	16
2.2.3.1 LOGISTIC REGRESSION	16
2.2.3.2 SUPPORT VECTOR MACHINE (SVM)	17
2.2.3.3 RANDOM FOREST	18
2.2.3.4 XGBOOST	18
CHAPTER THREE	21
METHODOLOGY	21
3.1 ARCHITECTURE OF THE DESIGNED PROPOSED PREDICTION MODEL	21
3.2 DATA PREPROCESSING	23

3.2.1	Bank Data Set	24
3.2.2	Normalizat ion	25
3.3	EXPERI MENTAL METHODS	25
3.3.1	Data set review	25
3.4	MODEL EVALUATI ON	26
3.4.1	Precisi on	27
3.4.2	Recall	27
3.4.3	F measure(F1 score)	28
3.4.4	Area under ROC curve(AUC)	28
CHAPTER FOUR		29
EXPERI MENTAL RESULTS AND DISCUSSI ON.		29
4.1	INTRODUCTI ON	29
4.2	EXPERI MENTAL RESULTS AND ANALYSI S	29
4.2.1	The experi mental result using Logistic Regressi on (LR)	43
4.2.1.1	Logistic regressi on using pri mal predicti on	43
4.2.1.2	Logistic regressi on with poly 2 predicti on	44
4.2.2	Support Vect or Machi ne (SVM) Experi mental result	44
4.2.2.1	SVM with RBF kernel	44
4.2.2.2	SVM with H oy kernel	45
4.2.3	The experi mental result using Rando m Forest Classifi er	45
4.2.4	Experi mental result using XGB Classifi er	46
4.3	DISCUSSI ON OF THE RESULT	46
4.3.1	Test model predicti on accuracy on test data	47
CHAPTER FIVE		49
CONCLUSI ON AND RECOMMENDATI ON.		49
5.1	CONCLUSI ON.	49
5.2	RECOMMENDATI ON&FUTURE WORKS	50
REFERENCE		51
ANNEXERS		53

LIST OF ACRONYMS

ANN	Artificial Neural Network
AUC	Area Under The Curve
BAGGING	Bootstrap Aggregation
CRISP- DM	Cross Industry Standard Procedure for Data Mining
CRM	Customer Relation Management
CSV	Comma Separated Value(s)
CV	Cross Validation
DF	Data Frame
DT	Decision Tree
EML	Ensemble Machine Learning
EML	Ensemble Machine Learning
FN	False Negative
FP	False Positive
GSCV	Grid Search Cross- Validation
KNN	KNearest Neighbor
LBFGS	Limited- memory Brodyen- Fletcher- Goldfarb- Shanno Algorithm
LSVM	Linear Support Vector Machine
ML	Machine Learning
ML	Machine Learning
NN	Neural Network
RBF	Gaussian radial basis function
RF	Random Forest
ROC	Receiver Operating Characteristics
SAG	Stochastic Average Gradient:
SVM	Support Vector Machine
TP	True Positive

LIST OF TABLES

Table 1:- Performance for bank marketing response prediction	11
Table 2:- Potential bank customer feature variables status	25
Table 3:- Potential bank customer dataset feature	24
Table 4:- Data set Data frame	31
Table 5:- Customer balance and yearly income ratio, tenure and age, no of transaction with age	34
Table 6:- Additional trained feature	34
Table 8:- Accuracy Results for logistic regression using primal prediction	44
Table 9:- Accuracy Results for logistic regression using poly 2 prediction.	44
Table 10:- Accuracy Results for SVM RBF kernel	45
Table 11:- Accuracy Results for SVM Poly kernel.	45
Table 12 :- Accuracy Results for logistic regression using poly 2 prediction.	45
Table 13:- Accuracy Results for logistic regression using poly 2 prediction.	46
Table 14:- model prediction Accuracy result.	48

LIST OF FIGURES

Figure 1. Architecture ensemble potential bank customer prediction	31
Figure 2. Unique count for each attribute variables	29
Figure 3. Potential customer and non potential customers	31
Figure 4. The 'Status' relation with categorical variables	32
Figure 5. Relations based on the continuous data attributes	33
Figure 6. Training ROC curve	41
Figure 7. Training result of balance and incremental ratio and Age and tenure	35
Figure 8. Model comparison for all used models	47
Figure 9. ROC of all models in the training	54
Figure 10. ROC curve of Randomforest.	48
Figure 11. Final ROC curve in test data	56

CHAPTER ONE

INTRODUCTION

1.1 BACKGROUND

For profit maximization of bank sector the key pillars are applying controllable expense, service providing efficiency improvement and having good strategy. To be excelling in service providing banking sectors need to know and manipulate the data of customers, Bank customers should be monitored and managed by appropriate corrective measures mostly that could be taken by Customer Relationship Management. The banking service profits are always directly related the service excellence and product varieties of banking service in order to Creating and implementing comprehensive models of customer profiles. [1]

Studying the customer classification and prediction techniques for bank customers and identifying the effective techniques are based on the different metrics like accuracy, error rate, recall, specificity and others will be useful for the banks to design the promotion strategy for the new product in particular and for the existing products in general. Prediction of which customer group will use for the new products based on the previous historical data. From our experiment, we were able to identify the best classification and prediction techniques for the bank data set based on efficiency. Banking industries have more customers and distributed branches; which is a large number to predict without the application of machine learning.

Hence customer prediction and classification shall be applied by customers using various prediction methodologies which yields difference prediction accuracy [2][3]. Applying a single methodology is prone to bias and over fitting. Now days to improve the accuracy of prediction applying ensemble technique provide valuable improvement for the customer prediction [4][5]. In this study we have developed new ensemble techniques to predicting customer using Ensemble machine learning technique.

1.2 STATEMENT OF THE PROBLEM

Customer behavior is one of the treasures for modern digital Banking Professionals [5]. With this newfound information, the bankers can explore the unique traits and habits of each customer 'bucket,' noting where, when, and why deposit or withdrawal occurs, which group mostly used which services of the bank and the like. Banker's insights might even provide the basis to partner or compete with other banking and financial sectors or domains that serve as natural magnets for a portion of the target user base [6].

To gain a vision of the customers who are using, how customers are viewing in online and responding we (the bankers) can compare their activities and interests against the activities of the general public. We were able to a predictive, two-way Banker-customer relationship. As per Harvard Business Review [21] obtaining a new customer for a company is multiple times more expensive than recalling a current one. Accordingly, nowadays most of the financial institutions are concerned with customer retention studies to prevent losing their arcade share and maximize their gained profit from existing customers. Appropriate customer classification and prediction technique supports a lot for guiding proper retention mechanisms by enabling customer service managers to know and understand their customer.

Even though different researchers study customer classification and prediction using various classification and prediction techniques the result of prediction and classification accuracy depends on the algorithms applied, hence in this study bank customers dataset predicted using ensemble technique using python programming to compare the accuracy of the classification and prediction. By using the bank sector dataset, we identified Potential customers to identify and act accordingly for customers and to devise the best mechanism. In the areas of banking and financial sectors, handling the customer's behavior and activities has become a crucial challenge for deciding on the potential business needs hence Ensemble methods (EM) applied to to maximizes classification and prediction accuracy and to minimize the classification and prediction error since various research works confirmed that the application of ensemble method shows a positive impact on classification [7][4][8] due to this applying ensemble method is optimal.

1.3 OBJECTIVE

The general and specific objectives of this study are given below

General Objective: The general objective of this research is to Classify and predict potential bank customers using the Ensemble machine learning method to compare the classification and prediction accuracy of six ensemble algorithms.

Specific Objectives: the specific objectives of this study are to

- ✎ State and explore the various ensemble classification techniques for bank customer prediction and classification
- ✎ Propose better suited ensemble classification techniques suited for bank customer prediction and classification

1.4 SCOPE AND LIMITATION

The scope for this paper is to study and compare logistic regression in the primal space (PS) and with different kernels, SVM in the primal and with different kernels, Random forest classification and XGB Ensemble models for potential bank customer prediction to find and compare accuracy, precision and error rate using online available bank customer dataset. This research is limited for only the online available telemarketing bank data and it can not be used for other bank customers.

1.5 METHODOLOGY

An important part when working with customer classification using an ensemble technique is getting hold of good quality data, which is difficult in the case of bank data due to customer privacy, bank customer data is most sensitive and secured and mostly the customer data only used for the bank data manipulation consumption. The datasets used for conducting the

experiments downloaded from online UCI data. In the study, We have applied SVM, RF, LR and XGB classifiers for classification and prediction in different scenarios for training and testing data to realize the accuracy results. Finally after training 80% of the data set we found best model to test are RF and XGB whereas while testing the dataset the best in classification accuracy is XGB classifier.

Implementation tools:- In order to achieve our objective, we used different environments and tools. Python programming language is used to develop the model. It is an interpreted, arranged abnormal state programming language with dynamic semantics. Its abnormal state worked in data structures, combined with dynamic composing and dynamic official; make it appealing for Rapid Application Development, just as for use as a scripting or glue language to interface existing segments together [9].

Python is an interpreted, high-level, general-purpose programming language, Created by Guido van Rossum and first released in 1991. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is regularly depicted as a "batteries included" language because of its complete standard library [9].

The main aims. The first, is to identify and visualize the factors that contribute to being a potential bank customer and the second is to build a prediction model which will classify if a customer is a potential customer or not. In addition to this based on the model performance and the probability to make easy for customer service management to focus on the actions that can be won or obtained with little effort in their effort to keep the customers potential and to protect the tendency of other customers who are not potential.

The two main tasks that we have done here are exploring the structure of our data, to understand the input space of the data set and to prepare the sets for exploratory and prediction tasks. To do so the following tasks have been done using python. Firstly, the important modules and libraries of python have been configured and imported for our work to calculate mathematical tasks and to draw a graph as per our need. The major libraries that we have been used for the experiment which includes panda, Numpy, Matplotlib and others [10].

Panda:- it is a Python Data Analysis Library which is quite a game changer to analyzing data with Python and the most preferred and widely used tools in data munging/wrangling

Nu mPy:- NumPy is the fundamental package for scientific computing with Python, It contains a powerful N-dimensional array object. NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. It provides fast and efficient operations on arrays of homogeneous data hence it extends python into a high-level language for manipulating numerical data.

Mat plot. pyplot: - is a collection of command style functions that make matplotlib work like MATLAB. Each plot function makes some change to a figure: e.g, creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.

Seaborn: - is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

Mat plotlib - is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter.

Sklearn preprocessing:- the transformations applied to your data before feeding it to the algorithm. sci-kit-learn library has a pre-built functionality under sklearn preprocessing. Sci-kit-learn is machine learning library for the Python programming language which is used for the application of classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

Import polynomial features: - Generate a new feature matrix consisting of all polynomial combinations of the features with degree less than or equal to the

Sklearn model selection:- Model selection is the process of choosing between different machine learning approaches like SVM, Logistic regression or choosing between different hyperparameters or sets of features for the same machine learning approach

Import cross_val_score:- Cross-validation is an important technique often used in machine learning to assess both the variability of a dataset and the reliability of any model trained using that data. It divides the dataset into some number of subsets (*fold*s), builds a model on each fold, and then returns a set of accuracy statistics for each fold. By comparing the accuracy statistics for all the folds, you can interpret the quality of the data set and understand whether the model is susceptible to variations in the data. Cross-validate also returns predicted results and probabilities for the dataset, so that you can assess the reliability of the predictions.

Sklearn model selection:- enables us to import `gridsearchCV` module which used to Find Parameters Producing the Highest Score. Now we are ready to conduct the grid search using `scikit-learn's GridSearchCV` which stands for grid search cross-validation. By default, the `GridSearchCV`'s cross-validation uses 3-fold `KFold` or `StratifiedKFold` depending on the situation. `GridSearchCV` implements a 'fit' method and a 'predict' method like any classifier except that the parameters of the classifier used to predict is optimized by cross-validation.

Sci.py.state:- SciPy builds on the NumPy array object and is part of the NumPy stack which includes tools like `Matplotlib`, `pandas`, and `SymPy`, and an expanding set of scientific computing libraries. [11]

Python fit models:- including `Sklearn.linear_model` for importing logistic regression and `Sklearn.svm` for importing SVM modules. [9]

Sklearn ensemble:- The goal of ensemble methods is to combine the predictions of several base estimators built with a given learning algorithm in order to improve generalizability/robustness over a single estimator.

RandomForest Classifier:- enables to import `sklearn.ensemble.RandomForestClassifier`. A `randomforest` is a meta estimator that fits a number of decision tree classifiers on various sub-

samples of the dataset and uses averaging to improve the predictive accuracy and control overfitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement if `bootstrap=True` (default).

Xgboost:- used to import XGBClassifier By Jason Brownlee on August 17, 2016 in XGBoost. XGBoost is an algorithm that has recently been dominating applied machine learning for structured or tabular data. XGBoost is an implementation of gradient boosted decision trees designed for speed and performance.

Python scoring functions:- includes Sklearn metrics to import `accuracy_score`, `classification_report`, `ROC_auc_score` and `Import roc_curve`. Scoring is also called prediction and is the process of generating values based on a trained machine learning model, given some new input data. The values or scores that are created can represent predictions of future values, but they might also represent a likely category or outcome.

Best Model selection functions:- includes `Model.best_score`, `Model.best_params_`, `Model.best_estimator_` modules

1.6 SIGNIFICANCE OF THE STUDY

Bank customer classification and prediction using ensemble method highly significant for banking sectors to identify potential customer's classification and prediction. The Implementation of LR, SVM, RF and XGB classifier for researchers owing to extend the study with algorithmic adoption to increase accuracy further to ease the identification of potential customer for the bank/ CRM to assist service providing efficiency and to incorporate while developing strategies.

In this study Logistic regression with primal and degree 2 parameters, SVM with RBF and poly kernel, Random forest and XGB classifier studied and compared for potential bank customer prediction to increase the accuracy by applying ensemble technique. Identifying potential customer is a very critical task for the organization continuity in general and for CRM in particular [7]. Trying to identify customers literally might result in a completely incorrect analysis of the data. Therefore, Ensemble classification can help to improve potential customer

classifications and help to provide bank benefit packages accordingly and to treat those other customers with various negotiating mechanisms.

It is clear that new customers can be much costlier than retaining existing ones according to Harvard Business Review. A potential customer may worth as Millions of \$ in future. Thus results of this study can be used as an input to the development of bank customer classification for different purposes while providing loan and launching new products.

1.7 THESIS ORGANIZATION

This thesis is organized into five chapters consisting of Introduction, Literature review, Methodology, Experimental results and discussion finally conclusion and recommendations.

The first chapter gives the general introduction of the thesis that contains an overview of the study, the Statement of problem, motivation, objectives, methodology, Scope of the study, limitation, Procedures, the study Significance and thesis organization.

The second chapter presents reviews made on different kinds of literatures regarding Ensemble machine learning approaches ensemble learning methods, potential customer prediction approaches and different ML techniques as well as previous related works review discussion for bank tele marketing customers of term deposit subscription.

Chapter three illustrates methodology of bank customer classification and prediction including the architecture of the designed proposed classification and prediction model, implementation approach and experimental settings, data preprocessing for potential bank customer prediction, experimental methods, hypothesis formulation, significance test, model evaluation and implementation tools.

The fourth chapter discusses the experimentation and discussion of the findings of how each six experiments and methodologies were implemented and discussion of the result. Finally, the conclusion and recommendations have been drawn from the findings of the study.

CHAPTER TWO

LITERATURE REVIEW

2.1 LITERATURE REVIEW DISCUSSION

Customers are the foundation for every business success, but all customers are not equally important for business. Since banking services are established to provide service for customers and gain profit, easily identifying the best potential customer shall be investigated and predicted timely before losing the valuable customers, to do so, applying EML on the selected sector customers [5]. As per the status of the classified customer business expansion or revision can be implemented.

In general classification is a scheme for information compression and as a transductive prediction error. Thus, the use of classification and prediction algorithms combinedly results in best prediction accuracy [Reference 12]. To classify and predict for a collected and prepared datasets for different classification algorithms implemented. The implementation includes LR in the PS and with different kernels, SVM and with different kernels and EM at different scales and for each scale we train the predictors with sets of predictions and finally, the prediction is combined by ensemble mechanism. Applying ensemble techniques by combining the output will yield higher accuracy and resilient to noise and class imbalance [12].

Accurately classified and predicted customers highly important for taking strategic actions especially for Customer relationship management (CRM). CRM became a great managerial strategy in many highly competitive organizations. The aim of CRM is to know the customer's profitability and recall profitable. CRM may collect customer data from various sources like from the database, from online or the pool to identify targeted customers and provide selected services for customer behavior. As a result, many companies have to measure their customer's value in order to recall or profit potential customers [13]. ML methods are used for classification and prediction purposes in the most application areas including in medical and banking areas. Some of the widely used Classification and predictive algorithms are Logistic regression, Naïve Bayes classifier, SVM, Random forest and neural network [5]. These algorithms are applied on bank customer data set and been analyzed significantly separately for different parameters.

Applying a single learning algorithm for bank customer classification and prediction the result will face three challenges and the challenges can be overcome by applying EML techniques. The three challenges are statistical problem, computational problem and the representation problem. A single algorithm has bias or overfit problems and can be overcome by ensemble. Customer classification, marketing response predictions and advantages of classification and prediction to improve customer loyalty and company profit [1]. To select one prediction from the other there are various evaluation metrics. These evaluation metrics are accuracy, error rate, recall, specificity. For such reason different algorithms yield different cumulative results [14].

As per the previous studies a single ML technique will not be suitable and best fit for all types of data sets and area of study [8]. Once after selecting the list of classification techniques, comparing each algorithm will not avoid errors at the negotiable level. To alleviate the accuracy gap of each ML algorithms' problems can be minimized by applying EML method. Ensemble techniques work in different ways, some work by building a lot of base classifiers and after that classify data focuses by taking a vote of their predictions [15]. This classifier which groups the classifiers, decisions combined and (typically by unweighted or weighted voting) to categorize new examples.

Different scholars clearly stated the importance and usage of data mining and ML applications. LR calculation is utilized for foreseeing factors with the limited arrangement of qualities. LR is based on maximum probability estimation rather than the estimation of least squares which is used in traditional multiple regression analysis, and hence requires more input data for better results. RF, however, represents the state of art in classification and regression in addition to this the experiment which is done using RF were more attractive smaller datasets the results obtained by implementing an RF on the entire dataset and those obtained using the combination of predictions obtained at different scales of clustering did not have a statistically significant difference [13] [8].

Elsalamony (2014) utilized three factual measures; order exactness, affectability, and explicitness on the bank dataset – He thought about and assessed the grouping execution of four distinct information mining procedures' models; Multilayer Perceptron Neural Network

(MLPNN), Tree Augmented Naïve-Bayes (TAN), Logistic (LR) and C5.0 Decision Tree Classifier. He announced that the C5.0 model accomplished somewhat preferable execution over the MLPNN, LR, and TAN. Nachev (2015) connected cross-validation and numerous keeps running for the parceling of train and test sets (70% and 30%) for the immediate showcasing reaction task. He discovered that the two concealed layers engineering proposed by Elsalamony (2014) could be rearranged into a solitary layer structure. He played out a near examination of Neural Networks (NN), LR, Naïve Bayes, Linear and Quadratic Discriminant Analysis (QDA) considering their presentation at different

In the following table 1.1 demonstrate the presentation results acquired by various creators lately when distinctive arrangement calculations were advanced for the bank client advertising forecast assignment utilizing comparative dataset. The most well-known measurement for execution assessment among creators is the AUC, yet a few creators restored the order blunder rates as execution metric. Three arrangement calculations to be specific; Support Vector Machine (SVM), Ensemble Machine Learning (EML), Logistic regression (LR), XGB classifier, Randomforest (RF) utilized for displaying the bank dataset in this investigation. While the examination isn't intended to recreate past investigations on the bank client advertising reaction expectation, none-the-less the exhibition of the RandomForest outfit will be contrasted and best in class results gotten by different creators that utilized comparable dataset so as to appropriately arrange the result in writing. Grafted by Prusty (2013) will fill in as standard for this investigation.

Author(s)	Year	Classification Algorithm	AUC	Remarks
Yiyan Jiang	2018	LED, SVM, NN, DT and LR	0.9203	LR outperforms using R language implementation
Clatunji	2016	RF, LR, CART	0.74	RF ensemble
Nachev	2015	NN	0.915	Data saturation, 3-fold cv
Prusty	2013	C4.5	0.939	Balanced, dataset, test validation
Gupta <i>et al</i>	2012	SVM	-	10 fold cross-validation
Moro <i>et al</i>	2011	SVM	0.938	1/3 test validation

Table 1.1:- Performance for bank marketing response prediction

2.2 ENSEMBLE LEARNING METHODS

EML algorithm is projected to do some classification and prediction for various applications such as gene expression [15][8], Bank customer and telemarketing response analysis [16] and for house price estimation [17]. Ensemble learning methods are becoming more important when the single model over fits and if Clustering and prediction results are worth the extra training. Generally, EL is a group learning in which individual models come together to achieve best accuracy. Due to this Ensemble learning which helps to improve the results of various machine learning algorithms to produce a predictive model. The two widely used ensemble models are bagging and boosting. Bagging (Bootstrap Aggregation) involves multiple models of same learning algorithm trained with subsets of data set randomly picked from the data set (training) whereas Boosting technique emphasizes on the data sets which gives the wrong prediction hence the weights are accustomed on the learning of previous model [8].

Ensemble methods (EMs) application has shown a rapid growth for several years in the ML community [18][19]. EL combines group the different models to reduce generalization error with compare to the individual predictors. That is if the individual predictions can be combined to form a single.

Even though there are so many ML methods a single ML technique will not be suitable and the best fit for all types of data sets. To alleviate the accuracy gap of each ML algorithms' problems can be minimized by applying EML method. An EML is a group of predictors to predict target variable and combines to minimize generalization error.

By definition EL is a composite model for classification, depends on various classification algorithms. EMs are said to be successful ML algorithms that combine different models to get an ensemble which should be more accurate than its component members [7]. The inclination for higher classification accuracy makes the ensemble preferable and important.

2.2.1 TYPES OF ENSEMBLE ALGORITHMS

Various types of algorithms are suitable for different application areas and data set types based on size and other criteria's. For this study we have implemented ensemble algorithms since EM algorithms have the better accuracy (low error), high consistency (avoiding overfitting) and the reduction of bias and variance error. Applying this EM will get the better output than compared to single model which have some problems like over fits, and experimental results worth extra training. There are two famous ensemble techniques those are bagging and boosting. EM techniques applied based on two families those are averaging methods and boosting methods. Averaging methods builds several estimators independently and then to average their predictions since the combined estimator is usually better than any of the single base estimator because its variance is reduced. A common example of average boosting includes bagging methods, RFs. By contrast, in boosting methods, base estimators are built sequentially and one tries to reduce the bias of the combined estimator. The motivation is to combine several weak models to produce a powerful ensemble which includes Ada Boost and Gradient Tree Boosting.

2.2.1.1 BAGGING

Bagging as Bootstrap Aggregation (Bagging):- refers aggregation of multiple models that use same learning algorithm trained with a subset of dataset randomly picked from training. Historically bagging was proposed by the distinguished statistician Leo Breiman in 1994 to improve classification by combining classification of randomly generated training sets. Bagging is a ML ensemble meta algorithm which used statistical classification and regression to improve the stability and accuracy of ML algorithms. Bagging works by classifying the training datasets into multiple bags of models and each model trained separately and combined and finally each bagging aggregated reduce the variance and helps to avoid over fitting. Several decision trees which are generated in parallel form the base learners of bagging technique. Data sampled with replacement is fed to these learners for training. The final prediction is then averaged.

2.2.1.2 BOOSTING

Boosting is a machine learning ensemble meta-algorithm for primarily reducing bias, and also variance in supervised learning, and a family of machine learning algorithms that convert weak learners to strong ones [4]. Boosting is based on the question posed by Kearns and Valiant "Can a set of weak learners create a single strong learner?" A weak learner is defined to be a classifier that is only slightly correlated with the true classification (it can label examples better than random guessing). In contrast, a strong learner is a classifier that is arbitrarily well-correlated with the true classification. Boosting method converts a set of weak learners into strong learners. The method to convert a weak learner into strong learner is by taking a family of weak learners, combine them and vote. This turns this family of weak learners into strong learners in mean time the training data kept into single bag and trained until the prediction accuracy improved.

The base learners in boosting are weak learners in which the bias is high, and the predictive power is just a bit better than random guessing. Each of these weak learners contributes some vital information for prediction, enabling the boosting technique to produce a strong learner by effectively combining of these weak learners. The final strong learner brings down both the bias and the variance.

In contrast to bagging techniques like RF, in which trees are grown to their maximum extent, boosting makes use of trees with fewer splits. Such small trees, which are not very deep, are highly interpretable. Parameters like the number of trees or iterations, the rate at which the gradient boosting learns, and the depth of the tree, could be optimally selected through validation techniques like k-fold cross-validation. Having a large number of trees might lead to overfitting. So, it is necessary to carefully choose the stopping criteria for boosting. There are different types of boosting algorithms for different data science applications some of them are Ada Boost, LPBoost, CoBoost, Brown Boost, Gradient Boosting and XGBoost.

Ada Boost :- it is the short for Adaptive Boosting is a machine learning meta-algorithm formulated by Yoav Freund and Robert Schapire, who won the 2003 Gödel Prize for their work. It can be used in conjunction with many other types of learning algorithms to improve performance. Ada Boost is sensitive to noisy data and outliers. In some problems it can be less susceptible to the overfitting problem than other learning algorithms. The individual learners can

be weak, but as long as the performance of each one is slightly better than random guessing, the final model can be proven to converge to a strong learner.

Linear Programming Boosting (LPBoost):- is a supervised classifier from the boosting family of classifiers.

CoBoost:- it is a semi-supervised training algorithm proposed by Collins and Singer in 1999. The original application for the algorithm was the task of Named Entity Classification using very weak learners. It can be used for performing semi-supervised learning in cases in which there exist redundancy in features.

BrownBoost:- is a boosting algorithm that may be robust to noisy datasets. BrownBoost is an adaptive version of the boost by majority algorithm. As is true for all boosting algorithms, BrownBoost is used in conjunction with other machine learning methods. BrownBoost was introduced by Yoav Freund in 2001.

2.2.2 CUSTOMER CLASSIFICATION

Customer classification is the process of identifying which part of customer observation fits on the base of a training dataset on the known group membership. Classifications to identify the which part of classes (sub-populations) a new thing of observation fits, on the base of a training data set covering observations (or instances) on which group membership is known and measured in the case of supervised learning. The task of clustering is to group into objects. Similar to this, objects in the similar group (denoted as a cluster) which more likely to be the same other than to which are indifferent groups.

Clustering is helpful for data analysis and as a preprocessing step for various learning tasks, utilize clustering in prediction can improve prediction precision [12]. Since Clustering is a plan for information compression. It will along these lines (when expressed as a transudative issue for effortlessness) in all likelihood improve the prediction error.

Classification is used to structure data in the required pattern. As indicated by a pre-defined metric data-points focuses on one group by definition exceptionally like each other than to information focuses from different groups. Classification appears to be helpful for predict as it is essentially a plan for information compression [20]. By compression, we learn something intriguing about the structure and the regularities in the data that can be utilized to maybe improve the expectation of accuracy.

2.2.3 APPLIED METHODOLOGIES

In this study from the various ML methods the following methodologies selected and described including their unique features and applications. The applied methodologies are: -LR, SVM, RF, and XGB classifiers.

2.2.3.1 LOGISTIC REGRESSION

Logistic Regression (LR) algorithm is utilized for anticipating factors with the limited arrangement of qualities. In LR the output is a probability distribution with esteem short of one. LR depends on maximum probability estimation instead of the least square's estimation used in customary different relapse examination, henceforth requires more information for better outcomes. It is a probabilistic approach and it provides feature statistical significance.

Linear regression - It works on any size of the dataset and gives data about the significance of features. It is the advantage of linear regression.

Polynomial Regression - Works on any size of the dataset and works a very well on nonlinear issues whereas it needs to pick the correct polynomial degree for a good bias/variance tradeoff.

Decision Tree Regression - It is Interpretability, no requirement for feature scaling since it takes a shot at both linear and nonlinear is whereas its poor results on too small datasets because of the event of overfitting effectively.

Random forest Regression - Powerful and accurate, good performance on many problems including nonlinear, no interpretably overfitting can easily occur

2.2.3.2 SUPPORT VECTOR MACHINE (SVM)

The Support vector machine (SVM) is a supervised learning method that creates input-output mapping capacities from a lot of marked preparing information. The mapping capacity can be either a grouping capacity which may be the classification of the information, or a relapse work. For order, nonlinear portion capacities are regularly used to change input information into a high-dimensional component space in which the information become increasingly detachable contrasted with the first information space. Most extreme edge hyperplanes are then made. The model consequently created relies upon just a subset of the preparation information close to the class limits. Additionally, the model delivered by Support Vector Regression disregards any preparation information that is adequately near the model forecast. SVMs are likewise said to have a place with "Kernel methods". We discuss the accuracy results and performance analysis by computing recall, precision and F-measure.

SVM algorithms use a set of mathematical functions that are defined as the kernel. The function of the kernel is to take data as input and transform it into the required form. Some of the common kernels used with SVMs and their short purposes:

- Polynomial kernel (It is popular in image processing)
- Gaussian kernel (It is a general-purpose kernel; used when there is no prior knowledge about the data)
- Gaussian radial basis function (RBF) (It is a general-purpose kernel; used when there is no prior knowledge about the data)
- Laplace RBF kernel (It is a general-purpose kernel; used when there is no prior knowledge about the data)
- Hyperbolic tangent kernel (We can use it in neural networks)
- Sigmoid kernel (We can use it as the proxy for neural networks)
- Bessel function of the first kind Kernel (We can use it to remove the cross term in mathematical functions)
- ANOVA radial basis kernel (We can use it in regression problems) and
- Linear splines kernel in one-dimension

SVM - The advantage is its performance, not influenced by outlier and not sensitivity to overfitting whereas it is not appropriate for nonlinear problems, not the best choice for a large number of features is the disadvantage.

2.2.3.3 RANDOM FOREST

A random forest is a supervised learning algorithm. It can be used both for classification and regression. It is also the most flexible and easy to use the algorithm. Literally a forest is comprised of trees. It is said that the more trees it has, the more robust a forest is. A random forest creates decision trees on randomly selected data samples, gets a prediction from each tree and selects the best solution by means of voting. It also provides a pretty good indicator of the feature importance.

Random forests have a variety of applications, such as recommendation engines, image classification, and feature selection. Random forests also offer a good feature selection indicator. Scikit-learn provides an extra variable with the model, which shows the relative importance or contribution of each feature in the prediction. It automatically computes the relevance score of each feature in the training phase.

RF Classification - It is a more powerful and accurate good performance on many problems including nonlinear is the advantage in contradiction no interpretability, overfitting can easily occur, need to choose decision for the enormous number is the negative part of the algorithm.

2.2.3.4 XGBOOST

XGBoost as we can see in the following Figure 2.1 is the most important Ensemble learning (EL) tool which is used for supervised learning. An advantage of utilizing the ensembles of decision tree methods like gradient boosting is that they can automatically provide estimates of highlight significance from a prepared predictive model. Predictions can be consolidated by uniform averaging, weighted averaging or group them together. Averaging prediction combination divides the number of qualities whereas Weighted is when estimates take diverse significance, so you duplicate by their weight (significance) at that point entirely everything up, at that point separate by the absolute weight by simple averaging as it were. the predictions in a

regression task (proportionate to casting a ballot in a characterization task) is most likely the least demanding approach to join them

In the first place, averaging every one of these expectations probably won't be productive as some of them may be poor indicators and in this way may demonstrate to be adverse to the forecast exactness. In this manner, a subset of the absolute number of forecasts acquired must be found the middle value of to improve exactness. Like referenced before, instead of uniform averaging a weighted averaging or the utilization of a troupe strategy could significantly improve the joined forecast. Good model training performance and ability of to build more accurate model are the advantage of XGBoost and taking more time for training due to iteration process are the disadvantage.

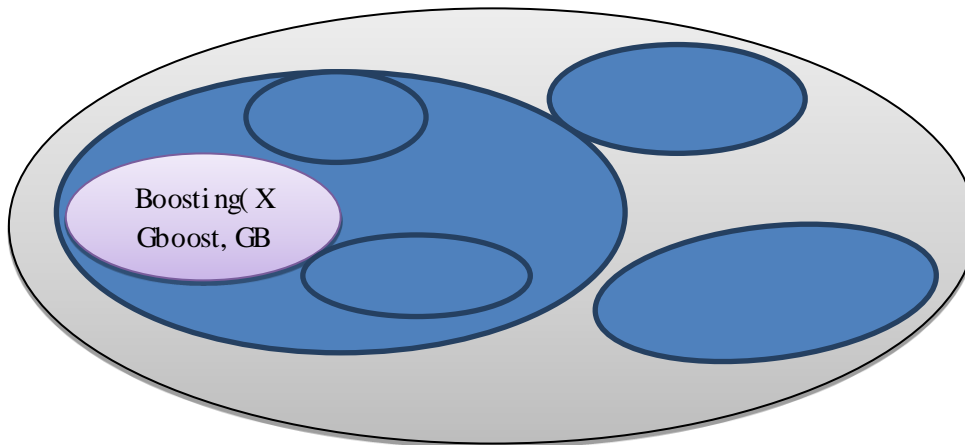


Figure 2.1 Machine learning classification [21]

Unique features of XGBoost:- Some features of XGBoost that make it preferable are:- regularization, Handling sparse data, Weighted quantile sketch, Block structure for parallel learning, Cache awareness, Out-of-core computing

Regularization- XGBoost has an alternative to punish complex models through both L1 and L2 regularization. Regularization helps in avoiding overfitting

Handling sparse data:- Missing information processing steps like one-hot encoding make information meager. XGBoost joins a sparsity-mindful split discovering calculation to deal with various sorts of sparsity designs in the information.

Weighted quantile sketch:- Most existing tree-based algorithms can locate the split points when the information focuses are of equivalent loads (utilizing quantile sketch calculation). In any case, they are not prepared to deal with weighted data. XGBoost has a distributed weighted quantile sketch algorithm to viably deal with weighted information.

Block structure for parallel learning:- For quicker processing XGBoost can utilize various centers on the CPU. This is conceivable in light of a block-based system in its framework plan. Data is arranged and put away in memory units called squares. In contrast to different algorithms this empowers the information design to be reused by ensuing cycles, rather than processing it once more. This component likewise serves helpful for steps like split finding and section sub-testing.

Cache awareness:- In XGBoost, non-constant memory access is required to get the angle insights by line record. Henceforth, XGBoost has been optimal use of hardware. This is finished by distributing allocating internal buffers in each thread, where the gradient insights can be put away.

Out-of-core computing:- This feature advances the accessible disk space and expands its utilization when taking care of enormous datasets that don't fit into memory. XGBoost remains game changer in the ML community.

CHAPTER THREE

METHODOLOGY

3.1 ARCHITECTURE OF THE DESIGNED PROPOSED PREDICTION MODEL

In this chapter, the design and implementation of bank customer classification and prediction using ensemble method elaborated in detail. We adopted our design from Yi yan Jiang which was developed for predicting the success of bank tele marketing using Logistic regression[5].

The general architecture of potential bank customer prediction is given in Figure 3-1. The architecture has five major phases;

The 1st is Bank data collection from online UCI bank dataset

The 2nd phase is Data pre-processing which used to refine our Data cleaning, feature selection, EDA, data transformation and data validation tasks have been done here

The 3rd phase is Implementing the selected algorithms LR with different kernels, SVM in the primal and with different Kernels, RF and at last EM using XGBoost

The 4th step is conducting data analysis and Evaluation to compute using the chosen data and the effectiveness of the proposed models realized by each algorithms accuracy, precision and recall,

The 5th and last step is the end of our job which is analyzing drawing conclusion based on the graphical and aggregated experimental result. On fig 3.1, we showed the interactive and connectiveness of each components in the model [22].

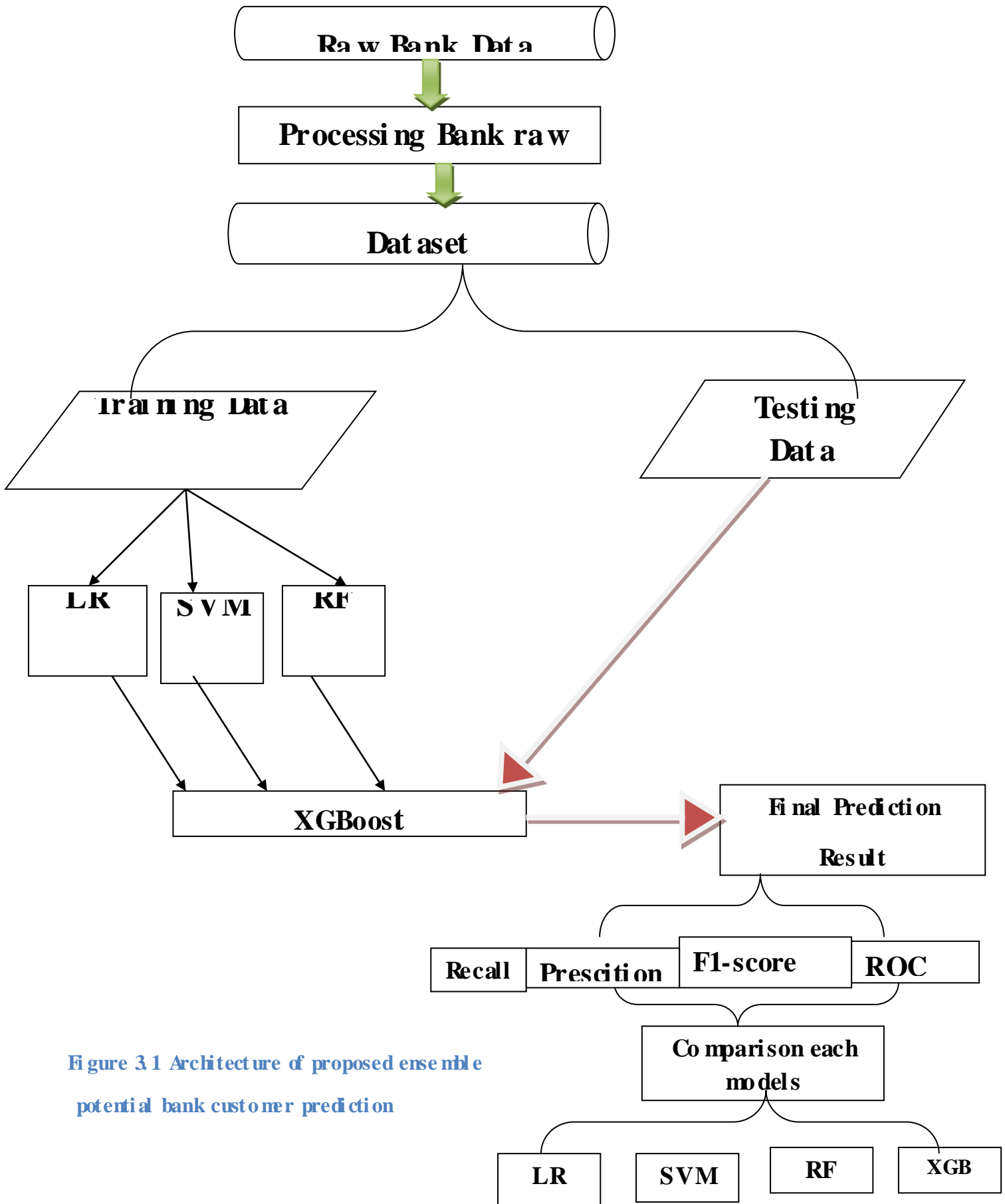


Figure 3.1 Architecture of proposed ensemble potential bank customer prediction

Row Bank Data- Are online available UCI bank dataset

Processing Bank Row Data- The collected row bank data should be processed due to three reasons, those are for fixing missing values, Data Standardization and to optimize variable sets.

Training data- 80% of bank dataset

Test data- 20% of bank dataset

LR, SVM, RF, XGBoost- are methods, training applied in each methods and for each the result of accuracy, F1-score, recall and precision and ROC value registered separately

Final Prediction Result- the comparison of LR, SVM, RF and XGB boost based on each testing Accuracy, AUC and ROC values

3.2 DATA PREPROCESSING

In ML, proper data exploration is the major critical task which has a great impact on the accuracy of the learning prediction since data exploration is the process describing the data by means of statistical and visualization techniques in order to bring important aspects of that data into focus for further analysis. Under data exploration, the following major tasks will be done those are Data feature variable exploration, Data collection, management of outliers, dimensionality reduction and data redundancy resolution.

In order to perform potential bank customer prediction, the data should first be converted into suitable format and arranged in tabular format the pre-preparing movement is imperative to improve the accuracy, efficiency, and scalability of the classification process. The data collected from online available UCI bank data. Therefore, the data must be processed and spoke to a brief and recognizable configuration or structure. Therefore, data preprocessing which used to refine data cleaning, duplicate removal, null value finding and correcting and balancing have been done, here The preprocessing assignment was executed utilizing using a programming language called Python (Python 3.6).

3.2.1 Bank Data Set

After collecting the detail data and Exploratory data analysis were done using Python. The dataset has been pre-prepared and has no missing qualities. Outline of the dataset presents the opportunity to assess certain characteristics

The final collected and prepared bank datasets are with 10,000 instances and 14 features including the row number of which 2037 instances considered as for potential ‘‘yes(1)’’ those are term deposit subscribers and 7963 instances of ‘‘no(0)’’ as a response our interest is in predicting term deposit customers which are 20%

Table 2- Potential bank customer dataset feature

	Feature	Feature description	Type
1	Row number	Number of the row in the dataset based on the chronological order	Numeric/continuous
2	CustomerId	Unique Id of customer given by	Numeric/continuous
3	Surname	Name of customer	Categorical/discrete
4	Nooftransaction	Number of times a customer used bank service in the given time period	Numeric/continuous
5	Gty	Location of a customer in the metropolitan cities	Categorical/discrete
6	Gender	Gender of customer either male or female	Categorical/discrete
7	Age	Age of customer	Numeric/continuous
8	Tenure	Stay of a customer by using bank service	Numeric/continuous
9	Balance	Remaining balance of a customer	Numeric/continuous
10	NumOfProducts	Number of a product provided by the bank	Numeric/continuous
11	Has D Card	Is a customer have a debit card or not	Categorical/discrete
12	inactive member	Customer status as active or inactive	Categorical/discrete
13	yearly incremental	The yearly income of a customer	Numeric/continuous
14	Potential		Categorical/discrete

Dimensionality Reduction:- It is important to decrease the dimensions of the dataset so as to lessen reduce redundancy. Dimensionality reduction should be possible in two distinct ways; one by keeping just the most significant variable from the informational index. This technique is called feature selection. The second technique is through the exploitation of redundant data, and by finding a smaller set of new

variables, each being in the mix of the information factors containing essentially the same data as the information variable.

3.2.2 Normalization

Next, we standardized the data and both axis in order to have a similar proportionate representation of the components. Different features have a different scale of unit measurement. For instance, the feature age is measured in years, balance is a currency unit. The z-score normalization in which the feature variables are standardized dependent on the mean and standard deviation of the features was used. To maintain a strategic distance from the impacts of the individual highlights we normalize by subtracting each instance of the feature variable from the mean at that point isolate the result by the standard time frame instead of deviation of the particular variables, using the bank-potential dataset.

3.3 EXPERIMENTAL METHODS

Experimental methods mainly aimed to accomplish identify and visualize which factors contribute to customer to be potential and to build a prediction model that will perform a customer is a potential customer to select term deposit or not and Preferably, based on model performance, choose a model that will attach the potential and the least participant customers in another way the potential and the customers who doesn't subscribe term deposit. The experimental method has five steps. Those are Dataset review and Preparation, Exploratory data analysis, feature engineering, Data preparation for model fitting and for model selection.

3.3.1 Data set review

In this section, we have identified the structure of data explored in order to understand the input space the data set and to prepare the sets for exploratory and prediction tasks. Practically the important libraries imported, the data frame imported, unique and null values have been checked

From data review the following facts considered

1. The balance is for a given date and depend on the balance at the end of fiscal year.
2. There are customers who are inactive and term deposit subscriber and have a balance in their account
3. Active members are those who uses their account once at least in the three months.
4. Number of transactions is both debit and credit transactions
5. Number of products is only in the number and each product can't be measured here.
6. Using exploratory Data Analysis the bank data (bank- Potential) was thoroughly explored using data visualization techniques and physical assessment of the data. Significant time was spent inspecting the dataset physically in tabular format in python.

3.4 MODEL EVALUATION

This activity is responsible for describing the evaluation parameters of the designed model and its results. Evaluation of the system is made with the evaluation parameter that compares the number of the data which are categorized correctly and incorrectly. The comparison is done between the data categorized by the proposed model system and that of the manually labeled (categorized) data. In order to have a common performance evaluation metric for the classification and ensemble algorithms and the classification accuracy (CA) will be used as the final test of performance. Other relevant metrics such as Precision and Recall will also be accorded a awareness in order to understand and appreciate the performance of the classification and ensemble algorithms on the bank dataset.

In this examination, the presentation of the proposed model is evaluated by taking about the experimental status of test accuracy, recall and f1 score tests in order, the precision is characterized as the quantity of true positives isolated by the whole of sum of true positives and false positives, which is communicated by Equation

3.4.1 Precision

Precision (P): It is very well considered as a proportion of precision, which is the level of examples marked as positive that are actually positive. Precision, in other words, is the fraction or percentage of detected or retrieved instances that are relevant by the classification algorithm. Accuracy is the number of true positives partitioned by the complete number of components marked as having a place with that class. High precision means that the majority of items labeled as for instance 'positive' indeed belong to the class 'positive' and is defined as.

- **True positives** are positive items that we correctly identified as positive for positive class and Negative items that we correctly identified as Negative for negative class.
- **False positives (or Type I errors)** are negative comments that we incorrectly identified as positive for positive class and positive comments that we incorrectly classified as negatives for the negative class.

Precision and recall reach their best value at 100% and worst at 0% while F-measure reach its best value one and worst zero.

3.4.2 Recall

Recall can be considered as a measure of completeness, which is the level of positive examples that are really marked as positive. Recall at the end of the day it is the portion or level of relevant instances that are detected and retrieved by the classification algorithm. The review of grouping is characterized as the number of true positives isolated by the all-out number of components that have a place with the positive classes.

Recall (R) is the number of true positives partitioned by the all-out number of items that really have a place with that class. A high recall implies that most of the 'positive' things were marked as having a place with the class 'positive'.

- **True negatives** are irrelevant items that we correctly identified as irrelevant. (negative comments not classified under positive for positive class and vice versa)
- **False negatives** (or **Type II errors**) are relevant items that we incorrectly identified as irrelevant. (positive comments that incorrectly not classified under positive for positive class and negative comments that incorrectly not classified under negative for negative class.)

3.4.3 F measure (F1 score)

F-measure (F1 score) is defined as the symphonious mean of precision and recall which is a measure that joins Recall and Precision into a single Measure of performance this is only the result of precision and recall divided by their normal the f measure which is appeared by

The F-measure, which is shown in Equation 3.3

$$(3.3)$$

3.4.4 Area under ROC curve (AUC)

The Area under Curve is a metric (usually less than 1.0) that measures the value of ROC. The AUC is a is mostly considered as a generalized measure for a classification algorithm's separation power for more than two classes. The AUC is considered as a more solid measurement and subsequently more acceptable than the classification precision Classification Accuracy (CA).

CA is a metric that estimates the presentation of a classification algorithm with its capacity to accurately order a binary or multi-class response. As such, the classification accuracy is a measure of the proportion of data instances for which the class prediction was correct.

CHAPTER FOUR

EXPERIMENTAL RESULTS AND DISCUSSION

4.1 INTRODUCTION

In this research, six experiments had done with the four learning algorithms Support Vector Machine (SVM), Logistic regression (LR), Randomforest (RF) and Extreme Gradient Boosting (XGB) classifier. The six experiments are Fit primal logistic regression, Fit Logistic regression with degree 2 polynomial kernels, Fit SVM with RBF kernel, fit SVM with Polly kernel, Fit Randomforest classifier, Fit extreme gradient boosting classifier. All the results are presented in the subsequent portion.

4.2 EXPERIMENTAL RESULTS AND ANALYSIS

After importing the above modules and libraries, the second immediate task is to read the processed dataframe (df) to python and to check the imported rows and attributes. In this study 10000 rows and 14 attributes before exploratory analysis and prediction modeling the identifying important attributes the need of data manipulations carried out. All columns are checked for null and their result is 0 which is no null value. As we can see from the result of figure 2 row number, customer id, surname, balance, yearly incremental attributes are specific to a customer. From those row number, customer id, surname is not required for the study since the value of each attribute is specific to customer and it is time and memory taking For each value unique values displayed as described in figure 2.

```
In [36]: # Get unique count for each variable
df.nunique()

Out[36]: RowNumber          10000
CustomerId          10000
Surname              2932
Nooftransaction      460
City                 3
Gender               2
Age                 70
Tenure               11
Balance              6382
NumOfProducts        4
HasDtCard            2
IsActiveMember       2
yearlyincremental    9999
Potential            2
dtype: int64
```

Figure 2- Unique count for each attribute variable

Feature Selection:- In this study so as to decide the most relevant features in order to get a relevant result from the experiment the determinant features should be identified. To do so from 10000 rows and 14 attributes before exploratory analysis and prediction modeling the identifying important attributes the need for data manipulations carried out.

It is important to understand the nature of the distribution of the bank dataset (bank-potential) and its features before any form of analysis is performed on the data. In data exploration feature selection is the major task, feature selection the process of selecting relevant features and discarding irrelevant features. In data exploration Anomalies, outliers and extraordinary qualities were effectively identified during data exploration.

Table 3:- Potential bank customer feature variables status

	Feature	Feature description	Type
1	Row number	Row number from 1 to 10,000	Numeric /continuous
2	CustomerId	8 digit Unique id of a customer	Numeric /continuous
3	Surname	Name of customer	Categorical/discrete
4	No. of transaction	350 to 850 number of transactions	Numeric /continuous
5	Geography	'Spain', 'France' and 'Germany'	Categorical/discrete
6	Gender	'male' or 'Female'	Categorical/discrete
7	Age	18 to 92 years	Numeric /continuous
8	Tenure	0 years to 10 years	Numeric /continuous
9	Balance	Remaining balance on the customer account 0 to 250898.01	Numeric /continuous
10	Number of Products	1 to 4 products	Numeric /continuous
11	Has D Card	1 for 'yes', 0 for 'no'	Categorical/discrete
12	Is Active Member	1 for 'yes', 0 for 'no'	Categorical/discrete
13	yearly increment	11.58 to 199992.5	Numeric /continuous
14	Potential	1 for 'yes', 0 for 'no'	Categorical/discrete

The data frame structure displayed in Table 5, Here our main interest is to get an understanding as to how the given attributes relate to the 'potential' status. The following figure displays potential and non-potential status. As we can see in figure 3 from the total dataset 79.6% are potential and 20.04% customers are non potential hence we are going to evaluate the potential level of the 79.6%. When proceeding our experiment by checking our variable data types, mostly we have a categorical variable and 5 continuous variables.

	Nooftr transaction	Geography	Gender	Age	Tenure	Balance	Num Of Products	HasDt Card	IsActive Member	yearly incremental	Potential	Balance yearly incomeRatio	Tenure ByAge	Noof Transaction Given Age
8159	684	Germany	Female	48	3	73309.38	1	0	0	21228.34	0	3.453373	0.0625	14.25
6332	647	Spain	Male	47	10	99835.17	1	0	1	89103.05	1	1.120446	0.212766	13.765957
8895	756	Spain	Male	41	6	149049.92	1	0	1	50422.36	0	2.956028	0.146341	18.439024
5351	552	France	Male	55	3	0	1	1	1	40333.94	1	0	0.054545	10.036364
4314	530	Germany	Female	36	7	0	2	1	0	80619.09	1	0	0.194444	14.72222

Table 4:- Data set Data frame

Proportion of term deposit subscriber and not subscriber

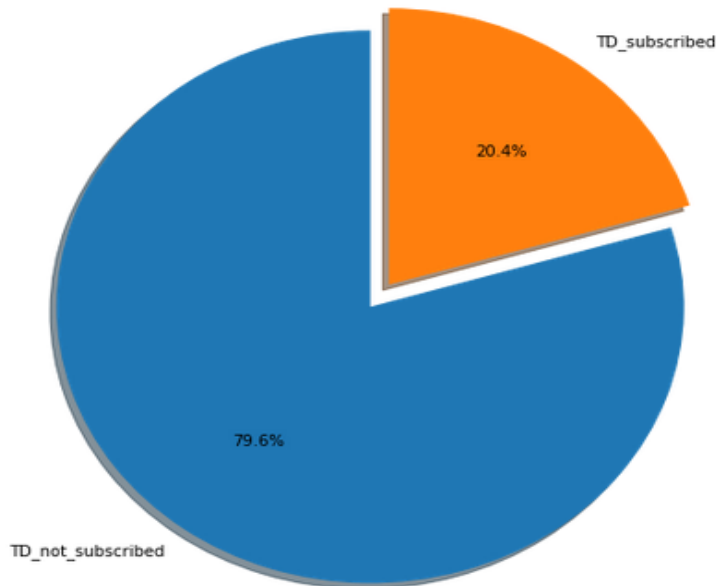


Figure 3:- Term deposit subscriber Potential customer status

The 'Status' relation with categorical variables represented as follows.

As we can see from fig 3 the proportion of term deposit is not related to customer and sometimes said to be inversely proportional. The proportion of Female customer term deposit or greater than that of male customers inactive customers subscribed more on term deposit other than active customers.

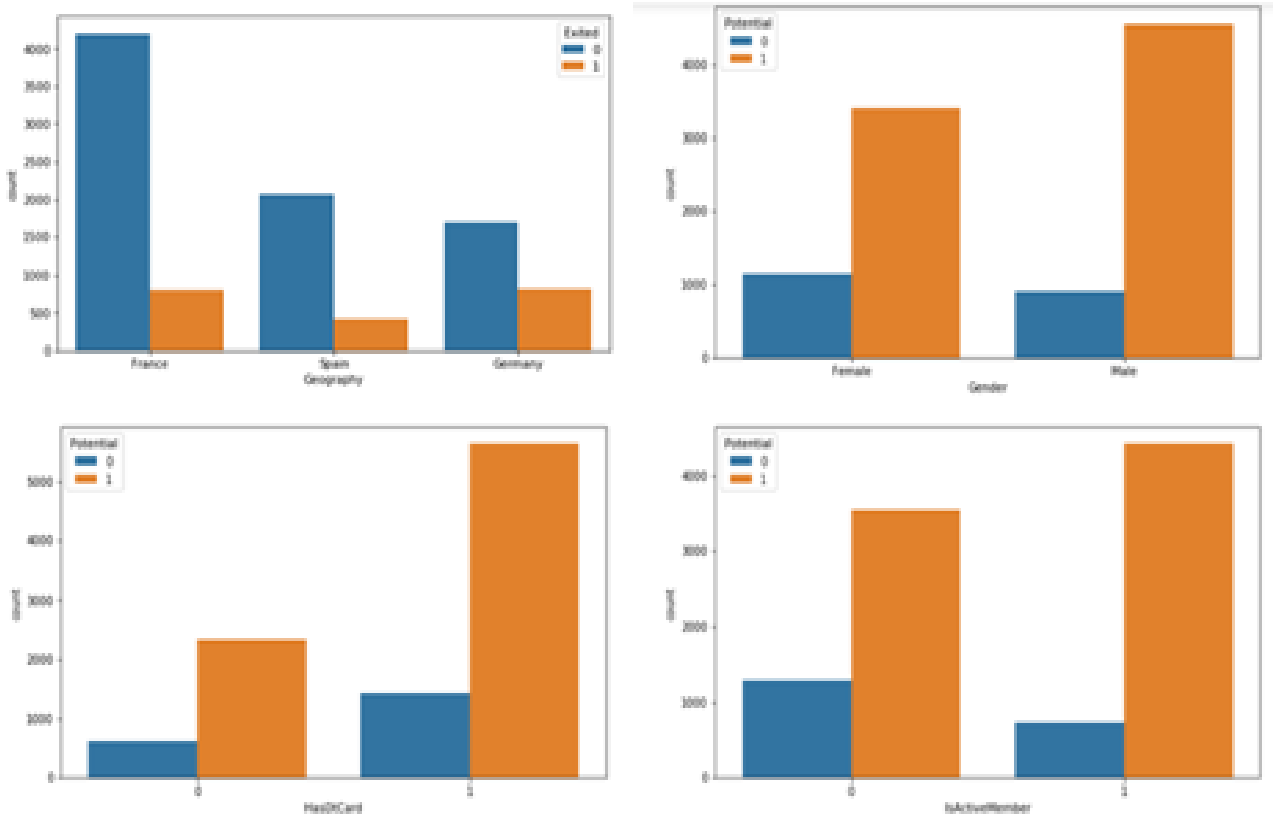


Figure 4:- The 'Status' relation with categorical variables

Relations based on the continuous data attributes

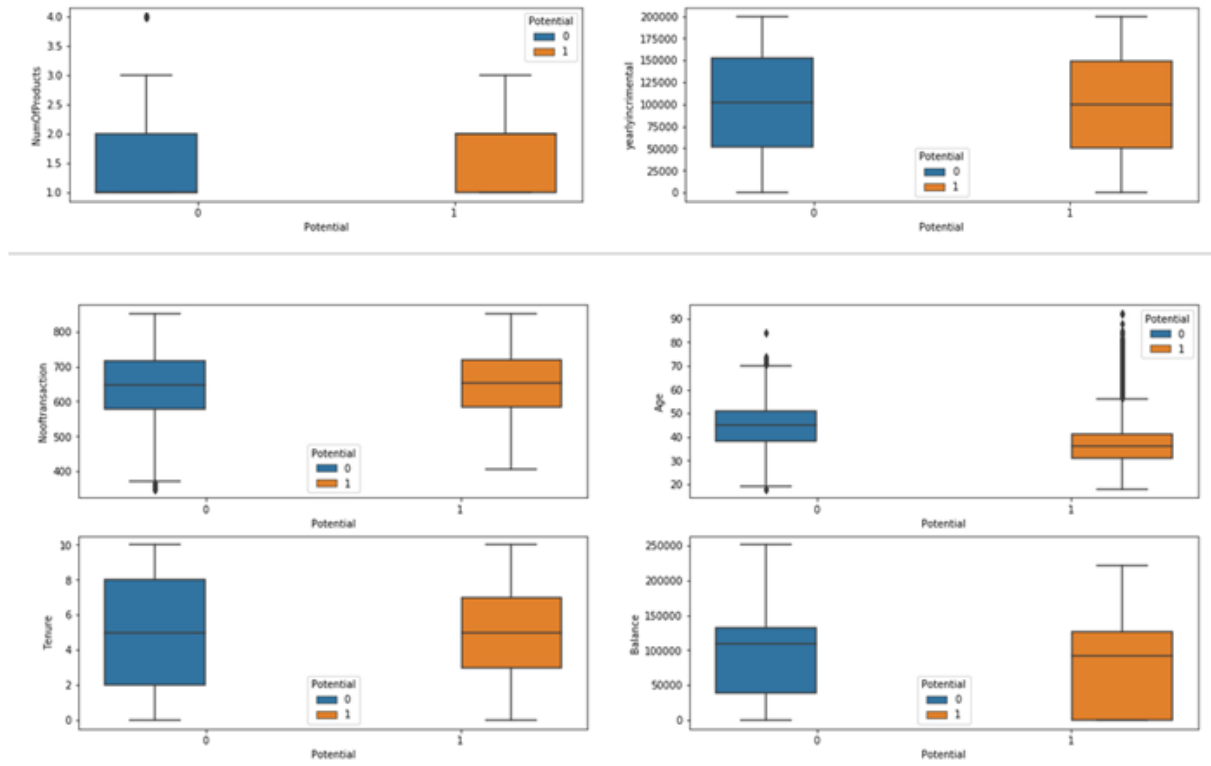


Figure 5:- Relations based on the continuous data attributes

1. Credit score is not significant for term deposit subscription
2. The older customers subscribe term deposit than the younger (there should be motivation for younger's)
3. Tenure, product and salary doesn't have +ve or -ve impact on term deposit subscription customer
4. A customer who has more balance doesn't subscribe term deposit which will be the potential for lending

Using the data set the following features In this part customer balance and yearly income ratio, tenure and age, nooftransaction with age have been visualized.

For easier manipulation columns shall be arranged by both continuous and categorical data types to be trained.

Table 5:- Customer balance and yearly income ratio, tenure and age, no of transaction with age

	Potential	Noof transaction	Age	Tenure	Balance	NumOfProducts	yearlyincremental	BalanceyearlyincomeRatio	TenureByAge	Noof transaction	HasDt Card	IsActive Member	City	Gender
8159	0	684	48	3	73309.38	1	21228.34	3.453373	0.0625	684	-1	-1	Spain	Female
6332	1	647	47	10	99835.17	1	89103.05	1.120446	0.212766	647	-1	1	France	Male
8895	0	756	41	6	149049.9	1	50422.36	2.956028	0.146341	756	-1	1	France	Male
5351	1	552	55	3	0	1	40333.94	0	0.054545	552	1	1	Germany	Male
4314	1	530	36	7	0	2	80619.09	0	0.194444	530	1	-1	France	Female

Feature Engineering :- The main objective of feature engineering is to add features that are likely to have impact on the bank potential customer. In the above we have seen that for continuous variables of test databases The primary task in the feature engineering is to split the training and testing data sets with 80% for training and 20% for testing which means we have used 8000 rows for training and 2000 rows for testing from the 10000 rows of data set.

Classification works by learning from labeled feature sets, or training data. Most papers that we have used for the researches the 20% of the total data used for testing and the remaining 80% for training for our research works [5][17].

In the feature engineering we have checked the relation of balance and Yearly incremental, customer Age with tenure status of a customer and Number of. The detailed result described in figure 7 in detail.

The Balance yearly incremental ratio, Tenure age, Credit Score Given Age have been trained and the resulting data frame have been displayed as follows by omitting Number of the transaction, City, Gender, Age, Tenure, Balance, Number of products, Handcard, Is Active Member, and yearly incremental features.

	Potential	BalanceyearlyincomeRatio	TenureByAge	Noof Transaction Given Age
8159	0	3.453373	0.0625	14.25
6332	1	1.120446	0.212766	13.765957
8895	0	2.956028	0.146341	18.439024
5351	1	0	0.054545	10.036364
4314	1	0	0.194444	14.722222

Table 6:- Additional trained feature

As we have seen in the above continuous data attributes balance and age have an impact on term deposit subscriptions but credit score, yearly incremental, salary and product doesn't have any significant role. After feature engineering Salary has little effect on the chance of term deposit subscriptions. In the case of Balance salary ratio, the customers with a higher balance salary ratio term deposit subscription status also increased.

As per the description in the 3.5.3 The Balance yearly incremental ratio Tenure By Age, No of transaction Given Age have been added and trained. From the below analysis Figure 7 the yearly incremental doesn't have an impact to consider as potential or not potential customer classification whereas the customer with significant bank balance are not potential and the bank is losing. Whereas after taking the balance and yearly incremental ratio we have seen that the yearly incremental has little impact on the chance of potential customer.

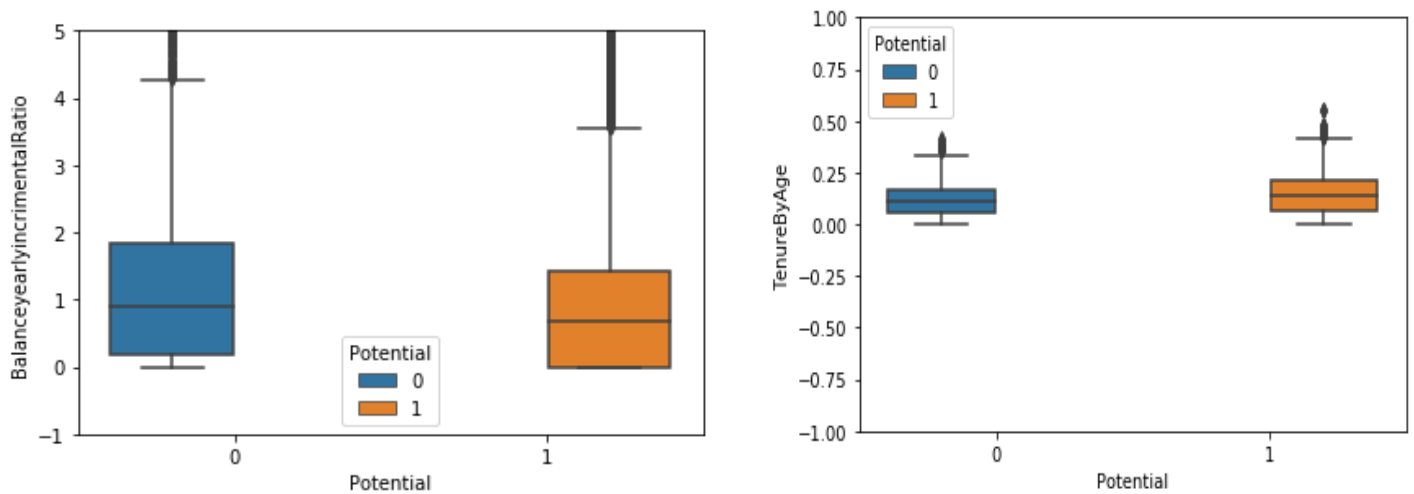


Figure 1:- Training result of balance and incremental ratio and Age and tenure

However as seen above after training balance and yearly incremental together, ensure and age there is a variance. The ratio of the bank balance and the yearly incremental indicates that customers with a higher balance yearly incremental ratio are more potential which would be a worrying advantage to the bank as this impacts their source of loan capital. Regarding the tenure and age since tenure is a function of age, we introduce a variable aiming to standardize tenure over age hence slightly contribute to potential customer.

Data Preparation for model fitting - To do the following tasks the data preparation for test data shall be prepared. The major tasks are predicting the added new features, Reorder the columns, change the 0 in categorical variables to -1 and encode the categorical variable, minimum maximum scaling and ensuring that the variables are ordered in the same way as per the desired.

Here the main task to Arrange columns by data type for easier manipulation by classifying as continuous variables (continuous_vars), Category Variables (cat_vars) to train the potential trained data and the sum of Continuous and category variables. The output displayed as the following table using the following code.

The other remaining task in data preparation for model fitting is to change from 0 to -1 for the hot variables in order to make the models to capture a negative relation. For Has D Card, Is Active Member, Cty and Gender instances to be trained, Ensuring that all one variables that appear in the train data appear in the subsequent data. Min Max scaling continuous variables based on minimum and maximum from the training data and to ensure the variables are ordered in the same way as was ordered in the training set. The last step for data preparation for model fitting is data preparation pipeline for test data which has the following task while preparing pipeline

```
# data prep pipeline for test data def DfPrepPipeline(df_predict, df_train_Cols, minVec, maxVec):  
# Add new features predicting Balanceyearly incomeRatio, TenureBy Age , Noof TranG ven Age  
# Reorder the columns( both continuous_vars and cat_vars)  
# Change the 0 in categorical variables to -1(for Has D Card and Is ActiveMember  
# One hot encode the categorical variables(cty and gender)  
# Ensure that all one hot encoded variables that appear in the train data appear in the subsequent  
data  
# Min Max scaling continuous variables  
# Ensure that The variables are ordered in the same way as was ordered in the train set  
df_predict = df_predict[df_train_Cols]  
return df_predict
```

(Detail code available at the end of the study in annex.

Model fitting and selection: -For model fitting we had implemented LR in the PS and with different kernels, SVM in the primal and with different kernels and EM. To check each model fitting practically we have imported support functions, fit models and scoring functions.

Table 4: python support, fit model and scoring functions

# Support functions	<pre> from sklearn.preprocessing import PolynomialFeatures from sklearn.model_selection import cross_val_score from sklearn.model_selection import GridSearchCV from scipy.stats import uniform </pre>
# Fit models.	<pre> from sklearn.linear_model import LogisticRegression from sklearn.svm import SVC from sklearn.ensemble import RandomForestClassifier from xgboost import XGBClassifier </pre>
# Scoring functions	<pre> from sklearn.metrics import accuracy_score sum of true.. positives and false from sklearn.metrics import classification_report from sklearn.metrics import roc_auc_score. from sklearn.metrics import roc_curve </pre>

Once after importing the above support functions, fit models and scoring functions. Training have been done on the potential data using various features. Primal Logistic regression trained using lbfgs solver where as liblinear solver for logistic regression degree 2. SVM with RBF kernel and poly kernel trained. Random forest classifier fit trained and extreme gradient boosting classifier trained. For each fit training the accuracy displayed including the detail feature results.

After training the fit model the next task is applying fit best model. For instance the fit and best fit model codes displayed as follows including the result respectively

```

# Fit primal logistic regression training
para_m_grid= {' C': [0.1, 0.5, 1, 10, 50, 100], ' max_iter': [250], ' fit_intercept': [ True], ' intercept_scaling': [1],
'penalty': [l2], 'td': [0.00001, 0.0001, 0.000001] }
log_primal_Grid=GridSearchCV(LogisticRegression(solver='lbfgs'), para_m_grid, cv=10, refit=True, verbose=0)
log_primal_Grid.fit(df_trainloc[:, df_traincolumns!='potential'], df_trainPotential)
best_model(log_primal_Grid)
output
0.810375
{' C': 0.1, ' fit_intercept': True, ' intercept_scaling': 1, ' max_iter': 250, ' penalty': 'l2', ' td': 1e-05}
LogisticRegression(C=0.1, class_weight=None, dual=False, fit_intercept=True, intercept_scaling=1, max_iter=250, multi_class='ovr', n_jobs=1, penalty='l2', random_state=None, solver='lbfgs', td=1e-05, verbose=0, warm_start=False)

```

```

# Fit primal logistic regression
log_primal = LogisticRegression(C=100, class_weight=None, dual=False, fit_intercept=True, intercept_scaling=1, max_iter=250, multi_class='warn', n_jobs=None, penalty='l2', random_state=None, solver='lbfgs', td=1e-05, verbose=0, warm_start=False)
log_primal.fit(df_trainloc[:, df_traincolumns!='potential'], df_trainpotential)
Output:
LogisticRegression(C=100, class_weight=None, dual=False, fit_intercept=True, intercept_scaling=1, max_iter=250, multi_class='warn', n_jobs=None, penalty='l2', random_state=None, solver='lbfgs', td=1e-05, verbose=0, warm_start=False)

```

Best model fit status

Fit model status with each six classifiers status described in the following table.

Classifier	Fit Best_Model status
best_model(log_primal_grid)	0.8151
best_model(log_pol2_grid)	0.8556
best_model(SVM_grid_RBF)	0.8519
best_model(SVM_grid_pol)	0.8545
best_model(RanFor_grid)	0.8631
best_model(XGB_grid)	0.8633

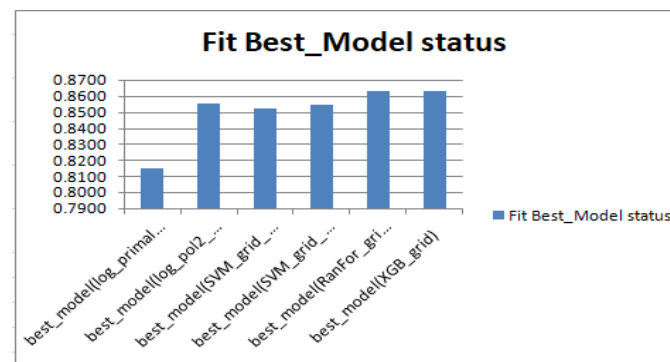
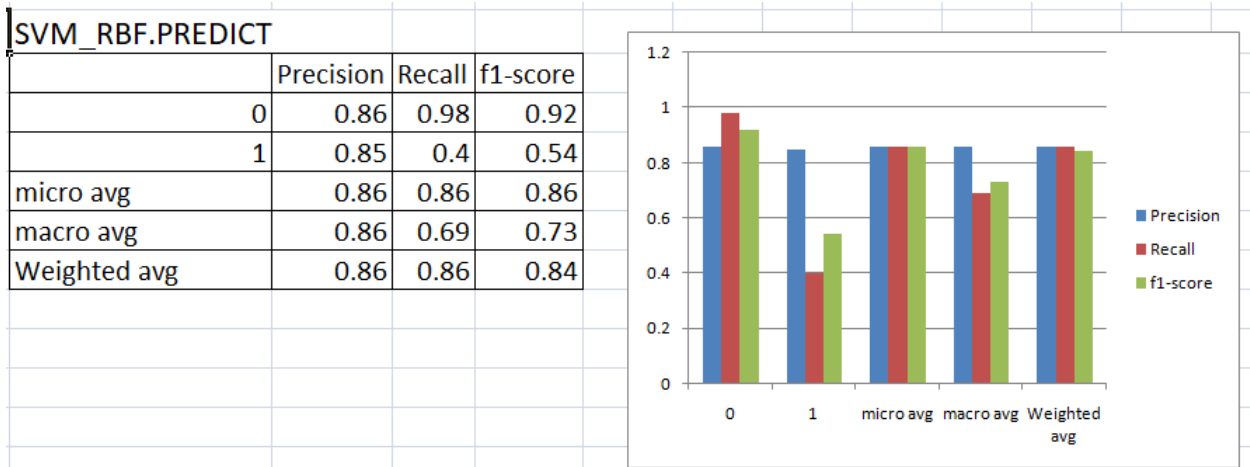
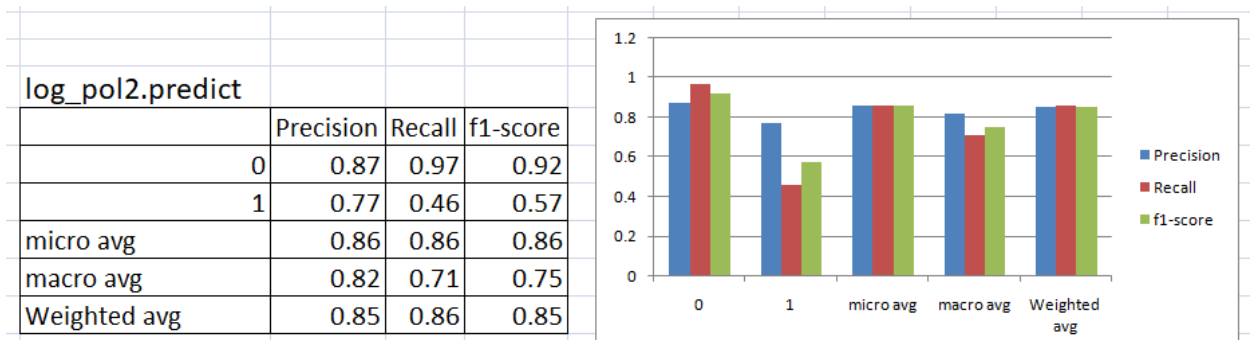
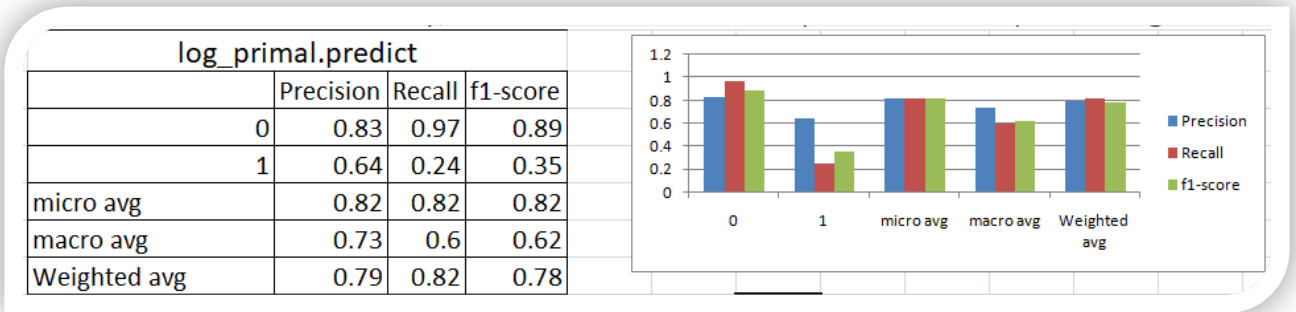


Figure 2 Best model fit status

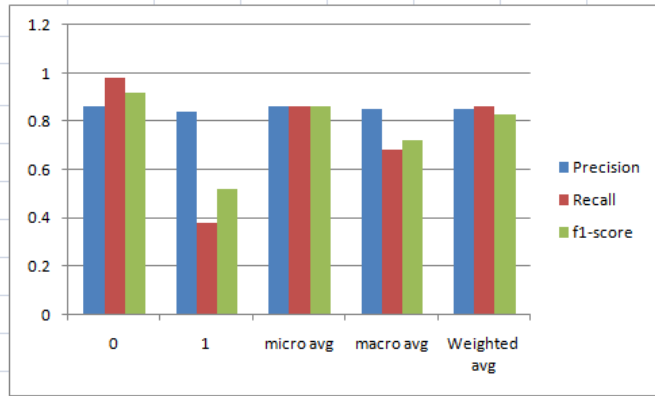
As described in the annex part fit and best fit training codes have been implemented for SVM LR RF and XGB classifiers and their respected accuracy result displayed

The final step is to Review best model fit accuracy: here our interest is on the performance in predicting the potential customer who is highly important.



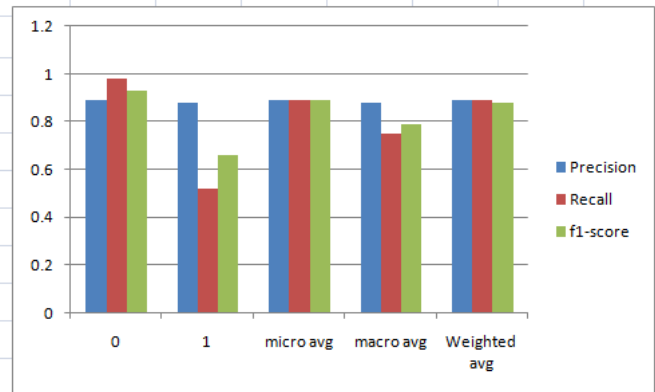
SVM_POL.PREDICT

	Precision	Recall	f1-score
0	0.86	0.98	0.92
1	0.84	0.38	0.52
micro avg	0.86	0.86	0.86
macro avg	0.85	0.68	0.72
Weighted avg	0.85	0.86	0.83



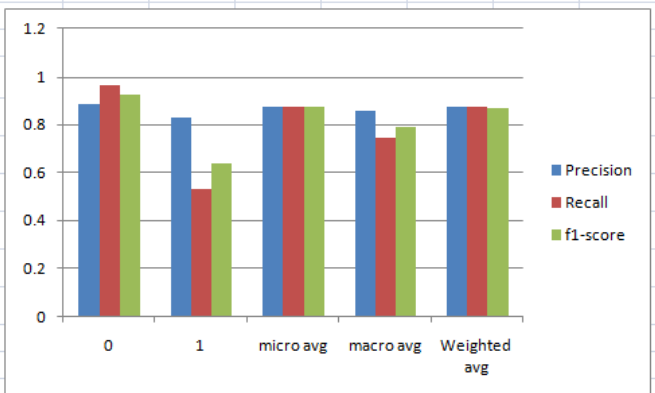
RF.predict

	Precision	Recall	f1-score
0	0.89	0.98	0.93
1	0.88	0.52	0.66
micro avg	0.89	0.89	0.89
macro avg	0.88	0.75	0.79
Weighted avg	0.89	0.89	0.88



XGB.predict

	Precision	Recall	f1-score
0	0.89	0.97	0.93
1	0.83	0.53	0.64
micro avg	0.88	0.88	0.88
macro avg	0.86	0.75	0.79
Weighted avg	0.88	0.88	0.87



All Weighted average status

	Precision	Recall	f1-score
log_primal.predict	0.79	0.82	0.78
log_pol2.predict	0.85	0.86	0.85
SVM_RBF.PREDICT	0.86	0.86	0.84
SVM_POL.PREDICT	0.85	0.86	0.83
RF.predict	0.89	0.89	0.88
XGB.predict	0.88	0.88	0.87

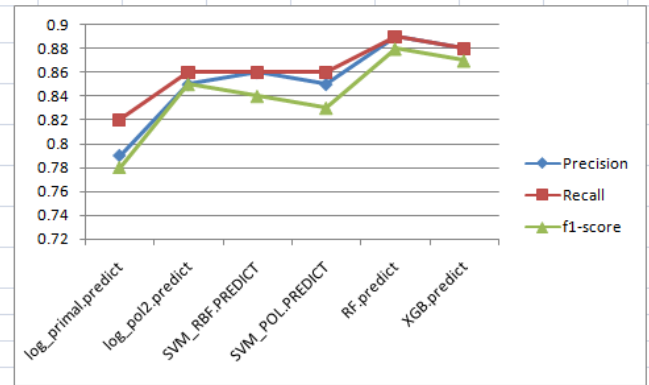


Figure 3 Comparison of each prediction algorithms and their precision, recall and F1score result.

As we can see in the above discussion Random forest best predicted than XGB and other predictions as well logistic regression primal products with low evaluation metric result where as logistic regression with degree 2 prediction is better.

```
Log_primal.fit(df_trainloc[:,df_train columns!= 'Potential'],df_train Potential)
log_pol2.fit(df_train_pol2,df_train Potential)
SVM_RBF.fit(df_trainloc[:,df_train columns!= 'Potential'],df_train Potential)
SVM_POL.fit(df_trainloc[:,df_train columns!= 'Potential'],df_train Potential)
RF.fit(df_trainloc[:,df_train columns!= 'Potential'],df_train Potential)
XGB.fit(df_trainloc[:,df_train columns!= 'Potential'],df_train Potential)
```

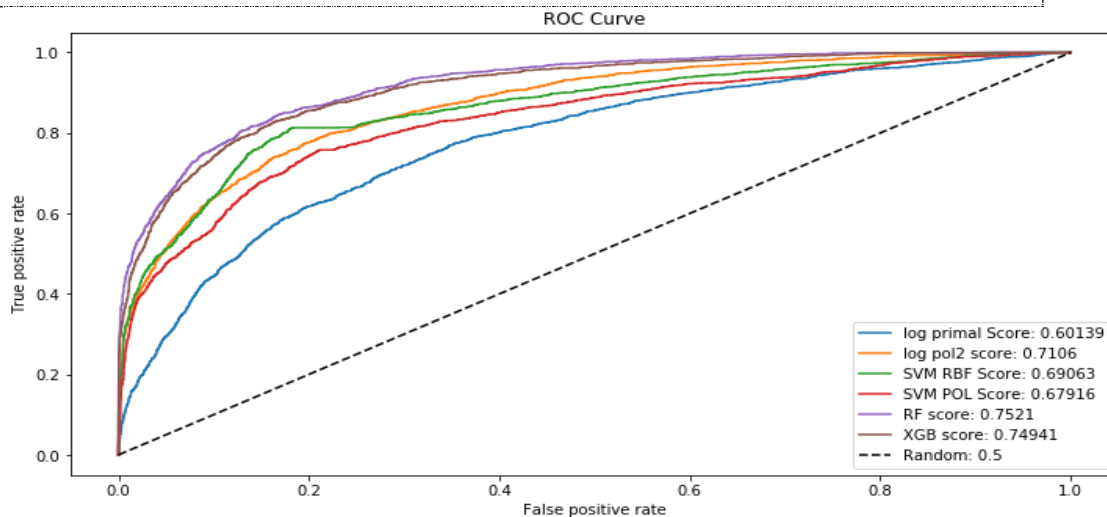


Figure 4:- Training ROC curve

The proportion of potential customer status result:- As we have seen from figure 3 the proportion of potential customer status about the 80% of the total customer is potential. So the baseline model could be to predict that 80% of the customers are potential. We need to ensure that the chosen model does predict with great accuracy this 80% as it is of interest to the bank to identify and keep this bunch as opposed to accurately predicting the customers that are kept in touch with the bank by granting the appropriate benefits packages.

Status relation of a potential customer with categorical variables result in the description

In addition to the above from figure 4, the following points have been noted. As we can see from the graph that visualizes the status relation of a potential customer with categorical variables. The 1st result observation is that the Majority of the data is from persons from Bahirdar. However, the proportion of potential customers is inversely related to the population of customers alluding to the bank possibly having a problem for Debre markos city customers (maybe not enough customer service resources allocated) in the areas where it has fewer clients. The 2nd result observation is the proportion of potential female customers is also greater than that of male customers. The 3rd interesting result observation is, the majority of the potential customers are those with credit cards. Given that the majority of the customers have credit cards could prove this to be just a coincidence. The 4th Unsurprising result is that the inactive members have a greater potential tendency and worryingly is that the overall proportion of inactive members is quite high suggesting that the bank may need a program implemented to turn this group to active customers as this will definitely have a positive impact on the potential customer.

Status relation of a potential customer with continuous variables result in description

The status relation of a potential customer with continuous variables described in Figure 5. From the image the noted results are the followings: There is no significant difference in the credit score distribution between potential and not a potential customer. The older customers are potential at more than the younger ones alluding to a difference in service preference in the age categories. The bank may need to review its target market or review the strategy for retention between the different age groups. With regard to the tenure, the clients on either extreme end (spent little time with the bank or a lot of time with the bank) are more likely potential customer

compared to those that are of average tenure. Worryingly, the bank is losing customers with the significant bank balance which is likely to hit their available capital for lending. Neither the product nor the salary has a significant effect on the likelihood to exit.

Data Preparation for model fitting

For best model fitting the following experiments have been handled, those are Arranging columns based on the data types for the sake of easier manipulation, and to make the model able to understand the negative relations for hot variables has a debit card, Is active member? city and gender training. At last the trained data frame scoring result described from table 8 to table 13.

4.2.1 The experimental result using Logistic Regression (LR)

Logistic regression which is a variation of ordinary regression that is used when the dependent (response) variable is dichotomous (takes two values). In logistic regression, a binary logistic model is used to estimate the probability of a binary response based on one or more predictor or independent variables. LR may use any of the following parameter's 'lbfgs', 'liblinear', 'sag', 'saga' and the optional (default = 'liblinear'). For small datasets, 'liblinear' is a good choice, whereas 'sag' and 'saga' are faster for large ones. Stochastic Average Gradient (SAG) method optimizes the sum of a finite number of smooth convex functions. Like the stochastic gradient (SG) methods, the SAG method's iteration cost is independent of the number of terms in the sum. However, by incorporating a memory of previous gradient values the SAG method achieves a faster convergence rate than black-box SG methods.

It is faster than other solvers for large datasets when both the number of samples and the number of features are large.

4.2.1.1 Logistic regression using primal prediction

	Precision	Recall	F1-score	Support
Potential(1)	0.83	0.97	0.89	6353
Not potential(0)	0.64	0.24	0.35	1647

Table 7:- Accuracy Results for logistic regression using primal prediction

4.2.1.2 Logistic regression with poly 2 prediction

	Precision	Recall	F1-score	Support
Potential(1)	0.87	0.97	0.92	6353
Not potential(0)	0.77	0.46	0.57	1647

Table 8:- Accuracy Results for logistic regression using poly 2 prediction

4.2.2 Support Vector Machine (SVM) Experimental result

The Support vector machine (SVM) is a supervised learning method that creates input-output mapping capacities from a lot of marked preparing information. **Kernel SVM** - It is a high performance on non-direct problems and not influenced by outliers, not sensitive to overfitting is prone of this algorithm and not the best preference for a large number of features.

SVM algorithms use a set of mathematical functions that are defined as the kernel. The function of the kernel is to take data as input and transform it into the required form. Some of the common kernels used with SVMs and their uses: Polynomial kernel (It is popular in image processing), Gaussian kernel (It is a general-purpose kernel; used when there is no prior knowledge about the data), Gaussian radial basis function (RBF) (It is a general-purpose kernel; used when there is no prior knowledge about the data), Laplace RBF kernel (It is a general-purpose kernel).

4.2.2.1 SVM with RBF kernel

Radial basis function kernel (**RBF kernel**) is a popular kernel function used in various kernelized learning algorithms. In particular, it is commonly used in support vector machine classification. The **gamma** parameter is the inverse of the standard deviation of the **RBF kernel** (Gaussian function), which is used as a similarity measure between two points. Intuitively, a small **gamma** value defines a Gaussian function with a large variance.

	Precision	Recall	F1-score	Support
Potential(1)	0.86	0.98	0.92	6353
Not potential(0)	0.85	0.40	0.54	1647

Table 9:- Accuracy Results for SVM RBF kernel

4.2.2.2 SVM with Hov kernel

Support vector machines (SVM) are a set of supervised learning methods used for classification, regression and outlier's detection. The advantages of support vector machines are effective in high dimensional spaces. Still effective in cases where a number of dimensions is greater than the number of samples.

	Precision	Recall	F1-score	Support
Potential(1)	0.86	0.98	0.92	6353
Not potential(0)	0.84	0.38	0.52	1647

Table 10:- Accuracy Results for SVM Poly kernel

4.2.3 The experimental result using Random Forest Classifier

	Precision	Recall	F1-score	Support
Potential(1)	0.89	0.98	0.93	6353
Not potential(0)	0.88	0.52	0.66	1647

Table 11 : Accuracy Results Random Forest Classifier

A random forest is a Meta estimator that fits various decision tree classifier on different sub-samples of the dataset and utilizes averaging to improve the predictive accuracy and command over-fitting. The train() class method builds this tree from the beginning, beginning with the leaf nodes. It at that point refines itself to minimize the number of choices expected to get with a name by putting the most enlightening features at the top.

4.2.4 Experimental result using XGB Classifier

XGBoost is an algorithm that has recently been dominating applied machine learning and Kaggle competitions for structured or tabular data. XGBoost is an implementation of gradient boosted decision trees designed for speed and performance.

XGBoost has a very useful function called ‘cv’ which performs cross-validation at each boosting iteration and thus returns the optimum number of trees required. Tune tree-specific parameters (max_depth, min_child_weight, gamma, subsample, colsample_bytree) for decided learning rate and a number of trees. XGBoost has been lauded as the holy grail of machine learning hackathons and competitions. From predicting ad click-through rates to classifying high energy physics events, XGBoost has proved its mettle in terms of performance - and speed.

XGBoost is an ensemble learning method. Ensemble learning offers a systematic solution to combine the predictive power of multiple learners. The resultant is a single model which gives the aggregated output from several models. The models that form the ensemble, also known as base learners, could be either from the same learning algorithm or different learning algorithms. Bagging and boosting are two widely used ensemble learners.

	Precision	Recall	F1-score	Support
Potential(1)	0.89	0.97	0.93	6353
Not potential(0)	0.83	0.53	0.64	1647

Table 12:- Accuracy Results for XGB Classifier

4.3 DISCUSSION OF THE RESULT

The training result of the six models graphically visualized in the following figure 4. The accuracy of extreme gradient boosting classifier scores 74.94% and Random forest with (75.21%) finally the accuracy of Logistic Regression both (primal and polynomial) 60.14% and 71.06%, SVM (Poly and RBF) 69.06% and 68%. From the six model, the best in Recall is a random forest.

Figure 5: - Model comparison for all used models

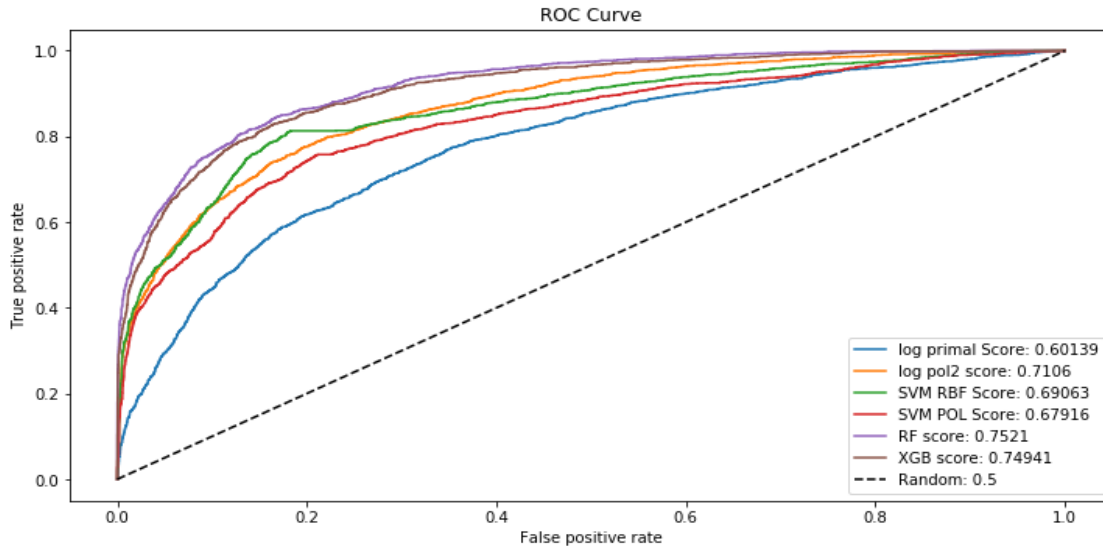


Figure 6- ROC of all models in the training

From the above results, our main aim is to predict potential customers which used for best service and package providing method as per the importance of the customer for the business hence the predicted potential customer put into some sort of scheme in order to provide better service and packages to do so the recall measure on the 1's is of more importance to the study than the overall accuracy score of the model.

From the review of the fitted models above, the best model that gives a decent balance of the recall and precision is the random forest where according to the fit on the training set, with a precision score on 1's of 0.89, out of all customers that the model thinks are potential, 89% do actually potential and with the recall score of 0.98 on the 1's, the model is able to highlight 98% of all those who are potential.

4.1 Test model prediction accuracy on test data

Once after training each model using training data and after receiving each model predictions the next remaining part is to test each models using test data and compare with previous results. the training data has 1996 rows and 17 columns.

RF. predict.test

	Precision	Recall	f1-score
0	0.87	0.98	0.92
1	0.79	0.38	0.51
mi cro avg	0.86	0.86	0.86
macro avg	0.83	0.68	0.72
Wei ghted avg	0.85	0.86	0.84

Table 13:- model prediction Accuracy result.

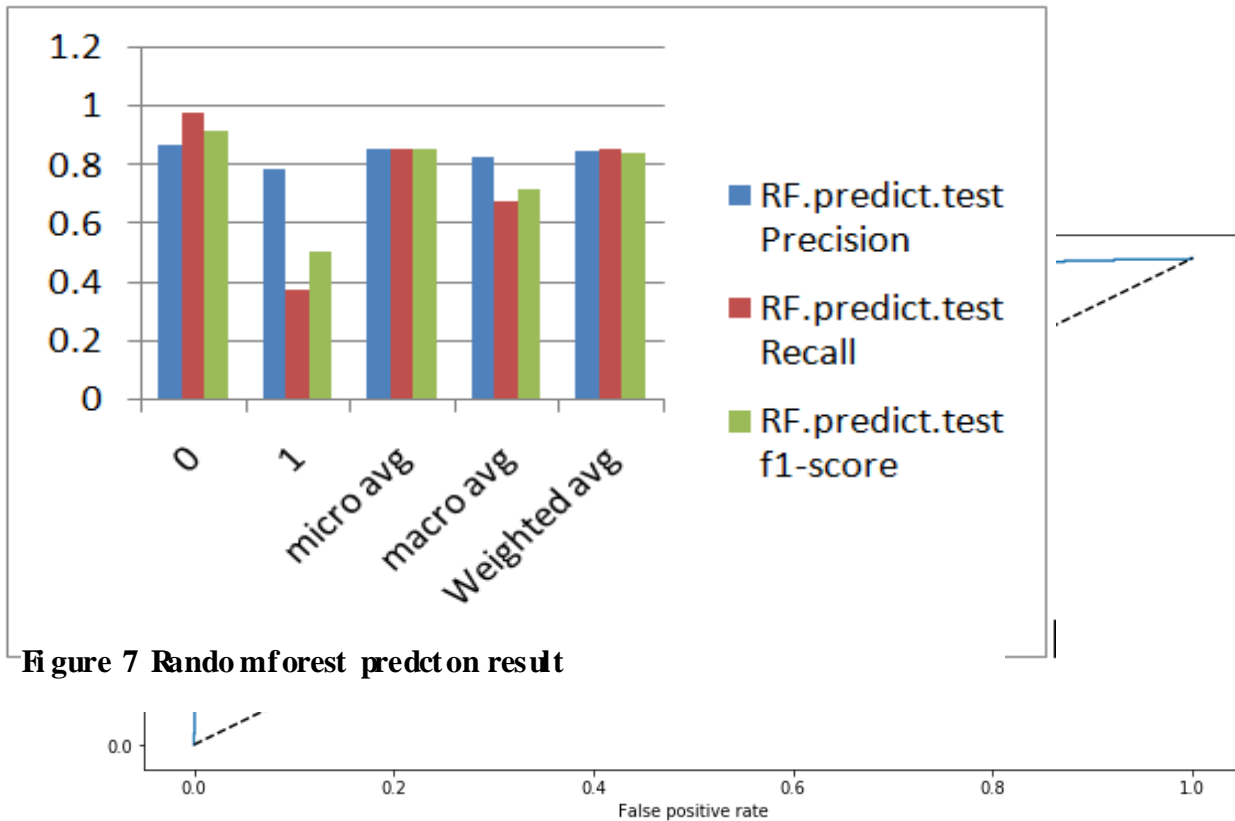


Figure 7 Randomforest predict on result

Figure 8:- ROC curve of Randomforest

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 CONCLUSION

The precision of the model test data is slightly higher with regard to precision those customers who subscribe to term deposit. However, in as much as the model has high accuracy, it still misses about half of those who are term deposited. This could be improved by providing retraining the model with more data over time while in the meantime working with the model to save the that would be potential customer.

In this Research, we compared the six models LR Primal, LR Poly2, SVM with RBF, SVM with Poly, RF, and XGB classifiers and observed ROC curve, XGB with RF Score shown tremendous result. But if you consider other features label like precision, recall, F1 Score the models shown the nearest to each other as we can see in the fig 11 ROC value of XGB is more better than with RF. Hence we can conclude that XGB ensemble classifier classifies and predict better than RF and others too. the final RF ROC result is 67.67% but XGB scores 79.94% which is a better result than the state of art classification and prediction algorithm

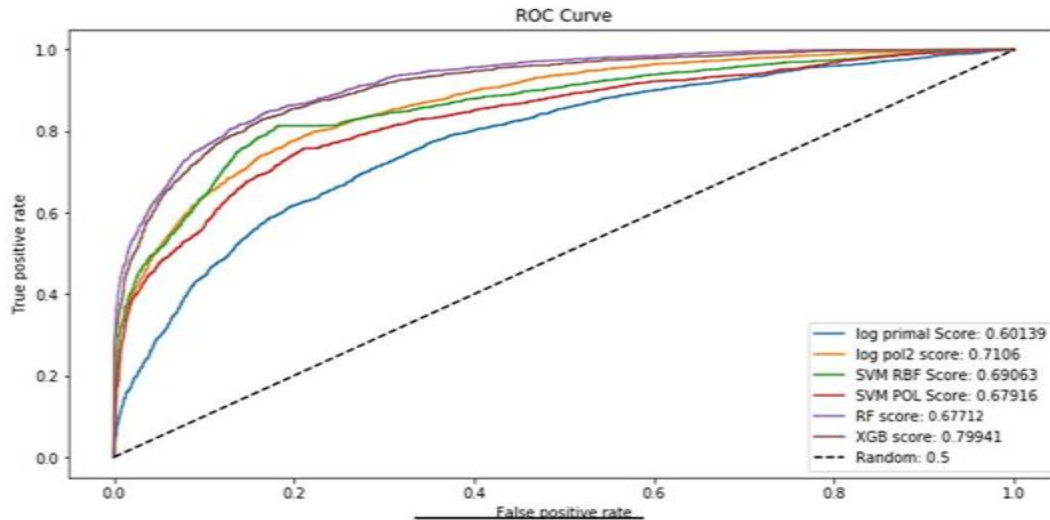


Figure 9: Final ROC curve in test data

5.2 RECOMMENDATION & FUTURE WORKS

Future research shall focus on using different ensemble methods, on the product types rather than on the number of products, on the number of transaction either credit or debit transaction impact on classification. Our impression is that mainly on the customer classification based on the target variable potential and non-potential here the prediction of customers who will leave the bank shall be predicted.

REFERENCE

- [1] J. Ahilman, M M Rian, W Marina, and K Margono, ‘Predicting and clustering customer to improve customer loyalty and company profit,’ *2014 2nd Int. Conf. Inf. Commun. Technol. ICoICT 2014*, pp. 331-334, 2014.
- [2] D R S R Manikandan, ‘Machine Learning Algorithms for Classification,’ *Int. J. Acad. Res. Dev.*, no. 2018, pp. 384-389, 2018.
- [3] J. Asare-Frenpong and M Jayabalan, ‘Predicting customer response to bank direct telemarketing campaign,’ *2017 Int. Conf. Eng. Technol. Technopreneurship, ICE2T 2017*, vol. 2017-Janua, pp. 1-4, 2017.
- [4] Z. Zhi-Hua, ‘Ensemble Methods: Foundations and Algorithms,’ *Chapman and Hall/CRC*, p. 23.
- [5] Y. Jiang, ‘Using Logistic Regression Model to Predict the Success of Bank Telemarketing,’ *Int. J. Data Sci. Technol.*, vol. 4, no. 1, pp. 35-41, 2018.
- [6] X. Hu, ‘A data mining approach for retailing bank customer attrition analysis,’ *Appl. Intell.*, vol. 22, no. 1, pp. 47-60, 2005.
- [7] O. Apampa, ‘Evaluation of Classification and Ensemble Algorithms for Bank Customer Marketing Response Prediction,’ *J. Int. Technol. Inf. Manag.*, vol. 25, no. 4, pp. 85-100, 2016.
- [8] C. W. Wang, ‘New ensemble machine learning method for classification and prediction on gene expression data,’ *Annu. Int. Conf. IEEE Eng. Med. Biol. - Proc.*, pp. 3478-3481, 2006.
- [9] (<https://www.python.org/about/>), ‘(<https://www.python.org/about/>)’.

ANNEXERS

```
# Read the data frame
df = pd.read_csv('I:\Bank_data\Bank_potential1.csv', delimiter=',')
df.shape

# Check columns list and missing values
df.isnull().sum()

# Get unique count for each variable
df.nunique()

# Proportion of Potential customer and non potential customer
labels = 'Potential', 'Not potential'
sizes = [df.Potential[df['Potential']==1].count(), df.Potential[df['Potential']==0].count()]
explode = (0, 0.1) #used to explode slice from the circle
fig1, ax1 = plt.subplots(figsize=(10, 8))
ax1.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=90)
ax1.axis('equal')
plt.title("Proportion of Potential customer and a customer with -ve response", size = 20)
plt.show()

#Python code which applied to train and test:
# Split Train test data
df_train = df.sample(frac=0.8, random_state=200)
```

```
df_test = df.drop(df_train.index)
print(len(df_train))
print(len(df_test))
```

```
# Arrange columns by data type for easier manipulation
continuous_vars = ['Nooftransacti on', 'Age', 'Tenure', 'Bal ance', 'NumOf Products', 'yearly ncom
e', 'Bal anceyearly ncomeati on', 'Tenure By Age', 'Nooftransacti on G ven Age']
cat_vars = ['Has D Card', 'Is Active M ember', 'Gty', 'Gender']
df_train = df_train[['Potenti d'] + continuous_vars + cat_vars]
df_train head()
```

```
# data prep pi peli ne for test data
```

```
def Df Prep Pi peli ne(df_predi ct, df_train_Cols, mi n Vec, max Vec):
```

```
# Add ne w feat ures
```

```
df_predi ct[' Bal anceyearly ncome Rati o'] =
df_predi ct. Bal ance/ df_predi ct. Bal anceyearly ncome Rati o
```

```
df_predi ct[' Tenure By Age'] = df_predi ct. Tenure/(df_predi ct. Age - 18)
```

```
df_predi ct[' Noof Tr ansacti on G ven Age'] = df_predi ct. Nooftransacti on/(df_predi ct. Age - 18)
```

```
# Reor der the col umns
```

```
conti nuous_vars =
```

```
[' Nooftransacti on', 'Age', 'Tenure', 'Bal ance', 'Nu mOf Products', 'yearly ncri ment al', 'Bal anceyearly nco
me Rati o',
```

```
    ' Tenure By Age', ' Noof Tr ansacti on G ven Age']
```

```
cat_vars = [' Has D Card', 'Is Acti ve M ember', "Gty", " Gender"]
```

```
df_predi ct = df_predi ct[['Pot enti al'] + conti nuous_vars + cat_vars]
```

```
# Change the 0 i n categori cal vari ables to -1
```

```
df_predi ct.loc[ df_predi ct. Has D Card == 0, ' Has D Card'] = -1
```

```
df_predi ct.loc[ df_predi ct. Is Acti ve M ember == 0, ' Is Acti ve M ember'] = -1
```

```
# One hot encode the categori cal vari ables
```

```
lst = ["Gty", " Gender"]
```

```

remove = list()

for i in lst:
    for j in df_predict[i].unique():
df_predict[i+'_'+j] = np.where(df_predict[i] == j, 1, -1)
remove.append(i)

df_predict = df_predict.drop(remove, axis=1)

# Ensure that all one hot encoded variables that appear in the train data appear in the
subsequent data

L = list(set(df_train_Cols) - set(df_predict.columns))

for l in L:
df_predict[str(l)] = -1

# Min Max scaling continuous variables based on min and max from the train data
df_predict[continuous_vars] = (df_predict[continuous_vars] - min_vec) / (max_vec - min_vec)

# Ensure that The variables are ordered in the same way as was ordered in the train set
df_predict = df_predict[df_train_Cols]

return df_predict

```

#Function of to give best model score and parameter

```

def best_model(model):
    print(model.best_score_)
    print(model.best_params_)
    print(model.best_estimator_)

def get_auc_scores(y_actual, method, method2):
    auc_score = roc_auc_score(y_actual, method);
    fpr_df, tpr_df, _ = roc_curve(y_actual, method2);
    return (auc_score, fpr_df, tpr_df)

```

```
# Fit primal logistic regression
param_grid = {'C': [0.1, 0.5, 1, 10, 50, 100], 'max_iter': [250], 'fit_intercept': [True], 'intercept_scaling': [1],
              'penalty': ['l2'], 'td': [0.00001, 0.0001, 0.000001]}
log_primal_Grid = GridSearchCV(LogisticRegression(solver='lbfgs'), param_grid, cv=10, refit=True, verbose=0)
log_primal_Grid.fit(df_train.loc[:, df_train.columns != 'potential'], df_train.Potential)
best_model(log_primal_Grid)
```

Output

```
0.810375
{'C': 0.1, 'fit_intercept': True, 'intercept_scaling': 1, 'max_iter': 250, 'penalty': 'l2', 'td': 1e-05}
LogisticRegression(C=0.1, class_weight=None, dual=False, fit_intercept=True,
intercept_scaling=1, max_iter=250, multi_class='ovr', n_jobs=1,
penalty=l2, random_state=None, solver='lbfgs', td=1e-05,
verbose=0, warm_start=False)
```

```
# Fit logistic regression with degree 2 polynomial kernel
param_grid = {'C': [0.1, 10, 50], 'max_iter': [300, 500], 'fit_intercept': [True], 'intercept_scaling': [1], 'penalty': ['l2'],
              'td': [0.0001, 0.000001]}
poly2 = PolynomialFeatures(degree=2)
df_train_pol2 = poly2.fit_transform(df_train.loc[:, df_train.columns != 'Exited'])
log_pol2_Grid = GridSearchCV(LogisticRegression(solver='liblinear'), param_grid, cv=5, refit=True, verbose=0)
log_pol2_Grid.fit(df_train_pol2, df_train.Potential)
best_model(log_pol2_Grid)
```

Output

```
0.995125
{'C': 50, 'fit_intercept': True, 'intercept_scaling': 1, 'max_iter': 300, 'penalty': 'l2', 'td': 1e-06}
LogisticRegression(C=50, class_weight=None, dual=False, fit_intercept=True,
intercept_scaling=1, max_iter=300, multi_class='ovr', n_jobs=1,
penalty=l2, random_state=None, solver='liblinear', td=1e-06,
verbose=0, warm_start=False)
```

```
# Fit SVM with RBF Kernel
```

```
param_grid = {'C': [0.5, 100, 150], 'gamma': [0.1, 0.01, 0.001], 'probability': [True], 'kernel': ['rbf']}
SVM_Grid = GridSearchCV(SVC(), param_grid, cv=3, refit=True, verbose=0)
```

```
SVM_grid_fit(df_trainloc[:, df_train_columns != 'Potential'], df_train_Potential)
best_model(SVM_grid)
```

Output

```
0.798375
{'C': 0.5, 'gamma': 0.1, 'kernel': 'rbf', 'probability': True}
SVC(C=0.5, cache_size=200, class_weight=None, coef0=0.0,
decision_function_shape='ovr', degree=3, gamma=0.1, kernel='rbf',
max_iter=1, probability=True, random_state=None, shrinking=True,
tol=0.001, verbose=False)
```

```
# Fit SVM with pol kernel
param_grid = {'C': [0.5, 1, 10, 50, 100], 'gamma': [0.1, 0.01, 0.001], 'kernel':
['poly'], 'degree': [2, 3]}
SVM_grid = GridSearchCV(SVC(), param_grid, cv=3, refit=True, verbose=0)
SVM_grid_fit(df_trainloc[:, df_train_columns != 'Potential'], df_train_Potential)
best_model(SVM_grid)
```

Output

```
0.8545
{'C': 100, 'degree': 2, 'gamma': 0.1, 'kernel': 'poly', 'probability': True}
SVC(C=100, cache_size=200, class_weight=None, coef0=0.0,
decision_function_shape='ovr', degree=2, gamma=0.1, kernel='poly',
max_iter=1, probability=True, random_state=None, shrinking=True,
tol=0.001, verbose=False)
```

```
# Fit randomforest classifier
param_grid = {'max_depth': [3, 5, 6, 7, 8], 'max_features':
[2, 4, 6, 7, 8, 9], 'n_estimators': [50, 100], 'min_samples_split': [3, 5, 6, 7]}
```

```
RanFor_grid = GridSearchCV(RandomForestClassifier(), param_grid, cv=5, refit=True,
verbose=0)
RanFor_grid.fit(df_trainloc[:, df_traincolumns != 'Potential'], df_trainpotential)
best_model(RanFor_grid)
```

Output

```
0.863125
{'max_depth': 8, 'max_features': 9, 'min_samples_split': 6, 'n_estimators': 50}
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
max_depth=8, max_features=9, max_leaf_nodes=None,
min_impurity_decrease=0.0, min_impurity_split=None,
min_samples_leaf=1, min_samples_split=6,
min_weight_fraction_leaf=0.0, n_estimators=50, n_jobs=None,
oob_score=False, random_state=None, verbose=0,
warm_start=False)
```

```
# Fit Extreme Gradient boosting classifier
param_grid={'max_depth':[5, 6, 7, 8], 'gamma':[0.01, 0.001, 0.001], 'min_child_weight':[1, 5, 10], 'learning_rate':[0.05, 0.1, 0.2, 0.3], 'n_estimators':[5, 10, 20, 100]}
xgb_grid=GridSearchCV(XGBClassifier(), param_grid, cv=5, refit=True, verbose=0)
xgb_grid.fit(df_trainloc[:, df_traincolumns != 'potential'], df_trainPotential)
best_model(xgb_grid)
```

Output

```
0.86325
{'gamma': 0.01, 'learning_rate': 0.1, 'max_depth': 7, 'min_child_weight': 5, 'n_estimators': 20}
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
colsample_bytree=1, gamma=0.01, learning_rate=0.1, max_delta_step=0,
max_depth=7, min_child_weight=5, missing=None, n_estimators=20,
n_jobs=1, nthread=None, objective='binary:logistic', random_state=0,
reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
silent=True, subsample=1)
```

```
In [36]: # Fit SVM with RBF Kernel
SVM_RBF = SVC(C=100, cache_size=200, class_weight=None, coef0=0.0, decision_function_shape='ovr', degree=3, gamma
          random_state=None, shrinking=True, tol=0.001, verbose=False)
SVM_RBF.fit(df_train.loc[:, df_train.columns != 'Potential'],df_train.Potential)

Out[36]: SVC(C=100, cache_size=200, class_weight=None, coef0=0.0,
            decision_function_shape='ovr', degree=3, gamma=0.1, kernel='rbf',
            max_iter=-1, probability=True, random_state=None, shrinking=True,
            tol=0.001, verbose=False)
```

```
In [54]: # Fit Random Forest classifier
RF = RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',max_depth=8, max_features=6, max
          min_impurity_split=None,min_samples_leaf=1, min_samples_split=3,min_weight_fraction_l
          oob_score=False, random_state=None, verbose=0,warm_start=False)
RF.fit(df_train.loc[:, df_train.columns != 'Potential'],df_train.Potential)

Out[54]: RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                                max_depth=8, max_features=6, max_leaf_nodes=None,
                                min_impurity_decrease=0.0, min_impurity_split=None,
                                min_samples_leaf=1, min_samples_split=3,
                                min_weight_fraction_leaf=0.0, n_estimators=50, n_jobs=None,
                                oob_score=False, random_state=None, verbose=0,
                                warm_start=False)
```

Fit best Models trained with each respective parameters for all six models

LogisticRegression.fit(df_train.loc[:, df_train.columns != 'Potential'],df_train.Potential)
 LogisticRegression.fit(df_train.pol2,df_train.Potential)
 SVM_RBF.fit(df_train.loc[:, df_train.columns != 'Potential'],df_train.Potential)
 SVM_POL.fit(df_train.loc[:, df_train.columns != 'Potential'],df_train.Potential)
 RF.fit(df_train.loc[:, df_train.columns != 'Potential'],df_train.Potential)
 XGB.fit(df_train.loc[:, df_train.columns != 'Potential'],df_train.Potential)

Server Information:

You are using Jupyter notebook

The version of the notebook server is: **5.5.0**

The server is running on this version of Python:

```
Python 3.6.5 | Anaconda, Inc. | (default, Mar 29 2018, 13:32:41) [MSC v.1900 64 bit (AMD64)]
```

Current Kernel Information:

```
Python 3.6.5 | Anaconda, Inc. | (default, Mar 29 2018, 13:32:41) [MSC v.1900 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 6.4.0 -- An enhanced Interactive Python. Type '?' for help
```

