```
import pandas as pd
df=pd.read_excel('titanic-passengers.xlsx')
df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fa
0	343	No	2	Collander, Mr. Erik Gustaf	male	28.0	0	0	248740	13.00
1	76	No	3	Moen, Mr. Sigurd Hansen	male	25.0	0	0	348123	7.65
2	641	No	3	Jensen, Mr. Hans Peder	male	20.0	0	0	350050	7.85
3	568	No	3	Palsson, Mrs. Nils (Alma Cornelia	female	29.0	0	4	349909	21.07

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

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

```
df['Embarked'].fillna('S',inplace=True)
df['Embarked'].value_counts()
```

```
S    646
C    168
Q     77
Name: Embarked, dtype: int64
```

```
df['Age'].fillna(df['Age'].mean(),inplace=True)
df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fa
0	343	No	2	Collander, Mr. Erik Gustaf	male	28.0	0	0	248740	13.00
1	76	No	3	Moen, Mr. Sigurd Hansen	male	25.0	0	0	348123	7.65

```
df['Cabin'].value_counts()
```

```
C23 C25 C27    4
B96 B98        4
G6             4
F33            3
D              3
              ..
D37            1
C101           1
E58            1
B38            1
A14            1
Name: Cabin, Length: 147, dtype: int64
```

```
df['Cabin'].fillna('G6',inplace=True)
df['Cabin'].value_counts()
```

```
G6             691
B96 B98          4
C23 C25 C27      4
F33              3
D                3
              ...
D37              1
C101             1
E58              1
B38              1
A14              1
Name: Cabin, Length: 147, dtype: int64
```

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

```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Cabin          0
Embarked       0
dtype: int64
```

```
df['Survived']=df['Survived'].map({"No":0,"Yes":1})
df['Survived'].value_counts()
```

```
0    549
1    342
Name: Survived, dtype: int64
```

```
onehot=pd.get_dummies(df['Sex'])
onehot.head()
```

	female	male
0	0	1
1	0	1
2	0	1
3	1	0
4	0	1

```
new_data=df.drop('Sex',axis=1)
new_data.head()
```

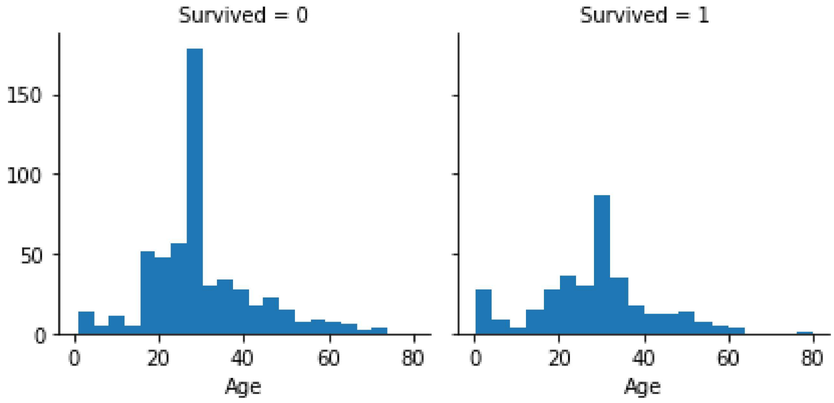
	PassengerId	Survived	Pclass	Name	Age	SibSp	Parch	Ticket	Fare	Cabi
0	343	0	2	Collander, Mr. Erik Gustaf	28.0	0	0	248740	13.0000	G
1	76	0	3	Moen, Mr. Sigurd Hansen	25.0	0	0	348123	7.6500	F G7
2	641	0	3	Jensen, Mr. Hans Peder	20.0	0	0	350050	7.8542	G
				Palsson						

```
new_data=new_data.join(onehot)
new_data.head()
```

	PassengerId	Survived	Pclass	Name	Age	SibSp	Parch	Ticket	Fare	Cabi
0	343	0	2	Collander, Mr. Erik Gustaf	28.0	0	0	248740	13.0000	G
1	76	0	3	Moen, Mr. Sigurd Hansen	25.0	0	0	348123	7.6500	F G7

```
import seaborn as sns
import matplotlib.pyplot as plt
d = sns.FacetGrid(df, col='Survived')
d.map(plt.hist, 'Age', bins=20)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f8feb035850>
```



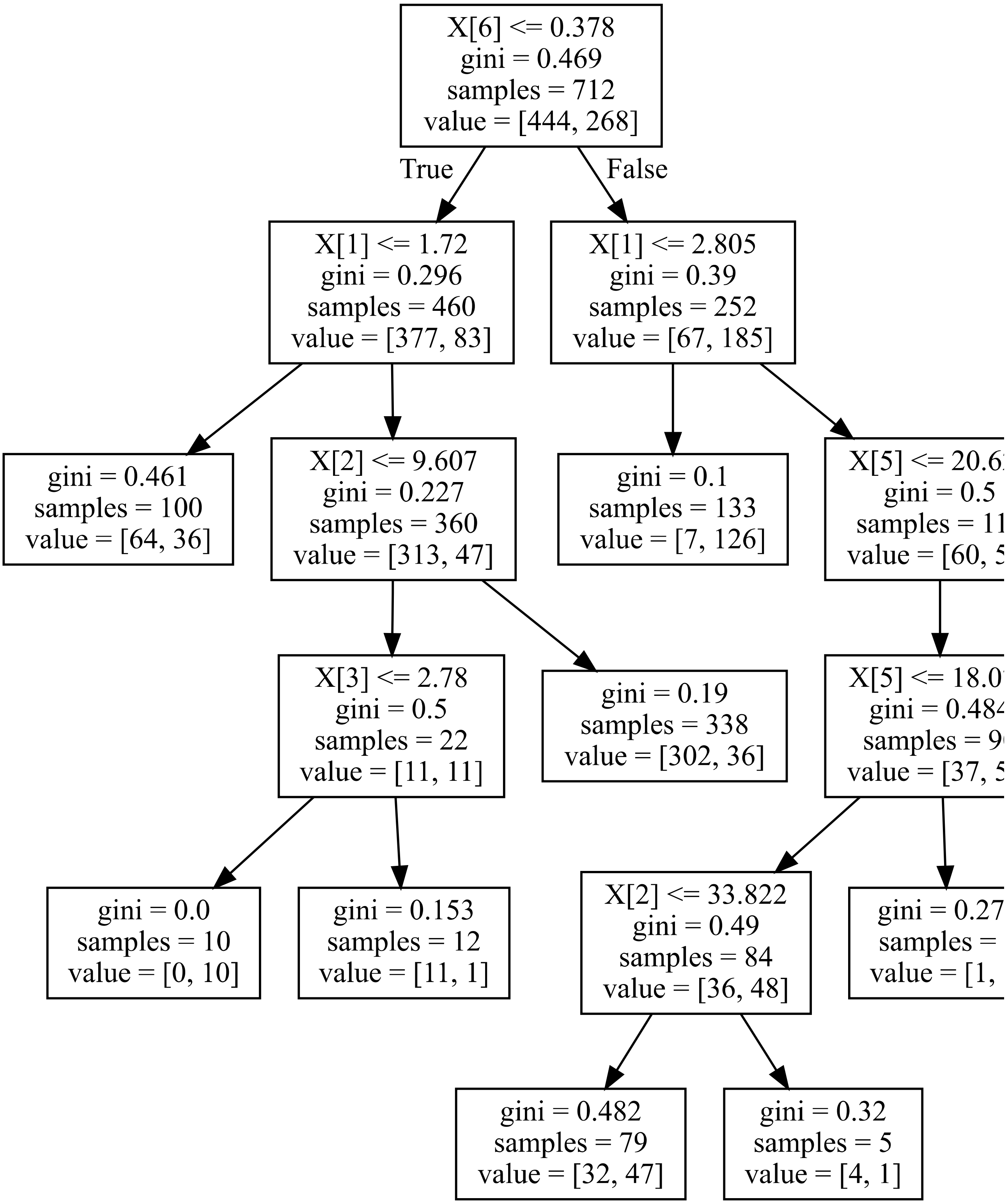
```
from sklearn.model_selection import train_test_split
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
#features extraction
x=new_data.drop(["Survived", "Name", "Cabin", "Ticket", "Embarked"], axis=1)
y= new_data["Survived"]

#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
t = tree.DecisionTreeClassifier(criterion="gini",splitter='random',max_leaf_nodes=10,min_s
t.fit(x_train, y_train)   #fitting our model
y_pred=t.predict(x_test)   # evaluating our model
print("score:{}".format(accuracy_score(y_test, y_pred)))
```

```
score:0.8268156424581006
```

```
import graphviz
from sklearn.tree import export_graphviz
dot_data=tree.export_graphviz(t,out_file=None)
graph=graphviz.Source(dot_data)
graph.render('data')
graph
```



```
from sklearn.ensemble import RandomForestClassifier
from sklearn import metrics
clf=RandomForestClassifier(n_estimators=10)
clf.fit(x_train, y_train)
y_pred=clf.predict(x_test)
print("Accuracy:", metrics.accuracy_score(y_test, y_pred))
```

```
    Accuracy: 0.8324022346368715
```

✓ 0 s terminée à 18:11 ● ✕