Problem Statement:

Predicting Survival in the Titanic Data Set We will be using a decision tree to make predictions about the Titanic data set from Kaggle. This data set provides information on the Titanic passengers and can be used to predict whether a passenger survived or not.

**Loading data and modules**

import numpy as np

import pandas as pd

import seaborn as sb

import matplotlib.pyplot as plt

import sklearn

from pandas import Series, DataFrame

from pylab import rcParams

from sklearn import preprocessing

from sklearn.linear\_model import LogisticRegression

from sklearn.cross\_validation import train\_test\_split

from sklearn import metrics

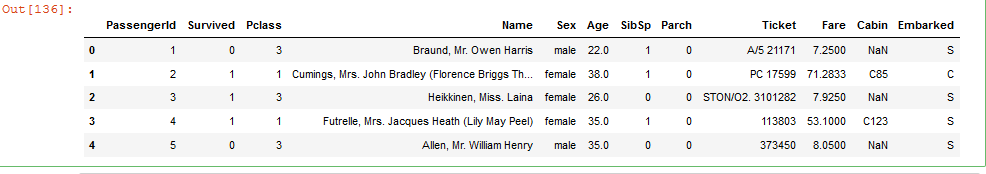
from sklearn.metrics import classification\_report

from sklearn import tree

from sklearn.model\_selection import train\_test\_split

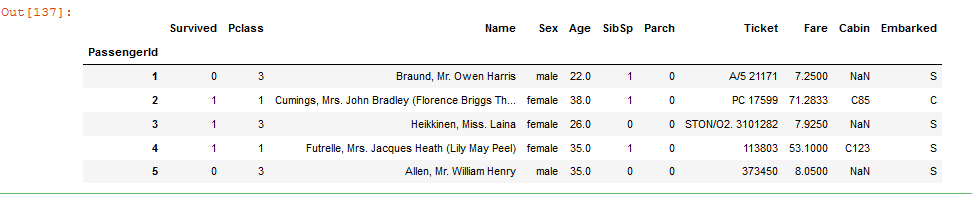
titanic = pd.read\_csv(‘https://raw.githubusercontent.com/BigDataGal/Python-for-Data-Science/master/titanic-train.csv’)

titanic.head()



**###Make PassengerID as first column###**

titanic.set\_index('PassengerId', drop = True, inplace = True)

titanic.head()

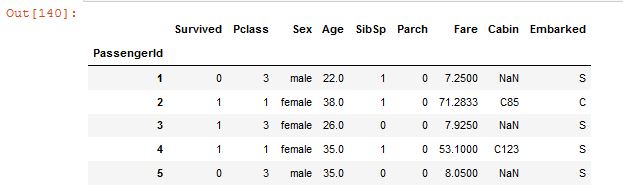
**##Delete Name colum##**

del titanic['Name']

titanic.head()

titanic['Embarked'].dropna(inplace = True)

titanic.head()



**##Delete Cabin column##**

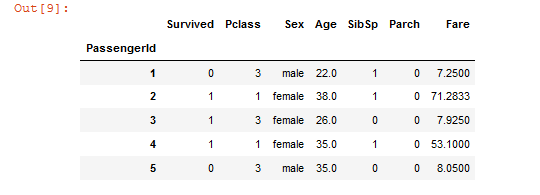
del titanic['Cabin']

titanic.head()

**##Delete Embarked column##**

del titanic['Embarked']

titanic.head()



#numerical value for Sex in the column gender

def gender(st):

if st == 'male':

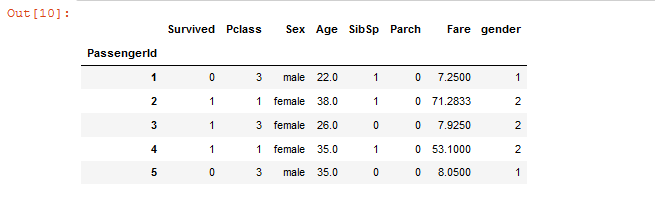
return 1

else:

return 2

titanic['gender'] = titanic.Sex.apply(gender)

titanic.head()



**##Delete Sex Column##**

del titanic['Sex']

**###Class 1 survival###**

class\_1\_surv = titanic[(titanic['Pclass'] == 1)].mean()['Survived']

class\_2\_surv = titanic[(titanic['Pclass'] == 2)].mean()['Survived']

class\_3\_surv = titanic[(titanic['Pclass'] == 3)].mean()['Survived']

class\_1\_surv

op7.PNG

**#graph based on class survival**

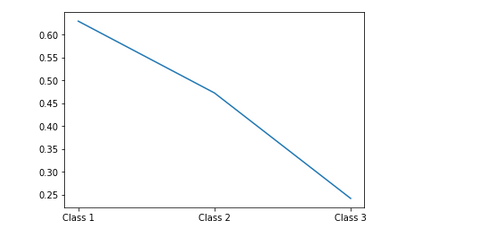
my\_xticks = ['Class 1','Class 2','Class 3']

x = [1,2,3]

plt.xticks(x, my\_xticks)

plt.plot(x, [class\_1\_surv, class\_2\_surv, class\_3\_surv])

plt.show()



males = len(titanic[titanic['gender'] == 1])

females = len(titanic[titanic['gender'] == 2])

# male v/s female pie chart

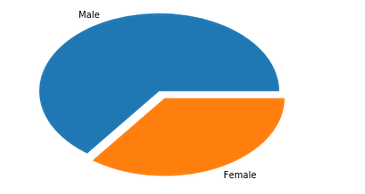
plt.pie([males,females],

labels = ['Male', 'Female'],

explode = [0.10, 0],

startangle = 0)

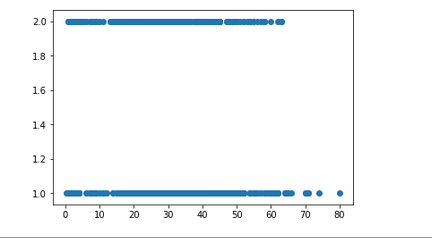
plt.show()



**#scatter graph based on age and gender**

plt.scatter(titanic['Age'], titanic['gender'])

plt.show()



**##Average of survival**

survived = titanic[titanic['Survived'] == 1]

surv\_avg = survived.mean()['Age']

not\_survived = titanic[titanic['Survived'] == 0]

nsurv\_avg = not\_survived.mean()['Age']

def fillavg(survv):

if survv == 1:

return surv\_avg

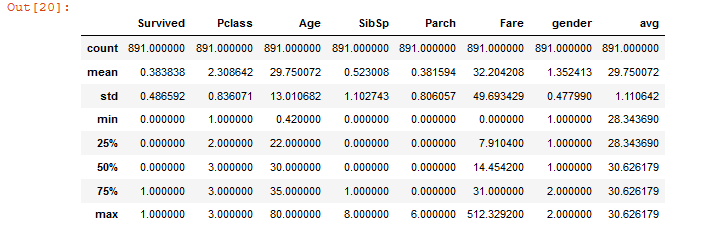
else:

return nsurv\_avg

titanic['avg'] = titanic.Survived.apply(fillavg)

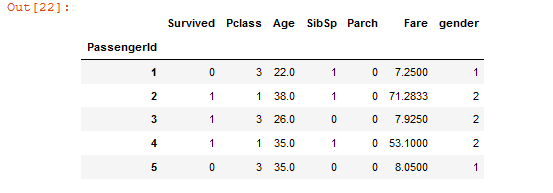
titanic.Age.fillna(titanic['avg'], inplace = True)

titanic.describe()



del titanic['avg']

titanic.head()



x\_train,x\_test,y\_train,y\_test = train\_test\_split(titanic.drop('Survived',axis =1),titanic['Survived'],

test\_size = 0.30,random\_state = 101)

**#Decision Tree**

clf1 = tree.DecisionTreeClassifier()

clf1.fit(x\_train, y\_train)

clf1.score(x\_test, y\_test)

op13.PNG