MANI BARATHI SP S(1832031)

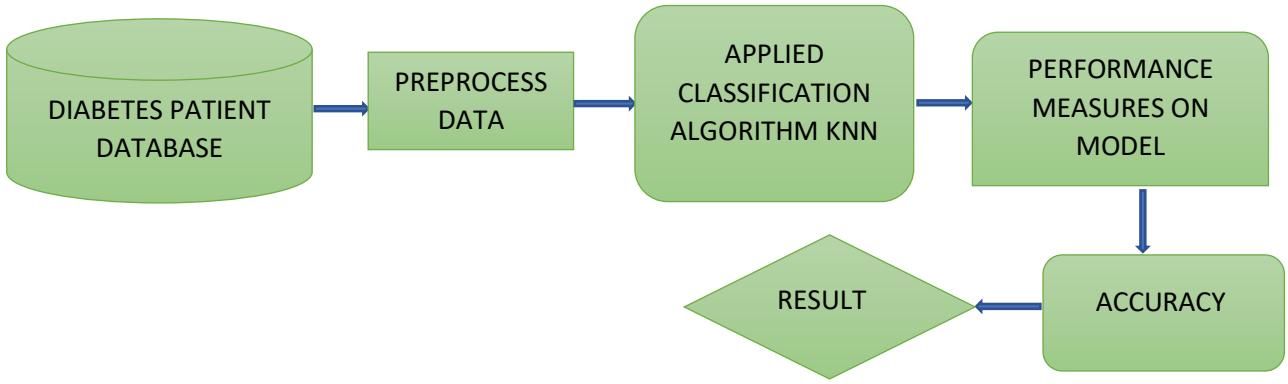
DEPARTMENT OF DATA SCIENCE

CLASSIFICATION OF KNN ALGORITHM USING KNN ALGORITHM

PROBLEM DESCRIPTION:

Aim is to classify a labelled dataset of fruit based on fruit height, width, mass and colour score is given in fruits.xlsx using the k Nearest Neighbour (KNN) algorithm.

METHODOLOGY:



DATA DESCRIPTION:

It consists of 59 rows and 6 columns.

fruit_label – represents the fruit apple=1,Madarin=2, Orange=3,lemon=4

fruit_name – contains name of the fruit

mass – mass of the fruit

width – width of the fruit

height – height of the fruit

color_score – color of the fruit

LOADING DATA:

To classify the type of fruit , I'm going to use KNN classifer for fruit dataset imp orted used pandas library .The data set has measurements (fruit_label , fruit_name , mass , width height , color_score)

SAMPLE DATASET:

	fruit_label	fruit_name	mass	width	height	color_score
0	1	apple	192	8.4	7.3	0.55
1	1	apple	180	8.0	6.8	0.59
2	1	apple	176	7.4	7.2	0.60
3	2	mandarin	86	6.2	4.7	0.80
4	2	mandarin	84	6.0	4.6	0.79
5	2	mandarin	80	5.8	4.3	0.77
6	2	mandarin	80	5.9	4.3	0.81
7	2	mandarin	76	5.8	4.0	0.81
8	1	apple	178	7.1	7.8	0.92
9	1	apple	172	7.4	7.0	0.89
10	1	apple	166	6.9	7.3	0.93
11	1	apple	172	7.1	7.6	0.92
12	1	apple	154	7.0	7.1	0.88
13	1	apple	164	7.3	7.7	0.70
14	1	apple	152	7.6	7.3	0.69

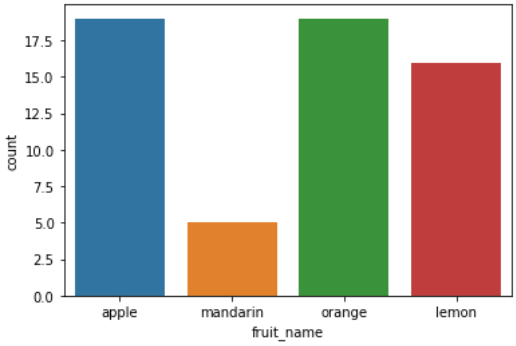
DATA EXPLORATION:

Data exploration is the initial step in data analysis, where users explore a large data set in an unstructured way to uncover initial patterns, characteristics, and points of interest. Data exploration can use a combination of manual methods and automated tools such as data visualizations, charts, and initial reports.

	fruit_name
apple	19
lemon	16
mandarin	5
orange	19

- From the dataset we got a information that there are 19 apples,16 lemons, 5 mandarin and 19 oranges.

BAR GRAPH:



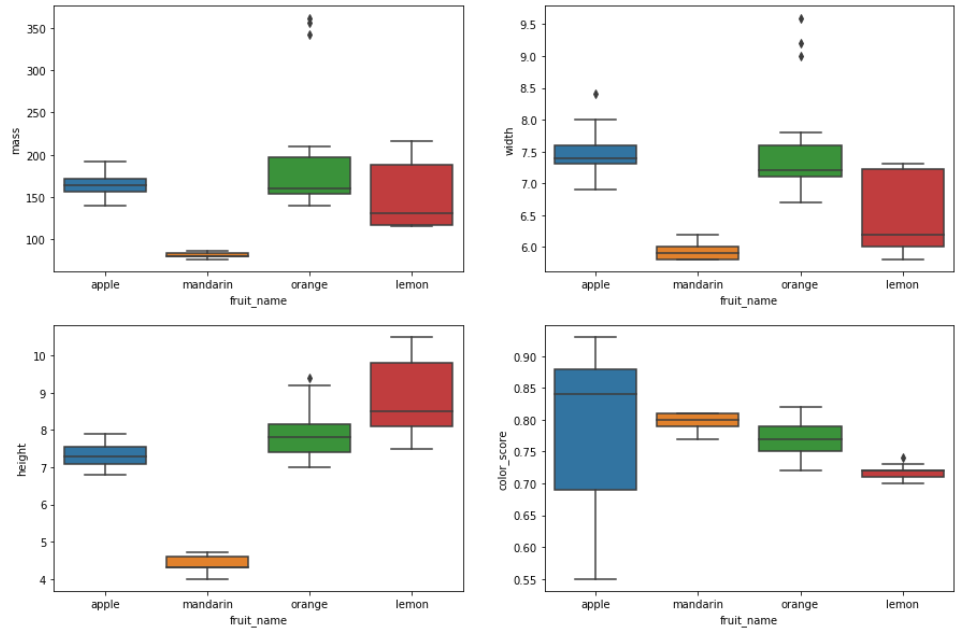
The above bar graph shows the counts of each fruit in the dataset.

DATA DESCRIPTION:

	fruit_label	mass	width	height	color_score
count	59.000000	59.000000	59.000000	59.000000	59.000000
mean	2.542373	163.118644	7.105085	7.693220	0.762881
std	1.208048	55.018832	0.816938	1.361017	0.076857
min	1.000000	76.000000	5.800000	4.000000	0.550000
25%	1.000000	140.000000	6.600000	7.200000	0.720000
50%	3.000000	158.000000	7.200000	7.600000	0.750000
75%	4.000000	177.000000	7.500000	8.200000	0.810000
max	4.000000	362.000000	9.600000	10.500000	0.930000

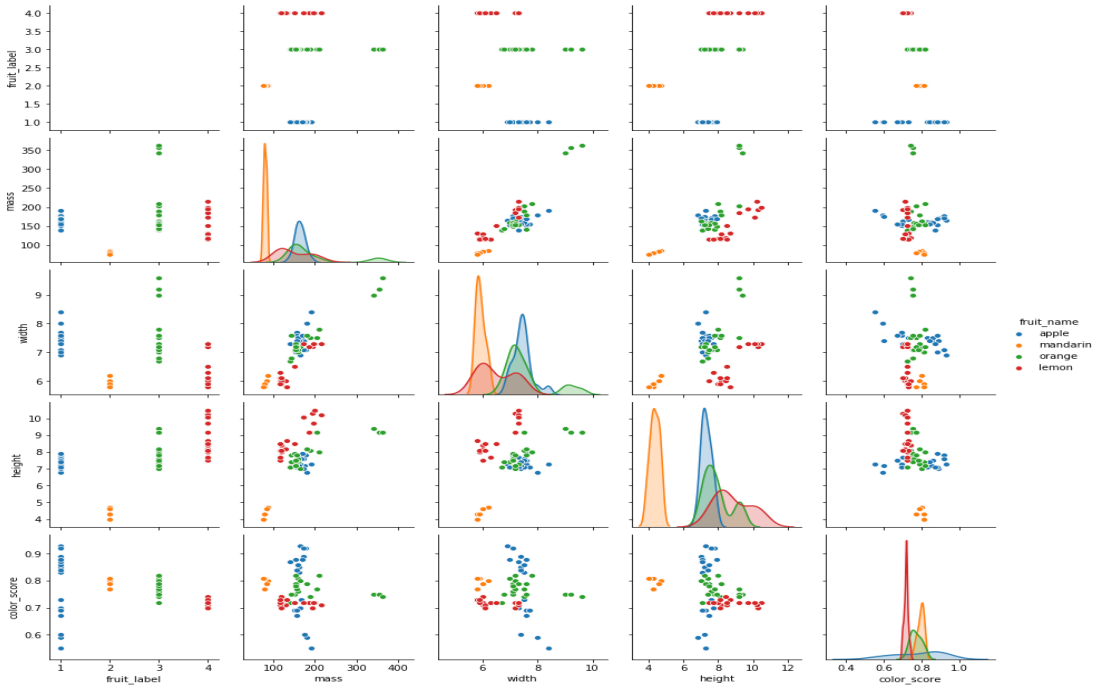
The above table shows the maximum , count , mean , standard deviation , minimum for each columns in the dataset.

BOX PLOT:



A box plot or boxplot (also known as box and whisker plot) is a type of chart often used in explanatory data analysis. Box plots visually show the distribution of numerical data and skewness through displaying the data quartiles (or percentiles) and averages. Box plot is mainly used for the outlier identification. Here in the dataset there is no outlier present in the dataset.

PAIR PLOT:



The pairplot is a kind of a comparison plot here we are compared fruit_label with fruit_name , mass , width height , color_score and so on. It says how the variables are related to each other.

K-NEAREST NEIGHBOR(KNN) ALGORITHM:

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm. K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems. K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.

STEPS:

- **Step-1:** Select the number K of the neighbors
- **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
- **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
- **Step-4:** Among these k neighbors, count the number of the data points in each category.
- **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
- **Step-6:** Our model is ready.

DISTANCE BETWEEN TWO POINTS:

First, I define a function called **minkowski_distance**, that takes an input of two data points (**mass and the witdh**) and a Minkowski power parameter **p,** and

returns the distance between the two points. Note that this function calculates distance exactly like the Minkowski formula I mentioned earlier. By making **p** an adjustable parameter, I can decide whether I want to calculate Manhattan distance (p=1), Euclidean distance (p=2), or some higher order of the Minkowski distance.

The calculated distance between two points is 12.94.

DISTANCE FUNCTION TO GET DISTANCE BETWEEN A TEST POINT AND ALL KNOWN DATA POINTS:

	Dist
0	197.85
1	184.91
2	180.70
3	87.00
4	84.69

It is the calculated distance between a test point and all known data points.

Sort distance measurements to find the points closest to the test point:

	dist
7	75.91
5	80.17
6	80.31
4	84.69
3	87.00

The 5 nearest neighbours

Use majority class labels of those closest points to predict the label of the test point:

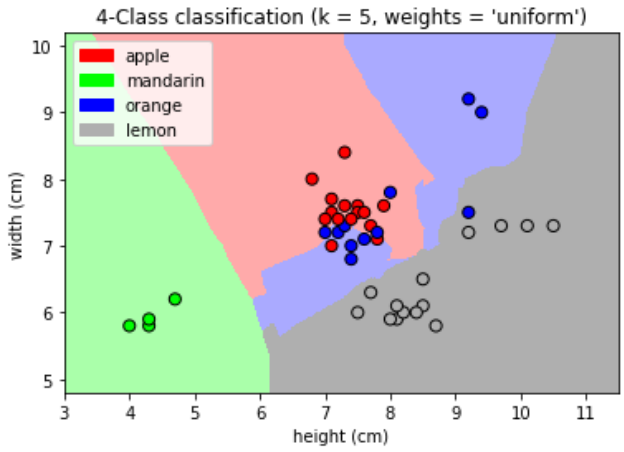
For this step, I use **collections.Counter** to keep track of the labels that coincide with the nearest neighbor points. I then use the **.most_common()** method to return the most commonly occurring label. If there is a tie between two or more labels for the title of "most common" label, the one that was first encountered by the **Counter()** object will be the one that gets returned. We got a value 2.

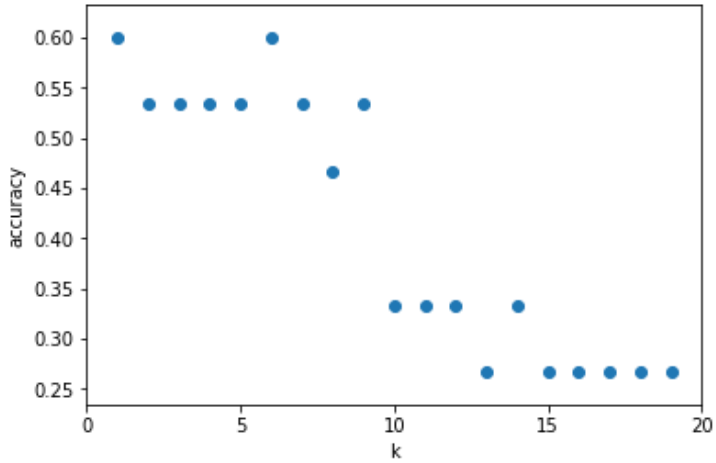
KNN MODEL:

Using the mean cross-validation, we can conclude that we expect the model to be around 95 % accurate on average. If we look at all the 10 scores produced by the 10-fold cross-validation, we can also conclude that there is a relatively high variance in the accuracy between folds, ranging from 100% accuracy to 95% accuracy. So, we can conclude that the model is very dependent on the particular folds used for training, but it also be the consequence of the small size of the dataset. We can see that 10-fold cross-validation accuracy does not result in performance improvement for this model.

Accuracy of K-NN classifier on training set: 0.95
Accuracy of K-NN classifier on test set: 1.00

RESULTS:





For this particular dateset, we obtain the highest accuracy when k=5

From the above graph we can infer that after assigning some values atlast 5 to the k gives the higher accuracy.

Focused on the prediction accuracy. Our objective is to learn a model that has a good generalization performance. Such a model maximizes the prediction accuracy. We identified the machine learning algorithm that is best-suited for the

problem at hand (i.e. fruit types classification); therefore, we compared different algorithms and selected the best-performing one.

CONCLUSION:

1. In this task, I build a kNN classifier model to classify the fruits according to the given mass, color,width,height. The model yields very good performance as indicated by the model accuracy which was found to be 0.95 with k=5.
2. With k=3, the training-set accuracy score is 0.93 while the test-set accuracy to be 0.1. These two values are quite comparable. So, there is no question of overfitting.
3. I have compared the model accuracy score which is 0.95 with null accuracy score which is 0.6071. So, we can conclude that our K Nearest Neighbors model is doing a very good job in predicting the class labels.
4. Our original model accuracy score with k=3 is 0.93. Now, we can see that we get same accuracy score of 0.95 with k=5. But, if we increase the value of k further, this would result in enhanced accuracy. With k=6,7,8 we get accuracy score of 0.93. So, it results in performance improvement. If we increase k to 9, then accuracy decreases again to 0.93. So, we can conclude that our optimal value of k is 5.
5. kNN Classification model with k=5 shows more accurate predictions and less number of errors than k=3 model. Hence, we got performance improvement with k=5.
6. ROC AUC of our model approaches towards 1. So, we can conclude that our classifier does a good job in predicting which fruit.
7. Using the mean cross-validation, we can conclude that we expect the model to be around 95 % accurate on average.
8. If we look at all the 10 scores produced by the 10-fold cross-validation, we can also conclude that there is a relatively high variance in the accuracy between folds, ranging from 100% accuracy to 95% accuracy. So, we can

conclude that the model is very dependent on the particular folds used for training, but it also be the consequence of the small size of the dataset.