

APPLIED MACHINE LEARNING LAB ASSIGNMENT

*Submitted to*

Manipal School of Information Sciences, MAHE, Manipal

|  |  |  |
| --- | --- | --- |
| **Registration No** | **Name** | **Branch** |
| **211057020** | **Aditi** | **Artificial Intelligence and Machine Learning** |



**TABLE OF CONTENTS**

1. Lab Assignment 1 3

2. Lab Assignment 2 9

3. Lab Assignment 3 12

4. Lab Assignment 4 17

5. Lab Assignment 5 21

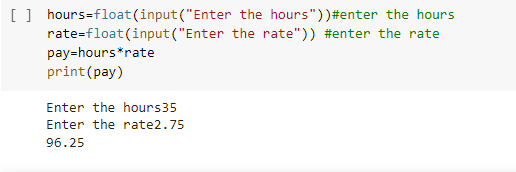
6. Lab Assignment 6 26

**LAB ASSIGNMENT 1**

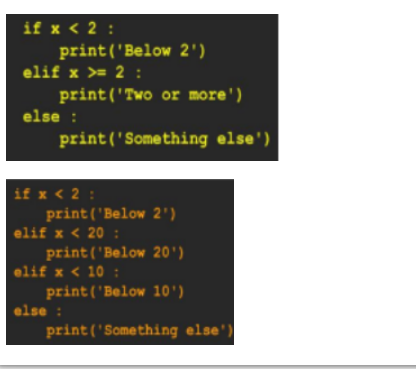
**1)** Write a program to prompt the user for hours and rate per hour to compute gross pay. Hour = 35 & Rate = 2.75

**Answer:**

We enter the hours and the rate as float value and then multiply hours and rate to get gross pay.

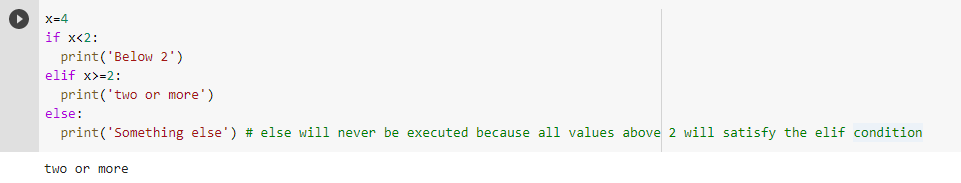
****

**2)** Which will never be printed out of the following two code sets:

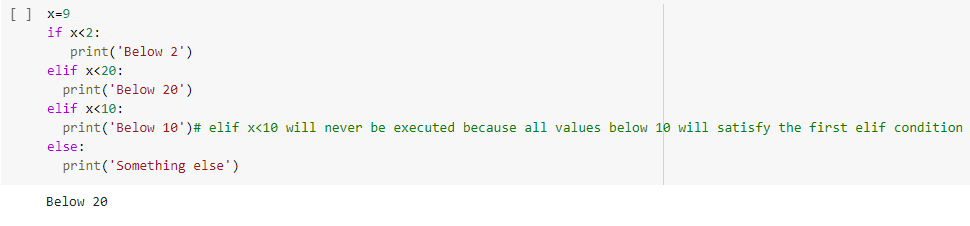


**Answer:**

In the first code block else will never execute as all the values above or equal to 2 are satisfying the “elif” condition and all the values below two are satisfying the “if” condition.



In the second block elif < 10 will never execute because all values below 10 will satisfy the first elif condition.



# 3) Rewrite the program-1 using try and except to prompt the user for hours and rate per hour to compute gross pay. (If non-numeric inputs entered, it should except). Hour = 35 & Rate = 2.75.

# Answer:

# When we enter a non numeric value it will execute the except block.

# 

# 4) Rewrite the program-1 with time-and-a half for overtime and create a function called paycomp which takes two parameters (hours and rate per hour). Hours = 45 and rate = 10.

# Answer:

# When hours is greater than 45 it will be multiplied with 1.5 times the rate.

# 

# 5) What is the code doing?

# 

# Answer:

# In the first case as the value of n=0 the while loop fails and only the last print statement executes.

# In the second case as the n value is not updated so as it enters the while loop it executes infinitely and the last print statement never executes.

# 

# 6) Consider the list of elements [9, 41, 23, 54, 33, 21, 8] use for loop to find

# 1. Largest number

# 2. Smallest number

# 3. Number of numbers

# 4. Number of odd numbers

# 5. Number of even numbers

# 6. Number of prime numbers

# 7. Sum and average of numbers

# 8. Filter the numbers greater than 20

# 9. Filter the numbers less than 15

# 10. Search for number 3

# Answer:

# 

# 

# 

# 

**7)** Illustrate the use of type operator and type conversion

# Answer:

# 

**8)** Illustrate the use of break and continue

**Answer:**

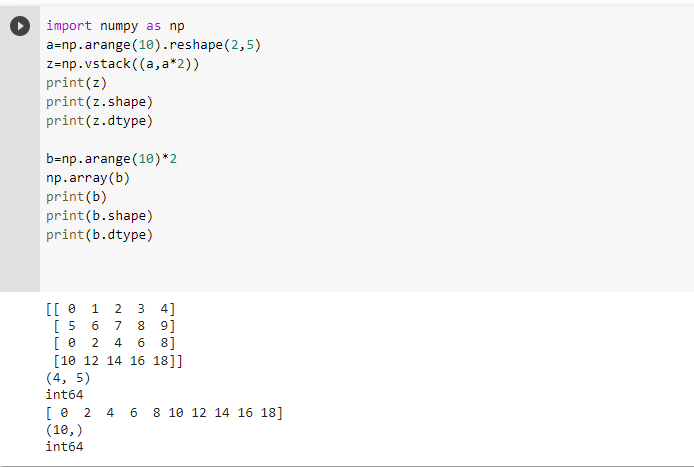
# 

**LAB ASSIGNMENT 2**

**1)** Find the shape and type of (a, a\*2)

**Answer:**

Use shape to find the shape and the type of the variable.

****

**2)** An array a= (4, -3, -2, 1, 3, 4).

a) Find sum of the elements in the array.

b) Find the small and big elements of the array.

c) Find the mean and standard deviation of the array elements.

d) Find the index of the biggest and smallest elements of the array.

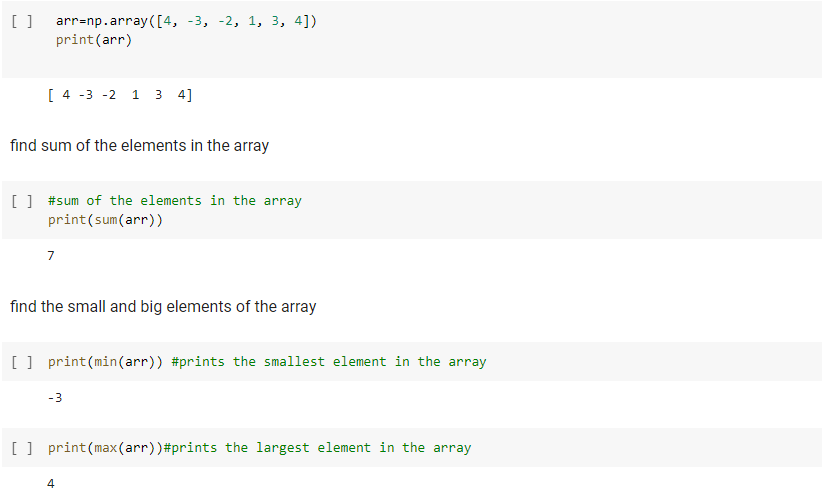
**Answer:**

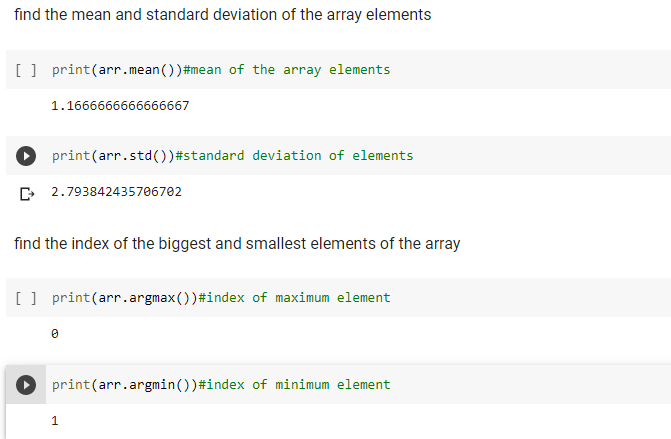
We can use sum to find the sum () of the elements in the array.

min () can be used to find the minimum elements and max () can be used to find the maximum elements in the array.

mean () and sd () can be used to find the mean and the standard deviation of the elements in array respectively.

argmax () and argmin () can be used to find the index of the largest element and the index of the smallest element in the array respectively.

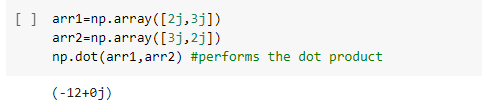
****

****

**3)** Find the dot product of (2j, 3j), (3j, 2j)

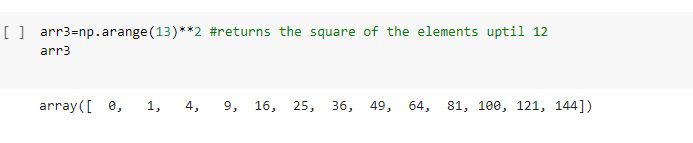
**Answer:**

Use dot function to perform dot product.



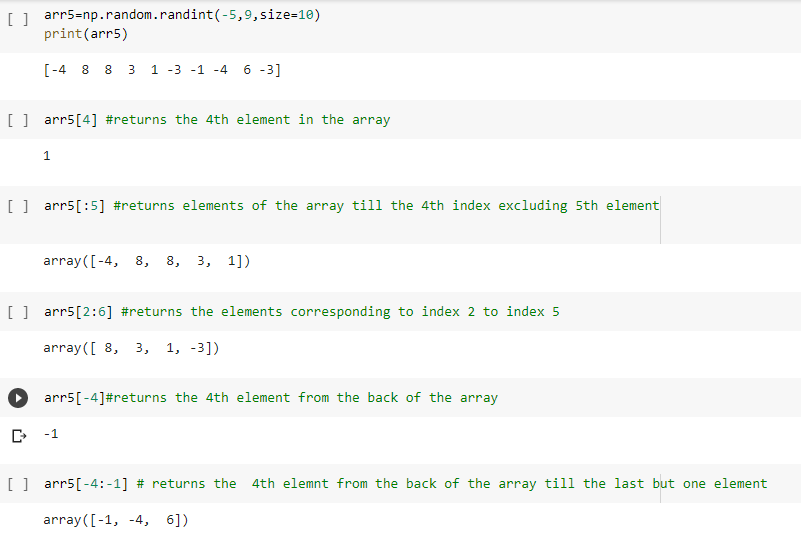
**4)** Create an array which contain square of all integer numbers between 0 and 13

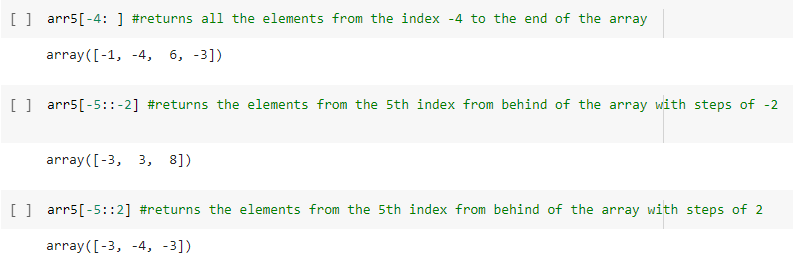
**Answer:**

****

**5)** Use bracket notation to get value particular index.

**Answer:**

****

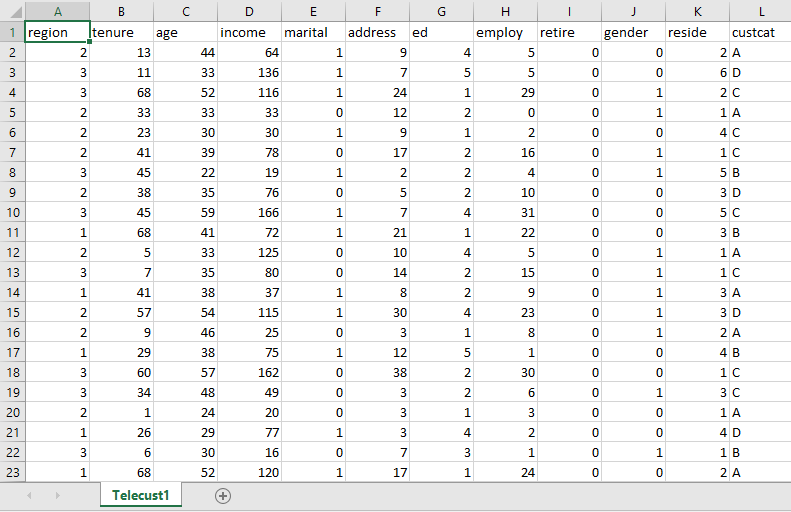
****

**LAB ASSIGNMENT 3**

**Implementing Decision Tree, KNN, Naïve Bayes on Tele-Customer Dataset**

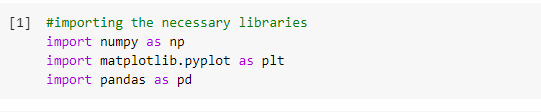
**Tele-Customer Dataset**

In this dataset the data of the telecommunication customers is stored like their age, income, marital status and on the basis of that they are classified into 4 categories A, B, C, and D.



**Decision Tree**

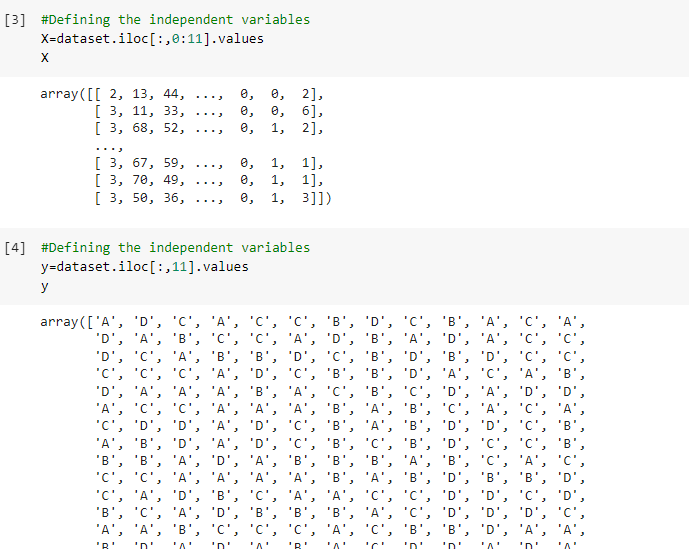
First step is to import the necessary libraries



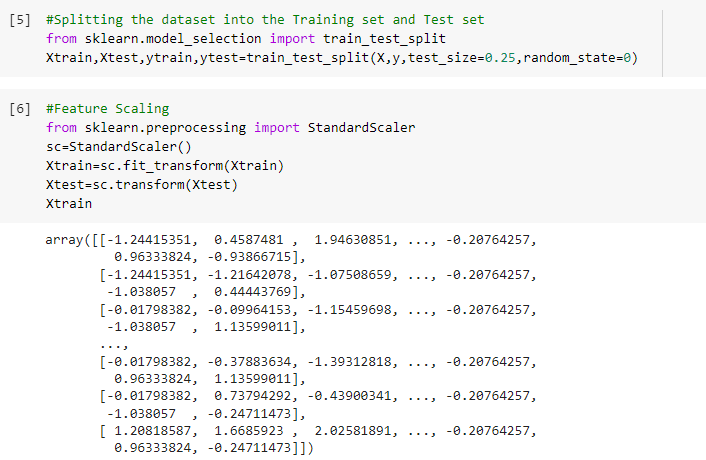
Second step is to import the dataset



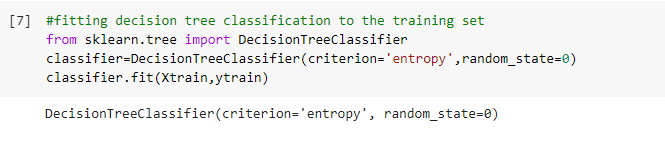
Third step is to define the dependent and the independent variables



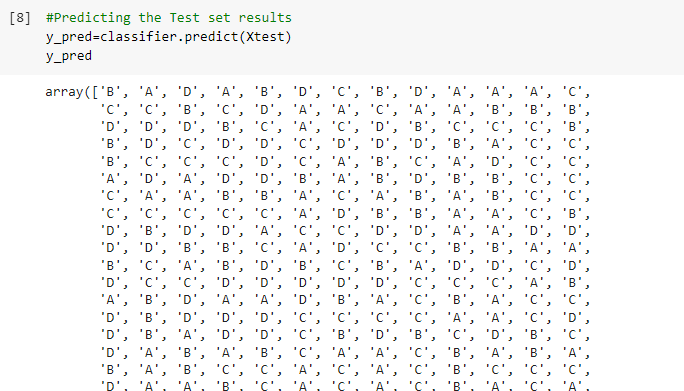
Fourth step is to split dataset into training and testing and as we want all the features in the data on a same scale we perform feature scaling.



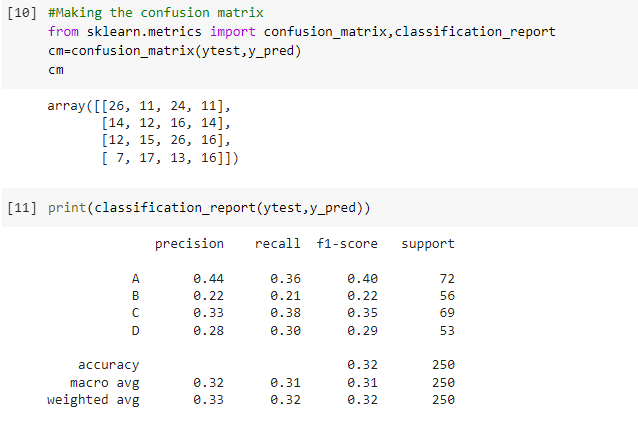
In the fifth step we fit the decision tree classification to the training set.

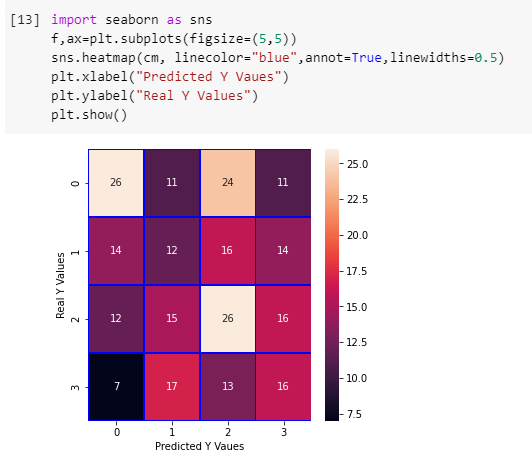


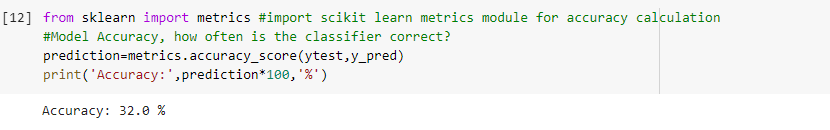
In the sixth step we perform the prediction.



In the next step we print the confusion matrix and the classification report.



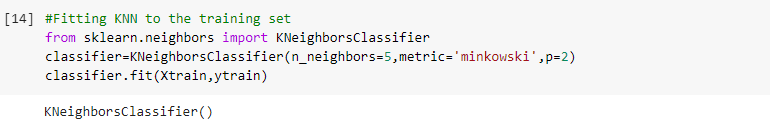
Finally we find the accuracy of the model.



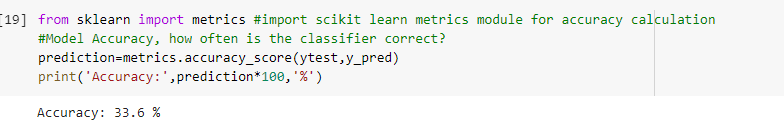
We can see that the decision tree gives 32% accuracy so the model needs to be improvised.

**KNN**

The same first four steps are followed for KNN as well. In the 5th step we fit the KNN model to the training data.



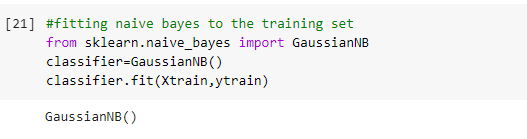
Again we follow the same steps performed for the decision tree and then finally we print the accuracy of the model.



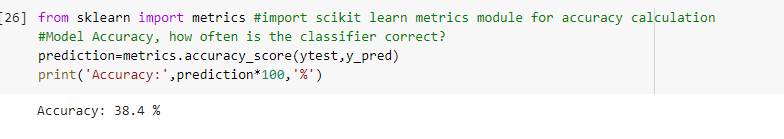
We can see a 1% difference in the accuracy of the model when compared to DT.

**NAÏVE BAYES**

Perform the same steps as for the DT and KNN in the fifth step fit the Naïve Bayes classifier on the training data.



After prediction print the accuracy of the model.



The accuracy has improved when compared to that of the DT and KNN.

**LAB ASSIGNMENT 4**

**Implementing hierarchical clustering on Customer Purchase Dataset**

# Hierarchical Clustering

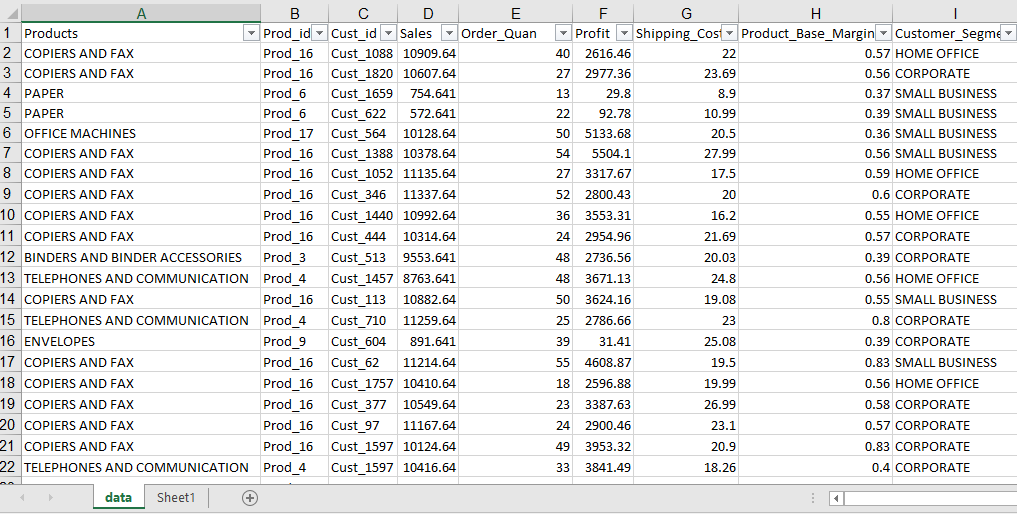
Hierarchical clustering(also called hierarchical cluster analysis or HCA) is a method of cluster analysis which seeks to build a hierarchy of clusters. Strategies for hierarchical clustering generally fall into two types:

* **Agglomerative**: This is a "bottom-up" approach: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.
* **Divisive**: This is a "top-down" approach: all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.

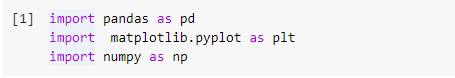
In general, the merges and splits are determined in a greedy manner. The results of hierarchical clustering are usually presented in a dendrogram.

**Customer Purchase Dataset**

In this dataset the information on the products purchased by a customer, the sales, shipping cost, profit and based on these divides the customers into 3 categories

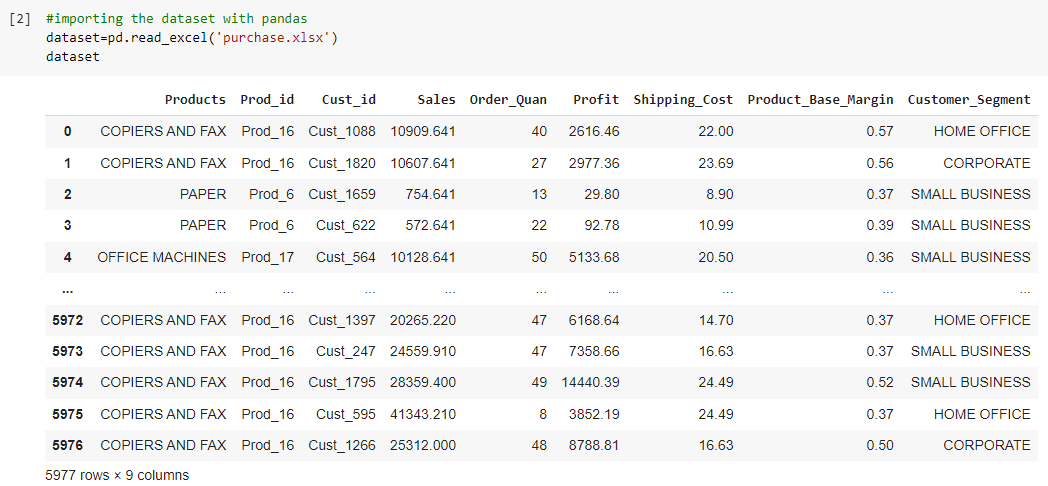


First step is importing the necessary libraries.

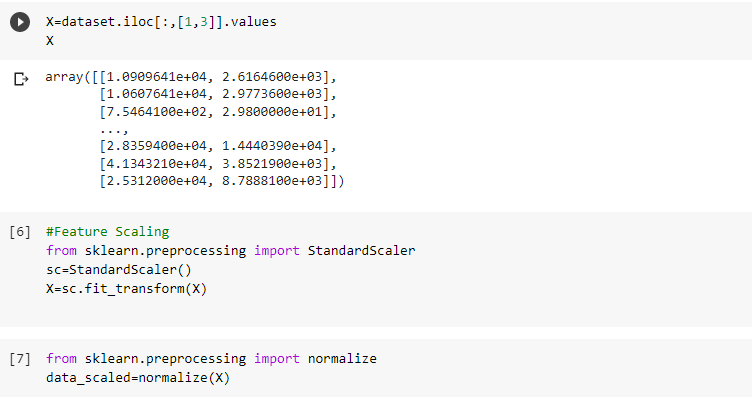


Second step we import the dataset.

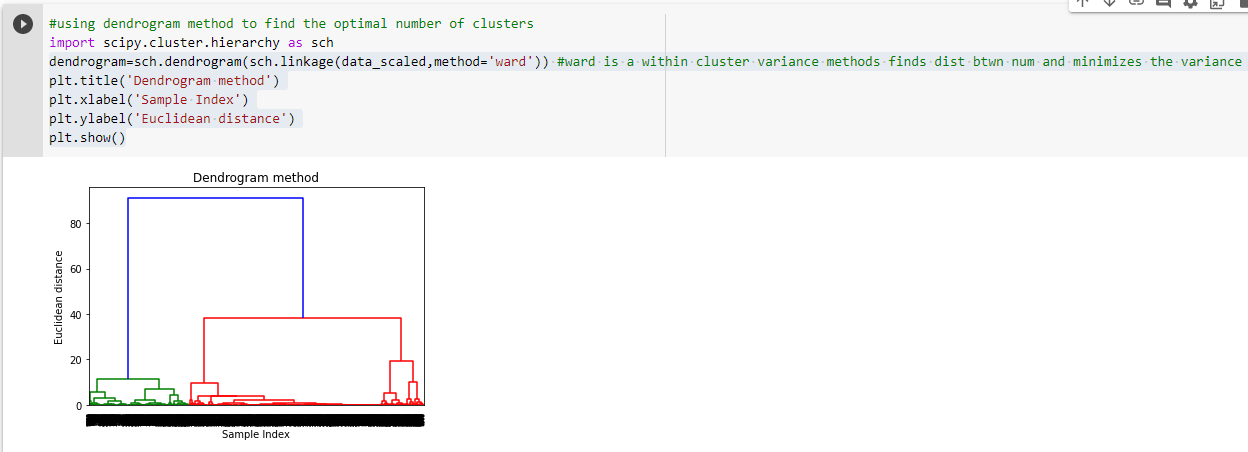
There are multiple products and each product is purchased for a different purpose. **Our aim is to make clusters from this data that can segment similar clients together**.



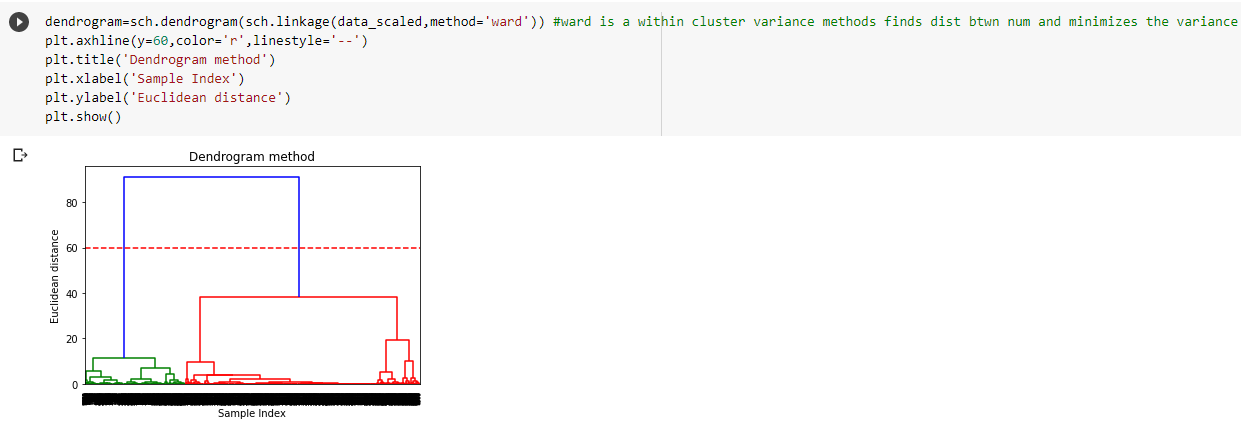
Third step we perform feature scaling and then normalize the data so that the scaling is same for each feature.



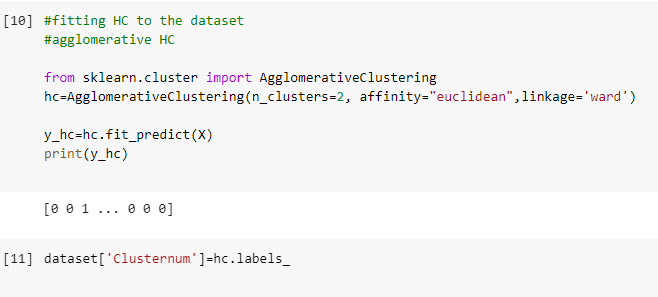
Fourth step we generate a dendrogram to help us decide with the number of clusters.



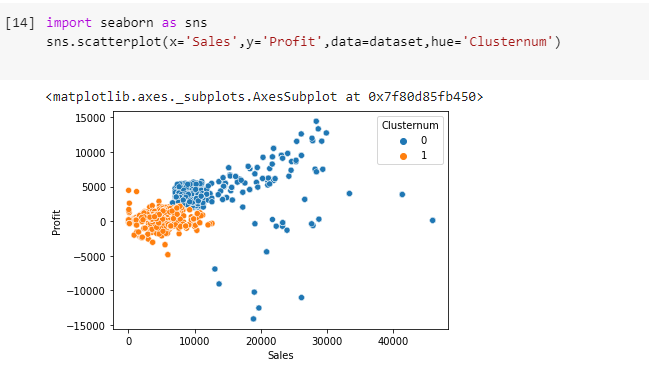
The X-axis contains the samples and the y axis contains the Euclidean distance. The vertical line with the maximum distance is the blue line so we can decide on the threshold value to be 60 and cut the dendrogram.



We have two clusters at this point and now we apply the agglomerative clustering.



We can see the values of 0s and 1s in the output since we defined 2 clusters. 0 represents the points that belong to the first cluster and 1 represents points in the second cluster. We will now visualize the two clusters.

****

**LAB ASSIGNMENT 5**

**Problem Set:-**

Use Numeric, Pandas, and other Python libraries

a.       Write the Python code to compute entropy and information gain

b.       Write the Python code to demonstrate a conditional probability

c.        Write the Python code to compute Euclidean Distance between data points

d.       Write the Python code to calculate covariance matrix, Eigenvalues, and Eigenvectors

e.        Write the Python code to calculate the following

a.       Accuracy

b.       Misclassification

c.        Type-1 and Type-2 error rates

d.       Sensitivity

e.        Specificity

**Answers:**

a) We import the necessary libraries and load the dataset.



2) To compute the counts of each unique value in the column

Counts = np.bincount (column)

3) We divide by the total column length to get the probability of each counts

probabilities = counts / len(column)

4) We can initialize a variable name entropy and set it to 0

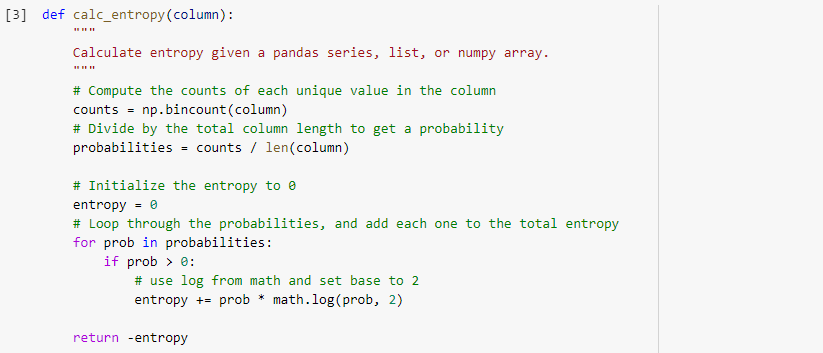
Entropy=0

5) Loop through the probabilities and add each one to the total entropy.

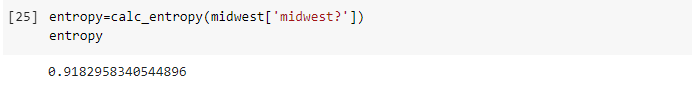
We use log from math module and set base 2 to compute the entropy.

6) In the final step we return the negative entropy

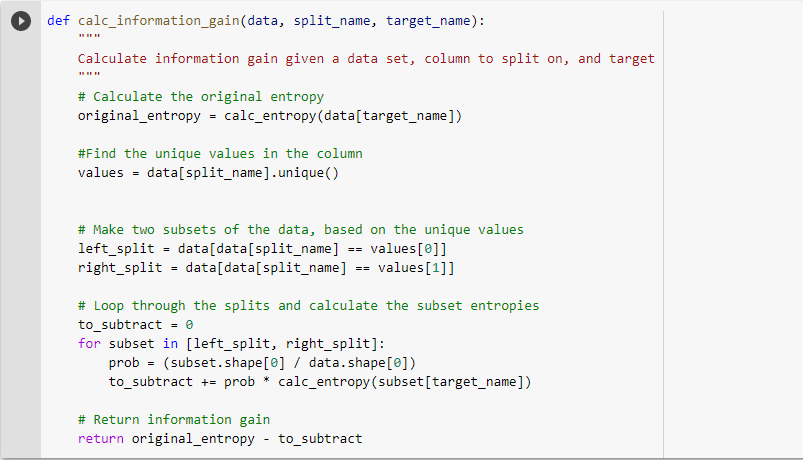
Return –entropy



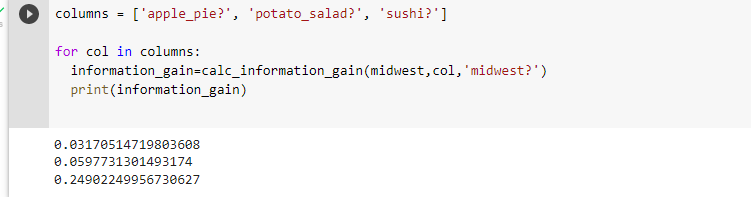
**Entropy result:**

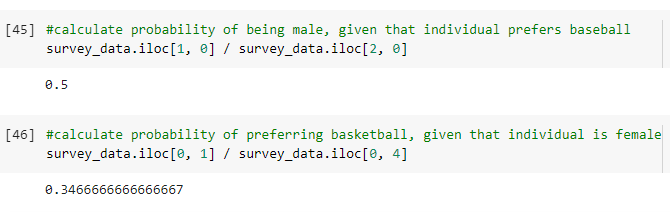
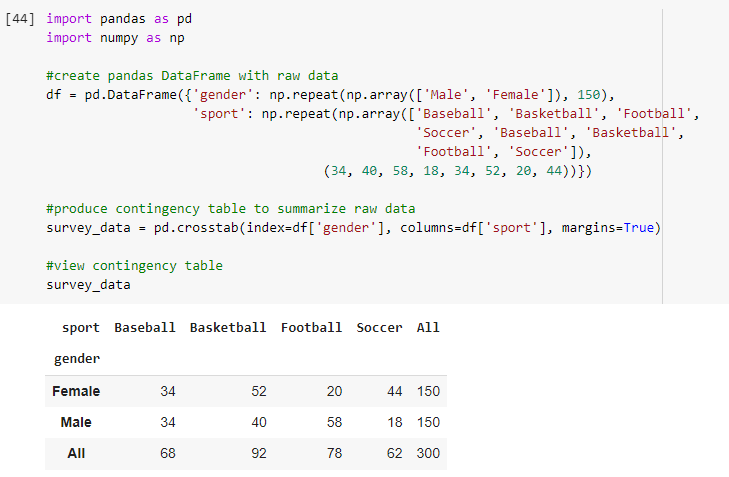


**Information gain**

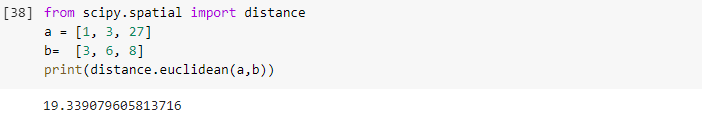
****

**Information Gain result:**

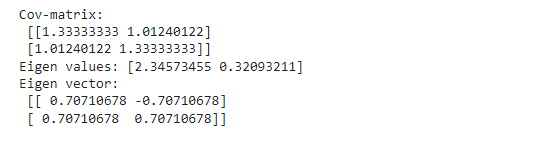
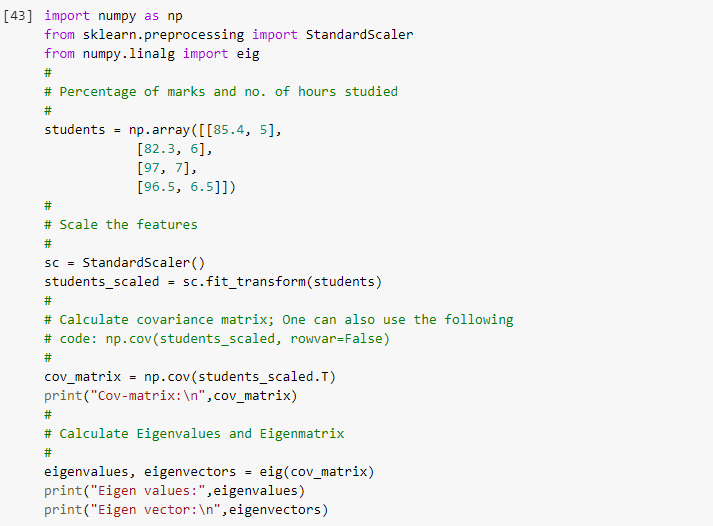
****

b) 

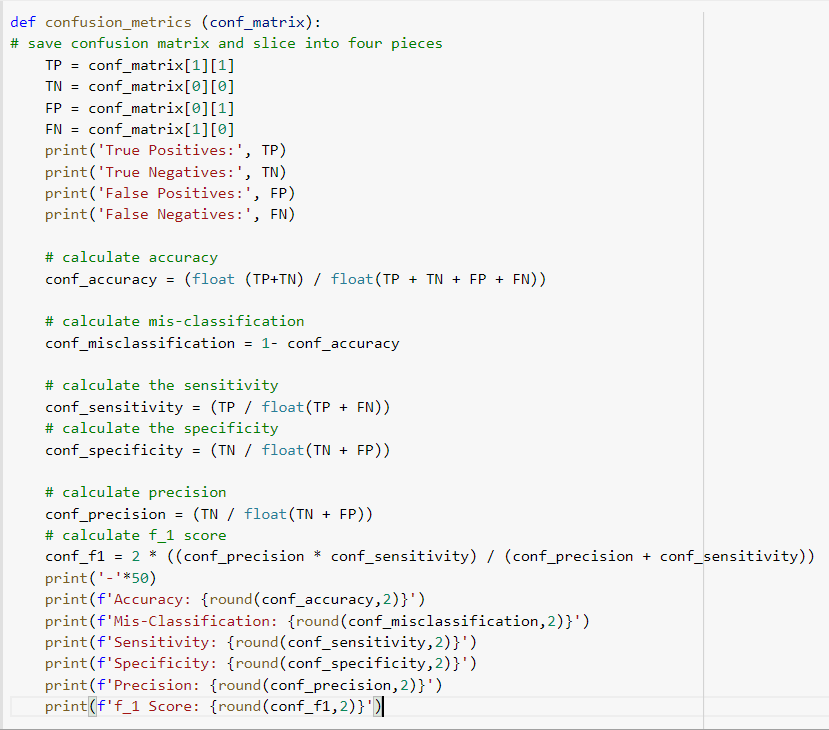
c) We can import the distance module from the Scipy.spatial library to calculate the Euclidean distance.



d) We can calculate the covariance matrix using numpy library and Eigenvalues and Eigenvectors from “linalg” library and importing “eig” module.



e) In the below code we need to pass the confusion matrix to the defined function.



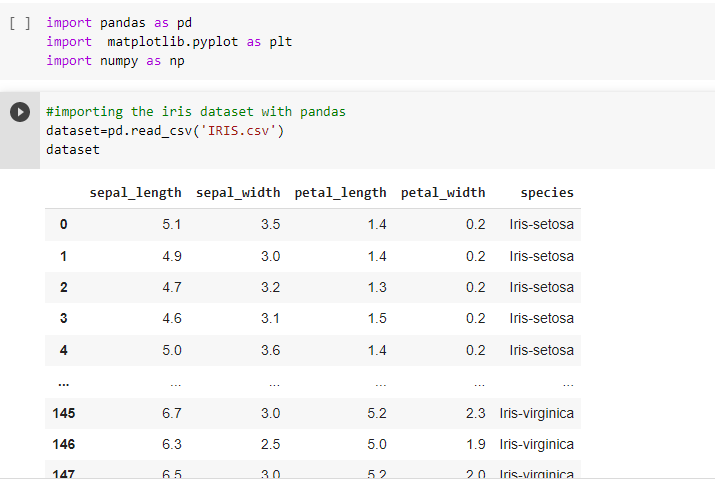
**LAB ASSIGNMENT 6**

Download iris dataset from UCI machine learning repository and use the dataset to develop the following classifiers and find the accuracy of the model. Compare and comment on the results:

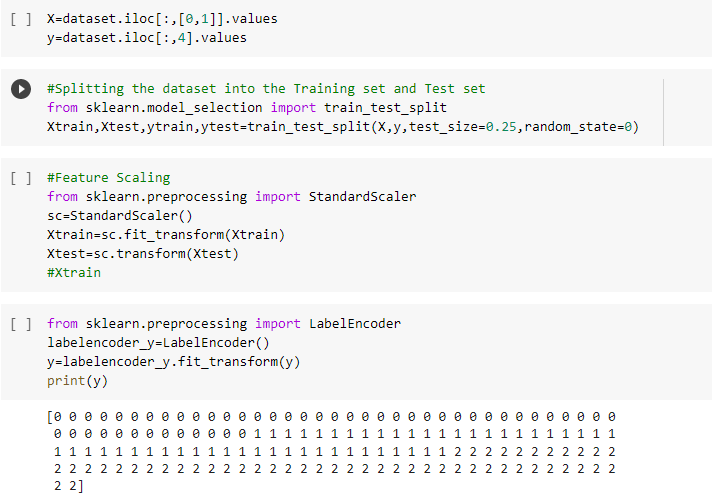
1. Decision Tree Classifier
2. Naïve Bayes Classifier
3. Logistic Regression classifier
4. K-NN classifier
5. Logistic Regression model and apply PAC
6. SVM – linear and non-linear classifiers
7. Random Forest Model
8. Ada Boost Model

**Answer:**

1. Import the libraries and the dataset.

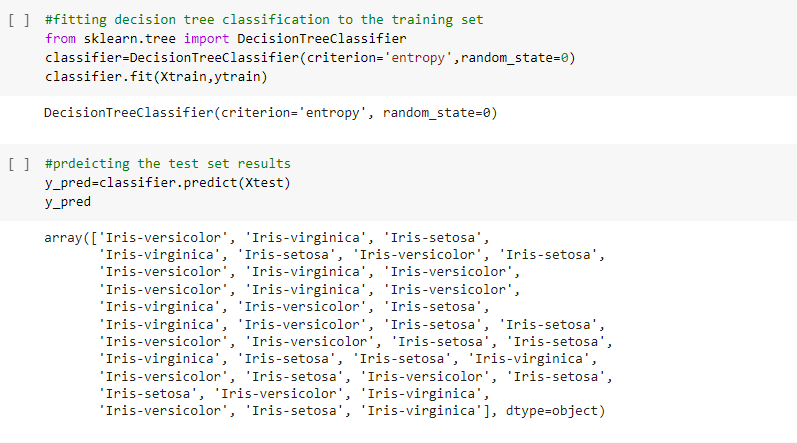
****

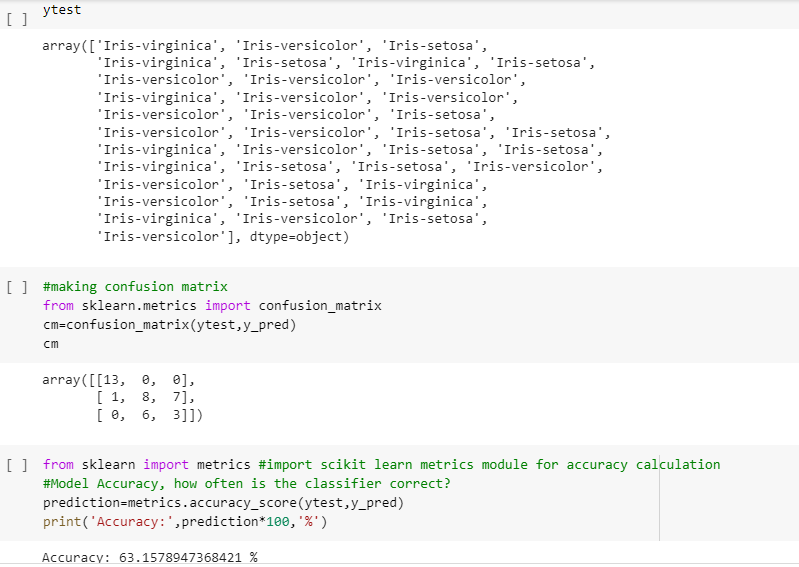
2. Split the dataset into train and test and perform feature scaling, and label encoding.



3. Fit the models and find their accuracy

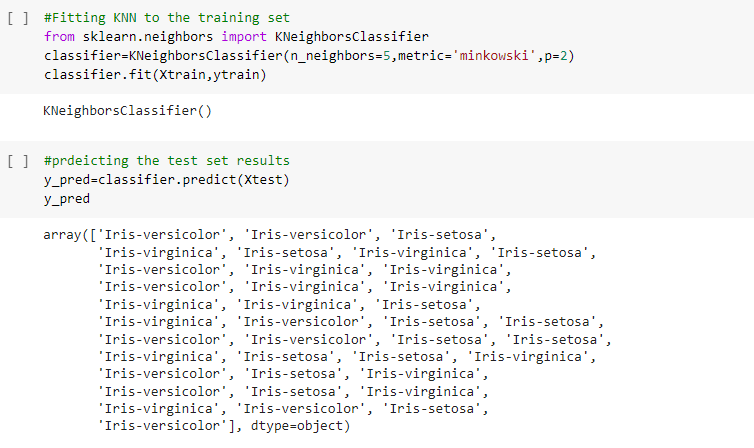
**Decision Tree**

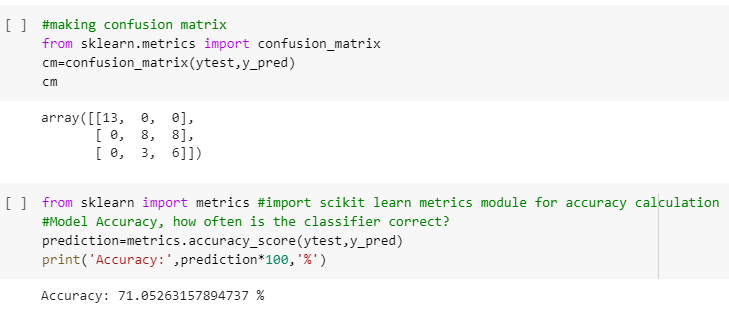
****

****

**Accuracy of DT is 63.15%.**

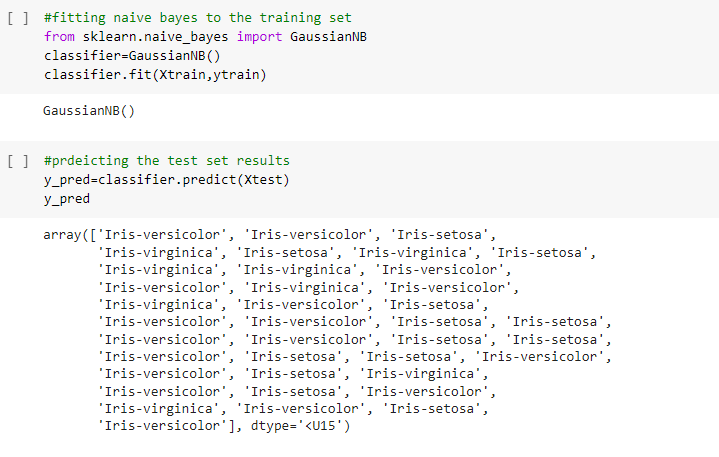
**KNN**

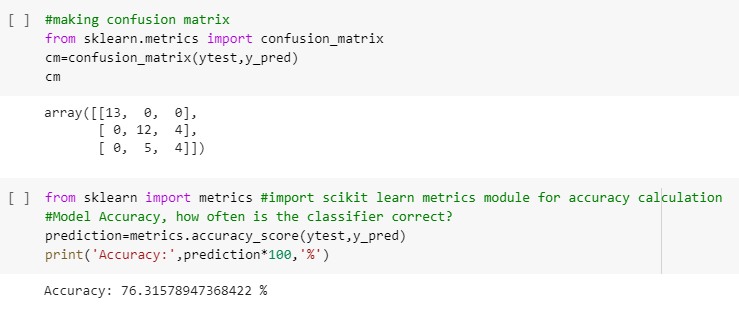
****

****

**Accuracy of KNN is 71.05%.**

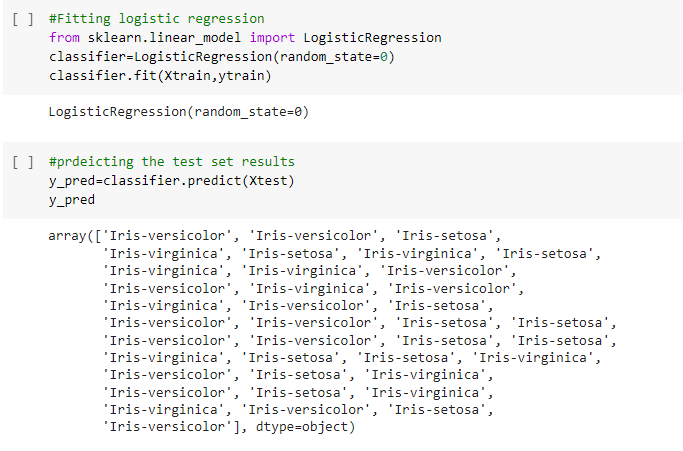
**Naïve Bayes**

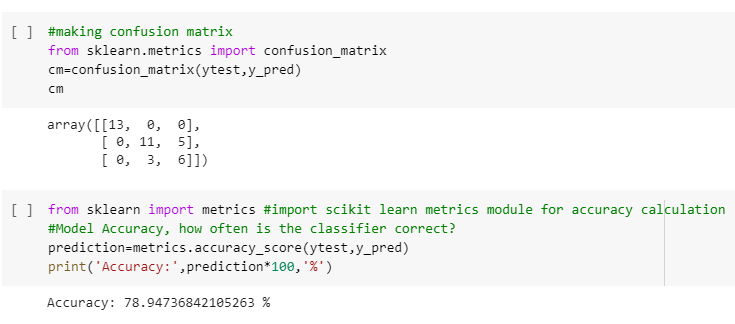
****

****

**Accuracy of Naïve Bayes is 76.31%.**

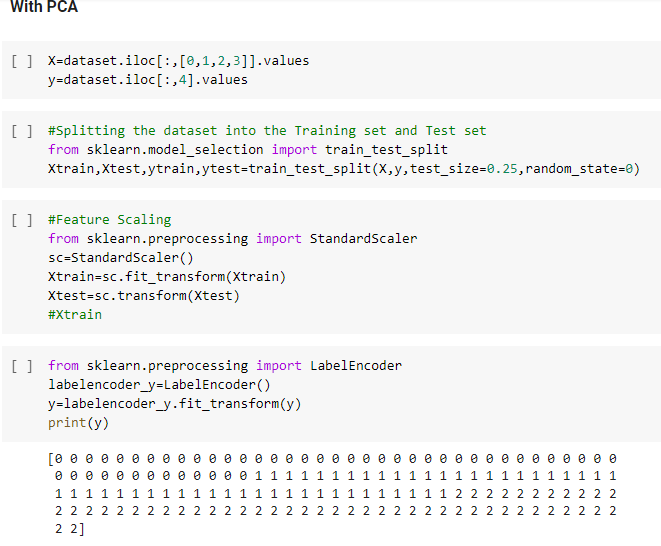
**Logistic Regression**

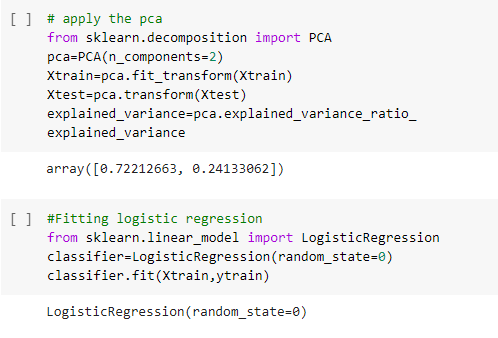
****

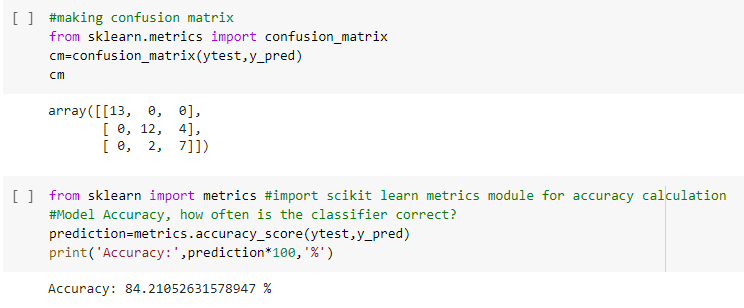
****

**Accuracy of Logistic Regression (without PCA) is 78.94%.**

**Logistic Regression without PCA**

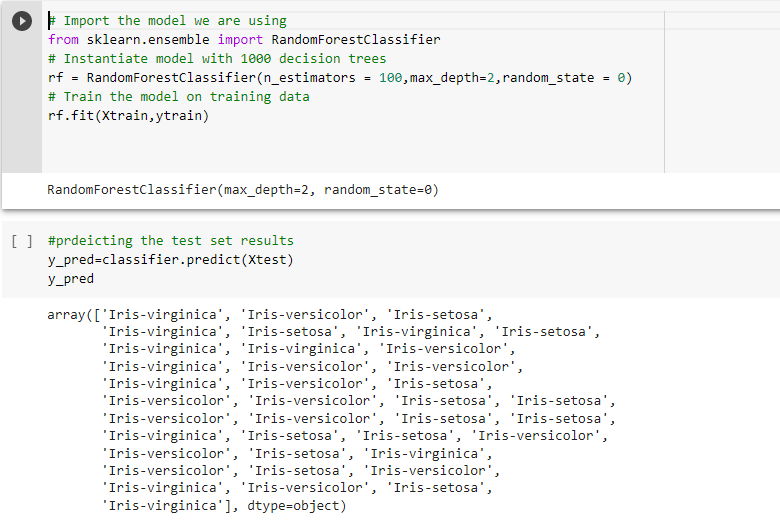
****

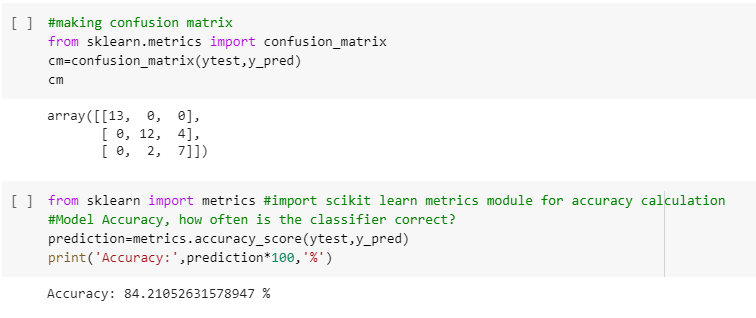
****

****

**Accuracy of Logistic Regression (with PCA) is 84.21%.**

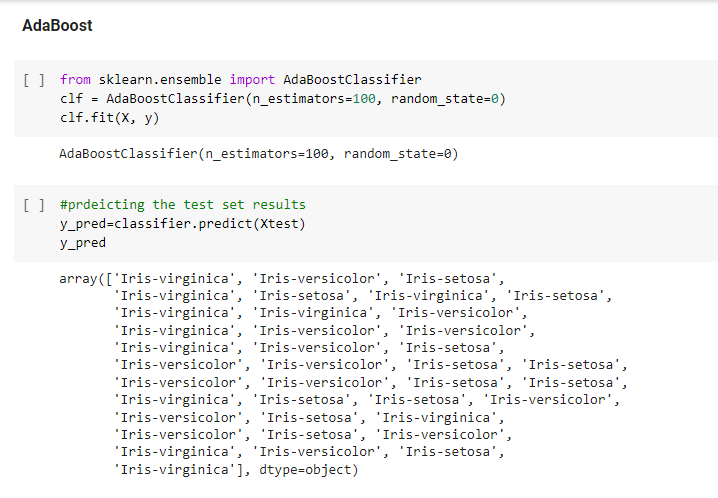
**Random Forest**

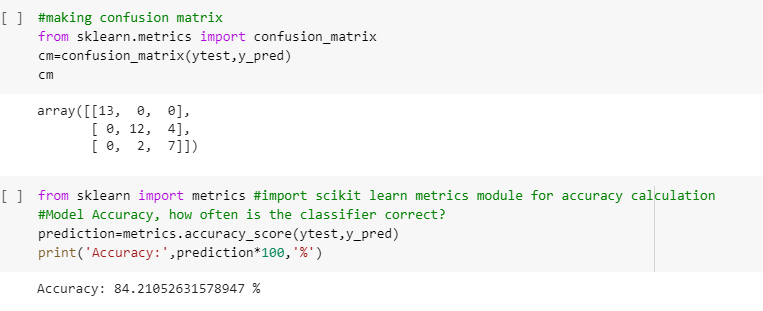
****

****

**Accuracy of Random Forest is 84.21%.**

**AdaBoost**

****

****

**Accuracy of AdaBoost is 84.21%.**

**SVM**

****

****

**Accuracy of SVM is 84.21%.**