**Using AzureML Data Mining Models to Evaluate**

**Santander Customer Transaction Prediction**

**Data Mining: ISM6136.001S19**

**By**

**Divya Macha**

**Kirti Tiwari**

**Srikrishna Srinivasan**

**Ritik Gupta**

**Shreyas Hastantram**

**May 1, 2019**

**Table of Contents**

[**Introduction**](#_upgoh9xpokiq) **2**

[**Dataset Description**](#_nqsucwxuh7e1) **4**

[**Logistic Regression**](#_a0byc05eycx4) **6**

[**Neural Networks**](#_ofrbtki2ntvt) **7**

[**Scored Model Result Evaluation**](#_u8ac49erbbrp) **8**

[**Interpretation and Comparison of Evaluation Model Results**](#_osvytv9ws76o) **9**

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# **Introduction**

Santander has been serving customers in the Northeast since 2013. Their mission is to help customers prosper. They do it with simple ways to spend, save, and manage customers money better.

**Project Goal:** The goal of this analysis is to use Microsoft Azure Machine Learning Studio to explore, modify, model and assess this data and need to predict whether the customers will perform transaction for the next time in future or not.



Questions that frame the exercise?

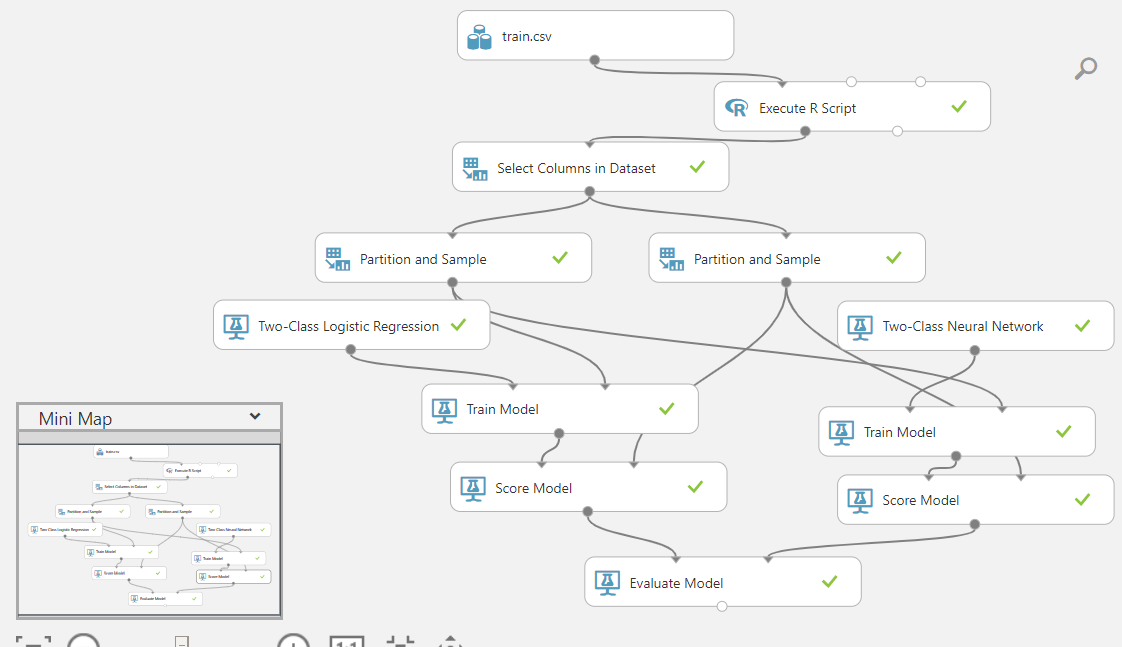
Santander uses a lot of binary classification to find out interesting insights like will the customer be able to pay the loan, is customer satisfied, etc.The goal here is to harness technology to predict whether a customer will make a specific transaction in the future or not for the leader or CEO to devise plan for strategic marketing. In the data which we have, we can observe that most of the customers won’t make a purchase i.e target is 0 all the time. So, Important concerns are why is most of the customers are not making purchase, Or on what conditions are they making a purchase. How can the company assess which customers will make a purchase. Furthermore, Is there some way to find a solution to make the customer, make transaction.

Recommendations for the problem?

If we look the problem as the data science problem, we need to predict target variable i.e will the customer make transaction or not the next time. And for that we need to fit the model to give us correct accuracy. Once we get the accurate target variables, CEO can look into factors affecting that and help guide marketing team.

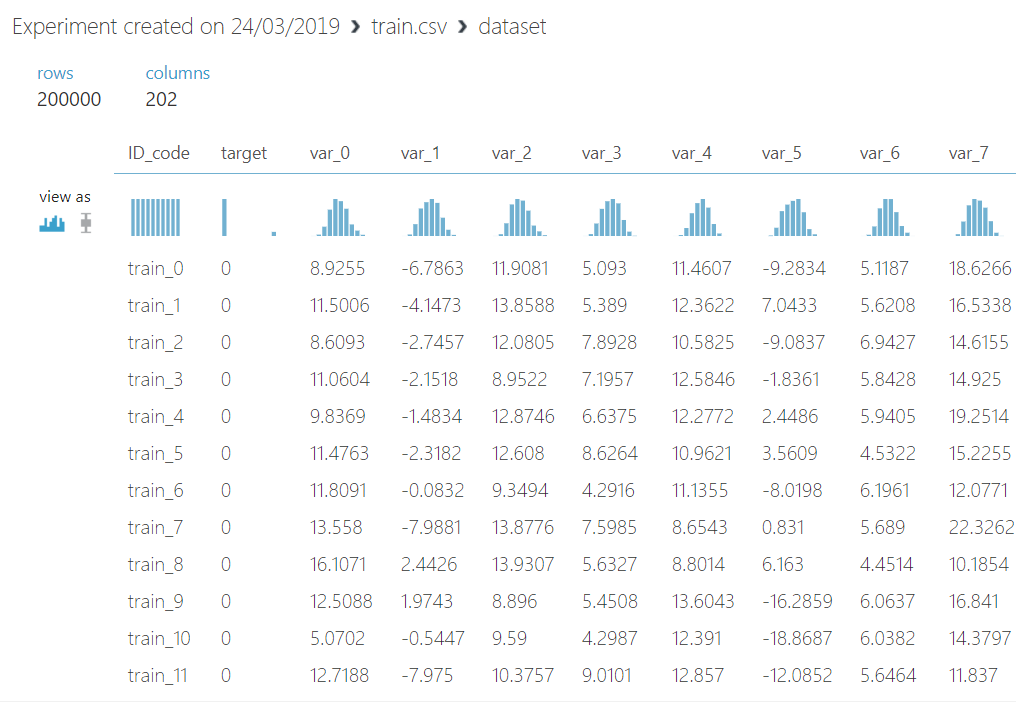
Models which we used to solve this real world problem is Logistic Regression and Neural Networks

**Here is our model,**



We have used Azure ML to build this model, train.csv is the dataset which is downloaded from the kaggle website.

# **Dataset Description**

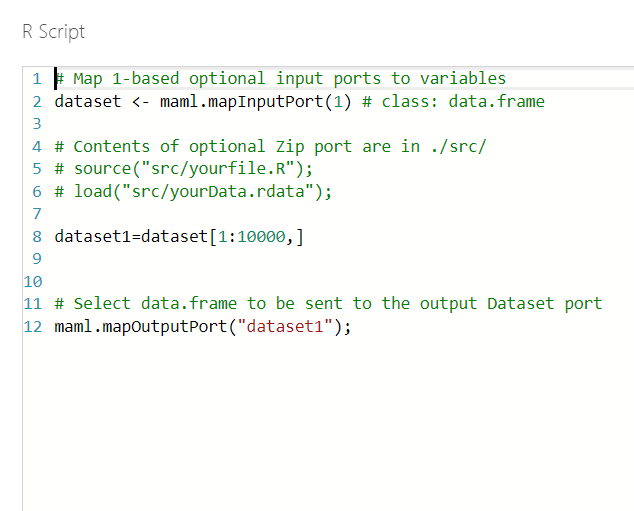


It is evident from the screenshot that there is 202 columns and 200000 rows,the variables names has been hidden from the dataset, they have generic names.

The company is not letting there trade secrets out, variables might also been scaled so that the person working on the data won’t be able to predict what this variable corresponds to in real world.

All the variables are distributed normally which indicates the fact that this is a real data and reliable data.

There are lot of zeros in the dataset when compared to 1 , this indicates there might be lot of failed transactions than the successful one’s.



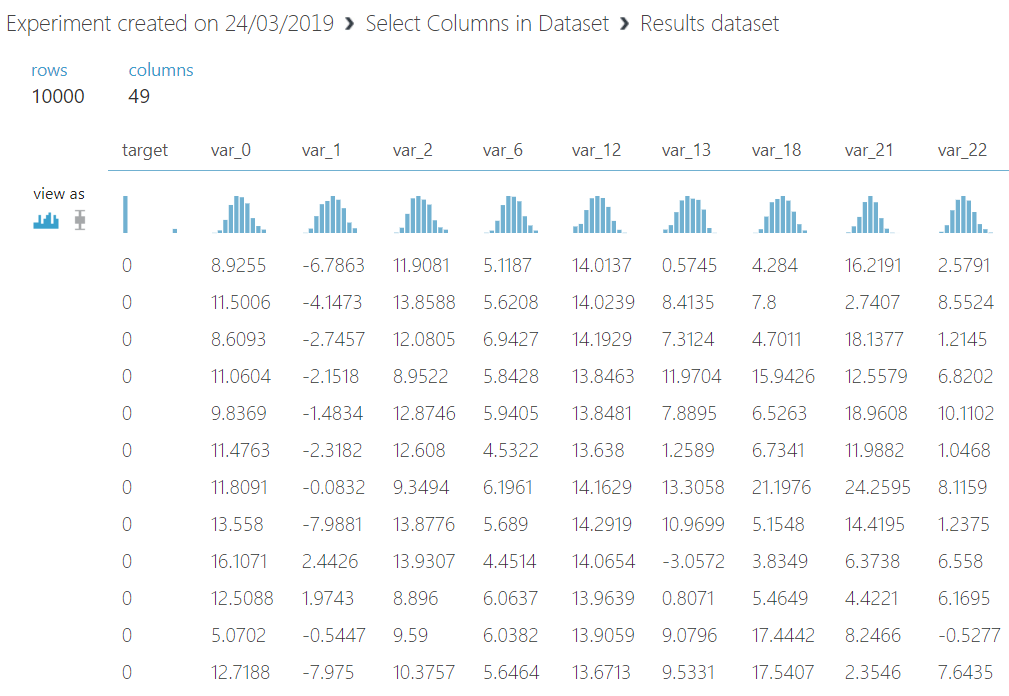
We wanted our model to perform quickly hence we wrote a simple r script to select 10000 rows from the dataset.

We can from the command dataset[1:10000,], it is saying select 10000 rows and select all the columns.

The same dataset will be sent as the output from the module.

We are using a rscript module in azure ml to perform this function.

Since there is 202 variables and we wanted few among them, we ran a feature selection model on the data and we found 49 columns are more correlated with the target variable.



# **Logistic Regression**

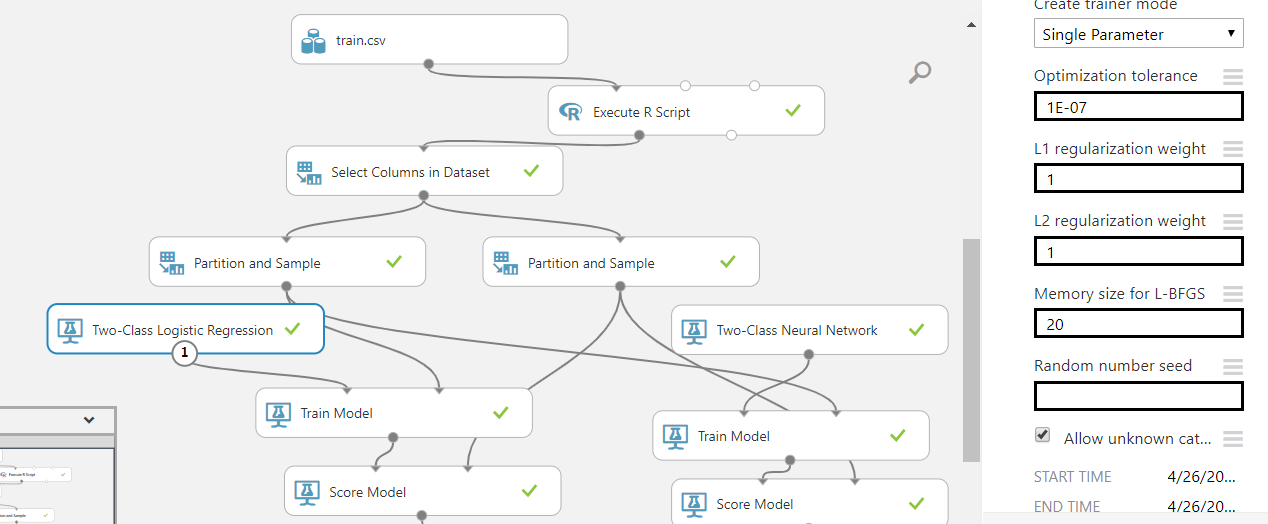
For this problem we used our first model as logistic regression. We chose logistic regression because we want to know the strength and statistical significance of each variable contributing to binary state of the target variable and in turn predicts its probability. We used L1 and L2 weights as 1 to prevent overfitting.

Using the two class logistic regression tab in the Azure ML we passed our features and sampled data. Based on this training data, logistic regression model will be calculated with each variable with some beta coefficient. This all will flush into sigmoid function to create a probabilistic value to determine the target. The sigmoid function is specified as

P(*target*=1)=1/1+*e*−*z*

*z*= B0+xB1+xB2+xB3………..

In our model all the x are the variables with their B terms indicating bent to probability



After, Applying logistic regression we train our model and after training we score our model to observe the table with probabilistic values to determine target.

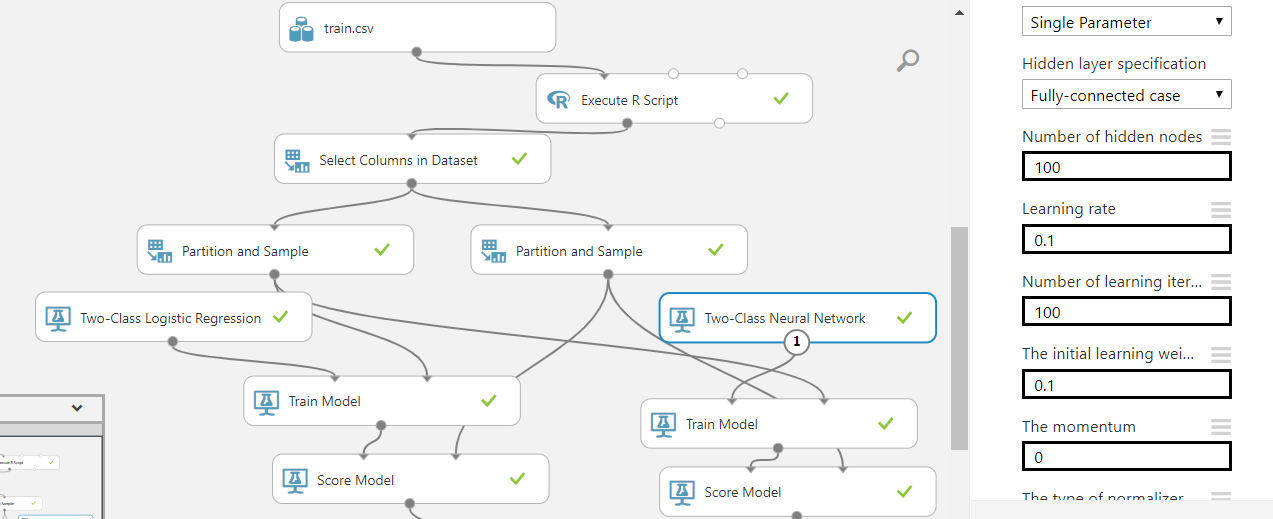
# **Neural Networks**

We have also applied Neural Network model so that we can compare the results with Logistic Regression and see which model has performed better .

We used Two-Class Neural Network here. It is one of the supervised learning methods for classification. It is a set of interconnected layers starting with Input Layer which has number of nodes equal to the number of features of the training data and based on that we will predict the outcome.The input layer is connected to the nodes in the hidden layer which is comprised of weighted edges and nodes and finally connected to the output layer. We can also customize some features for the hidden layer so that it can give better performance like number of hidden layers (default is 1),number of hidden nodes(default 100), learning rate,number of learning iterations, momentum etc.

As we used two-class neural network, all inputs must map to one of two nodes in the output layer.

Below is the screenshot of Neural Network model from Azure ML displaying its feature selection as well.

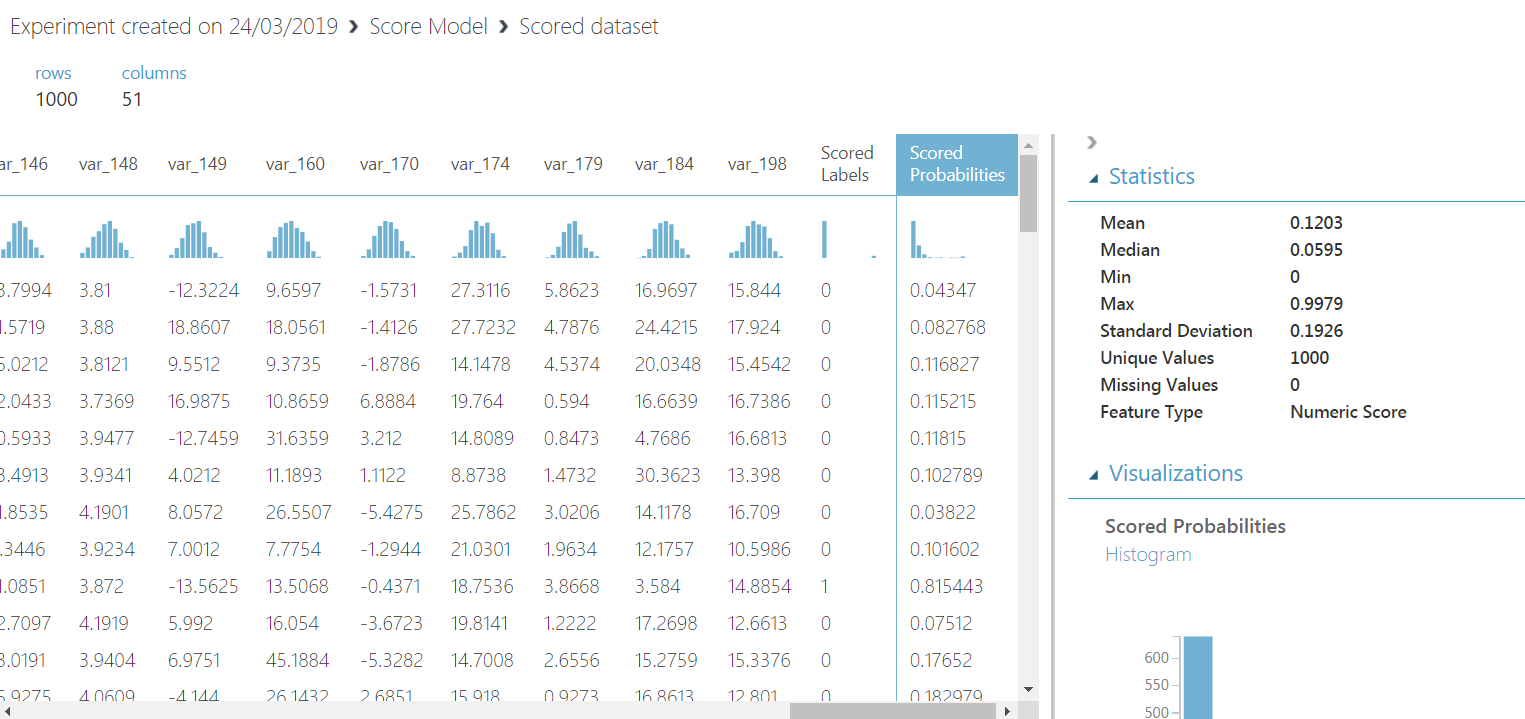


Next step is to train the model and validate its results and we can see the scored labels and scored probabilities which the model has predicted when we visualize the scored model results. We can check the Statistics like mean, median, standard deviation etc and based on that can determine if the model is good or not.

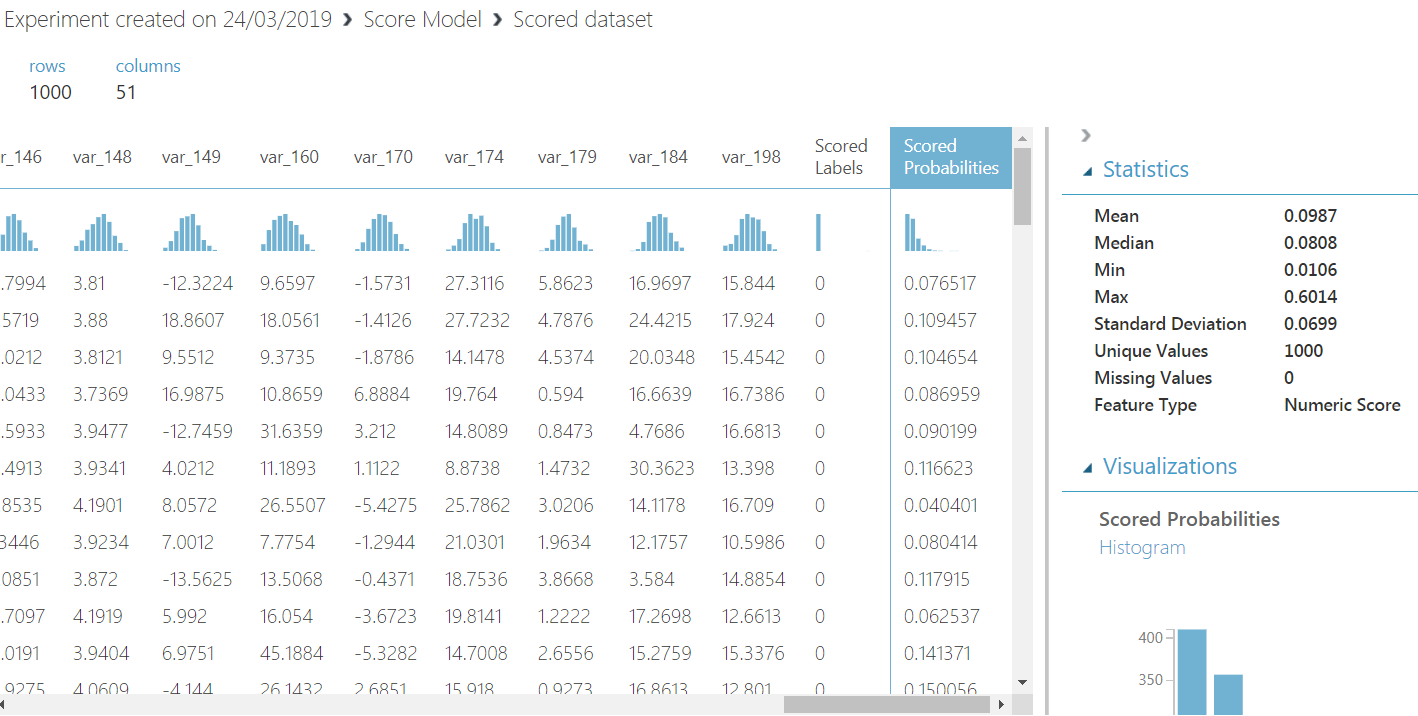
# **Scored Model Result Evaluation**

Below are the evaluation Results for the two model which we have used.

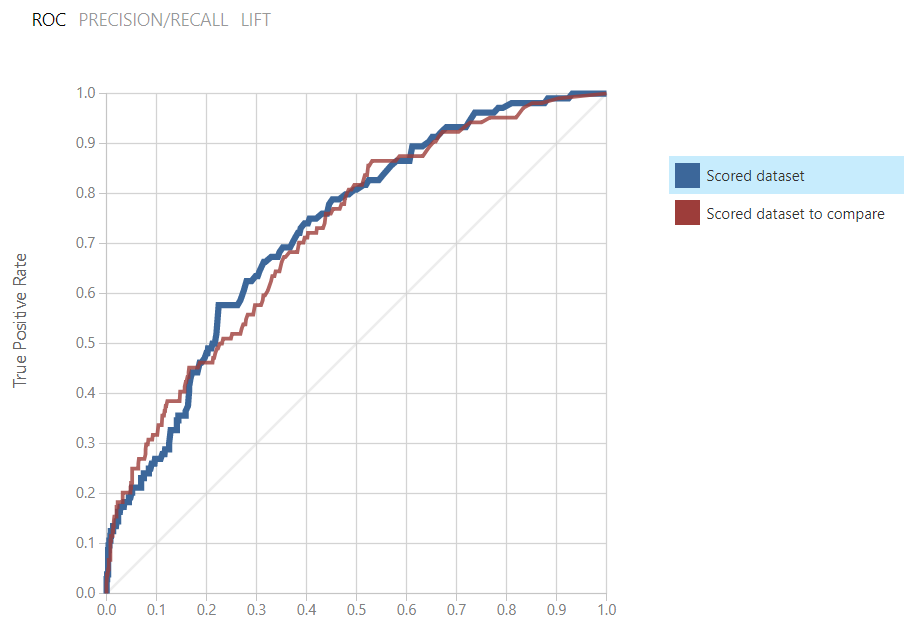
Score Model Results for Two-class Neural Network



The mean and standard deviation are not very close, therefore this is not as good compared to logistic regression (shown below)

Score Model Results for Two-class Logistic Regression :

# **Interpretation and Comparison of Evaluation Model Results**



ROC Plot and AUC measure:

1. Blue curve is for the model on the left and it is the Two class Logistic regression

2. Red curve is for the model on the right and it is the Two class Neural network

3. Lines connect lower left corner to upper right corner in the graph

4. Each point represents how model performs along two dimensions – false positive and true positive

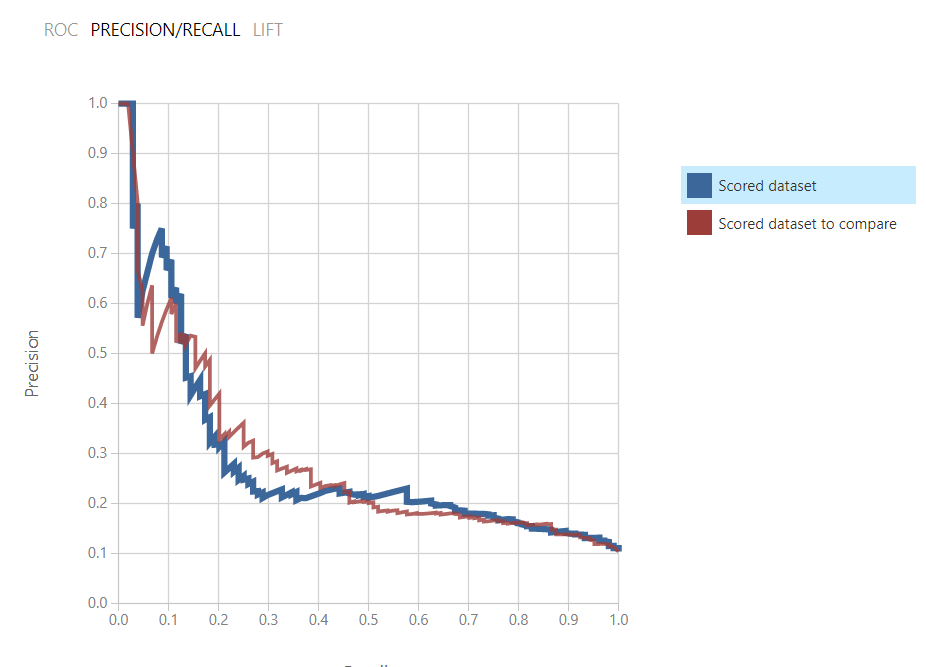
5. There is a threshold. By changing it we decrease the frequency of one type of error at the expense of increasing other type of error

6. The line which comes closest to upper-left corner provides best predictions

7. If the curve is towards the middle, it is not good in performance

8. The worst possible model is a diagonal line, it produces random predictions with the same distribution as class distribution

9. It strikes right balance between the two types of errors

Logistic regression is a better solution as seen above.

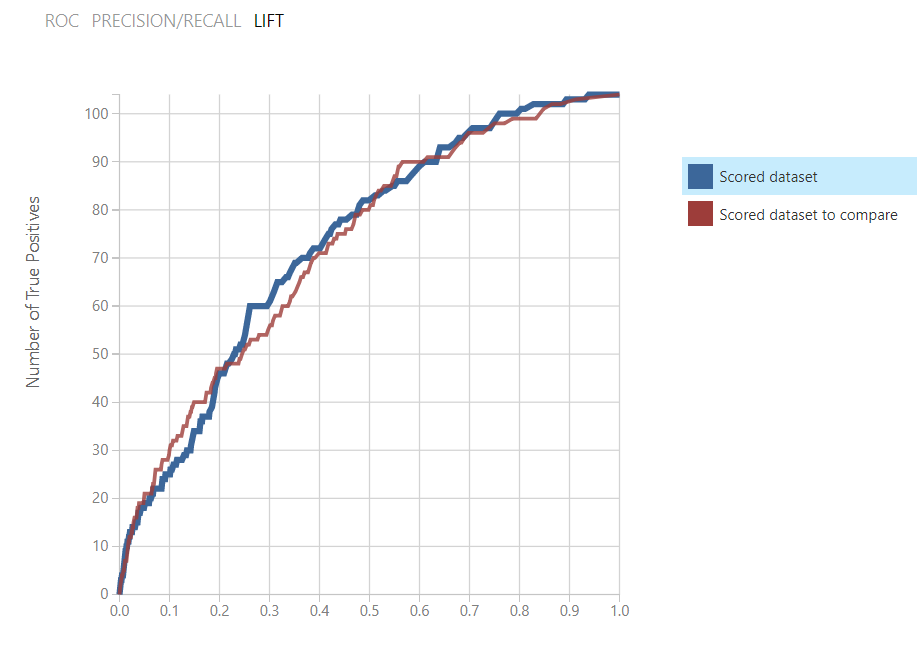
Precision / Recall plot

The curves are pulling to the left (usually it is expected to pull to right, but left is considered acceptable). Blue curve pulls better, hence it is a better model.

1. Recall = TP / (TP + FN)

2. Precision = TP / (TP + FP)

3. The curve to the upper right corner is the best model



Area Under the Curve / AUC:

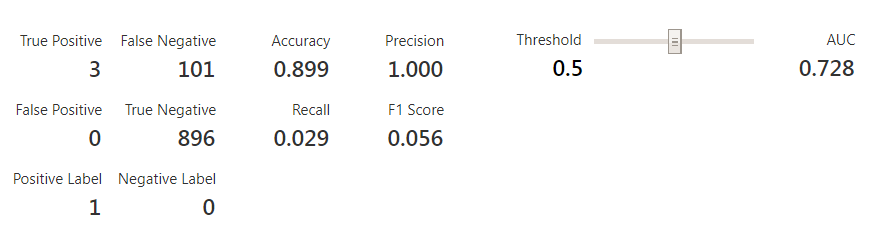
1. Shows the amount of the area under the ROC curve, value between 0 and 1

2. Value should never be less than 0.5 (diagonal)

The models are good, as they are pulling to the right, asway from diagonal.

False Positives: Actual value is NO and predicted value is YES

True Positives: Actual value is YES and predicted value is YES



**Accuracy** = Correctly predicted to Total = TP+TN / (TP + TN + FP + FN)

**Precision** = Correctly predicted positive to total predicted positive = TP / (TP + FP)

**Recall** (sensitivity) = Correctly predicted positive to all observations = TP / (TP+FN)

**F1 score** = weighted average of precision and recall = 2\*(Recall\*Precision)/(R+P)

The model has a very good Accuracy, as True Positives and True Negatives are much higher compared to False Positive and False Negative.

Precision is 1 because False Positives are 0

Recall is very low, because, True Positive is a small fraction compared to False Negatives.

# 