1.7.3 Import Libraries in Google Colab

ChatGPT

```
#Importing all the required libraries
import pandas as pd
import numpy as np
import sklearn
from pandas.core.dtypes.common import is_numeric_dtype
from sklearn.preprocessing import LabelEncoder
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
import matplotlib.pyplot as plt
import seaborn as sns
import warnings as wr #Ignores the warnings
wr.filterwarnings('ignore')
df=pd.read_csv("https://raw.githubusercontent.com/shakil1819/CSE442-Machine-Learning-Sessional/main/week%203%20-4/game%20data.
df
```

	Days	Outlook	Temprature	Routine	Wear Jacket?
0	1	Sunny	Cold	Indoor	No
1	2	Cloudy	Cold	Indoor	Yes
2	3	Cloudy	Warm	Outdoor	No
3	4	Sunny	Cold	Outdoor	Yes
4	5	Cloudy	Cold	Outdoor	Yes
5	6	Sunny	Warm	Outdoor	No
6	7	Cloudy	Warm	Indoor	No
7	8	Sunny	Warm	Indoor	No

1.7.5 Data Pre-processing

1.7.5.1 Checking the NULL values

```
df.drop('Days', axis=1, inplace=True)
```

1.7.5.2 Define dependent (target) and independent (predictor) features

```
y=df['Wear Jacket?']
x=df.drop('Wear Jacket?', axis=1)
#Feature Encoding:
for col in x.columns:
  if is_numeric_dtype(x[col]):
    continue
  else:
    x[col]=LabelEncoder().fit_transform(x[col])

```

```
x
```

	Outlook	Temprature	Routine
0	1	0	0
1	0	0	0
2	0	1	1
3	1	0	1
4	0	0	1
5	1	1	1
6	0	1	0
7	1	1	0

```
x.columns
```

```
Index(['Outlook', 'Temprature', 'Routine'], dtype='object')
```

1.7.5.3 Training Decision Tree Classifier

```
clf = DecisionTreeClassifier()
clf.fit(x,y)

```

DecisionTreeClassifier

DecisionTreeClassifier()

1.7.5.4 Prediction

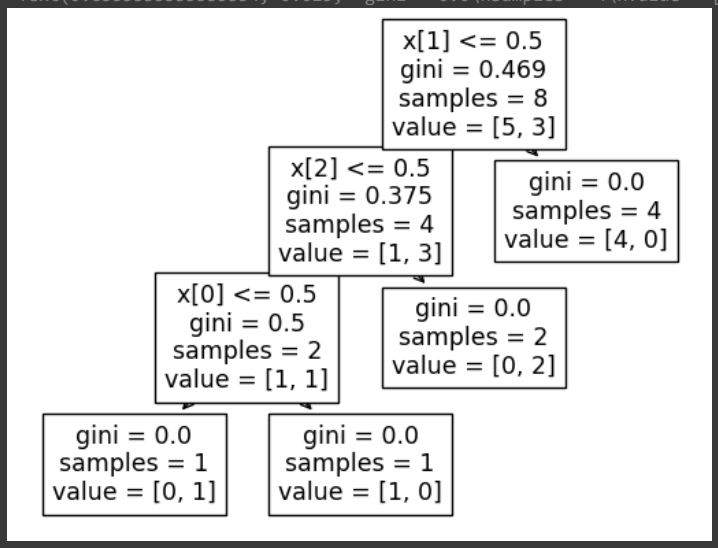
```
clf.predict([[1,0,0]])
clf.predict([[1,0,1]])

```

```
array(['Yes'], dtype=object)
```

1.7.6 Tree Represntation

```
tree.plot_tree(clf)
```

```
[Text(0.6666666666666666, 0.875, 'x[1] <= 0.5\ngini = 0.469\nsamples = 8\nvalue = [5, 3]'),
 Text(0.5, 0.625, 'x[2] <= 0.5\ngini = 0.375\nsamples = 4\nvalue = [1, 3]'),
 Text(0.3333333333333333, 0.375, 'x[0] <= 0.5\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.16666666666666666, 0.125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5, 0.125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.6666666666666666, 0.375, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.8333333333333334, 0.625, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]')]
```

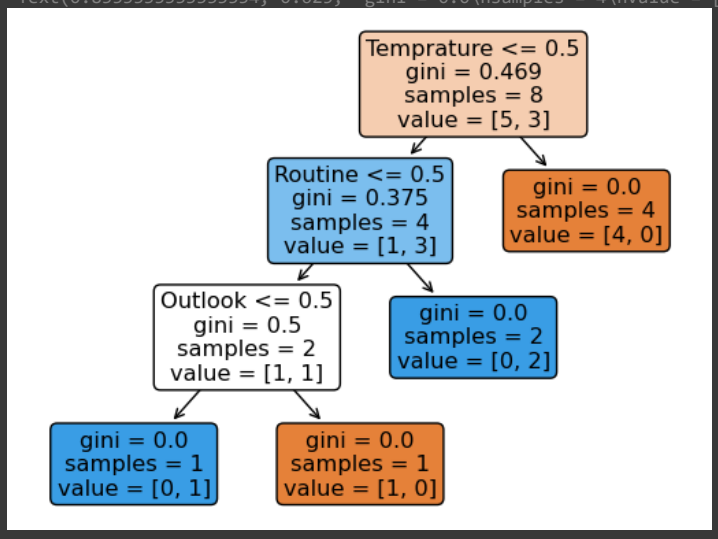



```
tree.plot_tree(clf, rounded=True, filled=True)
```

```
[Text(0.6666666666666666, 0.875, 'x[1] <= 0.5\ngini = 0.469\nsamples = 8\nvalue = [5, 3]'),
 Text(0.5, 0.625, 'x[2] <= 0.5\ngini = 0.375\nsamples = 4\nvalue = [1, 3]'),
```

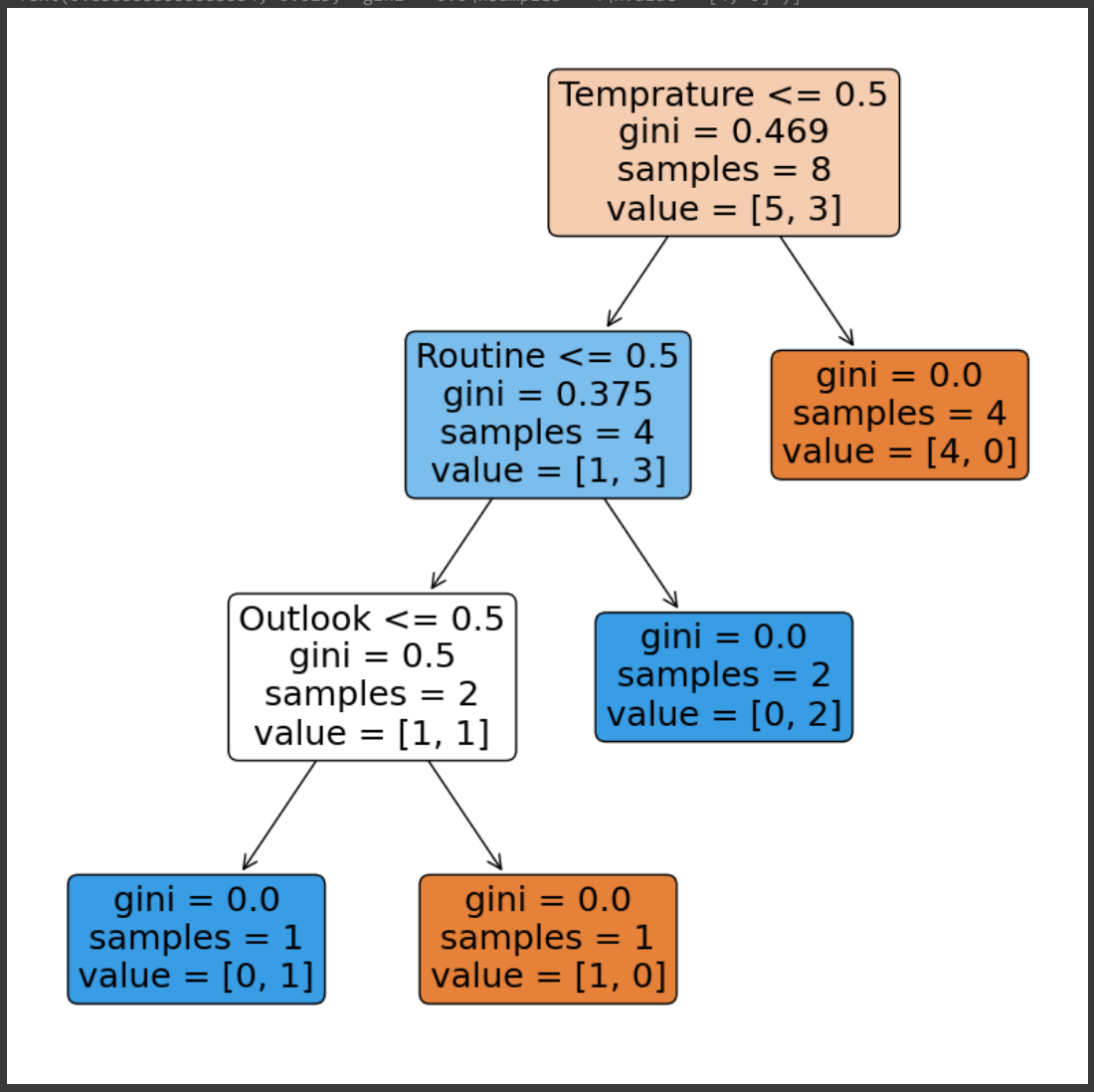
```
1 tree.plot_tree(clf, rounded=True, filled=True, feature_names=x.columns)
```

```
[Text(0.6666666666666666, 0.875, 'Temprature <= 0.5\ngini = 0.469\nsamples = 8\nvalue = [5,
3]'),
 Text(0.5, 0.625, 'Routine <= 0.5\ngini = 0.375\nsamples = 4\nvalue = [1, 3]'),
 Text(0.3333333333333333, 0.375, 'Outlook <= 0.5\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.16666666666666666, 0.125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5, 0.125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.6666666666666666, 0.375, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.8333333333333334, 0.625, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]')]
```




```
1 plt.figure(figsize=(10,10))
2 tree.plot_tree(clf, rounded=True, filled=True, feature_names=x.columns)
```

```
[Text(0.6666666666666666, 0.875, 'Temprature <= 0.5\ngini = 0.469\nsamples = 8\nvalue = [5, 3]'),
 Text(0.5, 0.625, 'Routine <= 0.5\ngini = 0.375\nsamples = 4\nvalue = [1, 3]'),
 Text(0.3333333333333333, 0.375, 'Outlook <= 0.5\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.16666666666666666, 0.125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5, 0.125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.6666666666666666, 0.375, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.8333333333333334, 0.625, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]')]
```



1.7.7 Text Representation

```
text_rep = tree.export_text(clf)
```

```
print(text_rep)
```

```
|--- feature_1 <= 0.50
|   |--- feature_2 <= 0.50
|   |   |--- feature_0 <= 0.50
|   |   |   |--- class: Yes
|   |   |--- feature_0 >  0.50
|   |   |   |--- class: No
|   |--- feature_2 >  0.50
|   |   |--- class: Yes
|--- feature_1 >  0.50
|   |--- class: No
```