

Experiment – 7

Student Name: Vivek Kumar

UID: 21BCS8129

Branch: BE-CSE(LEET)

Section/Group: WM-20BCS-616/A

Semester: 5th

Date of Performance: 2/11/2022

Subject Name: Machine Learning Lab

Subject Code: 20CSP-317

1. Aim/Overview of the practical:

Implement Decision Tree and compare the performance with Random Forest on any data set.

2. Task to be done/ Which logistics used:

Implement Decision Tree and compare the performance with Random Forest on any data set.

3. Steps for experiment/practical/Code:

```
from google.colab import drive
drive.mount('/content/drive')

import pandas as pd
data = pd.read_csv('/content/drive/MyDrive/Data/data.csv')
print(data)

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

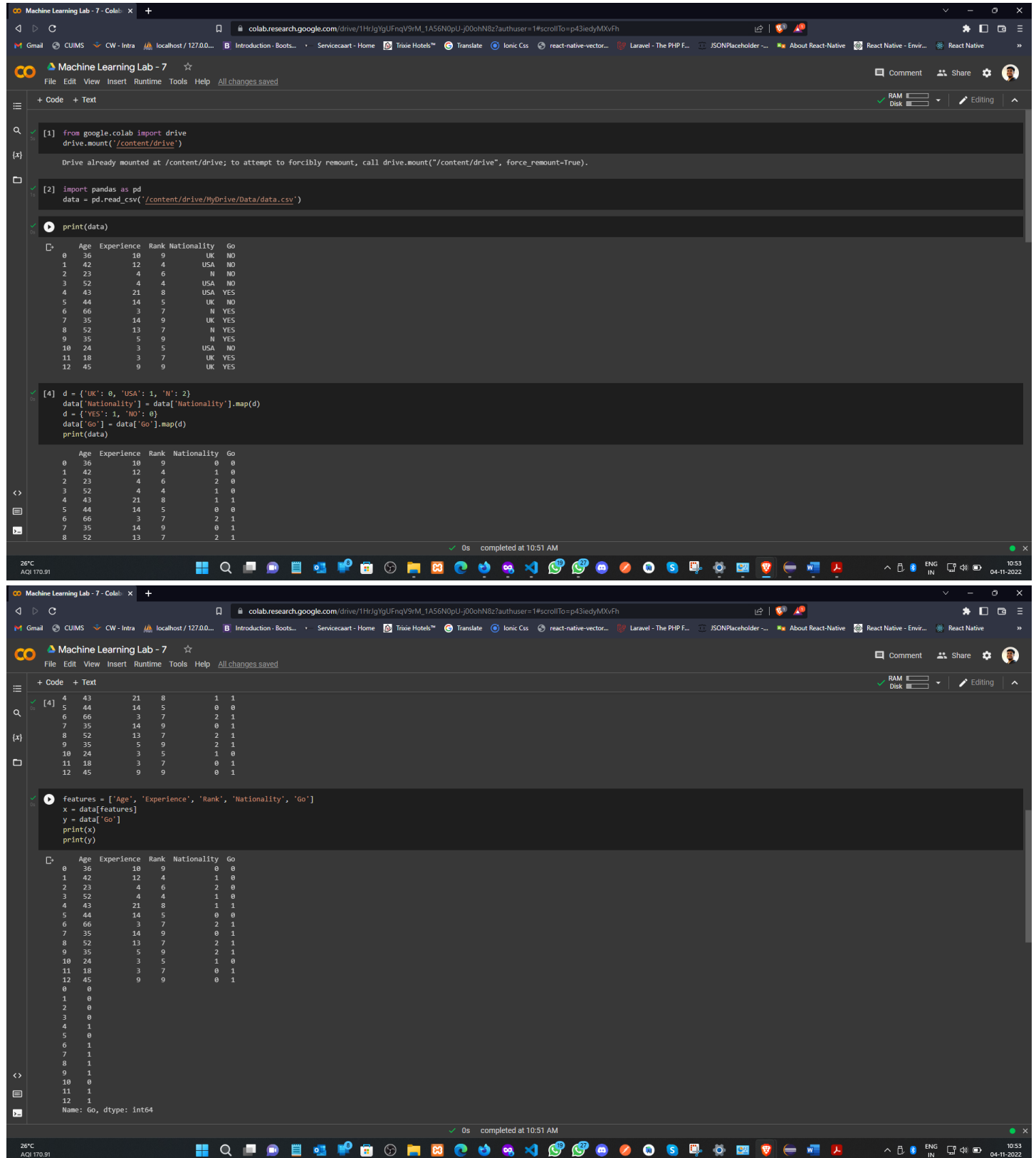
features = ['Age', 'Experience', 'Rank', 'Nationality', 'Go']
x = data[features]
y = data['Go']
print(x)
print(y)

from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
import matplotlib.pyplot as plt

dtree = DecisionTreeClassifier()
dtree = dtree.fit(x,y)
tree.plot_tree(dtree, feature_names=features)

print(dtree.predict([[40, 30, 10, 7, 1]]))
```

4. Result/Output/Writing Summary:

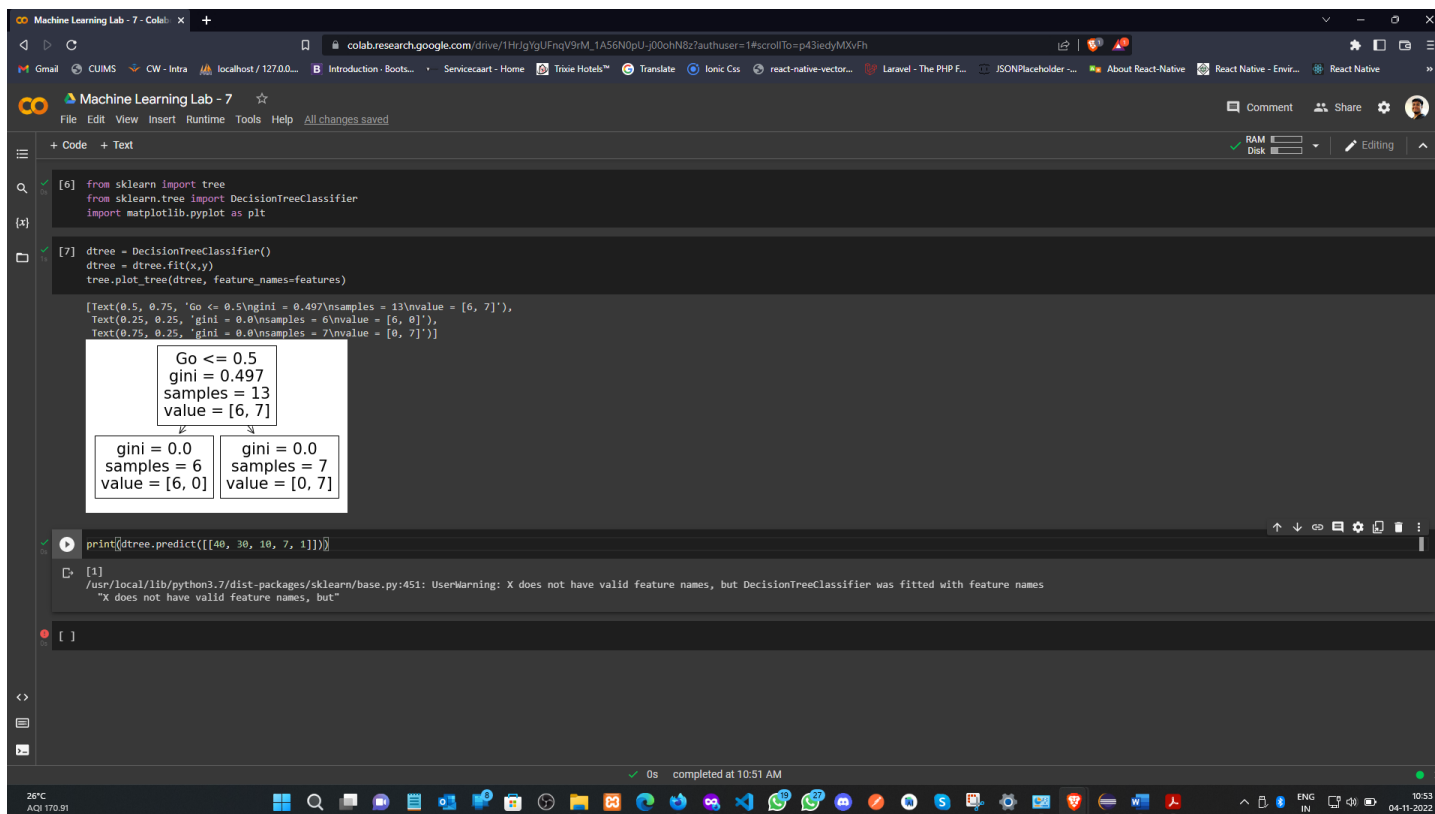


The first screenshot shows the initial code execution in a Google Colab notebook. The code imports the 'drive' module from 'google.colab', mounts the drive, and reads a CSV file named 'data.csv' from the '/content/drive/MyDrive/Data/' directory. The output displays a DataFrame with columns: Age, Experience, Rank, Nationality, and Go. The data is as follows:

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

The second screenshot shows the same notebook with additional code that filters the data based on 'Nationality' and 'Go' values. The code defines dictionaries 'd' for 'Nationality' and 'Go', and uses the 'map' function to filter the DataFrame. The output shows the filtered data, which is a subset of the original data where 'Nationality' is 'USA' or 'N' and 'Go' is 'YES'.

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



```

[6] from sklearn import tree
    from sklearn.tree import DecisionTreeClassifier
    import matplotlib.pyplot as plt

[7] dtree = DecisionTreeClassifier()
    dtree = dtree.fit(x,y)
    tree.plot_tree(dtree, feature_names=features)

[[Text(0.5, 0.75, 'Go <= 0.5\nGini = 0.497\nsamples = 13\nvalue = [6, 7]'),
  Text(0.25, 0.25, 'gini = 0.0\nsamples = 6\nvalue = [6, 0]'),
  Text(0.75, 0.25, 'gini = 0.0\nsamples = 7\nvalue = [0, 7]')]

print(dtree.predict([[40, 30, 10, 7, 1]]))

[1] /usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X does not have valid feature names, but DecisionTreeClassifier was fitted with feature names
  "X does not have valid feature names, but"
  
```

Learning outcomes (What I have learnt):

1. Understood the concept of Decision Tree.
2. Learnt how to load the dataset and map it.
3. Printing the data according to the feature available in the dataset.
4. Plot the Decision Tree and predict it.

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			