PREDICTING TITANIC SURVIVAL WITH LOGISTIC REGRESSION: A MACHINE LEARNING APPROACH

Niharikareddy Meenigea
Virginia International University
readdyniharika2@gmail.com

ABSTRACT

The sinking of RMS Titanic is one of the most infamous and disastrous shipwrecks ever. During it's voyage and early morning hours of April 15, 1912, the Titanic sank after colliding with an iceberg, killing an approximate of 1502 passengers and crew out of 2224 making it one of many of the deadliest commercial maritime in history. The entire world was deeply shocked and saddened after hearing the news of this disaster which resulted in improved ship safety legislation. It's architect, Thomas Andrews died in the disaster. An observation that came forth from the sinking of Titanic is the fact that certain individuals had a better chance at living than the others. Kids and women had been given the most priority. Social classes were heavily stratified in the early twentieth century, this was especially implemented on the Titanic Firstly, the goal is use and apply exploratory data analytics (EDA) to uncover previously hidden facts in the data set available. Then the task is to later apply various machine learning models to conclude the study of who has a better chance of surviving this disaster. The outcomes of application of the different machine learning models were then set side by side and analyzed based upon precision

KEYWORDS: RMS Titanic, Ship safety legislation, Exploratory data

INTRODUCTION

The disaster that happened over a century ago tore apart many parts of the Titanic during that night. Tragically, there were not enough lifeboats present to rescue all the 2224 passengers onboard. The deceased contain many men whose place were given to the children and women. The aim of this research paper is to accurately predict who would've survived the Titanic given a set of demographic information. A predictive model using passenger data was built so people's genders, their ages, what class of ticket they belonged from, and their socio-economic class, all contributed to whether they would be able to survive or regrettably sink with the Titanic. Predictive analysis is a method of determining important and useful patterns in broad data sets by combining statistical approaches. to determine significant and useful trends in large data. Survival is predicted using machine learning algorithms based on different feature combinations. Goal of this research is ,therefore, to conduct exploratory data analytics to extract different knowledge existing in available data set and to perceive the impact of every field with respect to the passengers' survival by the use of "Survival" field analytics in between each field of the data set. Data analysis on applied algorithms was performed and likewise the accuracy was tested. Based on this, different algorithms are compared, and the best performing model was selected. After analyzing the Titanic dataset, two predictions were generated. The first was to see what the lucky passengers had in common that helped them survive the shipwreck, while the second was to see if I would have survived had I been aboard the fateful ship by applying the tools of machine learning

METHODLOGY

In this research, we will be using Logical Regression to train our model and to see if it is accurate enough to find the mortality status of a random person based upon the input values given to that model.

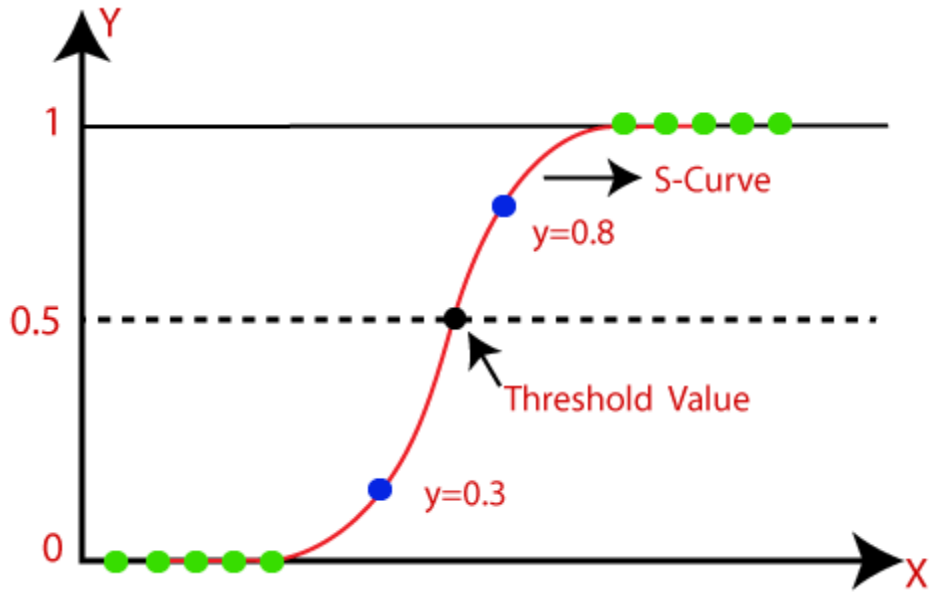

Fig.1 Line graph representing Logistic Function

Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

Importing dependencies:

We will be using: NumPy, pandas,matplotlib, seaborn,sklearn.

As we move ahead, you will get to know the use of each of these modules.

Now, we need to upload the downloaded dataset, into this program, so that our code can read the data and perform the necessary actions using it.

As we have downloaded a CSV file, we shall be using Pandas to store that data in a variable.

Our dataset is now stored in the variable named titanic_data.

To get a brief idea about how the data is loaded, we use the command "variable_name.head()" to get a glimpse of the dataset in the form of a table.

The output came out to be as follows:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

The meaning of the values (SibSp, Parch) can be found on the website from which we have downloaded the dataset.

We have learned from Kaggle while downloading the data set, that the data has 891 rows and 12 columns. Now, let's check how many cells are left empty in the table.

```
titanic_data.isnull().sum()
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

We cannot leave the cells empty, thus have to fill the tables with the most suitable values.

Handling the missing values:

Dropping the "Cabin" column from the data frame as it won't be of much importance

```
titanic_data = titanic_data.drop(columns='Cabin', axis=1)
```

Replacing the missing values in the "Age" column with the mean value

```
titanic_data['Age'].fillna(titanic_data['Age'].mean(), inplace=True)
```

Finding the mode value of the "Embarked" column as it will have occurred the maximum number of times

```
print(titanic_data['Embarked'].mode())
```

Replacing the missing values in the "Embarked" column with mode value

```
titanic_data['Embarked'].fillna(titanic_data['Embarked'].mode()[0], inplace=True)
```

Now let us check if there are still any cells remaining empty.

Running the isnull() command again, we get the satisfactory output, that no such empty cells are present.

We have already noticed from the table, there are two columns that contain string-type values: The "Sex" column and the "Berth" column.

Transformation into a categorical column

Let's convert that into integer type values, and transform it into a categorical column:

```
titanic_data.replace({'Sex':{'male':0,'female':1}, 'Embarked':{'S':0,'C':1,'Q':2}}, inplace=True)
```

Now if we run the titanic_data.head() command again, we find that the values have been replaced successfully.

We also see, that there are few columns, which are not of much importance in this process. Let us get rid of them

```
titanic_data= titanic_data.drop(columns = ['PassengerId','Name','Ticket','Survived'],axis=1)
```

Data Visualisation

Now, we will visualise the data in order to compare and see the data in a more readable format

We will be using the **SEABORN** python library for plotting the graphs

Firstly, we have to set the default seaborn theme by using the command:

```
sns.set()
```

Now that the seaborn library is in place, we will start by plotting a graph showing the number of people that survived and didn't survived

```
# making a count plot for "Survived" column
sns.countplot('Survived', data=titanic_data)
```

OUTPUT:

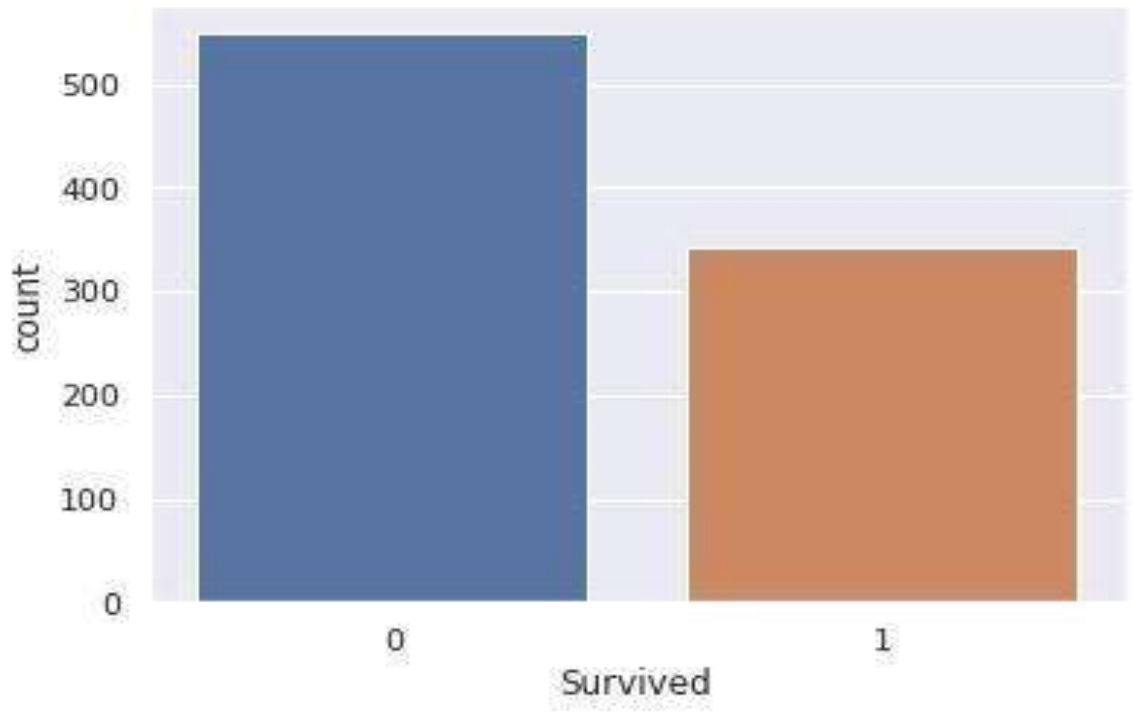

Fig.2 Bar graph showing the people who survived and who didn't survive

Next, we will see the number of male and female passengers on board

```
# making a count plot for "Survived" column
sns.countplot('Sex', data=titanic_data)
```

OUTPUT:

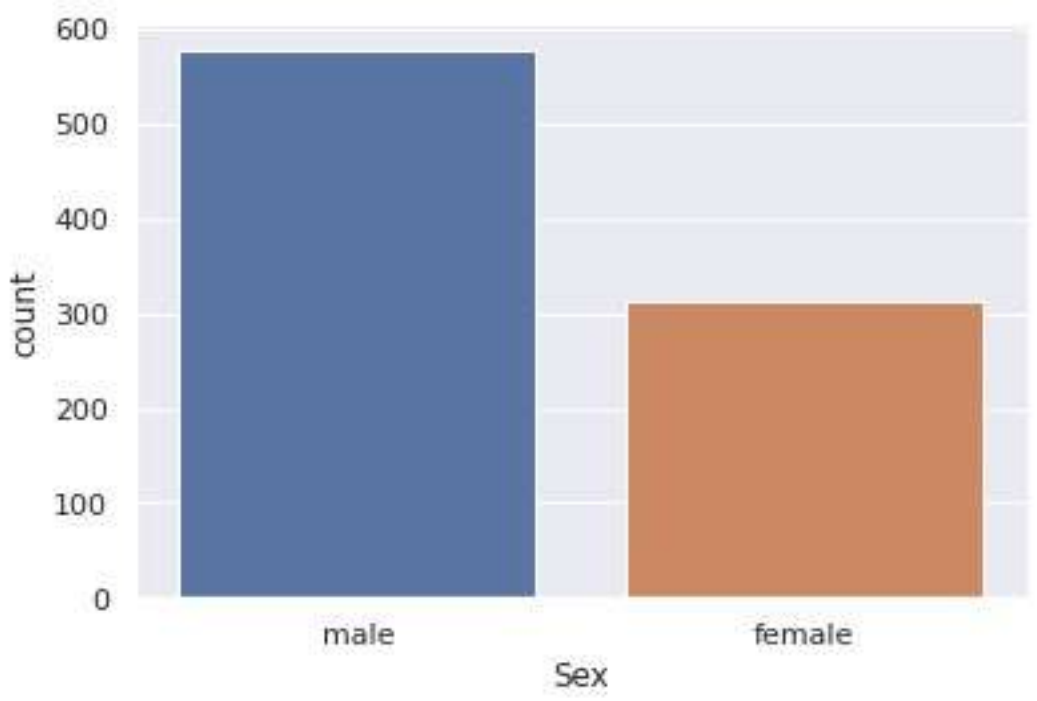

Fig.3 Bar graph showing the number of male and female passengers on board

While the number of male passengers outnumber the female passengers, the sex ratio of who survived however is a different story as females and children would get first priority than males.

To find that, we will use the following command to check how many survived and how many people didn't survive between both sexes

```
# number of survivors Gender wise
sns.countplot('Sex', hue='Survived', data=titanic_data)
```

OUTPUT:

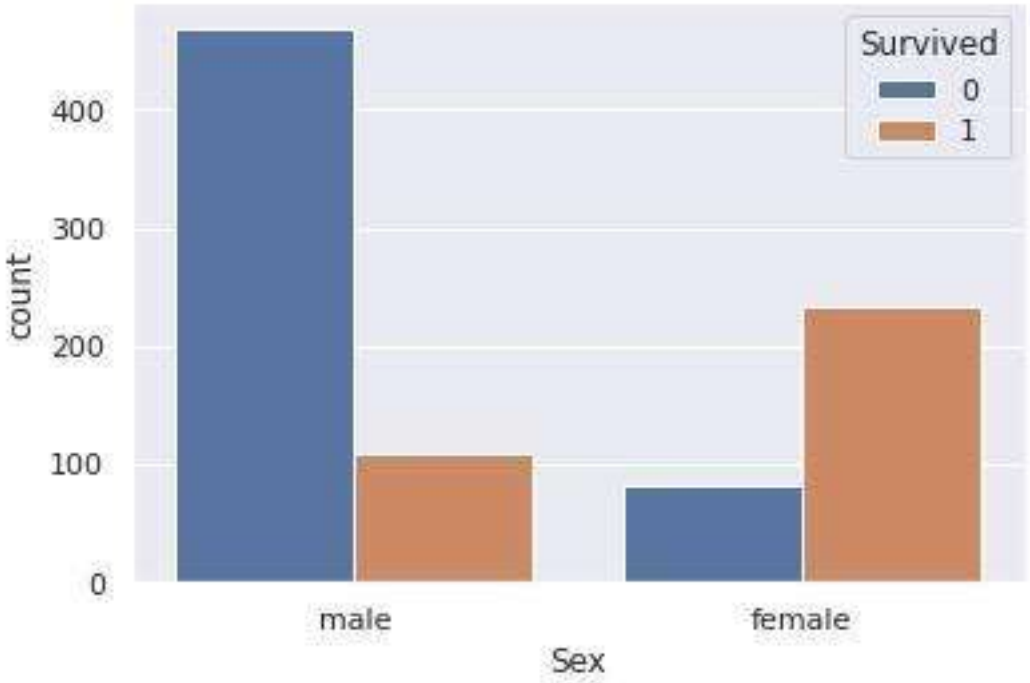

Fig.4 Bar graph showing the survival rate in both males and females

So, now we will check the number of people based on the three different classes available on the ship

```
# making a count plot for "Pclass" column
sns.countplot('Pclass', data=titanic_data)
```

OUTPUT:

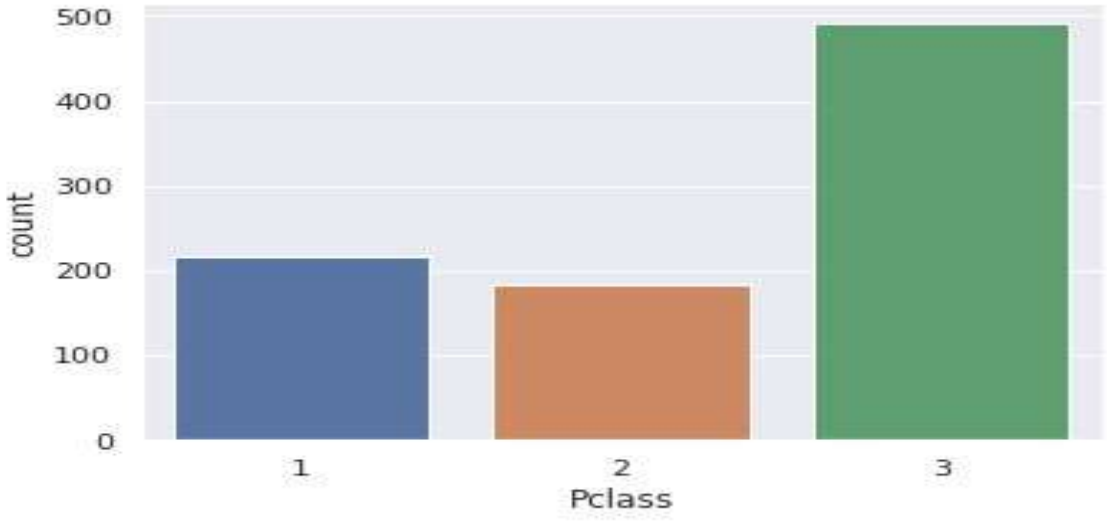

Fig.5 Bar graph showing the number of people onboard based on class

Now we will check the number of people who survived based on class

```
sns.countplot('Pclass', hue='Survived', data=titanic_data)
```

OUTPUT:

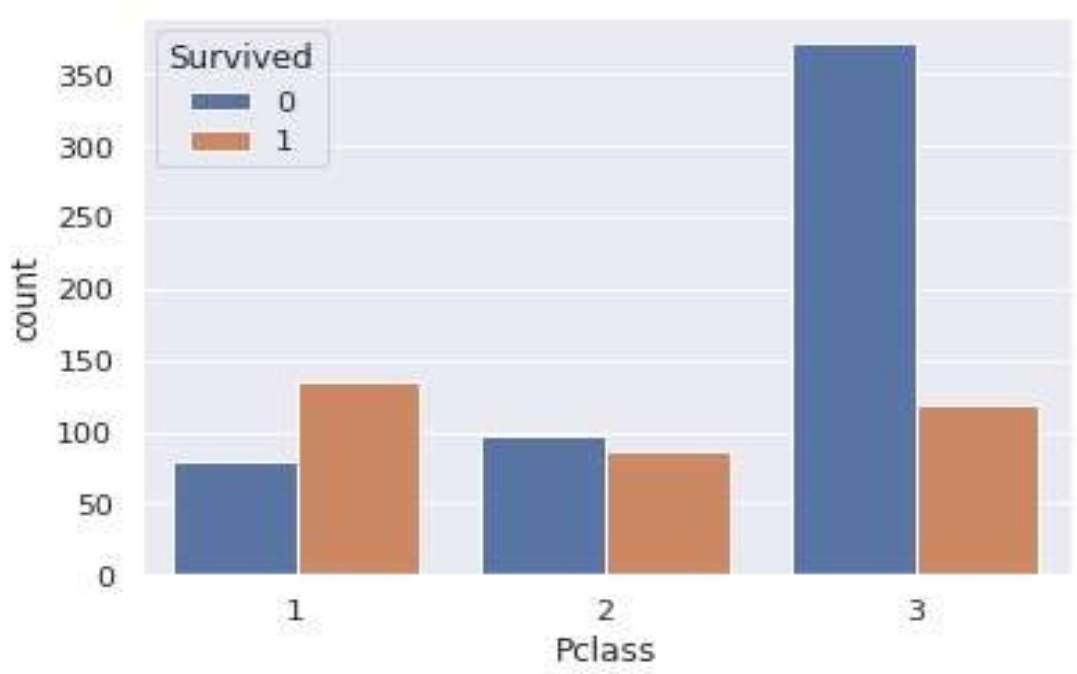

Fig.6 Bar graph showing the survival rate amongst the three different classes

Let's split the data into the target and feature variables.

```
X = titanic_data.drop(columns = ['PassengerId','Name','Ticket','Survived'],axis=1)
Y = titanic_data['Survived']
```

Here, X is the feature variable, containing all the features like Pclass, Age, Sex, Embarked, etc. excluding the Survived column.

Y, on the other hand, is the target variable, as that is the result that we want to determine,i.e, whether a person is alive.

Now, we will be splitting the data into four variables, namely, X_train, Y_train, X_test, Y_test.

```
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size=0.2, random_state=2)
```

Logical Regression:

Let's create a model named model

```
model = LogisticRegression()
```

Now let us train the model, with our training values(X_train , Y_train)

```
model.fit(X_train, Y_train)
```

The model trains in a way like this: "When the values of X are these, the value of Y is this."

Checking the accuracy:

Let's name a variable X_train_prediction, which will store all the predictive outputs of the values X_train

```
X_train_prediction = model.predict(X_train)
```

Now, to check how accurate was its prediction, we compare the values of X_train_prediction with Y_train, which was the original real-life data.

```
training_data_accuracy = accuracy_score(Y_train, X_train_prediction)
print('Accuracy score of training data : ', training_data_accuracy)
```

OUTPUT = 0.8075842696629213

Now, Let's try it again with X_test and Y_test:

```
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(Y_test, X_test_prediction)
print('Accuracy score of test data : ', test_data_accuracy)
```

The output came out to be 0.7821229050279329, which was very close to our test data prediction.

Thus our model is quite accurate as per the data we received.

Checking for a random person:

```
input_data = (3,0,35,0,0,8.05,0)
```

Now let's change these values to a NumPy array:

```
input_data_as_numpy_array = np.asarray(input_data)
```

As our model was trained in different dimensions, we need to reshape this to our target dimension.

```
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)
```

Now, Let's predict using our model:

```
prediction = model.predict(input_data_reshaped)
#print(prediction)
if prediction[0]==0:
    print("Dead")
if prediction[0]==1:
    print("Alive")
```

OUTPUT

APPENDICES(CODE)

import numpy as np import pandas as pd import matplotlib.pyplot as plt import seaborn as sns from sklearn.model_selection import train_test_split from sklearn.linear_model import LogisticRegression from sklearn.metrics import accuracy_score

t_data = pd.read_csv('/content/train.csv') t_data.head() t_data.shape t_data.info() t_data.isnull().sum() t_data['Age'].}llna(t_data['Age'].mean(), inplace=True)
print(t_data['Embarked'].mode()) print(t_data['Embarked'].mode()[0]) t_data['Embarked'].}llna(t_data['Embarked'].mode()[0], inplace=True) t_data.isnull().sum() t_data.describe() t_data['Survived'].value_counts()
sns.set() sns.countplot('Survived', data=t_data) t_data['Sex'].value_counts() sns.countplot('Sex', data=t_data) sns.countplot('Sex', hue='Survived', data=t_data) sns.countplot('Pclass', data=t_data) sns.countplot('Pclass', hue='Survived', data=titanic_data) t_data['Sex'].value_counts() t_data['Embarked'].value_counts() ti_data.replace({'Sex':{'male':0,'female':1}, 'Embarked':{'S':0,'C':1,'Q':2}}, inplace=True) ti_data.head()

X = t_data.drop(columns = ['PassengerId','Name','Ticket','Survived'],axis=1)

Y = t_data['Survived'] print(X) print(Y)

X_train, X_t, Y_train, Y_t = train_test_split(X,Y, test_size=0.2, random_state=2) print(X.shape, X_train.shape, X_t.shape) model = LogisticRegression() model.}t(X_train, Y_train)

X_train_prediction = model.predict(X_train)

print(X_train_prediction) training_data_accuracy = accuracy_score(Y_train, X_train_prediction) print('score of data accuracy : ', training_data_accuracy)

X_t_prediction = model.predict(X_t) print(X_t_prediction) test_data_accuracy = accuracy_score(Y_t, X_t_prediction) print('Accuracy score of test data : ', t_data_accuracy) input_data = (3,0,35,0,0,8.05,0)

input_data_as_numpy_array = np.asarray(input_data) input_data_reshaped = input_data_as_numpy_array.reshape(1,-1) prediction = model.predict(input_data_reshaped)

#print(prediction) if prediction[0]==0:

print("Dead") if prediction[0]==1: print("Alive")

CONCLUSION

In this research, we used logical regression to train our model to analyse the Titanic data. We first removed all the missing values, then changed certain char values into int in order to make it simpler for our machine learning model to understand and analyse, we visualised the data using the seaborn library and finally we implemented logical regression to check whether our training data is accurate enough to find the status of a random passenger. The dataset was acquired from Kaggle.

REFERENCES

[1]. Singh, A., Saraswat, S., & Faujdar, N. (2017, May). Analyzing Titanic disaster using machine learning algorithms. In 2017 International Conference on Computing, Communication and Automation (ICCCA) (pp. 406-411). IEEE.

[2]. Whitley, M. A. (2015). Using statistical learning to predict survival of passengers on the RMS Titanic.

[3]. Lam, E., & Tang, C. (2012). CS229 Titanic–Machine Learning From Disaster.

[4]. Wang, D., Peleg, M., Tu, S.W., Boxwala, A.A., Greenes, R.A., Patel, V.L. and Shortliffe, E.H., 2002. Representation primitives, process models and patient data in computer-interpretable clinical practice guidelines: A literature review of guideline representation models. International journal of medical informatics, 68(1-3), pp.59-70.

[5]. Kakde, Y. and Agrawal, S., 2018. Predicting survival on Titanic by applying exploratory data analytics and machine learning techniques. Int J Comput Appl, pp.32-38.

[6]. Cicoria, S., Sherlock, J., Muniswamaiah, M. and Clarke, L., 2014, May. Classification of titanic passenger data and chances of surviving the disaster. In Proceedings of Student-Faculty Research Day, CSIS (pp. 1-6).

[7]. Chatterjee, T. (2017). Prediction of survivors in titanic dataset: a comparative study using machine learning algorithms. Int J Emerg Res Manag Technol. Department of Management Studies, NIT Trichy, Tiruchirappalli, Tamilnadu, India.