**Problem Statement**: Create a model that classifies whether the given fruit is Apple or Orange

Steps in Problem Solving:

1. Create a Dataset
2. Create a Model
3. Train the Model
4. Predict the value

1 **Create a DataSet**: Dataset is nothing but the collaboration of feature and labels. Feature is the input variable that we give to the model and Label is the predicted or the output variable. In our case Texture and weight of the fruits is Feature and Label is the fruit (apple and orange)

|  |  |  |
| --- | --- | --- |
| Texture | Weight | label |
| Smooth | 130kg | Apple |
| Smooth | 140kg | Apple |
| Bumpy | 150kg | Orange |
| Bumpy | 170kg | Orange |

**Code:**

Feature = [[130,’Smooth’],[140,’Smooth’],[150,’Bumpy’],[170,’Bumpy]]

Labels = [‘Apple’,’Apple’,’Orange’,’Orange’]

**Output:**

Feature = [[130,1],[140,1],[150,0],[170,0]]

Labels=[1,1,0,0]

Now for calculation purpose we are going to change the value into int instead of a string value

Feature = [[130,1],[140,1],[150,0],[170,0]]

Labels=[1,1,0,0]

2 **Create a model**: We are going to use Decision Tree, in short decision tree is a non-parametric supervised learning algorithum, it makes prediction based on the input that we provide.

**Code:**

from sklearn import tree

clf=tree.DecisionTreeClassifier()

model is created and stored in a variable called clf

3 **Train the model**

We train the model by fitting the value into it, at this stage the model learn about the data

**Code:**

clf=clf.fit(Feature,Labels)

4 **Predit the value**: value where given into the trained model to predict the output

print(clf.predict([[160,0]]))

**output**

print(clf.predict([[160,0]]))

**[0]**

In out case 1 indicate Apple and 0 indicate orange so out model has predict the value to be orange

So here we go…….