Assignment – 5

GitHub link to access the source code and video which demonstrates my work. https://github.com/tanikonda7441/Assignment-5

Source code can be found under the source code file.

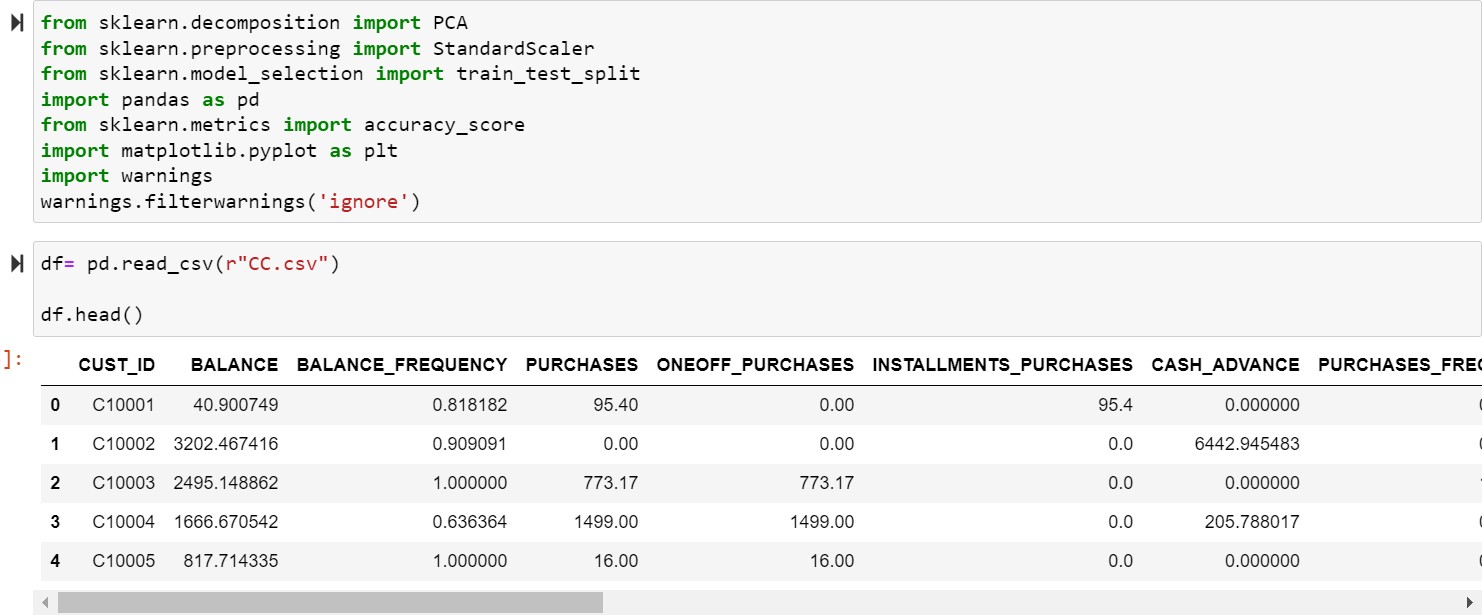
There is a README file which contains student Info, short description about assignment. A Video is available in README file under Video section.

# Principal Component Analysis

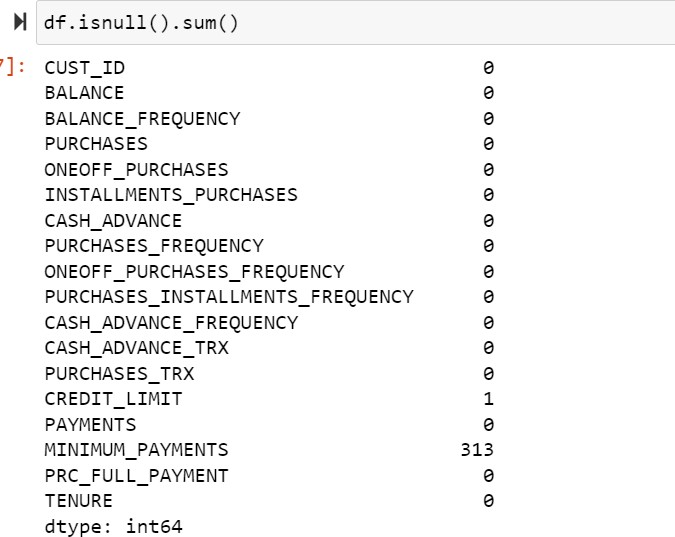
* 1. **Apply PCA on CC dataset.**

To do data analysis and apply machine learning algorithms on data, first I imported a few python libraries.

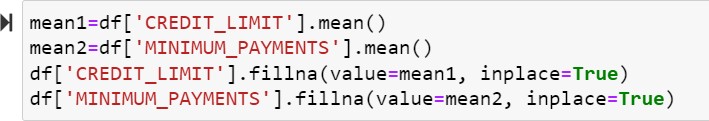
Using read\_csv method imported “CC“ data set. The head() method of pandas library results top most rows of a data set.



Isnull() method of pandas library checks for any values present in data set.



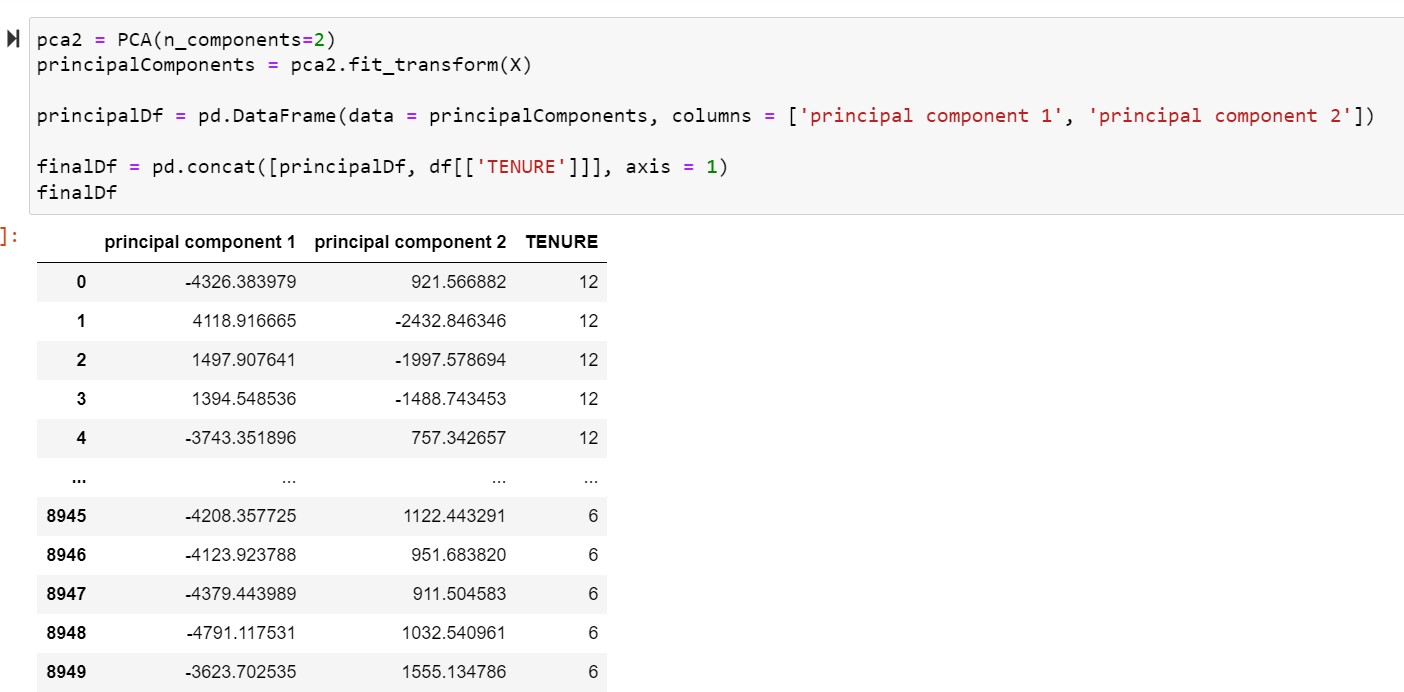
In this data set there are a few null values present in minimum payments and credit limit columns. So, these null values are replaced with their column mean value using fillna() method.



To perform PCA on this data set we don’t need the output labels because PCA does not rely on the output labels. Using the drop method, we removed a few columns which are unnecessary.

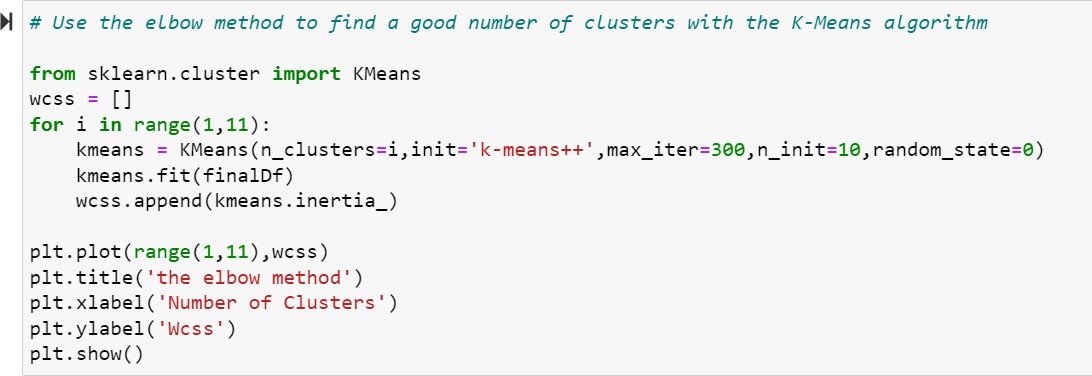


From sklearn python library we imported PCA method to perform PCA on the data set. PCA results in a data frame with features having maximum variance with other features by ignoring the duplicate features. Here we reduced the dimensionality of data into two components by keeping k value is equal to 2.

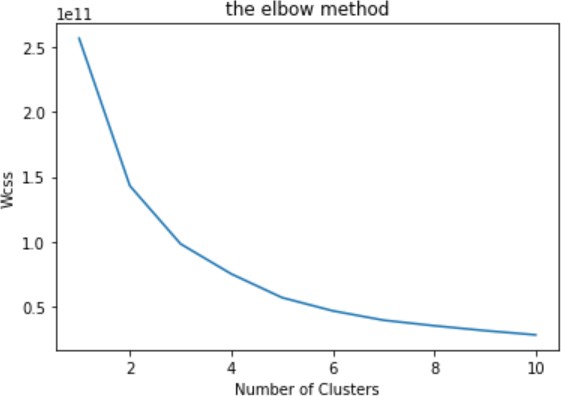


# Apply k-means algorithm on the PCA result and report your observation if the silhouette score has improved or not?

To perform k-means algorithm on a data set first we need to find the number of clusters required to fit our data together into clusters by using elbow method. This elbow method results in a graph from where we need to find the number of clusters value i.e., k value.

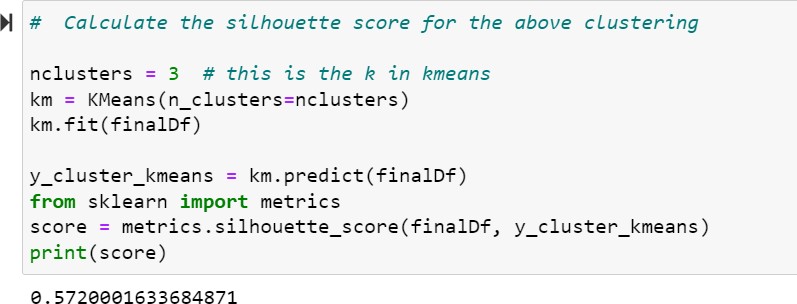


The elbow method results in a graph. From graph, the next point to point where the wcss value starts decreasing linearly will be the k value. From the graph below, from number of clusters is 2 the wcss value starts decreasing linearly. So, the number of clusters required to fit our data is 3 i.e., k value is 3. In k-means algorithm k is the number of clusters.



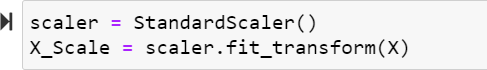
Using KMeans method of sklearn library, I applied K- Means algorithm on data set we got after performing PCA. After performing k-means on the PCA data we got a silhouette score of 57% which is higher than the silhouette score of raw data without performing PCA.

The silhouette score has been improved when we perform PCA on the data set. when we applied kmeans on the data set without performing PCA we got a silhouette score of 46.5%. After performing PCA we got a silhouette score of 57%. The silhouette score has been improved by more than 10%.

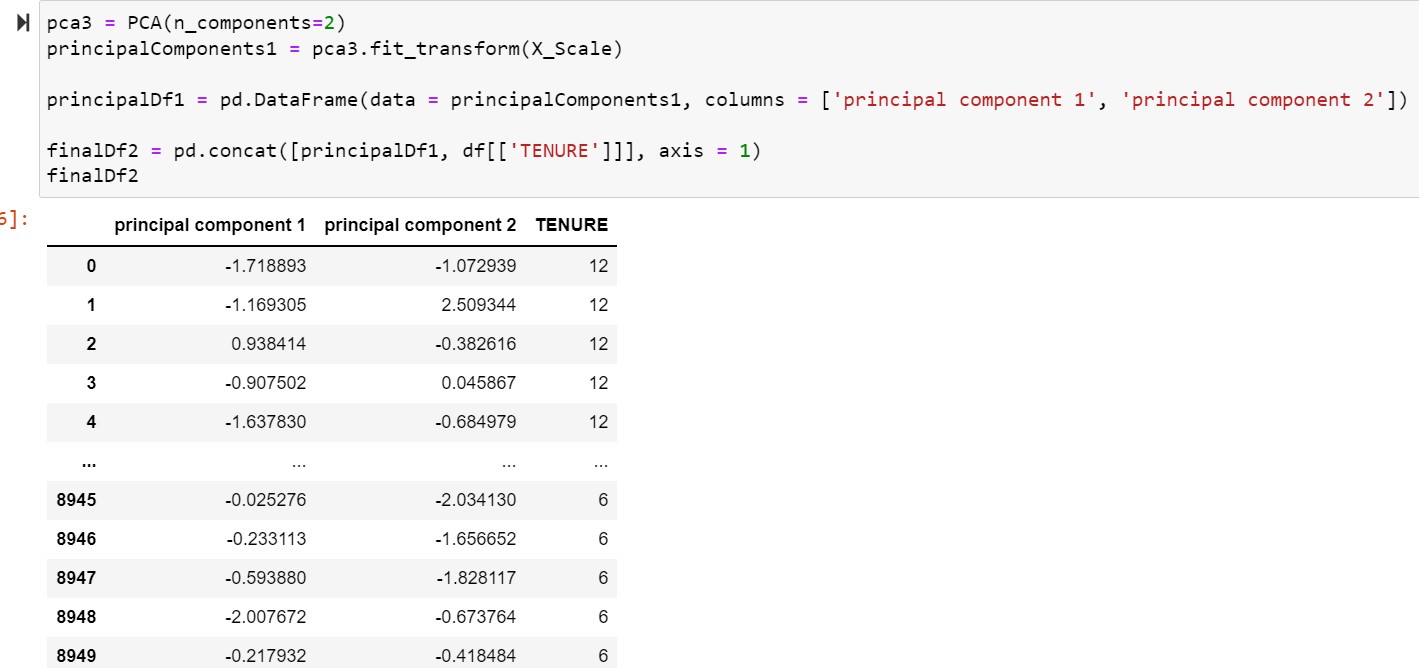


# Perform Scaling + PCA + K-Means and report performance.

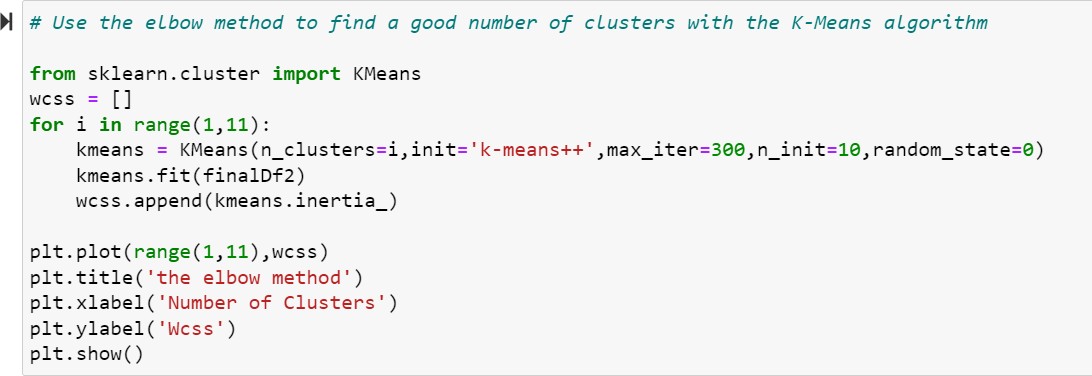
Using StandardScalar method we performed feature scaling on the data set. Feature scaling is used to normalize the range of all features.



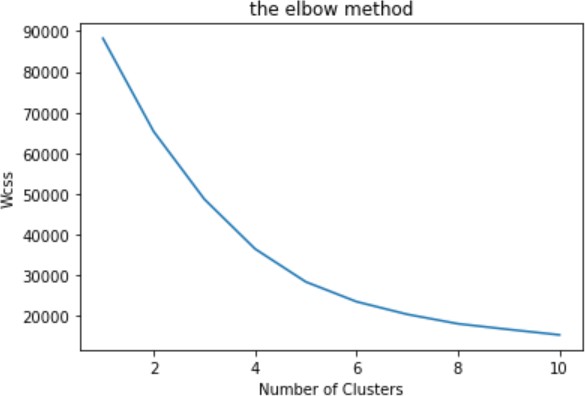
We are performing PCA on the feature scaled data set using the PCA method.



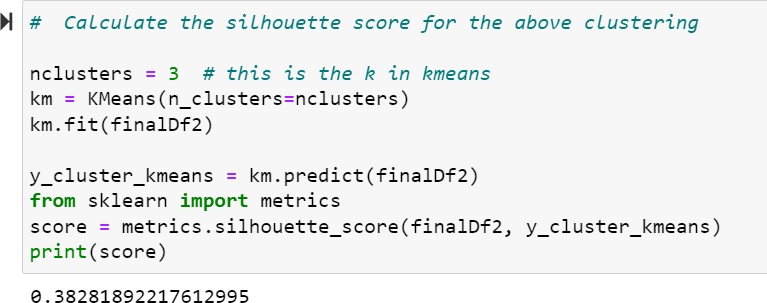
To perform k-means algorithm on a data set first we need to find the number of clusters required to fit our data together into clusters by using elbow method.



The elbow method results in a graph. From graph, the next point to point where the wcss value starts decreasing linearly will be the k value. From the graph below, from number of clusters is 2 the wcss value starts decreasing linearly. So, the number of clusters required to fit our data is 3 i.e., k value is 3.

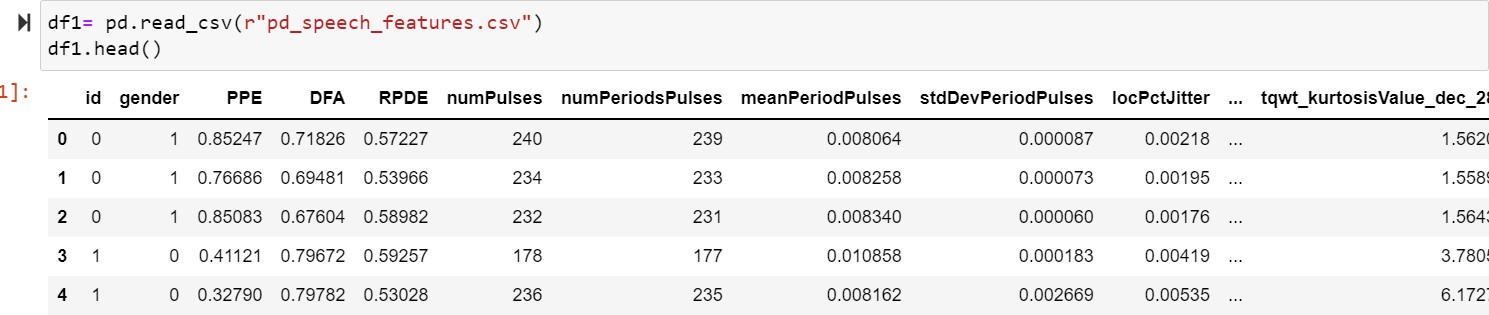


Using KMeans method of sklearn library, I applied K- Means algorithm by taking k value as 3 on data set, we got after performing feature scaling and PCA. After performing k-means on this data we got a silhouette score of 38%.



# Use pd\_speech\_features.csv

Using read\_csv method imported a csv file. The head() method of pandas library results top most rows of a data set.



# Perform Scaling

Using StandardScalar method we performed feature scaling on the data set. Feature scaling is used to normalize the range of all features.

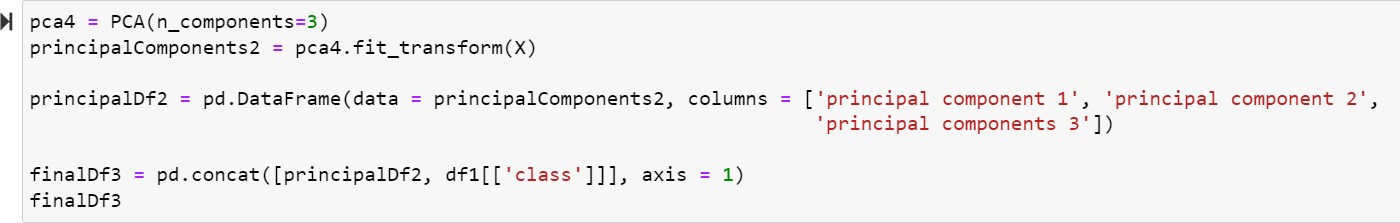


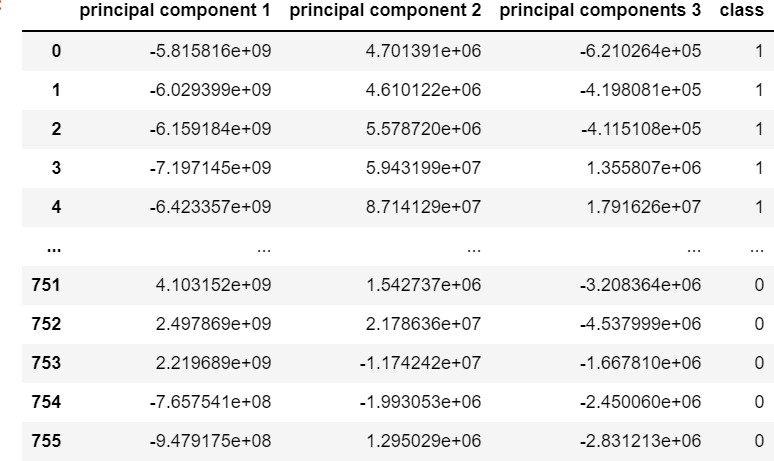
# Apply PCA (k=3)

To perform PCA on this data set we don’t need the output labels because PCA does not rely on the output labels. Using the drop method, we removed a class column which is unnecessary.



To run PCA on the data set, we imported the PCA method from the Sklearn Python package. By discarding the duplicate features, PCA produces a data frame with features that have the greatest variation with other characteristics. Here, we kept the k value at 3, which reduced the data's dimension to three components.



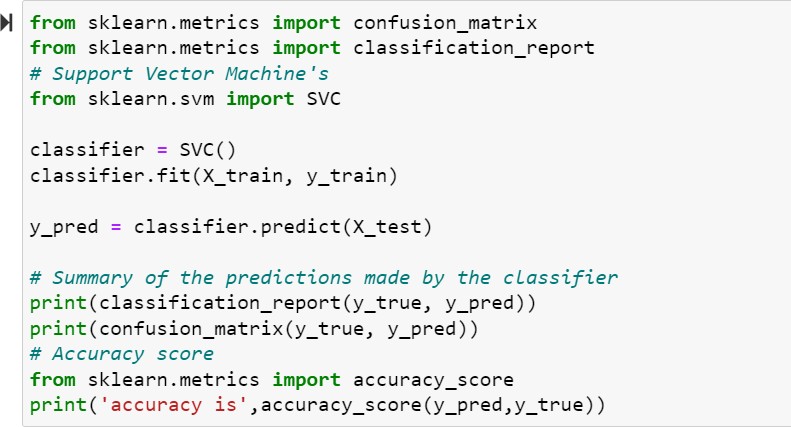


# Use SVM to report performance

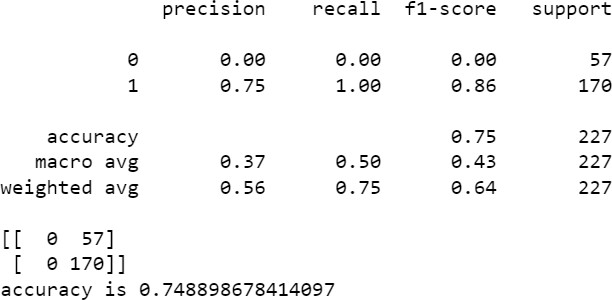
sklearn module contains train\_test\_split method to split our data set into training and testing data sets. In this method, test\_size defines how much proportion of data to be in the test data set. When we change test\_size value whole analysis results will change.



Support vector machine algorithm is applied to the data set we got after performing PCA using sklearn module. We got an accuracy of 74.8% when we trained SVM on our data set.



The Machine learning algorithms are being compared with each other based on different metrices like F1-score, accuracy, recall and precision. Based on those values we are going to determine which algorithm is the most accurate and complete classifier for the dataset.

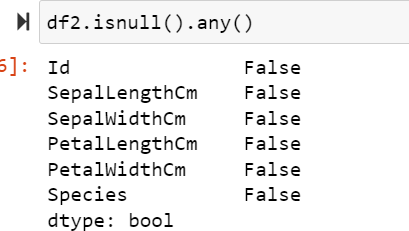


# Apply Linear Discriminant Analysis (LDA) on Iris.csv dataset to reduce dimensionality of data to k=2.

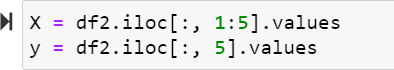
A csv file was imported using the read\_csv method. The top rows of a data set are returned by the pandas library's head() method.



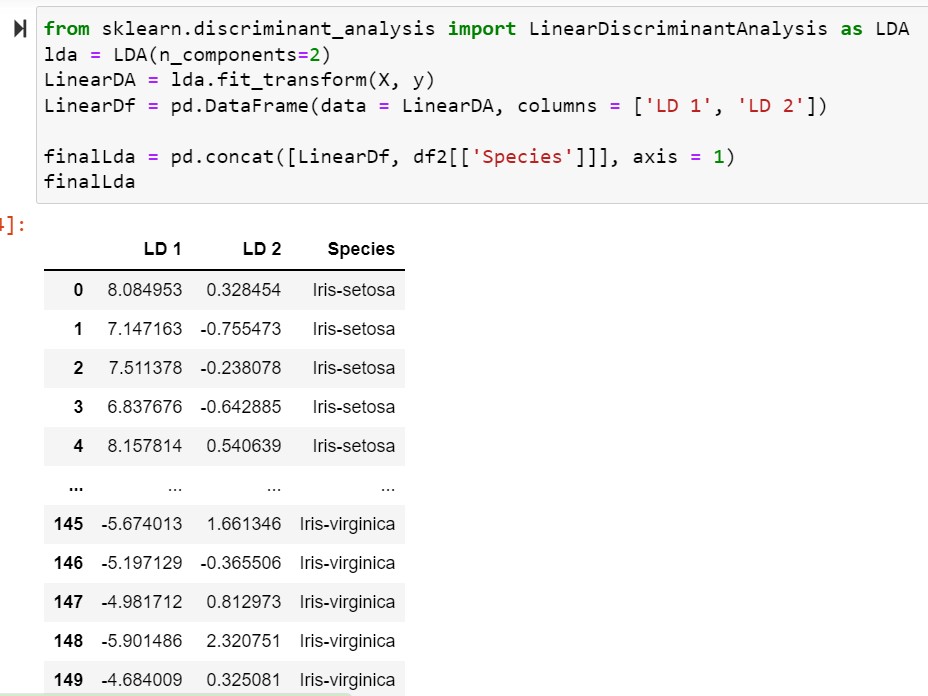
Isnull() method of pandas library checks for any values present in data set. In this iris dataset there are no null values.



To perform LDA on this data set we need the output labels because LDA rely on these output labels to reduce the dimensionality of data based on output classes.



The LinearDiscriminantAnalysis class of the sklearn.discriminant\_analysis library can be used to Perform LDA in Python. By setting n\_components value as 2 we will get the results in two linear discriminates. We execute the fit and transform methods to retrieve our results.



# Briefly identify the difference between PCA and LDA

Dimensionality reduction in machine learning refers to the process of collecting a collection of major variables to reduce the number of random variables being considered. Principle Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are two main algorithms in dimensionality reduction.

PCA is an unsupervised while LDA is a supervised dimensionality reduction technique.

PCA gets the results without depending on the output labels. PCA results a data set with maximum variance between the features by ignoring the duplicates of other features. Since the variance between the features is independent of the outcome, PCA does not consider the output labels.

LDA depends on the output labels. Based on the output labels information LDA reduces the feature set dimensions and finds a decision boundary. The data points are then projected to new dimensions so that the clusters are as distinct from one another as possible, and the individual components of a cluster are as near the cluster centroid as possible.