CMPE 239: Final Project Report

Telemarketing and Machine Learning – A Performance Analysis

Submitted by: Team

Apoorva Gupta [010001043]

Rakesh Balusa [010020361]

Moksha Bhargav Vanam [010011638]

Submitted to: Dr. Magdalini Eirinaki

Date of Submission: November 30, 2015.

Table of Contents

CHAPTER 1. INTRODUCTION

1.1 Motivation

The trend of telemarketing is becoming a huge part of direct marketing where the users are offered services or products over the phone call. This technique is quite interactive and is adopted by many banks to present long term deposits to their customers and communication is done over the phone. These marketing calls if not directed properly can result in customer annoyance and failure of the campaign. Being a customer, we also face this annoyance a number of times when a marketing call offering irrelevant product offer comes to us. Hence we decided to work on a bank dataset targeting telemarketing in a machine learning approach.

1.2 Objective

For a telemarketing business to be success, it is very important to contact right customers with right offers. We used bank dataset to perform our analysis. This dataset consists of two sections, one is the customer details like age, job and marital status and so on. The other section had details of the campaign calls such as when was the user last contacted, duration of the call etc. The objective of this analysis was to compare the outcomes of different algorithms using the bank dataset and perform predictions of probability of the customer accepting the offer.

1.3 Literature / Market Review

1. http://www.experts.umich.edu/pubDetail.asp?t=&id=33749240206&
2. http://mlr.cs.umass.edu/ml/datasets/Bank+Marketing

Chapter 2. System Design & Implementation Details

2.1 Algorithms

2.1.1 Decision Trees

We used Decision trees to build a few predictive models, where we take observations from a dataset and map them to reach to a conclusion about the target value. It is a type of predictive model where a data is trained using values and prediction is made on the system. These consist of nodes which are the input features and branches to all the possible values for that input feature. These are of two types: Classification and Regression. Classification Tree is the one where the predicted outcome is a class belonging to the dataset. Whereas, Regression Tree gives some predicted outcome which can be some value. In our analysis we used Classification Trees: Two Class Boosted Decision Trees. This class of decision trees combine weak learners into a single strong learner, in a recursive way and gives a cost effective function.

2.1.2 Decision Forest

Decision Forests is an ensemble learning technique used for classification/regression purposes. They create a large number of decision trees while training the model and votes on the most popular output class. This voting is a kind of aggregation where every tree will give an output which is non-normalized frequency of histogram labels. This aggregation method adds up these histograms and normalizes the result to get the probabilities of each label. They have a high prediction confidence and address the problem of over fitting seen in the decision trees. Individual trees are created using two resampling methods: Bagging and Replicate. In Bagging, each tree is grown on a new sample whereas in Replication each tree is trained on the same sample data.

2.1.3 Multi-class Neural Network

Neural network is a set of layers which are inter-connected where the inputs lead to outputs using a series of weighted nodes/edges. The edges' weights are learned while training the neural network based on input data. The graph direction will then proceed through hidden layers from the inputs and all the nodes are connected by weighted edges to the nodes in next layer. These networks are used for prediction based tasks where only one or a few layers are hidden. According to recent searches Deep-neural networks are becoming an effective method for more complex tasks where successive layers increase the level of semantic depth.

2.1.4 Logistic Regression

Logistic regression presents a regression model where the dependent variable (DV) is categorical. Our dataset had binary dependent variable i.e. it can take only two values. This model predicts the probability of the response based on the input features. Logistic regression measures the relationship between the dependent variable which is categorical and one/more variable that are dependent and estimates the probabilities using a function. This function is a cumulative logistic distribution where the probability that an example belongs to class 1 is the formula:

$$p(x;\beta 0,\ldots, \beta D\text{-}1).$$

Where: x is a D-dimensional vector containing the values of all the features of the instance.
p is the logistic distribution function.

β{0},…., β {D-1} are the unknown parameters of the logistic distribution.
This algorithm works towards finding the optimal values by maximizing the probability of log of input parameters.

2.1.5 Decision Jungle

Random decision trees and forests were considered to play a huge part in machine learning but they have a limitation. Given enough data, the nodes count in decision trees grows exponentially with depth. This exponential growth comes with the cost of accuracy. Decision jungles are an extension to the decision forests. They consist of an ensemble/collection of decision directed acyclic graphs (DAGs). These are compact and powerful discriminative models of classification. In DAG we have multiple paths from root to leaf. Advantages of decision jungle:

a. Lower memory footprint and better generalization performance

b. Non-parametric models that can represent non-linear decision boundaries.

c. Perform integrated feature selection and classification.

2.2 Technologies & Tools used

2.2.1 Python

Python is object oriented, interpreted and interactive programming language. We used python to clean our dataset and categorize the values. Here is a snippet of the code:

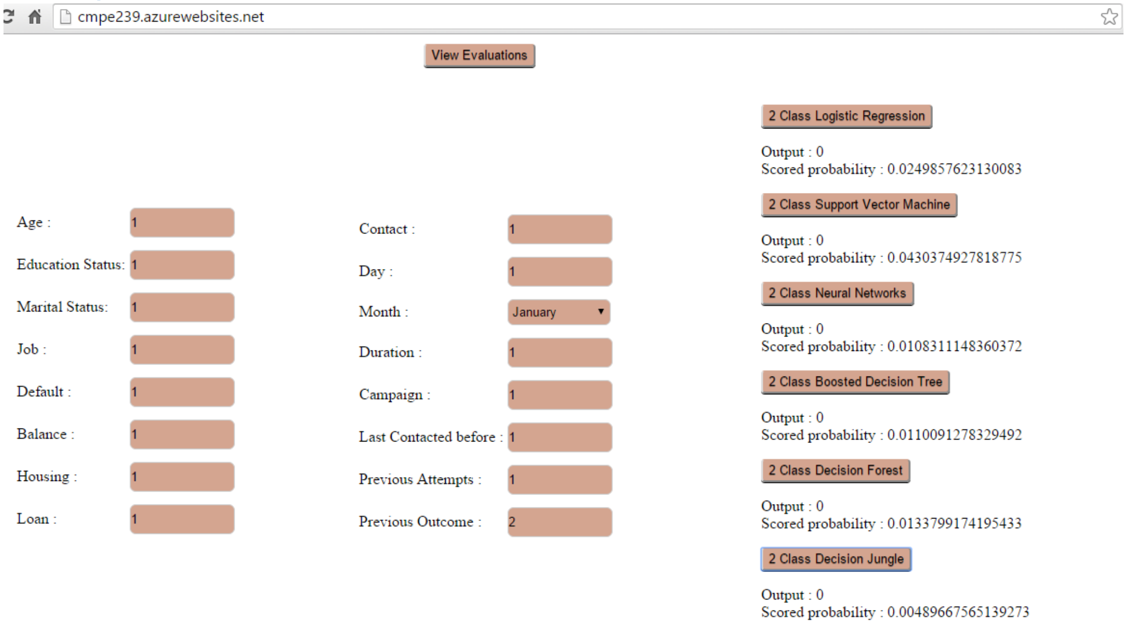
```
def get_marital(input):
  val_dict = {
    'married' : 1,
    'single' : 2,
    'divorced' : 3
  }
  for key,value in val_dict.iteritems():
    if input==key:
      return value
    else if input not in val_dict.keys():
      return input
```

where we are giving the marital status' value as 1,2,3 depending on whether they are married, single or divorced respectively.

2.2.2 Microsoft Azure Machine Learning

It provides us with a tool - Azure Machine Learning Studio. It provides an interactive and visual workspace where we can build a model and training dataset. The datasets and analysis modules are dragged and dropped to form an experiment, the results can be visualized and exported.

2.3 GUI / Screenshots



cmpe239.azurewebsites.net

View Evaluations

Age : 1

Education Status: 1

Marital Status: 1

Job : 1

Default : 1

Balance : 1

Housing : 1

Loan : 1

Contact : 1

Day : 1

Month : January ▾

Duration : 1

Campaign : 1

Last Contacted before : 1

Previous Attempts : 1

Previous Outcome : 2

2 Class Logistic Regression

Output : 0
Scored probability : 0.0249857623130083

2 Class Support Vector Machine

Output : 0
Scored probability : 0.0430374927818775

2 Class Neural Networks

Output : 0
Scored probability : 0.0108311148360372

2 Class Boosted Decision Tree

Output : 0
Scored probability : 0.0110091278329492

2 Class Decision Forest

Output : 0
Scored probability : 0.0133799174195433

2 Class Decision Jungle

Output : 0
Scored probability : 0.00489667565139273

Figure 1: Web UI - User Details

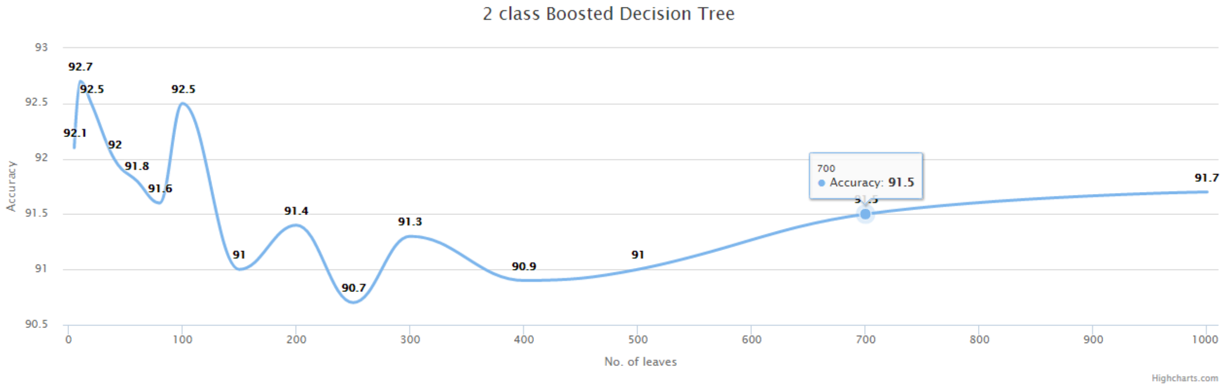


Figure 2: Accuracy for Two-Class Boosted Decision Tree

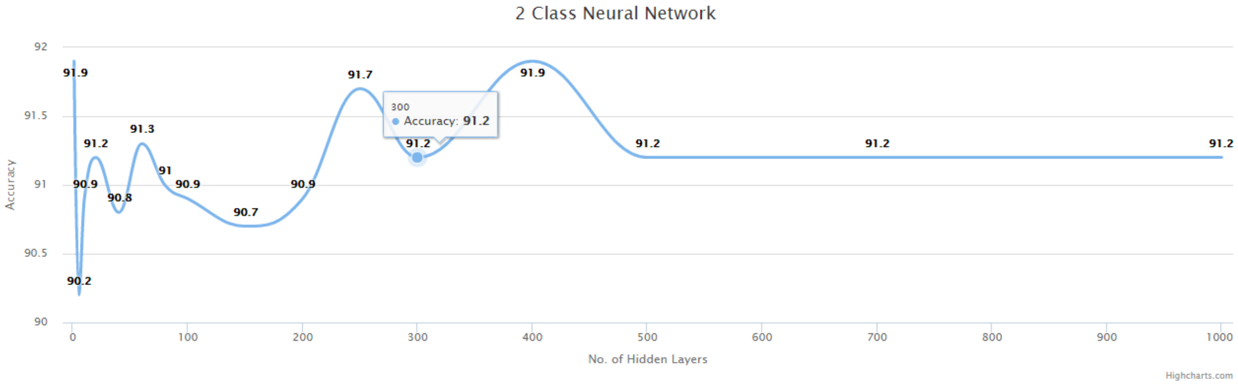
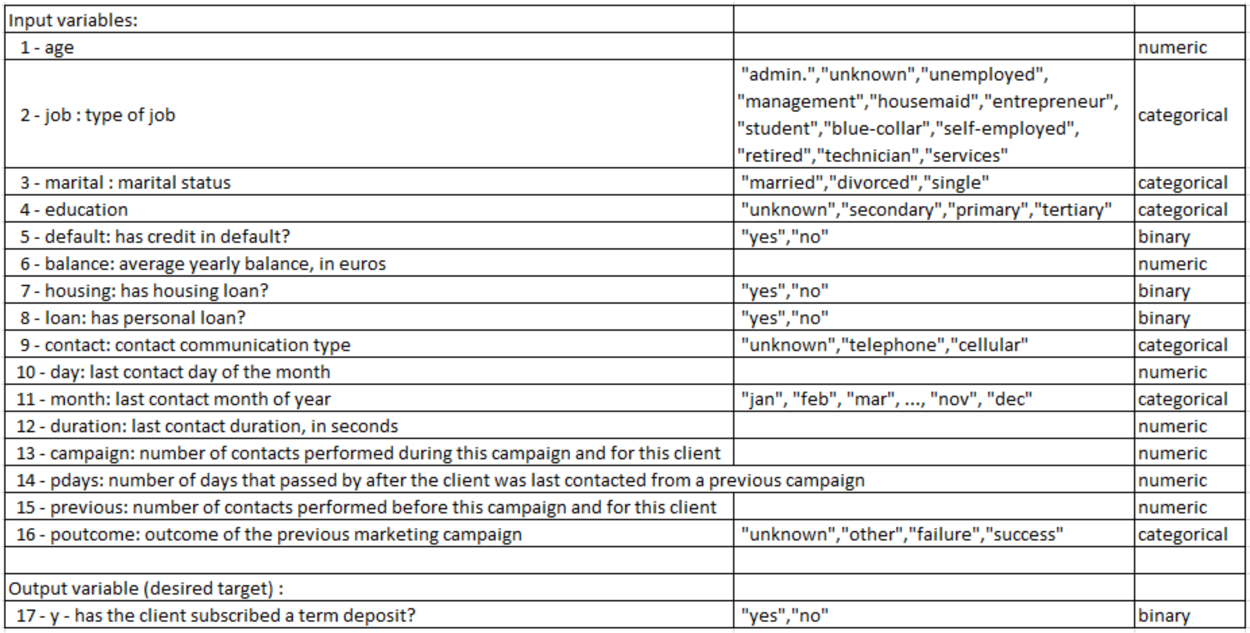


Figure 3: Accuracy for Two-Class Neural Network

Chapter 3. Experiments / Proof Of Concept Evaluation

3.1 Dataset

We used real data provided by UCI Machine Learning repository which was collected from a Portuguese retail bank from May 2008 to June 2013, in a total of 41,188 phone contacts. This dataset is related with the direct marketing campaigns of the banking institution. These campaigns were based on the calls made to the client and more often the same client was needed to contact more than once to access if the product was subscribed or not. This dataset has a total of 16 attributes as shown:



Input variables:		
1 - age		numeric
2 - job : type of job	"admin.","unknown","unemployed", "management","housemaid","entrepreneur", "student","blue-collar","self-employed", "retired","technician","services"	categorical
3 - marital : marital status	"married","divorced","single"	categorical
4 - education	"unknown","secondary","primary","tertiary"	categorical
5 - default: has credit in default?	"yes","no"	binary
6 - balance: average yearly balance, in euros		numeric
7 - housing: has housing loan?	"yes","no"	binary
8 - loan: has personal loan?	"yes","no"	binary
9 - contact: contact communication type	"unknown","telephone","cellular"	categorical
10 - day: last contact day of the month		numeric
11 - month: last contact month of year	"jan", "feb", "mar", ..., "nov", "dec"	categorical
12 - duration: last contact duration, in seconds		numeric
13 - campaign: number of contacts performed during this campaign and for this client		numeric
14 - pdays: number of days that passed by after the client was last contacted from a previous campaign		numeric
15 - previous: number of contacts performed before this campaign and for this client		numeric
16 - poutcome: outcome of the previous marketing campaign	"unknown","other","failure","success"	categorical
Output variable (desired target) :		
17 - y - has the client subscribed a term deposit?	"yes","no"	binary

Table 1: Dataset Attributes - Input Variables

3.2 Methodology

Firstly, the data was checked for duplicate rows and the values were changed to numerical values for every feature. For instance, the education level of the client was changed from primary, secondary, tertiary or unknown to 1, 2, 3 and 4 respectively. The clean dataset was then uploaded to Microsoft azure and experiments were created. Along with the dataset, the algorithm for each and every training model is added. The data is then split for training and the model is scored and evaluated. The evaluation results can be visualized in form of ROC and the statistical data to get the accuracy, precision and other factors. Every algorithm model was tested recursively by changing the testing parameters such as number of trees, maximum depths and hidden layers and so on. Results from all the variations were recorded and compared with the same as well as different algorithms.

n-folds Cross Validation

We used n-fold Cross validation where the data is split into training and testing dataset. If n subsets are used as test data sets, n-1 are used as training data sets and the average error across 'n' data-sets is computed. The advantage of this approach is that every data point appears in test set exactly once and in the training dataset 'n-1' times.

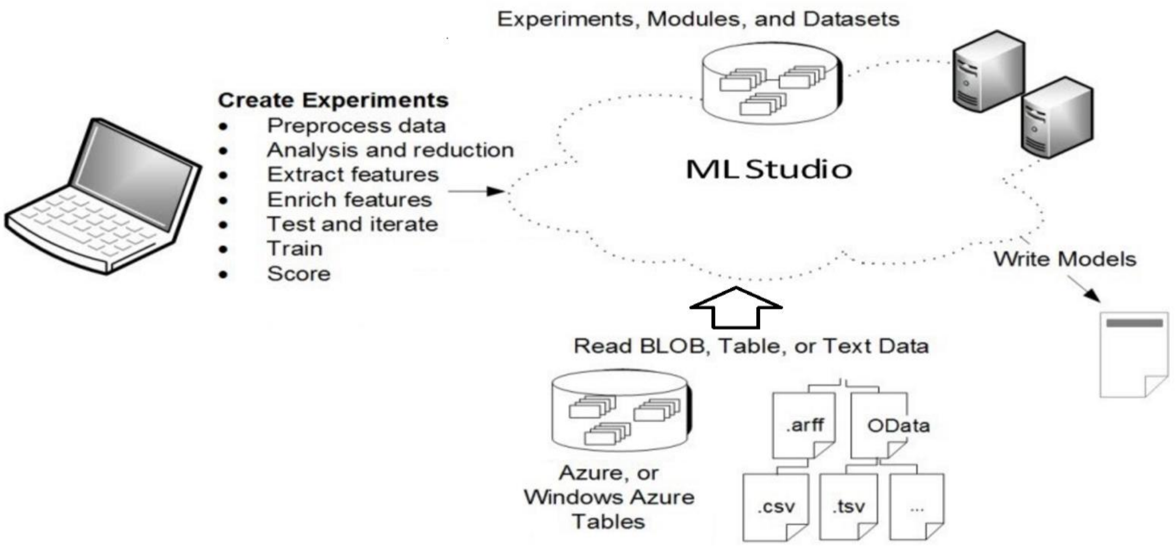
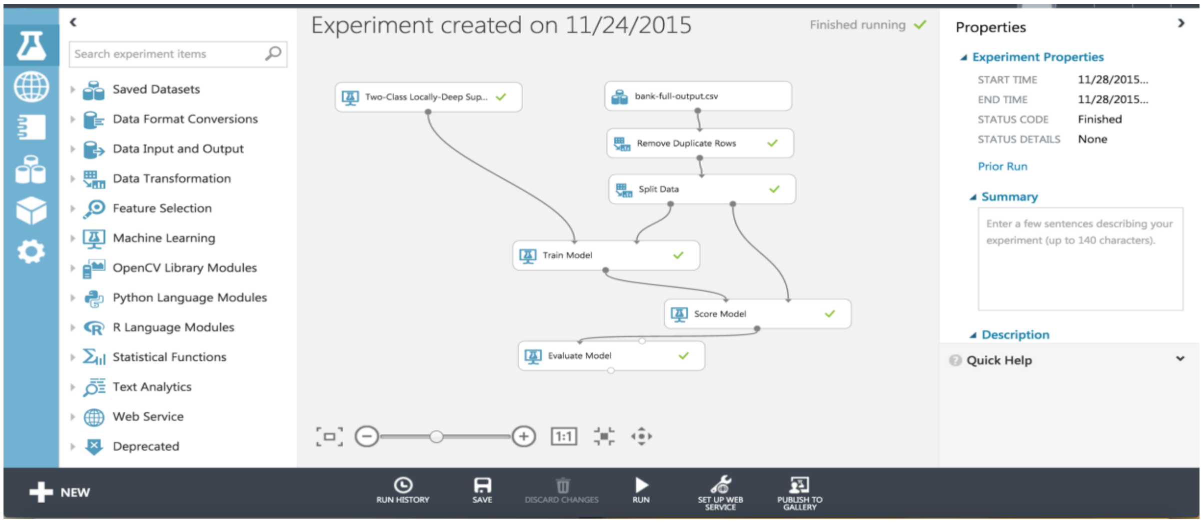


Figure 4: Methodology using Azure ML Studio



Experiment created on 11/24/2015

Finished running ✓

Properties

Experiment Properties

START TIME	11/28/2015...
END TIME	11/28/2015...
STATUS CODE	Finished
STATUS DETAILS	None

Prior Run

Summary

Enter a few sentences describing your experiment (up to 140 characters).

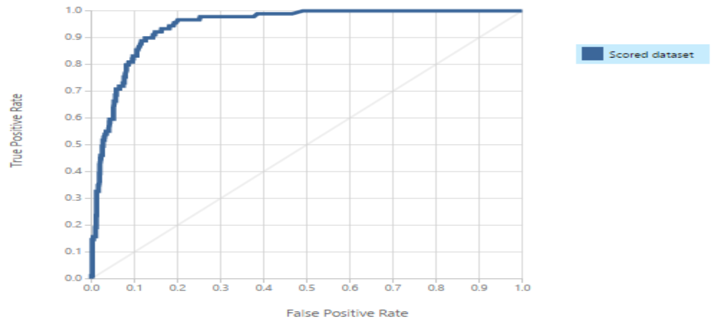
Description

Quick Help

Figure 5: Running Experiment in ML Studio

3.3 Graphs

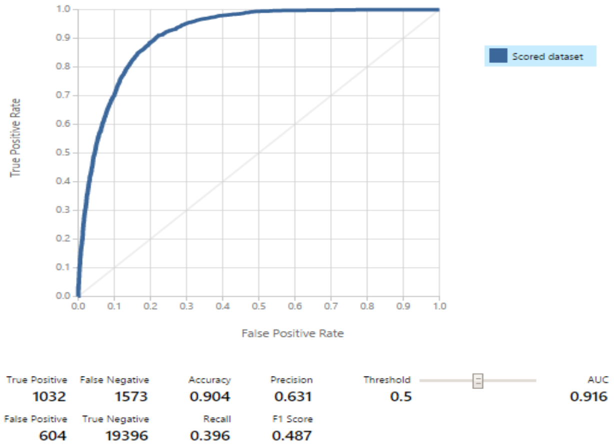
3.3.1 Two class Boosted Decision Tree



True Positive	False Negative	Accuracy	Precision	Threshold	AUC
47	42	0.927	0.662	0.5	0.943
False Positive	True Negative	Recall	F1 Score		
24	791	0.528	0.588		

Figure 6: ROC curve - Two Class Boosted Decision Tree

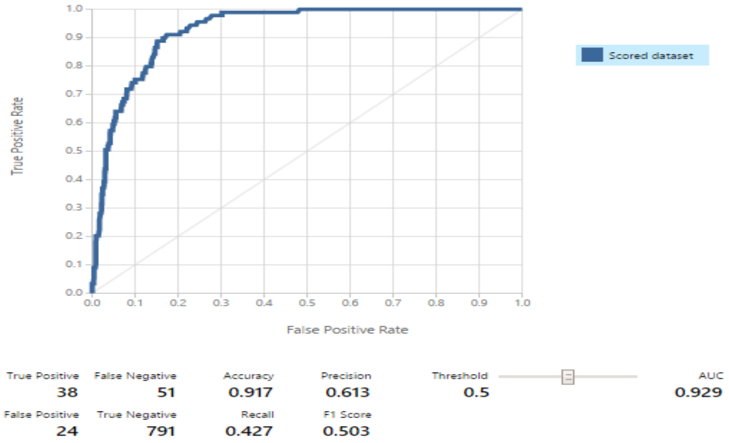
3.3.2 Two-class decision forest



True Positive	False Negative	Accuracy	Precision	Threshold	AUC
1032	1573	0.904	0.631	0.5	0.916
False Positive	True Negative	Recall	F1 Score		
604	19396	0.396	0.487		

Figure 7: ROC curve - Two Class Decision Forest

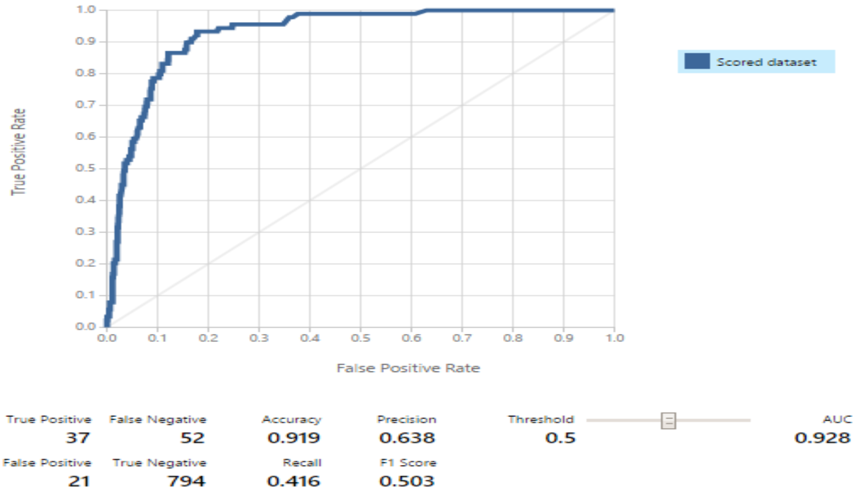
3.3.3 Decision Jungle



True Positive	False Negative	Accuracy	Precision	Threshold	AUC
38	51	0.917	0.613	0.5	0.929
False Positive	True Negative	Recall	F1 Score		
24	791	0.427	0.503		

Figure 8: ROC curve - Decision Jungle

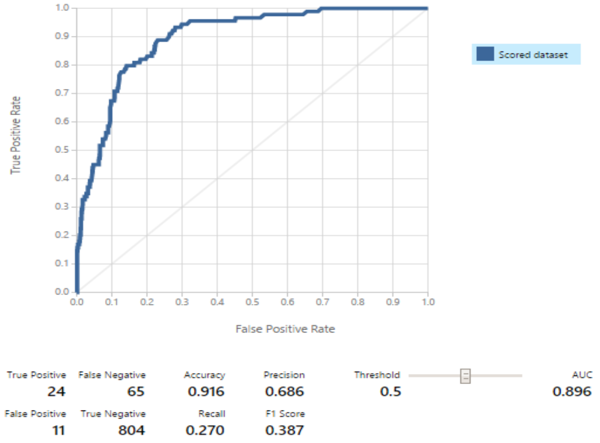
3.3.4 Multiclass Neural Network



True Positive	False Negative	Accuracy	Precision	Threshold	AUC
37	52	0.919	0.638	0.5	0.928
False Positive	True Negative	Recall	F1 Score		
21	794	0.416	0.503		

Figure 9: ROC curve - Multiclass Neural Network

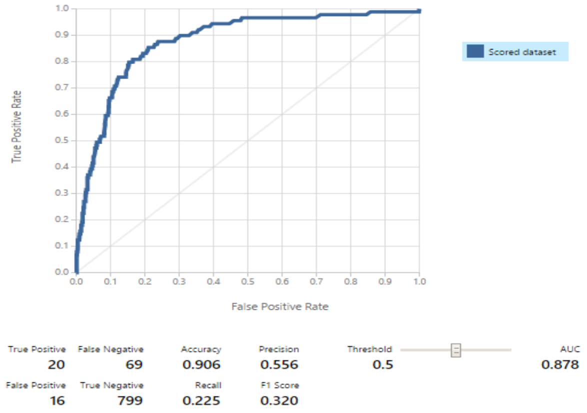
3.3.5 Logistic Regression



True Positive	False Negative	Accuracy	Precision	Threshold	AUC
24	65	0.916	0.686	0.5	0.896
False Positive	**True Negative**	**Recall**	**F1 Score**		
11	804	0.270	0.387		

Figure 10: ROC curve - Logistic Regression

3.3.6 Support Vector Machine



True Positive	False Negative	Accuracy	Precision	Threshold	AUC
20	69	0.906	0.556	0.5	0.878
False Positive	**True Negative**	**Recall**	**F1 Score**		
16	799	0.225	0.320		

Figure 11: ROC curve - Support Vector Machine

3.4 Analysis Of Results

The below table gives an overview of all the results from the experiments and shows the best parameter metric to be selected for a given algorithm:

Algorithm	Best Parameter	Maximum Accuracy
Two-Class Neural Network	hidden layers : 1,400	91.9
Two-Class Decision Jungle	no. of decision directed acyclic graphs: 20	91.7
Two-Class Support Vector Machine	-	90.6
Two-Class Boosted Decision Tree	no. of leaves : 10	92.7
Two-Class Logistic Regression	-	91.6
Two-Class Decision Forest	maximum depth : 8 number of trees : 25	90.4

CHAPTER 4. CONCLUSION

After performing a number of experiments with the dataset using the above mentioned 6 algorithms by changing parameters for each and every algorithm we found out that "Two-Class Boosted Decision Tree" gave the best accuracy (**92.7%**) for this dataset.

CHAPTER 5. PROJECT PLAN / TASK DISTRIBUTION

5.1 Break Into Components And Clearly Explain

5.1.1 Who was assigned what task

Name	Task
Exploring datasets	Rakesh Balusa, Moksha Bhargav , Apoorva Gupta
Exploring Algorithms and Technologies	Rakesh Balusa, Moksha Bhargav , Apoorva Gupta
Exploring Tools and Technologies	Rakesh Balusa, Moksha Bhargav , Apoorva Gupta
Cleaning of Dataset	Rakesh Balusa, Moksha Bhargav , Apoorva Gupta
Algorithms and Graphs	Rakesh Balusa, Moksha Bhargav , Apoorva Gupta
UI	Rakesh Balusa, Moksha Bhargav , Apoorva Gupta
Final Testing	Rakesh Balusa, Moksha Bhargav , Apoorva Gupta

5.1.2 Who ended up doing what task

Name	Task
Exploring datasets	Rakesh Balusa
Exploring Algorithms and Technologies	Rakesh Balusa, Moksha Bhargav , Apoorva Gupta
Exploring Tools and Technologies	Rakesh Balusa, Apoorva Gupta
Cleaning of Dataset	Rakesh Balusa, Moksha Bhargav
Algorithms and Graphs	Rakesh Balusa, Moksha Bhargav , Apoorva Gupta
UI	Moksha Bhargav
Final Testing	Rakesh Balusa, Moksha Bhargav , Apoorva Gupta