VSIT

Business Intelligence
Practical #7

Name	Sanjeev Gupta	**Roll Number**	21302B0023
Class	TYBscIT	**Division**	C
Subject/Course:	Business intelligence		
Topic	Decision Tree Classifier		

What is Decision Tree Classifier?

A decision tree is a flowchart-like tree structure where each internal node denotes the feature, branches denote the rules and the leaf nodes denote the result of the algorithm. It is a versatile supervised machine-learning algorithm, which is used for both classification and regression problems. It is one of the very powerful algorithms. And it is also used in Random Forest to train on different subsets of training data, which makes random forest one of the most powerful algorithms in machine learning.

A decision tree classifier is composed of one root node, several internal nodes, and several terminal nodes.

Decision tree classifiers are used successfully in many diverse areas. Their most important feature is the capability of capturing descriptive decisionmaking knowledge from the supplied data. Decision tree can be generated from training sets

What is Entropy and Information Gain?

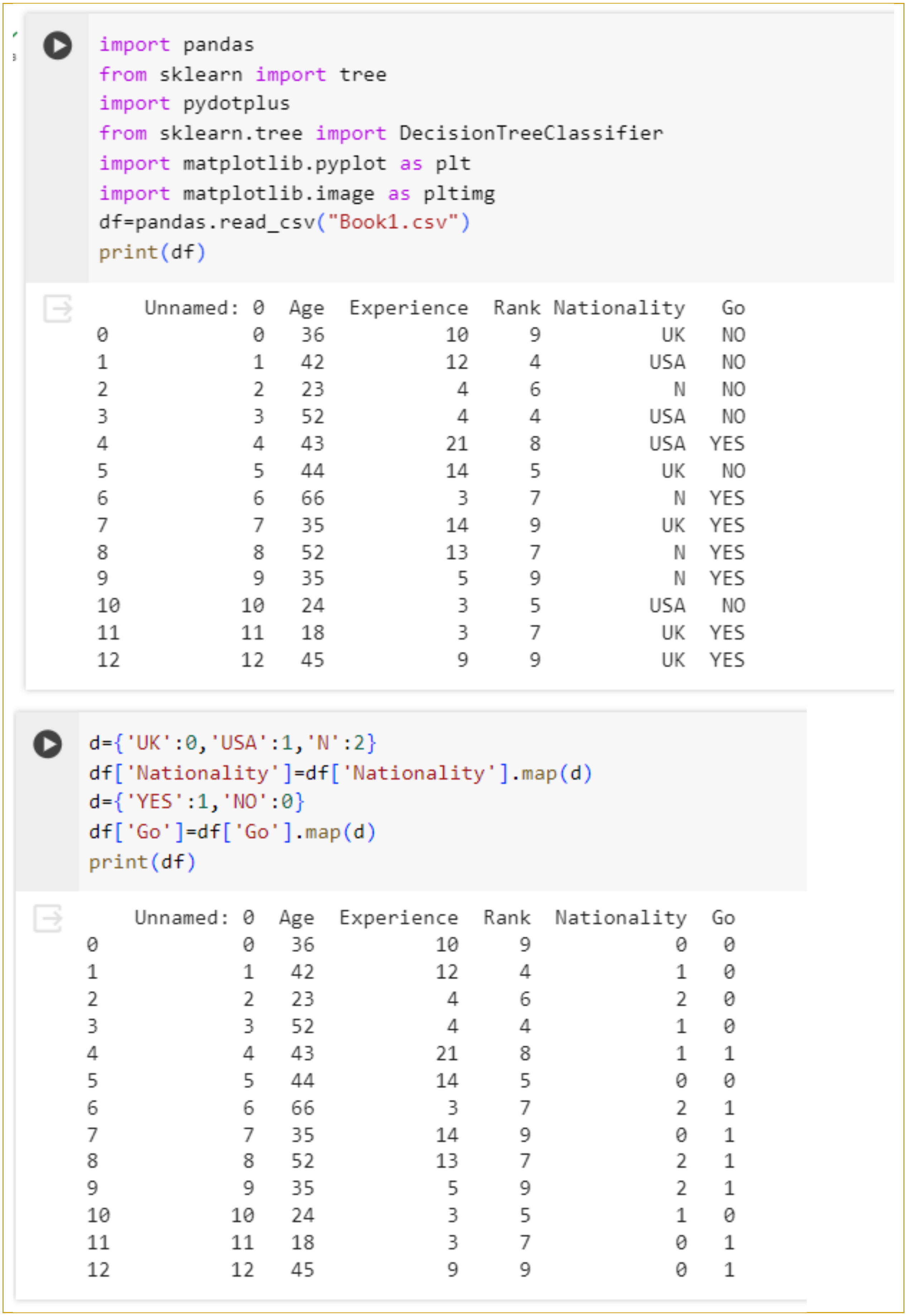
Entropy is the measures of impurity, disorder or uncertainty in a bunch of examples. Entropy controls how a Decision Tree decides to split the data.

Information gain (IG) measures how much "information" a feature gives us about the class.It tells us how important a given attribute of the feature vectors is. **Information gain (IG)** is used to decide the ordering of attributes in the nodes of a decision tree.

Information gain (IG) is calculated as follows:

Information Gain = entropy(parent) – [average entropy(children)]

What are the steps to perform Decision Tree Classifier?

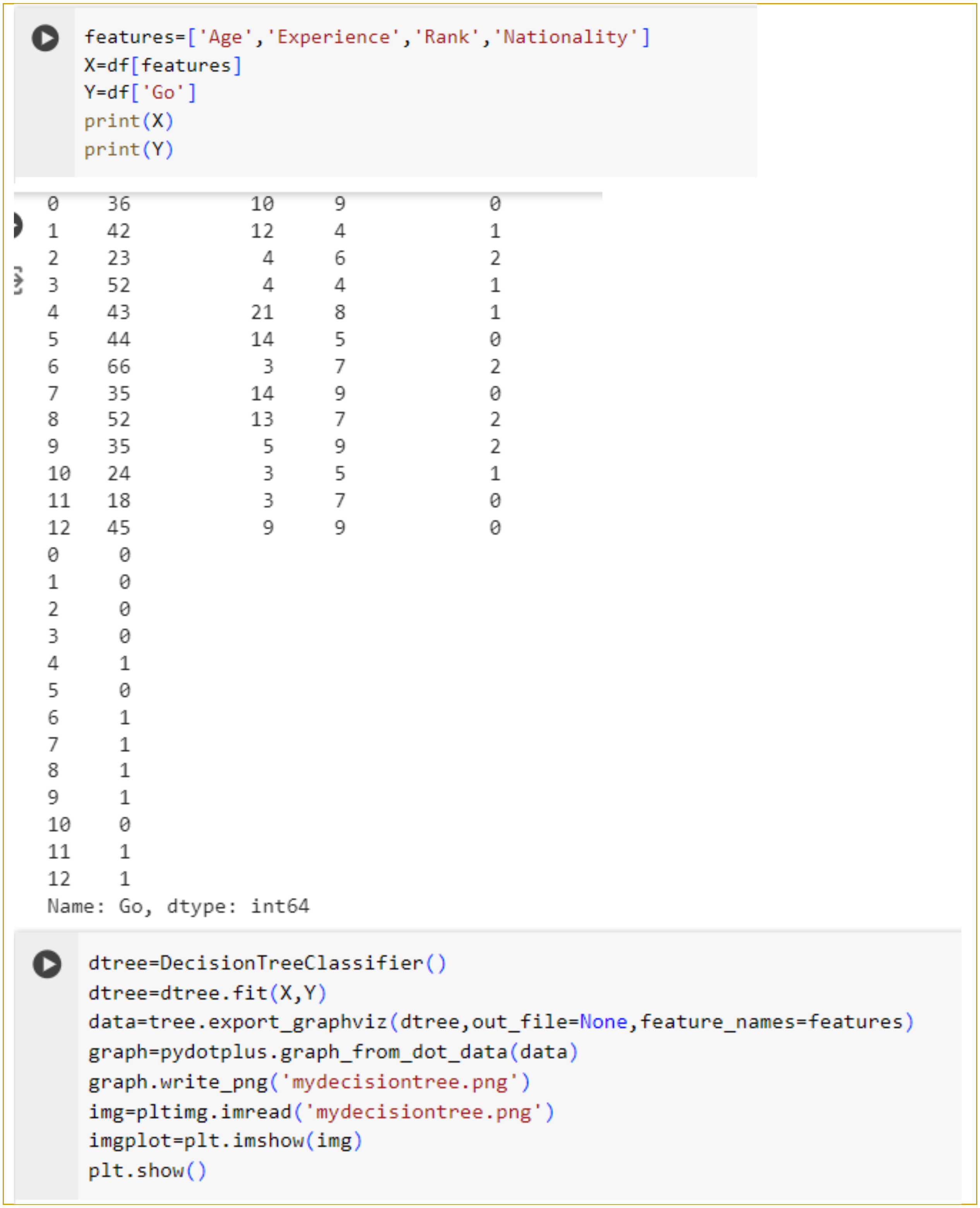


```
import pandas
from sklearn import tree
import pydotplus
from sklearn.tree import DecisionTreeClassifier
import matplotlib.pyplot as plt
import matplotlib.image as pltimg
df=pandas.read_csv("Book1.csv")
print(df)
```

```
    Unnamed: 0  Age  Experience  Rank Nationality   Go
0            0   36          10     9          UK   NO
1            1   42          12     4         USA   NO
2            2   23           4     6           N   NO
3            3   52           4     4         USA   NO
4            4   43          21     8         USA  YES
5            5   44          14     5          UK   NO
6            6   66           3     7           N  YES
7            7   35          14     9          UK  YES
8            8   52          13     7           N  YES
9            9   35           5     9           N  YES
10          10   24           3     5         USA   NO
11          11   18           3     7          UK  YES
12          12   45           9     9          UK  YES
```

```
d={'UK':0,'USA':1,'N':2}
df['Nationality']=df['Nationality'].map(d)
d={'YES':1,'NO':0}
df['Go']=df['Go'].map(d)
print(df)
```

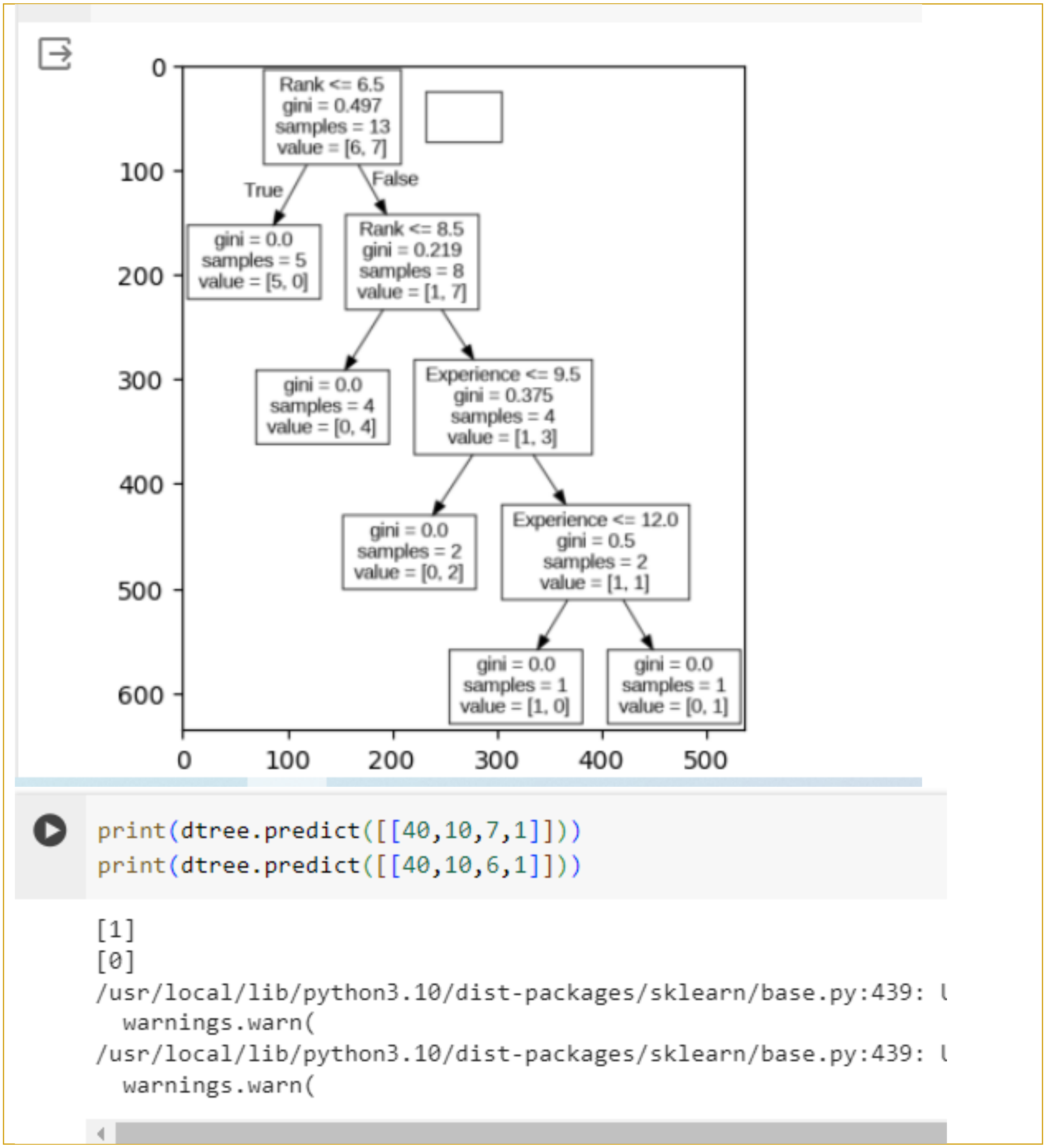
```
    Unnamed: 0  Age  Experience  Rank  Nationality  Go
0            0   36          10     9            0   0
1            1   42          12     4            1   0
2            2   23           4     6            2   0
3            3   52           4     4            1   0
4            4   43          21     8            1   1
5            5   44          14     5            0   0
6            6   66           3     7            2   1
7            7   35          14     9            0   1
8            8   52          13     7            2   1
9            9   35           5     9            2   1
10          10   24           3     5            1   0
11          11   18           3     7            0   1
12          12   45           9     9            0   1
```



```
features=['Age','Experience','Rank','Nationality']
X=df[features]
Y=df['Go']
print(X)
print(Y)
```

```
0    36          10    9          0
1    42          12    4          1
2    23           4    6          2
3    52           4    4          1
4    43          21    8          1
5    44          14    5          0
6    66           3    7          2
7    35          14    9          0
8    52          13    7          2
9    35           5    9          2
10   24           3    5          1
11   18           3    7          0
12   45           9    9          0
0     0
1     0
2     0
3     0
4     1
5     0
6     1
7     1
8     1
9     1
10    0
11    1
12    1
Name: Go, dtype: int64
```

```
dtree=DecisionTreeClassifier()
dtree=dtree.fit(X,Y)
data=tree.export_graphviz(dtree,out_file=None,feature_names=features)
graph=pydotplus.graph_from_dot_data(data)
graph.write_png('mydecisiontree.png')
img=pltimg.imread('mydecisiontree.png')
imgplot=plt.imshow(img)
plt.show()
```



```
print(dtree.predict([[40,10,7,1]]))
print(dtree.predict([[40,10,6,1]]))
```

```
[1]
[0]
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: U
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: U
  warnings.warn(
```