Decision tree and Random forest

```
import pandas as pd
import numpy as np
from sklearn.tree import DecisionTreeClassifier
df=pd.read_csv("/content/demodt.txt",sep=",")
features=pd.DataFrame(columns=["Literacy","Cleanliness","Crime_Rate"])
features["Literacy"]=df["Literacy"]
features["Cleanliness"]=df["Cleanliness"]
features["Crime_Rate"]=df["Crime_Rate"]
target=df["Good"]
model=DecisionTreeClassifier()
model.fit(features,target)
Literacy=int(input("Enter Literacy Rate:"))
Cleanliness=int(input("Enter Cleanliness Rate:"))
Crime_Rate=int(input("Enter Crime Rate:"))
```

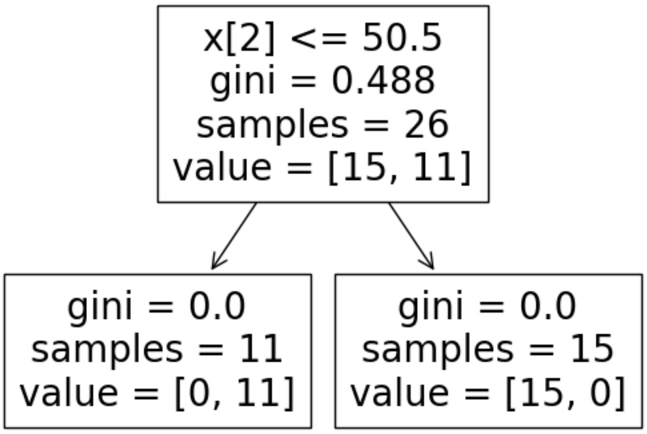
```
Enter Literacy Rate:30
Enter Cleanliness Rate:55
Enter Crime Rate:22
```

```
p=model.predict([[Literacy,Cleanliness,Crime_Rate]]) #cr=90
if(p==1):
  print("This State is in Good State")
else:
  print("This State is in Bad State")
```

```
This State is in Good State
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but DecisionTreeClass
  warnings.warn(
```

```
from sklearn.tree import export_graphviz,plot_tree
plot_tree(model)
```

```
[Text(0.5, 0.75, 'x[2] <= 50.5\ngini = 0.488\nsamples = 26\nvalue = [15, 11]'),
 Text(0.25, 0.25, 'gini = 0.0\nsamples = 11\nvalue = [0, 11]'),
 Text(0.75, 0.25, 'gini = 0.0\nsamples = 15\nvalue = [15, 0]')]
```



Random Forest

```
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestClassifier
df=pd.read_csv("/content/demodt.txt",sep=",")
features=pd.DataFrame(columns=["Literacy","Cleanliness","Crime_Rate"])
features["Literacy"]=df["Literacy"]
features["Cleanliness"]=df["Cleanliness"]
features["Crime_Rate"]=df["Crime_Rate"]
target=df["Good"]
model=RandomForestClassifier(n_estimators=10)#No of trees in the forest
#more the number more is the accuracy and time taken to run
#default values is 100 and it is increased from 10 to 100 in version 0.22 of sklearn.ensemble
model.fit(features,target)
Literacy=int(input("Enter Literacy Rate:"))
Cleanliness=int(input("Enter Cleanliness Rate:"))
Crime_Rate=int(input("Enter Crime Rate:"))
p=model.predict([[Literacy,Cleanliness,Crime_Rate]]) #cr=90
if(p==1):
  print("This State is in Good State")
else:
  print("This State is in Bad State")
```

```
Enter Literacy Rate:30
Enter Cleanliness Rate:22
Enter Crime Rate:11
This State is in Good State
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but RandomForestClass
  warnings.warn(
```

```
#Saving all decision trees of RandomForest
import os
output_dir="tree_visualizations"
os.makedirs(output_dir,exist_ok=True)


for i , tree in enumerate(model.estimators_):
  tree_dot_file=os.path.join(output_dir,f"tree_{i}.dot")
  tree_png_file=os.path.join(output_dir,f"tree_{i}.png")



from sklearn.tree import export_graphviz
export_graphviz(tree, out_file=tree_dot_file, feature_names=["Literacy", "Cleanliness", "Crime_Rate"],
                    class_names=[str(cls) for cls in model.classes_], filled=True, rounded=True)


command=f"dot -Tpng {tree_dot_file} -o {tree_png_file}"
os.system(command)
print(f"tree {i} visualization saved to {tree_png_file}")
```

```
tree 9 visualization saved to tree_visualizations/tree_9.png
```