**Curneu MedTech Innovation Private Limited**

**Name : GOWTHAM SG**
**College register no : 1632009**
**Task number : SD03Q03**
**Repository link : https://github.com/Gowtham2709/Curnue_tasks.git**

**Task - 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.

1. Generate scatter plots for various combination of parameters and do the feature
2. engineering meaning thereby which parameters of best suited to build the classifier.
3. Split the data into test and training split.
4. Building a classifier using KNN from scratch.
5. Figure out the best value of k with highest r_score.
6. Run at least three test cases on the parameter and assess the fruit using the classifier.

**Remark** :Only use python.

**AIM**

To develop K - nearest neighbor classifier model for fruits dataset containing attributes such as height, mass, width and color score to predict the fruit name for the data given by the user.

**DATASET**

**Name** : Fruit dataset
**Record size** : 60 records
**Attributes** : Fruit label, fruit name, mass, width, length, color score
**Sample dataset**

| | fruit_label | fruit_name | mass | width | height | color_score |
|---|---|---|---|---|---|---|
| 2 | 1 | apple | 192 | 8.4 | 7.3 | 0.55 |
| 3 | 1 | apple | 180 | 8 | 6.8 | 0.59 |
| 4 | 1 | apple | 176 | 7.4 | 7.2 | 0.6 |
| 5 | 2 | mandarin | 86 | 6.2 | 4.7 | 0.8 |
| 6 | 2 | mandarin | 84 | 6 | 4.6 | 0.79 |
| 7 | 2 | mandarin | 80 | 5.8 | 4.3 | 0.77 |
| 8 | 2 | mandarin | 80 | 5.9 | 4.3 | 0.81 |
| 9 | 2 | mandarin | 76 | 5.8 | 4 | 0.81 |
| 10 | 1 | apple | 178 | 7.1 | 7.8 | 0.92 |
| 11 | 1 | apple | 172 | 7.4 | 7 | 0.89 |
| 12 | 1 | apple | 166 | 6.9 | 7.3 | 0.93 |
| 13 | 1 | apple | 172 | 7.1 | 7.6 | 0.92 |

## METHODOLOGY

**Tool used :** Google colaboratory (Python 3.6)

### K - Nearest neighbor

The k-nearest neighbors (KNN) algorithm is a simple, supervised machine learning algorithm that can be used to solve both classification and regression problems. It's easy to implement and understand, but has a major drawback of becoming significantly slows as the size of that data in use grows.

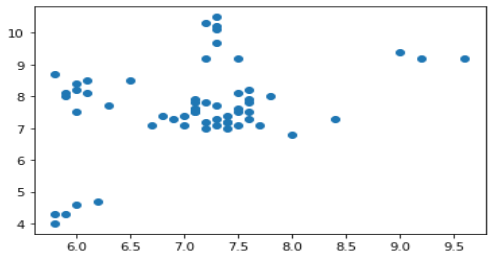
We can implement a KNN model by following the below steps:

1. Load the data
2. Initialise the value of k
3. For getting the predicted class, iterate from 1 to total number of training data points
    1. Calculate the distance between test data and each row of training data. Here we will use Euclidean distance as our distance metric since it's the most popular method. The other metrics that can be used are Chebyshev, cosine, etc.
    2. Sort the calculated distances in ascending order based on distance values
    3. Get top k rows from the sorted array
    4. Get the most frequent class of these rows
    5. Return the predicted class.

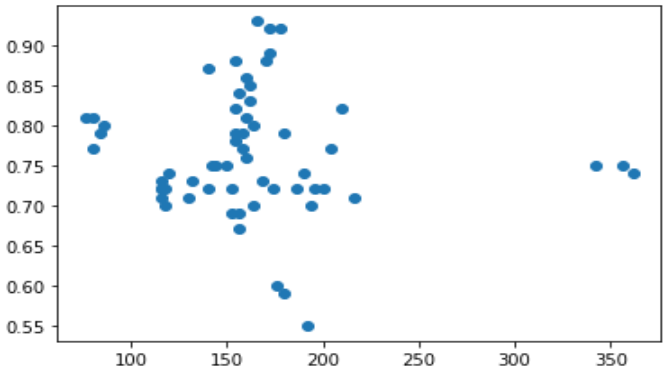
### Tasks

**Generate scatter plots for various combination of parameters and do the feature engineering meaning thereby which parameters of best suited to build the classifier.**

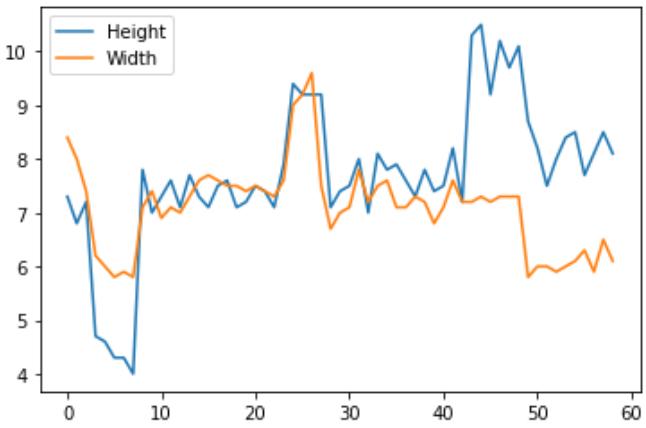
Scatter plots the attributes combinations such as width ans height, mass and color score and a line graph for width and length is also constructed and the distribution of data values are identified and it the conclusion made is that width, height, mass and color score is used as the features for building KNN model.Following are the visualizations made for the feature engineering.




**Scatter plot for width and height**




**Scatter plot for mass and color score**




**Line graph for height and width**

**Split the data into test and training split.**

For model training and testing the given dataset is spiltted into test and train by using python package SKlearn.model_selection. Following is the code snippet for used for splitting the data.

***Code snippet***

```
from sklearn.model_selection import train_test_split
X=fruits[['mass','width','height']]
Y=fruits['fruit_label']
X_train,X_test,y_train,y_test=train_test_split(X,Y,random_state=0)
```

**Building a classifier using KNN**

KNN classifier is build by using sklearn's kneighborclassifier package with n_neighborequal to 5 and with uniform weights.Following is the output for fitting the KNN model.

***Output***

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=5, p=2,
                     weights='uniform')
```

**Figure out the best value of k with highest r_score.**

KNN score is found for the developed model.

```
knn.score(X_test,y_test)
0.5333333333333333
```

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

**For prediction mass, weight and lengthare the parameters given.**

**Case 1**

```
prediction1=knn.predict([['100','6.3','8']])
predct[prediction1[0]]
```

**Output :** lemon

**Case 2**

```
prediction2=knn.predict([['300','7','10']])
predct[prediction2[0]]
```

**Output :** orange
**Case 3**

```
prediction3=knn.predict([['150','8','15']])
predct[prediction3[0]]
```

**Output :** apple