Experiment 2.4

Student Name: Jitesh Kumar **UID: 20BCS2334**
Branch: CSE **Section/Group 20BCS_WM_903(A)**
Semester: 5th **Date of Performance:11/4/22**
Subject Name: Machine Learning Lab **Subject Code: CSP-317**

1. Aim/Overview of the practical:

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

2. Source Code:

```
import pandas as pd
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
import matplotlib.pyplot as plt
    from sklearn.model_selection import train_test_split
    Bill_Auth_data = pd.read_csv('bill_authentication.csv')
    Bill_Auth_data = pd.read_csv('bill_authentication.csv')
   X = Bill_Auth_data.drop('Class',axis=1)
    Y = Bill_Auth_data['Class']
   X_train, X_test, y_train, y_test = train_test_split(
    X, Y, test_size = 0.3, random_state=50)
  dtree = DecisionTreeClassifier()
    dtree.fit(X_train, y_train)
    y_pred = dtree.predict(X_test)
    y_pred
    from sklearn.metrics import confusion_matrix, accuracy_score
    plt.figure(figsize=(50, 50))
```

```
from sklearn.ensemble import RandomForestClassifier
classifier_rf = RandomForestClassifier(random_state=0, n_jobs=-1, max_depth=10,
n_estimators=100, oob_score=True)
classifier_rf.fit(X_train, y_train)
y_pred_RF = classifier_rf.predict(X_test)
y_pred_RF
cm_RF= confusion_matrix(y_test, y_pred_RF)
print(f"Confusion Matrix for RF:\n{cm_RF}\n")
acc_RF = accuracy_score(y_test, y_pred_RF)
print(f"Accuracy Score: {acc_RF}")
```

Result/Output

In [7]:
```
import pandas as pd
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
```

In [8]:
```
Bill_Auth_data = pd.read_csv('bill_authentication.csv')
```

In [9]:
```
Bill_Auth_data.head()
```

Out[9]:

	Variance	Skewness	Curtosis	Entropy	Class
0	3.62160	8.6661	-2.8073	-0.44699	0
1	4.54590	8.1674	-2.4586	-1.46210	0
2	3.86600	-2.6383	1.9242	0.10645	0
3	3.45660	9.5228	-4.0112	-3.59440	0
4	0.32924	-4.4552	4.5718	-0.98880	0

In [11]:
```
X = Bill_Auth_data.drop('Class',axis=1)
Y = Bill_Auth_data['Class']
```

In [13]:
```
X
```

Out[13]:

	Variance	Skewness	Curtosis	Entropy
0	3.62160	8.66610	-2.8073	-0.44699
1	4.54590	8.16740	-2.4586	-1.46210
2	3.86600	-2.63830	1.9242	0.10645
3	3.45660	9.52280	-4.0112	-3.59440
4	0.32924	-4.45520	4.5718	-0.98880
...	...	...	...	...
1367	0.40614	1.34920	-1.4501	-0.55949
1368	-1.38870	-4.87730	6.4774	0.34179
1369	-3.75030	-13.45860	17.5932	-2.77710
1370	-3.56370	-8.38270	12.3930	-1.28230
1371	-2.54190	-0.65804	2.6842	1.19520

1372 rows × 4 columns

In [14]:
```
Y
```

Out[14]:
```
0       0
1       0
2       0
3       0
4       0
       ..
1367    1
1368    1
1369    1
1370    1
1371    1
Name: Class, Length: 1372, dtype: int64
```

In [16]:
```
X_train, X_test, y_train, y_test = train_test_split(
X, Y, test_size = 0.3, random_state=50)
```

In [17]:
```
dtree = DecisionTreeClassifier()
dtree.fit(X_train, y_train)
```

Out[17]:
```
DecisionTreeClassifier()
```

In [18]:
```
 y_pred = dtree.predict(X_test)
```

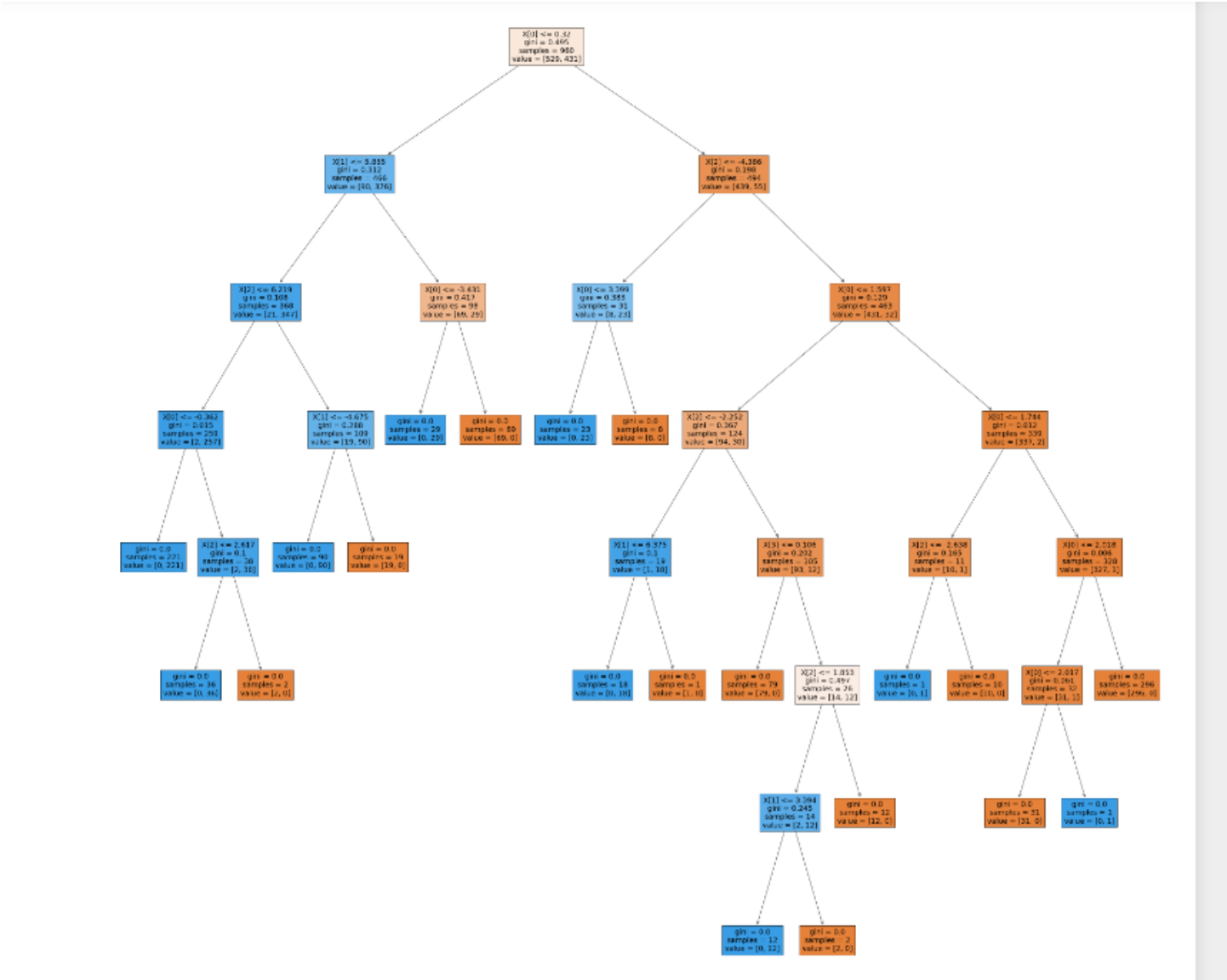
In [22]:
```
from sklearn.metrics import confusion_matrix, accuracy_score
```

In [23]:
```
cm_DT = confusion_matrix(y_test, y_pred)
print(f"Confusion Matrix for DT:\n{cm_DT}\n")
acc_DT = accuracy_score(y_test, y_pred)
print(f"Accuracy Score: {acc_DT}")
```

```
Confusion Matrix for DT:
[[231   2]
 [  4 175]]

Accuracy Score: 0.9854368932038835
```

In [34]:
```
 plt.figure(figsize=(50, 50)) # Resize figure
tree.plot_tree(dtree, filled=True)
plt.show()
```



In [25]:
```
from sklearn.ensemble import RandomForestClassifier
```

In [26]:
```
classifier_rf = RandomForestClassifier(random_state=0, n_jobs=-1, max_depth=10,
n_estimators=100, oob_score=True)
```

In [28]:
```
 classifier_rf.fit(X_train, y_train)
```

Out[28]:
```
RandomForestClassifier(max_depth=10, n_jobs=-1, oob_score=True, random_state=0)
```

In [29]:
```
y_pred_RF = classifier_rf.predict(X_test)
```

In [30]:
```
y_pred_RF
```

Out[30]:
```
array([1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1,
       1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1,
       1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0,
       1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0,
       1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1,
       0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0,
       1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0,
       1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1,
       1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0,
       0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0,
       0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0,
       1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1,
       0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1,
       0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0,
       1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0,
       0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0,
       1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0,
       0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0], dtype=int64)
```

In [31]:
```
cm_RF= confusion_matrix(y_test, y_pred_RF)
print(f"Confusion Matrix for RF:\n{cm_RF}\n")
```

```
Confusion Matrix for RF:
[[232   1]
 [  1 178]]
```

In [33]:
```
acc_RF = accuracy_score(y_test, y_pred_RF)
print(f"Accuracy Score: {acc_RF}")
```

```
Accuracy Score: 0.9951456310679612
```

Learning outcomes (What I have learnt):

1. Learnt to analyze the data.
2. Learnt to import various libraries.
3. Learnt to read csv files.
4. Learnt to implement decision tree and random forest algorithm .
5. Learnt to train and test the data.