**Coimbatore Institute of Technology**

**Pre-Assessment Test – Curnue**

**Task-01**

**Name : Indhu Sree S**

**Roll no : 1832023**

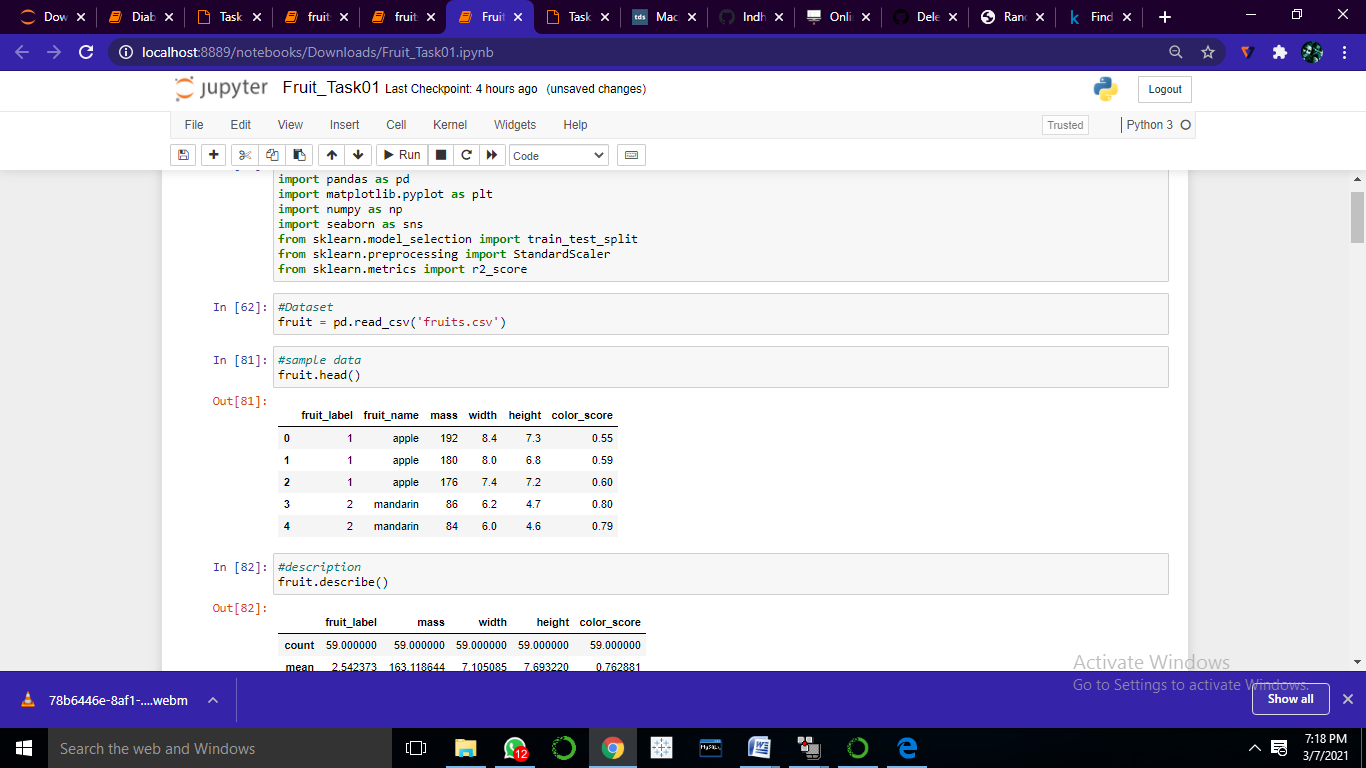
**M.Sc Data Science**

**Ques : SD03Q03**

**Task:**

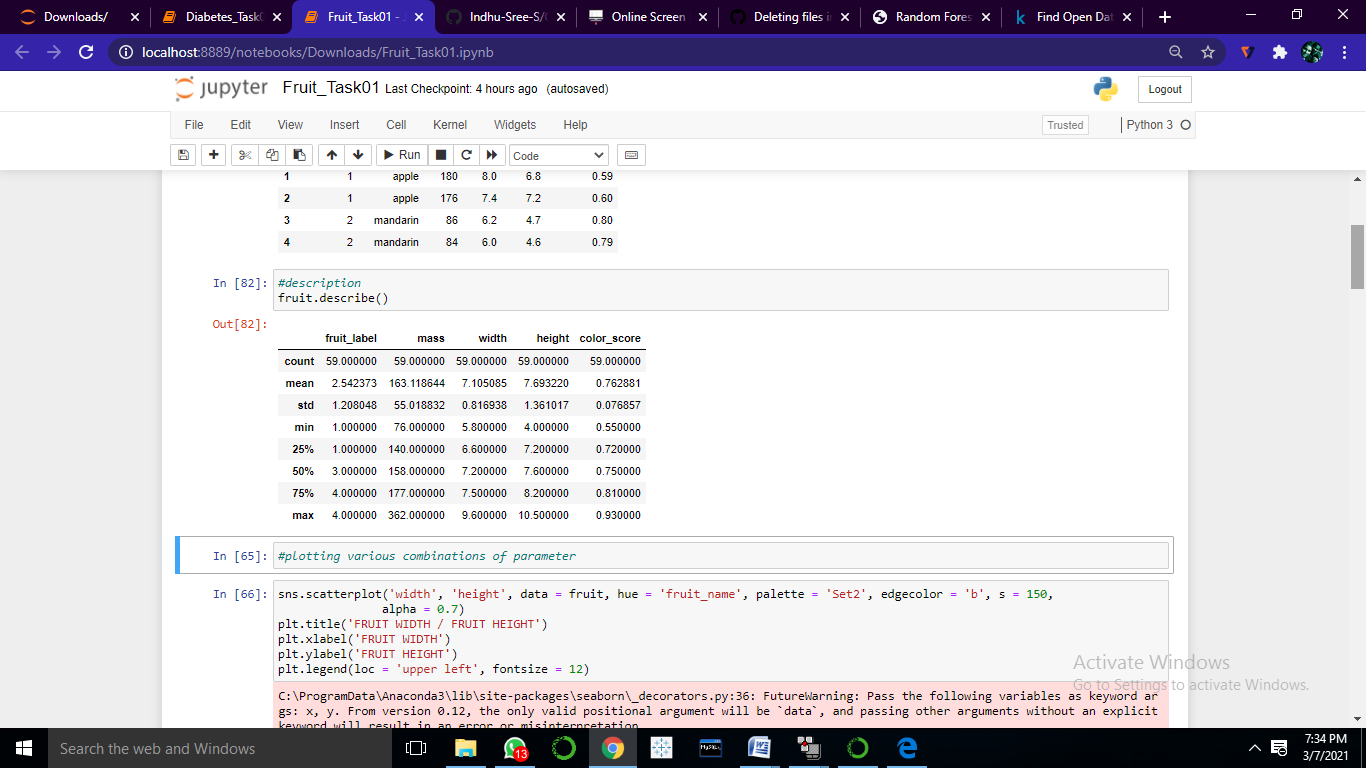
A dataset labelled based on fruit height, width, mass and colour score is given in fruits.xlsx. A classifier based on k Nearest Neighbour (KNN) algorithm is to be crafted for classification.

**Dataset :**



* Mass : mass of the given fruit
* Width : width of the given fruit
* Height : height of the given fruit
* Color\_score : color score of the given fruit
* Fruit\_name : Depending upon the mass, width, height, color score – name of the fruit and label is classified.
* Therefore classification of fruits based on mass, width, height and their color score

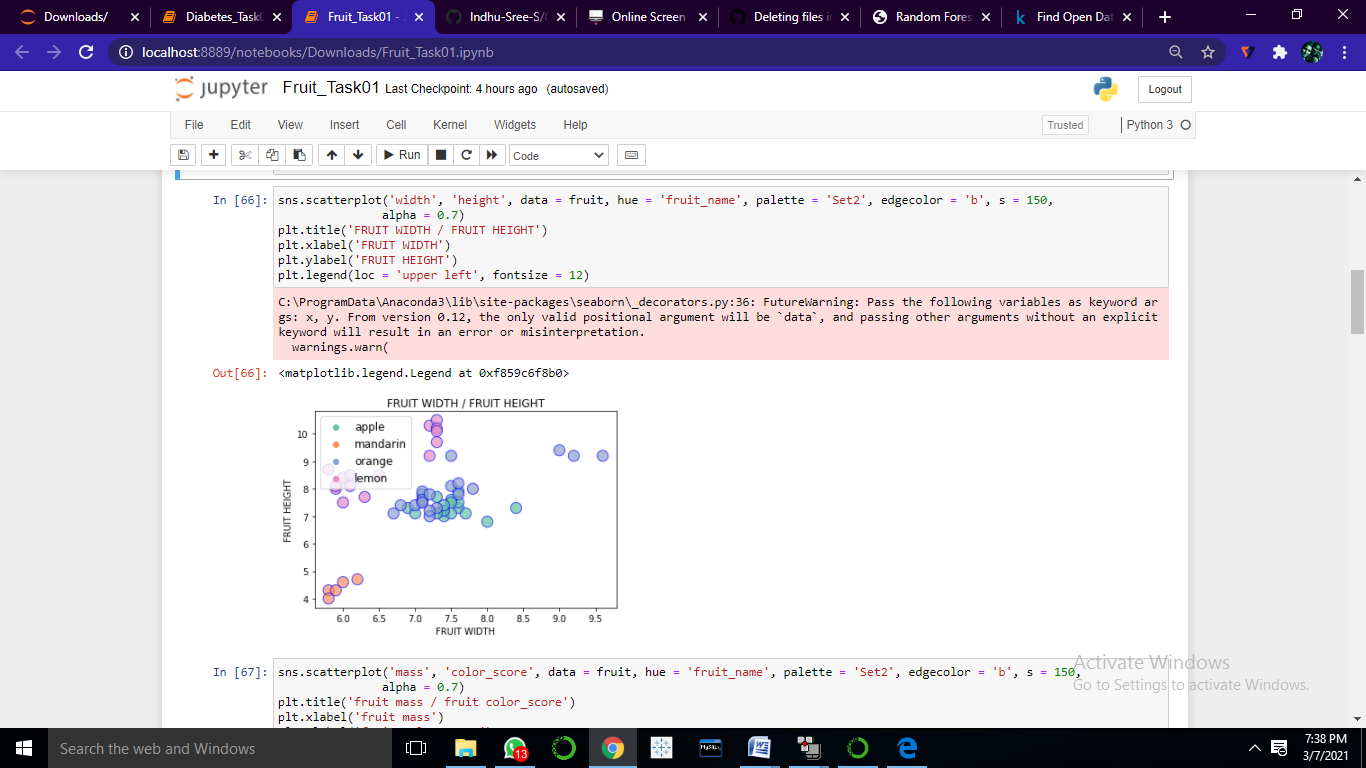
**Description:**



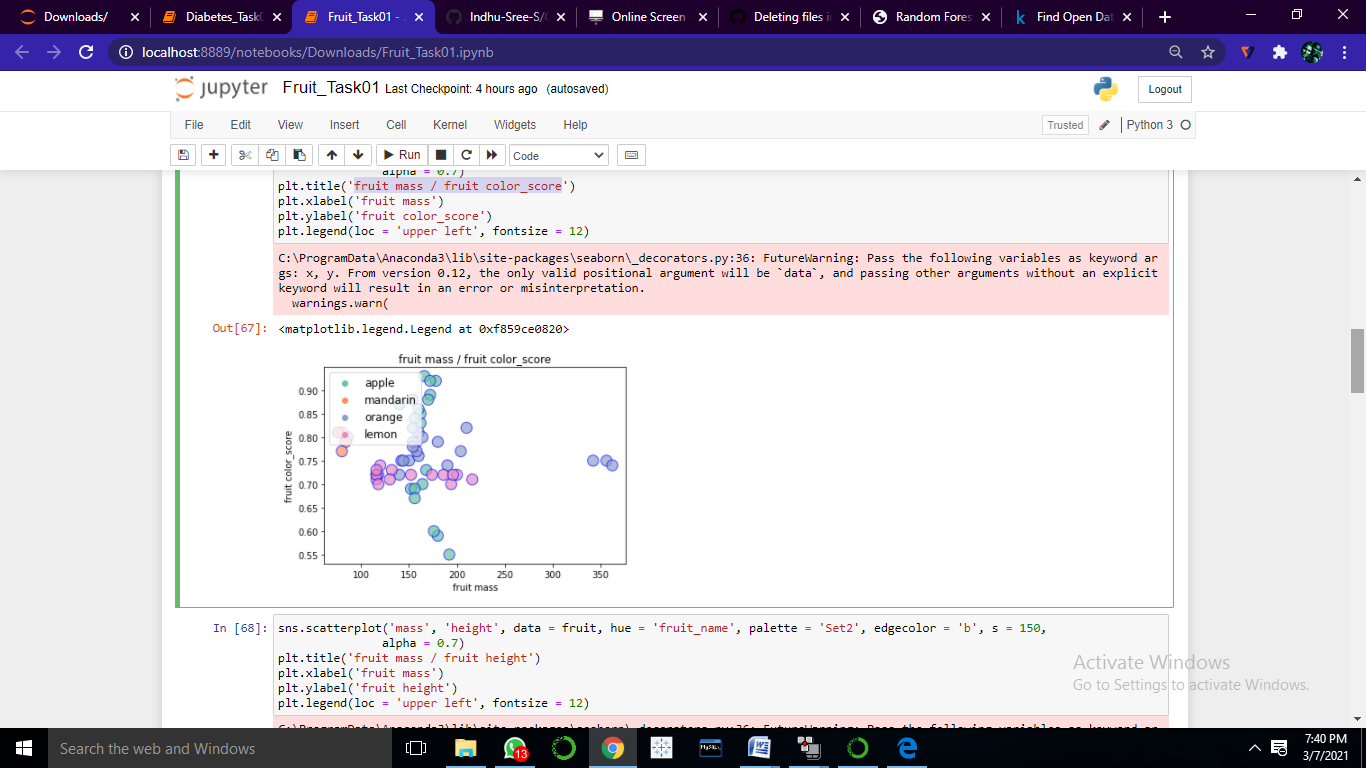
Above table shows that mean, minimum, maximum, quartiles, standard deviation and count of each variable that have numerical values.

**Visualization:**

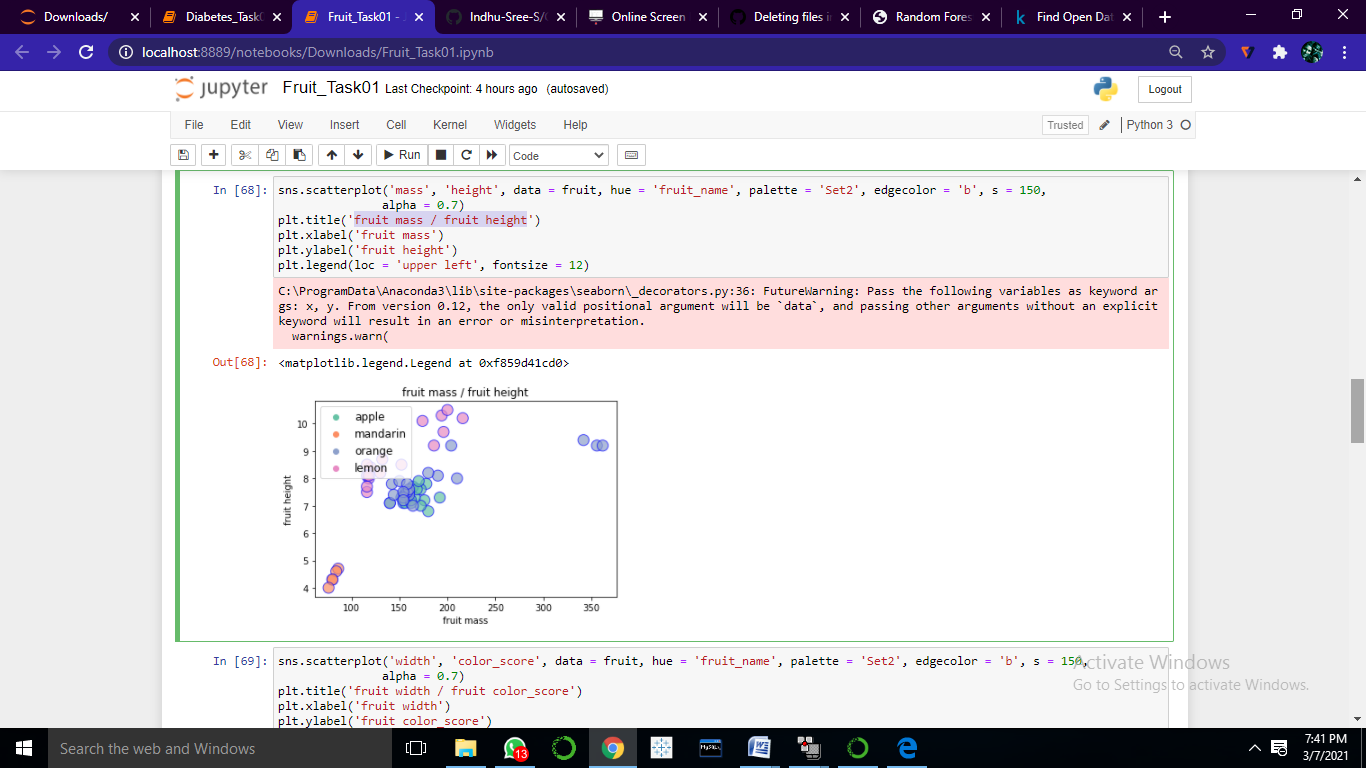
* Relationship between (FRUIT WIDTH / FRUIT HEIGHT)



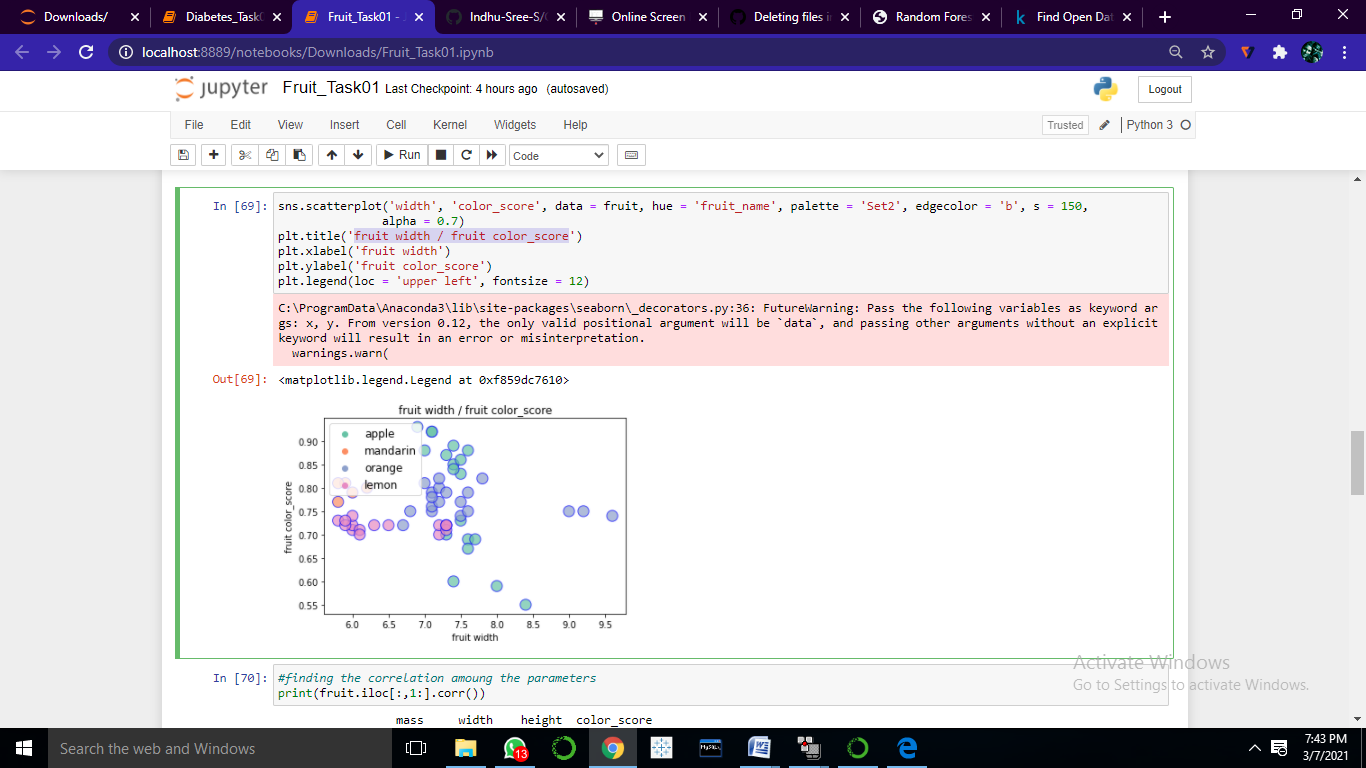
* Relationship between (fruit mass / fruit color\_score)



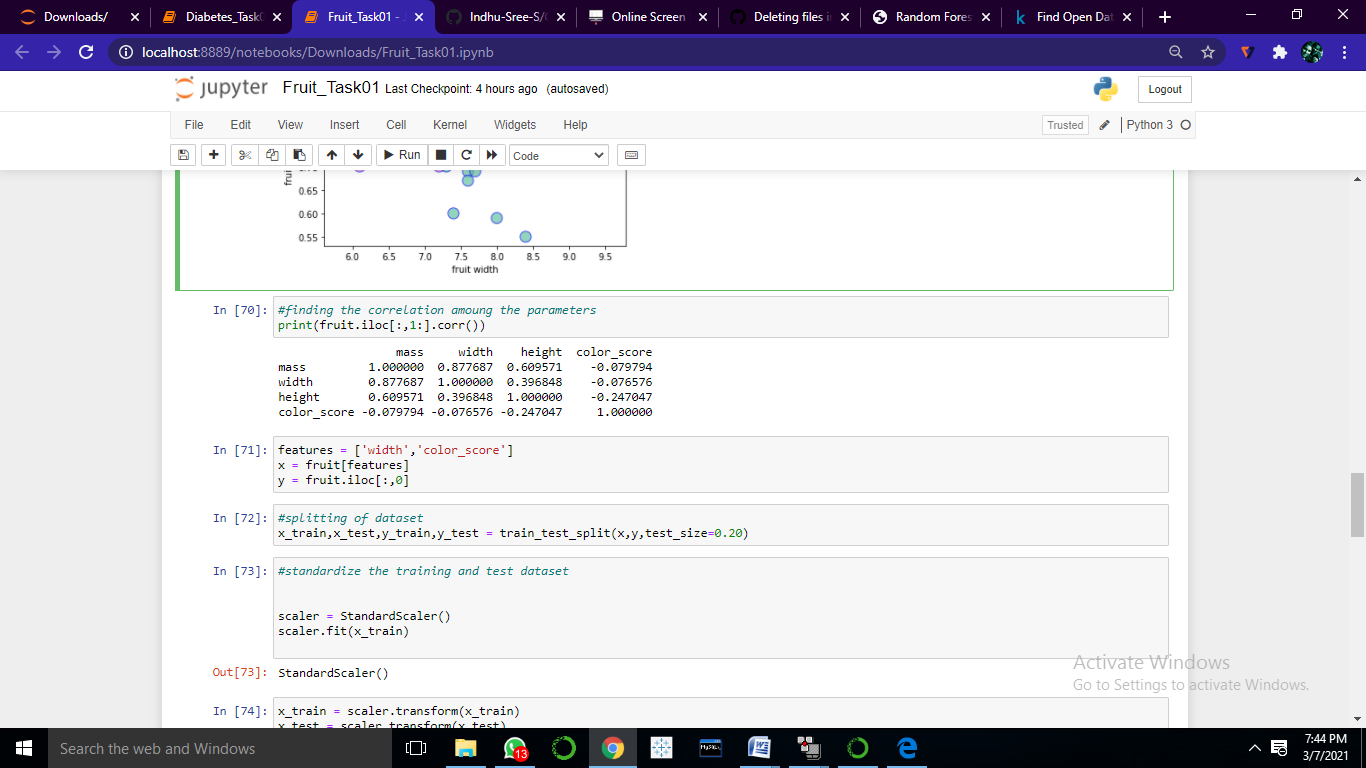
* Relationship between(fruit mass / fruit height)



* Relationship between(fruit width / fruit color\_score)



**Correlation:**

****

Above table shows the correlation between variables. Correlation means association - more precisely it is a measure of the extent to which two variables are related. Here Color score values are lowly correlated with other variables.

**k-Nearest Neighbors:**

* Calculate Euclidean Distance.
* Get Nearest Neighbors.
* Make Predictions.

**Euclidean Distance:**

* 1. Rows of data are mostly made up of numbers and an easy way to calculate the distance between two rows or vectors of numbers is to draw a straight line

Euclidean Distance = sqrt(sum i to N (x1\_i – x2\_i)^2)

def euclid\_dist(X1,X2):

dist = np.sum((X1 - X2)\*\*2)

return np.sqrt(dist)

**Function of knn predict:**

def knn\_predict(x\_train, x\_test, y\_train, y\_test, k):

# Counter to help with label voting

from collections import Counter

# Make predictions on the test data

# Need output of 1 prediction per test data point

y\_pred = []

for test\_point in x\_test:

distances = []

for train\_point in x\_train:

distance = euclid\_dist(test\_point, train\_point)

distances.append(distance)

# Storing distances in a dataframe

df\_dists = pd.DataFrame(data=distances, columns=['dist'],

index=y\_train.index)

# Sort distances and considering the k closest points

df\_nn = df\_dists.sort\_values(by=['dist'], axis=0)[:k]

# Create counter object to track the labels of k closest neighbors

counter = Counter(y\_train[df\_nn.index])

# Get most common label of all the nearest neighbors

prediction = counter.most\_common()[0][0]

# Append prediction to output list

y\_pred.append(prediction)

return y\_pred

* Neighbors for a new piece of data in the dataset are the k closest instances, as defined by our distance measure.
* To locate the neighbors for a new piece of data within a dataset we must first calculate the distance between each record in the dataset to the new piece of data. We can do this using our distance function prepared above.
* Once distances are calculated, we must sort all of the records in the training dataset by their distance to the new data. We can then select the top k to return as the most similar neighbors.

**Prediction**

47 4

18 1

25 3

21 1

12 1

4 2

31 3

20 1

43 4

37 3

36 3

38 3

Name: fruit\_label, dtype: int64

[4, 3, 3, 1, 1, 2, 3, 1, 4, 3, 3, 3]

Accuracy: 0.9166666666666666

r 2\_score : 0.7318435754189945

**Interpretation:**

Fruits’s dataset is classified by using K-Nearest Neighbor Classifier. It has an accuracy of 92% for this classification. The r2 score of the classifier is having 73%. The best K value is 16, which is calculated by r score value and the minimum error value. Thus the K-Nearest Neighbor classifier was implemented in Fruit to classify and predict them.