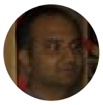
Un-supervised vs Supervised learning



Rajaram VR
Jun 26 · 6 min read

Hello, I just started my initial steps into Data science and Machine learning, and got introduced to “Supervised Learning” techniques as “Classifiers (Decisiontreeclassifer from sklearn kit) , and on the un-supervised learning , with “Clustering”.

What i set out to do was plain and simple, I picked up a dataset “Breast cancer — wisconsin” (https://www.kaggle.com/uciml/breast-cancer-wisconsin-data/download) , and set this objective,

a) Perform clustering (k-means) , use evaluation methods like silhoutte score and WSS (within sum of squares) to find optimal “clusters”.

b) Perform a Decisiontreeclassifier model , and the traditional train vs test samples and evaluate the model with ROC/AUC

c)Compare the clustering model output with the efficiency of Decisiontreeclassifer model outcome

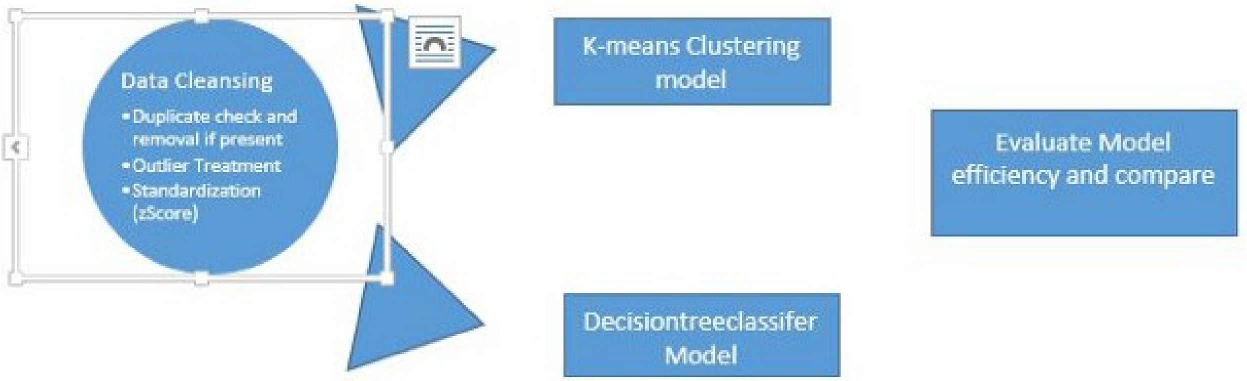
The comparison outcome, presented a surprise to me, where without the target / class variables, the accuracy with just clustering , **was close to 95 %** match to the actual class variables in the data set, better than Supervised learning **(with 70 : 30** , train to test split up, the accuracy **was 92 %)**. now does this mean, this will work for larger samples also, is to be validated with larger data sets.

Let us get started , ***Data insights :***

Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image.

Total rows — 569 , columns — 32 (including class variable, called Diagnosis, with outcome as Malignant (M) and Benign (B).

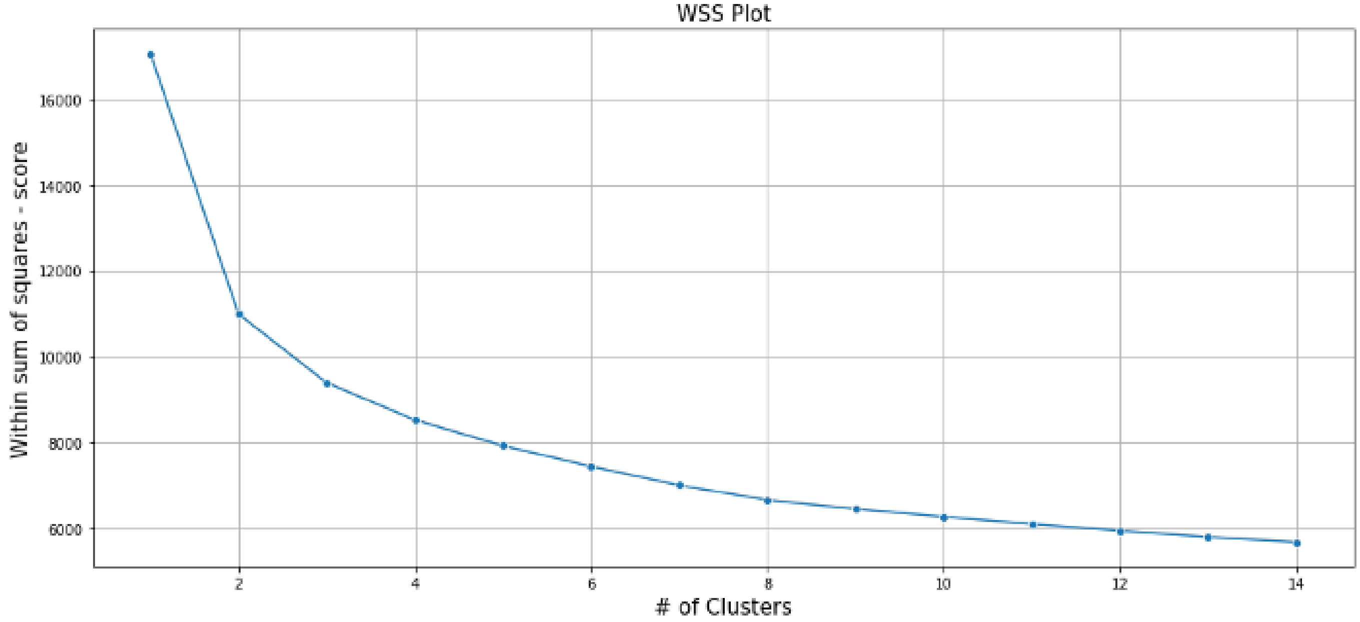
Data Process steps, were ,


Data Cleansing
•Duplicate check and removal if present
•Outlier Treatment
•Standardization (zScore)
K-means Clustering model
Evaluate Model efficiency and compare
Decisiontreeclassifer Model

The Data set did not have any missing or duplicate values, but had a lot of outliers, across most of the columns, and the treatment of outliers was a IQR based outlier treatment. This was followed by standardization (zScore) as the scale and range was different between features.

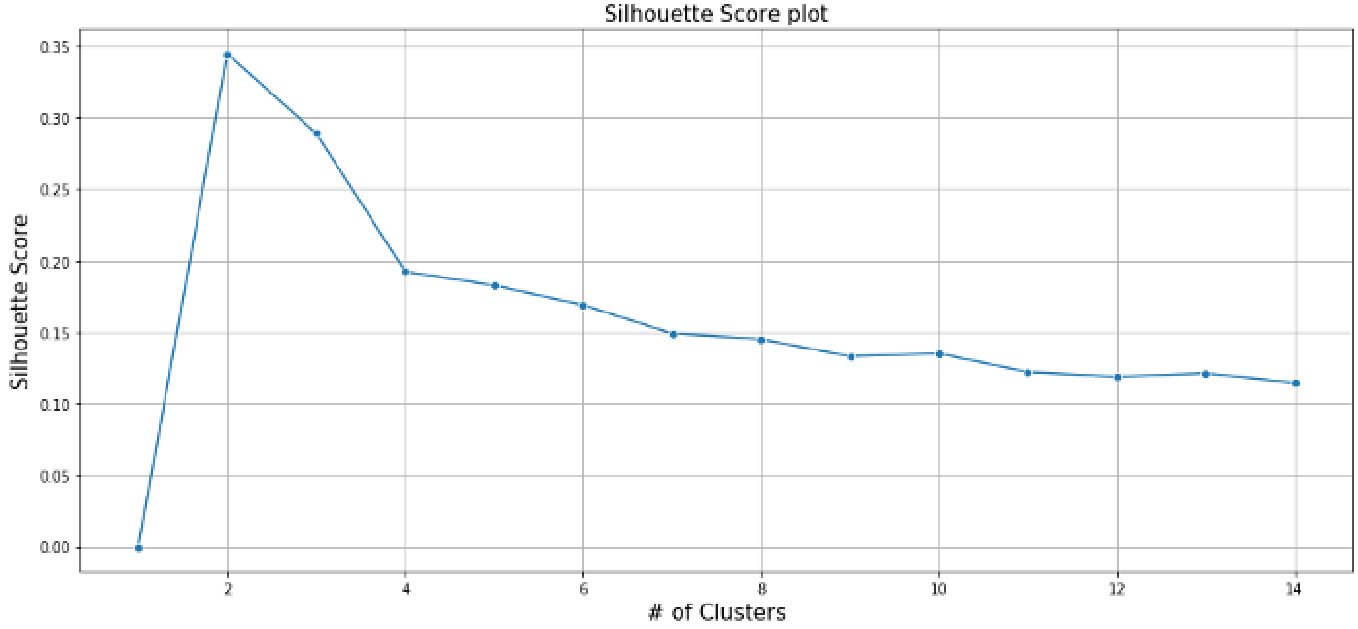
This cleansed data, was separated into features and classes (target) and then provided as input to both K-Means Clustering and also Decisiontreeclassifier models.

K-Means Clustering ,is an Un-supervised ML algorithm. It is traditionally used to identify number of clusters that can be formed given a data set. If we ignore the fact that this original data set had a target variable, and just look at only features, and if we assume we dont know the number of "buckets" in the target variable , the mechanism to decide how many clusters should be optimal for a data set is very business specific and also driven by many factors. given this data set, and looking at WSS plot below, it might seem there is no significant change in WSS score , from 8 onwards. so # of clusters from this plot could seem like # 8. But given the business scenario of "Cancer data set" , it wouldnt make sense to have 8 class buckets or target, or may be ??



Within Sum of Squares (WSS) plot

Further validate this , let us look at silhouette score..



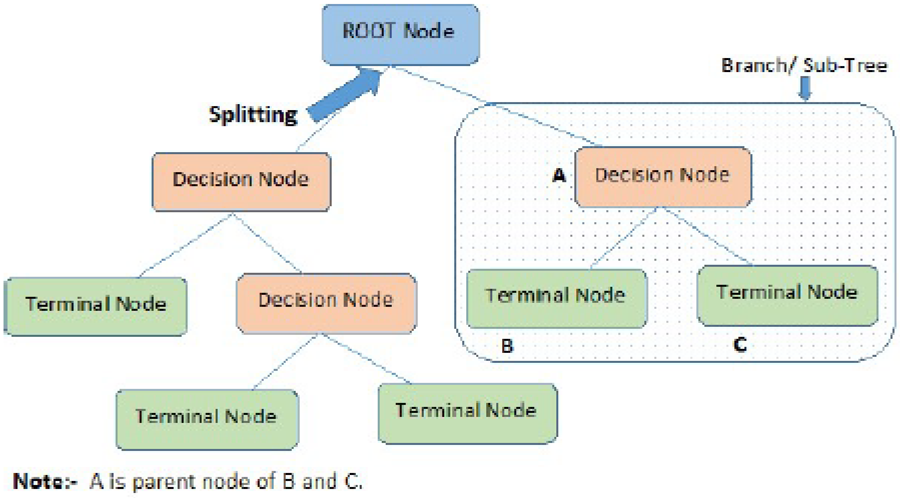
Silhouette Score plot

Silhouette score, is a measure of average distance between clusters and how tightly observations are clustered in a cluster. Though the WSS plot indicates a optimal cluster of 8, the silhouette score is more apt , as the optimal cluster is indicative of the “average score” , and the maximum average score as per above Silhouette score plot is at **2 clusters.**

Supervised Learning — Decisiontreeclassifier

As the name suggests, a supervised learning model requires both features and the target variable (class variable) , in this supervised learning approach, we are going to use Decisiontree to classify rows based on features, into a binary decision outcome 1 or 0. In the business scenario considered, 1 would mean Malignant (M) and 0 would mean Benign (B).

In this model approach, a decision tree,is constructed at the end, and it typically has the Root node, branch / sub-tree (under root nodes), as depicted below, the higher the number of branches, it would mean the model is overfitted, and less branches would mean under fitting. An optimal approach or number of branches is what would be required, and the evaluation mechanism of such models, are varied, and we would use confusion matrix and Area under curve AUC and ROC (Receiver operating characteristics) to validate the model efficiency. A decision tree is depicted as below,



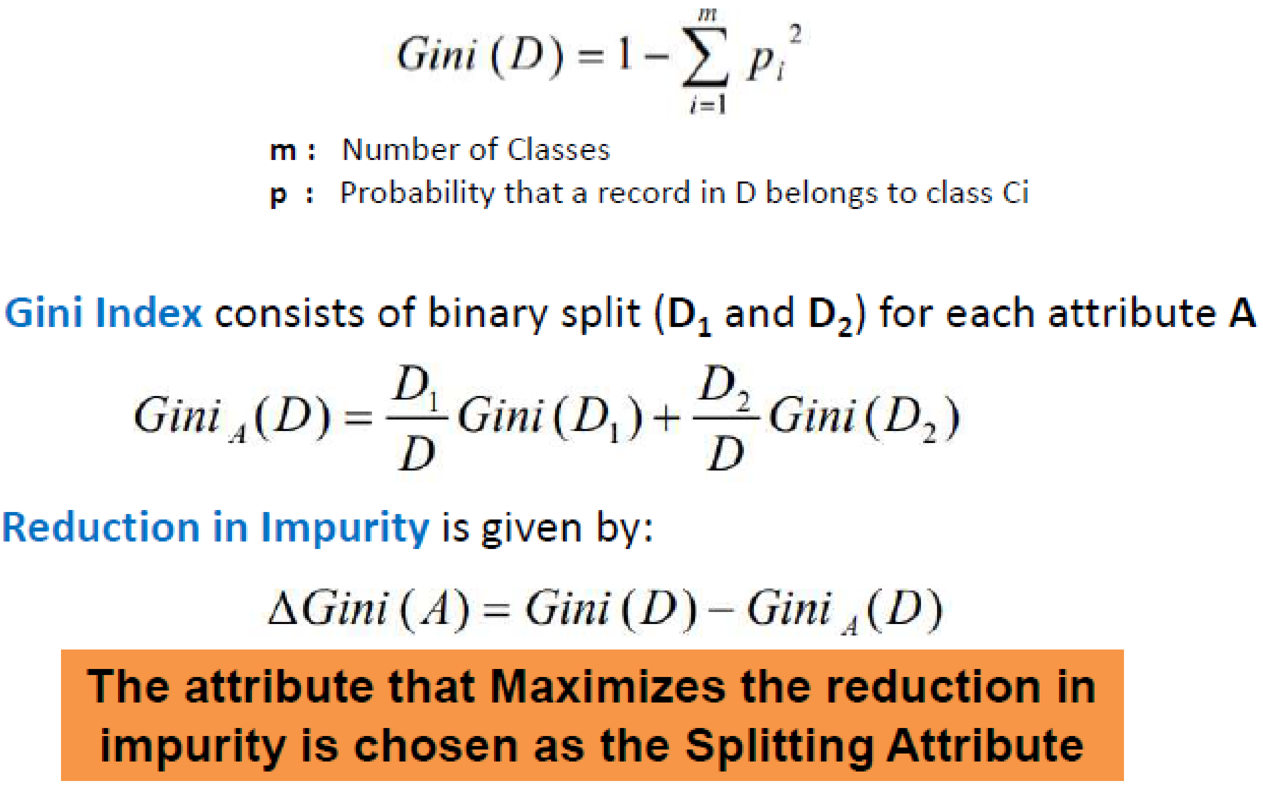
Decision Tree

"Decisiontreeclassifier" modeling available under sklearn.tree module

dt_model = DecisionTreeClassifier(criterion = 'gini' ,random_state=123)

before invoking the model , and fitment , we split the data set into training and testing samples, using the sklearn.modelselection module and library train_test_split. we present a 70:30 split between train and test .

A quick glance at the parameters, we would using “Gini” criteria, which is , basically,



$$Gini\,(D) = 1 - \sum_{i=1}^{m} p_i^{\,2}$$

m : Number of Classes

p : Probability that a record in D belongs to class Ci

Gini Index consists of binary split (D_1 and D_2) for each attribute **A**

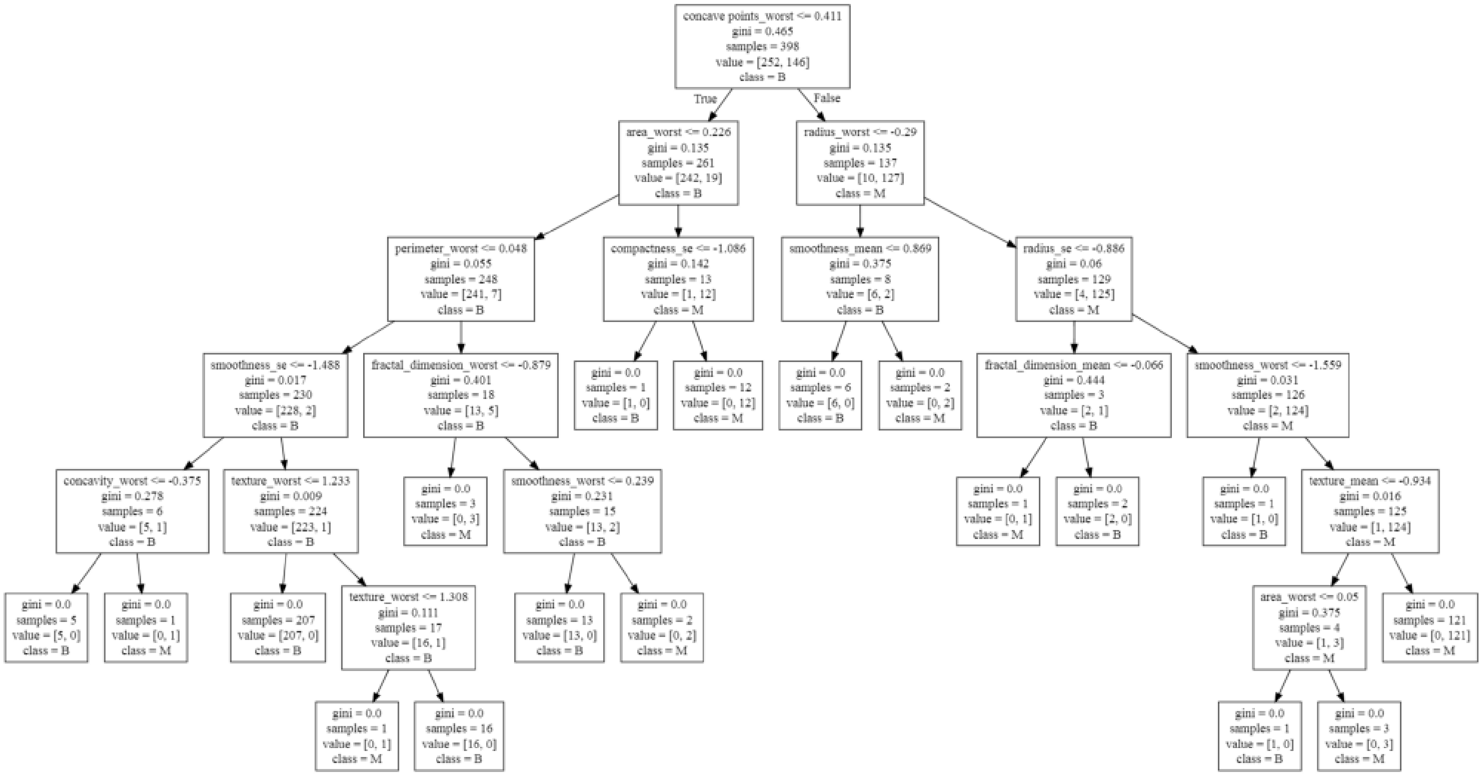
$$Gini_A(D) = \frac{D_1}{D} Gini\,(D_1) + \frac{D_2}{D} Gini\,(D_2)$$

Reduction in Impurity is given by:

$$\Delta Gini\,(A) = Gini\,(D) - Gini_A(D)$$

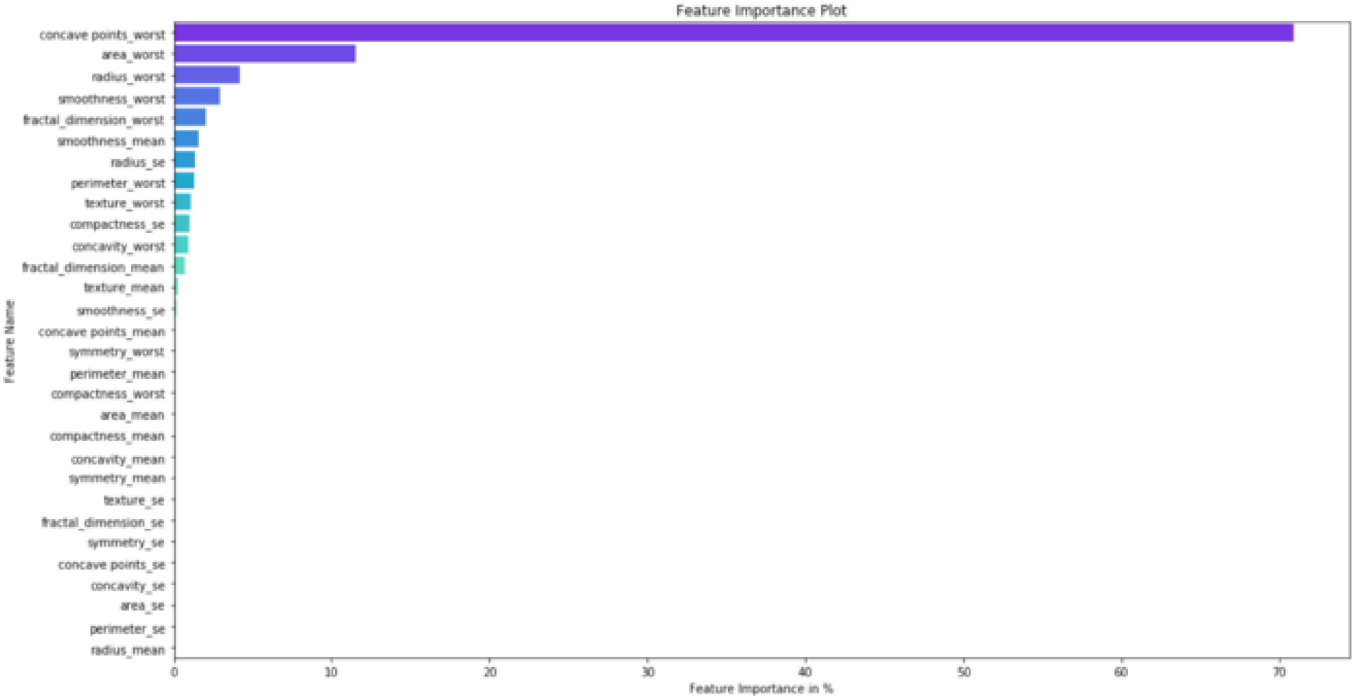
The attribute that Maximizes the reduction in impurity is chosen as the Splitting Attribute

Gini Criteria



Decision Tree view after model is trained

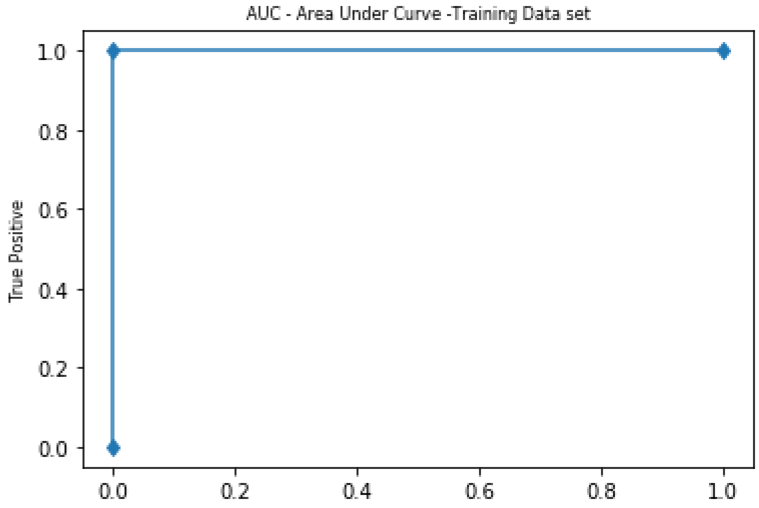
Decision tree , generated above, depicts 6 branches, for the training set, the decision tree classifier model, provides a "features_importance" for every feature. this is critical to understand how each feature influences the final outcome and what is the weight of each , the below plot depicts which features have higher influences,



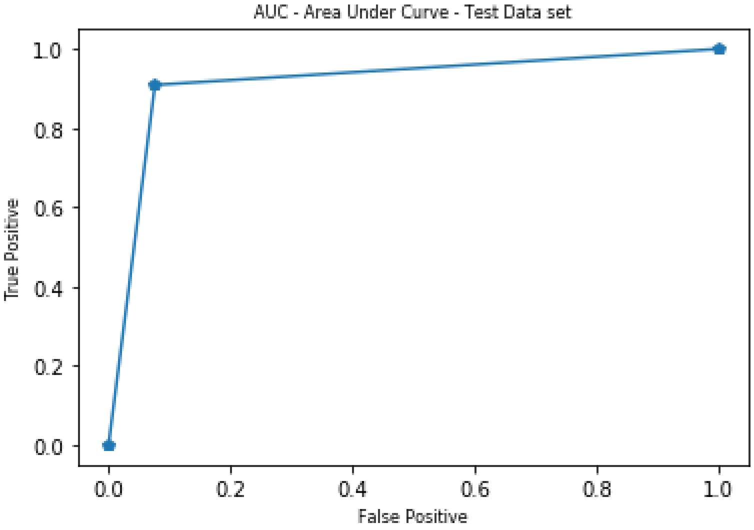
Features Importance

If we notice the feature "Concave Points_worst" has the highest influence of 70.9 % , and the root_node uses this feature to basically generate the split to the next branch.

Let us look at the AUC plots , for training data set, test data set and comparison of K-Means with test and training data set

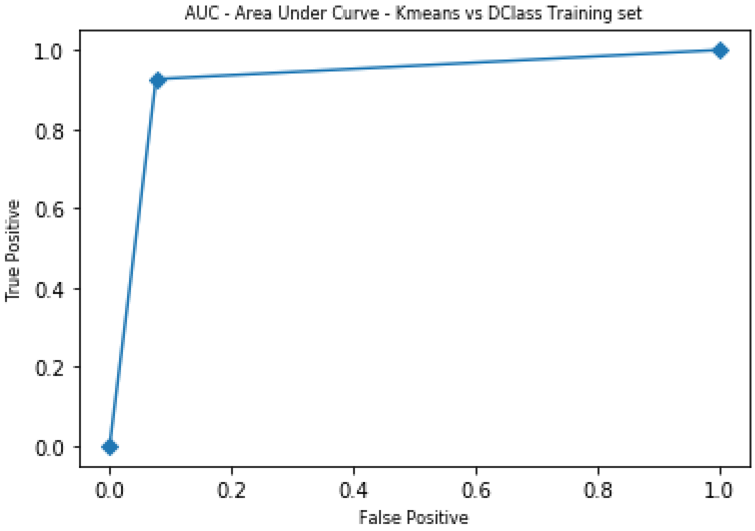


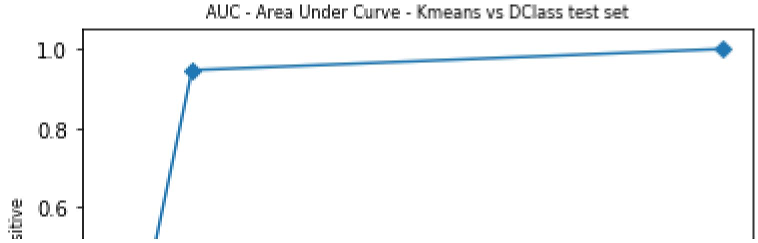
False Positive

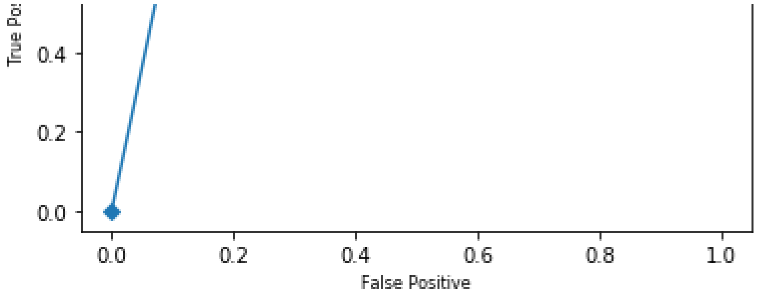


If you notice, the AUC is less for test data set, @ **0.916,** while training data set was **at 1**, meaning the train labels when compared to the actual data set , class variables (diagnosis) were exactly matching.

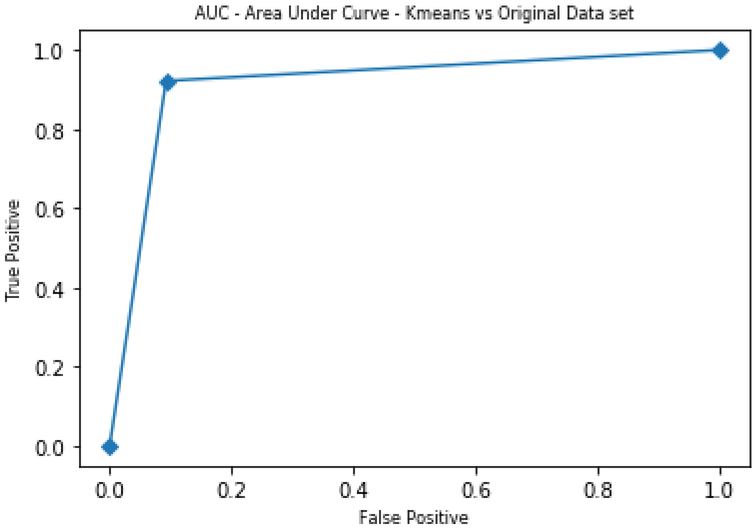
When comparing the AUC between kMeans outcome and train , test respectively,





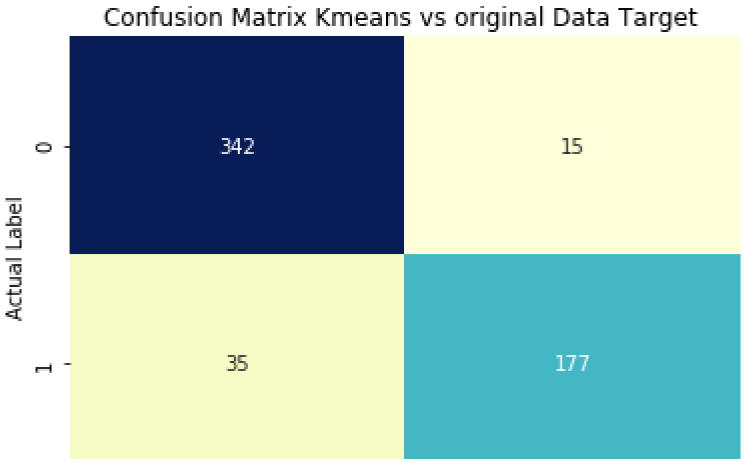


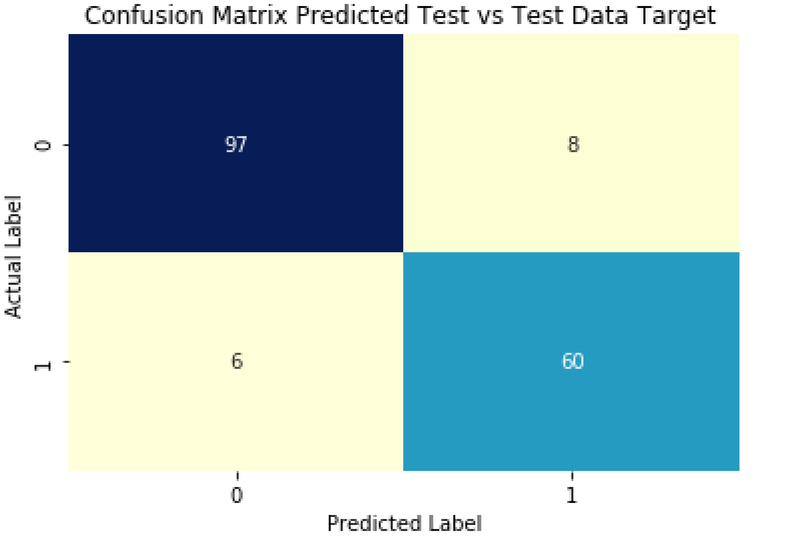
The AUC scores for Kmeans outcome (only training samples) compared to Train data set , **is 0.925 and for test it is 0.908 .**

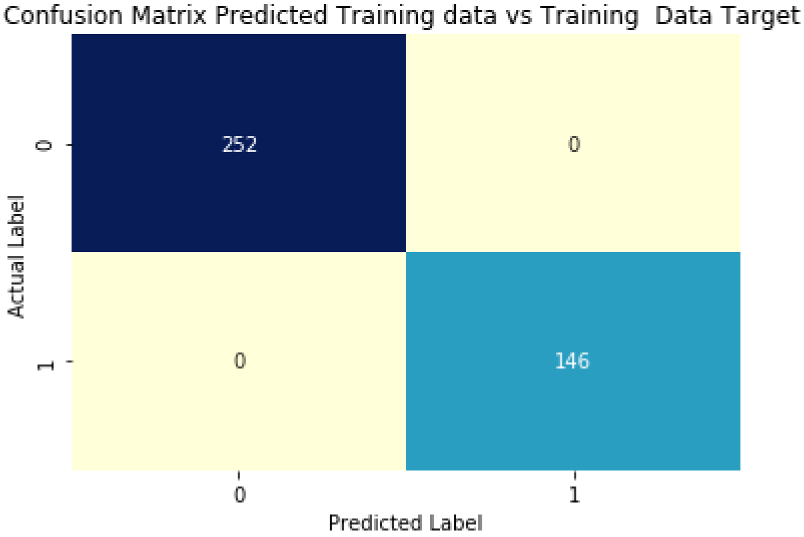


When comparing kMeans score with original data (Diagnosis) class labels, the above AUC indicates a high conformance between kMeans output and Diagnosis class (Benign — 0 , Malignant — 1) . The AUC score was also high **0.915.**

The following are the confusion matrix , for different scenarios,







The consolidated final data, with comparison between "Diagnosis" original target (class) variable, Kmeans clustering output, Training prediction, Test prediction , is available at www.kaggle.com/dataset/db445d1fbfe498ff91c204aafe9beac8b2afee3eaaf32fbaa2ae0832da791f7b

> *In conclusion, **KMeans clustering provides similar accuracy and fit , even though it is un-supervised learning,** when compared to Decisiontreeclassifier which is a supervised learning.*

Unsupervised Learning K Means Clustering Decision Tree Classifier Confusion Matrix