Assignment 2 Version 2: A mathematical essay on logistic regression

S, Karthik
ME18B149 - IDDD Data Science
Indian Institute of Technology Madras
Chennai, India
me18b149@smail.iitm.ac.in

***Abstract*—This document is an overview of the concept of logistic regression. It gives a general idea of the topic, along with the mathematical aspects, as well as an application involving logistic regression based on data from a real-world problem.**

***Index Terms*—Logistic Regression, Machine Learning, Data Science, Titanic survival problem**

I. Introduction

Supervised learning is a machine learning task that learns a model or a function to give an output to every possible input value based on previous data at various instants. Logistic regression is a technique that can be used to handle binary classification problems.

The representation is a model that combines a specific set of input values (x) the solution to which is the predicted output is either 0 and 1 which then determines the output classification of that specific input.

The specific problem discussed is the survival rate of the passengers involved in the titanic shipwreck. Logistic regression will be used to map certain features of the passengers to their survival.

This paper is a case study through which the principles of logistic regression are implemented. It aims to examine which parameters influence the survival rate of the passengers. Moreover, for a set of passengers whose survival is unknown, a model created using the known data will be used to predict the same for the unknown population.

II. Logistic regression

A. Logistic Function

Logistic regression is named for the function used at the core of the method, the logistic function.

The logistic function, also called the sigmoid function was developed by statisticians to describe properties of population growth in ecology, rising quickly and maxing out at the carrying capacity of the environment. It's an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits.

$$sig(t) = \frac{1}{1 + e^{-t}} \tag{1}$$

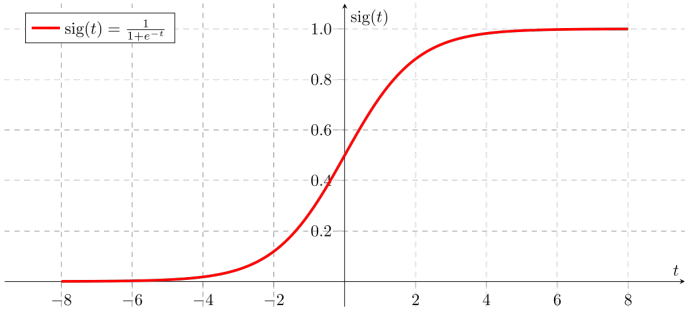

Fig. 1. The logistic function

B. Model Structure

Logistic regression uses an equation as the representation, very much like linear regression.

Input values (x) are combined linearly using weights or coefficient values (θ_i) to predict an output value (y). A key difference from linear regression is that the output value being modeled is a binary values (0 or 1) rather than a numeric value.

Mathematically, the model can be represented as follows

$$z = \boldsymbol{\theta}^T \boldsymbol{x}$$

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = sig(z) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \boldsymbol{x}}} \tag{2}$$

where $h_{\boldsymbol{\theta}}(\boldsymbol{x})$ is the sigmoid output for a given input, $\mathbf{x} = (x_0, x_1, ..., x_n)^T$ is a n-dimensional input feature vector with $x_0 = 1$ (x_0 is not an actual input, it is written along with the inputs for convenience), $\boldsymbol{\theta} = (\theta_0, \theta_1, ..., \theta_n)^T$ are the model parameters and n is the number of input parameters.

C. Significance of $h_{\boldsymbol{\theta}}(\boldsymbol{x})$

The output from the sigmoid is the estimated probability. This is used to infer how confident can predicted value be actual value when given an input $\boldsymbol{x}$. Mathematically this can be written as

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = P(y = 1 | \boldsymbol{x}; \boldsymbol{\theta}) \tag{3}$$

D. Decision Boundary

The sigmoid output, i.e, the probability is a number between 0 and 1. However, the final output is either 0 or 1. This means we need to analyse the probability values and determine which

class the output is. This is where we can set a threshold value k such that if the probability of the output being 1 is greater than the threshold then we can fix it to be 1. Mathematically

$$\begin{aligned} \hat{y} &= 1; \text{if } h_{\boldsymbol{\theta}}(\boldsymbol{x}) > k \\ &= 0; \text{ otherwise} \end{aligned} \tag{4}$$

where $\hat{y}$ is the final output which is the classification predicted for the given inputs. Typically, having k as 0.5 is a fair choice. But the data scientist is free to choose based on the bias required in that particular problem.

E. Cost Function

Cost Function is a measure of how wrong the model is in terms of its ability to estimate the relation between inputs and outputs. The cost function for linear regression (Mean Squared Error) is not used here because logistic regression's cost function will be a non-convex function which means the gradient descent approach will most likely end up in a local minima. Hence, Log loss cost function is used in the case of logistic regression.

$$J(h_{\boldsymbol{\theta}}(\boldsymbol{x}), y) = -y \log h_{\boldsymbol{\theta}}(\boldsymbol{x}) - (1-y)\log(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x})) \tag{5}$$

where y is the observed or given output. In this case, we can see that the above function works how a cost function is expected to be. If y = 1, then the function will be $-\log h_{\boldsymbol{\theta}}(\boldsymbol{x})$ which means as the predicted value is close to 0 (farther from true value), then the cost will be high. Similarly, y = 0 will also have the same effect.

F. Solution Objective

Given a dataset of inputs $\boldsymbol{x}$ and outputs ($\boldsymbol{y}$), the objective of Logistic Regression is to find a function i.e, $h_{\boldsymbol{\theta}}(\boldsymbol{x})$ that minimizes the cost function. Mathematically,

$$minimize_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \tag{6}$$

G. Learning $\boldsymbol{\theta}$ through Gradient Descent

Gradient descent is an optimization algorithm used to minimize the cost function by moving in the direction opposite to that of the gradient of the cost function in an iterative manner. This method will always give a result and is computationally efficient with slight variations in how the data is used (For example, data is processed in batches in Mini-Batch Gradient descent)

To find the gradient, we need to differentiate Equation (5) with respect to $\boldsymbol{\theta}$. This will lead to the following equation

$$\frac{\partial J}{\partial \theta_k} = (h_{\boldsymbol{\theta}}(\boldsymbol{x}) - y)x_k \; ; \quad for \; k \in [1, n] \tag{7}$$

The mathematical implementation of gradient descent is given in Algorithm (1) where α is known as the learning rate which can be appropriately set by the user.

Some of the major choices that have to be made for implementing gradient descent are

Algorithm 1 Gradient Descent

$\boldsymbol{\theta_0}$ = random_init()
for $i = 0, 1, 2, \ldots, num\_iterations - 1$
    $\boldsymbol{\theta^{i+1}} \leftarrow \boldsymbol{\theta^i} - \alpha(h_{\boldsymbol{\theta}}(\boldsymbol{x}) - y)\boldsymbol{x}$
end

1) Parameter Initialization: Ideally, one should choose $\boldsymbol{\theta_0}$ to be close to the minimum. Since in most cases, it is difficult to make a guess of such a choice, they are initialized to zero. A common workaround for this problem is to run the algorithm several times from widely dispersed starting points, and then pick the best result.

2) Stopping Criteria: Stopping criteria is a condition which determines that the gradient descent algorithm has converged. Ideally we want the algorithm to stop when the minimum has been reached. There are multiple ways of defining a stopping criteria 1. Setting the maximum number of iterations apriori 2. Stopping when the change in cost function is less than a certain threshold 3. Stopping when the change in parameters is less than a certain threshold 4. Stopping when the change in gradient is less than a certain threshold

3) Learning Rate: There is a tradeoff in selecting the value of learning rate. If the value is set high, then the gradients can blow up which might lead to skipping the minimum. If it is set to a very low value, then it will take a lot of iterations to converge. A viable solution is to adjust the learning rate dynamically while having a decay factor when it is near the minimum.

III. The Problem

A. Overview and objectives

The sinking of the Titanic is one of the most infamous shipwrecks in history. On April 15, 1912, during her maiden voyage, the widely considered "unsinkable" RMS Titanic sank after colliding with an iceberg. Unfortunately, there weren't enough lifeboats for everyone on-board, resulting in the death of 1502 out of 2224 passengers and crew. While there was some element of luck involved in surviving, it seems some groups of people were more likely to survive than others.

In this assignment, we attempt to build a predictive model that answers the question: "what sorts of people were more likely to survive?" using passenger data (i.e name, age, gender, socio-economic class, etc). Using the predictive model, we predict the survival outcome for some of the passengers for whom it is unknown.

B. Reading the data

We will be using python throughout this assignment. Two similar datasets include passenger information like name, age, gender, socio-economic class, etc. One dataset is titled "train.csv" and the other is titled "test.csv". The train file will contain the details of a subset of the passengers on board (891 to be exact) and importantly, will reveal whether they survived or not, also known as the "ground truth". The test file

dataset contains similar information but the "ground truth" is undetermined for each passenger.

IV. Exploratory Data Analysis

The datasets contains the passenger data for all the passengers onboard the Titanic. The complete details of all the attributes are shown in Fig 2.

Variable	Definition	Key
survival	Survival	0 = No, 1 = Yes
pclass	Ticket class	1 = 1st, 2 = 2nd, 3 = 3rd
sex	Sex	M / F
Age	Age in years	
sibsp	# of siblings / spouses aboard the Titanic	
parch	# of parents / children aboard the Titanic	
ticket	Ticket number	
fare	Passenger fare	
cabin	Cabin number	
embarked	Port of Embarkation	C = Cherbourg, Q = Queenstown, S = Southampton

Fig. 2. Exhaustive list of all the attributes in the given datasets

Our first task is to analyze the data and form visible conclusions and trends from the same with the help of plots.

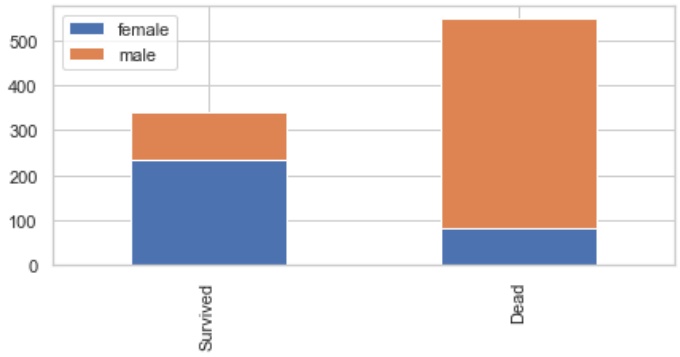

Fig. 3. Bar chart - Genderwise distribution in training data

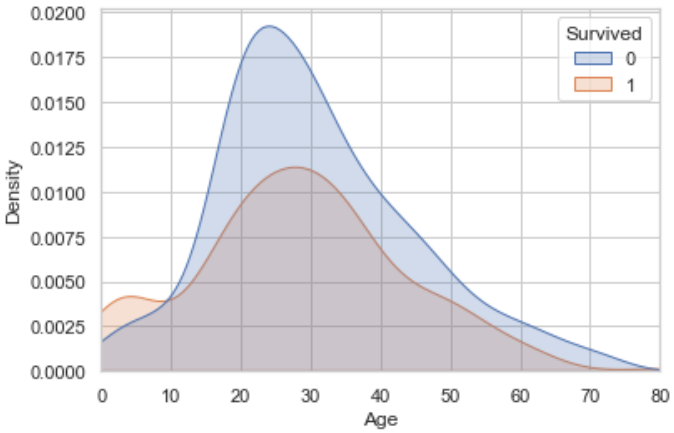

Fig. 4. KDE plot - Age distribution in training data

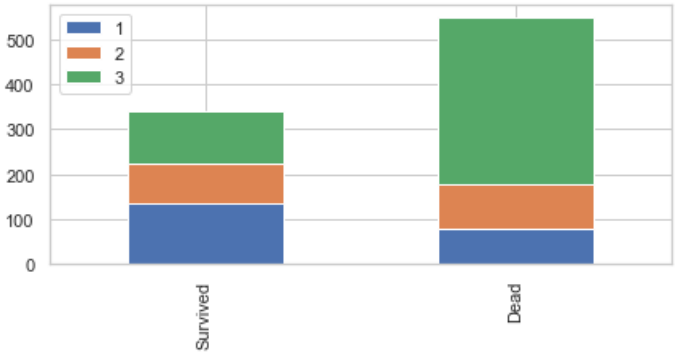

Fig. 5. Bar chart - Class distribution in training data

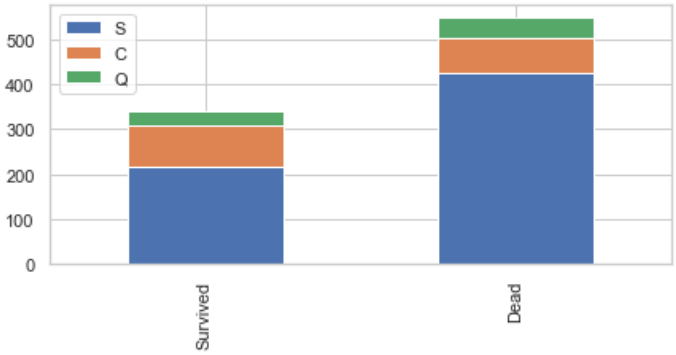

Fig. 6. Bar chart - Embarked location distribution in training data

A. Gender and Age

From Fig 3, we observe that the proportion of females who have survived is much greater than men. Similarly from Fig 4, we can see that more fraction of children survived than all other age categories.

Hence, we can conclude that the rescue boats prioritized women and children first. Also since children occupy lesser space, they could have been accommodated in small spaces more easily than adults.

B. Class and Fare

It is evident that passengers in the more expensive classes are prioritized more and also the safety systems would be better built in those cabins. Hence from Fig 5, we can see that the proportion of 1st class passengers have survived more than that of other classes which is coherent with Fig 7 where the population with higher fares had a better survival rate than the others.

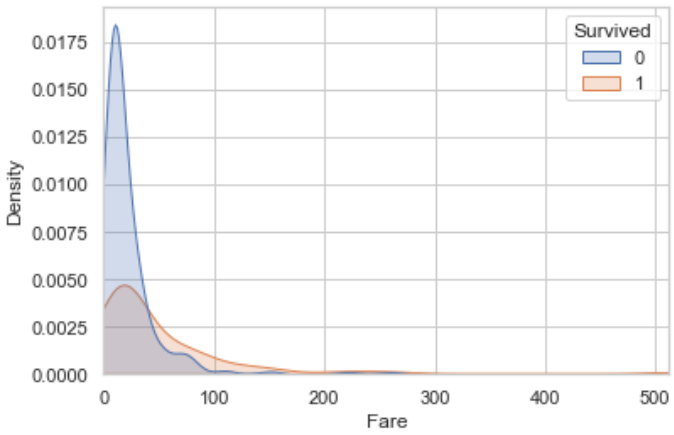

Fig. 7. KDE plot - Fare distribution in training data

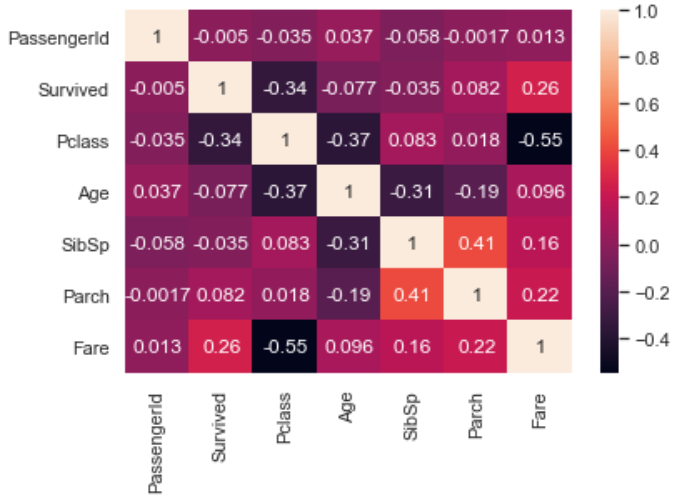

Fig. 8. Correlation matrix - training data

C. Other parameters

1) Place embarked: Fig 6 shows that the fraction of people who survived are less for those who embarked from Southampton and Queenstown and more for those from Cherbourg.

2) Siblings and Parents: From the correlation matrix (Fig 8), we can see that the number of siblings is directly correlated to the number of parents travelling. This is reasonable because siblings are most likely to travel as a family and hence, will travel along with their parents too.

V. Data cleaning

After the initial data analysis, the next task will be to filter out the attributes that are not required or redundant in this analysis. Let us first check the number of unfilled values in each of the parameters. Table I gives the number of empty cells under every feature.

1) Unwanted data: Attributes like Name, PassengerId and Ticket are unique to every individual and would not contribute to any trend in the model. Hence, these parameters are removed from our analysis.

TABLE I
Empty values under each variable

Feature	Train data	Test data
PassengerId	0	0
Pclass	0	0
Name	0	0
Sex	0	0
Age	177	86
SibSp	0	0
Parch	0	0
Ticket	0	0
Fare	0	1
Cabin	687	327
Embarked	2	0

2) Redundant data: Cabin name is an attribute which is very vast and has a lot of choices so wouldn't contribute much to the model. Also, since all cabins of a particular class are usually next to each other, it is the class which is more important. Since class of the passenger exists as Pclass, the Cabin data becomes unnecessary and can be removed.

3) Unreported data: It is observed that Age, Fare, Embarked and Cabin are unreported. Following shows the decision made regarding each of the parameters.

a) Age: Since Age is a very important parameter in this context of this problem, eliminating it from our analysis is not possible. So it was decided to impute the parameter with the median age.

b) Fare: Similar to Age, it was decided to impute the missing values with the median fare value.

c) Embarked: The missing entries were filled with the place where maximum passengers embarked i.e, the mode of the distribution

d) Cabin: As discussed before, this is a redundant parameter. Hence, it can be removed from our analysis.

VI. Data preparation

Since there are parameters which are continuous and almost unique if we directly consider the value(Age, Fare) and also parameters which are strings (Embarked), we need to make all the parameters compatible for the logistic regression model.

TABLE II
Categorization of Age

Greater than	Lesser than	Class number
0	15	0
15	30	1
30	40	2
40	60	3
60	-	4

For Age and Fare, it is necessary to group the values into classes. Based on the distribution (Fig 4 and Fig 7), the parameters were classified as shown in Table II and Table III

For Embarked, it was decided to convert the letter codes to numbers as shown in Table IV.

TABLE III
CATEGORIZATION OF FARE

Greater than	Lesser than	Class number
0	20	0
20	40	1
40	60	2
60	100	3
100	-	4

TABLE IV
NUMERIFICATION OF PLACE EMBARKED

Acronym	Place	Number
S	Southampton	0
C	Cherbourg	1
Q	Queenstown	2

Similarly, Sex was mapped to numbers with male being assigned 0 and female being assigned 1.

The above were done for both the train and the test datasets. After data preparation, the datasets are compatible for the logistic regression model. Some sample entries of the datasets are shown in Table V and Table VI

TABLE V
SAMPLE ENTRIES OF MODEL COMPATIBLE TRAINING DATASET

Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	3	0	1.0	1	0	0.0	0
1	1	1	2.0	1	0	3.0	1
1	3	1	1.0	0	0	0.0	0
1	1	1	2.0	1	0	2.0	0
0	3	0	2.0	0	0	0.0	0

TABLE VI
SAMPLE ENTRIES OF MODEL COMPATIBLE TEST DATASET

S.No	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	3	0	2.0	0	0	0.0	2
1	3	1	3.0	1	0	0.0	0
2	2	0	4.0	0	0	0.0	2
3	3	0	1.0	0	0	0.0	0
4	3	1	1.0	1	1	0.0	0

VII. MODEL PREPARATION AND TRAINING

Since the datasets are now model compatible, the next step is to create the model. A simple logistic regression model was chosen which was implemented from the sklearn package in Python. The training dataset was split into training and validation data, with 10% of training data for validation.

4 different models were created which differ by the regularization implemented. All the models were trained using the training data and the accuracy checked using the validation data. The results are summarized in Table VII

From this we can see that L2 regularization gives the maximum validation accuracy. That model was chosen as the final model to predict the entries in the test dataset. Using this model, the ground truth for the passnegers in the test dataset were determined. The sample of the final outcome is shown in Table VIII

TABLE VII
TRAINING AND VALIDATION RESULTS

Model No.	Regularization	Validation accuracy
1	None	84.44
2	L1	84.44
3	L2	85.56
4	Elasticnet	84.44

VIII. OBSERVATIONS FROM TEST DATASET PREDICTIONS

- Out of the 418 passengers, 164 are predicted to have survived and 254 dead.
- Among the 164 survivors, 145 are female and 19 are male. On the other hand, 247 among the 254 dead were men which indicates that a high fraction of men did not survive. (Fig 9)
- The trend in parameters fare, class and embarked in the test data predictions seem to be similar to that of the training data, where 153 1st class passengers survived and 65 of them died.(Fig 11,12,13)
- One interesting observation regarding the Age distribution is that the fraction of children(age 0-10) surviving is lower than those who didn't in the test dataset (Fig 10) which is opposite to the training dataset (Fig 4).

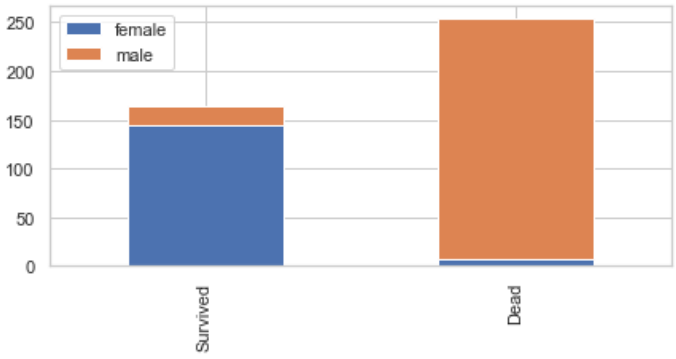

Fig. 9. Bar chart - Genderwise distribution in test data

IX. HYPERPARAMETER TUNING

Since we were able to find a model manually with a validation accuracy of 85.56%, we now can vary some more parameters and perform a grid search with the set of parameters to check if there is a model around the current optimum which is more optimal. The parameters involved in the hyperparameter tuning are shown in Table IX.

Grid Search with Cross Validation (GridSearchCV) was implemented using sklearn where the validation accuracy for the best model came out to be **85.56%**. The parameters for this best model are shown in Table X.

TABLE VIII
TEST DATASET AFTER PREDICTION

PassengerId	Name	Age	Sex	Survived
892	Kelly, Mr. James	34.5	male	0
893	Wilkes, Mrs. James (Ellen Needs)	47.0	female	0
894	Myles, Mr. Thomas Francis	62.0	male	0
895	Wirz, Mr. Albert	27.0	male	0
896	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	22.0	female	1

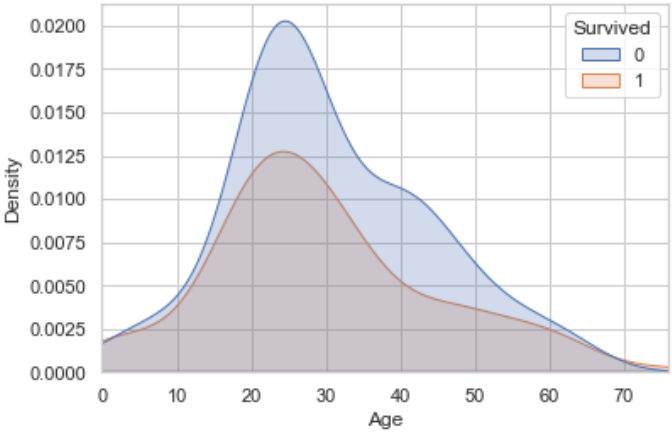

Fig. 10. KDE plot - Age distribution in test data

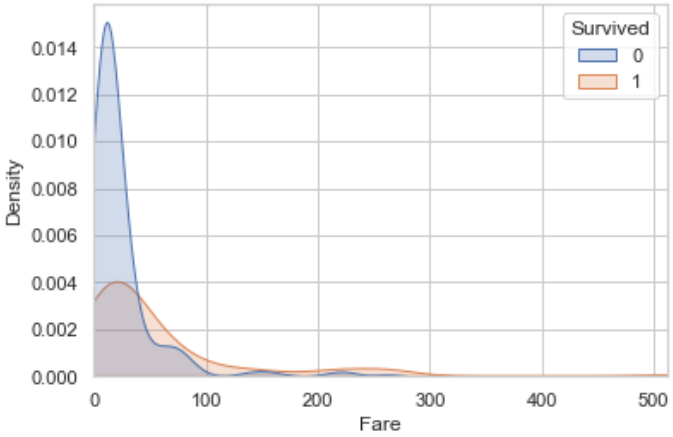

Fig. 13. KDE plot - Fare distribution in test data

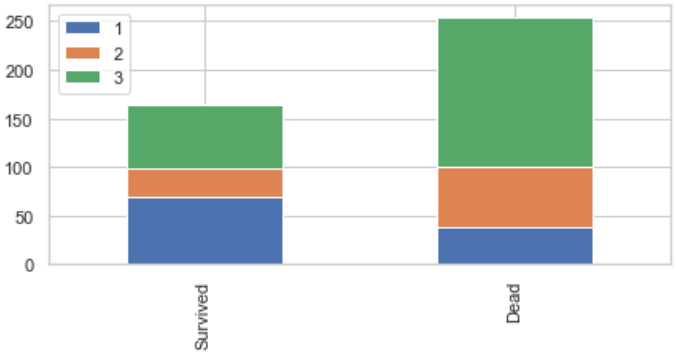

Fig. 11. Bar chart - Class distribution in test data

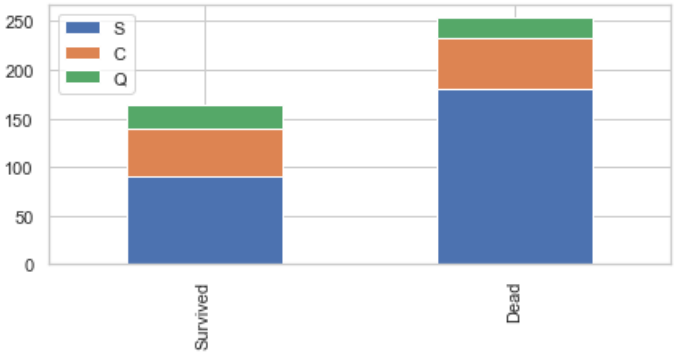

Fig. 12. Bar chart - Embarked location distribution in test data

TABLE IX
HYPERPARAMETER TUNING - GRID SEARCH

S.No	Parameter	Values
1	C	linspace(-5,5,50)
2	solver	'newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'
3	L1-ratio	0,0.25,0.5,0.75,1
4	class$_weight$	'None','balanced'

Hence, we can see that doing a grid search gives a model with the same accuracy as we had done with manual tuning. Therefore, the maximum possible accuracy that is reached by using the LogisticRegression() function in sklearn in **85.56%**.

X. CONCLUSION

The effects of various factors which could have affected the survival outcome of the passengers involved in the sinking of the titanic were analyzed in great detail. Various factors greatly affected the outcome like Gender, Age, Number of siblings and

TABLE X
GRID SEARCH BEST MDOEL

S.No	Parameter	Value
1	C	0.102
2	solver	'liblinear'
3	L1-ratio	0
4	class$_weight$	'None'

parents, the class in which the passenger is travelling and even the place from which the journey was embarked.

A logistic regression model was constructed on the basis of the known data which gave a validation accuracy of 85.56%. Using this model, the survival outcome was predicted for a population for which it was unknown. The trends in the predictions were analyzed and discussed in great detail, which gave a variety of insights to this incident.

Further avenues could be explored using this data where they can be classified using various classifiers available to study how the parameters might possibly affect each other and also how they affect the ground truth. Also, implementing techniques more advanced than Logistic regression might give a model with better accuracy and predictions which would improve on the current logistic regression model.

REFERENCES

[1] Müller, Andreas and Sarah Guido, "Introduction to machine learning with Python: a guide for data scientists", 2016.

[2] Logistic Regression — Detailed Overview, Towards Data Science.

[3] Dr. Kaushik Mitra, Dr. Ramkrishna Pasumarthy, "Regression", EE4708: Data Analytics Laboratory - Week 6

[4] Jason Brownlee, Logistic Regression for Machine Learning, Machine Learning Mastery