DECISION TREE IN PYTHON

DECISION TREE
ENTROPY

$$E = -\sum p(X) \cdot \log_2(p(X))$$

$$p(X) = \frac{\#x}{n}$$

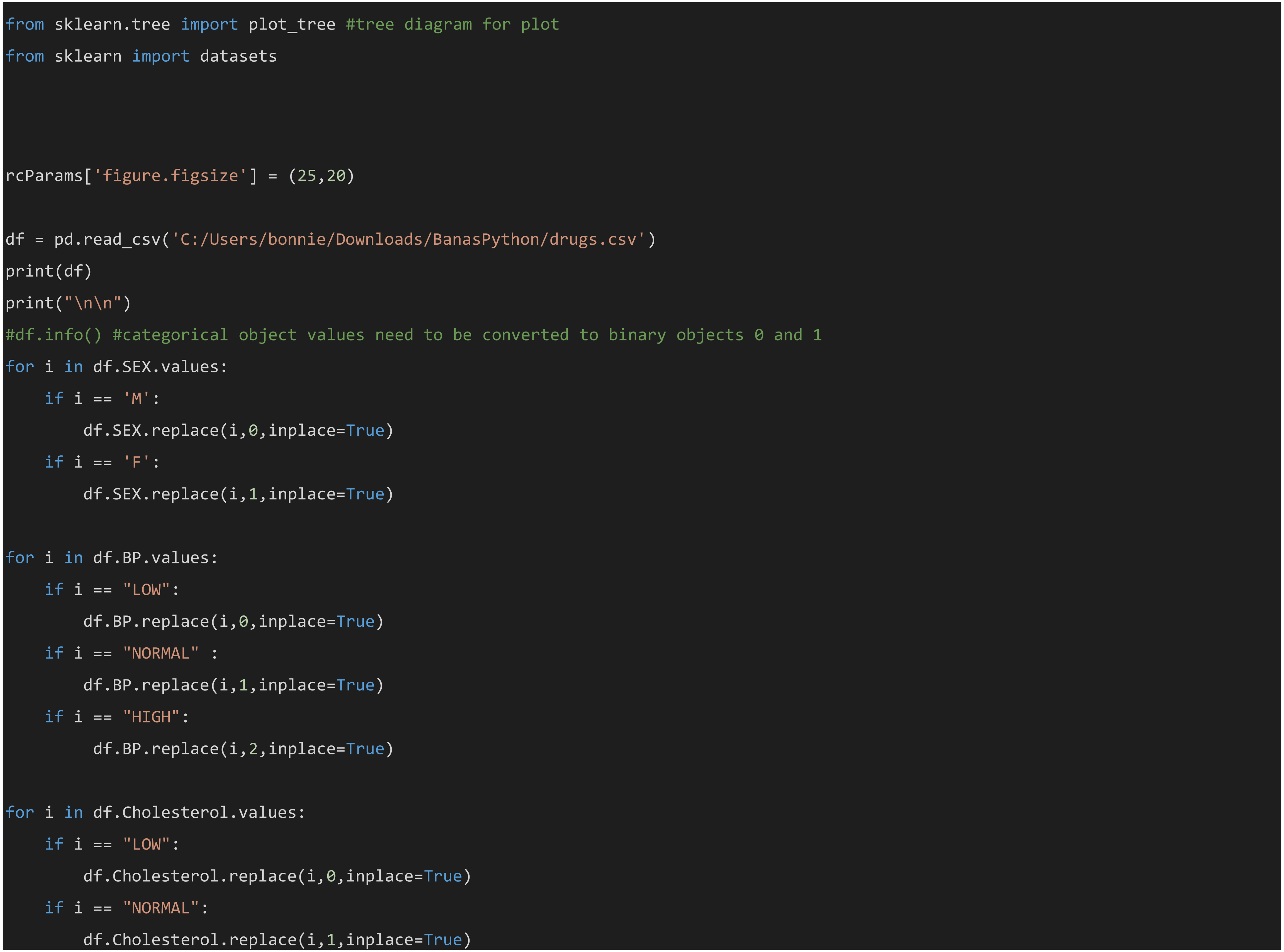
```
def entropy(y):
    hist = np.bincount(y) #calculate the number of occurrances of all class labels
    ps = hist/len(y) #where len(y) is total number of class labels
    return np.sum([p*np.log2(p) for p in ps if p > 0] )
```

## Information Gain

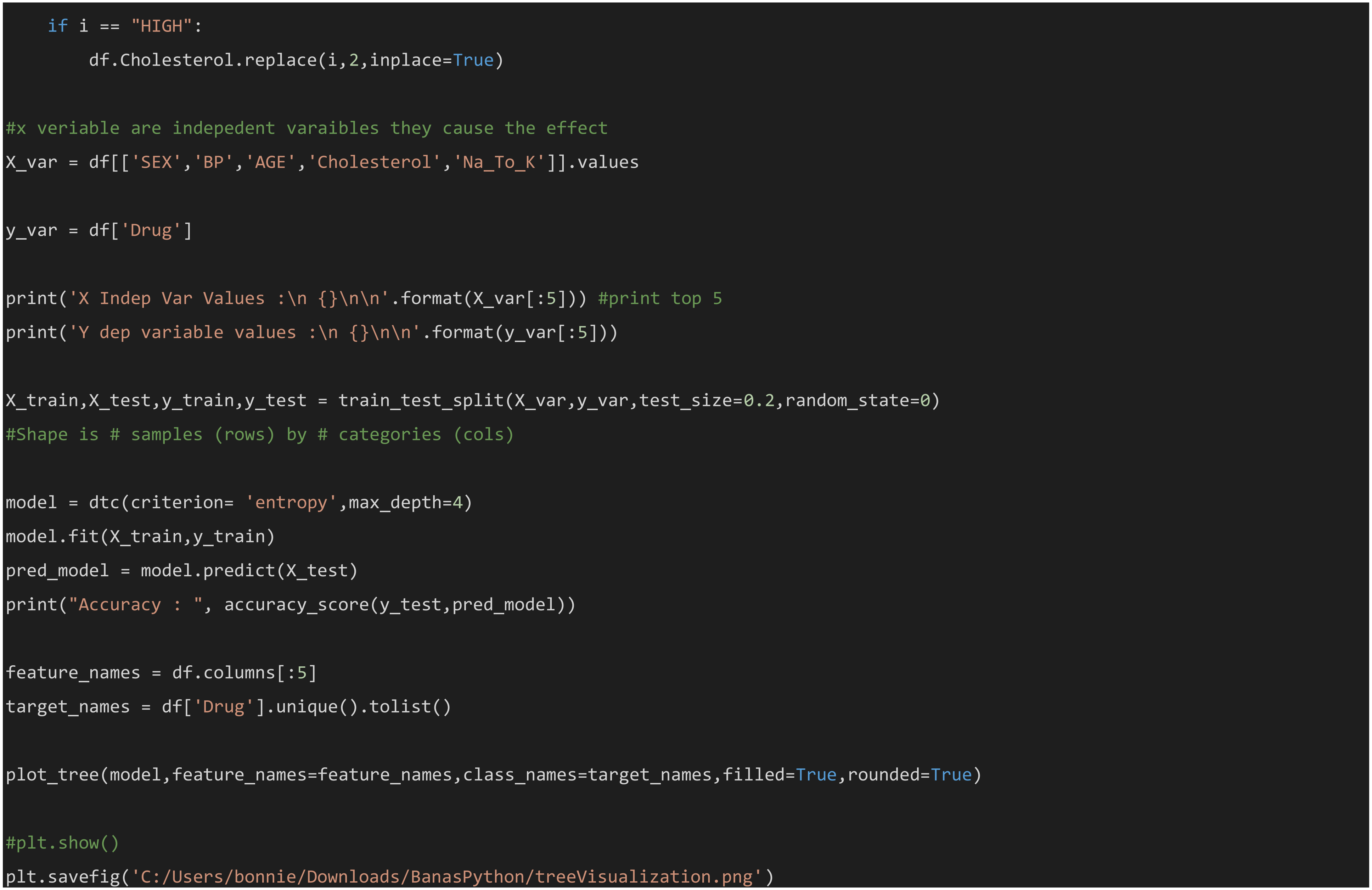
$$IG = E(parent) - [weighted\ average] \cdot E(children)$$

**BEGIN DECISION TREE DEMO**

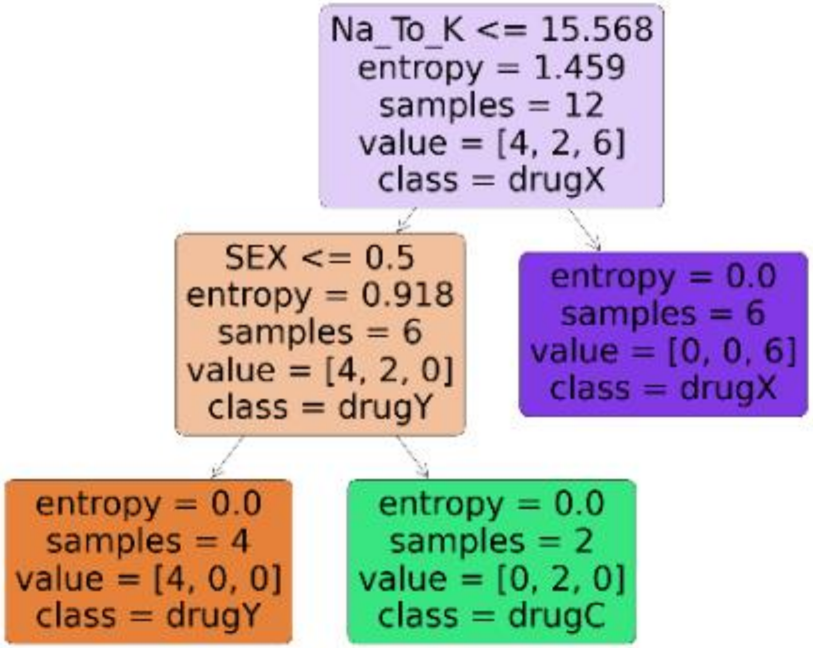
```
#Decision Tree in Python
import pandas as pd  #data processing
import numpy as np  #working with arrays
import matplotlib.pyplot as plt  #visualization
from matplotlib import rcParams #figure size
from sklearn.tree import DecisionTreeClassifier as dtc #tree algorithm
from sklearn.model_selection import train_test_split #splitting the data into training and test data
from sklearn.metrics import accuracy_score #model precision
```



```
from sklearn.tree import plot_tree #tree diagram for plot
from sklearn import datasets



rcParams['figure.figsize'] = (25,20)

df = pd.read_csv('C:/Users/bonnie/Downloads/BanasPython/drugs.csv')
print(df)
print("\n\n")
#df.info() #categorical object values need to be converted to binary objects 0 and 1
for i in df.SEX.values:
    if i == 'M':
        df.SEX.replace(i,0,inplace=True)
    if i == 'F':
        df.SEX.replace(i,1,inplace=True)

for i in df.BP.values:
    if i == "LOW":
        df.BP.replace(i,0,inplace=True)
    if i == "NORMAL" :
        df.BP.replace(i,1,inplace=True)
    if i == "HIGH":
         df.BP.replace(i,2,inplace=True)

for i in df.Cholesterol.values:
    if i == "LOW":
        df.Cholesterol.replace(i,0,inplace=True)
    if i == "NORMAL":
        df.Cholesterol.replace(i,1,inplace=True)
```



```
    if i == "HIGH":
        df.Cholesterol.replace(i,2,inplace=True)

#x veriable are indepedent varaibles they cause the effect
X_var = df[['SEX','BP','AGE','Cholesterol','Na_To_K']].values

y_var = df['Drug']

print('X Indep Var Values :\n {}\n\n'.format(X_var[:5])) #print top 5
print('Y dep variable values :\n {}\n\n'.format(y_var[:5]))

X_train,X_test,y_train,y_test = train_test_split(X_var,y_var,test_size=0.2,random_state=0)
#Shape is # samples (rows) by # categories (cols)

model = dtc(criterion= 'entropy',max_depth=4)
model.fit(X_train,y_train)
pred_model = model.predict(X_test)
print("Accuracy : ", accuracy_score(y_test,pred_model))

feature_names = df.columns[:5]
target_names = df['Drug'].unique().tolist()

plot_tree(model,feature_names=feature_names,class_names=target_names,filled=True,rounded=True)

#plt.show()
plt.savefig('C:/Users/bonnie/Downloads/BanasPython/treeVisualization.png')
```


Na_To_K <= 15.568
entropy = 1.459
samples = 12
value = [4, 2, 6]
class = drugX
SEX <= 0.5
entropy = 0.918
samples = 6
value = [4, 2, 0]
class = drugY
entropy = 0.0
samples = 6
value = [0, 0, 6]
class = drugX
entropy = 0.0
samples = 4
value = [4, 0, 0]
class = drugY
entropy = 0.0
samples = 2
value = [0, 2, 0]
class = drugC