**Microsoft Azure Project**

(Audumbar S Deo)

**Data:**

Data is taken from Kaggle ( <https://www.kaggle.com/c/titanic/data> ) training and testing data is given separately.

**Size of training data:** 891 observations explained in 12 features.

**Size of testing data:** 418observations explained in 12 features.

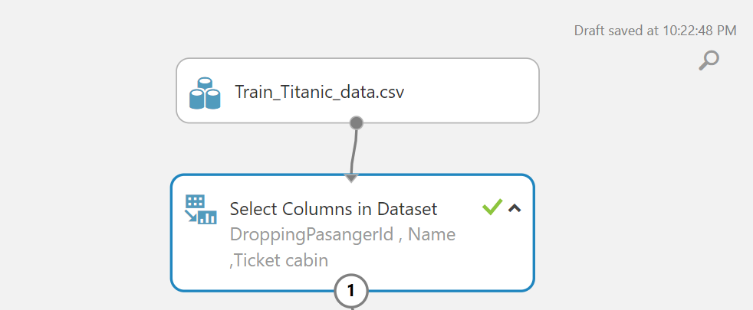
The dataset also has 12 columns that each record an attribute about each occupant’s circumstances and demographic. For this particular experiment classification model is built that can predict whether or not someone would survive the Titanic disaster given the same circumstances and demographic.

**Pre-processing and Data exploration**

First some preprocessing is required to use the data. Begin by identifying columns that add little-to-no value for predictive modeling.

These columns will be dropped.

PassengerID, Name, Ticket, Cabin etc.

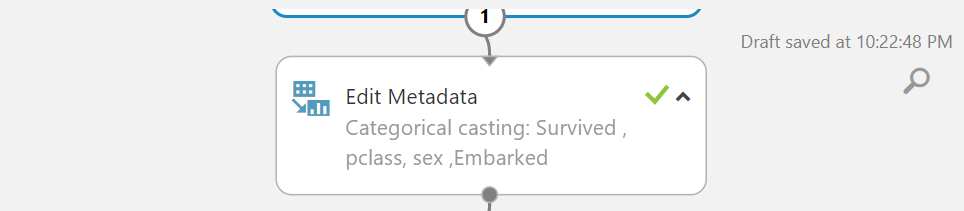


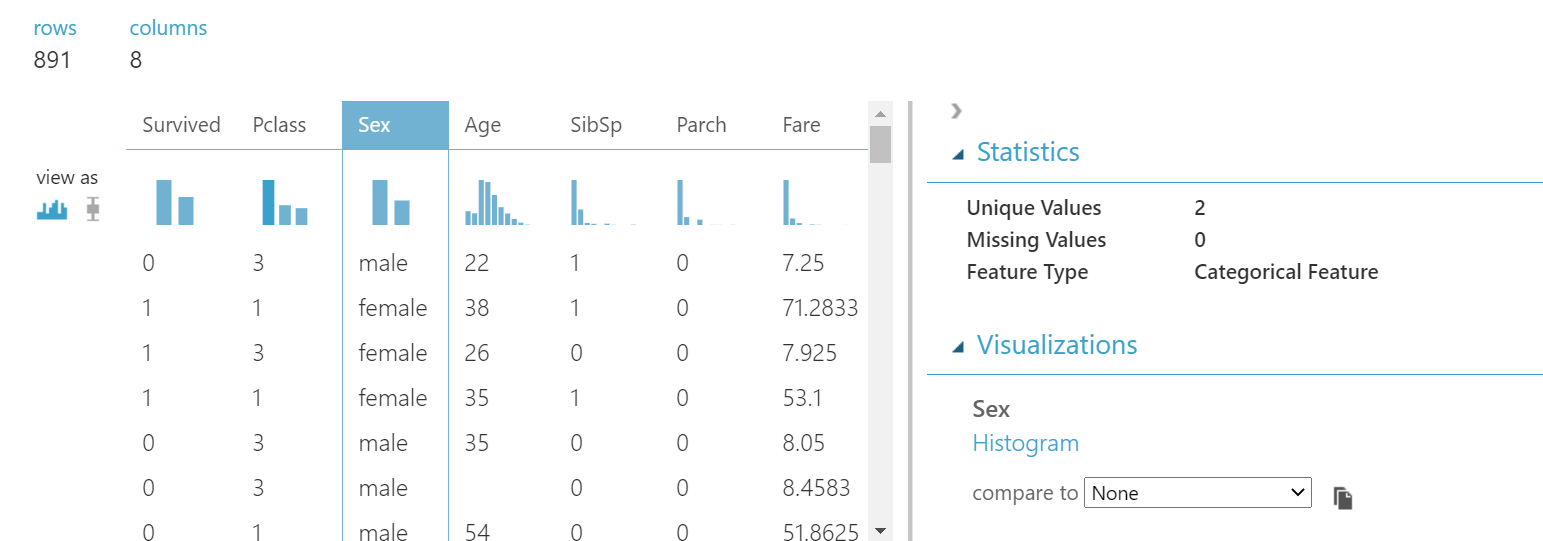
**Define Categorical variable**

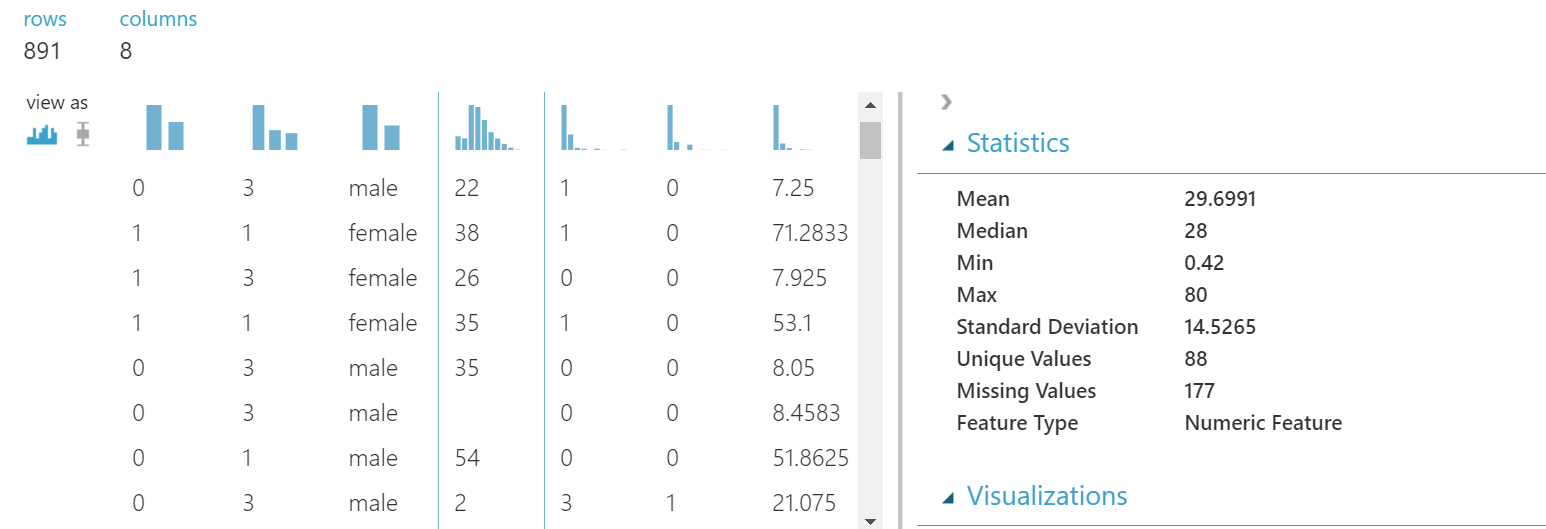
We must now define which values are non-continuous by casting them as categorical.

Using “edit metadata” convert following variable as type : categorical

Variables: Survived, pclass, sex, Embarked

****





**Clean missing data**

Most algorithms are unable to account for missing values and some treat it inconsistently from others. To address this, we must make sure our dataset contains no missing, “null,” or “NA” values. There are many ways to address missing values.

Drag in a "Summarize Data" module and connect it to your "Edit Metadata" module. Run the experiment and visualize the summary output.

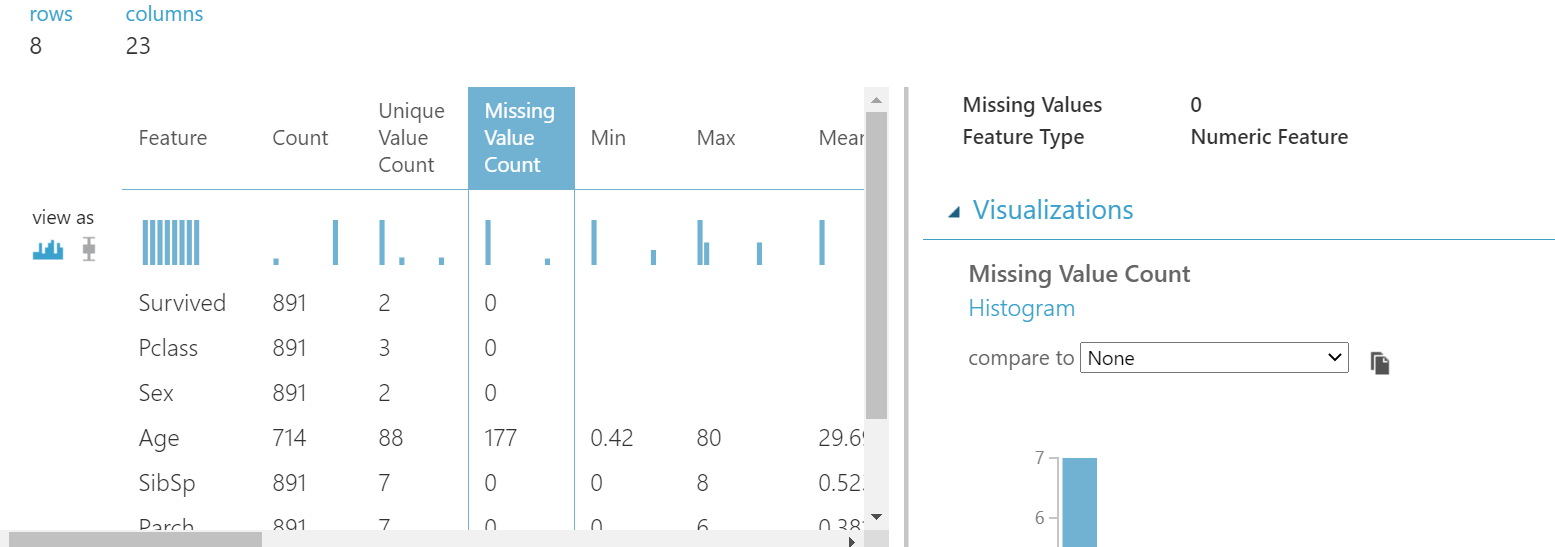
There is ‘Missing value count” for each attribute.

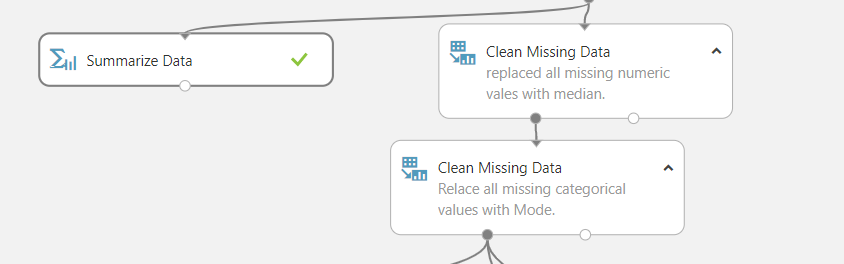
So, in this dataset, “177” missing values are there in age column and 2 missing values are there in embarked column.

“Age” is the numeric column and “embarked” is categorical column.

Use “clean missing data” module in Microsoft Azure, First replace all the missing values with median for numeric variable. and again use “clean missing data” to replace missing values in categorical variables using mode.

Summerised data

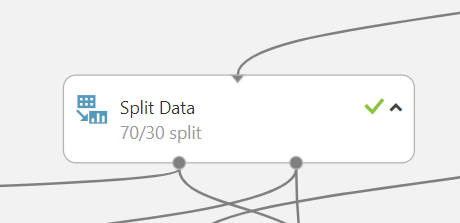




**Building Model**

It is extremely important to randomly partition your data prior to training an algorithm to test the validity and performance of your model. A predictive model is worthless to us if it can only accurately predict known values. Withhold data represents data that the model never saw when it was training its algorithm.

Split Data into 70% train and 30% test dataset using ‘Split data” module.



**Select an algorithm**

In this project I have selected two classification algorithms.

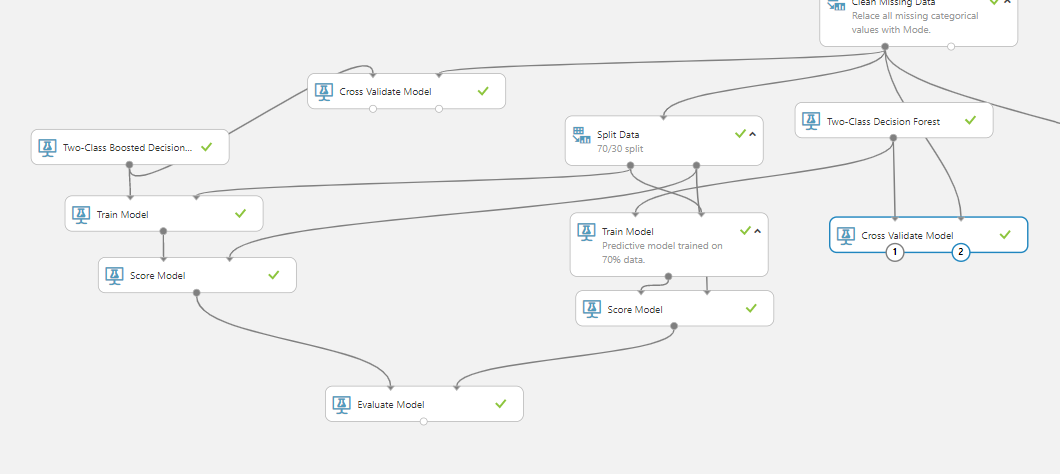
1. **Two class Decision forest**
2. **Two class boosted decision tree.**

**Train Model**

**Using ‘train model” module use the 70% trained data to train the model on both the algorithms.**

**To score the model, drag in a “Score Model” module. Connect the “Train Model” to the left input node of the “Score Model,” and the 30% withhold data to the right input node of the “Score Model.” Finally, to evaluate the performance of model, drag in an “Evaluate Model” module and connect its left input to the output of the “Score Model.”**

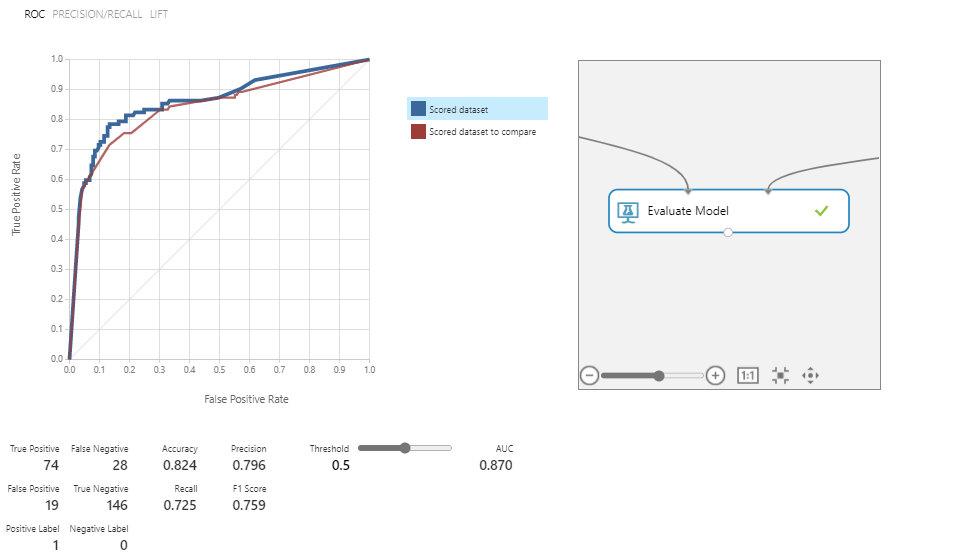
**And use “cross validation score” module to check the performance of both the model.**

****

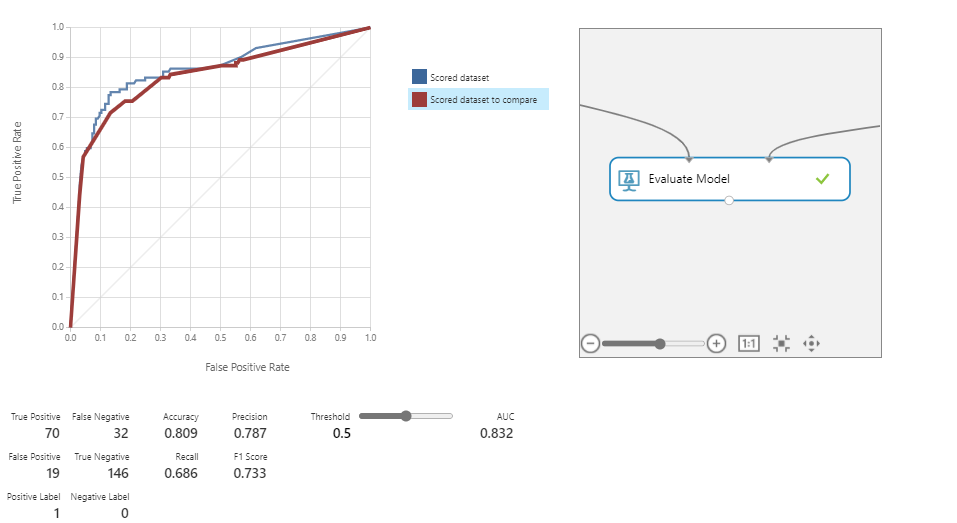
**Comparison of both models**

**Comparison of both models using ‘evaluate model’ module. Here accuracy of both the models and Auc Roc curve is given to compare the two models.**

**Accuracy for Two class boosted decision tree algorithm.**

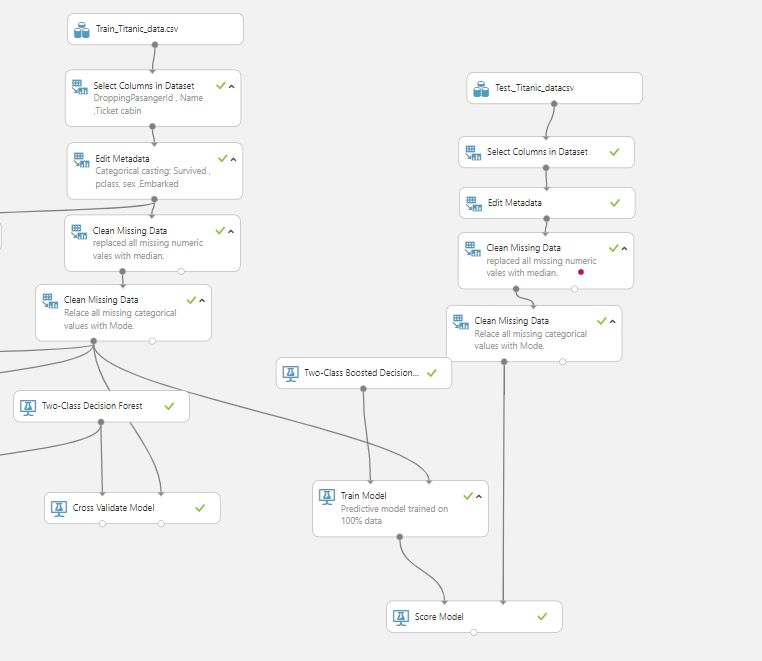
****

**Accuracy for Two class decision forest algorithm.**

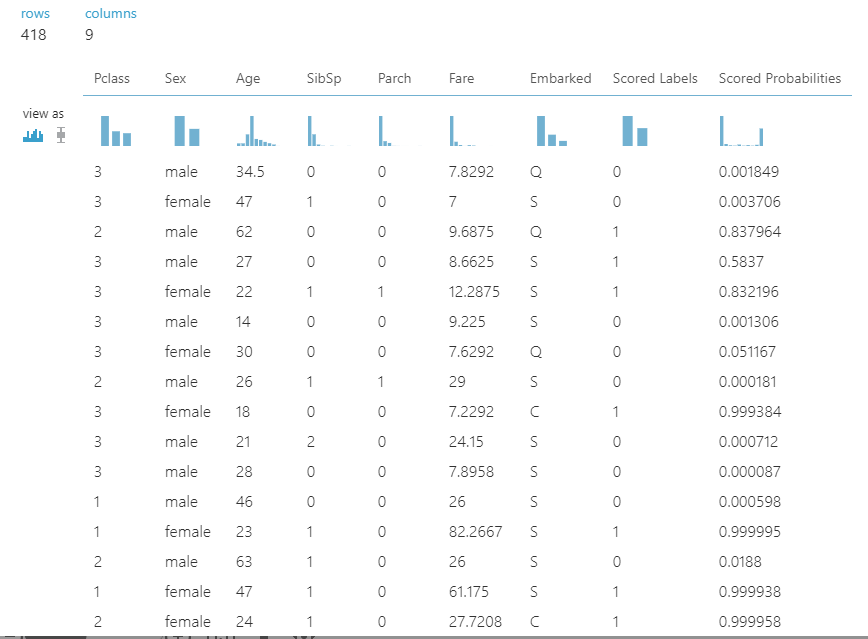
****

So, from both the algorithms two class boosted decision tree algorithm has achieved higher accuracy. This algorithm is used to predict the classes on unseen data.

Test data provided on Kaggle is imported and all the data cleaning processes performed on the training data are applied on this testing data and “two class boosted decision tree algorithm” is used to predict the final output.



Final score of predicted output:



Final Project outline:

