**SD03Q07**

**Problem Statement 1: -**

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.

• Generate scatter plots for various combination of parameters and do the feature

engineering meaning thereby which parameters of best suited to build the classifier.

• Split the data into test and training split.

• Building a classifier using KNN from scratch.

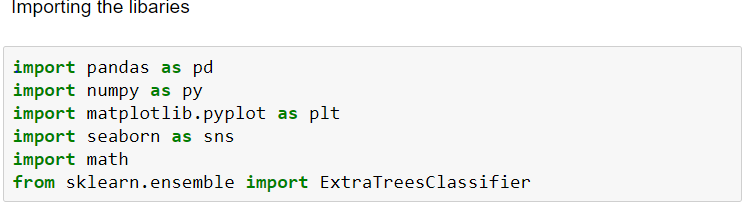
• Figure out the best value of k with highest r\_score.

• Run at least three test cases on the parameter and assess the fruit using the classifier.

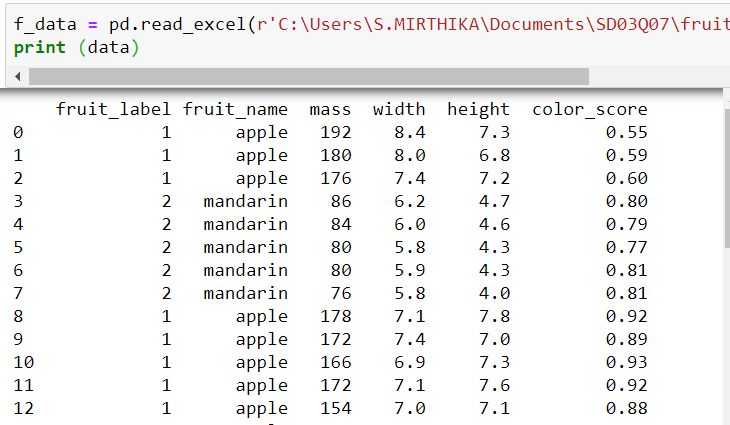
• Only use python

**Code:**

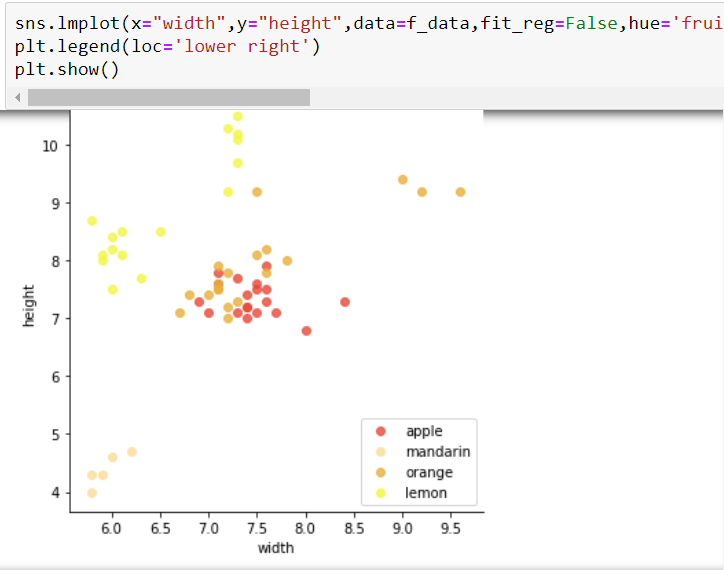
1. Import the required libraries.



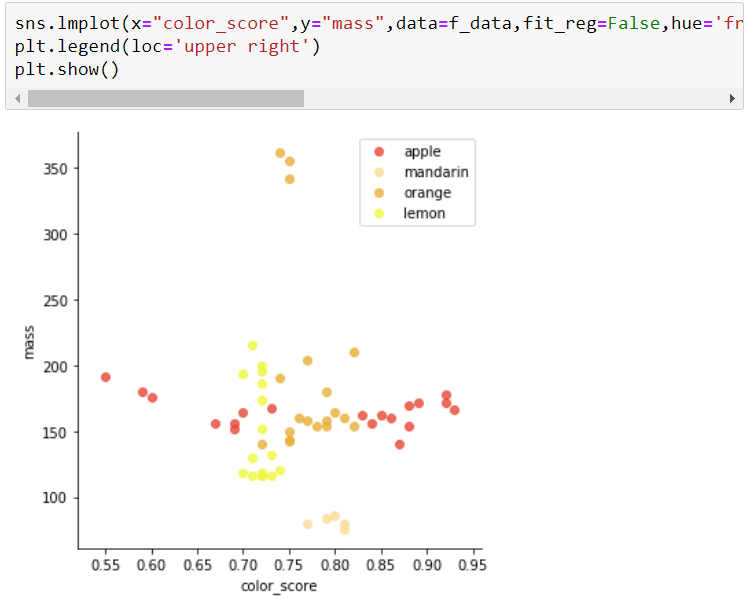
2. Read the fruits.xlxs dataset using read\_excel



3. Plot a scatter plot for the variable’s width, height and fruit\_name



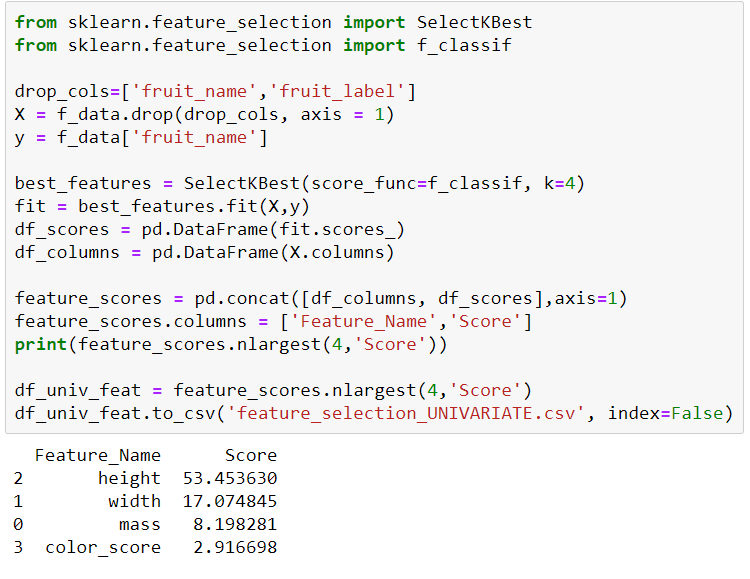
4. Plot a scatter plot for the variable’s color\_score, mass, fruit\_name



5. Univariate Selection

Statistical tests can be used to select those features that have the strongest relationship with the output variable.

The scikit-learn library provides the SelectKBest class that can be used with a suite of different statistical tests to select a specific number of features.

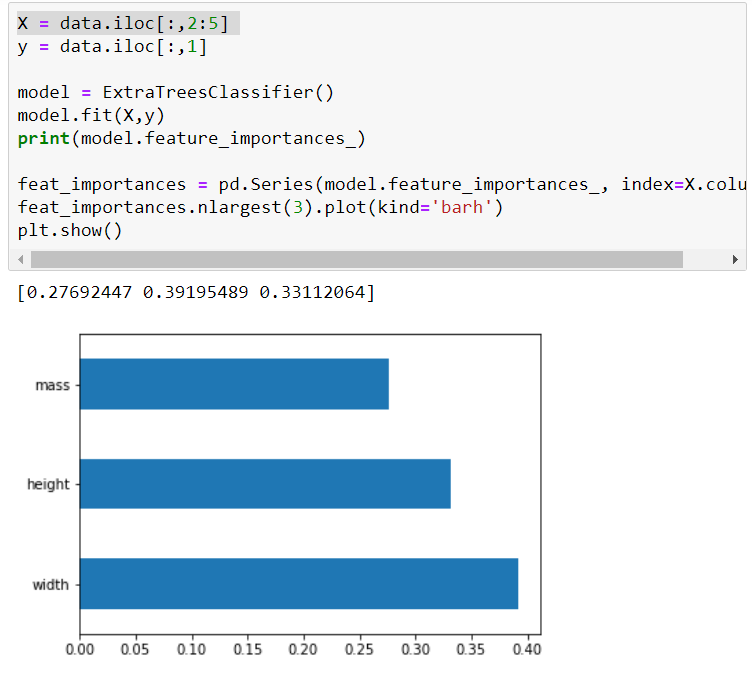


6. Feature Importance

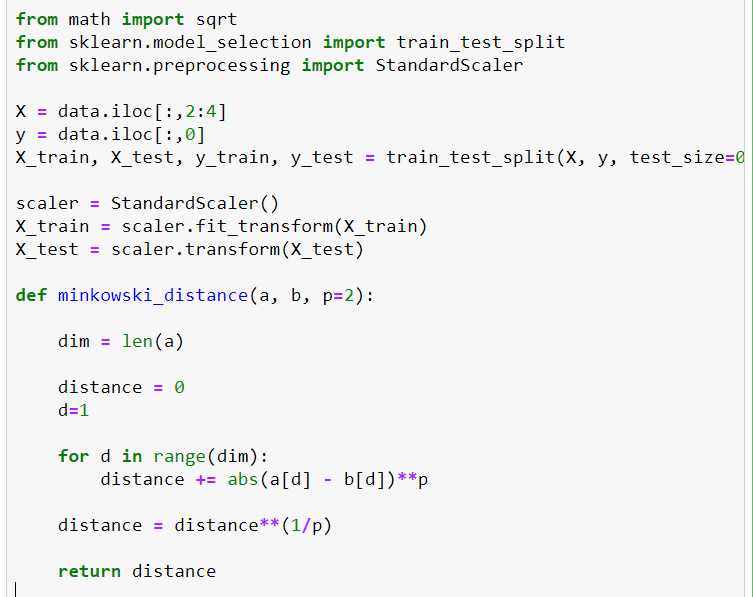
You can get the feature importance of each feature of your dataset by using the feature importance property of the model.

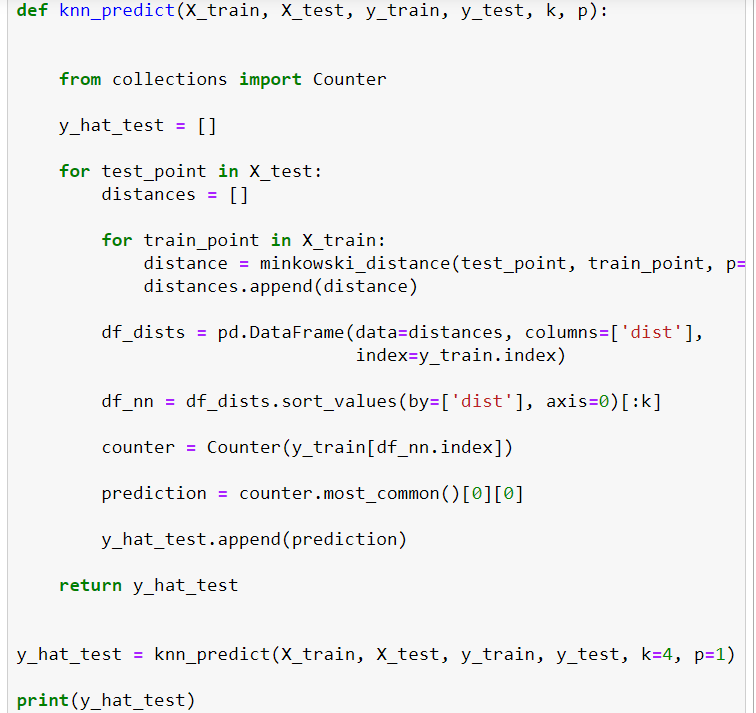
Feature importance gives you a score for each feature of your data, the higher the score more important or relevant is the feature towards your output variable.

Feature importance is an inbuilt class that comes with Tree Based Classifiers, we will be using Extra Tree Classifier for extracting the top 3 features for the dataset.



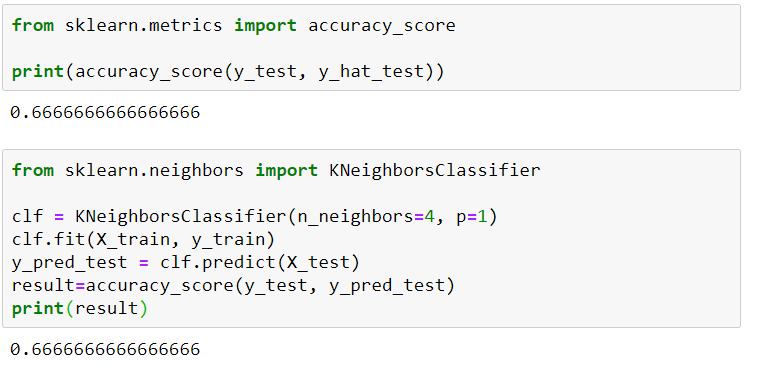
7. The next step is to split the dataset into train and test data with test\_size=0.3 and create a function for calculating Minkowski distance and create a function called knn\_predict that takes in all of the training and test data, k, and p, and returns the predictions my KNN classifier makes for the test set (y\_hat\_test).

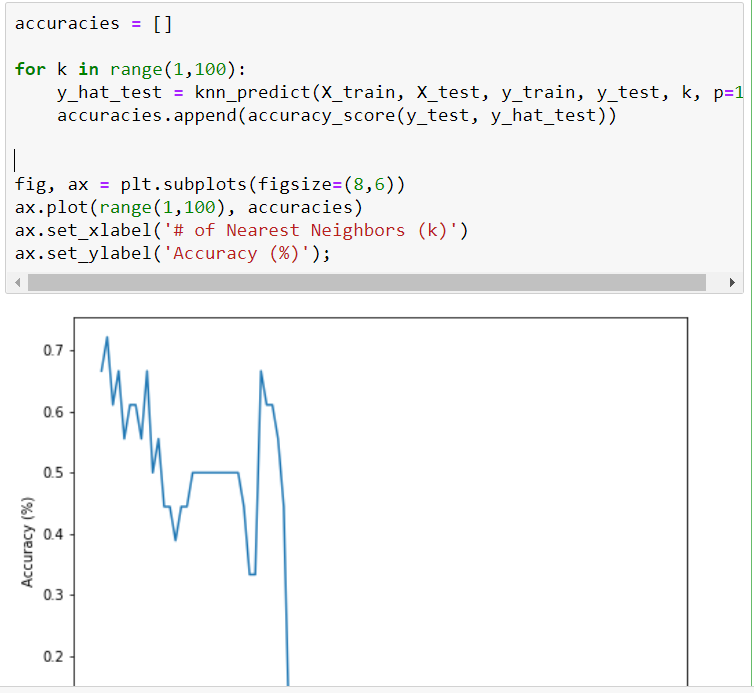






8. I used accuracy\_score from sklearn to get the accuracy score of the model.





**Problem Statement 2: -**

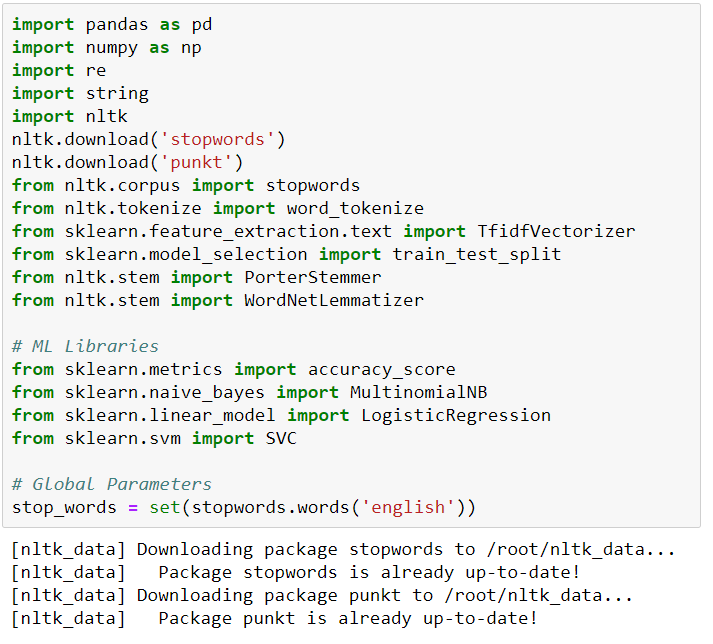
Perform Sentimental Analysis from scratch using python. In any Twitter post, there are various

sentiments behind it. So perform sentimental analysis and find out the different sentiments in

the dataset.

**Code:**

1. Let’s load the necessary libraries.



2. I defeined a function load\_dataset to read our dataset and I have created remove\_unwanted\_cols function to remove unwanted columns from our dataset and I have defined preprocess\_tweet\_text function to preprocess the tweet data by removing the urls, users, punctuations, stopwords.

Now I want to see how well the given sentiments are distributed across the train dataset. One way to accomplish this task is by understanding the common words by plotting wordclouds.

A wordcloud is a visualization wherein the most frequent words appear in large size and the less frequent words appear in smaller sizes.

