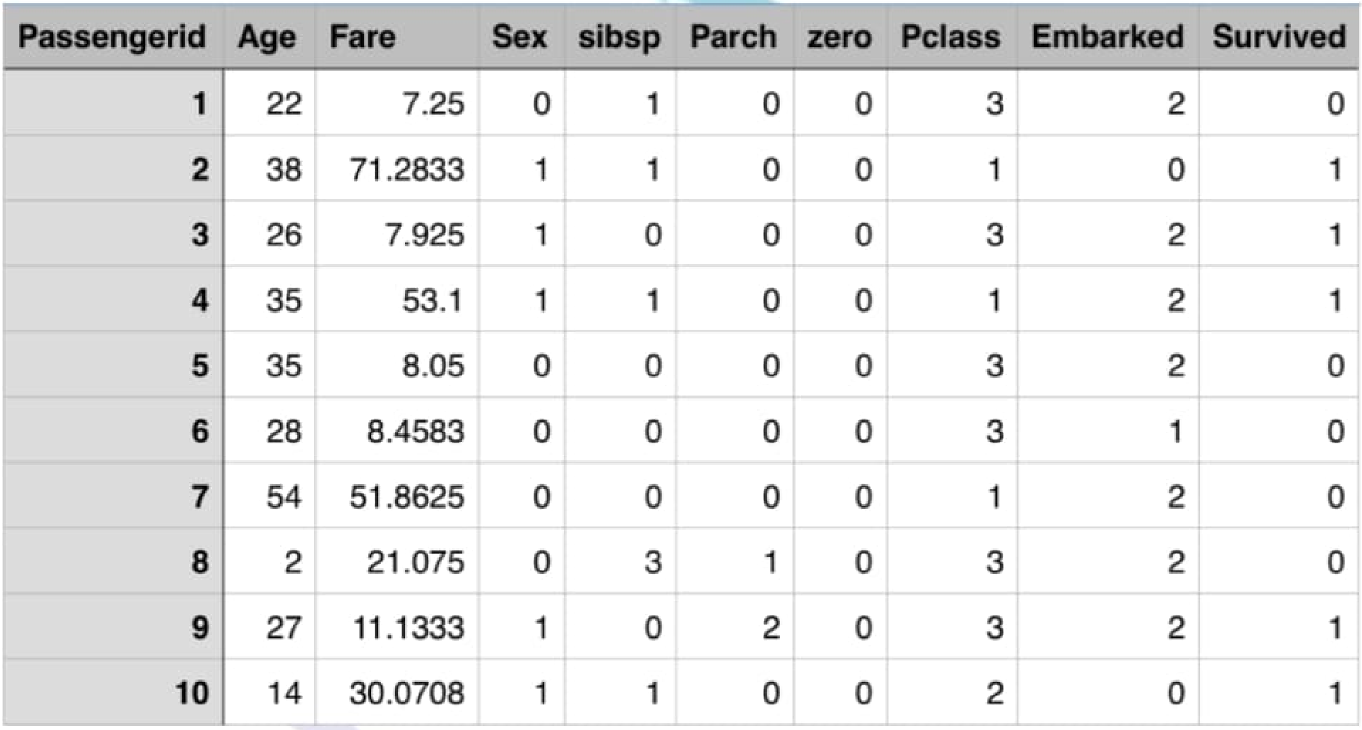
Application 12

Supervised Machine Learning

Logistic Regression

There is one data set which contains information about the passengers from Titanic.
This data set describe multiple features about survived and non survived passengers.

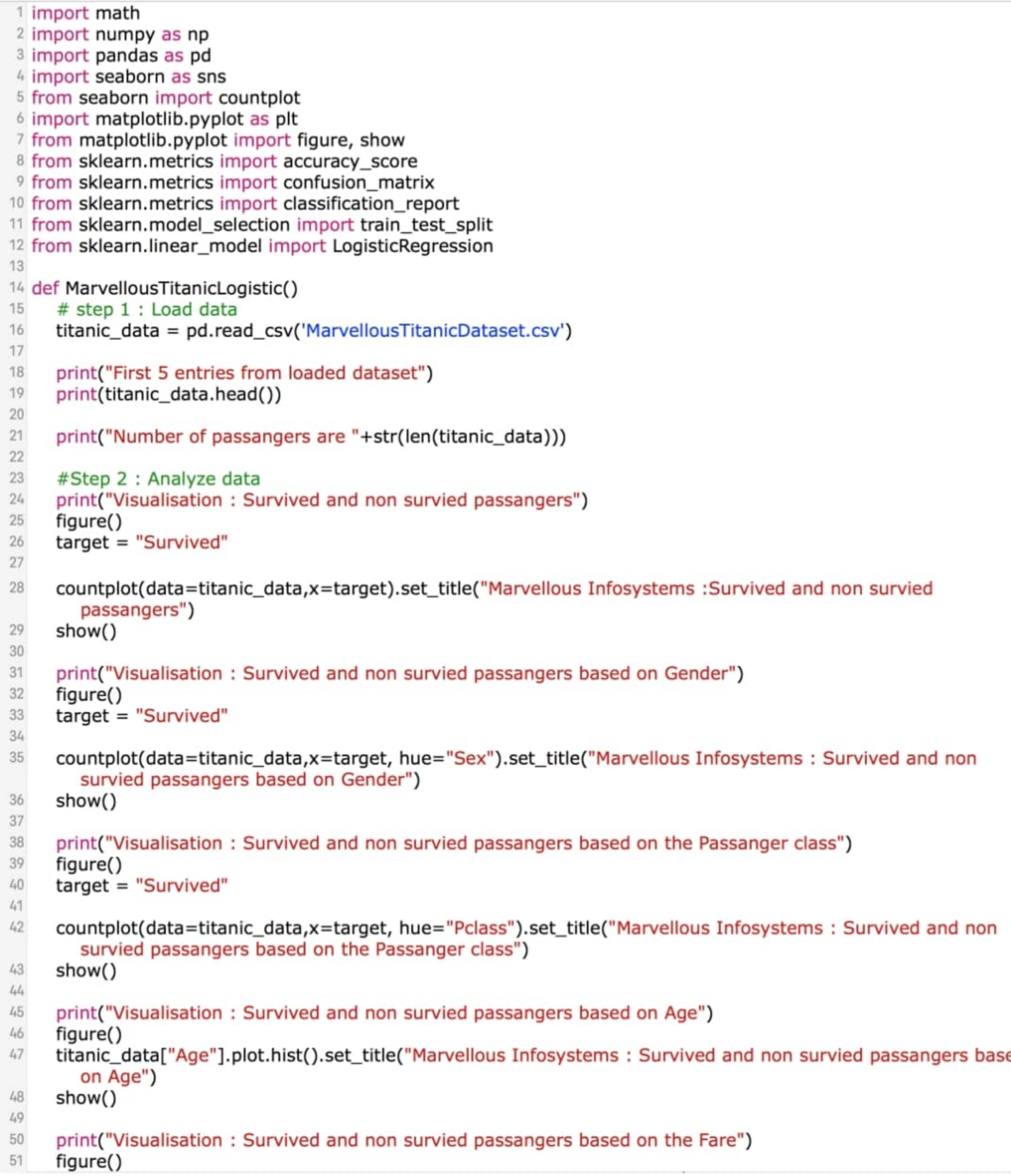


Passengerid	Age	Fare	Sex	sibsp	Parch	zero	Pclass	Embarked	Survived
1	22	7.25	0	1	0	0	3	2	0
2	38	71.2833	1	1	0	0	1	0	1
3	26	7.925	1	0	0	0	3	2	1
4	35	53.1	1	1	0	0	1	2	1
5	35	8.05	0	0	0	0	3	2	0
6	28	8.4583	0	0	0	0	3	1	0
7	54	51.8625	0	0	0	0	1	2	0
8	2	21.075	0	3	1	0	3	2	0
9	27	11.1333	1	0	2	0	3	2	1
10	14	30.0708	1	1	0	0	2	0	1

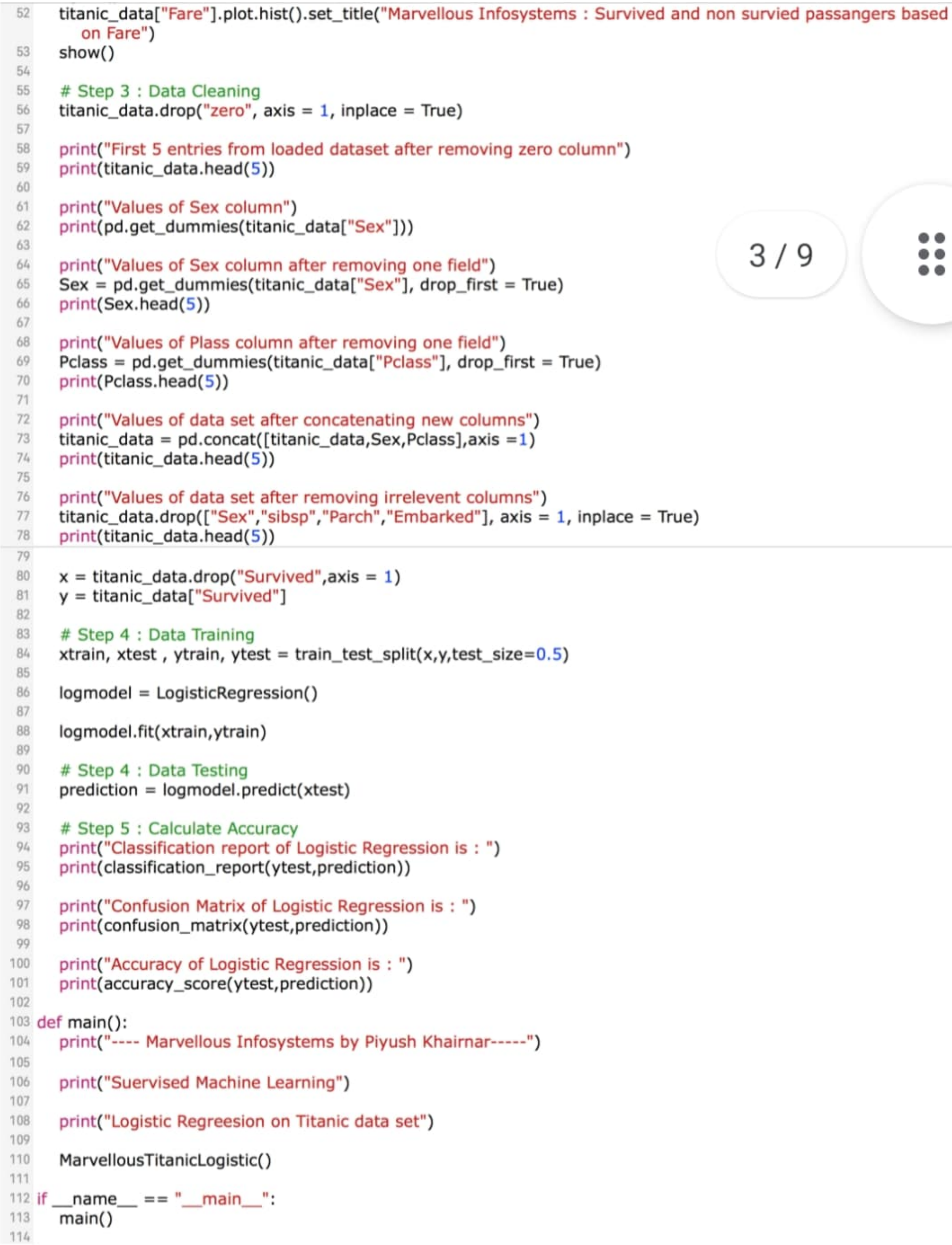
Consider below characteristics of Machine Learning Application :

Classifier :	**Logistic Regression**
DataSet :	**Titanic Dataset**
Features :	**Passenger id,Gender, Age, Fare, Class etc**
Labels :	**-**

Consider below application which uses Logistic Regression algorithm from skit learn library to train above data set and predict whether passenger survived or not.

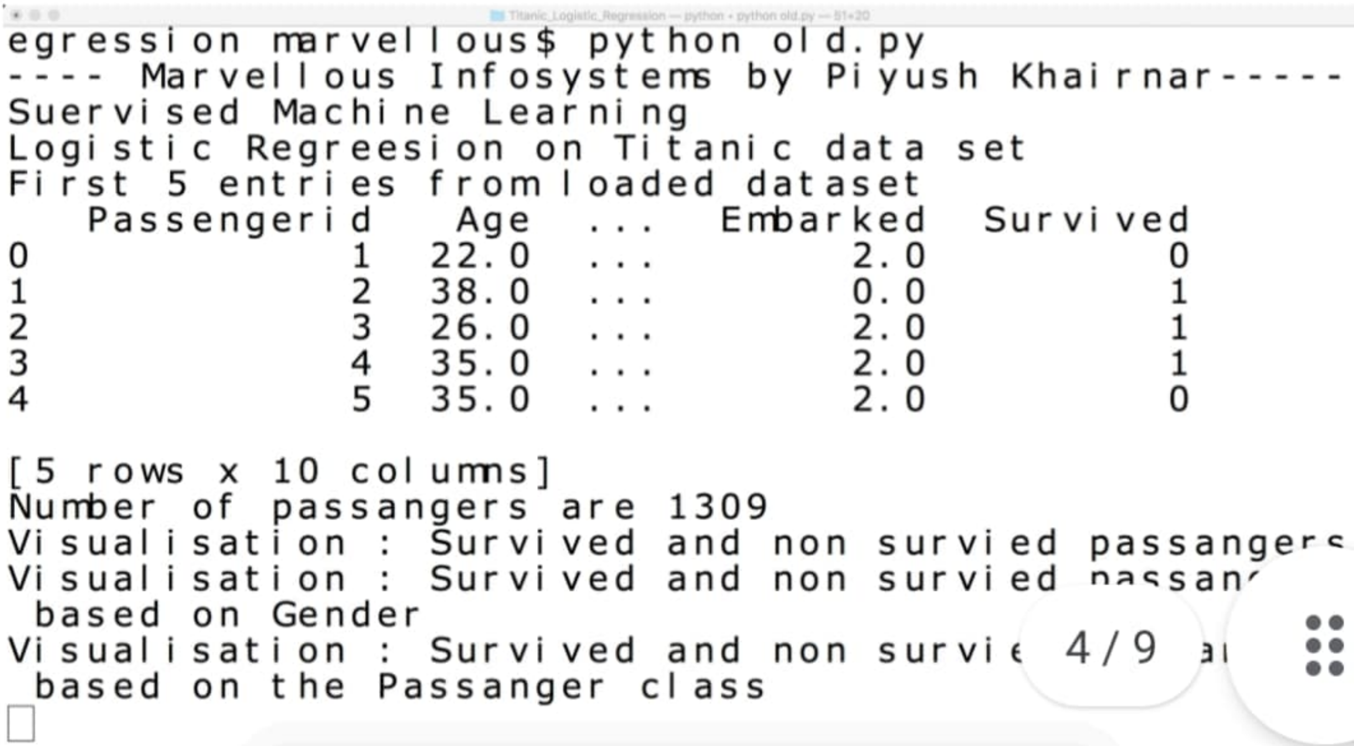


```
import math
import numpy as np
import pandas as pd
import seaborn as sns
from seaborn import countplot
import matplotlib.pyplot as plt
from matplotlib.pyplot import figure, show
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

def MarvellousTitanicLogistic()
    # step 1 : Load data
    titanic_data = pd.read_csv('MarvellousTitanicDataset.csv')

    print("First 5 entries from loaded dataset")
    print(titanic_data.head())

    print("Number of passangers are "+str(len(titanic_data)))

    #Step 2 : Analyze data
    print("Visualisation : Survived and non survied passangers")
    figure()
    target = "Survived"

    countplot(data=titanic_data,x=target).set_title("Marvellous Infosystems :Survived and non survied passangers")
    show()

    print("Visualisation : Survived and non survied passangers based on Gender")
    figure()
    target = "Survived"

    countplot(data=titanic_data,x=target, hue="Sex").set_title("Marvellous Infosystems : Survived and non survied passangers based on Gender")
    show()

    print("Visualisation : Survived and non survied passangers based on the Passanger class")
    figure()
    target = "Survived"

    countplot(data=titanic_data,x=target, hue="Pclass").set_title("Marvellous Infosystems : Survived and non survied passangers based on the Passanger class")
    show()

    print("Visualisation : Survived and non survied passangers based on Age")
    figure()
    titanic_data["Age"].plot.hist().set_title("Marvellous Infosystems : Survived and non survied passangers base on Age")
    show()

    print("Visualisation : Survived and non survied passangers based on the Fare")
    figure()
```

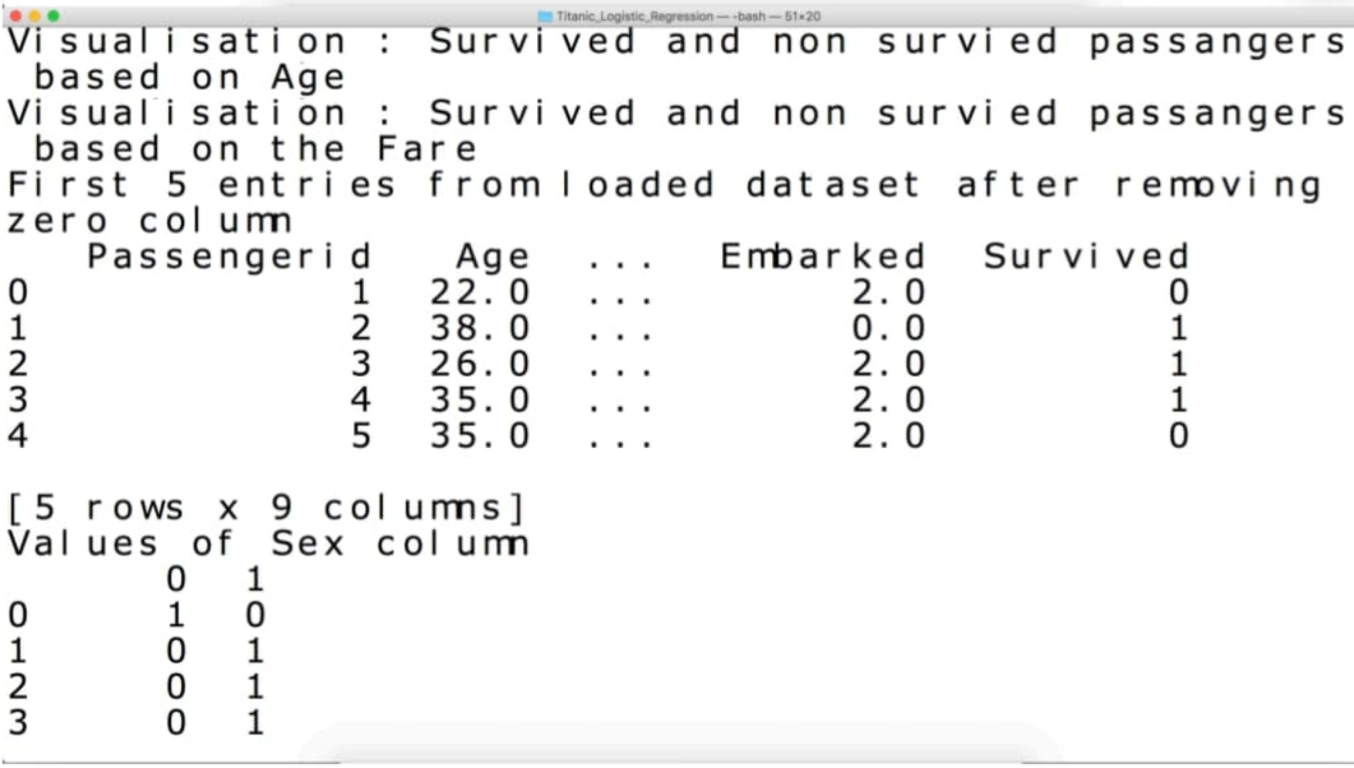


```
    titanic_data["Fare"].plot.hist().set_title("Marvellous Infosystems : Survived and non survied passangers based on Fare")
    show()

    # Step 3 : Data Cleaning
    titanic_data.drop("zero", axis = 1, inplace = True)

    print("First 5 entries from loaded dataset after removing zero column")
    print(titanic_data.head(5))

    print("Values of Sex column")
    print(pd.get_dummies(titanic_data["Sex"]))

    print("Values of Sex column after removing one field")
    Sex = pd.get_dummies(titanic_data["Sex"], drop_first = True)
    print(Sex.head(5))

    print("Values of Plass column after removing one field")
    Pclass = pd.get_dummies(titanic_data["Pclass"], drop_first = True)
    print(Pclass.head(5))

    print("Values of data set after concatenating new columns")
    titanic_data = pd.concat([titanic_data,Sex,Pclass],axis =1)
    print(titanic_data.head(5))

    print("Values of data set after removing irrelevent columns")
    titanic_data.drop(["Sex","sibsp","Parch","Embarked"], axis = 1, inplace = True)
    print(titanic_data.head(5))

    x = titanic_data.drop("Survived",axis = 1)
    y = titanic_data["Survived"]

    # Step 4 : Data Training
    xtrain, xtest , ytrain, ytest = train_test_split(x,y,test_size=0.5)

    logmodel = LogisticRegression()

    logmodel.fit(xtrain,ytrain)

    # Step 4 : Data Testing
    prediction = logmodel.predict(xtest)

    # Step 5 : Calculate Accuracy
    print("Classification report of Logistic Regression is : ")
    print(classification_report(ytest,prediction))

    print("Confusion Matrix of Logistic Regression is : ")
    print(confusion_matrix(ytest,prediction))

    print("Accuracy of Logistic Regression is : ")
    print(accuracy_score(ytest,prediction))

def main():
    print("---- Marvellous Infosystems by Piyush Khairnar-----")

    print("Suervised Machine Learning")

    print("Logistic Regreesion on Titanic data set")

    MarvellousTitanicLogistic()

if __name__ == "__main__":
    main()

```

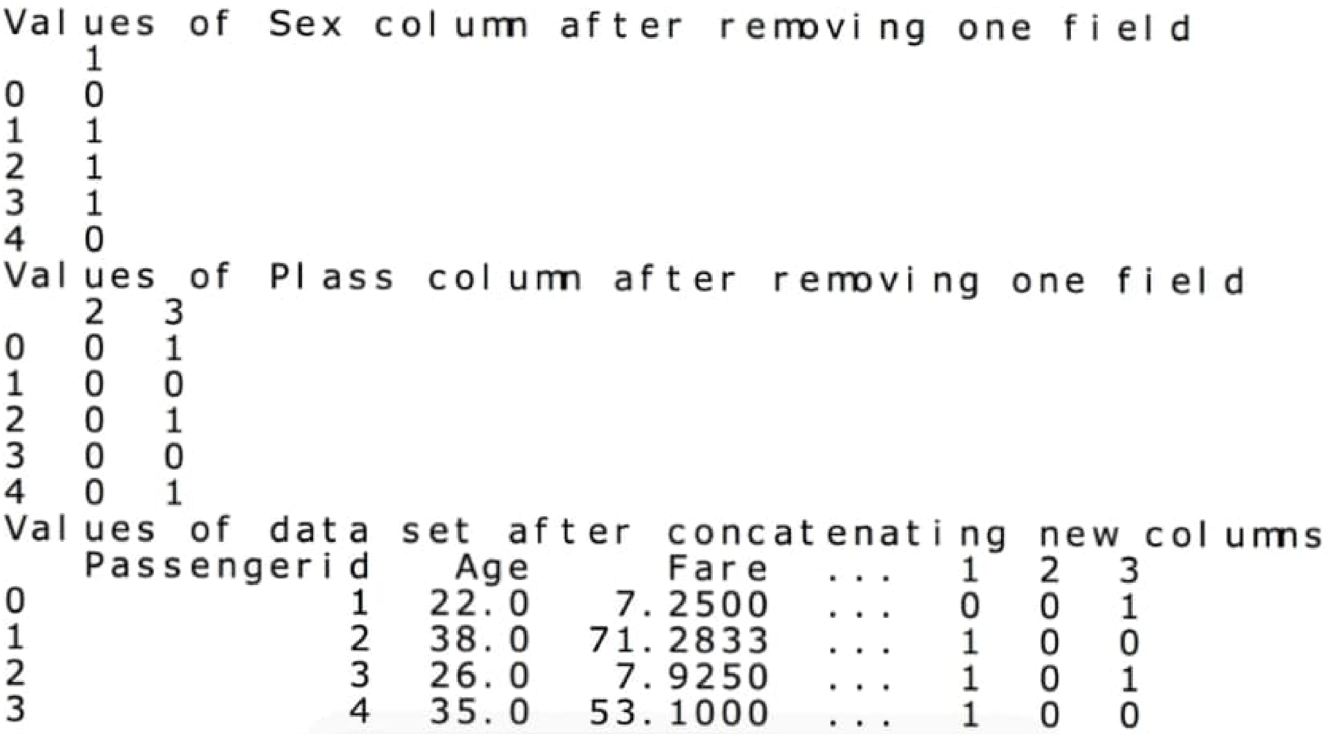
Output of above application


Titanic_Logistic_Regression — python • python old.py — 51×20

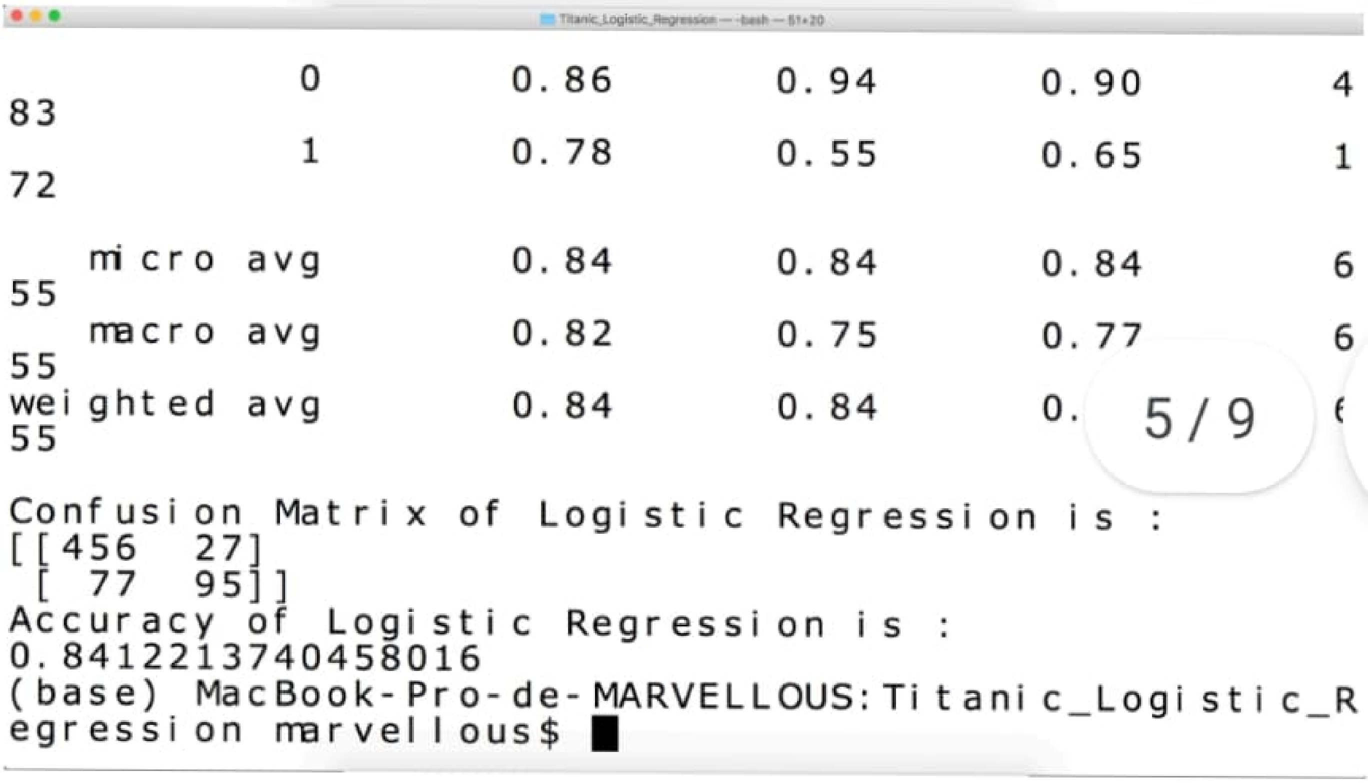
```
egression marvellous$ python old.py
---- Marvellous Infosystems by Piyush Khairnar-----
Suervised Machine Learning
Logistic Regreesion on Titanic data set
First 5 entries from loaded dataset
   Passengerid   Age  ...  Embarked  Survived
0            1  22.0  ...       2.0         0
1            2  38.0  ...       0.0         1
2            3  26.0  ...       2.0         1
3            4  35.0  ...       2.0         1
4            5  35.0  ...       2.0         0

[5 rows x 10 columns]
Number of passangers are 1309
Visualisation : Survived and non survied passangers
Visualisation : Survived and non survied passan
 based on Gender
Visualisation : Survived and non survie
 based on the Passanger class
```



```
Visualisation : Survived and non survied passangers
 based on Age
Visualisation : Survived and non survied passangers
 based on the Fare
First 5 entries from loaded dataset after removing
zero column
   Passengerid   Age  ...  Embarked  Survived
0            1  22.0  ...       2.0         0
1            2  38.0  ...       0.0         1
2            3  26.0  ...       2.0         1
3            4  35.0  ...       2.0         1
4            5  35.0  ...       2.0         0

[5 rows x 9 columns]
Values of Sex column
     0  1
0    1  0
1    0  1
2    0  1
3    0  1
```

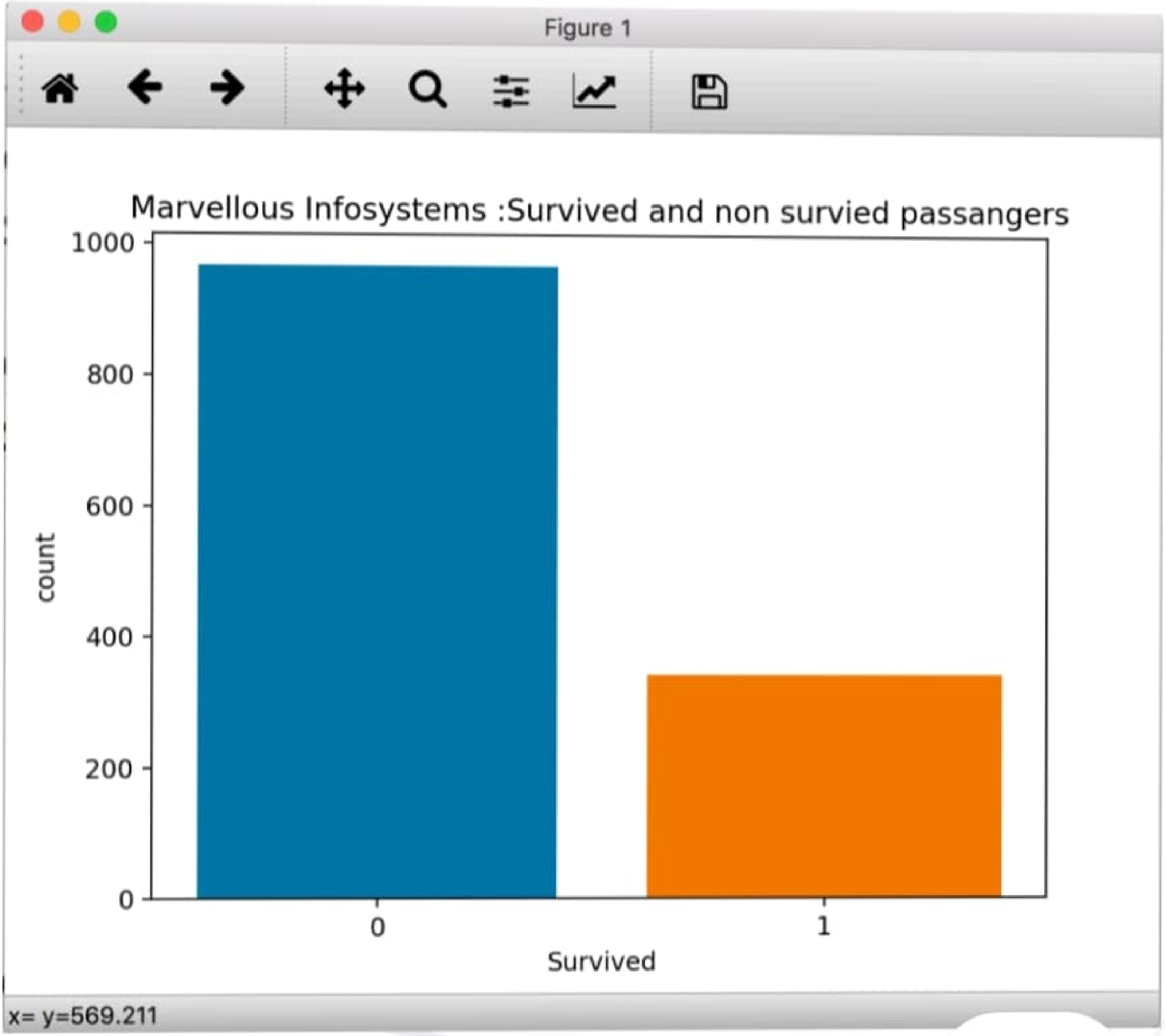


```
Values of Sex column after removing one field
   1
0  0
1  1
2  1
3  1
4  0
Values of Plass column after removing one field
   2  3
0  0  1
1  0  0
2  0  1
3  0  0
4  0  1
Values of data set after concatenating new columns
   Passengerid   Age      Fare  ...  1  2  3
0            1  22.0    7.2500  ...  0  0  1
1            2  38.0   71.2833  ...  1  0  0
2            3  26.0    7.9250  ...  1  0  1
3            4  35.0   53.1000  ...  1  0  0
```

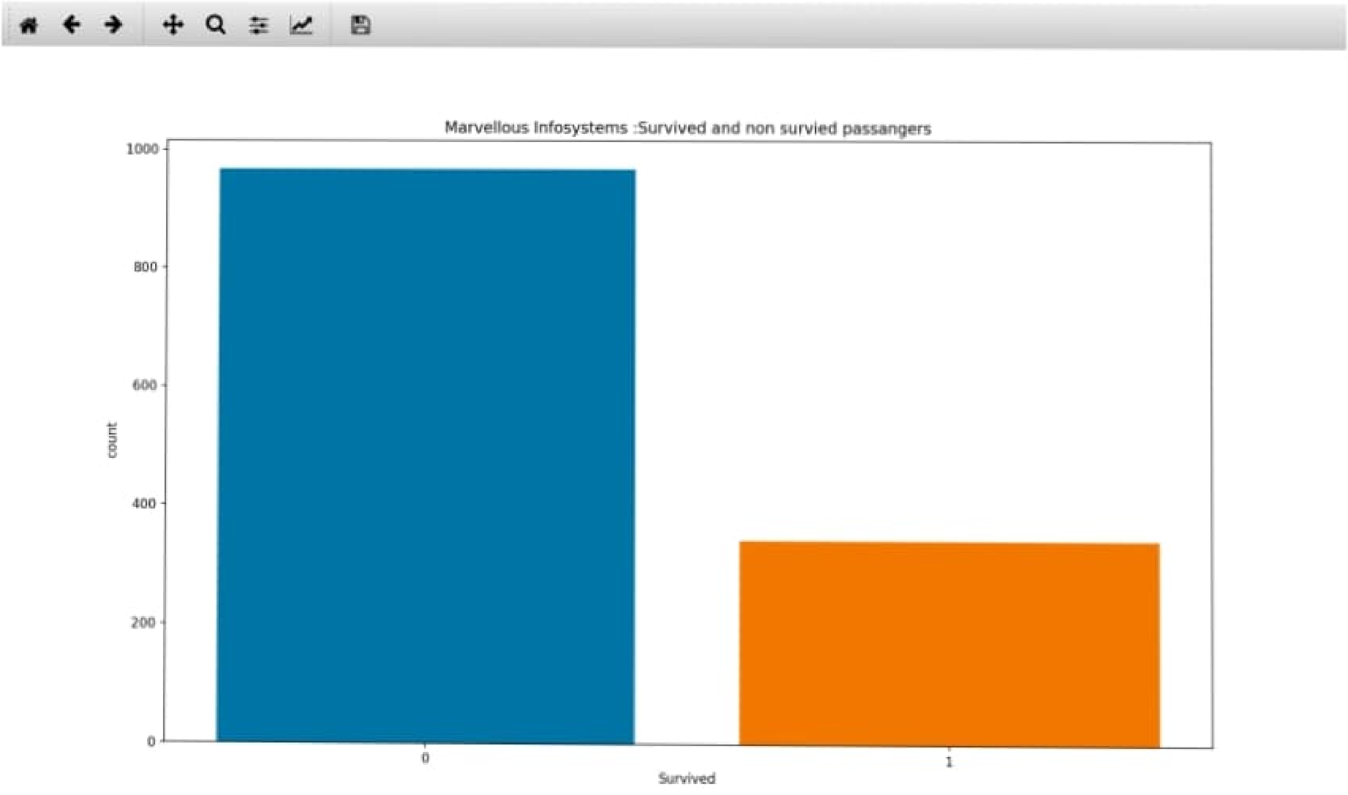


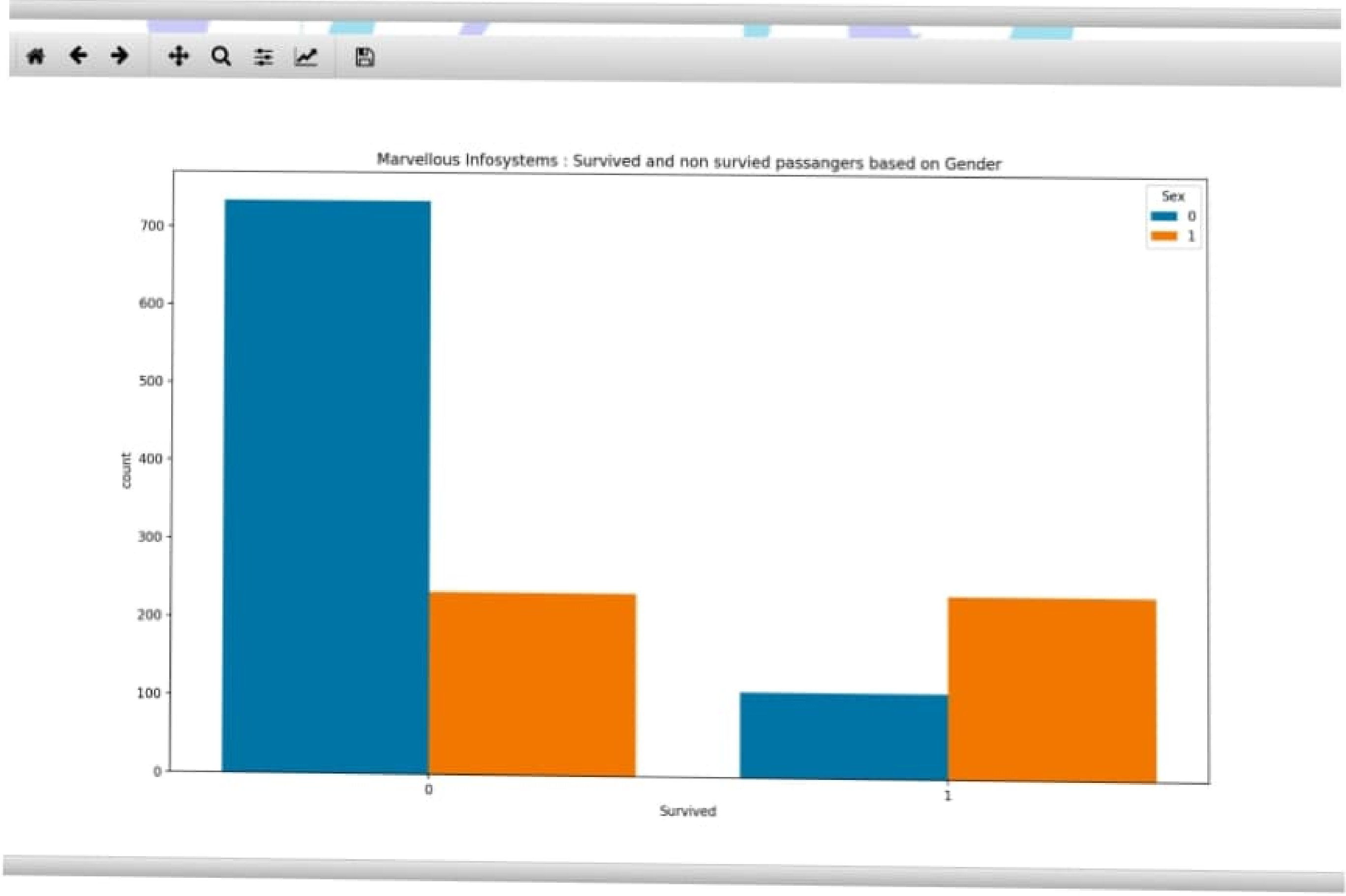
```
            0       0.86      0.94      0.90       4
83
            1       0.78      0.55      0.65       1
72

    micro avg       0.84      0.84      0.84       6
55
    macro avg       0.82      0.75      0.77       6
55
weighted avg        0.84      0.84      0.           6
55

Confusion Matrix of Logistic Regression is :
[[456  27]
 [ 77  95]]
Accuracy of Logistic Regression is :
0.8412213740458016
(base) MacBook-Pro-de-MARVELLOUS:Titanic_Logistic_R
egression marvellous$ █
```

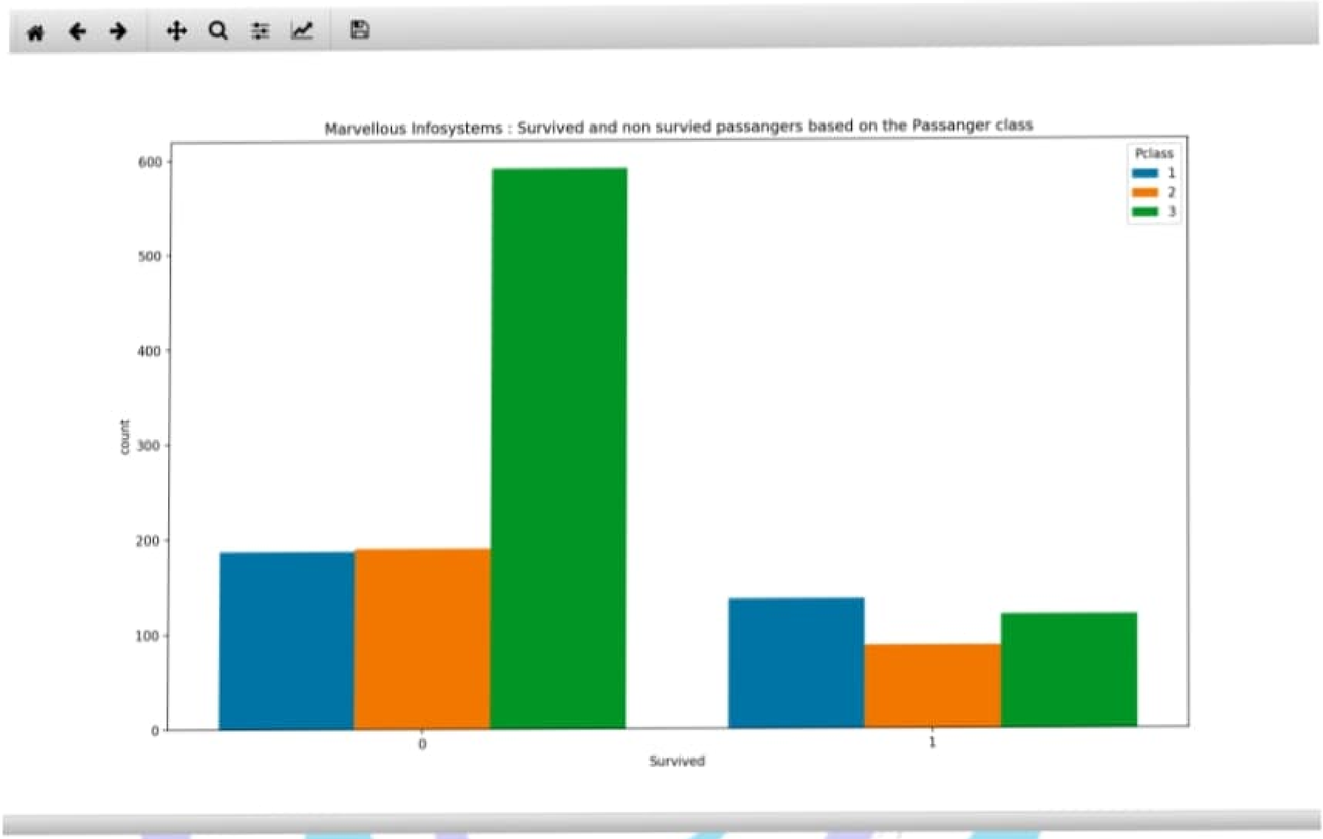
Graphs generated by Data Analysis



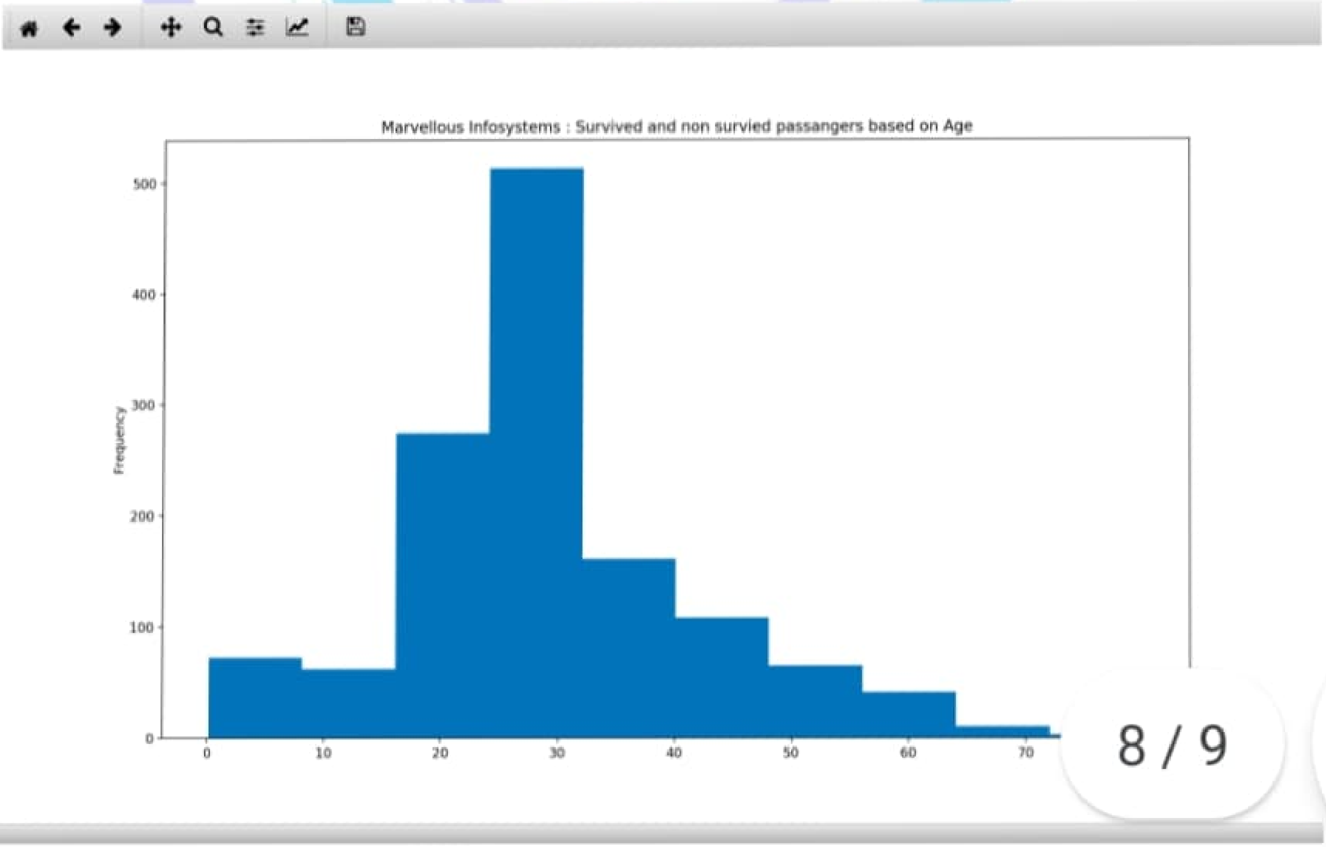
6 / 9


Marvellous Infosystems :Survived and non survied passangers
1000
800
600
400
200
0
count
0
1
Survived

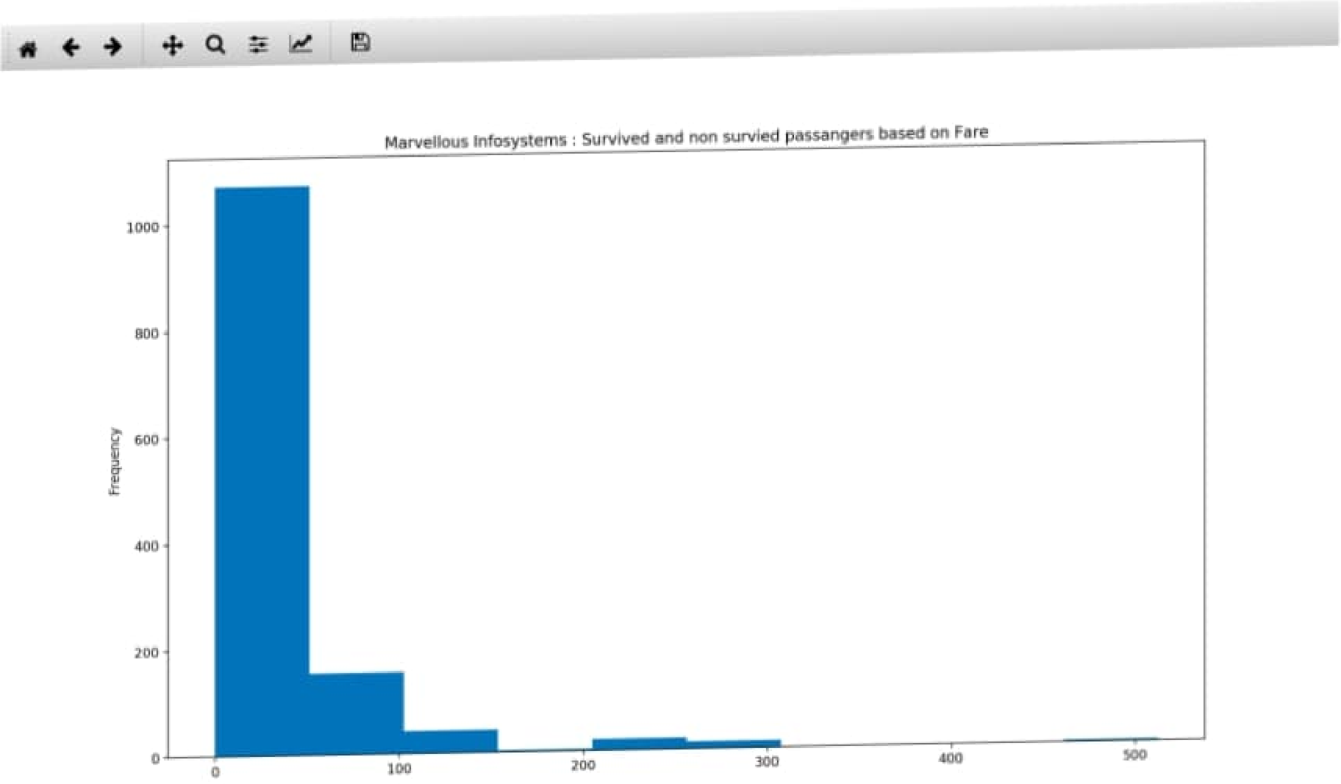

Marvellous Infosystems : Survived and non survied passangers based on Gender
Sex
0
1
700
600
500
400
300
200
100
0
count
0
1
Survived



Marvellous Infosystems : Survived and non survied passangers based on the Passanger class



Marvellous Infosystems : Survived and non survied passangers based on Age


Marvellous Infosystems : Survived and non survied passangers based on Fare
Frequency
0
200
400
600
800
1000
0
100
200
300
400
500