In [1]:

```
#Decision Tree
import pandas as pd
import matplotlib as plt
from sklearn.preprocessing import LabelEncoder
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:

```
data=pd.read_csv('telecom_users.csv')
```

In [3]:

```
from sklearn.preprocessing import LabelEncoder
encoder=LabelEncoder()
data['PhoneService']=encoder.fit_transform(data['PhoneService'])
data['MultipleLines']=encoder.fit_transform(data['MultipleLines'])
data['InternetService']=encoder.fit_transform(data['InternetService'])
data['OnlineSecurity']=encoder.fit_transform(data['OnlineSecurity'])
data['OnlineBackup']=encoder.fit_transform(data['OnlineBackup'])
data['DeviceProtection']=encoder.fit_transform(data['DeviceProtection'])
data['InternetService']=encoder.fit_transform(data['InternetService'])
data['TechSupport']=encoder.fit_transform(data['TechSupport'])
data['StreamingTV']=encoder.fit_transform(data['StreamingTV'])
data['StreamingMovies']=encoder.fit_transform(data['StreamingMovies'])
data['PaymentMethod']=encoder.fit_transform(data['PaymentMethod'])
data['Churn']=encoder.fit_transform(data['Churn'])
data['gender']=encoder.fit_transform(data['gender'])
data['Partner']=encoder.fit_transform(data['Partner'])
data['TotalCharges']=encoder.fit_transform(data['TotalCharges'])
data.head()
```

Out[3]:

	Unnamed: 0	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	M
0	1869	7010-BRBUU	1	0	1	Yes	72	1	
1	4528	9688-YGXVR	0	0	0	No	44	1	
2	6344	9286-DOJGF	0	1	1	No	38	1	
3	6739	6994-KERXL	1	0	0	No	4	1	
4	432	2181-UAESM	1	0	0	No	2	1	

5 rows × 22 columns

In [4]:

```
#Decision Tree Prediction
#importing relevant libraries
from sklearn.model_selection import train_test_split
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
#features extraction
x = data[['tenure','PhoneService','MultipleLines','InternetService','OnlineSecurity','Onlin
y = data['Churn']
#splitting data
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size=0.20,random_state=10)

#applying tree algorithm
tree = tree.DecisionTreeClassifier()
tree.fit(x_train, y_train)   #fitting our model
y_pred=tree.predict(x_test)   # evaluating our model
print("score:{}".format(accuracy_score(y_test, y_pred)))
```

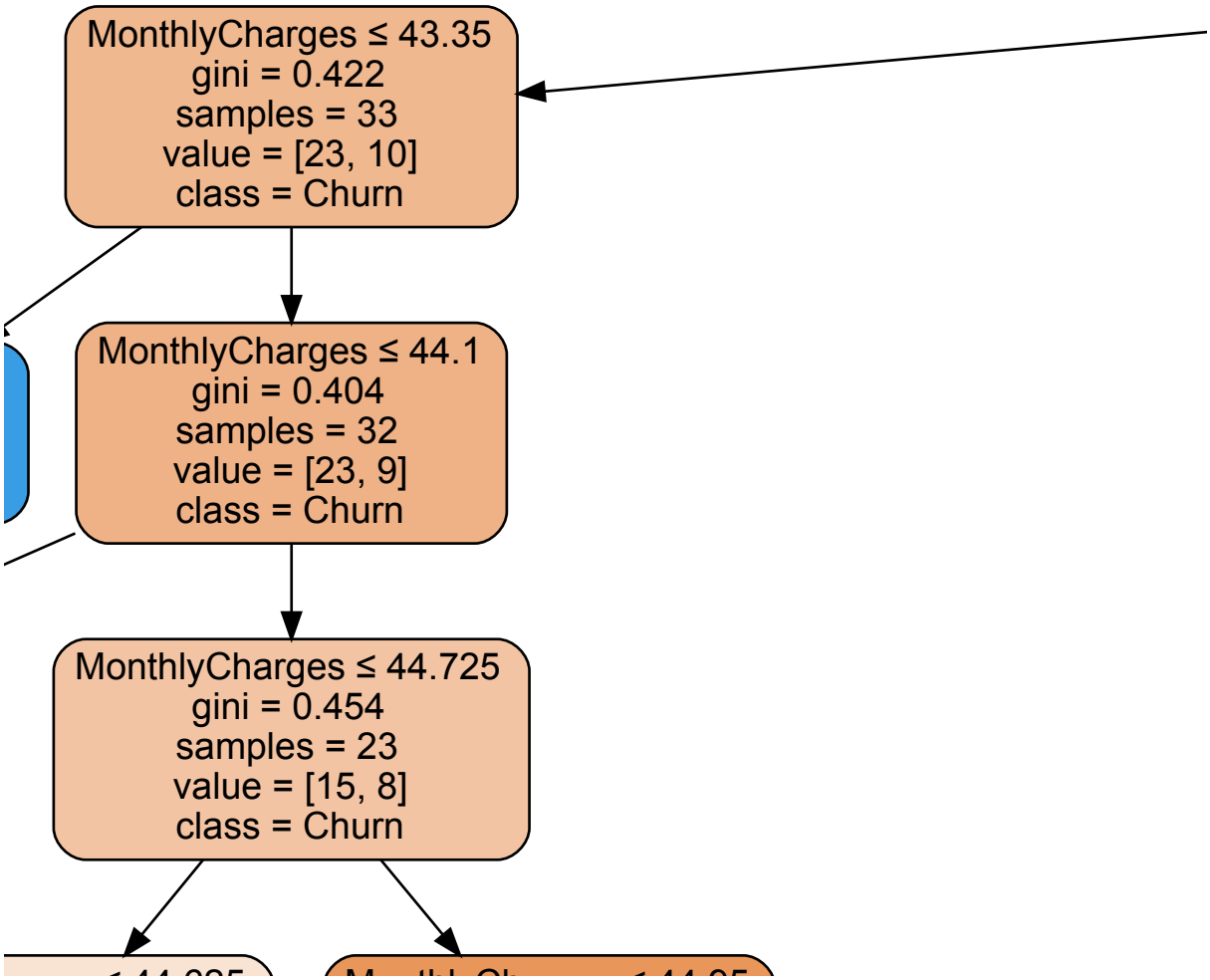
```
score:0.7178631051752922
```

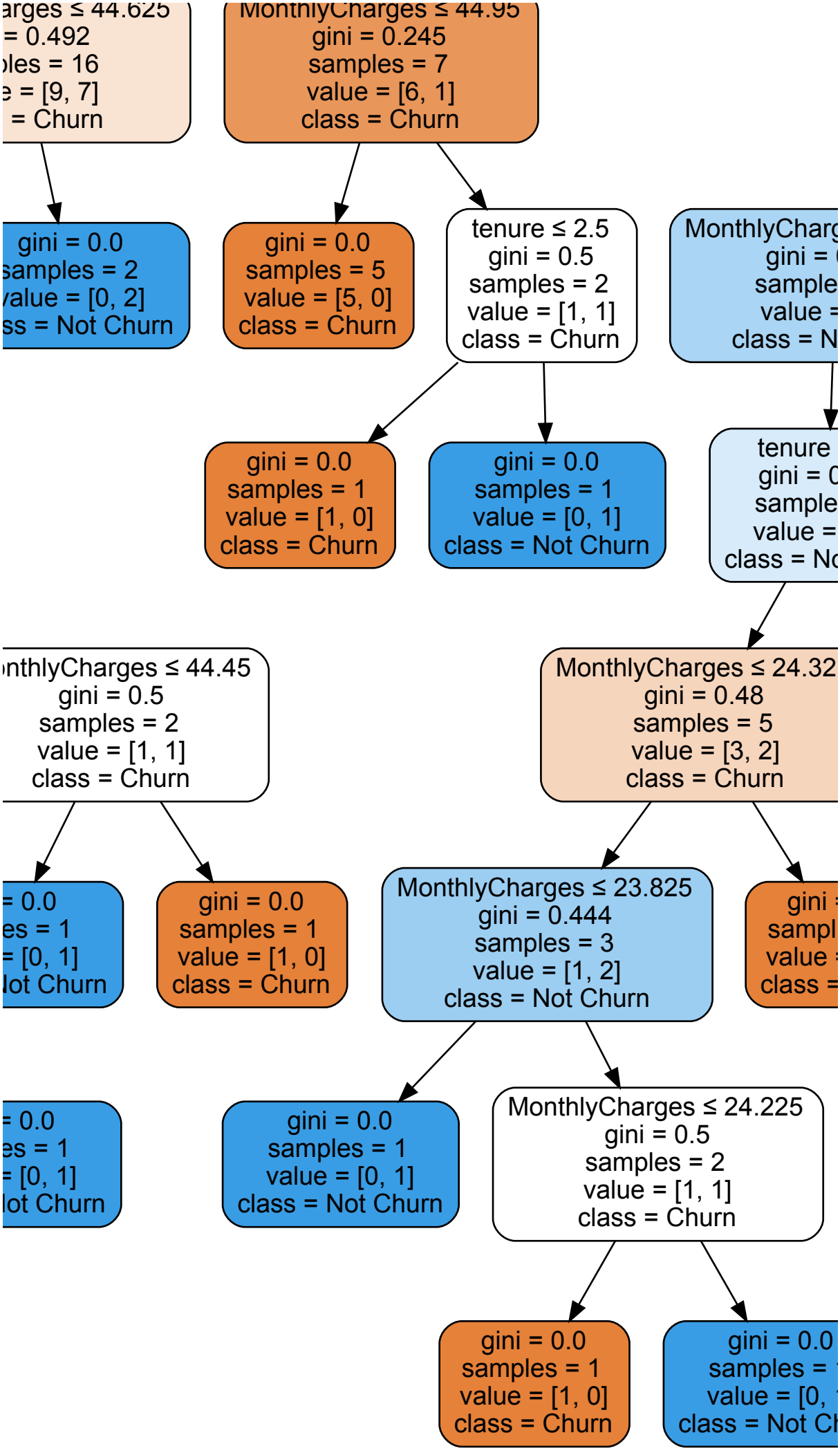
In [6]:

```
#Decision Tree Prediction
#importing relevant libraries
from sklearn.model_selection import train_test_split
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
#features extraction
x = data[['tenure','PhoneService','MultipleLines','InternetService','OnlineSecurity','Onlin
y = data['Churn']
#splitting data
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size=0.20,random_state=10)

#applying tree algorithm
clf = tree.DecisionTreeClassifier()
clf.fit(x_train, y_train)   #fitting our model
import graphviz
dot_data = tree.export_graphviz(clf, out_file=None,
                         feature_names=x.columns.values,
                         class_names=['Churn','Not Churn'],
                         filled=True, rounded=True,
                         special_characters=True)
graph = graphviz.Source(dot_data)
graph
```

Out[6]:


MonthlyCharges ≤ 43.35
gini = 0.422
samples = 33
value = [23, 10]
class = Churn
MonthlyCharges ≤ 44.1
gini = 0.404
samples = 32
value = [23, 9]
class = Churn
MonthlyCharges ≤ 44.725
gini = 0.454
samples = 23
value = [15, 8]
class = Churn


arges ≤ 44.625
= 0.492
les = 16
e = [9, 7]
= Churn
MonthlyCharges ≤ 44.95
gini = 0.245
samples = 7
value = [6, 1]
class = Churn
gini = 0.0
samples = 2
value = [0, 2]
ss = Not Churn
gini = 0.0
samples = 5
value = [5, 0]
class = Churn
tenure ≤ 2.5
gini = 0.5
samples = 2
value = [1, 1]
class = Churn
MonthlyCharg
gini =
sample
value =
class = N
gini = 0.0
samples = 1
value = [1, 0]
class = Churn
gini = 0.0
samples = 1
value = [0, 1]
class = Not Churn
tenure
gini = 0
sample
value =
class = No
nthlyCharges ≤ 44.45
gini = 0.5
samples = 2
value = [1, 1]
class = Churn
MonthlyCharges ≤ 24.32
gini = 0.48
samples = 5
value = [3, 2]
class = Churn
= 0.0
es = 1
= [0, 1]
lot Churn
gini = 0.0
samples = 1
value = [1, 0]
class = Churn
MonthlyCharges ≤ 23.825
gini = 0.444
samples = 3
value = [1, 2]
class = Not Churn
gini
sampl
value
class
= 0.0
es = 1
= [0, 1]
lot Churn
gini = 0.0
samples = 1
value = [0, 1]
class = Not Churn
MonthlyCharges ≤ 24.225
gini = 0.5
samples = 2
value = [1, 1]
class = Churn
gini = 0.0
samples = 1
value = [1, 0]
class = Churn
gini = 0.0
samples =
value = [0,
class = Not Ch

In [ ]:

```

```