DMML Lab Assignment

Shubham Kumar*

April 12, 2021

1 Aim:

To write a program in python to create a decision tree.
The program takes a labeled dataset as input and produces a decision tree.
Data set is taken in a ratio of 70:30 for training and data.

2 Code and Output:

[1]:
```
import pandas as pd
from sklearn import *
from sklearn.tree import export_graphviz
from sklearn.metrics import *
```

[2]:
```
df = pd.read_csv('hcvdat0.csv')
```

[3]:
```
df.shape
```

[3]:
```
(615, 14)
```

[4]:
```
df.columns
```

[4]:
```
Index(['Unnamed: 0', 'Category', 'Age', 'Sex', 'ALB', 'ALP', 'ALT', 'AST',
       'BIL', 'CHE', 'CHOL', 'CREA', 'GGT', 'PROT'],
      dtype='object')
```

[5]:
```
df[df.isnull().any(axis=1)].head()
```

[5]:

	Unnamed: 0	Category	Age	Sex	ALB	ALP	ALT	AST	BIL	CHE	\
121	122	0=Blood Donor	43	m	48.6	45.0	10.5	40.5	5.3	7.09	
319	320	0=Blood Donor	32	f	47.4	52.5	19.1	17.1	4.6	10.19	
329	330	0=Blood Donor	33	f	42.4	137.2	14.2	13.1	3.4	8.23	
413	414	0=Blood Donor	46	f	42.9	55.1	15.2	29.8	3.6	8.37	
424	425	0=Blood Donor	48	f	45.6	107.2	24.4	39.0	13.8	9.77	

*Mtech-1st yr: Information Security, Atal Bihari Vajpayee Indian Institute of Information Technology and Management, Gwalior, India-474015

```
     CHOL  CREA   GGT  PROT
121   NaN  63.0  25.1  70.0
319   NaN  63.0  23.0  72.2
329   NaN  48.0  25.7  74.4
413   NaN  61.0  29.0  71.9
424   NaN  88.0  38.0  75.1
```

[6]:
```
df= df.dropna()
```

[7]:
```
df.shape
```

[7]:
```
(589, 14)
```

[8]:
```
df.drop(df.columns[0],axis=1,inplace=True)
```

[9]:
```
df.head()
```

[9]:
```
         Category  Age Sex   ALB   ALP   ALT   AST   BIL    CHE  CHOL   CREA  \
0  0=Blood Donor   32   m  38.5  52.5   7.7  22.1   7.5   6.93  3.23  106.0
1  0=Blood Donor   32   m  38.5  70.3  18.0  24.7   3.9  11.17  4.80   74.0
2  0=Blood Donor   32   m  46.9  74.7  36.2  52.6   6.1   8.84  5.20   86.0
3  0=Blood Donor   32   m  43.2  52.0  30.6  22.6  18.9   7.33  4.74   80.0
4  0=Blood Donor   32   m  39.2  74.1  32.6  24.8   9.6   9.15  4.32   76.0

    GGT  PROT
0  12.1  69.0
1  15.6  76.5
2  33.2  79.3
3  33.8  75.7
4  29.9  68.7
```

[10]:
```
from sklearn.preprocessing import LabelEncoder
```

[11]:
```
category=LabelEncoder()
sex=LabelEncoder()
df['category_df']=category.fit_transform(df['Category'])
df['sex_df']=category.fit_transform(df['Sex'])
df.head()
```

[11]:
```
         Category  Age Sex   ALB   ALP   ALT   AST   BIL    CHE  CHOL   CREA  \
0  0=Blood Donor   32   m  38.5  52.5   7.7  22.1   7.5   6.93  3.23  106.0
1  0=Blood Donor   32   m  38.5  70.3  18.0  24.7   3.9  11.17  4.80   74.0
2  0=Blood Donor   32   m  46.9  74.7  36.2  52.6   6.1   8.84  5.20   86.0
3  0=Blood Donor   32   m  43.2  52.0  30.6  22.6  18.9   7.33  4.74   80.0
4  0=Blood Donor   32   m  39.2  74.1  32.6  24.8   9.6   9.15  4.32   76.0
```

	GGT	PROT	category_df	sex_df
0	12.1	69.0	0	1
1	15.6	76.5	0	1
2	33.2	79.3	0	1
3	33.8	75.7	0	1
4	29.9	68.7	0	1

[12]:
```
data_class=df['Category'].copy()
data_class=list(dict.fromkeys(data_class))
print(data_class)
df.head()
```

```
['0=Blood Donor', '0s=suspect Blood Donor', '1=Hepatitis', '2=Fibrosis',
'3=Cirrhosis']
```

[12]:

	Category	Age	Sex	ALB	ALP	ALT	AST	BIL	CHE	CHOL	CREA	\
0	0=Blood Donor	32	m	38.5	52.5	7.7	22.1	7.5	6.93	3.23	106.0	
1	0=Blood Donor	32	m	38.5	70.3	18.0	24.7	3.9	11.17	4.80	74.0	
2	0=Blood Donor	32	m	46.9	74.7	36.2	52.6	6.1	8.84	5.20	86.0	
3	0=Blood Donor	32	m	43.2	52.0	30.6	22.6	18.9	7.33	4.74	80.0	
4	0=Blood Donor	32	m	39.2	74.1	32.6	24.8	9.6	9.15	4.32	76.0	

	GGT	PROT	category_df	sex_df
0	12.1	69.0	0	1
1	15.6	76.5	0	1
2	33.2	79.3	0	1
3	33.8	75.7	0	1
4	29.9	68.7	0	1

[13]:
```
from sklearn.model_selection import train_test_split
```

[14]:
```
df = df.drop(['Category','Sex'], axis=1)
y=df['category_df'].copy()
X = df.drop(['category_df'], axis=1)
X_train, X_test, y_train, y_test =train_test_split(X,y,test_size=0.
 ↪3,random_state=500)
```

[15]:
```
from sklearn.tree import DecisionTreeClassifier
```

[16]:
```
model = DecisionTreeClassifier(max_leaf_nodes = 15, random_state = 0 )
model.fit(X_train,y_train)
```

[16]:
```
DecisionTreeClassifier(max_leaf_nodes=15, random_state=0)
```

[17]:
```
from sklearn.metrics import accuracy_score,␣
 ↪confusion_matrix,classification_report
```

[18]:
```
y_predicted = model.predict(X_test)
print("Accuracy : ", accuracy_score(y_test,y_predicted)*100)
print("Report : ", classification_report(y_test,y_predicted))
print(confusion_matrix(y_test,y_predicted))
```

```
Accuracy :  94.35028248587571
Report :                precision    recall  f1-score   support

           0       0.99      0.98      0.98       164
           1       0.00      0.00      0.00         1
           2       0.40      0.50      0.44         4
           3       0.50      0.50      0.50         2
           4       0.57      0.67      0.62         6

    accuracy                           0.94       177
   macro avg       0.49      0.53      0.51       177
weighted avg       0.95      0.94      0.95       177

[[160   1   0   0   3]
 [  1   0   0   0   0]
 [  0   1   2   1   0]
 [  0   0   1   1   0]
 [  0   0   2   0   4]]
```

[19]:
```
dot_data = export_graphviz( model,out_file = 'tree.dot',
                           feature_names = X_train.columns, class_names =␣
 ↪data_class,
                           filled = True, rounded = True,
                           special_characters = True
                          )
```

[20]:
```
!dot -Tpng tree.dot -o tree.png
```

[ ]:
```

```