```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
df=pd.read_csv("/content/churn.csv")
```

```
df.head()
```

	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrC
0	619	France	Female	42	2	0.00	1	
1	608	Spain	Female	41	1	83807.86	1	
2	502	France	Female	42	8	159660.80	3	
3	699	France	Female	39	1	0.00	2	
4	850	Spain	Female	43	2	125510.82	1	

```
df.isnull().sum()
```

```
CreditScore        0
Geography          0
Gender             0
Age                0
Tenure             0
Balance            0
NumOfProducts      0
HasCrCard          0
IsActiveMember     0
EstimatedSalary    0
Exited             0
dtype: int64
```

```
df.drop(["Gender","Age","IsActiveMember","HasCrCard","EstimatedSalary"], axis="colu
```

	CreditScore	Geography	Tenure	Balance	NumOfProducts	Exited
0	619	France	2	0.00	1	1
1	608	Spain	1	83807.86	1	0
2	502	France	8	159660.80	3	1

```
df['Geography'].unique()
```

```
array(['France', 'Spain', 'Germany'], dtype=object)
```

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
df.Geography=le.fit_transform(df.Geography)
```

```
x=df[['CreditScore','Geography','Tenure','Balance','NumOfProducts']]
y=df['Exited']
```

9999	792	France	4	130142.79	1	0

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test= train_test_split(x,y,train_size=0.7)
```

```
print(x_train)
print(x_test)
print(y_train)
print(y_test)
```

```
      CreditScore  Geography  Tenure    Balance  NumOfProducts
7821          777          0       2  134571.50              1
4272          640          2       3   77826.80              1
958           531          2       8  132576.25              1
74            519          0       9       0.00              2
8712          469          2       5       0.00              2
...           ...        ...     ...        ...            ...
6352          741          0       9       0.00              2
8050          707          0       2       0.00              2
7577          615          1       3   86920.86              1
2652          601          0       0       0.00              2
6750          618          0       2       0.00              4

[7000 rows x 5 columns]
      CreditScore  Geography  Tenure    Balance  NumOfProducts
7687          754          0       5  146622.35              1
4258          782          1       7   98556.89              2
4182          550          1       5  121016.23              1
9908          492          1       9  170295.04              2
5657          496          0       0   90963.49              1
...           ...        ...     ...        ...            ...
6317          450          0       7  117199.80              1
2080          721          0       3   44020.89              1
5744          749          2       1  124209.02              1
6712          599          0       4       0.00              1
150           754          2       7       0.00              2

[3000 rows x 5 columns]
7821    0
```

```
4272    0
958     0
74      0
8712    0
       ..
6352    0
8050    0
7577    0
2652    0
6750    1
Name: Exited, Length: 7000, dtype: int64
7687    1
4258    0
4182    1
9908    0
5657    0
       ..
6317    0
2080    1
5744    0
6712    0
150     0
Name: Exited, Length: 3000, dtype: int64
```

```
from sklearn.tree import DecisionTreeClassifier
```

```
treemodel= DecisionTreeClassifier(max_depth=3)
```

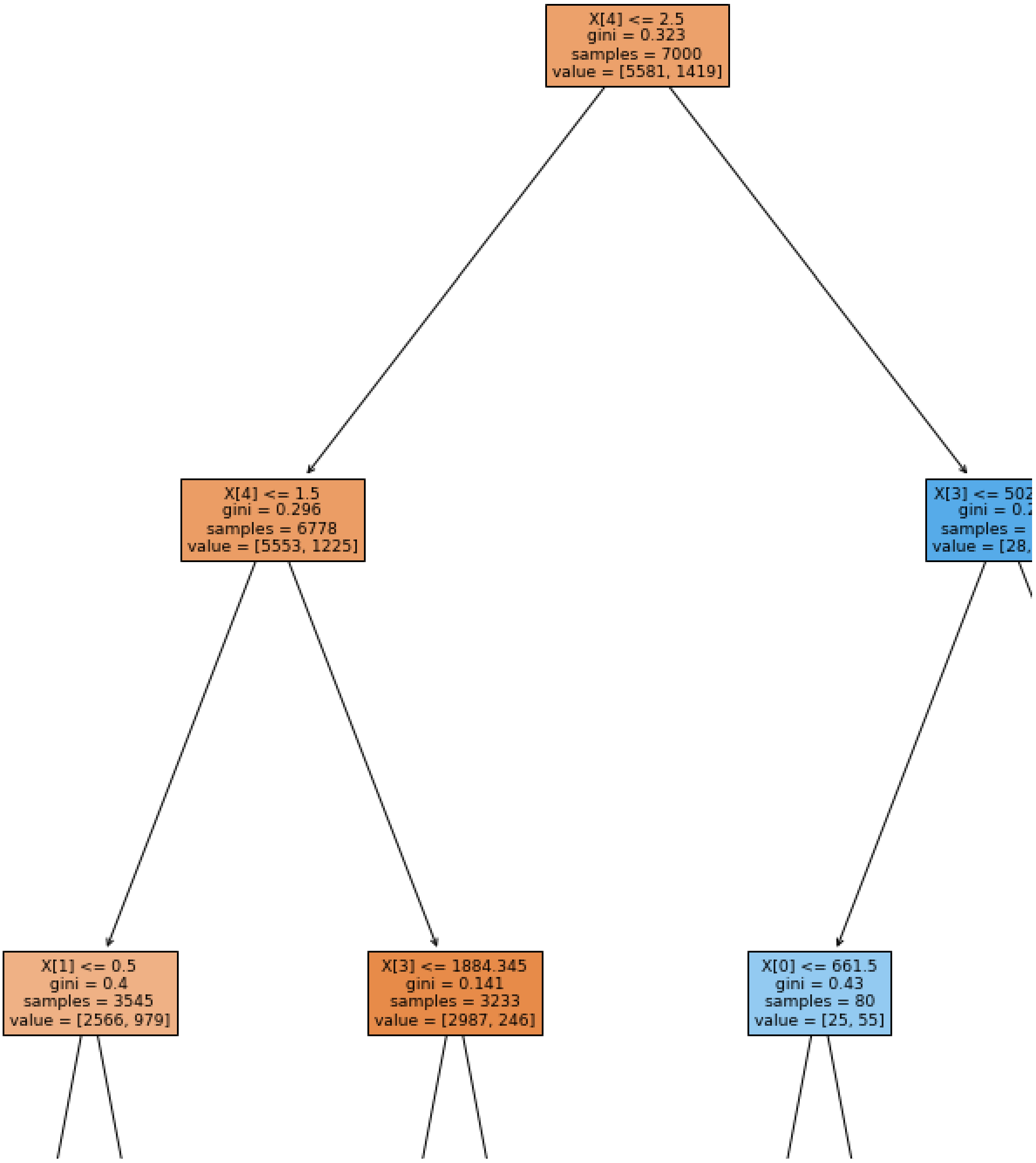
```
treemodel.fit(x_train,y_train)
```

```
DecisionTreeClassifier(max_depth=3)
```

```
from sklearn import tree
```

```
plt.figure(figsize=(15,20))
tree.plot_tree(treemodel, filled=True)
```

```
[Text(0.5, 0.875, 'X[4] <= 2.5\ngini = 0.323\nsamples = 7000\nvalue = [5581, 1
 Text(0.25, 0.625, 'X[4] <= 1.5\ngini = 0.296\nsamples = 6778\nvalue = [5553,
 Text(0.125, 0.375, 'X[1] <= 0.5\ngini = 0.4\nsamples = 3545\nvalue = [2566, 9
 Text(0.0625, 0.125, 'gini = 0.349\nsamples = 1763\nvalue = [1366, 397]'),
 Text(0.1875, 0.125, 'gini = 0.44\nsamples = 1782\nvalue = [1200, 582]'),
 Text(0.375, 0.375, 'X[3] <= 1884.345\ngini = 0.141\nsamples = 3233\nvalue = [
 Text(0.3125, 0.125, 'gini = 0.064\nsamples = 1831\nvalue = [1770, 61]'),
 Text(0.4375, 0.125, 'gini = 0.229\nsamples = 1402\nvalue = [1217, 185]'),
 Text(0.75, 0.625, 'X[3] <= 50210.0\ngini = 0.22\nsamples = 222\nvalue = [28,
 Text(0.625, 0.375, 'X[0] <= 661.5\ngini = 0.43\nsamples = 80\nvalue = [25, 55
 Text(0.5625, 0.125, 'gini = 0.303\nsamples = 43\nvalue = [8, 35]'),
 Text(0.6875, 0.125, 'gini = 0.497\nsamples = 37\nvalue = [17, 20]'),
 Text(0.875, 0.375, 'X[3] <= 140980.32\ngini = 0.041\nsamples = 142\nvalue = [
 Text(0.8125, 0.125, 'gini = 0.017\nsamples = 114\nvalue = [1, 113]'),
 Text(0.9375, 0.125, 'gini = 0.133\nsamples = 28\nvalue = [2, 26]')]
```

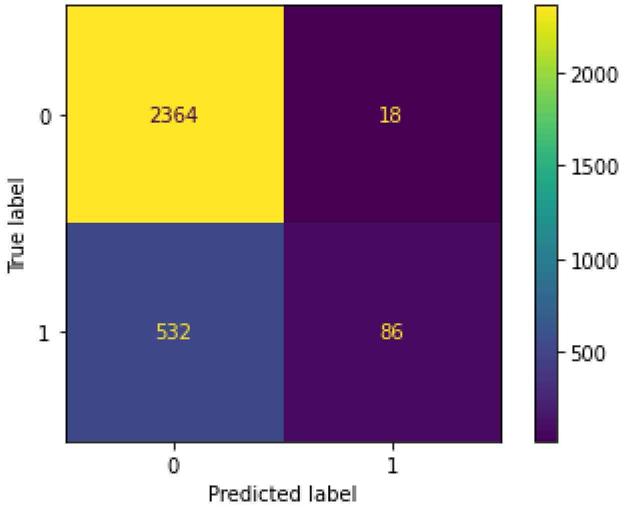


```
                  /    \                      /    \                      /    \

y_pred = treemodel.predict(x_test)
                 /       \                     /       \                     /       \

from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
                /         \                     /         \                     /         \

ac=accuracy_score(y_pred, y_test)
print(ac)
```

```
    0.8166666666666667
```

```
confusion_matrix(y_pred, y_test)
plot_confusion_matrix(treemodel, x_test, y_test)
```

```
    /usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: Future
      warnings.warn(msg, category=FutureWarning)
    <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7f0a64218a
```



```
report=classification_report(y_pred, y_test)
print(report)
```

```
                  precision    recall  f1-score   support

               0       0.99      0.82      0.90      2896
               1       0.14      0.83      0.24       104

        accuracy                           0.82      3000
       macro avg       0.57      0.82      0.57      3000
    weighted avg       0.96      0.82      0.87      3000
```

✓ 0s completed at 3:33 PM ● ×