A Report on Open Ended Activity

Submitted for the course

"Foundations of Data Science"
V Semester, B Section

by

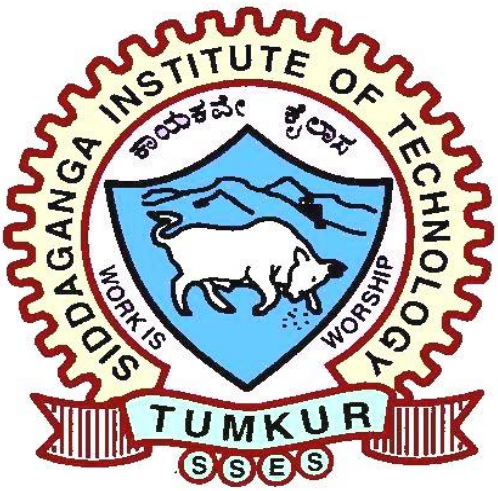
Name : Mohammed Sayeed USN : 1si18cs057

Name : MNM Varun USN : 1si18cs052

under the guidance of

K BHARGAVI

Assistant Professor, Dept of CSE, SIT



Department of Computer Science & Engineering
Siddaganga Institute of Technology, Tumakuru-3
(An Autonomous Institute affiliated to VTU Belagavi, Approved by AICTE)

2020-21

Activity Stats:

Programming Language Used:

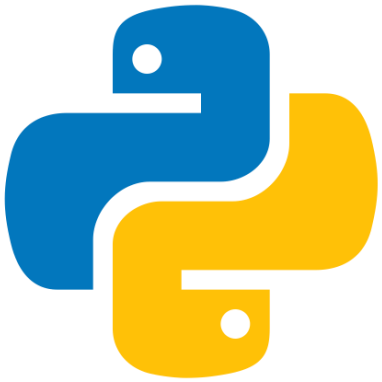
-Python

Tool for Simulation and Visualizing Decision Tree:

-Google Collaborator(Alternative to Jupiter notebook)

Libraries and Dependencies:

-NumPy, Pandas and Matplotlib



Description of Dataset:

This data set contains description of different types of drugs consumed by an individual with respect to various factors like Blood Pressure, Cholesterol and in take ratio of sodium to potassium ratio present in the drug.

Link :
https://drive.google.com/file/d/14A_UFR314mxscQaOEbgBi6lWEkVxiciA/view?usp=sharing

FOUNDATIONS OF DATA SCIENCE

Problem Statement: Building and Visualizing Decision Tree For a data set.

Theory: Decision tree is an algorithm which is mainly applied to data classification scenarios. It is a tree structure where each node represents the features and each edge represents the decision taken. Starting from the root node we go on evaluating the features for classification and take a decision to follow a specific edge. Whenever a new data point comes in, recursively same method is applied again and again and then the final conclusion is taken when all the required features are studied or applied to the classification scenario. Decision tree algorithm is a supervised learning model used in predicting a dependent variable with a series of training variables.

Steps involved in building decision tree:

1.Pre-processing the data: In this step we pre-process our data to get numeric values for different text values we have in the data. This is useful to train and test the sample data about the decision to use certain drug for a given value of age, sex, BP etc.

2.Converting the dependent variable: we also convert the dependent variable into numerical values so that it can be used in the training as well as the evaluation data set.

3.Training the dataset: Next we use 30 percent of the supplied data as a training data set. This will be use as the basis for creating the classification for the remaining 70 percentages which we will call as test data.

4.Getting the result from trained dataset: Now we will apply the decision tree to see the result for the trained data set. Here we will create a tree image based on the input we have and using the criteria called entropy. And finally, we calculate the accuracy of the decision tree.

```
# The PreRequisites that we actually need

from six import StringIO
from sklearn import tree
import matplotlib.image as mpimg
import pydotplus
import matplotlib.pyplot as plt
from sklearn import metrics
import numpy as np
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
```

```
# read and process the data from csv file
my_data = pd.read_csv("drug200.csv", delimiter=",")
my_data[0:5]
my_data.shape
X = my_data[['Age', 'Sex', 'BP', 'Cholesterol', 'Na_to_K']].values

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])
```

```
#Training Dataset
le_Chol = preprocessing.LabelEncoder()
le_Chol.fit(['NORMAL', 'HIGH'])
X[:, 3] = le_Chol.transform(X[:, 3])
y = my_data["Drug"]

X_trainset, X_testset, y_trainset, y_testset = train_test_split(
    X, y, test_size=0.3, random_state=3)
# set up the tree properties
drugTree = DecisionTreeClassifier(criterion="entropy", max_depth=4)
drugTree
drugTree.fit(X_trainset, y_trainset)
predTree = drugTree.predict(X_testset)
print(" Accuracy: ", metrics.accuracy_score(y_testset, predTree))
%matplotlib inline
```

```
# building the image
dot_data = StringIO()
filename = "drugtree.png"
featureNames = my_data.columns[0:5]
targetNames = my_data["Drug"].unique().tolist()
out = tree.export_graphviz(drugTree, feature_names=featureNames,
 out_file=dot_data, class_names=np.unique(
    y_trainset), filled=True,  special_characters=True, rotate=F
alse)
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_png(filename)
img = mpimg.imread(filename)
plt.figure(figsize=(100, 200))
plt.imshow(img, interpolation='nearest')
```

Final Code for the problem :

```
# The PreRequisites that we actually need

from six import StringIO
from sklearn import tree
import matplotlib.image as mpimg
import pydotplus
import matplotlib.pyplot as plt
from sklearn import metrics
import numpy as np
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn import preprocessing
from sklearn.model_selection import train_test_split

# read and process the data
my_data = pd.read_csv("drug200.csv", delimiter=",")
my_data[0:5]
my_data.shape
X = my_data[['Age', 'Sex', 'BP', 'Cholesterol', 'Na_to_K']].values

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])

#Training Dataset
le_Chol = preprocessing.LabelEncoder()
le_Chol.fit(['NORMAL', 'HIGH'])
X[:, 3] = le_Chol.transform(X[:, 3])
y = my_data["Drug"]

X_trainset, X_testset, y_trainset, y_testset = train_test_split(
    X, y, test_size=0.3, random_state=3)

# set up the tree properties
drugTree = DecisionTreeClassifier(criterion="entropy", max_depth=7)
drugTree
```

```
drugTree.fit(X_trainset, y_trainset)
predTree = drugTree.predict(X_testset)

print(" Accuracy: ", metrics.accuracy_score(y_testset, predTree))
%matplotlib inline

# building the image
dot_data = StringIO()
filename = "drugtree.png"
featureNames = my_data.columns[0:5]
targetNames = my_data["Drug"].unique().tolist()
out = tree.export_graphviz(drugTree, feature_names=featureNames,
out_file=dot_data, class_names=np.unique(
   y_trainset), filled=True,  special_characters=True, rotate=False)

graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_png(filename)
img = mpimg.imread(filename)
plt.figure(figsize=(100, 200))
plt.imshow(img, interpolation='nearest')
```

Output :

```
DecisionTrees's Accuracy:  0.9890829694323144
<matplotlib.image.AxesImage at 0x7f203c1639e8>
```