Machine Learning Assignment 2
M22MA003

Selecting the dataset :

Given a vector G = [v1 v2 v3 v4], where v1, v2, v3 and v4 can take either of 0|1 values.
IITJ roll number : M22MA003
The values v1 to v4 are as calculated:a
v1 = 1 if yy>= 20 -- 1
v2 v3 v4 is the binary equivalent of the mod operation of abc with 4, where v2 is the MSB and v4 is the LSB -- 011
vector G : 1011
The count of 1s in the vector G - 3
Taking 3rd dataset : Count = 3: Link to dataset

Task 1 a

The given dataset has 9 attributes.

At first, K-means clustering algorithm is applied using all the nine features but it cannot be shown in graphical format because of high dimensional structure.

1. Importing the input dataset.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

2. Standardizing the Input Dataset:

	0	1	2	3	4	5	6	7	8
0	0.639947	0.848324	0.149641	0.907270	-0.692891	0.204013	0.468492	1.425995	1.365896
1	-0.844885	-1.123396	-0.160546	0.530902	-0.692891	-0.684422	-0.365061	-0.190672	-0.732120
2	1.233880	1.943724	-0.263941	-1.288212	-0.692891	-1.103255	0.604397	-0.105584	1.365896
3	-0.844885	-0.998208	-0.160546	0.154533	0.123302	-0.494043	-0.920763	-1.041549	-0.732120
4	-1.141852	0.504055	-1.504687	0.907270	0.765836	1.409746	5.484909	-0.020496	1.365896
...	...	...	...	...	...	...	...	...	...
763	1.827813	-0.622642	0.356432	1.722735	0.870031	0.115169	-0.908682	2.532136	-0.732120
764	-0.547919	0.034598	0.046245	0.405445	-0.692891	0.610154	-0.398282	-0.531023	-0.732120
765	0.342981	0.003301	0.149641	0.154533	0.279594	-0.735190	-0.685193	-0.275760	-0.732120
766	-0.844885	0.159787	-0.470732	-1.288212	-0.692891	-0.240205	-0.371101	1.170732	1.365896
767	-0.844885	-0.873019	0.046245	0.656358	-0.692891	-0.202129	-0.473785	-0.871374	-0.732120

768 rows × 9 columns

3. The labels given by K-means clustering to each data row is saved into a dataframe. Below is the screenshot of the same: 768 data rows and their respective predicted labels or cluster number.

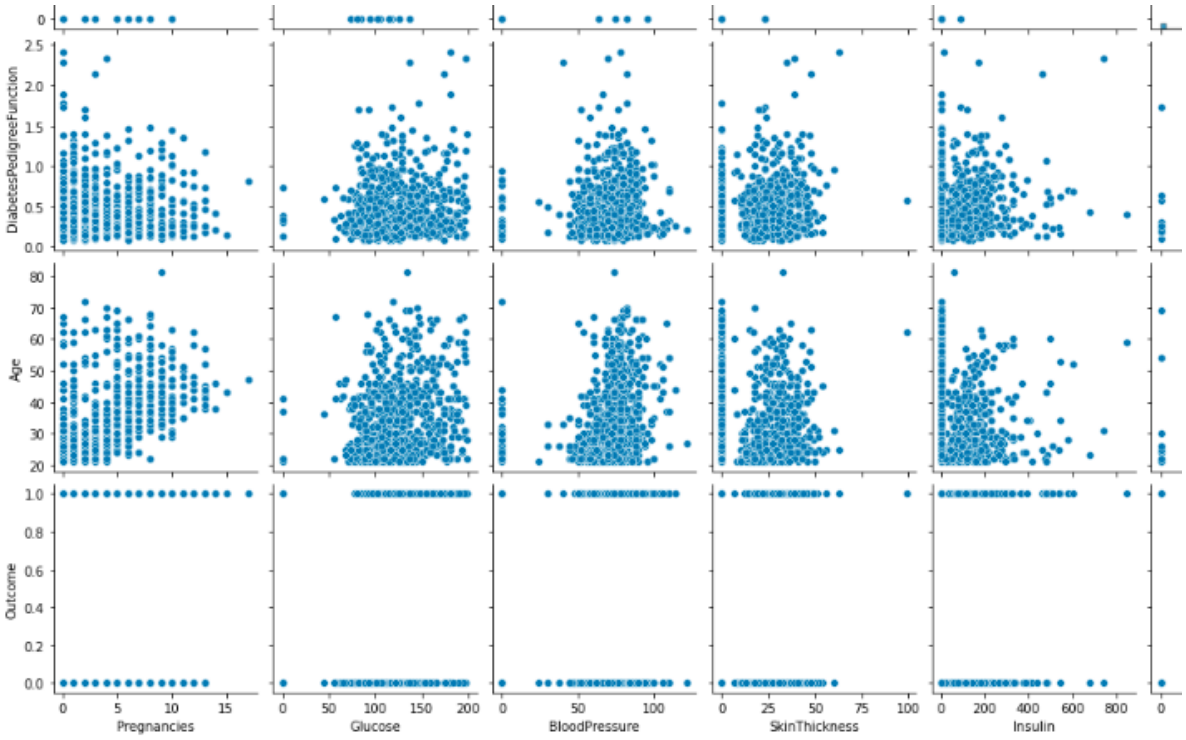
```
       0
0    0.0
1    1.0
2    0.0
3    1.0
4    0.0
..   ...
763  0.0
764  1.0
765  1.0
766  0.0
767  1.0

[768 rows x 1 columns]
```

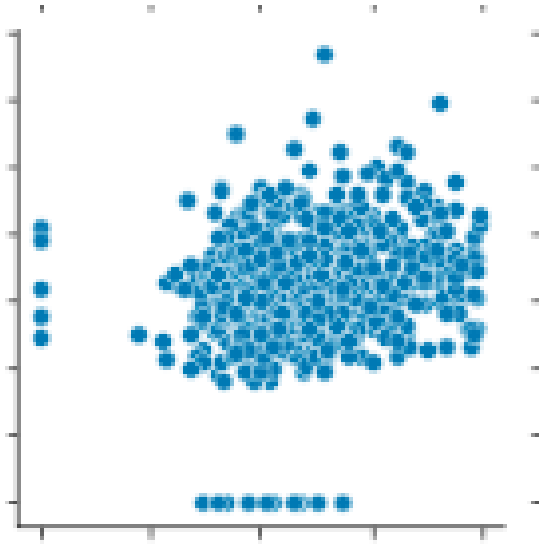
4. Here, first row indicates the index of 1st data row and the label provided to it is in class values [0,1] because The “Outcome” column from the given dataset also had two unique labels.

5. To display the K-means in graphical format, two features are to be chosen. To first display the unclustered data, pairplot is drawn between all possible pairs of given features.

6. Below is the screenshot examples for the same:

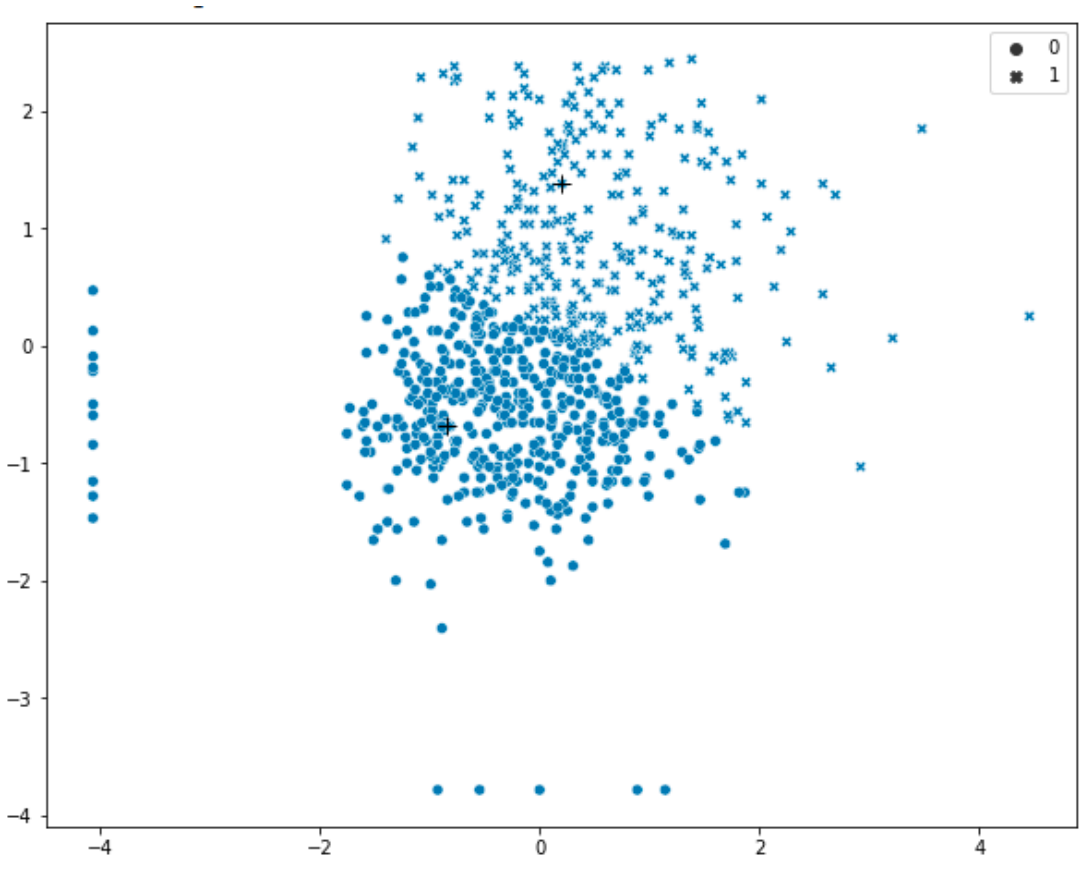


7. Selecting two features for K-mean clustering, the unclustered data looks like:

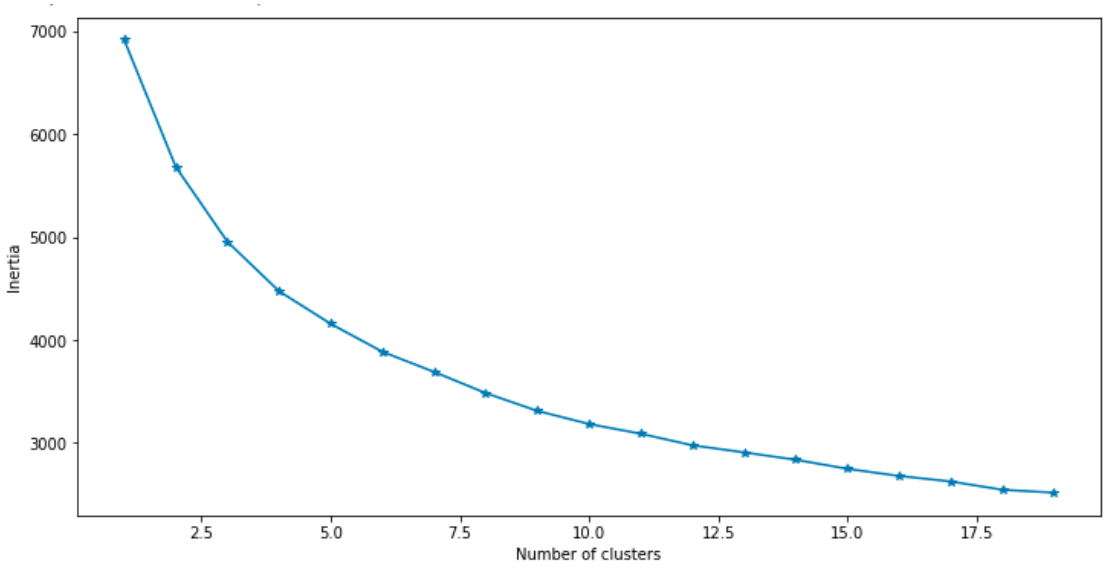


8. After applying K-Means algorithm, the clusters are formed:

0 cluster is represented using Cross symbol.
1 cluster is represented using Circle symbol.



9. Using Elbow Method to identify the optimal value of K.



10. Link to Colab File : [ML_ Assignment2_ Task1]

References: https://www.youtube.com/watch?v=vtuH4VRq1AU
https://towardsdatascience.com/visualizing-data-with-pair-plots-in-python-f228cf529166
https://www.youtube.com/watch?v=FqIGui0rwh4

Task 2 a:

PCA is applied on both Training dataset and Test dataset.

First Displaying Training Dataset

1. Imported the dataset

```
#Load the Dataset
train_data = pd.read_csv(io.BytesIO(uploaded['train.csv']))
#display the head (first 5 rows) of the dataset
train_data.head()
```

	tBodyAcc-mean()-X	tBodyAcc-mean()-Y	tBodyAcc-mean()-Z	tBodyAcc-std()-X	tBodyAcc-std()-Y	tBodyAcc-std()-Z	tBodyAcc-mad()-X	tBodyAcc-mad()-Y	tBodyAcc-mad()-Z	tBodyAcc-max()-X	...	fBodyBody(
0	0.288585	-0.020294	-0.132905	-0.995279	-0.983111	-0.913526	-0.995112	-0.983185	-0.923527	-0.934724	...	
1	0.278419	-0.016411	-0.123520	-0.998245	-0.975300	-0.960322	-0.998807	-0.974914	-0.957686	-0.943068	...	
2	0.279653	-0.019467	-0.113462	-0.995380	-0.967187	-0.978944	-0.996520	-0.963668	-0.977469	-0.938692	...	
3	0.279174	-0.026201	-0.123283	-0.996091	-0.983403	-0.990675	-0.997099	-0.982750	-0.989302	-0.938692	...	
4	0.276629	-0.016570	-0.115362	-0.998139	-0.980817	-0.990482	-0.998321	-0.979672	-0.990441	-0.942469	...	

5 rows × 563 columns

2. Scaling the dataset

```
Train Data is:
```

	0	1	2	3	4	5	6	7	8	9	...	552	553	554	555	556
0	0.200642	-0.063683	-0.419628	-0.868814	-0.939441	-0.737529	-0.859817	-0.939019	-0.766437	-0.856036	...	0.025960	-0.276399	-0.360603	0.062940	-0.778427
1	0.055948	0.031486	-0.253908	-0.875426	-0.923902	-0.849304	-0.868531	-0.921998	-0.848928	-0.871359	...	-0.897357	-0.767990	0.133011	-0.021461	-1.218805
2	0.073515	-0.043416	-0.076295	-0.869039	-0.907760	-0.893785	-0.863137	-0.898854	-0.896701	-0.863323	...	-0.260878	-0.438316	-0.377840	0.391976	0.151207
3	0.066696	-0.208422	-0.249712	-0.870626	-0.940022	-0.921805	-0.864503	-0.938124	-0.925279	-0.863323	...	0.591045	0.463155	-0.135025	-0.033637	1.037851
4	0.030469	0.027587	-0.109848	-0.875188	-0.934878	-0.921343	-0.867384	-0.931789	-0.928028	-0.870260	...	-0.138515	-0.240313	0.340406	0.268486	1.125918
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
7347	0.358361	-0.967904	-1.273005	0.913859	1.095963	1.628612	0.821169	1.174137	1.609686	1.247727	...	-0.876738	-0.829197	-0.591277	1.846034	0.325923
7348	-0.009044	0.243731	-0.676787	0.824887	1.026150	1.586100	0.726245	1.023755	1.658407	1.076279	...	-0.260847	-0.180290	0.166951	1.948561	-1.459501
7349	-0.015668	0.016781	1.132221	0.862975	0.810002	2.100249	0.768781	0.880813	2.266963	0.941403	...	1.034784	1.044548	0.131018	-0.599877	1.406760
7350	0.215866	-0.028123	-0.867710	0.860922	0.794902	2.086778	0.754697	0.944019	2.235301	1.047375	...	1.155541	0.913569	-0.326769	1.558312	1.525574
7351	1.096202	0.129199	-1.672681	0.749198	0.843051	1.868249	0.596891	1.004776	2.017853	1.354644	...	-0.249363	-0.375352	-0.857491	-0.022141	-0.106555

7352 rows × 562 columns

3. HeatMap to display correlation between all the 557 columns inside the dataset.

4. Taking n_components argument of PCA function as 2

```
PCA(n_components=2)
```

5. Transformed into components 1and 2

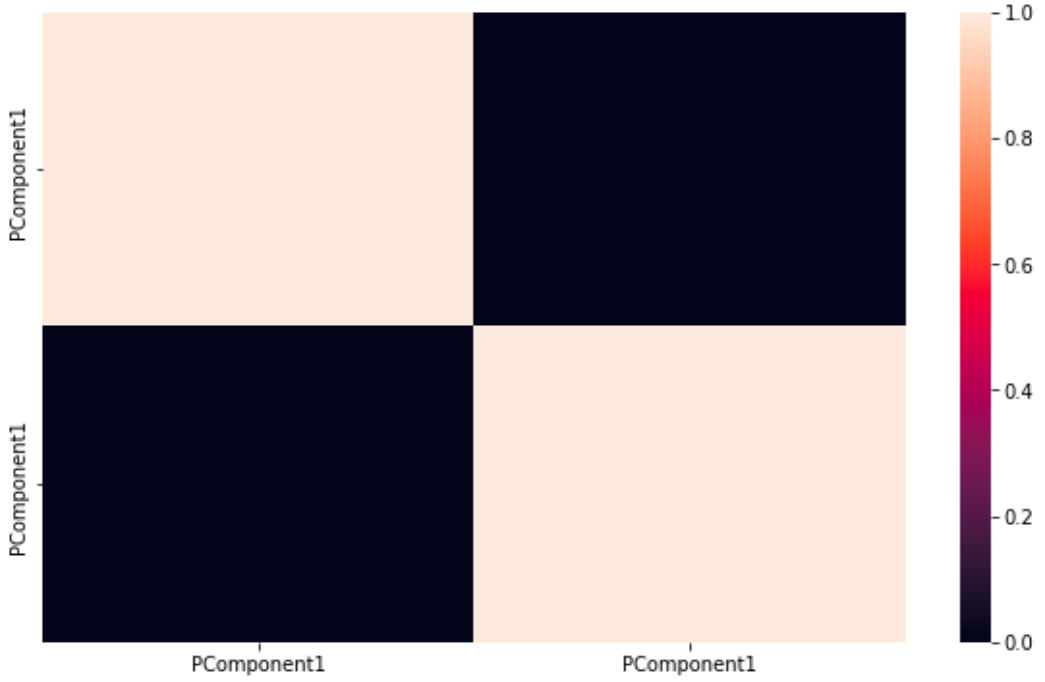
	PComponent1	PComponent1
0	-16.127876	2.165066
1	-15.285529	1.400438
2	-15.126334	2.486807
3	-15.340192	3.929089
4	-15.534146	4.611814

6. Checking the data size after transformation.

```
[ ]  len(data_train_pca)

     7352
```

7. Checking the correlation between the two components



8. Following the same steps for Test dataset:

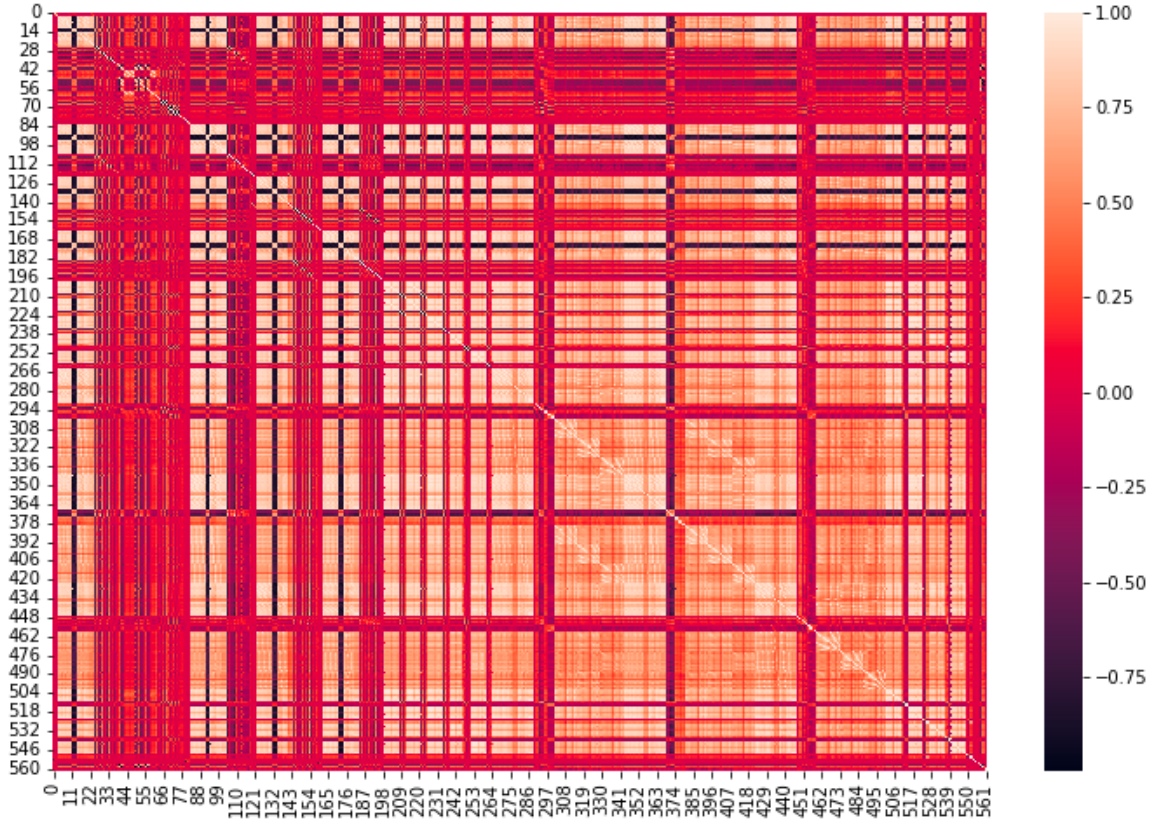
Imported the dataset and Scaling the dataset

Test Data is:

	0	1	2	3	4	5	6	7	8	9	...	552	553	554	555	556
0	-0.277708	-0.210631	2.193076	-0.787267	-0.833212	-0.093444	-0.808091	-0.839013	-0.104178	-0.824746	...	-0.166391	-0.344765	0.003566	0.357573	-1.363900
1	0.198660	0.182591	-0.250264	-0.876984	-0.929061	-0.858050	-0.897145	-0.928949	-0.863269	-0.824746	...	0.491021	0.012257	-0.264092	0.030787	-0.747232
2	0.024587	-0.318055	-0.228485	-0.921598	-0.934054	-0.907107	-0.916891	-0.933813	-0.912639	-0.910982	...	0.274821	-0.134989	-0.119671	0.446072	0.037919
3	-0.061057	-0.573037	-0.213709	-0.923837	-0.940817	-0.919082	-0.919152	-0.941599	-0.927776	-0.909740	...	-0.210678	-0.441714	-0.066443	0.338512	0.472695
4	0.013823	-0.387881	-0.494639	-0.921679	-0.929036	-0.949979	-0.916133	-0.923846	-0.951396	-0.909740	...	-0.810527	-0.796939	-0.022276	-0.098529	1.097348
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2942	0.597072	-1.380212	0.217067	0.789690	0.744135	1.154623	0.740507	0.779257	1.132204	1.235951	...	-0.311121	-0.488935	-1.019623	0.769650	1.330760
2943	1.476032	-0.829444	0.057817	0.747216	1.085580	1.206230	0.695181	1.026724	1.025466	1.235951	...	-0.135013	-0.326438	-2.207635	-0.846487	-1.098551
2944	1.254458	1.862441	-0.173180	0.688432	0.943345	1.057851	0.657591	0.811598	0.810778	0.896354	...	0.500453	0.423089	-0.555875	0.190504	1.034263
2945	-0.601088	1.411385	0.278129	0.704246	0.563664	1.175343	0.646262	0.506579	0.971675	0.896354	...	0.227458	-0.061034	1.307070	-1.849401	1.400679
2946	-1.987586	-0.022264	-0.669914	0.687443	0.633521	1.294567	0.546054	0.634369	1.138143	0.669183	...	0.647419	0.520325	1.766026	-0.655615	1.316783

2947 rows × 562 columns

HeatMap to display correlation between all the 557 columns inside the dataset.



Taking n_components argument of PCA function as 2

```
PCA(n_components=2)
```

Transformed into components 1and 2

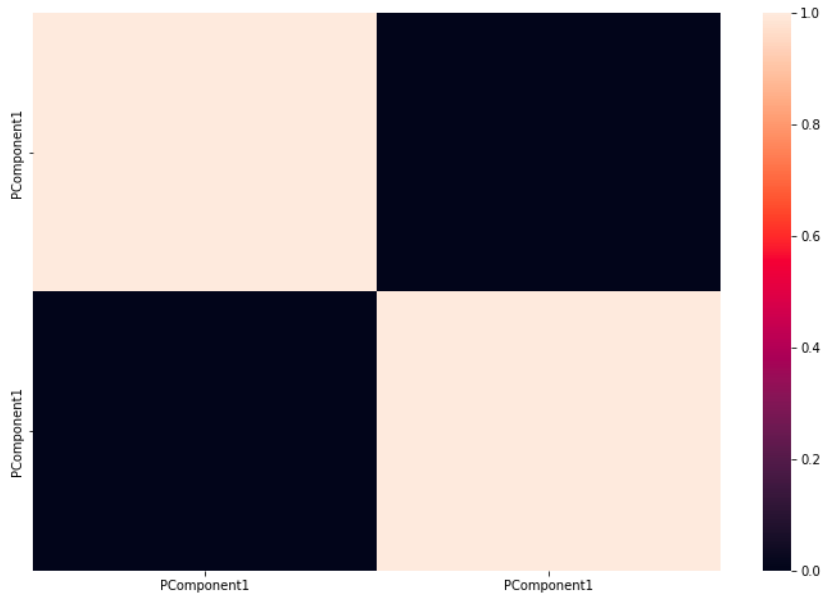
	PComponent1	PComponent1
0	-10.885514	-2.516958
1	-14.132175	-1.361279
2	-15.411709	1.992661
3	-15.673719	1.309247
4	-14.882018	-2.341210

Checking the data size after transformation.

```
[13] len(data_test_pca)

     2947
```

Checking the correlation between the two components



Task 2 b:

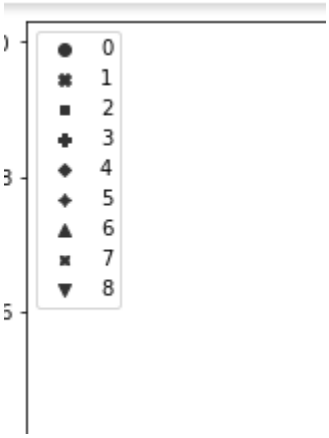
Applying K-Means algorithm to the training dataset using the PComponent1 and PComponent2

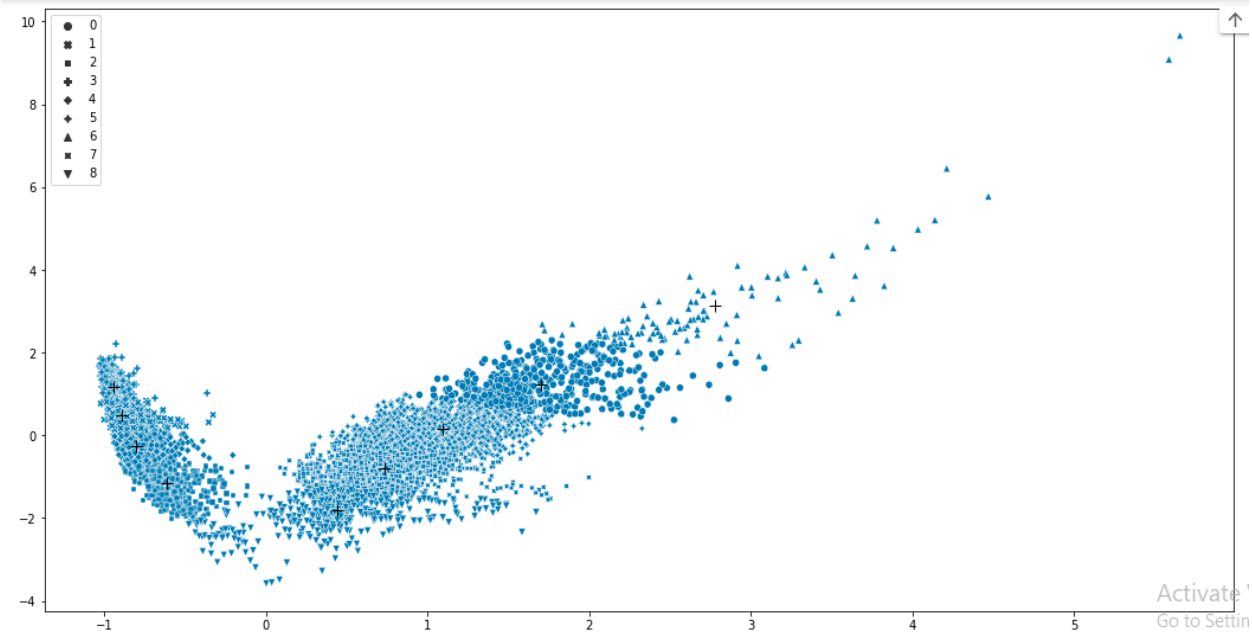
	PComponent1	PComponent1
0	-16.127876	2.165066
1	-15.285529	1.400438
2	-15.126334	2.486807
3	-15.340192	3.929089
4	-15.534146	4.611814

Cardinality of the target label is 9.

Taking Clusters count as 9.

➔ Each cluster is represented by a different symbol as mentioned in the plot for respective cluster count.





Colab Link : [ML_Assignment2_Task2.ipynb]

References:
https://www.youtube.com/watch?v=83x5X66uWK0
https://www.youtube.com/watch?v=o0NNUeWNnL4