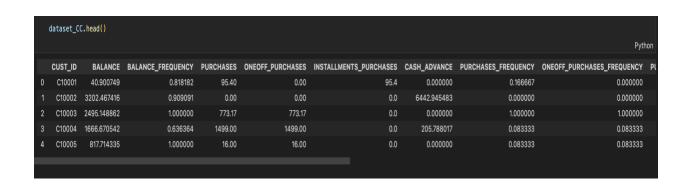
Assignment 5

• Importing all the required libraries for the Assignments:

```
dataset CC = pd.read csv('datasets//CC.csv')
   dataset_CC.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8950 entries, 0 to 8949
Data columns (total 18 columns):
    Column
                                     Non-Null Count Dtype
0
    CUST_ID
                                     8950 non-null object
1
    BALANCE
                                     8950 non-null float64
                                     8950 non-null float64
2
    BALANCE_FREQUENCY
3
    PURCHASES
                                     8950 non-null float64
4
    ONEOFF_PURCHASES
                                     8950 non-null float64
5
    INSTALLMENTS_PURCHASES
                                     8950 non-null float64
    CASH_ADVANCE
                                     8950 non-null float64
    PURCHASES_FREQUENCY
                                     8950 non-null float64
8
   ONEOFF_PURCHASES_FREQUENCY
                                     8950 non-null float64
    PURCHASES_INSTALLMENTS_FREQUENCY 8950 non-null
                                                     float64
9
10 CASH_ADVANCE_FREQUENCY
                                     8950 non-null float64
11 CASH_ADVANCE_TRX
                                     8950 non-null
                                                     int64
12 PURCHASES_TRX
                                     8950 non-null int64
13 CREDIT_LIMIT
                                     8949 non-null float64
14 PAYMENTS
                                     8950 non-null float64
                                     8637 non-null
15 MINIMUM_PAYMENTS
                                                     float64
16 PRC_FULL_PAYMENT
                                     8950 non-null
                                                     float64
17 TENURE
                                     8950 non-null
                                                     int64
dtypes: float64(14), int64(3), object(1)
memory usage: 1.2+ MB
```

• Read the datasets given and getting the info about the datasets



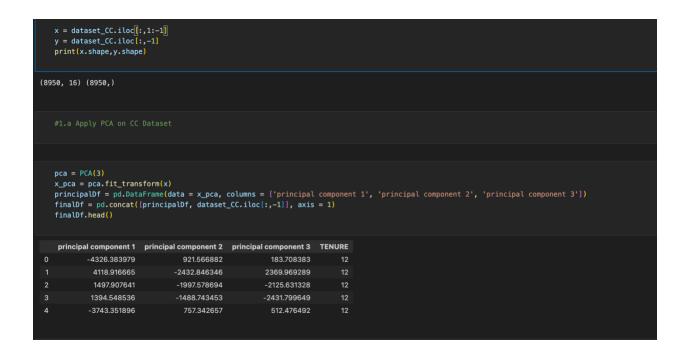
• By using the head() function we successfully printed the n number of rows for the corresponding datasets

```
dataset_CC.isnull().any()
CUST_ID
                                     False
BALANCE
                                     False
BALANCE_FREQUENCY
                                     False
PURCHASES
                                     False
ONEOFF_PURCHASES
                                     False
INSTALLMENTS_PURCHASES
                                     False
CASH_ADVANCE
                                     False
PURCHASES_FREQUENCY
                                     False
ONEOFF_PURCHASES_FREQUENCY
                                     False
PURCHASES_INSTALLMENTS_FREQUENCY
                                     False
CASH_ADVANCE_FREQUENCY
                                     False
CASH_ADVANCE_TRX
                                     False
PURCHASES_TRX
                                     False
                                      True
CREDIT_LIMIT
PAYMENTS
                                     False
MINIMUM_PAYMENTS
                                     True
PRC_FULL_PAYMENT
                                     False
TENURE
                                     False
dtype: bool
```

• Using the isnull() function to check whether we have any null values in it or not.

```
dataset_CC.fillna(dataset_CC.mean(), inplace=True)
   dataset_CC.isnull().any()
CUST_ID
                                     False
BALANCE
                                     False
BALANCE_FREQUENCY
                                     False
PURCHASES
                                     False
ONEOFF_PURCHASES
                                     False
INSTALLMENTS_PURCHASES
                                     False
CASH_ADVANCE
                                     False
PURCHASES_FREQUENCY
                                     False
ONEOFF_PURCHASES_FREQUENCY
                                     False
PURCHASES_INSTALLMENTS_FREQUENCY
                                     False
CASH_ADVANCE_FREQUENCY
                                     False
CASH_ADVANCE_TRX
                                     False
PURCHASES_TRX
                                     False
CREDIT_LIMIT
                                     False
PAYMENTS
                                     False
                                     False
MINIMUM_PAYMENTS
PRC_FULL_PAYMENT
                                     False
TENURE
                                     False
dtype: bool
```

• Using the fillna() function, it will fill out all the NA values to the values mentioned. i.e., mean



Apply the PCA to the CC dataset and the result is shown below:

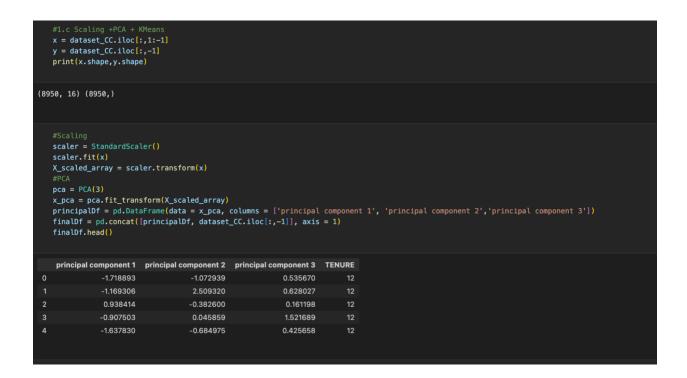
• The number of components that is given is three.

```
X = final(f.iloc(s,0:-1)
y = final(f.iloc(s,0:
```

 Applying the K- means in the PCA results we got and to check the accuracy and the silhouette score and it is taken as given.

		pre	cision		recall	f1-:	score	su	pport	
	0		0.00		1.00		0.00		0.0	
	1		0.00		1.00		0.00		0.0	
	2		0.00		1.00		0.00		0.0	
	6		1.00		0.00		0.00		204.0	
	7		1.00		0.00		0.00		190.0	
	8		1.00		0.00		0.00		196.0	
	9		1.00		0.00		0.00		175.0	
	10		1.00 1.00		0.00		0.00	236.0		
	11				0.00		0.00	365.0		
	12		1.00		0.00		0.00	7	584.0	
accı	uracy						0.00	8	950.0	
macro	o avg		0.70		0.30		0.00 8950.0			
weighted	d avg		1.00		0.00		0.00	950.0		
0]]	0	0	0	0	0	0	0	0	0]	
[0	0	0	0	0	0	0	0	0	0]	
[0	0	0	0	0	0	0	0	0	0]	
[175	1	28	0	0	0	0	0	0	0]	
[173	2	15	0	0	0	0	0	0	0]	
[169	0	27	0	0	0	0	0	0	0]	
[149	0	26	0	0	0	0	0	0	0]	
[188	1	47	0	0	0	0	0	0	0]	
[5389	126 2	069	0	0	0	0	0	0	0]]	
Accuracy	y for	our	Trainin	g da	ataset	with	PCA:	0.0		
Sihouet	te Sco	re:	0.5109	307	2743194	68				

• Result of Silhouette score is 0.51.



Applying scaling and PCA as given in the question.

• StandardScaler() Standardize features by removing the mean and scaling to unit variance.

```
0.00
                         1.00
                                            0.0
                0.00
                0.00
                        1.00
                                  0.00
                                           0.0
                       1.00
                0.00
                                  0.00
                                           0.0
                1.00
                         0.00
                                  0.00
                                          139.0
                        0.00
                                  0.00
        10
                1.00
                         0.00
                                  0.00
                                          151.0
                                        262.0
               1.00
                         0.00
                                  0.00
                                  0.00 4974.0
               1.00
                         0.00
                                  0.00 5907.0
0.00 5907.0
0.00 5907.0
  macro avg
weighted avg
                                            01
                                       0 0]
[ 96 28
[ 89 27
[ 107
Accuracy for our Training dataset with PCA: 0.0
Sihouette Score: 0.3812076198524835
\nSihouette Score- ranges from -1 to +1 , a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring
```

• Here we achieved the silhouette score 0.38 for the train sets.

```
# predict 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 or 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 cluster and poorly matched to neighboring clusters.
"""
```

			pre	cision	ا	recall	f1-	-score	SI	upport
		0		0.00		1.00		0.00		0.0
		1		0.00		1.00		0.00		0.0
		2		0.00		1.00		0.00		0.0
		6		1.00		0.00		0.00		65.0
		7		1.00		0.00		0.00		55.0
		8		1.00		0.00		0.00		68.0
		9		1.00		0.00		0.00		57.0
		10		1.00		0.00		0.00		85.0
		11		1.00		0.00		0.00		103.0
		12		1.00		0.00		0.00	:	2610.0
	aco	uracy						0.00		3043.0
		o avg		0.70		0.30		0.00		3043.0
WE	eighte	d avg		1.00		0.00		0.00		3043.0
[]		0	0	0	0	0	0	0	0	0]
		0	0	0	0	0	0	0	0	0]
		0	0	0	0	0	0	0	0	0]
		21	3	0	0	0	0	0	0	0]
	42	12	1	0	0	0	0	0	0	0]
	57	10	1	0	0	0	0	0	0	0]
	35	22	0	0	0	0	0	0	0	0]
	63	17	5	0	0	0	0	0	0	0]
•										
	[1763	450	397	0	0	0	0	0	0	0]]
				Trainin				ı PCA:	0.0	
Si	Lhouet	te Sc	ore:	0.3833	3223	4096896	i4			

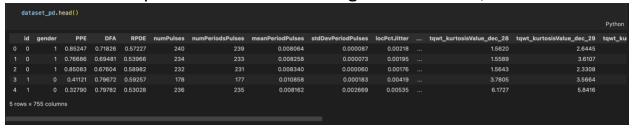
• Here we achieved the silhouette score 0.3833 for the test sets.

Question 2:

```
dataset_pd = pd.read_csv('datasets//pd_speech_features.csv')
   dataset_pd.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 756 entries, 0 to 755
Columns: 755 entries, id to class
dtypes: float64(749), int64(6)
memory usage: 4.4 MB
```

• Read the speech features.csv and get the info of I,



 Head function is used to get the n number of rows for the corresponding datasets mentioned.

```
dataset_pd.isnull().any()
id
                             False
gender
                             False
PPE
                             False
DFA
                             False
RPDE
                             False
                             . . .
tqwt_kurtosisValue_dec_33
                             False
tqwt_kurtosisValue_dec_34
                             False
tqwt_kurtosisValue_dec_35
                            False
tqwt_kurtosisValue_dec_36
                             False
class
                             False
Length: 755, dtype: bool
```

• Checking is there any null values or not.

• Scaling the datasets and finding the PCA for the three components as given in the question and the results are taken.

```
from sklearn.svm import SVC
   svmClassifier = SVC()
   svmClassifier.fit(X_train, y_train)
   y_pred = svmClassifier.predict(X_test)
   print(classification_report(y_test, y_pred, zero_division=1))
   print(confusion_matrix(y_test, y_pred))
   glass_acc_svc = accuracy_score(y_pred,y_test)
   print('accuracy is',glass_acc_svc )
   score = metrics.silhouette_score(X_test, y_pred)
   print("Sihouette Score: ",score)
                         recall f1-score support
             precision
                 0.67
                           0.42
                                    0.51
                                               62
                 0.84
                           0.93
                                               196
                                    0.88
                                               258
                                    0.81
   accuracy
                                  0.70
                 0.75 0.68
                                               258
  macro avg
weighted avg
                 0.80
                           0.81
                                    0.79
                                               258
[[ 26 36]
[ 13 183]]
accuracy is 0.810077519379845
Sihouette Score: 0.2504463929631217
```

• By using the support vector machine to report the performance with the accuracy of 0.81 and the Silhouette score od 0.25.

```
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
   dataset_iris = pd.read_csv('datasets//Iris.csv')
   dataset_iris.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
                  Non-Null Count Dtype
    Column
0 Id
                 150 non-null int64
1 SepalLengthCm 150 non-null
                                float64
2 SepalWidthCm 150 non-null float64
3 PetalLengthCm 150 non-null float64
4 PetalWidthCm 150 non-null
                                  float64
5 Species 150 non-null
                                  object
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB
   dataset_iris.isnull().any()
Ιd
                False
SepalLengthCm
               False
SepalWidthCm
                False
PetalLengthCm
               False
PetalWidthCm
               False
Species
               False
dtype: bool
```

- Apply Linear Discriminant Analysis (LDA) on Iris.csv dataset to reduce dimensionality of data to k=2.
- Also, to find is there any null values related to that.

```
x = dataset_iris.iloc[:,1:-1]
y = dataset_iris.iloc[:,-1]
print(x.shape,y.shape)

(150, 4) (150,)

X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=0)

sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
le = LabelEncoder()
y = le.fit_transform(y)

from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA
lda = LDA(n_components=2)
X_train = lda.fit_transform(X_train, y_train)
X_test = lda.transform(X_test)
print(X_train.shape,X_test.shape)

(105, 2) (45, 2)
```

• By using the scaling and LDA to reduce the dimensionality of data to k=2 and gave the results.

Question number 4:

1. Briefly identify the difference between PCA and LDA

Both LDA and PCA rely on linear transformations and aim to maximize the variance in a lower dimension. PCA is an unsupervised learning algorithm while LDA is a supervised learning algorithm. This means that PCA finds directions of maximum variance regardless of class labels while LDA finds directions of maximum class separability.

PCA:

It reduces the features into a smaller subset of orthogonal variables, called principal components – linear combinations of the original variables. The first component captures the largest variability of the data, while the second captures the second largest, and so on.

LDA:

LDA finds the linear discriminants in order to maximize the variance between the different categories while minimizing the variance within the class.

Video link:

https://drive.google.com/file/d/1J1vkGzL2yK28sNufseqZjLtbi-I6aAQS/view?usp=share link

GitHub link: https://github.com/Jitheandra/Assignment5

Thank you,
Jitheandra Subramaniam.