

▼ ASSIGNMENT NO.5

NAME-: MORE PANKAJ SAMPAT

ROLL NO-: 054

CLASS-: TE COMPUTER

ACADEMIC YEAR-: 2022-23

SUB-: AIML HONOUR COURSE

Problem Statement-: Perform clustering of the iris dataset based on all variables using Gaussian mixture models. Use PCA to visualize clusters.

```
1 import numpy as np
2 import pandas as pd
3 import seaborn as sns
4 import matplotlib.pyplot as plt
5 from sklearn.preprocessing import StandardScaler
6 from sklearn.mixture import GaussianMixture
7 from sklearn.metrics.cluster import adjusted_rand_score
```

```
1 from google.colab import files
2 uploaded = files.upload()
3
```

Choose Files iris.csv

- **iris.csv**(text/csv) - 3975 bytes, last modified: 2/6/2014 - 100% done
Saving iris.csv to iris.csv

▼ Importing and reading dataset

```
1 import pandas as pd
2 import io
3
4 irisdata = pd.read_csv(io.BytesIO(uploaded['iris.csv']))
```

```
1 x= irisdata.iloc[:,4]
2 y= irisdata.iloc[:,-1]
```

▼ Standardizing Dataset using sklearn

```
1 sc =StandardScaler()
2 sc.fit(x)
3 std_array =sc.transform(x)
4 X = pd.DataFrame(std_array,columns = x.columns)
```

▼ Gaussian Mixture model

```
1 cluster =GaussianMixture(n_components=3)
2 cluster.fit(X)
3 y_pred =cluster.predict(X)
4 score = adjusted_rand_score(y,y_pred)
5 score
```

0.9038742317748124

▼ Using PCA to visualize data

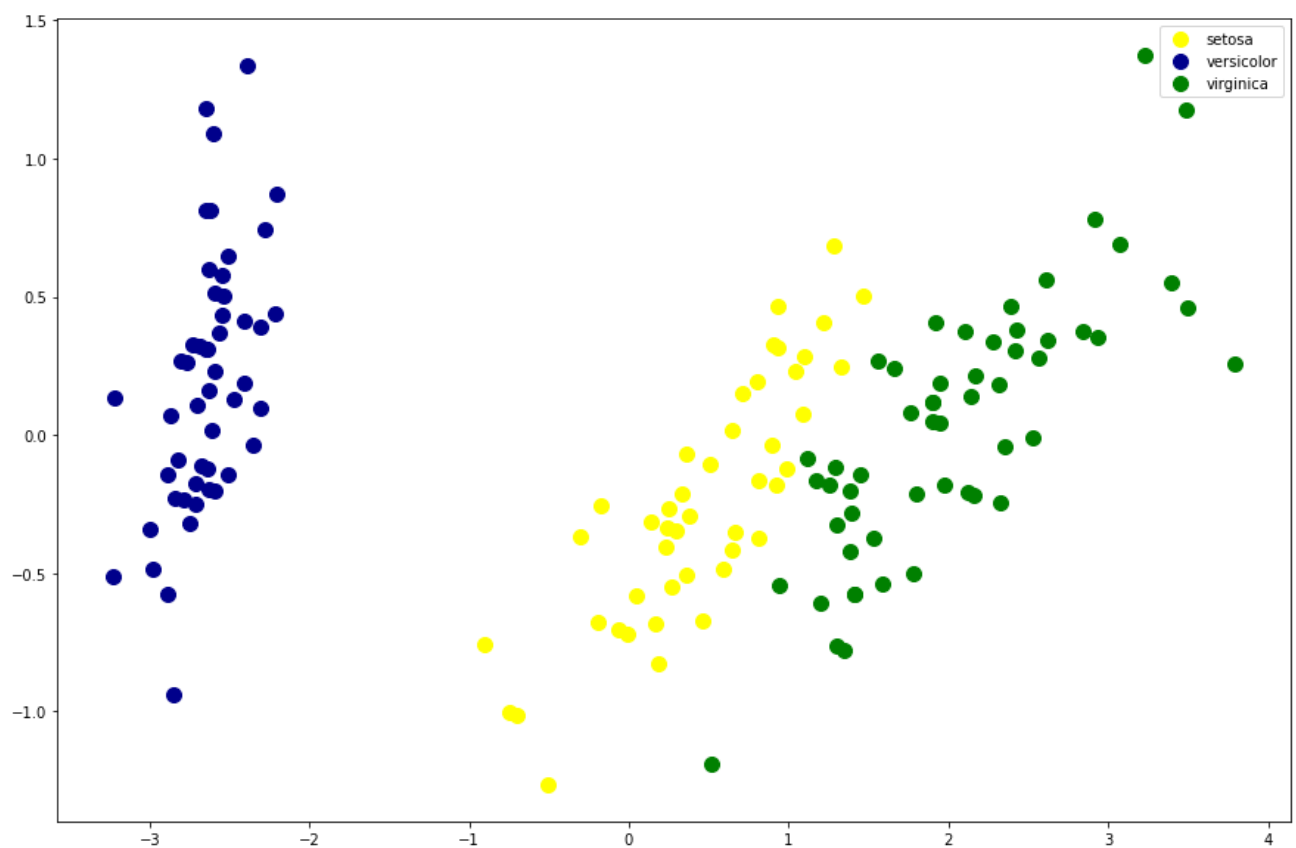
```
1 from sklearn.decomposition import PCA
2
3 pca =PCA(n_components=2)
4 pca_array =pca.fit_transform(irisdata.drop(['variety'],axis=1))
5 pca_df =pd.DataFrame(pca_array,columns=["PC1", "PC2"])
6 pca_df.head()
```

	PC1	PC2
0	-2.684126	0.319397
1	-2.714142	-0.177001
2	-2.888991	-0.144949
3	-2.745343	-0.318299
4	-2.728717	0.326755

```
1 col_code = {0:"yellow",1:"darkblue",2:"green"}
2 label = {0:"setosa",1:"versicolor",2:"virginica"}
3
4 pca_df["labels"]= pd.DataFrame(y_pred)
5 groups = pca_df.groupby('labels')
6
7 # Grouping instances based on species
8 groups.mean()
```

	PC1	PC2	
labels			
0	0.452518	-0.248189	
1	-2.642415	0.190885	
2	2.031954	0.029531	

```
1 fig, ax =plt.subplots(1,1,figsize =(15,10))
2 for name, group in groups:
3     ax.plot(group.PC1,group.PC2,color =col_code[name],label =label[name],marker='o',lin
4 ax.legend()
5 plt.show()
```



[Colab paid products](#) - [Cancel contracts here](#)

✓ 0s completed at 4:41 PM

