DATA MINING END SEMESTER PRACTICAL EXAM

SET- 5

NAME: AKASH SAMPATH

ROLL: 18BCS009

Date: 23/06/2021

Download a suitable dataset for classification from any Repository. List the attributes and its type in a word Doc.

Note: I have used SVM classification of my repository and have listed the attributes and its type

In [1]:
```
from sklearn.model_selection import train_test_split
```

In [2]:
```
import pandas as pd

data = pd.read_csv("apples_and_oranges.csv")
```

In [3]:
```
data
```

Out[3]:

	Weight	Size	Class
0	69	4.39	orange
1	69	4.21	orange
2	65	4.09	orange
3	72	5.85	apple
4	67	4.70	orange
5	73	5.68	apple
6	70	5.56	apple
7	75	5.11	apple
8	74	5.36	apple
9	65	4.27	orange
10	73	5.79	apple
11	70	5.47	apple
12	74	5.53	apple
13	68	4.47	orange
14	74	5.22	apple
15	65	4.48	orange
16	69	4.66	orange
17	75	5.25	apple
18	67	4.18	orange
19	74	5.50	apple
20	66	4.13	orange
21	70	4.83	orange
22	69	4.61	orange
23	68	4.08	orange

	Weight	Size	Class
24	67	4.25	orange
25	71	5.35	apple
26	67	4.01	orange
27	70	4.22	orange
28	74	5.25	apple
29	71	5.26	apple
30	73	5.78	apple
31	66	4.68	orange
32	72	5.72	apple
33	73	5.17	apple
34	68	4.83	orange
35	69	4.11	orange
36	69	4.76	orange
37	74	5.48	apple
38	70	5.59	apple
39	73	5.03	apple

Load the dataset and set the target and feature variables. Split the dataset into training and test dataset. Build decision tree classifier with Entropy criteria. Perform Prediction for test dataset using Entropy and print the results in the form of confusion matrix, accuracy, and classification report. visualize the decision tree.

Dataset used is balance scale data

In [29]:
```
import numpy as np
import pandas as pd
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.tree import export_graphviz
```

In [5]:
```
# Function importing Dataset
def importdata():
    balance_data = pd.read_csv(
'https://archive.ics.uci.edu/ml/machine-learning-'+
'databases/balance-scale/balance-scale.data',
    sep= ',', header = None)

    # Printing the dataswet shape
    print ("Dataset Length: ", len(balance_data))
    print ("Dataset Shape: ", balance_data.shape)

    # Printing the dataset obseravtions
    print ("Dataset: ",balance_data.head())

    return balance_data
```

In [6]:
```
# Function to split the dataset
def splitdataset(balance_data):

    # Separating the target variable
    X = balance_data.values[:, 1:5]
    Y = balance_data.values[:, 0]

    # Splitting the dataset into train and test
    X_train, X_test, y_train, y_test = train_test_split(
    X, Y, test_size = 0.3, random_state = 100)

    return X, Y, X_train, X_test, y_train, y_test
```

In [8]:
```
# Function to perform training with entropy.
def tarin_using_entropy(X_train, X_test, y_train):

    # Decision tree with entropy
    clf_entropy = DecisionTreeClassifier(
            criterion = "entropy", random_state = 100,
            max_depth = 3, min_samples_leaf = 5)

    # Performing training
    clf_entropy.fit(X_train, y_train)
    return clf_entropy
```

In [9]:
```
# Function to make predictions
def prediction(X_test, clf_object):

    # Predicton on test with giniIndex
    y_pred = clf_object.predict(X_test)
    print("Predicted values:")
    print(y_pred)
    return y_pred
```


In [10]:
```
# Function to calculate accuracy
def cal_accuracy(y_test, y_pred):

    print("Confusion Matrix: ",
        confusion_matrix(y_test, y_pred))

    print ("Accuracy : ",
    accuracy_score(y_test,y_pred)*100)

    print("Report : ",
    classification_report(y_test, y_pred))
```

In [25]:
```
# Driver code
def main():

    # Building Phase
    data = importdata()
    X, Y, X_train, X_test, y_train, y_test = splitdataset(data)
    clf_gini = train_using_gini(X_train, X_test, y_train)
    clf_entropy = tarin_using_entropy(X_train, X_test, y_train)

    # Operational Phase
    print("Results Using Gini Index:")
```

```
print("Results Using Entropy:")
# Prediction using entropy
y_pred_entropy = prediction(X_test, clf_entropy)
cal_accuracy(y_test, y_pred_entropy)
```

,

In [30]:
```
# Calling main function
if __name__=="__main__":
    main()
```

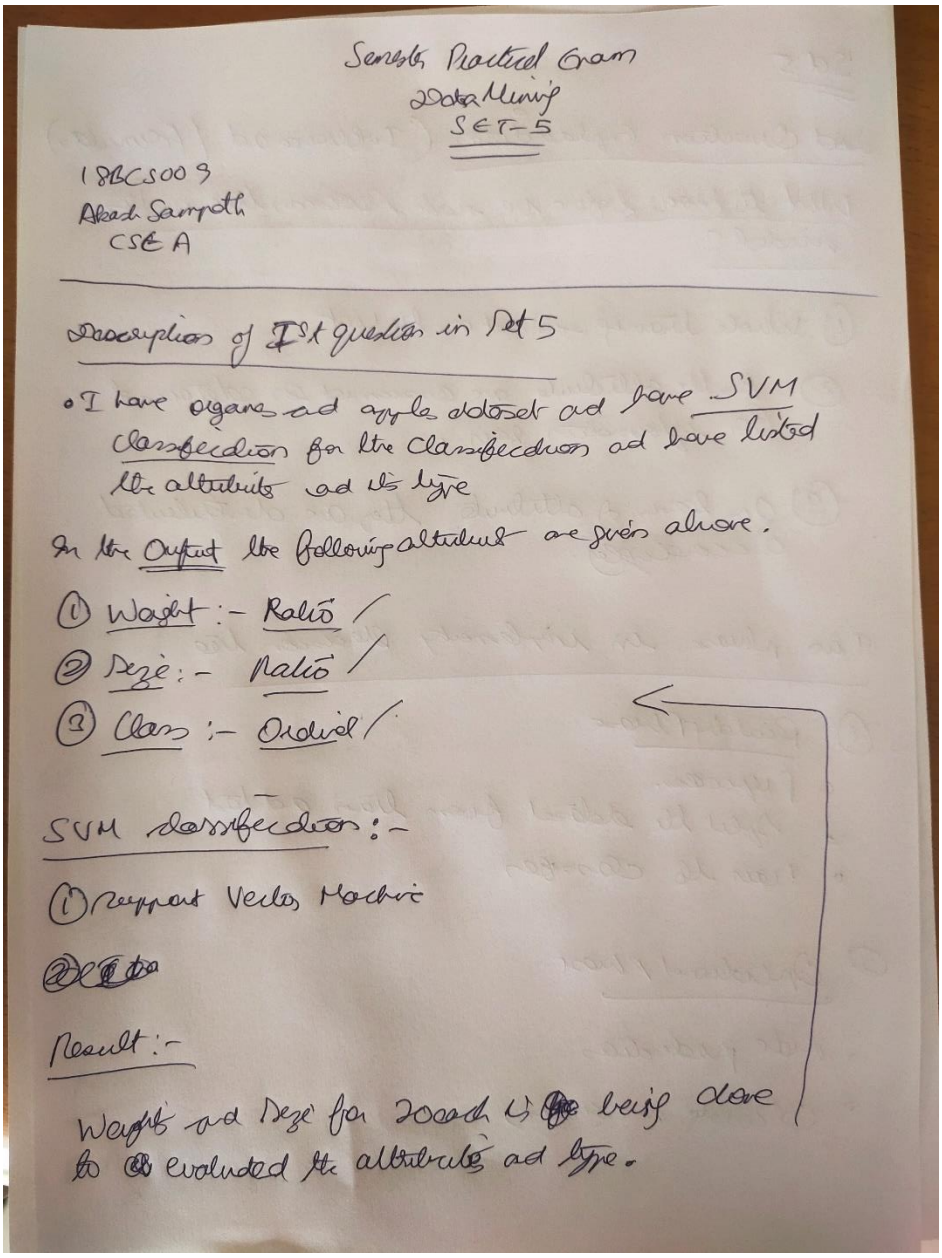
```
Dataset Length:  625
Dataset Shape:  (625, 5)
Dataset:     0  1  2  3  4
0  B  1  1  1  1
1  R  1  1  1  2
2  R  1  1  1  3
3  R  1  1  1  4
4  R  1  1  1  5
Results Using Gini Index:
Predicted values:
['R' 'L' 'R' 'R' 'R' 'L' 'R' 'L' 'L' 'L' 'R' 'L' 'L' 'L' 'R' 'L' 'R' 'L'
 'L' 'R' 'L' 'R' 'L' 'L' 'R' 'L' 'L' 'L' 'R' 'L' 'L' 'L' 'R' 'L' 'L' 'L'
 'L' 'R' 'L' 'L' 'R' 'L' 'R' 'L' 'R' 'R' 'L' 'L' 'R' 'L' 'R' 'R' 'L' 'R'
 'R' 'L' 'R' 'R' 'L' 'L' 'R' 'R' 'L' 'L' 'L' 'L' 'L' 'R' 'R' 'L' 'L' 'R'
 'R' 'L' 'R' 'L' 'R' 'R' 'R' 'L' 'R' 'L' 'L' 'L' 'L' 'R' 'R' 'L' 'R' 'L'
 'R' 'R' 'L' 'L' 'L' 'R' 'R' 'L' 'L' 'L' 'R' 'L' 'R' 'R' 'R' 'R' 'R' 'R'
 'R' 'L' 'R' 'L' 'R' 'R' 'L' 'R' 'R' 'R' 'R' 'R' 'L' 'R' 'L' 'L' 'L' 'L'
 'L' 'L' 'L' 'R' 'R' 'R' 'R' 'L' 'R' 'R' 'R' 'L' 'L' 'R' 'L' 'R' 'L' 'R'
 'L' 'L' 'R' 'L' 'L' 'R' 'L' 'R' 'L' 'R' 'R' 'R' 'L' 'R' 'R' 'R' 'R' 'R'
 'L' 'L' 'R' 'R' 'R' 'R' 'L' 'R' 'R' 'R' 'L' 'R' 'L' 'L' 'L' 'L' 'R' 'R'
```

```
Confusion Matrix:  [[ 0  6  7]
 [ 0 67 18]
 [ 0 19 71]]
Accuracy :  73.40425531914893
Report :                precision    recall  f1-score   support

           B       0.00      0.00      0.00        13
           L       0.73      0.79      0.76        85
           R       0.74      0.79      0.76        90

    accuracy                           0.73       188
   macro avg       0.49      0.53      0.51       188
weighted avg       0.68      0.73      0.71       188
```

```
Results Using Entropy:
Predicted values:
['R' 'L' 'R' 'L' 'R' 'L' 'R' 'L' 'R' 'R' 'R' 'R' 'L' 'L' 'R' 'L' 'R' 'L'
 'L' 'R' 'L' 'R' 'L' 'L' 'R' 'L' 'R' 'L' 'R' 'L' 'R' 'L' 'R' 'L' 'L' 'L'
 'L' 'L' 'R' 'L' 'R' 'L' 'R' 'L' 'R' 'R' 'L' 'L' 'R' 'L' 'L' 'R' 'L' 'L'
 'R' 'L' 'R' 'R' 'L' 'R' 'R' 'R' 'L' 'L' 'R' 'L' 'L' 'R' 'L' 'L' 'L' 'R'
 'R' 'L' 'R' 'L' 'R' 'R' 'R' 'L' 'R' 'L' 'L' 'L' 'L' 'R' 'R' 'L' 'R' 'L'
 'R' 'R' 'L' 'L' 'L' 'R' 'R' 'L' 'L' 'L' 'R' 'L' 'L' 'R' 'R' 'R' 'R' 'R'
 'R' 'L' 'R' 'L' 'R' 'R' 'L' 'R' 'R' 'L' 'R' 'R' 'L' 'R' 'R' 'R' 'L' 'L'
 'L' 'L' 'L' 'R' 'R' 'R' 'R' 'L' 'R' 'R' 'R' 'L' 'L' 'R' 'L' 'R' 'L' 'R'
 'L' 'R' 'R' 'L' 'L' 'R' 'L' 'R' 'R' 'R' 'R' 'R' 'L' 'R' 'R' 'R' 'R' 'R'
 'R' 'L' 'R' 'L' 'R' 'R' 'L' 'R' 'L' 'R' 'L' 'R' 'L' 'L' 'L' 'L' 'L' 'R'
 'R' 'R' 'L' 'L' 'L' 'R' 'R' 'R']
Confusion Matrix:  [[ 0  6  7]
 [ 0 63 22]
 [ 0 20 70]]
Accuracy :  70.74468085106383
```

Report :

	precision	recall	f1-score	support
B	0.00	0.00	0.00	13
L	0.71	0.74	0.72	85
R	0.71	0.78	0.74	90
accuracy			0.71	188
macro avg	0.47	0.51	0.49	188
weighted avg	0.66	0.71	0.68	188



Semester Practical Exam
Data Mining
SET-5

18BCS009
Akash Sampath
CSE A

Description of 1st question in Set 5

- I have given and apple dataset and have SVM classification for the classification and have listed the attributes and its type

In the Output the following attributes are given above.

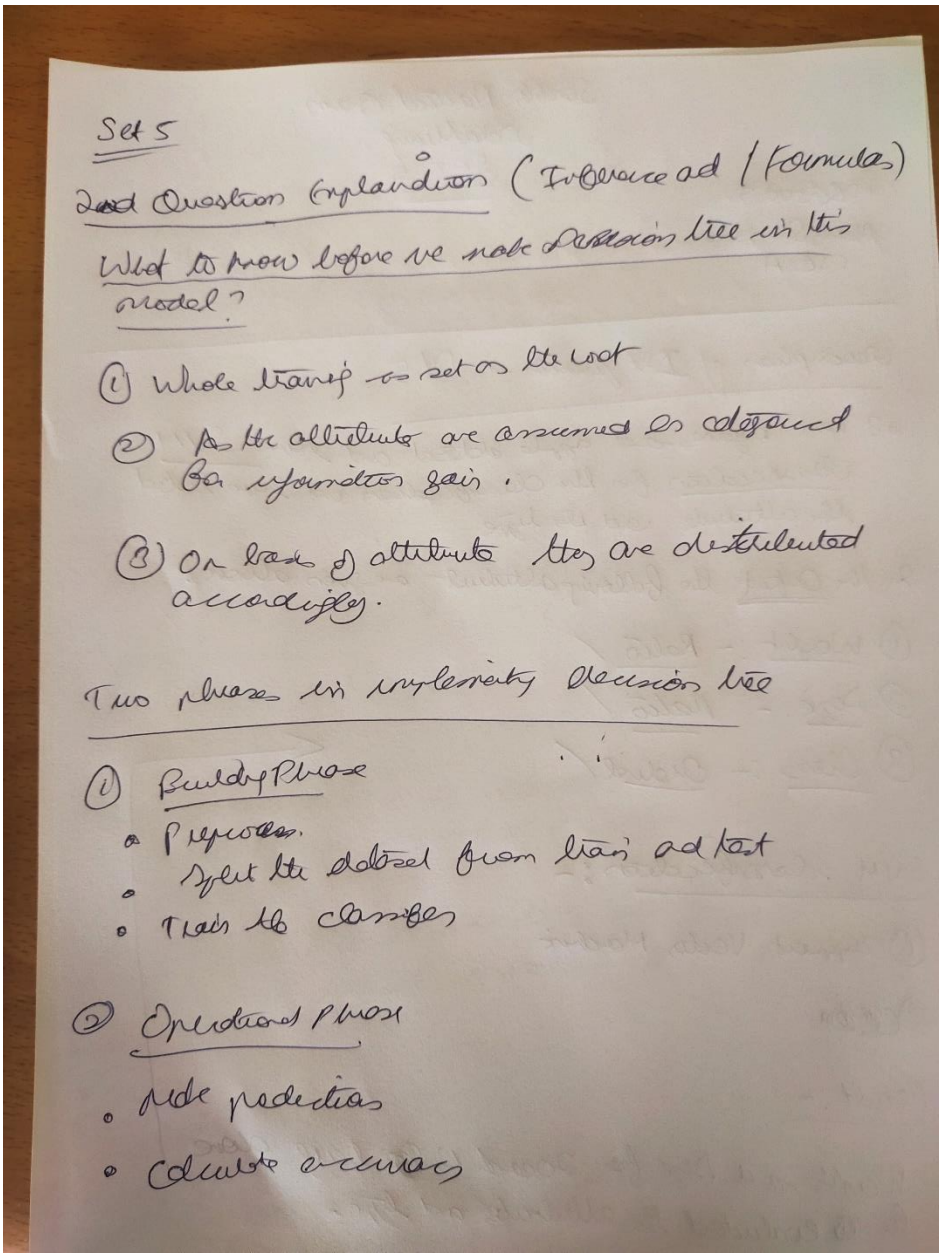
1. Weight :- Ratio
2. Size :- Ratio
3. Class :- Ordinal

SVM classification :-

1. Support Vector Machine
2. [illegible]

Result :-

Weight and Size for 2nd and 3rd being done to be evaluated the attributes and type.



Set 5

2nd Question Explanation (Theoretical / Formulas)

What to know before we make Decision tree in this model?

(1) Whole training set at the root

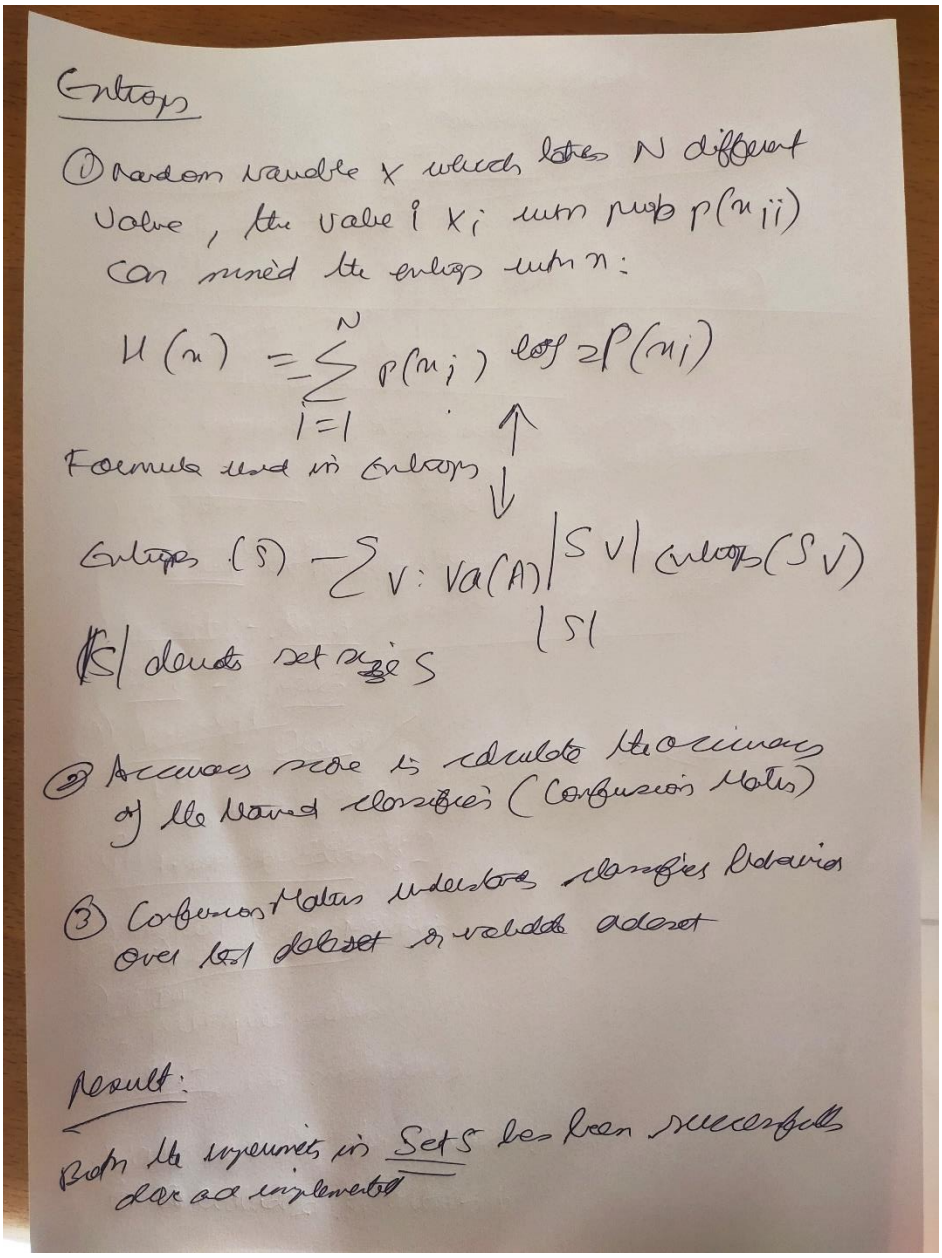
(2) As the attributes are assumed to categorical for information gain.

(3) On basis of attribute they are distributed accordingly.

Two phases in implementing decision tree

(1) Building Phase
- Preprocess.
- Split the dataset from train and test
- Train the classifier

(2) Operational Phase
- Make predictions
- Calculate accuracy


Entropy
① Random variable X which takes N different value, the value i x_i with prob $p(x_i)$ can mead the entropy with x:
$H(x) = -\sum_{i=1}^{N} p(x_i) \log_2 p(x_i)$
Formula used in Entropy
$Entropy(S) = -\sum_{v: Val(A)} \frac{|S_v|}{|S|} Entropy(S_v)$
$|S|$ denote set size S
② Accuracy score is calculate the accuracy of the trained classifier (Confusion Matrix)
③ Confusion Matrix understand classifier behavior over test dataset or validate dataset
Result:
Both the experiments in Set 5 has been successfully done as implemented