F1-scores and performance of three different clustering algorithms on Iris dataset

Krishna Mathur, https://ai-techsystems.com

Abstract—**This report applies machine learning algorithm to classify the data in the Iris dataset. Classification is done using clustering algorithms and further the results of these algorithms are compared on the basis of F1- score and accuracy.**

Index Terms— **Accuracy, Clustering algorithms, DBSCAN, F-1 score, Hierarchical clustering, Iris dataset, K-Means.**

I. Introduction

MMachine Learning is the field of study that gives computers the capability to learn without being explicitly programmed. Machine learning is a type of AI that allows computer programs to adjust when exposed to new data, in effect, "learning" without being explicitly programmed. Machine learning algorithms are often categorized as supervised or unsupervised.

1. Supervised machine learning – These algorithms can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training.
2. Unsupervised machine learning – These algorithms are used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system doesn't figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.

II. Clustering Algorithms

It is basically a type of unsupervised learning method . An unsupervised learning method is a method in which we draw references from datasets consisting of input data without labeled responses. Generally, it is used as a process to find meaningful structure, explanatory underlying processes, generative features, and groupings inherent in a set of examples.
Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups. It is basically a collection of objects on the basis of similarity and dissimilarity between them.

For Example - The data points in the graph below clustered together can be classified into one single group. We can distinguish the clusters, and we can identify that there are 3 clusters in the below picture.

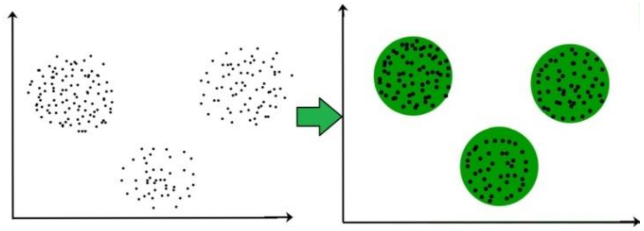


Fig.1

A. *K-means clustering algorithm* –

The algorithm begins by selecting k points as starting centroids ('centers' of clusters). We can just select any k random points, or we can use some other approach, but picking random points is a good start. Then, we iteratively repeat two steps:

1. Assignment step: each of m points from our dataset is assigned to a cluster that is represented by the closest of the k centroids. For each point, we calculate distances to each centroid, and simply pick the least distant one.
2. Update step: for each cluster, a new centroid is calculated as the mean of all points in the cluster. From the previous step, we have a set of points which are assigned to a cluster. Now, for each such set, we calculate a mean that we declare a new centroid of the cluster.

After each iteration, the centroids are slowly moving, and the total distance from each point to its assigned centroid gets lower and lower. The two steps are alternated until convergence, meaning until there are no more changes in cluster assignment. After a number of iterations, the same set of points will be assigned to each centroid.

B. *Hierarchical clustering*

The algorithm starts with all the data assigned to one of their own clusters and then joins the two most recent clusters to the same cluster. Finally, when there is only one cluster left, the algorithm ends. Hierarchical Clustering uses the distance based approach between the neighbor data-points for clustering. Each data point is linked to its nearest neighbors. Hierarchical clustering starts by treating each observation as a separate cluster. The main output of Hierarchical Clustering is a dendrogram, which shows the hierarchical relationship between the clusters. The Hierarchical clustering algorithm steps are -

1. In the initial step, we calculate the proximity of individual points and consider all the data points as individual clusters.
2. In step two, similar clusters are merged together and formed as a single cluster.
3. We again calculate the proximity of new clusters and merge the similar clusters to form new clusters. We repeat this step until only one cluster is left.

C. *DBSCAN*

DBSCAN stands for Density Based Spatial Clustering of Applications with Noise. Given that DBSCAN is a density based clustering algorithm, it does a great job of seeking areas in the data that have a high density of observations, versus areas of the data that are not very dense with observations. DBSCAN can sort data into clusters of varying shapes as well, another strong advantage.

DBSCAN Algorithm –

1- Find all the neighbor points within eps and identify the core points or visited with more than MinPts neighbors.

2- For each core point if it is not already assigned to a cluster, create a new cluster.

3- Find recursively all its density connected points and assign them to the same cluster as the core point.
A point a and b are said to be density connected if there exist a point c which has a sufficient number of points in its neighbors and both the points a and b are within the eps distance. This is a chaining process. So, if b is neighbor of c, c is neighbor of d, dis neighbor of e, which in turn is neighbor of a implies that b is neighbor of a.

4- Iterate through the remaining unvisited points in the dataset. Those points that do not belong to any cluster are noise.

III. Implementation

In order to implement the clustering algorithm on the Iris dataset we need to follow these steps –

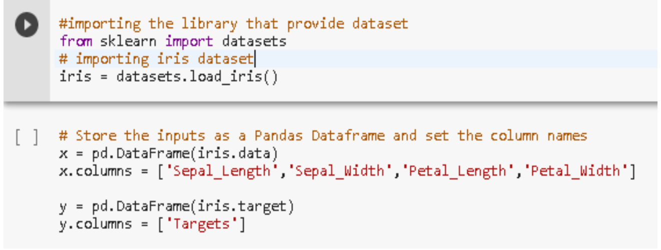
1- Load the Iris dataset

2- Create object of that clustering algorithm.

3- Fit the data into the object.

4- Evaluate the model.

We used Google colaboratory to build the model. Initially we load the iris dataset from Scikit learn library which is common step for all the algorithms and then store the data in the pandas dataframe.



```
#importing the library that provide dataset
from sklearn import datasets
# importing iris dataset
iris = datasets.load_iris()
```

```
[ ] # Store the inputs as a Pandas Dataframe and set the column names
    x = pd.DataFrame(iris.data)
    x.columns = ['Sepal_Length','Sepal_Width','Petal_Length','Petal_Width']

    y = pd.DataFrame(iris.target)
    y.columns = ['Targets']
```

Fig.2

In this plot we have Petal length on the x-axis and Petal width on the y-axis and the colors Red, Green and black belongs to 3 different types of iris i.e., Versicolor, Setosa, Virginica.

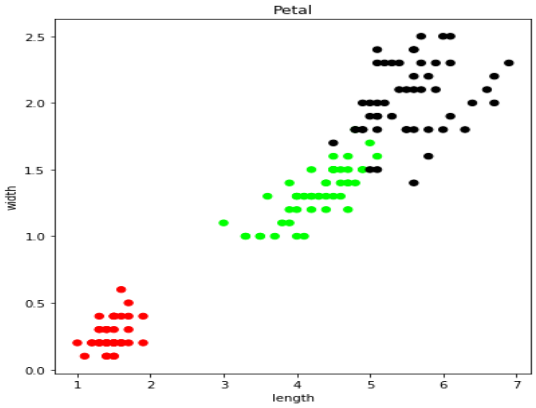
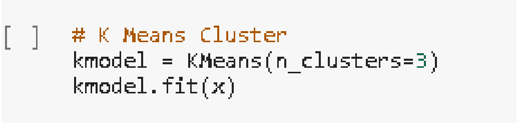


Fig.3

A. *K-Means model*

Creating the object and fitting data in the object.



```
[ ] # K Means Cluster
    kmodel = KMeans(n_clusters=3)
    kmodel.fit(x)
```

Fig.4

Plotting the graph predicted by K-means model –

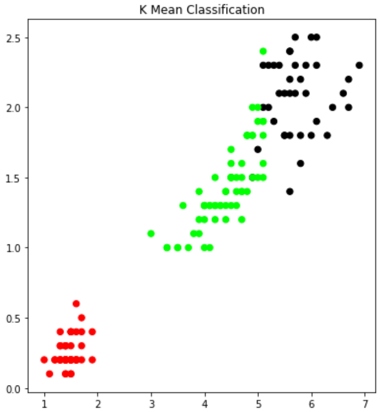
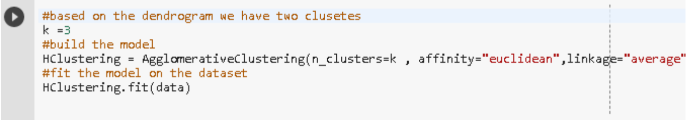


Fig.5

B. *Hierarchical model –*

We will first create a dendogram to see the clusters and then fir the number of cluster in model object.



```
#based on the dendrogram we have two clusetes
k =3
#build the model
HClustering = AgglomerativeClustering(n_clusters=k , affinity="euclidean",linkage="average"
#fit the model on the dataset
HClustering.fit(data)
```

Fig.6

Plotting the graph predicted by Hierarchical model –

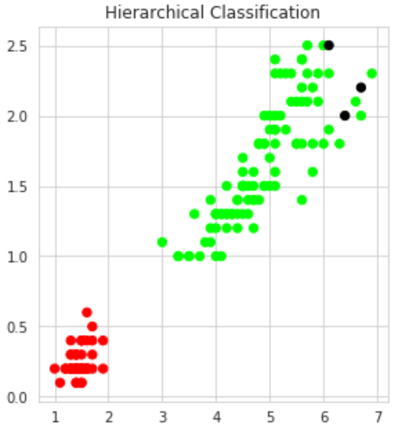
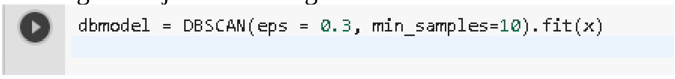


Fig.7

C. *DBSCAN model-*

Creating the object and fitting the data.



```
dbmodel = DBSCAN(eps = 0.3, min_samples=10).fit(x)
```

Fig.8

Plotting the graph predicted by DBSCAN –

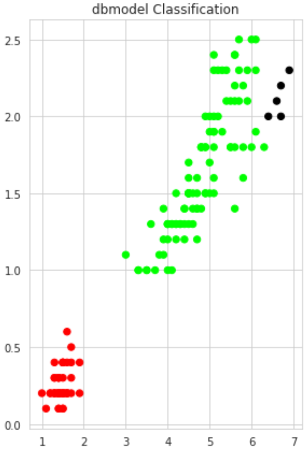
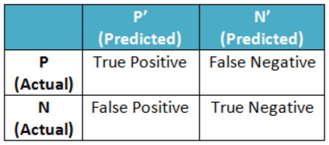


Fig.8

IV. Evaluation

We evaluate the model to check whether the model is properly working or not. And the other purpose to evaluate the model is to modify the model and to get the better result. In this paper we have used the classification Accuracy metrics and F-1 Scores. F1 score combines precision and recall relative to a specific positive class -The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst at 0. Confusion Matrix is a performance measurement for machine learning classification. It is a table with 4 different combinations of predicted and actual values.



	P' (Predicted)	N' (Predicted)
P (Actual)	True Positive	False Negative
N (Actual)	False Positive	True Negative

Fig.9

True Positive(TP) – You predicted Positive and it's True.
True Negative(TN) – You predicted Negative and it's True.
False Positive(FP) – You predicted Positive and it's False.
False Negative(FN) – You predicted Negative and it's False.

Accuracy = TP+TN/(TP+TN+FP+FN)
Recall = TP/(TP+FN)
Precision = TP/(TP+FP)
F-measure = 2*(Recall * Precision)/(Recall + Precision)

Plotting the graph to compare Accuracy of these models-

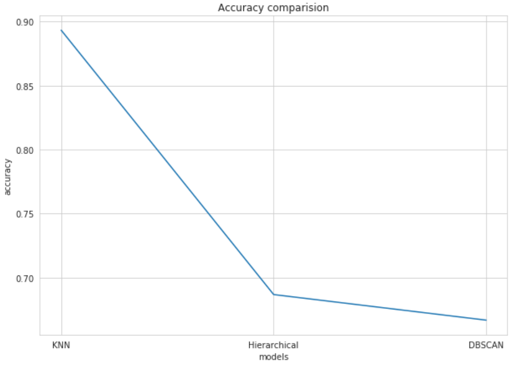


Fig.10

Plotting the Graph to compare F1-Score –

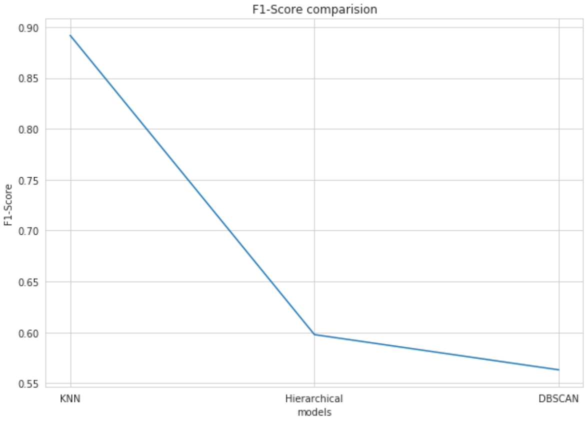


Fig.11

V. CONCLUSION

I tried to build the models that are able to recognize the iris species accurately on the basis of 3 classes, but some sample provides the misclassified result. All the models are able to Classify 2 Classes with higher percentage of accuracy. But the accuracy of KNN model is much better overall in terms of classifying all 3 classes and same in the case of F1-Score.