Machine Learning Autum-22
Assignment - 2
Report

Submitted by,
Aftab Hussain(22CS60R54)
Ritik Thool (22CS60R41)

Question 1:

Loading data
Loading the dataset from the file as a pandas DataFrame and also setting the column name as provided by the file “abalone.names”

Applying One-Hot-Encoding to categorical variable

The dataset has only one categorical variable named “Sex” and it can have three possible values {M, F, I} corresponds to male, female and infant.
Rings is the label for the instances, so we have dropped feature “Rings” from the dataset.

Since, we have to apply PCA on the dataset hence we have to have the feature as numerical variable.
After OHE of “Sex” the data look like,

	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Sex_F	Sex_I	Sex_M
0	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150	0	0	1
1	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.070	0	0	1
2	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210	1	0	0
3	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.155	0	0	1
4	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.055	0	1	0

Scaling of features

Scaling the feature will ensure that any particular feature shall not influence more than the other while doing the dimensionality reduction by Principal component analysis(PCA)
We have used the StandardScaler here, and after scaling the dataset look like,

	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Sex_F	Sex_I	Sex_M
0	-0.574558	-0.432149	-1.064424	-0.641898	-0.607685	-0.726212	-0.638217	-0.674834	-0.688018	1.316677
1	-1.448986	-1.439929	-1.183978	-1.230277	-1.170910	-1.205221	-1.212987	-0.674834	-0.688018	1.316677
2	0.050033	0.122130	-0.107991	-0.309469	-0.463500	-0.356690	-0.207139	1.481846	-0.688018	-0.759488
3	-0.699476	-0.432149	-0.347099	-0.637819	-0.648238	-0.607600	-0.602294	-0.674834	-0.688018	1.316677
4	-1.615544	-1.540707	-1.423087	-1.272086	-1.215968	-1.287337	-1.320757	-0.674834	1.453451	-0.759488

Applying PCA

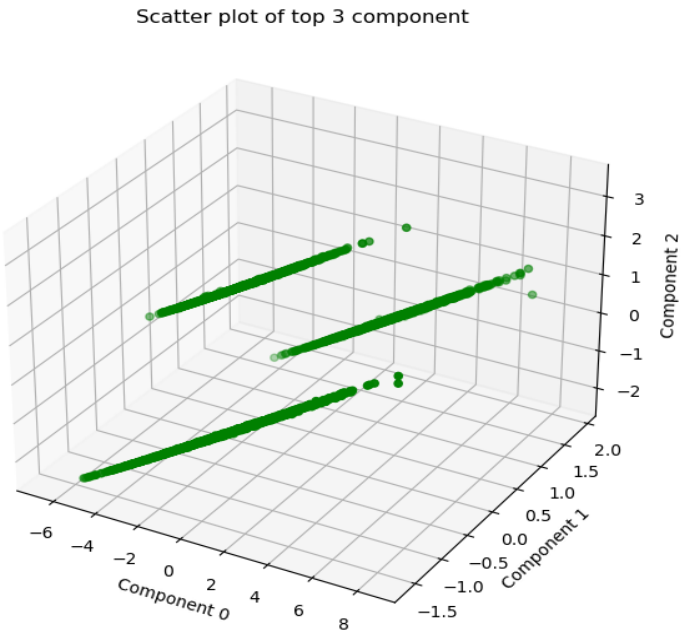
PCA is used to do dimensionality reduction. In our case we had initially 10 features including the One-Hot-Encoded features. It has been asked that we have to preserve 95% of the total variance. By following an iterative approach, we found out that reducing 10 components to 4 components preserves 96.3% of total variance.

After applying PCA, the reduced data look like,

:

	0	1	2	3
0	-1.442862	-1.523664	-1.151518	-0.412017
1	-2.971084	-1.549403	-1.652189	-0.143979
2	-0.165524	1.484398	-1.147274	0.162377
3	-1.207120	-1.509323	-1.102478	0.190170
4	-4.020928	0.184969	0.558237	-0.243602

And if we plot the top 3 components of the reduced 4 component, what we got is,



K-means clustering

Algorithm

1. We start with randomly selection of k centroids from the whole set of points
2. Create clusters w.r.t k centroids
3. Calculate new centers for the clusters.
4. Check if the consecutive centers have changed much or not.
5. If not, then repeat 2 - 5 otherwise break. The last calculated cluster is the final cluster.

The condition of convergence that has been followed in the implementation is that, we first see the minimum distance between any two current centroid of the clusters, say the minimum distance is *m,* and then check if all of the new centroid is changed at least by tolerance % of the *m*.

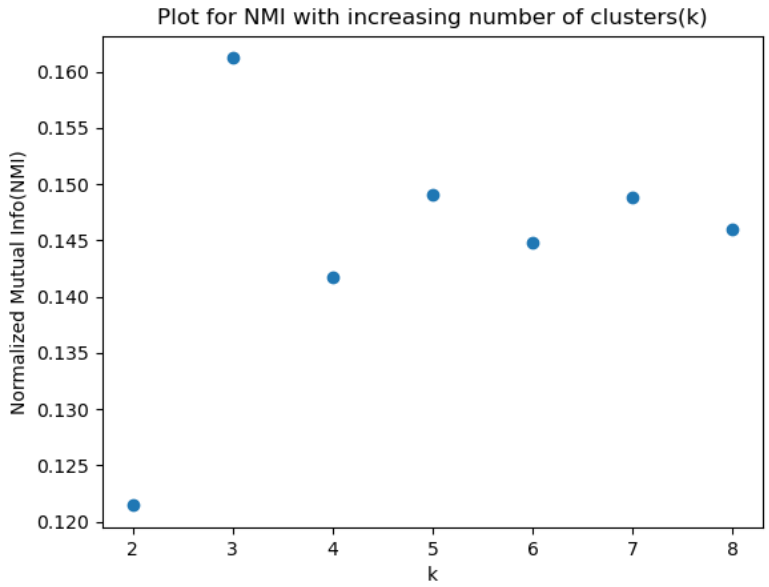
The gist of the algorithm can be seen by this code snippet

```
while(1):
    new_centeroids = get_new_centeroids(df, clusters, k)
    if is_converged(centeroids, new_centeroids, tol, k) == True:
        break

    new_clusters = get_new_clusters(df, new_centeroids, k)

    centeroids = new_centeroids
    clusters = new_clusters
```

After the Cluster has been formed, we assign labels to the clusters, and then we find the Normalized Mutual Information score by using sklearn metrics.
The function takes two arguments, first being the ground truth label of the initial points in the dataset and second being the cluster labels in which the points belongs in the final clustering.

Finally, we iteratively repeated this process with varying the value of k, that is the number of clusters and plotted k vs NMI in a scatter graph using matplotlib.

Result is a plot which look like,



From the PCA plot, the presence of 3 clusters was clear which has been justified by the results also.
For k = 3, the Normalized Mutual Information score having value 0.16122031364812683 is the maximum.
(The result might change with multiple runs of the program because we are selecting initial centroid randomly)