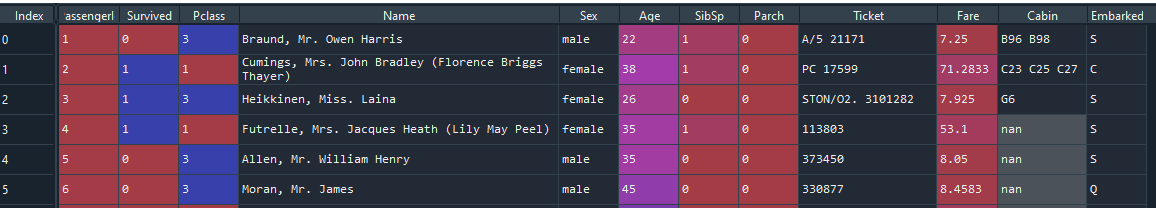
import pandas as pd

import numpy as np

from sklearn import tree

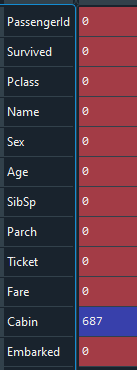
from sklearn import preprocessing

titanic\_train = pd.read\_csv("D:\\ML\\dataset\\train.csv")



**## to chack the null values**

titanic\_train\_null = titanic\_train.isnull().sum()



titanic\_train['Cabin']=titanic\_train['Cabin'].mode()

**####### Mode value to replace cabin columns ## this column is unwanted.**

lable\_encoder = preprocessing.LabelEncoder()

encoded\_sex = lable\_encoder.fit\_transform(titanic\_train['Sex'])

**####### To change text file to number**

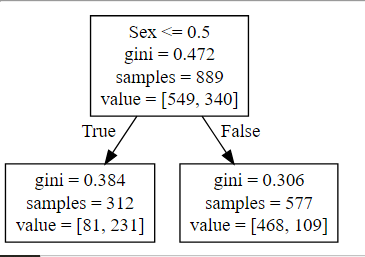
**To Model Creation**

tree\_model = tree.DecisionTreeClassifier()

tree\_model.fit(X=pd.DataFrame(encoded\_sex),y=pd.DataFrame(titanic\_train['Survived']))

with open("dtree.dot",'w') as f:

f=tree.export\_graphviz(tree\_model,feature\_names=['Sex'],out\_file=f)



**Add more independent variable**

predictors = pd.DataFrame([encoded\_sex,titanic\_train['Age'],titanic\_train['Pclass'],titanic\_train['Fare']]).T

###### To add the more independent variable use T

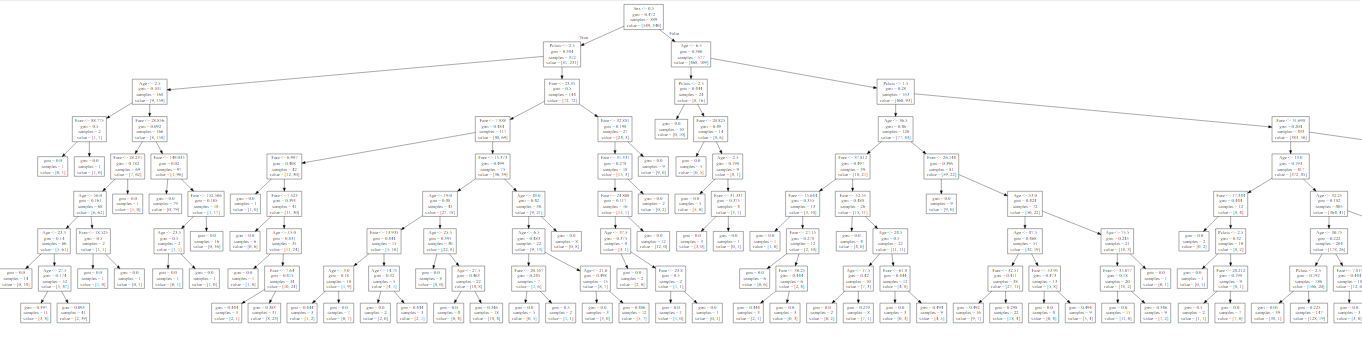
tree\_model = tree.DecisionTreeClassifier(max\_depth=8)

###### max\_depth = 8 that is idv = 4 and dv values =2 so 4X2 = 8

tree\_model.fit(X=predictors,y=titanic\_train['Survived'])

with open("Dtree1.dot",'w') as f:

f = tree.export\_graphviz(tree\_model,feature\_names=["Sex","Age","Pclass","Fare"],out\_file=f)



**Model Accuracy**

tree\_model.score(X=predictors,y=titanic\_train['Survived'])



**Decision Tree helpful for predication**

titanic\_test = pd.read\_csv("D:\\ML\\dataset\\test.csv")

titanic\_test\_null = titanic\_test.isnull().sum()

encode\_sex\_test = lable\_encoder.fit\_transform(titanic\_test['Sex'])

**###### Label encoder to change text column to number columns**

test\_features = pd.DataFrame([encode\_sex\_test,titanic\_test['Pclass'],titanic\_test['Age'],titanic\_test['Fare']]).T

#test\_preds = tree\_model.predict(X=pd.DataFrame([encode\_sex\_test,titanic\_test['Pclass'],titanic\_test['Age'],titanic\_test['Fare']]))

#predicted\_output = pd.DataFrame({"Passenger\_ID":titanic\_test["PassengerId"],"Survive":test\_preds})

# predicted\_output.to\_csv("output.csv",index=False)

**RandomForestClassifer**

from sklearn.ensemble import RandomForestClassifier

lable\_encoder = preprocessing.LabelEncoder()

titanic\_train['Sex'] = lable\_encoder.fit\_transform(titanic\_train["Sex"])

titanic\_train['Embarked'] = lable\_encoder.fit\_transform(titanic\_train["Embarked"])

rf\_model = RandomForestClassifier(n\_estimators=1000,max\_features=2,oob\_score=True)

**###### n\_estimatore = 1000 run iteration, max\_features = only two split,oob\_score = model accuracy**

features = ["Sex","Pclass","SibSp","Age","Fare","Embarked"]

rf\_model.fit(X = titanic\_train[features],y=titanic\_train['Survived'])

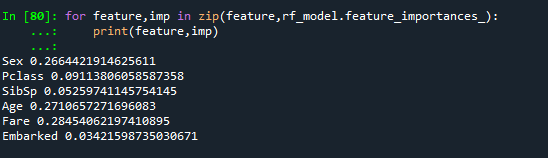
print("OOB Accuracy:",rf\_model.oob\_score\_)



**##### To Find important Variable**

for feature,imp in zip(feature,rf\_model.feature\_importances\_):

print(feature,imp)



**###### import features are “Sex”,”Age”,”Embarked” it contain higher value’s.**

**Build the Decision Tree**

tree\_model = tree.DecisionTreeClassifier()

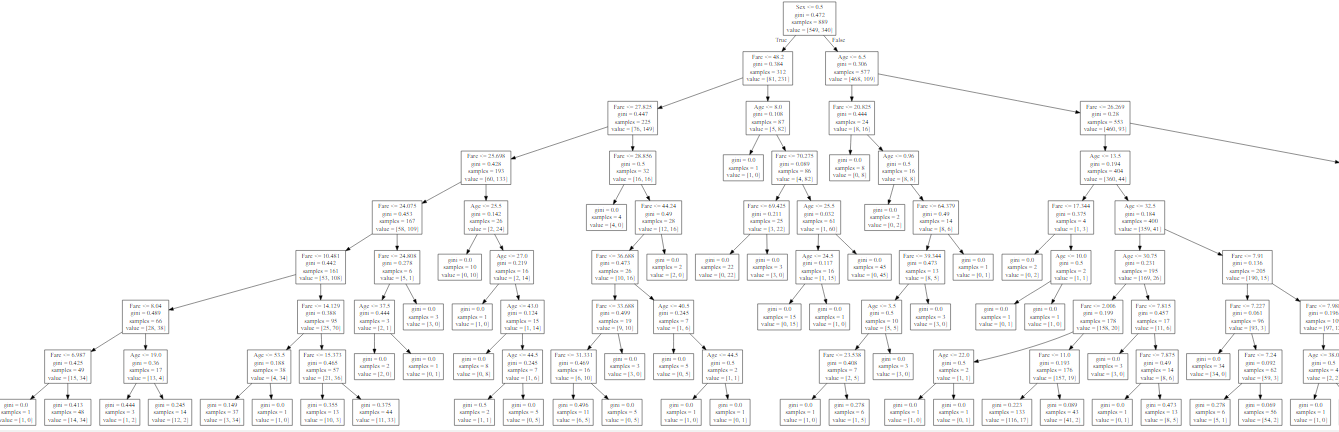
predictors = pd.DataFrame([encoded\_sex,titanic\_train['Age'],titanic\_train['Fare']]).T

tree\_model = tree.DecisionTreeClassifier(max\_depth=8)

tree\_model.fit(X=predictors,y=titanic\_train['Survived'])

with open("Dtree3.dot",'w') as f:

f = tree.export\_graphviz(tree\_model,feature\_names=["Sex","Age","Fare"],out\_file=f)



**###### Model Accuracy**

tree\_model.score(X=predictors,y=titanic\_train['Survived'])



This model is 87% accuracy, so that