PROJECT REPORT

On DRUG prediction using

Decision Tree algorithm

Department of Computer Science & Engineering

Dr B R Ambedkar National Institute of Technology,

Jalandhar

Submitted To

Dr Kuldeep Kumar Sir

Submitted by

Suneeta Singh

21203028

Table of Content

1.INTRODUCTION

1.1 MACHINE LEARNING

Machine learning is a part of Artificial intelligence which used for software applications to become more accurate by predicting outomes without using explicite programming.Machine Learning algorithms take historical data as input to predict output values.

Many of leading companies as facebook,google,uber makes machine learning as a central part of their operaions.Machine learning is often classified into different parts:

1.1.1 Types

1.Supervised Learning: Algorithm used labeled types of data for training.

2.Unsupervised learning: In this type of algorithm ,involves training on unlabeled data.

3.Semi-Supervised Learning: This algorithms used mix type of data.Some Used labeled data and some used unlabeled data for training.

Now here we are discussed about Decision tree algorithms.

1.2 DECISION TREE

Decision trees are a type of Supervised machine learning that can be used for both classification and regression problems,but mostly used for solving classification problems.Decision tree is powerful analytical model that has the ability to comprehend data with minimal preprocessing time.In decision tree data is splitting continuously according to given parameter.

A Decision tree consists of nodes in which internal nodes represent feature of dataset,branches represent decision rules and the leaf nodes represent the outcome.

Decision trees classify based on sorting them from root to leaf node.Each node in trees specify a test of some attributes and each branch from that nodes corresponds to some values for that attributes.

1.3 APPLICATION OF DECISION TREE

1.Decision tree modeling can be used for making predictions.

2.As many organization had created their database to enhance their services.Deision tree is possible way to extract information from database.

3.It is an approach to manage costumer relationship that is investigate how individual users access online services.

2. ABOUT PROJECT

In this project we are implementing Drug medication cases using Decision tree model. We have collected data about set of patient ,where everyone suffered from same illness.In their course of treatment duration time,each patient had responded to one of the given 5 medications,Drug A,Drug B,Drug C,Drug X,Drug Y.

Now we have to buid a model to find out which drug would be more appropriate for future patient with same sickness.There are different features of this dataset are Sex,Blood Pressure,Age and the Cholesterol of the patient and the target of this is Drug that each patient responded too.We can use the training part of dataset to build model and use it to predict class of new unknown patient.

Here we assume feature of Dataset is X parameter and Y as a Target parameter.

2.1 METRICS USED

There are few metrics that we can use to build a decision tree model,but here we would use only two most famous metrics:GINI Index and Entropy and Information gain metrics.

2.1.1 GINI INDEX

Gini Index score used to evaluate that how splitting is good by mixed the classes that splits in two groups.It could have a score between 0 and 1 values,where 0 is for all observations belongs to one class and 1 is for random distribution of the observations within the classes.

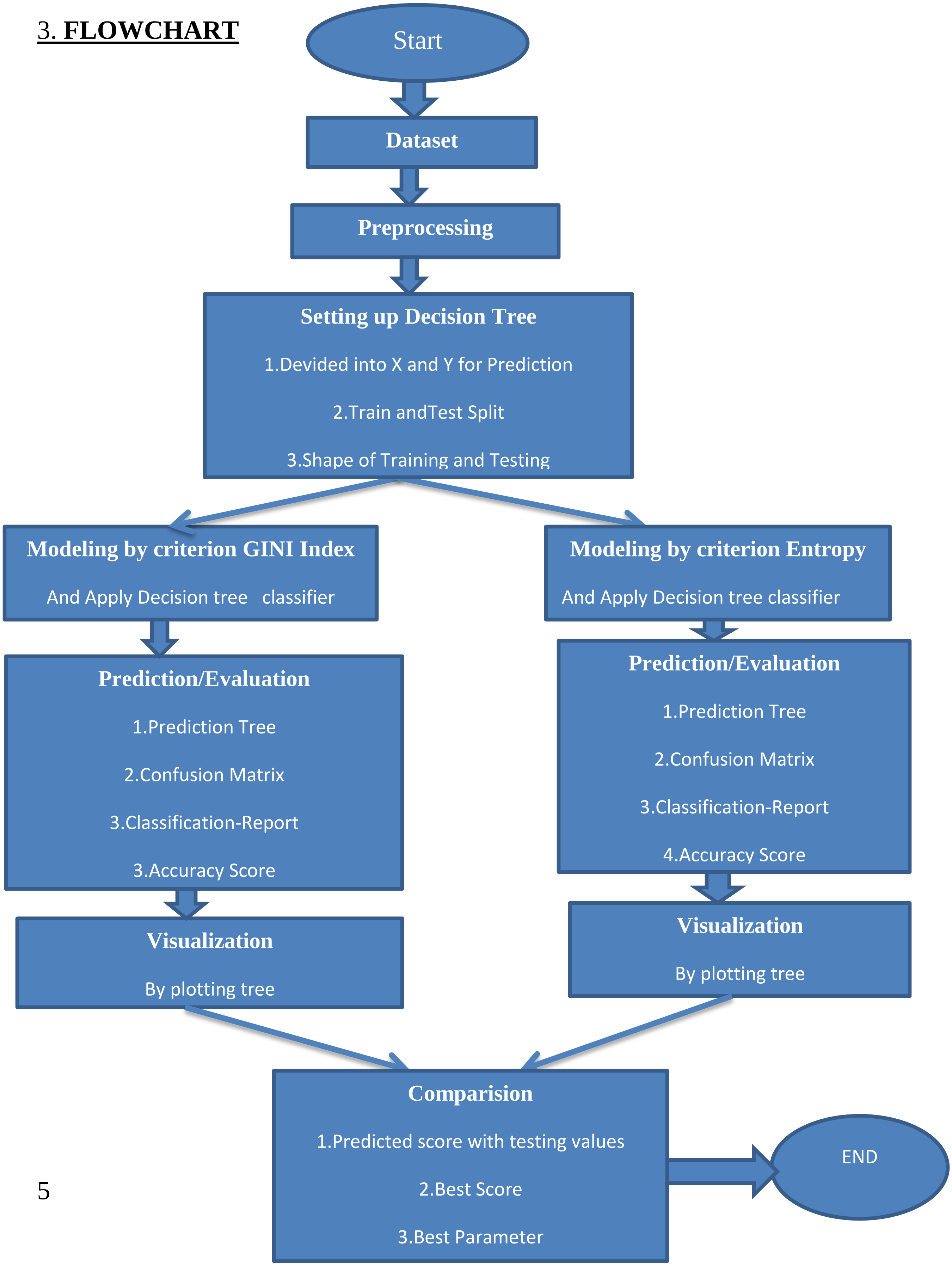
2.1.2 ENTROPY AND INFORMATION GAIN

For entropy we can say that ,measurement of impurity within a dataset.

And Information gain is differenc of entropy before and after the splitting of a feature.

Both the method measured the impurity used to spilitting but in different way. Gini Index calculates the binary split impurity , where as Information Gain measures the entropy before and after the splitting. .

3. FLOWCHART


Start
Dataset
Preprocessing
Setting up Decision Tree
1.Devided into X and Y for Prediction
2.Train andTest Split
3.Shape of Training and Testing
Modeling by criterion GINI Index
And Apply Decision tree   classifier
Modeling by criterion Entropy
And Apply Decision tree classifier
Prediction/Evaluation
1.Prediction Tree
2.Confusion Matrix
3.Classification-Report
3.Accuracy Score
Prediction/Evaluation
1.Prediction Tree
2.Confusion Matrix
3.Classification-Report
4.Accuracy Score
Visualization
By plotting tree
Visualization
By plotting tree
Comparision
1.Predicted score with testing values
2.Best Score
3.Best Parameter
END

4. Steps to Build Decision Tree model

4.1 Downloading and Loading the Dataset

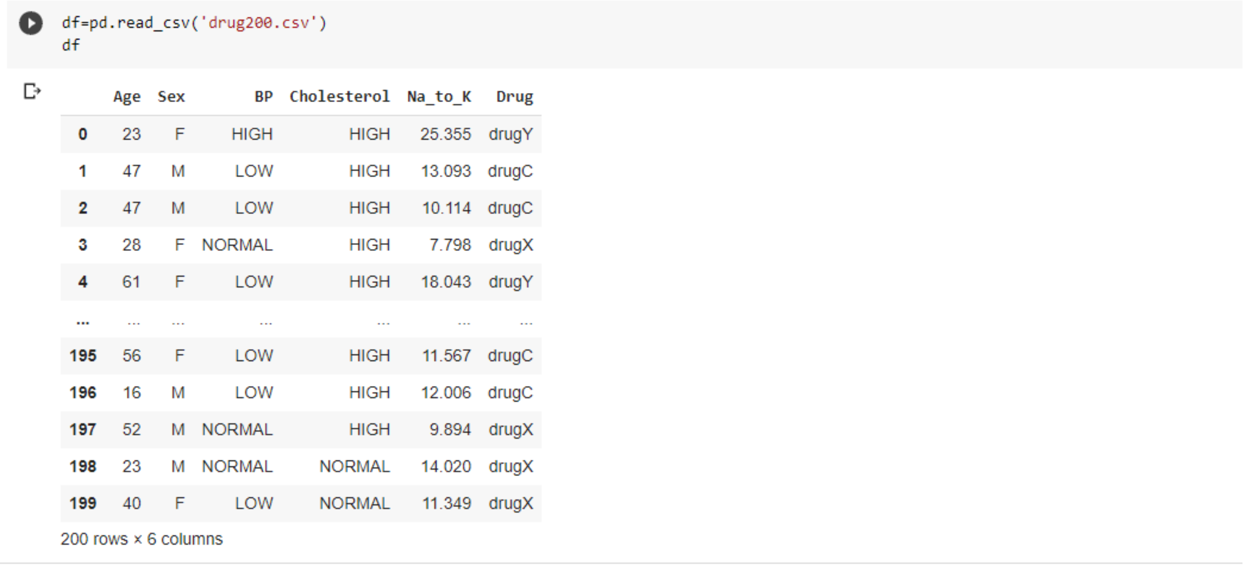
4.1.1 Downloaded the dataset from https://drive.google.com/file/d/1Us7u4Sy13MvRMXTMk2RgfE68OGHdGYUX/view?usp=sharing .

4.1.2 .Create a new file in Google Collab Plateform.

4.1.3 Imported all the libraries numpy,pandas,DecisionTreeClassifier from sklearn.

4.1.4 Loaed the dataset named as “drug200.csv” into Collab file.

4.1.5 Open and read the data file using pandas dataframe.



```
df=pd.read_csv('drug200.csv')
df
```

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	HIGH	HIGH	25.355	drugY
1	47	M	LOW	HIGH	13.093	drugC
2	47	M	LOW	HIGH	10.114	drugC
3	28	F	NORMAL	HIGH	7.798	drugX
4	61	F	LOW	HIGH	18.043	drugY
...	...	...	...	...	...	...
195	56	F	LOW	HIGH	11.567	drugC
196	16	M	LOW	HIGH	12.006	drugC
197	52	M	NORMAL	HIGH	9.894	drugX
198	23	M	NORMAL	NORMAL	14.020	drugX
199	40	F	LOW	NORMAL	11.349	drugX

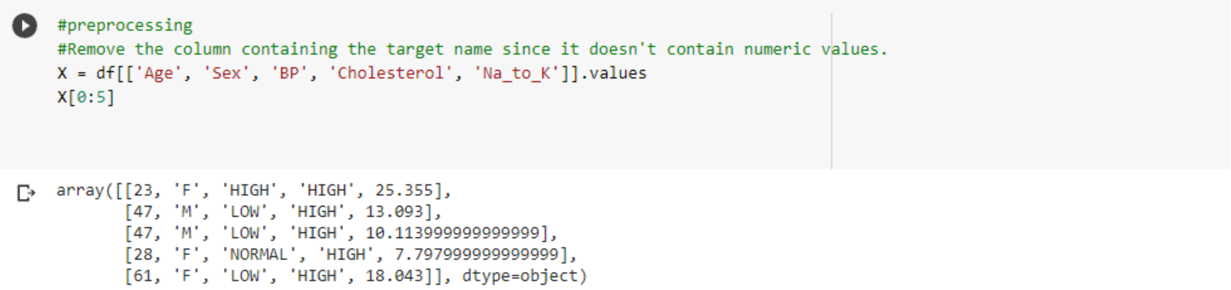
200 rows × 6 columns

4.2 Preprocessing of Data

Using as ‘df’ the drug200.csv data read by pandas, and variables are:

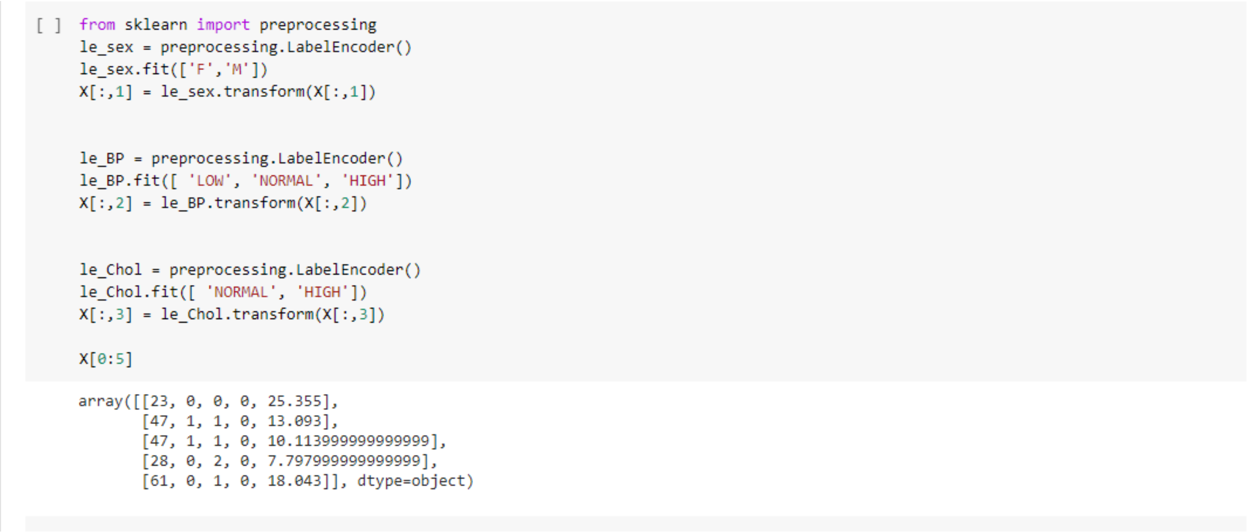
X is the Feature Matrix that is input data (data of df) and Y is the response vector (target or output data)

Remove the column of target name because it does not contain numerical values.



```
#preprocessing
#Remove the column containing the target name since it doesn't contain numeric values.
X = df[['Age', 'Sex', 'BP', 'Cholesterol', 'Na_to_K']].values
X[0:5]
```

```
array([[23, 'F', 'HIGH', 'HIGH', 25.355],
       [47, 'M', 'LOW', 'HIGH', 13.093],
       [47, 'M', 'LOW', 'HIGH', 10.113999999999999],
       [28, 'F', 'NORMAL', 'HIGH', 7.797999999999999],
       [61, 'F', 'LOW', 'HIGH', 18.043]], dtype=object)
```



```
[ ] from sklearn import preprocessing
    le_sex = preprocessing.LabelEncoder()
    le_sex.fit(['F','M'])
    X[:,1] = le_sex.transform(X[:,1])


    le_BP = preprocessing.LabelEncoder()
    le_BP.fit([ 'LOW', 'NORMAL', 'HIGH'])
    X[:,2] = le_BP.transform(X[:,2])


    le_Chol = preprocessing.LabelEncoder()
    le_Chol.fit([ 'NORMAL', 'HIGH'])
    X[:,3] = le_Chol.transform(X[:,3])

    X[0:5]
```

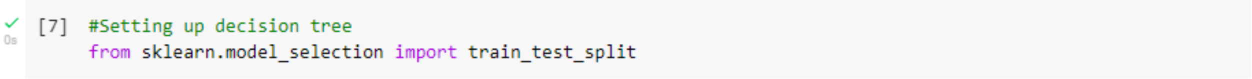
```
array([[23, 0, 0, 0, 25.355],
       [47, 1, 1, 0, 13.093],
       [47, 1, 1, 0, 10.113999999999999],
       [28, 0, 2, 0, 7.797999999999999],
       [61, 0, 1, 0, 18.043]], dtype=object)
```

4.2.1 Fill the target variable.

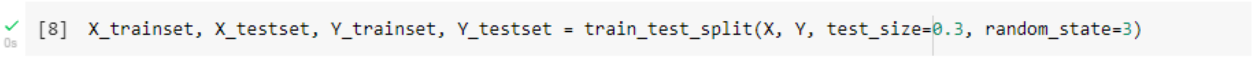
4.3 Setting up the Decision Tree

4.3.1 Devided the data into X and Y where X is set of attribute used for prediction and Y attribute used for whose value needs to be predicted.

Imported train -test split from library sklearn.model_selection and this return 4 parameters X_trainset, X_testset, Y_trainset, Y_testset.



```
[7] #Setting up decision tree
    from sklearn.model_selection import train_test_split
```

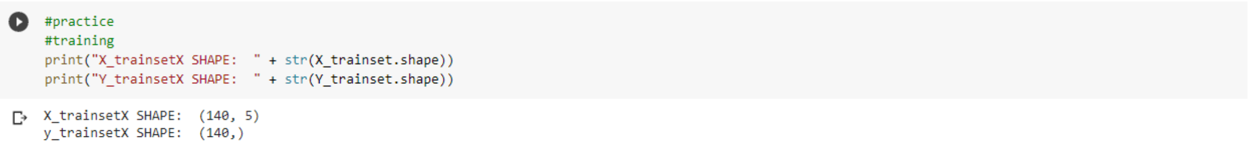


```
[8] X_trainset, X_testset, Y_trainset, Y_testset = train_test_split(X, Y, test_size=0.3, random_state=3)
```

Here test_size represents the ratio of testing dataset and random_state ensures that obtatin the same splits.And 30% dataset to testing and 70% dataset to training.

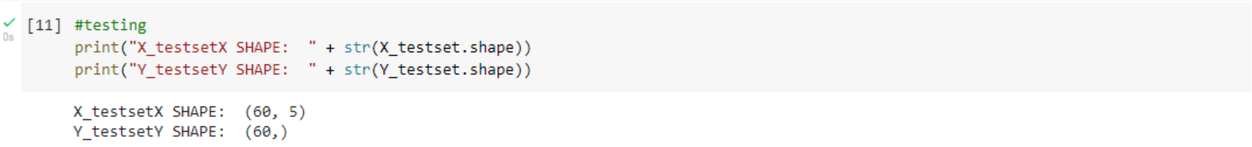
4.4 Practice

4.4.1 Print the shape of X_trainset ,Y_trainset,X_testset and Y_testset and confirm that dimensions match.



```
#practice
#training
print("X_trainsetX SHAPE:  " + str(X_trainset.shape))
print("Y_trainsetX SHAPE:  " + str(Y_trainset.shape))
```

```
X_trainsetX SHAPE:  (140, 5)
y_trainsetX SHAPE:  (140,)
```



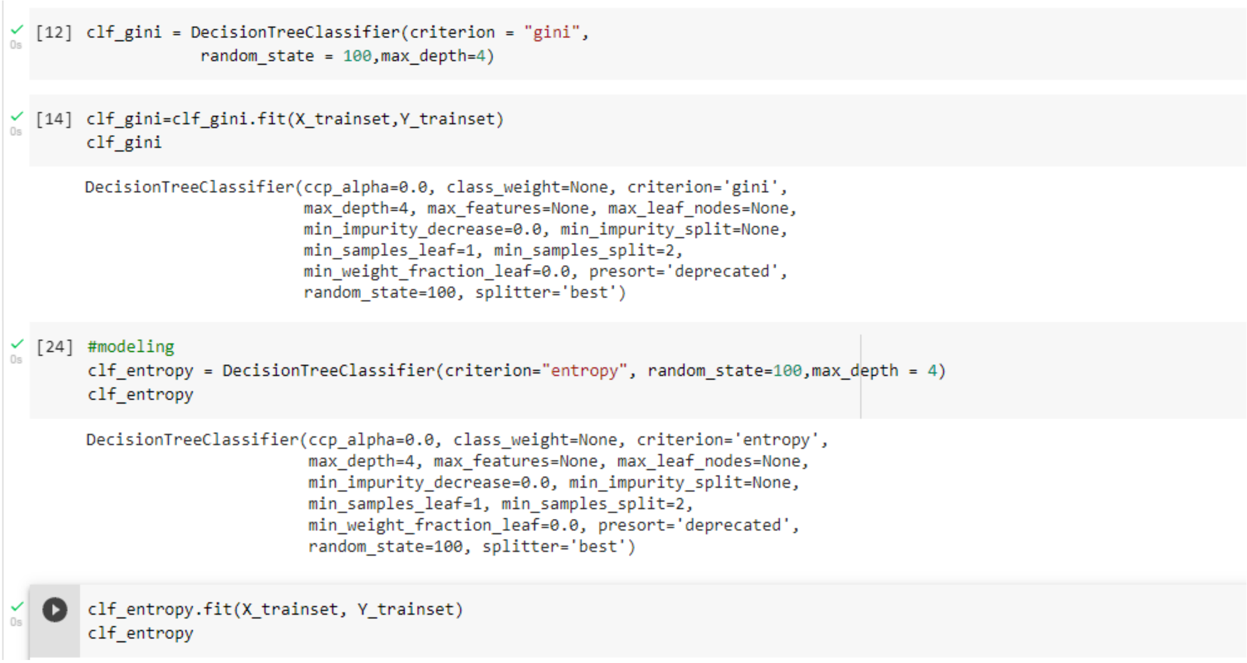
```
[11] #testing
     print("X_testsetX SHAPE:  " + str(X_testset.shape))
     print("Y_testsetY SHAPE:  " + str(Y_testset.shape))
```

```
X_testsetX SHAPE:  (60, 5)
Y_testsetY SHAPE:  (60,)
```

4.5 Modeling

4.5.1 Apply Decision Tree Classification into training data and train the model .

4.5.2 Did this once using   Criterian Gini Index and once with Criterian Entropy.



```
[12] clf_gini = DecisionTreeClassifier(criterion = "gini",
                 random_state = 100,max_depth=4)
```

```
[14] clf_gini=clf_gini.fit(X_trainset,Y_trainset)
     clf_gini
```

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                       max_depth=4, max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=100, splitter='best')
```

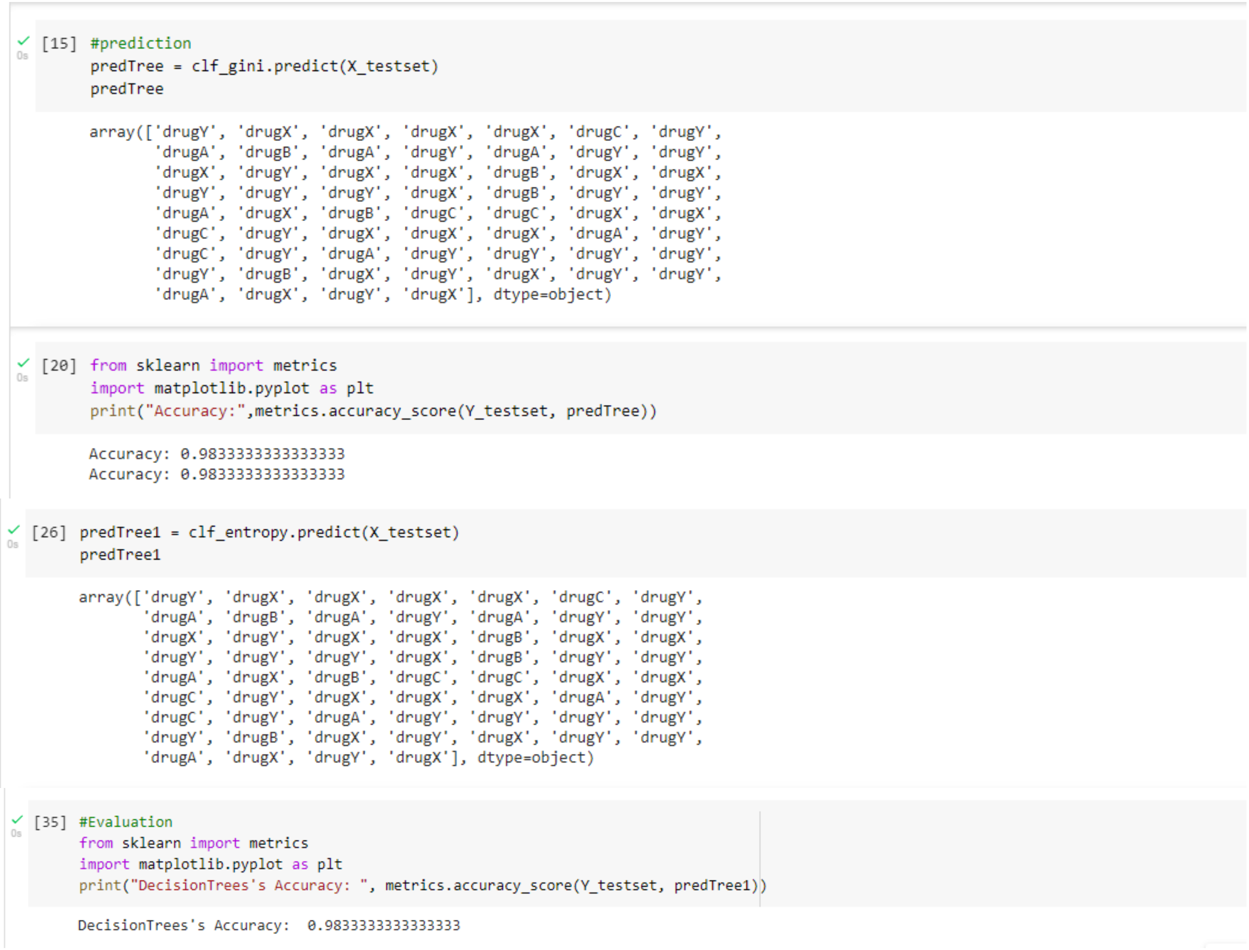
```
[24] #modeling
     clf_entropy = DecisionTreeClassifier(criterion="entropy", random_state=100,max_depth = 4)
     clf_entropy
```

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='entropy',
                       max_depth=4, max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=100, splitter='best')
```

```
clf_entropy.fit(X_trainset, Y_trainset)
clf_entropy
```

4.6 Prediction /Evaluation

4.6.1 Now import the metrics from sklearn.plot confusion matrix and classification report and get the predicted accuracy score of our model by both metrics Gini index and Entropy.



```
[15] #prediction
     predTree = clf_gini.predict(X_testset)
     predTree

     array(['drugY', 'drugX', 'drugX', 'drugX', 'drugX', 'drugC', 'drugY',
            'drugA', 'drugB', 'drugA', 'drugY', 'drugA', 'drugY', 'drugY',
            'drugX', 'drugY', 'drugX', 'drugX', 'drugB', 'drugX', 'drugX',
            'drugY', 'drugY', 'drugY', 'drugX', 'drugB', 'drugY', 'drugY',
            'drugA', 'drugX', 'drugB', 'drugC', 'drugC', 'drugX', 'drugX',
            'drugC', 'drugY', 'drugX', 'drugX', 'drugX', 'drugA', 'drugY',
            'drugC', 'drugY', 'drugA', 'drugY', 'drugY', 'drugY', 'drugY',
            'drugY', 'drugB', 'drugX', 'drugY', 'drugX', 'drugY', 'drugY',
            'drugA', 'drugX', 'drugY', 'drugX'], dtype=object)
```

```
[20] from sklearn import metrics
     import matplotlib.pyplot as plt
     print("Accuracy:",metrics.accuracy_score(Y_testset, predTree))

     Accuracy: 0.9833333333333333
     Accuracy: 0.9833333333333333
```

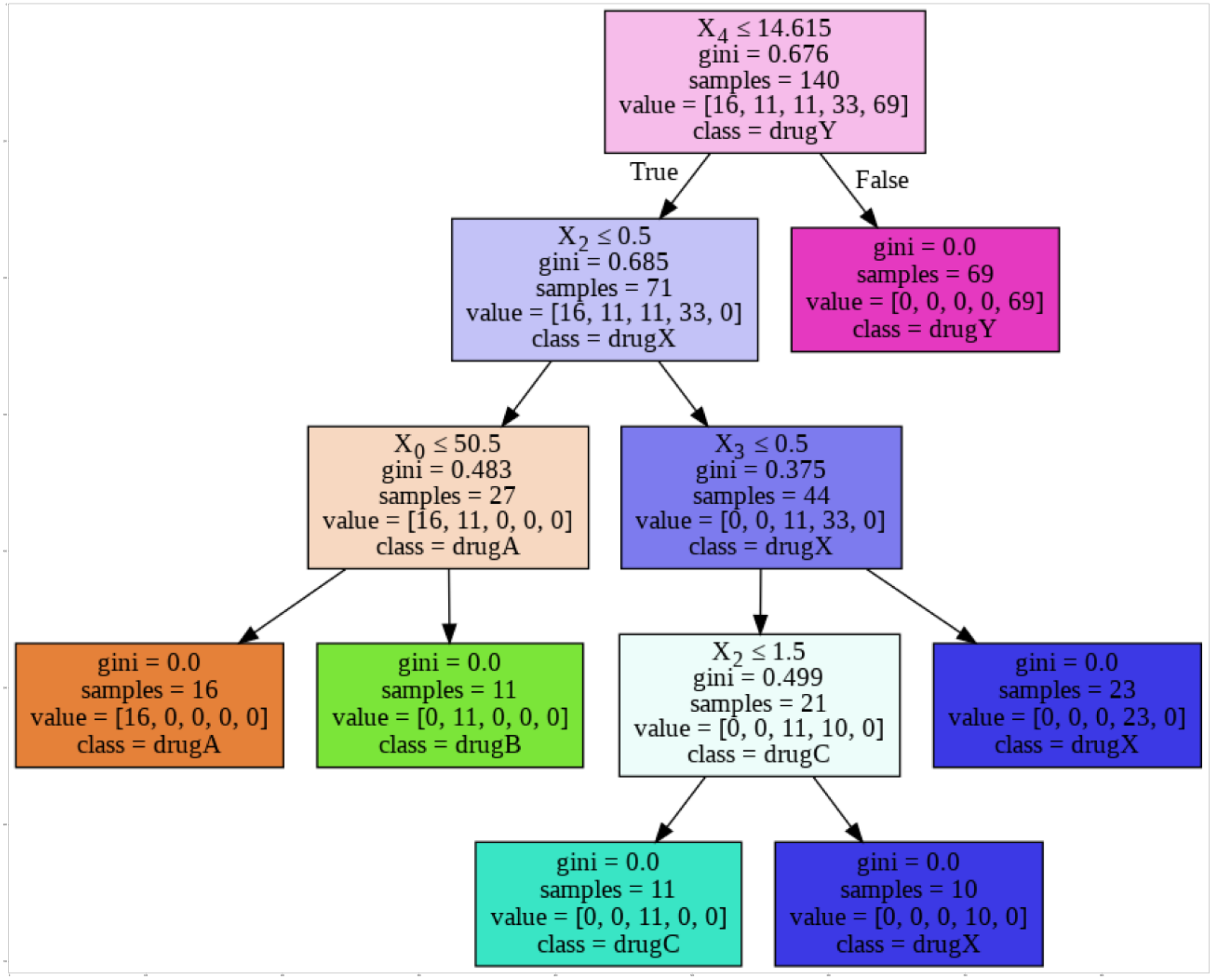
```
[26] predTree1 = clf_entropy.predict(X_testset)
     predTree1

     array(['drugY', 'drugX', 'drugX', 'drugX', 'drugX', 'drugC', 'drugY',
            'drugA', 'drugB', 'drugA', 'drugY', 'drugA', 'drugY', 'drugY',
            'drugX', 'drugY', 'drugX', 'drugX', 'drugB', 'drugX', 'drugX',
            'drugY', 'drugY', 'drugY', 'drugX', 'drugB', 'drugY', 'drugY',
            'drugA', 'drugX', 'drugB', 'drugC', 'drugC', 'drugX', 'drugX',
            'drugC', 'drugY', 'drugX', 'drugX', 'drugX', 'drugA', 'drugY',
            'drugC', 'drugY', 'drugA', 'drugY', 'drugY', 'drugY', 'drugY',
            'drugY', 'drugB', 'drugX', 'drugY', 'drugX', 'drugY', 'drugY',
            'drugA', 'drugX', 'drugY', 'drugX'], dtype=object)
```

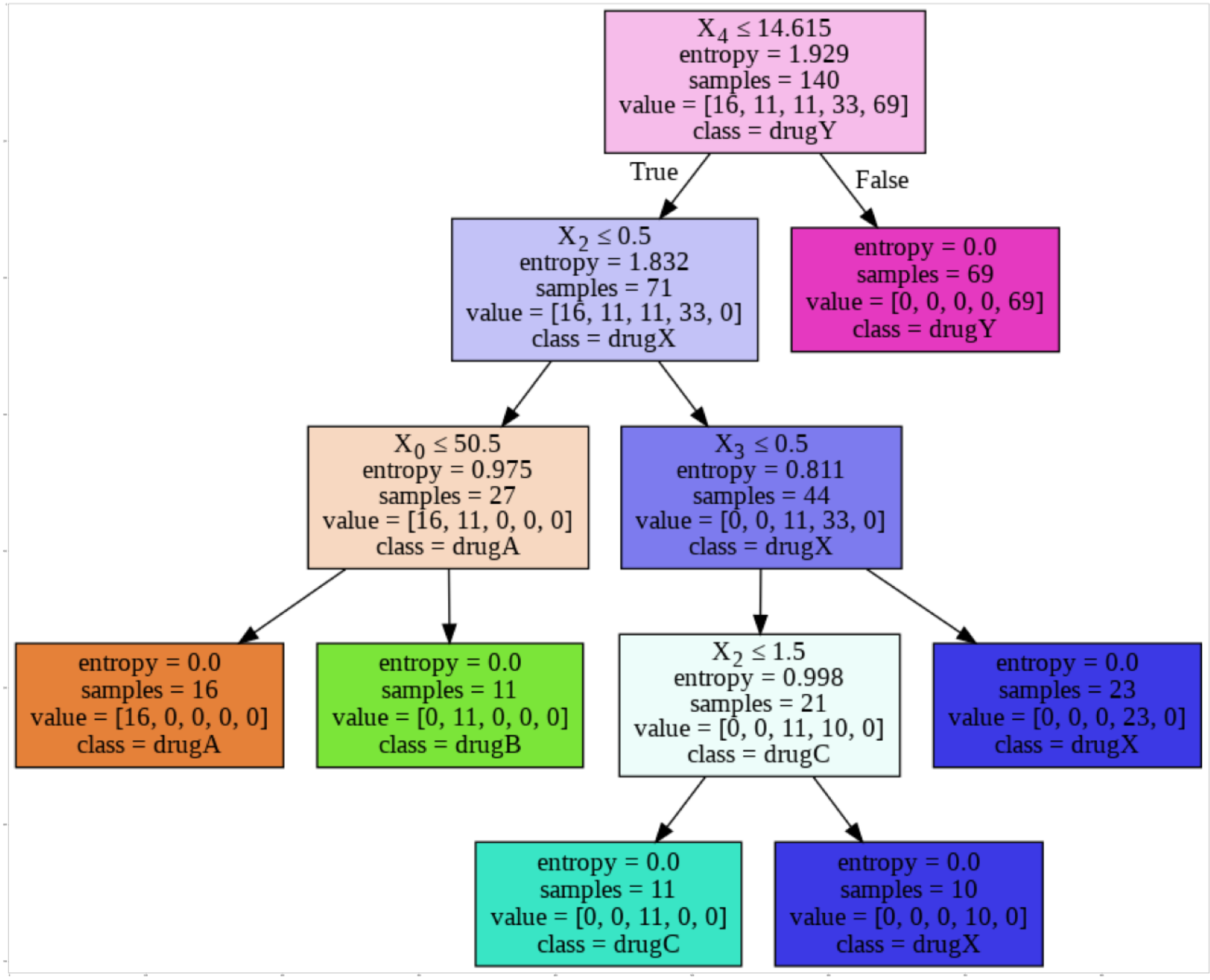
```
[35] #Evaluation
     from sklearn import metrics
     import matplotlib.pyplot as plt
     print("DecisionTrees's Accuracy: ", metrics.accuracy_score(Y_testset, predTree1))

     DecisionTrees's Accuracy:  0.9833333333333333
```

4.7 Visualization of Model

4.7.1 Imported all the libraries StringIO,pydotplus,matplotlib   for plotting the tree.

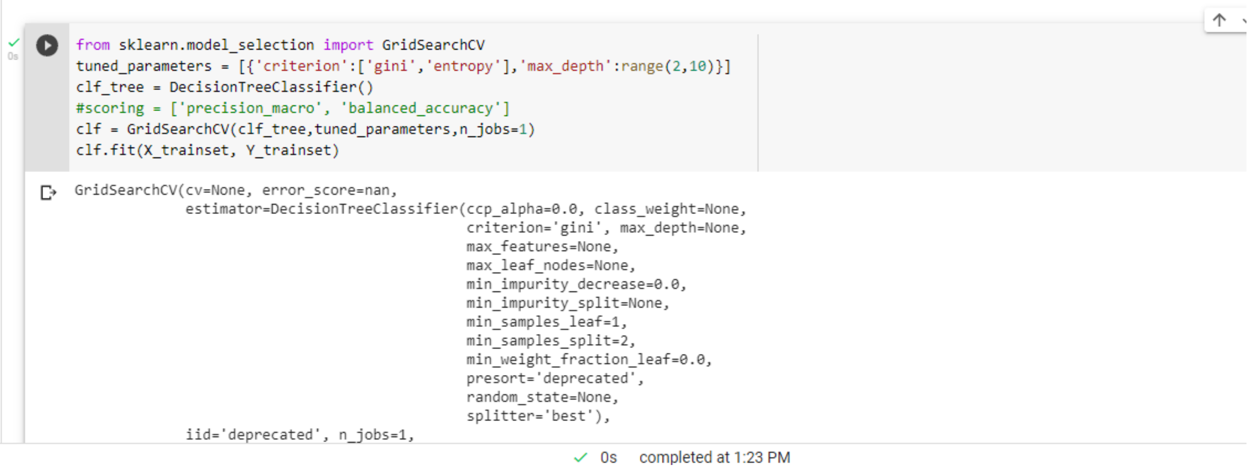
4.7.2 Load tree figure as ‘Drugtree.png ‘ and plot both the trees ,once by Gini criterian and other by entropy.


$X_4 \leq 14.615$
gini = 0.676
samples = 140
value = [16, 11, 11, 33, 69]
class = drugY
True
False
$X_2 \leq 0.5$
gini = 0.685
samples = 71
value = [16, 11, 11, 33, 0]
class = drugX
gini = 0.0
samples = 69
value = [0, 0, 0, 0, 69]
class = drugY
$X_0 \leq 50.5$
gini = 0.483
samples = 27
value = [16, 11, 0, 0, 0]
class = drugA
$X_3 \leq 0.5$
gini = 0.375
samples = 44
value = [0, 0, 11, 33, 0]
class = drugX
gini = 0.0
samples = 16
value = [16, 0, 0, 0, 0]
class = drugA
gini = 0.0
samples = 11
value = [0, 11, 0, 0, 0]
class = drugB
$X_2 \leq 1.5$
gini = 0.499
samples = 21
value = [0, 0, 11, 10, 0]
class = drugC
gini = 0.0
samples = 23
value = [0, 0, 0, 23, 0]
class = drugX
gini = 0.0
samples = 11
value = [0, 0, 11, 0, 0]
class = drugC
gini = 0.0
samples = 10
value = [0, 0, 0, 10, 0]
class = drugX



4.8 Comparision

4.8.1 At last imported GridSearchCV for comparision and compare both the predicted accuracy score with testing values.

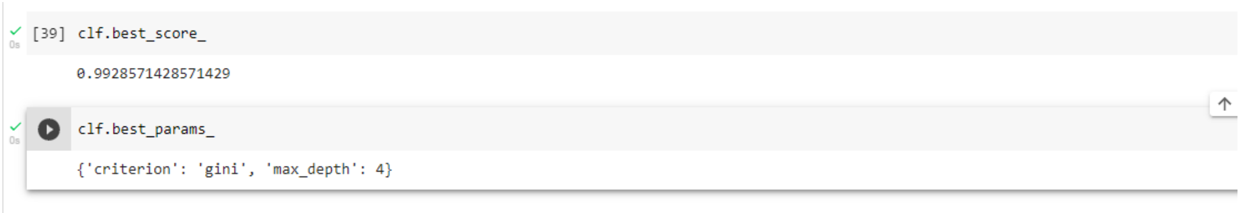


```
from sklearn.model_selection import GridSearchCV
tuned_parameters = [{'criterion':['gini','entropy'],'max_depth':range(2,10)}]
clf_tree = DecisionTreeClassifier()
#scoring = ['precision_macro', 'balanced_accuracy']
clf = GridSearchCV(clf_tree,tuned_parameters,n_jobs=1)
clf.fit(X_trainset, Y_trainset)
```

```
GridSearchCV(cv=None, error_score=nan,
             estimator=DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None,
                                              criterion='gini', max_depth=None,
                                              max_features=None,
                                              max_leaf_nodes=None,
                                              min_impurity_decrease=0.0,
                                              min_impurity_split=None,
                                              min_samples_leaf=1,
                                              min_samples_split=2,
                                              min_weight_fraction_leaf=0.0,
                                              presort='deprecated',
                                              random_state=None,
                                              splitter='best'),
             iid='deprecated', n_jobs=1,
```

✓ 0s completed at 1:23 PM

4.8.2 And it was 99.2 percentile by criterian Gini and maximum depth 4.



```
[39] clf.best_score_

     0.9928571428571429

     clf.best_params_

     {'criterion': 'gini', 'max_depth': 4}
```

So,finalized the Model.

5. CONCLUSION

Above Decision Tree model shows how algorithm works and will be useful for future reference.Tree gives visual representation of all possible outcomes **.**

THANKYOU !