**Assignment**

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

**Url** = https://raw.githubusercontent.com/BigDataGal/Python-for-Data-

Science/master/titanic-train.csv

titanic = pd.read\_csv(url)

titanic.columns =

['PassengerId','Survived','Pclass','Name','Sex','Age','SibSp','Parch','Ti

cket','Fare','Cabin','E mbarked']

You use only Pclass, Sex, Age, SibSp (Siblings aboard), Parch

(Parents/children aboard), and Fare to predict whether a passenger

survived.

**Task:** Deploy this assignment in any cloud platform.(Try to look for free cloud platform)

**Assignment:** Submit assignment’s deployable link only.

**1. Loading Libraries and Dataset**

**import pandas as pd**

**import numpy as np**

**import seaborn as sns**

**import matplotlib.pyplot as plt**

**from sklearn.tree import export\_graphviz**

***#import pydotplus***

**from IPython.display import Image**

**from sklearn.externals.six import StringIO**

**import graphviz**

**import os**

**from sklearn.impute import SimpleImputer**

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

**from sklearn.linear\_model import LogisticRegression**

**import pandas\_profiling as pp**

**from sklearn.metrics import accuracy\_score,f1\_score,precision\_score,roc\_auc\_score,recall\_score**

**from sklearn.metrics import classification\_report**

**from sklearn.tree import DecisionTreeClassifier**

**url = 'https://raw.githubusercontent.com/BigDataGal/Python-for-Data-Science/master/titanic-train.csv'**

**titanic = pd.read\_csv(url)**

**titanic.head()**

### 2. Exploratory Data Analysis

features **=** ["Pclass","Sex","Age","SibSp","Parch","Fare"]

X **=** titanic[features]

X**.**head()

y **=** titanic["Survived"]

*# Checking for Null Values*

X**.**isnull()**.**sum()

*# Imputing Null values of Age with mean*

X["Age"]**.**fillna(X["Age"]**.**mean(),inplace**=True**)

X**.**isnull()**.**sum()

X**.**info()

X**.**describe()

sns**.**countplot(x**=**titanic["Survived"])**.**set(title **=** 'survived')

sns**.**distplot(X["Age"])**.**set(title **=** "Distribution of Age")

sns**.**barplot(x**=**titanic["Pclass"],y**=**titanic["Survived"])

plt**.**figure(figsize**=**(10,10))

sns**.**heatmap(X**.**corr(),annot**=True**)

X['Sex'] **=** pd**.**get\_dummies(X['Sex'])**.**values

X**.**head()

### 3. Model Building

*# Spliting dataset into train and test*

X\_train,X\_test,y\_train,y\_test **=** train\_test\_split(X,y,test\_size**=**0.3,random\_state**=**42)

*# importing classifier*

**from** sklearn.tree **import** DecisionTreeClassifier

print(f'Shape of train feature:{X\_train**.**shape}')

print(f'Shape of test feature :{X\_test**.**shape}')

print(f'Shape of train target :{y\_train**.**shape}')

print(f'Shape of test target :{y\_test**.**shape}')

**from** sklearn.metrics **import** accuracy\_score,classification\_report,confusion\_matrix

**from** sklearn.model\_selection **import** cross\_val\_score

**from** collections **import** Counter

**import** scikitplot.metrics **as** skplt

*# Instantiating Decision tree model*

model\_dt **=** DecisionTreeClassifier()

*# Fitting on train data*

model\_dt**.**fit(X\_train,y\_train)

*# Predicting on test dataset*

y\_pred **=** model\_dt**.**predict(X\_test)

*# Plotting the confusion matrix*

skplt**.**plot\_confusion\_matrix(y\_test ,y\_pred)

accuracy\_score(y\_test,y\_pred)