# **Machine Learning Assignment -5**

Name : Sachit Kumar Tadishetty

700# : 700734682 UCM ID : SXT46820

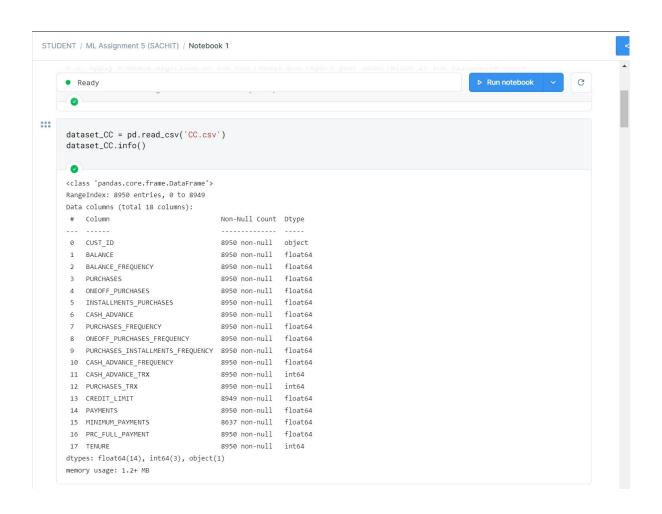
### VideoLink:

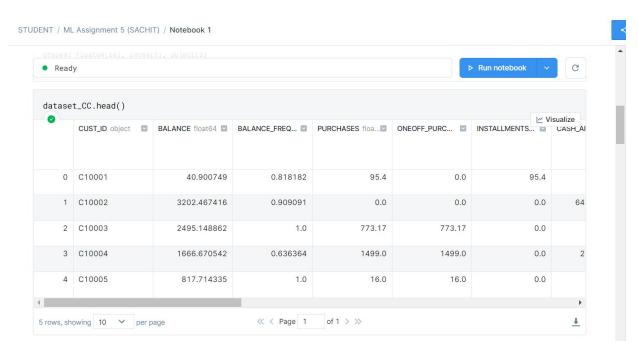
https://drive.google.com/drive/folders/1kj57lrUnTj2skKkZDH7UgScC-2lM5PB3?usp=share link

## Github Link: https://github.com/sachit46820/ML-Assignment

- 1) Principal Component Analysis
  - a. Apply PCA on CC dataset.
  - b. Apply k-means algorithm on the PCA result and report your observation if the silhouette score has improved or not?
  - c. Perform Scaling+PCA+K-Means and report performance.

```
STUDENT / ML Assignment 5 (SACHIT) / Notebook 1
    Ready
     # importing required libraries for assignment 5 here
    import numpy as np
     import matplotlib.pyplot as plt
    import pandas as pd
    import seaborn as sns
     from sklearn import preprocessing, metrics
     from sklearn.preprocessing import StandardScaler, LabelEncoder
    from sklearn.model_selection import train_test_split
     from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
     from sklearn.decomposition import PCA
     from sklearn.cluster import KMeans
     sns.set(style="white", color_codes=True)
     import warnings
     warnings.filterwarnings("ignore")
    # Principal Component Analysis
    # a. Apply PCA on CC dataset.
    # b. Apply k-means algorithm on the PCA result and report your observation if the silhouette score
    # has improved or not?
     # c. Perform Scaling+PCA+K-Means and report performance.
```





STUDENT / ML Assignment 5 (SACHIT) / Notebook 1 Ready C ... dataset\_CC.isnull().any() 0 CUST\_ID False BALANCE False BALANCE\_FREQUENCY False PURCHASES False ONEOFF\_PURCHASES False INSTALLMENTS\_PURCHASES False CASH\_ADVANCE False

False

False

False

False

False

False

True

False

True

False

False

PURCHASES\_FREQUENCY

CASH\_ADVANCE\_TRX

MINIMUM\_PAYMENTS

PRC\_FULL\_PAYMENT

PURCHASES\_TRX

CREDIT\_LIMIT

PAYMENTS

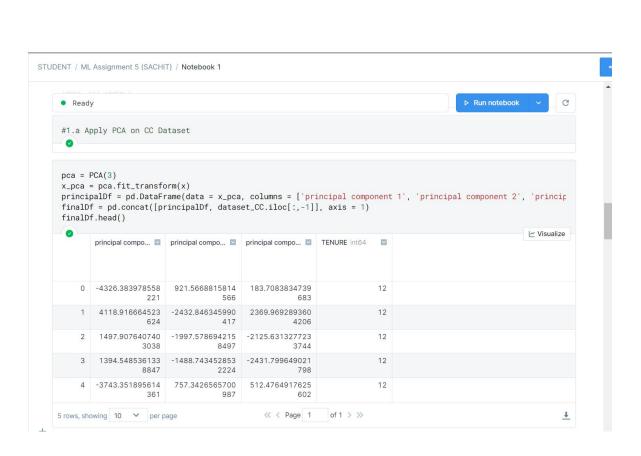
TENURE

dtype: bool

ONEOFF\_PURCHASES\_FREQUENCY

CASH\_ADVANCE\_FREQUENCY

PURCHASES\_INSTALLMENTS\_FREQUENCY



```
Ready
     #1.b Apply K Means on PCA Result
X = finalDf.iloc[:,0:-1]
      y = finalDf.iloc[:,-1]
***
      nclusters = 3 # this is the k in kmeans
      km = KMeans(n_clusters=nclusters)
      km.fit(X)
      # predict the cluster for each data point
      y_cluster_kmeans = km.predict(X)
      # Summary of the predictions made by the classifier
      print(classification_report(y, y_cluster_kmeans, zero_division=1))
print(confusion_matrix(y, y_cluster_kmeans))
      train_accuracy = accuracy_score(y, y_cluster_kmeans)
print("\nAccuracy for our Training dataset with PCA:", train_accuracy)
      #Calculate sihouette Score
      score = metrics.silhouette_score(X, y_cluster_kmeans)
      print("Sihouette Score: ",score)
      Sinouette Score- ranges from -1 to +1 , a high value indicates that the object is well matched to its own clus
```

```
STUDENT / ML Assignment 5 (SACHIT) / Notebook 1
    Ready
                                                                                                                      C
     #1.c Scaling +PCA + KMeans
     x = dataset_CC.iloc[:,1:-1]
y = dataset_CC.iloc[:,-1]
     print(x.shape,y.shape)
      0
     (8950, 16) (8950,)
000
     X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.34,random_state=0)
     \# this is the k in kmeans
     km = KMeans(n_clusters=nclusters)
     km.fit(X_train,y_train)
     # predict the cluster for each training data point
     y_clus_train = km.predict(X_train)
     # Summary of the predictions made by the classifier
     print(classification_report(y_train, y_clus_train, zero_division=1))
     print(confusion_matrix(y_train, y_clus_train))
     train_accuracy = accuracy_score(y_train, y_clus_train)
     print("Accuracy for our Training dataset with PCA:", train_accuracy)
     #Calculate sihouette Score
     score = metrics.silhouette_score(X_train, y_clus_train)
     print("Sihouette Score: ",score)
     Sihouette Score- ranges from -1 to +1, a high value indicates that the object is well matched to its own clus
```

```
Ready
           precision recall f1-score support
               0.00
                       1.00
                                 0.00
                                          0.0
                0.00
                        1.00
                                 0.00
                                          0.0
               1.00
                                        139.0
                        0.00
                                0.00
               1.00
                        0.00
                                0.00
                                        135.0
         8
               1.00
                        0.00
                                0.00
                                        128.0
         9
               1.00
                        0.00
                                0.00
                                        118.0
        10
               1.00
                       0.00
                                0.00
                                        151.0
        11
               1.00
                        0.00
                                 0.00
                                        262.0
              1.00
                                0.00
                                       4974.0
   accuracy
                                       5907.0
                                0.00
                        0.30
  macro avg
               0.70
                                0.00
                                       5907.0
                                       5907.0
weighted avg
               1.00
                        0.00
                                 0.00
       0
                0
                                     0
                                          0]
   0 123 16
                                          0]
   0 126
           9
                0
                    0
                         0
                             0
                                 0
                                     0
                                          01
   0 110 18
                         0
                             0
                                          0]
   0 106 12
                0
                    0
                         0
                             9
                                 0
   1 121 29
                     0
                         0
                             0
   3 211 48
                                          0]]
Accuracy for our Training dataset with PCA: 0.0
Sihouette Score: 0.5216744364662849
'\nSihouette Score- ranges from -1 to +1 , a high value indicates that the object is well matched to its own cluster 🕏
```

#### STUDENT / ML Assignment 5 (SACHIT) / Notebook 1

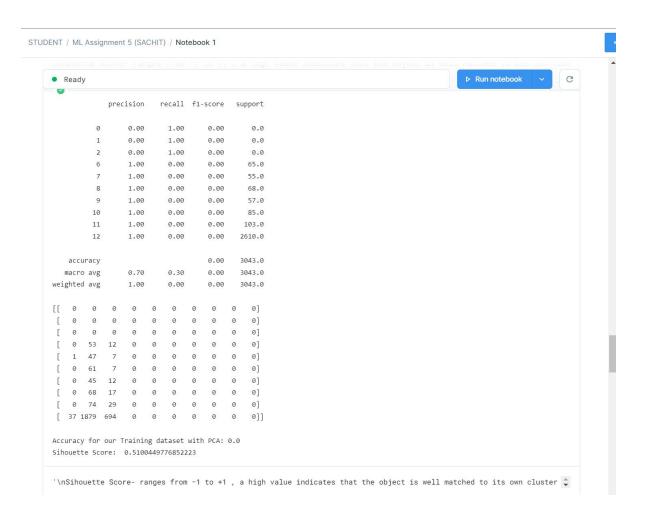
```
Provided the cluster for each testing data point
y_clus_test = km.predict(X_test)

# Summary of the predictions made by the classifier
print(classification_report(y_test, y_clus_test, zero_division=1))
print(confusion_matrix(y_test, y_clus_test))

train_accuracy = accuracy_score(y_test, y_clus_test)
print("\nAccuracy for our Training dataset with PCA:", train_accuracy)

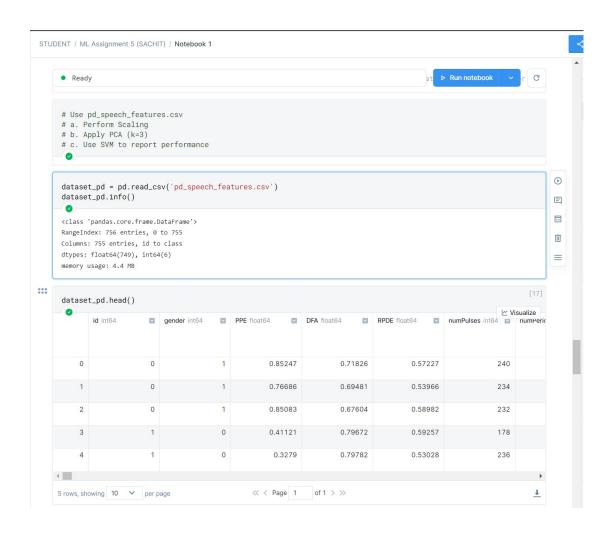
#Calculate sihouette Score
score = metrics.silhouette_score(X_test, y_clus_test)
print("Sihouette Score: ",score)
"""

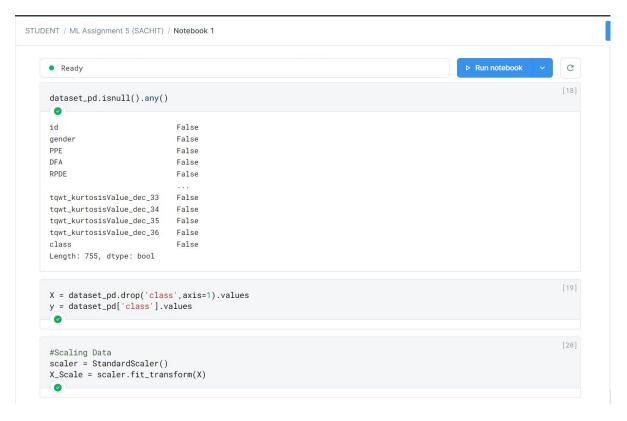
Sihouette Score- ranges from -1 to +1 , a high value indicates that the object is well matched to its own clus
```

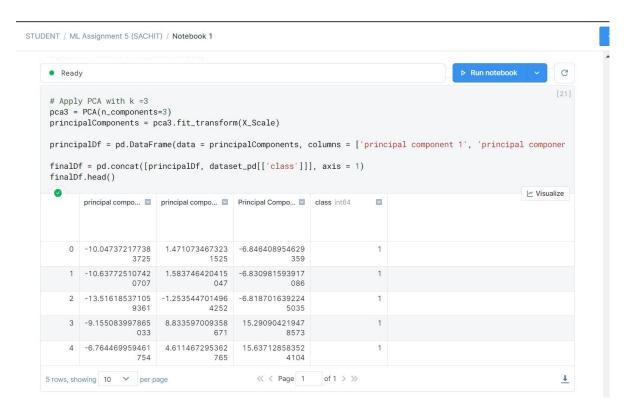


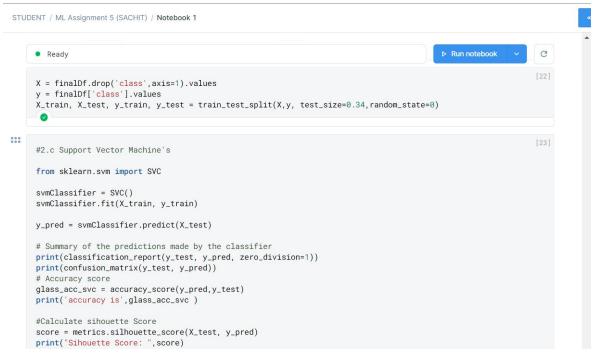
# 2. Use pd speech features.csv

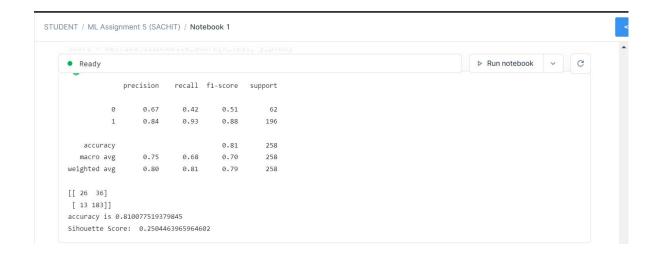
- a. Perform Scaling.
- b. Apply PCA (k=3).
- c. Use SVM to report performance.



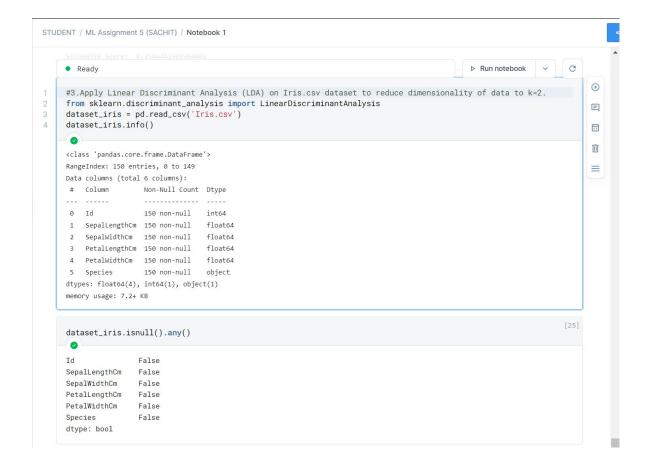


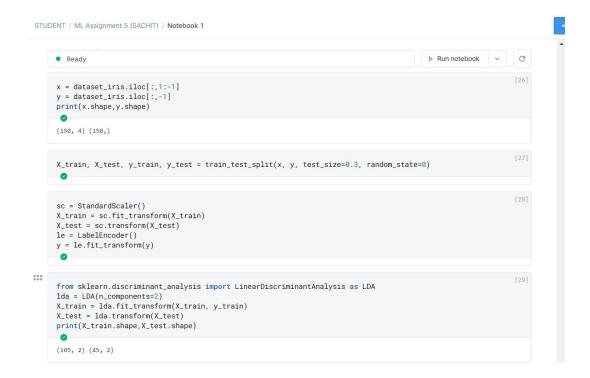




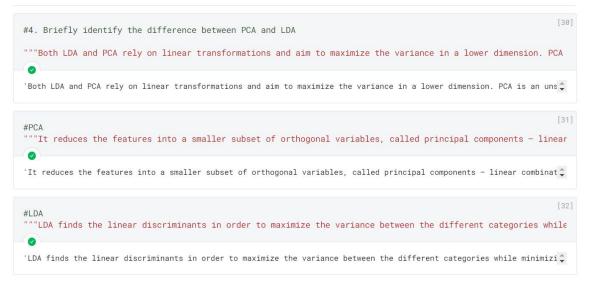


3. Apply Linear Discriminant Analysis (LDA) on Iris.csv dataset to reduce dimensionality of data to k=2.





4. Briefly identify the difference between PCA and LDA.



#### PCA:

- Principal component analysis (PCA) is surely the most known and simple unsupervised dimensionality reduction method.
- The first component captures the largest variability of the data, while the second captures the second largest, and so on.

#### LDA:

- Linear discriminant analysis (LDA) is a supervised machine learning and linear algebra approach for dimensionality reduction.
- LDA finds the linear discriminants in order to maximize the variance between the different categories while minimizing the variance within the class.