

**0**

1

0

3

Braund, Mr. Owen Harris

male

22.0

1

0

A/5 21171

7.2500

NaN

S

**2**

3

1

3

Heikkinen, Miss. Laina

female

26.0

0

0

STON/O2.

3101282

7.9250

NaN

S

**4**

5

0

3

Allen, Mr. William Henry

male

35.0

0

0

373450

8.0500

NaN

S

**6**

7

0

1

McCarthy, Mr. Timothy J

male

54.0

0

0

17463

51.8625

E46

S

**8**

9

1

3

Johnson, Mrs. Oscar W

(Elisabeth Vilhelmina Berg)

female

27.0

0

2

347742

11.1333

NaN

S

**Assignment**

In [1]:

**import pandas as pd**

**import numpy as np**

In [2]:

s=pd.read\_csv("train.csv" )

In [3]:

s.head(10 )

1. ut[3]:
2. **assengerId Survived Pclass Name Sex Age SibSp Parch Ticket Fare Cabin Embarked**

**1** 2 1 1

**3** 4 1 1

Cumings, Mrs. John Bradley

(Florence Briggs Th...

Futrelle, Mrs. Jacques Heath

(Lily May Peel)

female 38.0 1 0 PC 17599 71.2833 C85 C

female 35.0 1 0 113803 53.1000 C123 S

**5** 6 0 3 Moran, Mr. James male 45.0 0 0 330877 8.4583 NaN Q

**7** 8 0 3 Palsson, Master. Gosta Leonard male 2.0 3 1 349909 21.0750 NaN S

**9** 10 1 2

Nasser, Mrs. Nicholas (Adele

Achem)

female 14.0 1 0 237736 30.0708 NaN C

**Checking for null values**

In [4]:

s.isnull() .sum()

1. ut[4]:
2. assengerId 0

Survived 0

Pclass 0

Name 0

Sex 0

Age 0

SibSp 0

Parch 0

Ticket 0

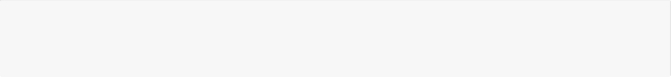
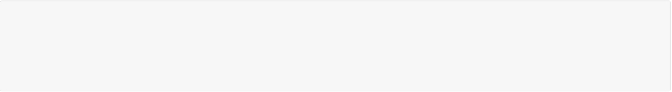
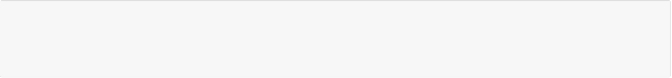
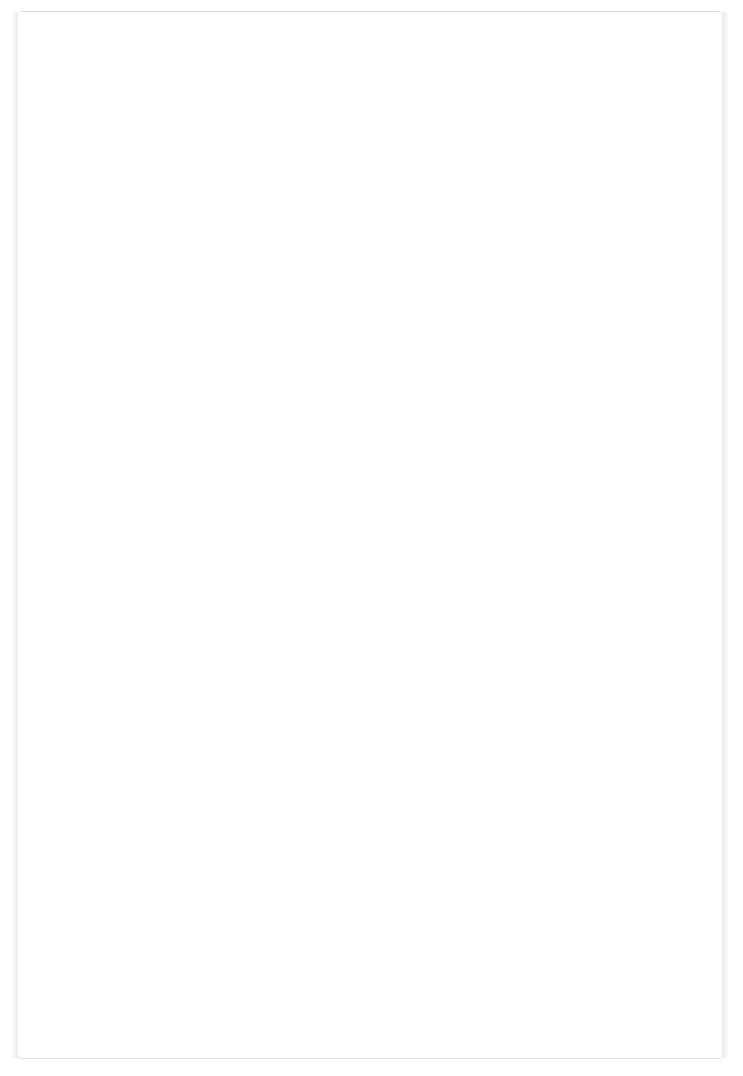
Fare 0

Cabin 687

Embarked 0

dtype: int64

**converting string categorical values into numeric values** In [5]:



**0**

1

0

3

Braund, Mr. Owen Harris

1

22

1

0

A/5 21171

7

NaN

S

**2**

3

1

3

Heikkinen, Miss. Laina

0

26

0

0

STON/O2.

3101282

7

NaN

S

**4**

5

0

3

Allen, Mr. William Henry

1

35

0

0

373450

8

NaN

S

**from sklearn import** preprocessing

l\_encoder =preprocessing.LabelEncoder()

encoded\_g =l\_encoder .fit\_transform(s['Sex'])

s['Sex']=encoded\_g

**convert float value into numeric value**

In [19]:

df=pd.DataFrame(s)

df.Age = df.Age.astype(int)

df1=pd.DataFrame(s)

df1.Fare = df1.Fare .astype(int)

s.head()

1. ut[19]:
2. **assengerId Survived Pclass Name Sex Age SibSp Parch Ticket Fare Cabin Embarked**

**1** 2 1 1

**3** 4 1 1

Cumings, Mrs. John Bradley

(Florence Briggs Th...

Futrelle, Mrs. Jacques Heath (Lily

May Peel)

0 38 1 0 PC 17599 71 C85 C

0 35 1 0 113803 53 C123 S

**implementing Decision Tree**

In [20]:

**from sklearn import** tree

predictors=pd.DataFrame([s['Age'],s['Sex'],s['Fare']]).T

tree\_model=tree .DecisionTreeClassifier(max\_depth=6)

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

Out[20]:

DecisionTreeClassifier(ccp\_alpha=0.0, class\_weight=None, criterion='gini',

max\_depth=6, max\_features=None, max\_leaf\_nodes=None,

min\_impurity\_decrease=0.0, min\_impurity\_split=None,

min\_samples\_leaf=1, min\_samples\_split=2,

min\_weight\_fraction\_leaf=0.0, presort='deprecated',

random\_state=None, splitter='best')

**ln[20]:**

**with** open ("DT\_Attrition" ,'w') **as** f:

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

In [21]:

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

Out[21]:

0.829021372328459

**The accuracy of the model is approximately 83%**

**Testing the dataset**

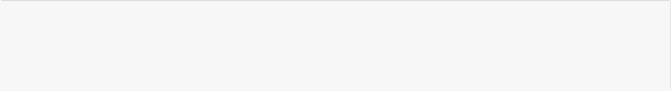
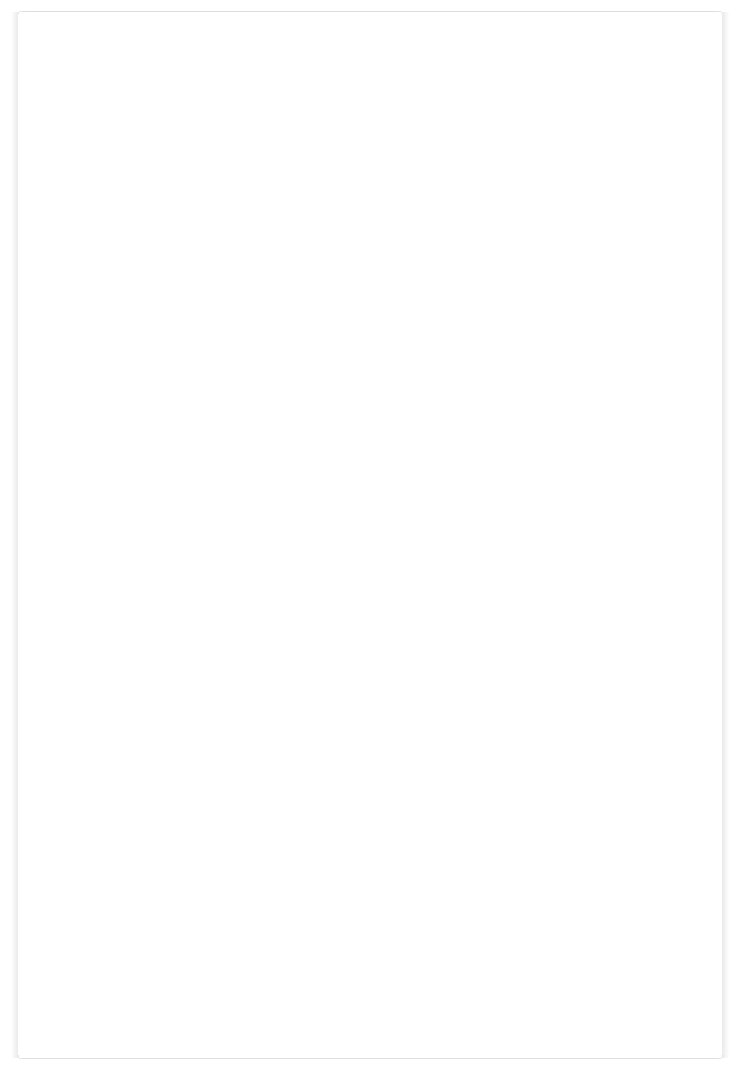
In [22]:

st=pd.read\_csv('test.csv')

new\_age=np.where(st['Age'].isnull(),28,st['Age'])

st["Age"]=new\_age

encoded\_gen =l\_encoder .fit\_transform(st[ 'Sex'])



**0**

892

3

Kelly, Mr. James

1

34

0

0

330911

7

Q

**2**

894

2

Myles, Mr. Thomas Francis

1

62

0

0

240276

9

Q

**4**

896

3

Hirvonen, Mrs. Alexander (Helga E

Lindqvist)

0

22

1

1

3101298

12

S

**0**

892

0

**2**

894

0

**4**

896

1

**412**

1305

0

**414**

1307

0

**416**

1309

0

encoded\_gen =l\_encoder .fit\_transform(st[ 'Sex'])

st['Sex']=encoded\_gen

In [23]:

df=pd.DataFrame(st)

df.Age = df.Age.astype(int)

df1=pd.DataFrame(st)

df1.Fare = df1.Fare .astype(int)

st.head()

1. ut[23]:
2. **assengerId Pclass Name Sex Age SibSp Parch Ticket Fare Embarked**

**1** 893 3 Wilkes, Mrs. James (Ellen Needs) 0 47 1 0 363272 7 S

**3** 895 3 Wirz, Mr. Albert 1 27 0 0 315154 8 S

In [26]:

test\_features=pd.DataFrame([st["Age"],st['Sex'],st['Fare']]).T

In [27]:

test\_preds=tree\_model.predict(X=test\_features)

In [28]:

predicted\_output=pd.DataFrame({ 'PassengerId':st["PassengerId"],'Survived':test\_preds})

In [30]:

predicted\_output

1. ut[30]:
2. **assengerId Survived**

**1** 893 1

**3** 895 0

**...** ... ...

**413** 1306 1

**415** 1308 0

417 rows × 2 columns

In [32]:

predicted\_output.to\_csv('att\_output.csv' )