### ML\_Assignment\_5

Sushmitha Virri

#### #700742289

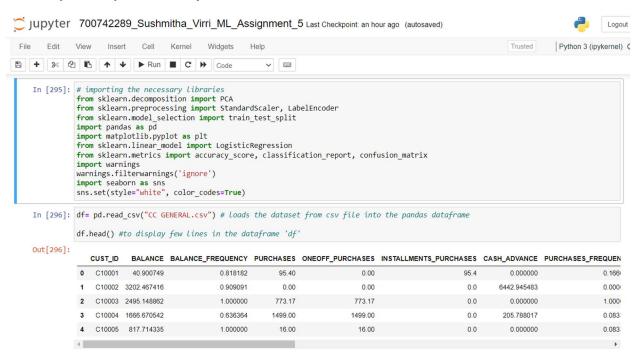
Github Link: https://github.com/Sushmitha-

Virri/MLAssignments21627/blob/main/700742289 Sushmitha Virri ML Assignment 5.ipynb

#### Drive Video Link:

https://drive.google.com/file/d/10XSx5hnwqFJPNDWGBte3bJkWGiMZHB0X/view?usp=sharing

#### 1. Principal Component Analysis



In the above python script, we have imported the required modules from the libraries, including pandas for data manipulation, StandardScaler for feature scaling, PCA for dimensionality reduction, KMeans for clustering and metrics evaluation.

We have loaded the CC GENEREAL dataset from csv file to pandas data frame.

head() function displays the first few lines of the dataframe.

Next we use df.isnull().any() function to check whether there are any missing values. If any, fill them with mean of each column using fillna() function.

'inplace = True' argument will make changes to the original dataframe and is displayed in the output

#### Jupyter 700742289\_Sushmitha\_Virri\_ML\_Assignment\_5 Last Checkpoint: an hour a File Edit View Insert Cell Kernel Widgets Help **4** 38 ► Run C Code In [297]: df.shape # using shape attribute to get the shape of df Out[297]: (8950, 18) In [298]: df.isnull().any() Out[298]: CUST\_ID False BALANCE False BALANCE\_FREQUENCY False False **PURCHASES** 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 True **PAYMENTS** False MINIMUM\_PAYMENTS True PRC\_FULL\_PAYMENT False **TENURE** False dtype: bool In [299]: df.fillna(df.mean(), inplace = True) df.isnull().any() Out[299]: CUST ID False False BALANCE BALANCE\_FREQUENCY False **PURCHASES** False ONEOFF PURCHASES False INSTALLMENTS PURCHASES False CASH ADVANCE False

False

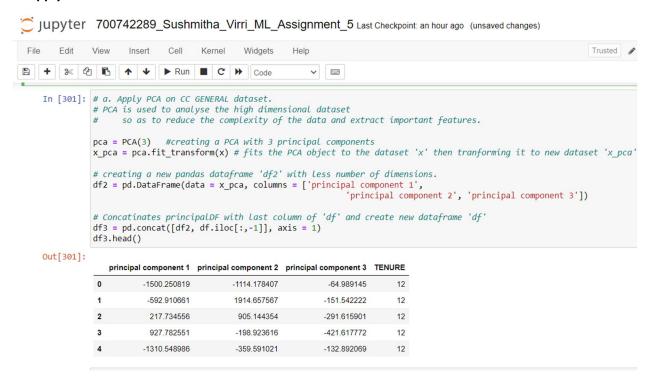
PURCHASES FREQUENCY



The above code selects a subset of columns at index 1,2,3,4 from pandas Dataframe 'df' and assigns it to 'x' and selects all rows and last column and assign it to 'y'.

Then it prints the shape of 'x' and 'y' which represent the number of rows and colums in the output cell.

#### a. Apply PCA on CC dataset.



In this code we applied PCA on CC GENERAL dataset. In the first line we created a 'pca' object with 3 principal components.

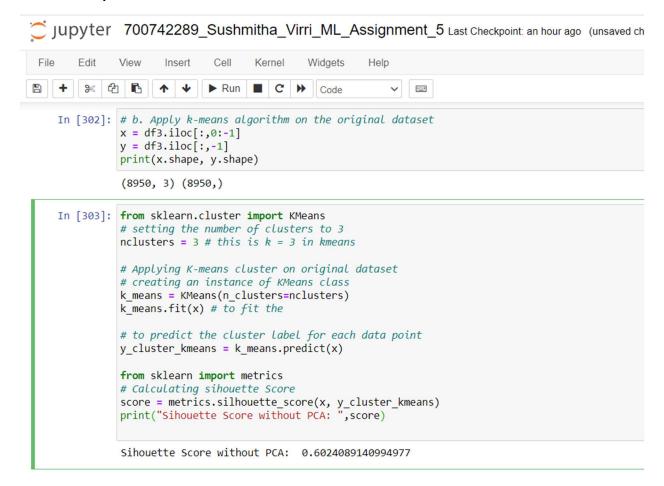
In the second line PCA object is fit to the original dataset 'x' and transformed to a new dataset 'x' pca'

The third statement creates a new pandas dataframe 'df2' with principal component columns.

Next statement concatenates 'df2' with last column of 'df' and assigned to a new dataframe 'df3'

The output cell shows reduced dimensions of the dataset.

# b. Apply k-means algorithm on the PCA result and report your observation if the silhouette score has improved or not?

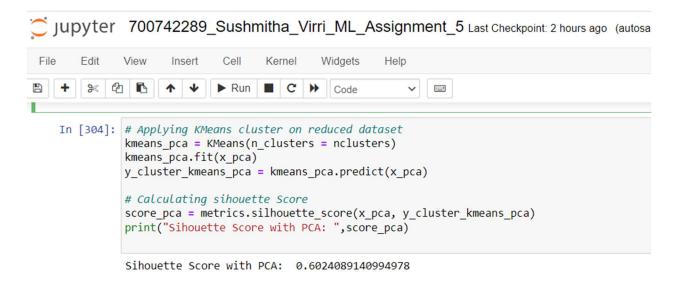


The above screenshot talks about the application of KMeans clustering algorithm from scikit-learn library to the original dataset.

The number of clusters is set to 3 and stored in variable nclusters. KMeans class instance is created with this value and fit() method is called to fit the data.

Predict() method is used to predict the cluster label for each datapoint.

silhouette\_score() function calculates the score with input values as original dataset and predicted labels.



The above code apples KMeans clustering to preprocessed dataset with PCA.

An instance of KMeans class is created with nclusters. fit() method is called on this instance with input as 'x\_pca' and predict() method is then called on kmeans\_pca to predict the cluster labels for PCA transformed dataset x pca.

Silhouette score is then calculated with PCA-transformed dataset and cluster labels as input.

**Performance:** From the above we can observe that the silhouette score with PCA is slightly less than the silhouette score without PCA.

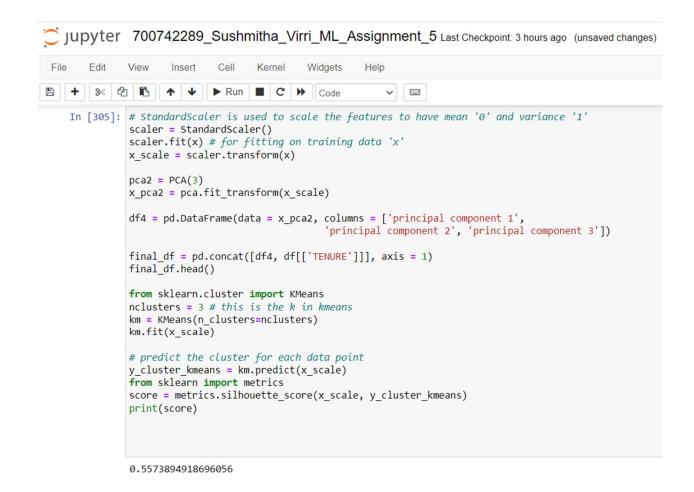
**Reason**: Implementing PCA has reduced the number of features in the dataset.

#### c. Perform Scaling+PCA+K-Means and report performance.

To perform dimensionality reduction using PCA and clustering using K-means on the dataset.

The code for this creates an instance of StandardScaler class and fits it on the training data 'x' and then transforms it into fitted scaler.

The code then performs feature scaling using StandardScaler, dimensionality reduction using PCA, clustering using KMeans and then evaluates clustering performance using silhouette score.

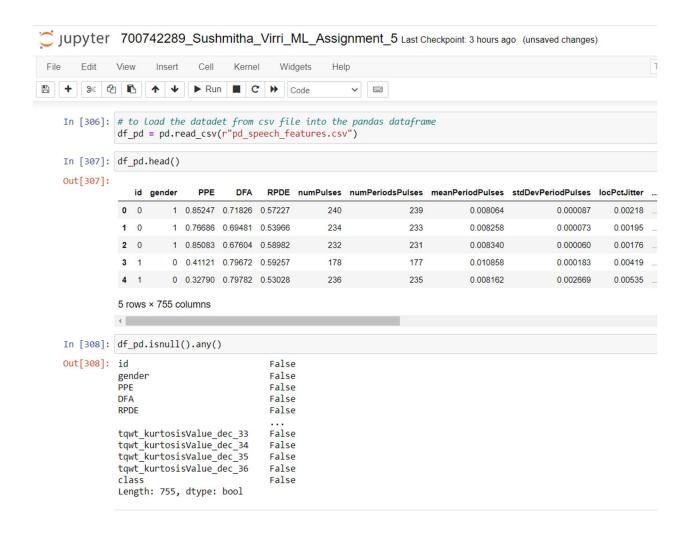


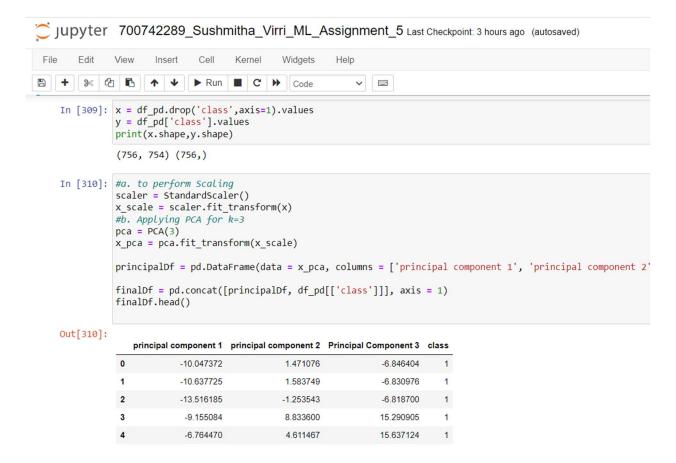
#### Performance report:

We can observe from the output that the silhouette score after applying Scaling + PCA + K-means is reduced when compared to the silhouette scores without PCA and with PCA.

From this we can say that as the silhouette score is not improved that is it is not greater than the previous values and hence the performance is not improved.

- 2. Use pd\_speech\_features.csv
- a. Perform Scaling
- b. Apply PCA (k=3)
- c. Use SVM to report performance





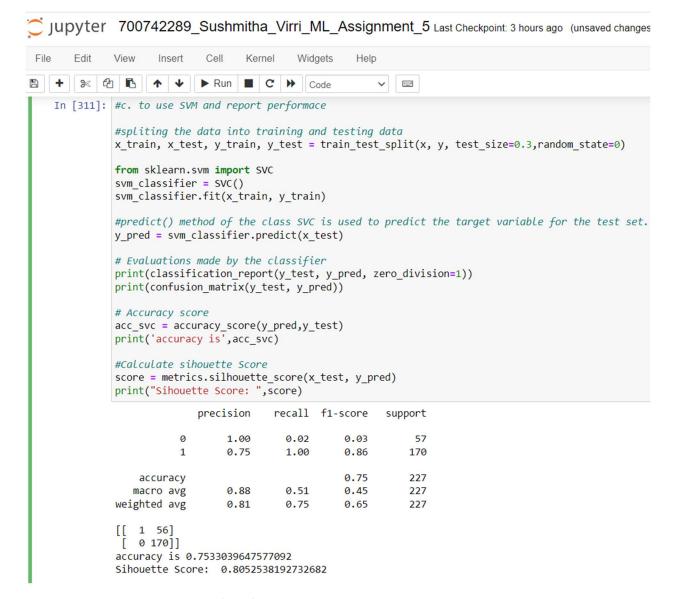
#### a. Scalaing:

In the first step we scale the data using StandardScaler() function and stored in the variable 'x\_scale'.

#### b. PCA:

The next step is to perform PCA on the scaled data using PCA() function with k=3 and store it in the variable 'x\_pca'.

Next we create a dataframe 'principalDF' with three principal components. The code then concatenates 'principalDF' with 'class' column using the pd.concat() function.

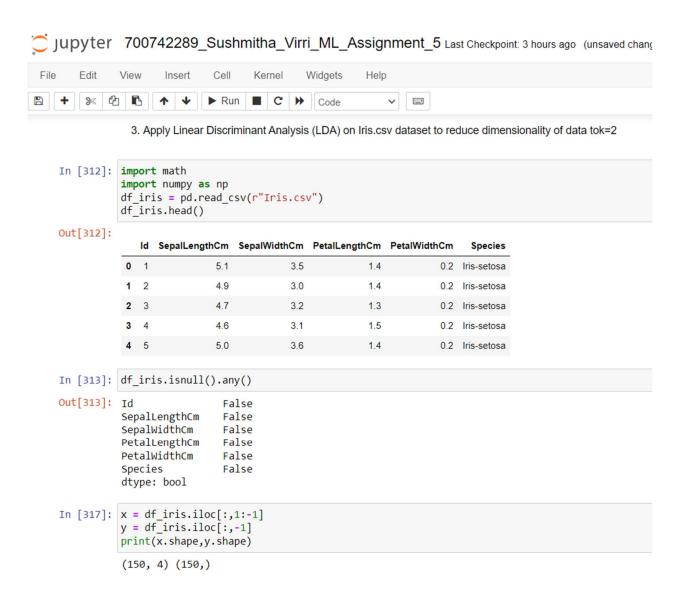


#### c. Support Vector Machine(SVM):

The data is split into training and testing sets using train\_test\_split() function. The SVM classifier is trained on the training set using the fit() method.

classification\_report() and confusion\_matrix() are used to determine the performance of the classifier.

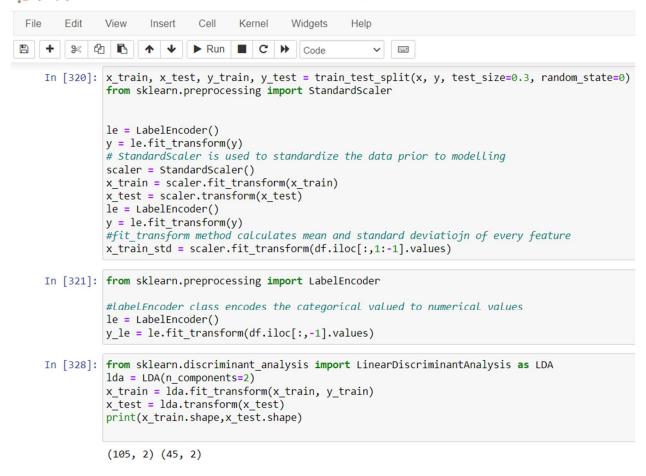
silhouette\_score() measures the similarity of data points within the cluster and the dissimilarity of data points among different clusters.



The above code reads the data from 'iris.csv' file inti the pandas dataframe 'df\_iris'. Displays the first five columns. Next checks for any missing values in the dataframe, then separates dataframe into 2 parts using 'iloc' method.

- 1. 'x' which contain all features of dataset except target variable.
- 2. 'y' which contain only the target variable.

## Jupyter 700742289\_Sushmitha\_Virri\_ML\_Assignment\_5 Last Checkpoint: 3 hours ago (autosaved)



The above code performs following operations:

Splitting the data into training and testing sets using train test split' function from the sklearn library.

Encoding the target variable using 'LabelEncoder'

Standardizing the target variable using 'StandardScaler'

Then applying (LDA) Linear Discriminant Analysis to reduce feature dimension to 2.

Thus the code does preprocessing and reduced the dimensions using LDA.

#### 4. Explain briefly the distinction between PCA and LDA.

LDA and PCA both use linear transformations to maximize variance in a smaller dimension. The PCA method is an unsupervised learning algorithm, whereas the LDA algorithm is a supervised learning system. This means that PCA seeks maximum variance directions regardless of class labels, whereas LDA finds maximum class separability directions.

LDA and PCA both use linear transformations to maximize variance in a lower dimension. The PCA method is an unsupervised learning algorithm, whereas the LDA algorithm is a supervised learning system. This indicates that PCA finds maximum variance directions regardless of class labels, whereas LDA finds maximum class separability directions.

PCA: It condenses the characteristics into a smaller set of orthogonal variables known as principal components, which are linear combinations of the original variables. The first component captures the most variability in the data, the second the second, and so on. It reduces the features to a smaller group of orthogonal variables called principal components - linear combinations of the original variables. The first component captures the most variability in the data, the second the second most, and so on.

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