VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI-590018, KARNATAKA

A PROJECT REPORT

On

"DISASTER SURVIVAL PREDICTION"

Submitted in Partial Fulfillment for the Award of the Degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING

Submitted By:

UPMA MAURYA	**1SG18CS122**
SRISHTI KUMARI	**1SG18CS114**
SIMRAN MAHTO	**1SG18CS108**
SABHYATA CHAUDHARY	**1SG18CS093**

Under the Guidance of:

Prof. Shobha.K

Assistant Professor

Department of Computer Science and Engineering

(Accredited by NBA)

SAPTHAGIRI COLLEGE OF ENGINEERING

(Affiliated to Visvesvaraya Technological University, Belagavi, Approved by AICTE, NEW DELHI)

ISO 9001-2015 & 14001-2015 Certified, Accredited by NAAC with 'A' Grade

14/5, Chikkasandra, Hesarghatta Main Road, Bengaluru – 560057

2021-2022

SAPTHAGIRI COLLEGE OF ENGINEERING

(Affiliated to Visvesvaraya Technological University, Belagavi& Approved by AICTE, New Delhi.)
ISO 9001-2015 & 14001-2015 Certified, Accredited by NAAC with 'A' Grade
Bengaluru-560057

Department of Computer Science and Engineering

(Accredited by NBA)

Certificate

Certified that the Project Work entitled **"DISASTER SURVIVAL PREDICTION"** carried out by **UPMA MAURYA (1SG18CS122), SRISHTI KUMARI (1SG18CS114), SIMRAN MAHTO (1SG18CS108), SABHYATA CHAUDHARY (1SG18CS093),** bonafide students of **Sapthagiri College of Engineering**, in partial fulfillment for the award of **Bachelor of Engineering** in **Computer Science and Engineering** of **Visvesvaraya Technological University, Belagavi** during the academic year 2021-2022. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the Department Library. The project report has been approved as it satisfies the academic requirements in respect of **Project Work Phase II (18CSP85)** prescribed for the said degree.

Signature of the Guide	**Signature of the HOD**	**Signature of the Principal**
Prof. Shobha.K	**Dr. Kamalakshi Naganna**	**Dr. H Ramakrishna**
Assistant Professsor	**Professor & Head**	**Principal**

EXTERNAL EXAMINATION:

Name of the Examiners	**Signature with Date**
1._____	_____
2._____	_____

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement ground my efforts with success.

I consider it is a privilege to express my gratitude and respect to all those who guided me in completion of technical seminar.

I am, grateful to thank our Principal **Dr. H Ramakrishna**, Sapthagiri College of Engineering, who patronized throughout our career & for the facilities provided to carry out this work successfully.

It's a great privilege to place on record my deep sense of gratitude to our beloved HOD **Dr. Kamalakshi Naganna** of Computer Science & Engineering, who patronized throughout our career & for the facilities provided to carry out this work successfully.

I am also grateful to thank my Technical seminar Guide **Prof. Shobha.K, Assistant Professor** of CSE department for **his/her** invaluable support and guidance.

I would also like to thank the teaching and non-teaching staff members who have helped me directly or indirectly during the technical seminar.

Lastly but most importantly I would like thank my family and friends for their co-operation and motivation to complete this seminar successfully.

UPMA MAURYA (1SG18CS122)
SRISHTI KUMARI (1SG18CS114)
SIMRAN MAHTO (1SG18CS108)
SABHYATA CHAUDHARY (1SG18CS093)

SAPTHAGIRI COLLEGE OF ENGINEERING

(Affiliated to Visvesvaraya Technological University, Belagavi& Approved by AICTE, New Delhi.)
ISO 9001-2015 & 14001-2015 Certified, Accredited by NAAC with 'A' Grade
#14/5, Chikkasandra, Hesaraghatta Main Road,
Bengaluru-560057

Department of Computer Science and Engineering

(Accredited by NBA)

DECLARATION

We **UPMA MAURYA (1SG18CS122), SRISHTI KUMARI (1SG18CS114), SIMRAN MAHTO (1SG18CS108), SABHYATA CHAUDHARY (1SG18CS093),** bonafide students of **Sapthagiri College of Engineering,** hereby declare that the project entitled "**DISASTER SURVIVAL PREDICTION**" submitted in partial fulfilment for the award of Bachelor of Engineering in **Computer Science & Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2021-2022 is our original work and the project has not formed the basis for the award of any other degree, fellowship or any other similar titles.

UPMA MAURYA	**1SG18CS122**
SRISHTI KUMARI	**1SG18CS114**
SIMRAN MAHTO	**1SG18CS108**
SABHYATA CHAUDHARY	**1SG18CS093**

ABSTRACT

The objective is to first explore hidden or previously unknown information by applying exploratory data analytics on available dataset and then apply different machine learning models to complete the analysis of what sorts of people were likely to survive. After this, the results of applying machine learning models are compared and analyzed on the basis of accuracy. Predicting what sorts of people are likely to survive based on the predictive analysis and using tools from machine learning to predict whether the passengers will survive the tragedy. To develop a disaster response web application that will categorize messages into different categories such as medical, food and direct these messages to the right organization in order to provide speedy recovery.

Table of Contents

List of Figures

CHAPTER 1

INTRODUCTION

FIG 1.1: DISASTER SURVIVAL PREDICTION

Machine learning means the application of any computer-enabled algorithm that can be applied against a data set to find a pattern in the data. This encompasses basically all types of data science algorithms, supervised, unsupervised, segmentation, classification, or regression". Few important areas where machine learning can be applied are Handwriting Recognition, Language Translation, Speech Recognition, Image Classification, Autonomous Driving. Some features of machine learning algorithms can be observations that are used to form predictions for image classification, the pixels are the features, For voice recognition, the pitch and volume of the sound samples are the features and for autonomous cars, data from the cameras, range sensors, and GPS.

The goal of the project was to predict the survival of passengers based off a set of data. In this project, we will analyze the Disaster dataset and make two predictions. One prediction to see which passengers on board the ship would survive and then another prediction to see if we would've survived. The historical data has been split into two groups, a 'training set' and a 'test set'. For the training set, we are provided with the outcome (whether or not a passenger survived). We used this set to build our model to generate predictions for the test set. For each passenger in the test set, we had to predict whether or not they survived the sinking. Our score was the percentage of correctly predictions.

Disaster Response Web-Application

In 2019, there were a total of 409 natural disasters worldwide. The irony is that we are right now in the middle of a global pandemic due to Covid19. During a disaster or following the disaster, millions of people communicate either directly or via social media to get some help from the government or disaster relief and recovery services. If the affected person is tweeting it or even sending a message to the helpline service chances are that the message will be lost in the thousands of messages received. Sometimes it's because a lot of people are just tweeting and very few people are needing help and organizations do not have enough time to filter out these many messages manually.

So, in this project, we will be building a disaster response web application that will classify the message into different categories like medical supplies, food, or block road and direct them to the right organization to provide speedy recovery!

1.1 OVERVIEW

- The objective is to perform exploratory data analytics to mine various information in the dataset available and to know effect of each field on survival of passengers by applying analytics between every field of dataset with survival field.
- The prediction are done for newer data sets by applying machine learning algorithm. The data analysis will be done on applied algorithms and accuracy will be checked.
- Different algorithms are compared on the basis of accuracy and the best performing model is suggested for prediction.
- The data analysis will be performed using the passenger's dataset in order to find out the survival likelihood. In this case, Random Forest, to come up with models that can best predict what types of passengers are most likely to survive.
- After disaster, millions of people communicate either directly or via social media to get some help from the government. There are chances that the message will be lost so, organizations don't have enough time to manually filter out all of these messages.
- Therefore, the objective is to develop a disaster response web application that will categorize messages into different categories such as medical ,food and direct these messages to the right organization in order to provide speedy recovery

1.2 Problem Statement

The model uses a disaster dataset that comprises various information about passengers (name, age, socio-economic class, etc). At a later stage, the model is able to predict the likelihood that the arbitrary passenger could have survived.

1.3 Objectives

- The main objective of our project is to predict the number of passenger survived after tragedy and also to develop a prediction system for future, so that if this type of tragedy like flood, cyclone, storm will happen, survivors can be detected easily.
- To develop a disaster response web application that will categorize messages into different categories such as medical, food and direct these messages to the right organization in order to provide speedy recovery.
- The objective is to first explore hidden or previously unknown information by applying exploratory data analytics on available dataset and then apply different

machine learning models to complete the analysis of what sorts of people were likely to survive. After this the results of applying machine learning models are compared and analyzed on the basis of accuracy.

1.4 MACHINE LEARNING

Machine learning (ML) is the study of computer algorithms that improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, speech recognition, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks. A subset of machine learning is closely related to computational statistics, which focuses on making predictions using computers; but not all machine learning is statistical learning. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a related field of study, focusing on exploratory data analysis through unsupervised learning. In its application across business problems, machine learning is also referred to as predictive analytics.

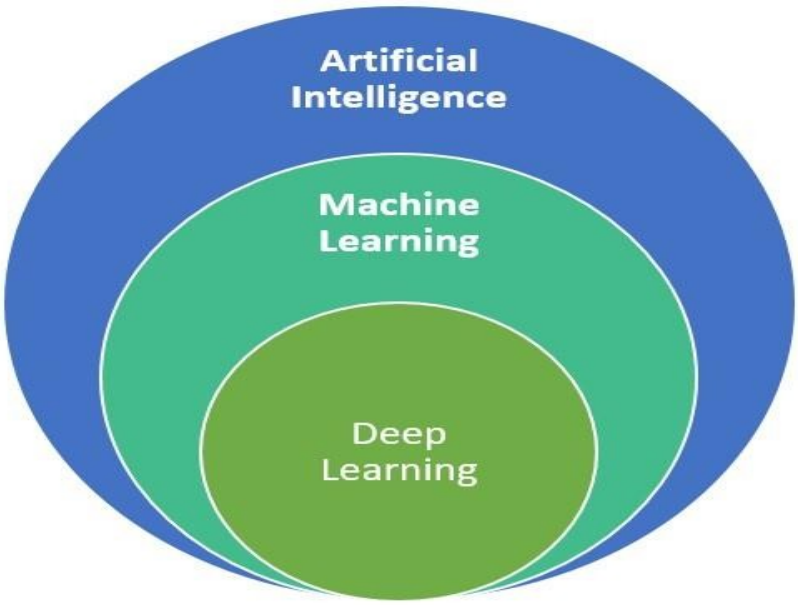


Fig 1.2: Fields of Artificial Intelligence

1.5 MACHINE LEARNING ALGORITHMS

1.5.1 SUPERVISED LEARNING

A support-vector machine is a supervised learning model that divides the data into regions separated by a linear boundary. Here, the linear boundary divides the black circles from the white. Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs. The data is known as training data and consists of a set of training examples. In the mathematical model, each training example is represented by an array or vector, sometimes called a feature vector, and the training data is represented by a matrix. Through iterative optimization of an objective function, supervised learning algorithms learn a function that can be used to predict the output associated with new inputs. An optimal function will allow the algorithm to correctly determine the output for inputs that were not a part of the training data. An algorithm that improves the accuracy of its outputs or predictions over time is said to have learned to perform that task. Types of supervised learning algorithms include active learning, classification and regression. Classification algorithms are used when the outputs are restricted to a limited set of values, and regression algorithms are used when the outputs may have any numerical value within a range. As an example, for a classification algorithm that filters emails, the input would be an incoming email, and the output would be the name of the folder in which to file the email. Similarity learning is an area of supervised machine learning closely related to regression and classification, but the goal is to learn from examples using a similarity function that measures how similar or related two objects are. It has applications in ranking, recommendation systems, visual identity tracking, face verification, and speaker verification. This algorithm consists of a target / outcome variable (or dependent variable) which is to be predicted from a given set of predictors (independent variables). Using these set of variables, we generate a function that map inputs to desired outputs

.

1.5.2 UN-SUPERVISED LEARNING

Unsupervised learning refers to the use of artificial intelligence (AI) algorithms to identify patterns in data sets containing data points that are neither classified nor labeled. The algorithms are thus allowed to classify, label and/or group the data points contained within the data sets without having any external guidance in performing that task. In other words, unsupervised learning allows the system to identify patterns within data sets on its own. In unsupervised learning, an AI system will group unsorted information according to

similarities and differences even though there are no categories provided. Unsupervised learning algorithms can perform more complex processing tasks than supervised learning systems. Additionally, subjecting a system to unsupervised learning is one way of testing AI. However, unsupervised learning can be more unpredictable than a supervised learning model. While an unsupervised learning AI system might, for example, figure out on its own how to sort cats from dogs, it might also add unforeseen and undesired categories to deal with unusual breeds, creating clutter instead of order. AI systems capable of unsupervised learning are often associated with generative learning models, although they may also use a retrieval-based approach. Chat-bots, self-driving cars, facial recognition programs, expert systems and robots are among the systems that may use either supervised or unsupervised learning approaches, or both. In this algorithm, we do not have any target or outcome variable to predict / estimate. It is used for clustering population in different groups, which is widely used for segmenting customers in different groups for specific intervention. Examples of Unsupervised Learning: Apriori algorithm, K-means. Cluster analysis is the assignment of a set of observations into subsets (called clusters) so that observations within the same cluster are similar according to one or more predesignated criteria, while observations drawn from different clusters are dissimilar. Different clustering techniques make different assumptions on the structure of the data, often defined by some similarity metric and evaluated, for example, by internal compactness, or the similarity between members of the same cluster, and separation, the difference between clusters.

NAÏVE BAYES

As a benchmark, we first implemented Naïve Bayes [1]. For the Naïve Bayes model, we considered the following features: 1) sex, 2) passenger class, 3) age, and 4) fare. We chose to use the multinomial event model and Laplace smoothing. First we had to find P(died) and P(survived) by tallying up the number of passengers that survived and dividing by the total number of samples. Both gender and passenger class took on discrete values. Gender has two values, male and female, and passenger class has three values, 1st , 2nd, and 3rd class. Next, we had to estimate the conditional probabilities of these features given whether a passenger survived. For example, to find we had to count the number of male survivors and divide that by the total number of survivors. The same can be done for other features and values. Before computing the parameter estimates for the features age and fare, they first need to be discretized. For age, we grouped the ages into buckets of size 5. We used a default value of -1 for samples for which age information was not provided. Similarly, for

fare, we grouped the fare prices into buckets of size 20 and used -1 for samples with no fare information. After discretizing age and fare, we found the estimate for conditional probabilities of those features given whether a passenger died or survived the same way as before.

The conditional probability computed from the training set is given by

$$p(F = f|S = s) = \frac{\sum_j^n 1\{f_j = f, s_j = s\}}{n}$$

Where is a particular feature and is survival. We iterate through all the training examples. This gives us a conditional probability distribution based on survival. Using this conditional probability distribution, we can compute the probability that a test point survives given the feature set. We use the maximum a posteriori or MAP decision rule as shown below.

Pclass	Sex	Age	Fare	Accuracy
Yes	Yes	Yes	Yes	76.79%
Yes	Yes	No	Yes	75.36%
No	Yes	Yes	Yes	74.40%
Yes	No	Yes	Yes	65.79%
No	Yes	No	No	76.79%

Table 1. Naïve Bayes Accuracy – Using Different Features

SVM

To improve our classification, we used support vector machines [2]. We considered the following features: 1) passenger class, 2) sex, 3) age, 4) number of siblings, 5) patriarchal status, 6) fare, and 7) place of embarkation. We used a Gaussian radial basis function as our kernel and set the tolerance to e= .001

Pclass	Sex	Age	Sibsp	Parch	Fare	Embarked	Accuracy
Yes	Yes	No	No	No	No	Yes	77.99%
No	Yes	Yes	Yes	Yes	No	No	74.88%
No	Yes	No	Yes	No	Yes	No	73.21%
No	No	No	No	Yes	Yes	Yes	67.70%
No	No	Yes	No	Yes	No	Yes	64.83%
No	Yes	Yes	Yes	No	Yes	Yes	61.96%
No	No	Yes	No	No	Yes	Yes	58.13%

Table 2. SVM Accuracy – Using Different Features

Unlike Naïve Bayes, no extra data cleaning was needed. Iterating through all possible feature combinations, we were able to achieve an accuracy rate of 77.99% on the test data set using only three features. The three features that achieve this rate were class, sex and place of embarkation. Using age, fare, and place of embarkation resulted in the worst accuracy of 58.13%. It is interesting to note that this accuracy would be less if we had just

guessed that all test points died (accuracy of 63.23%). This suggests that perhaps class and sex are strong indicators of survival whereas age and fare are weaker indicators of survival. In figure 2, we see the SVM learning curve using the features class, sex and place of embarkation. At around 400 samples, the training curve has reached its asymptotic value of 77.99% and any additional sample does not improve the accuracy.

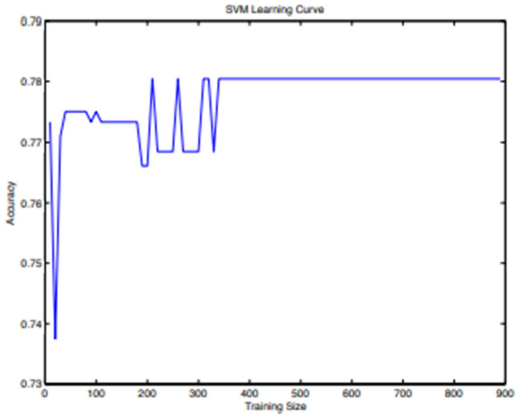


Figure 2. SVM Learning curve using class, sex, and place of embarkation

DECISION TREE

Our prediction system is based on growing Decision Trees to predict the survival status. Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label. Decision tree learning uses a decision tree as a predictive model to go from observations about an item to conclusions about the item's target value. It is one of the predictive modeling approaches used in statistics, data mining, and machine learning. A tree can be “learned” by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called recursive partitioning.

The recursion is completed when the subset at a node all has the same value of the target variable, or when splitting no longer adds value to the predictions. The construction of decision tree classifier does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery.

Decision trees can handle high dimensional data. In general decision tree classifier has good accuracy. Decision tree induction is a typical inductive approach to learn knowledge on classification. Tree models where the target variable can take a discrete set of values are called classification trees; in these tree structures, leaves represent class labels and branches

represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees. In decision analysis, a decision tree can be used to visually and explicitly represent decisions

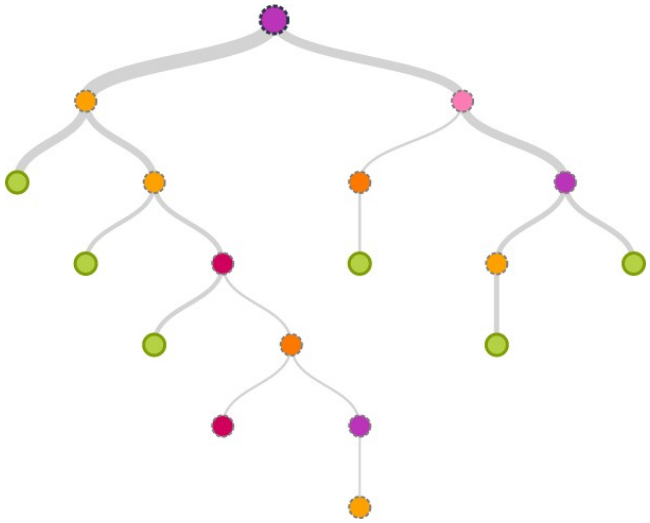


Fig 1.3: Decision Tree

In data mining, a decision tree describes data, but the resulting classification tree can be an input for decision making.

The basic algorithm for growing Decision Tree:

1. Start at the root node as parent node
2. Split the parent node based on field X[i] to minimize the sum of child nodes uncertainty (maximize information gain)
3. Assign training samples to new child nodes
4. Stop if leave nodes are pure or early stopping criteria is satisfied, otherwise repeat step 1 and 2 for each new child node.

Stopping Rules:

1. The leaf nodes are pure
2. A maximal node depth is reached
3. Splitting a node does not lead to an information gain

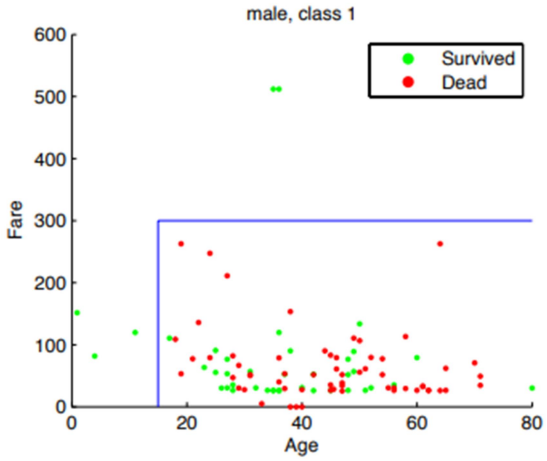


Figure 3. Optimal decision boundaries for the subgroup male, class 1

Figure 4. Optimal decision boundaries for the subgroup male, class 2

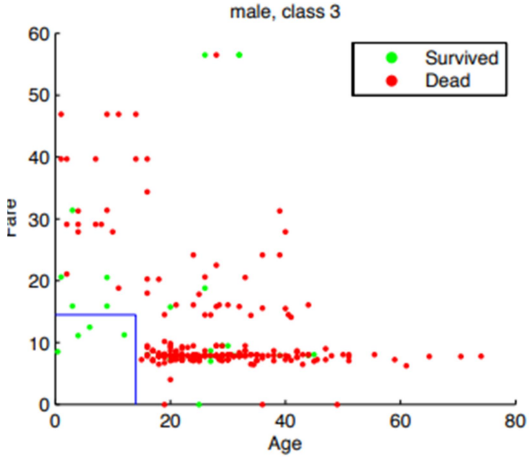


Figure 5. Optimal decision boundaries for the subgroup male, class 3

Figure 6. Fare vs. Age and survival in the subgroup female, class 1

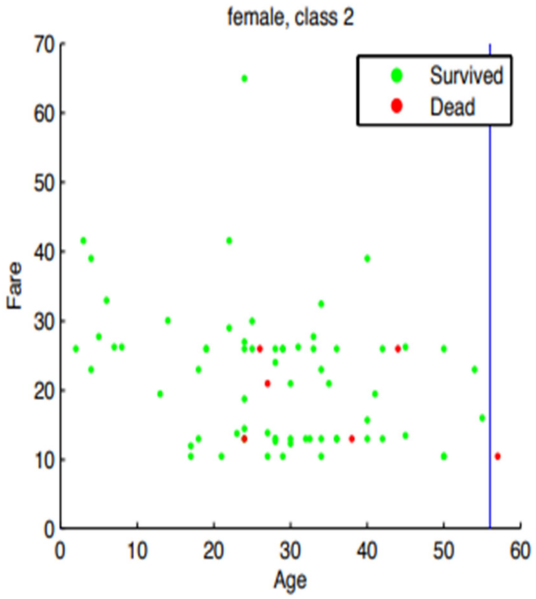


Figure 7. Optimal decision boundaries for the subgroup female, class 2

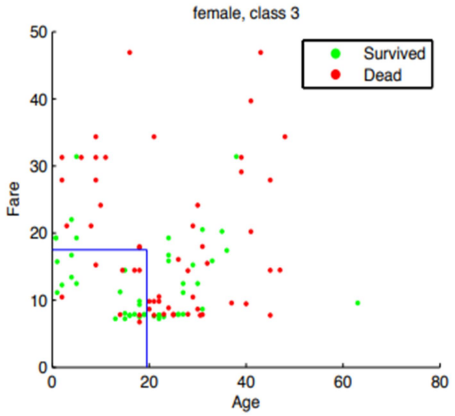


Figure 8. Optimal decision boundaries for the subgroup female, class 3

SUPPORT VECTOR MACHINES

Support-vector machines (SVMs), also known as support-vector networks, are a set of related supervised learning methods used for classification and regression. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that predicts whether a new example falls into one category or the other. An SVM training algorithm is a non-probabilistic, binary, linear classifier, although methods such as Platt scaling exist to use SVM in a probabilistic classification setting. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high dimensional feature spaces.

REGRESSION ANALYSIS

Regression analysis encompasses a large variety of statistical methods to estimate the relationship between input variables and their associated features. Its most common form is linear regression, where a single line is drawn to best fit the given data according to a mathematical criterion such as ordinary least squares.

The latter is often extended by regularization (mathematics) methods to mitigate overfitting and bias, as in ridge regression. When dealing with non-linear problems, go-to models include polynomial regression logistic regression or even kernel regression, which introduces non-linearity by taking advantage of the kernel trick to implicitly map input variables to higher-dimensional space.

RANDOM FOREST AND EXTRA TREES

One common issue with all machine learning algorithms is Overfitting. For Decision Tree, it means growing too large tree (with strong bias, small variation) so it loses its ability to generalize the data and to predict the output. In order to deal with overfitting, we can grow several decision trees and take the average of their predictions. The library SciKit-Learn provides to such algorithm Random Forest and ExtraTrees.

In Random Forest, we grow N decision trees based on randomly selected subset of the data and randomly selected M fields, where $M = \sqrt{total\ \#\ of\ fields}$. In ExtraTrees, in addition to randomness of subsets of the data and of field, splits of nodes are chosen randomly.

A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as bagging. The basic idea behind this is to combine

multiple decision trees in determining the final output rather than relying on individual decision trees. Random Forest has multiple decision trees as base learning models. We randomly perform row sampling and feature sampling from the dataset forming sample datasets for every model. This part is called Bootstrap.

We need to approach the Random Forest regression technique like any other machine learning technique:

• Design a specific question or data and get the source to determine the required data.

• Make sure the data is in an accessible format else convert it to the required format.

• Specify all noticeable anomalies and missing data points that may be required to achieve the required data.

• Create a machine learning model

• Set the baseline model that you want to achieve

• Train the data machine learning model.

• Provide an insight into the model with test data

• Compare the performance metrics of both the test data and the predicted data from the model.

• If the expectations are not satisfied, you can try improving your model accordingly or dating your data or use another data modeling technique.

• At this stage you interpret the data you have gained and report accordingly.

BAYESIAN NETWORK :

A Bayesian network, belief network, or directed acyclic graphical model is a probabilistic graphical model that represents a set of random variables and their conditional independence with a directed acyclic graph (DAG). For example, a Bayesian network could represent the probabilistic relationships between diseases and symptoms. Given symptoms, the network can be used to compute the probabilities of the presence of various diseases. Efficient algorithms exist that perform inference and learning. Bayesian networks that model sequences of variables, like speech signals or protein sequences, are called dynamic Bayesian networks.

1.6 ORGANIZATION OF THE PROJECT REPORT

• Chapter 1: Discuss the introduction about the wireless sensor network, irrigation management, problem statement and objectives are discussed.

• Chapter 2: Contains the information about the works on the survey.

• Chapter 3: Discuss the complete details about the requirement of the system and describe the functional and non-functional requirements also software and hardware used.

• Chapter 4: Discuss the information about the system architecture; it has the design of the system.

• Chapter 5: Discuss briefly about the functionalities and the various models that are implemented.

• Chapter 6: Discuss the various experiments and results conducted.

• Chapter 7: Discuss the different test cases and its results. Also, the snapshots of the output are included.

• Chapter 8: Discuss conclusions made from the results and future improvements that can be done are included.

CHAPTER 2
LITERATURE REVIEW

2.1 LITERATURE SURVEY

SL no.	**Title**	**Advantages/Disasvantages**
1	Predicting the Survival Rate of Titanic Disaster Using Machine Learning Approaches. April 2020, IEEE	Limited to calculate accuracy of train set and test set using cross validation technique.
2	Analyzing Titanic disaster using machine learning algorithms. May 2017, IEEE	This research work compares the algorithms on the basis of the percentage on a test dataset.
3	Predicting likelihood or survivals using machine learning. March 2021, IEEE	Different machine learning algorithms are used to get the maximum accuracy.
4	Exploratory Data Analysis and Machine Learning on Titanic Disaster Dataset. January 2021, IEEE	Result was inaccurate because of absent data. Need to compute missing values to get accurate result.

2.2 LITERATURE REVIEWS

[1]" Analyzing Titanic disaster using machine learning algorithms" Aakriti singh, Shipra Saraswat

Titanic disaster occurred 100 years ago on April 15, 1912, killing about 1500 passengers and crew members. The fateful incident still compel the researchers and analysts to understand what can have led to the survival of some passengers and demise of the others. With the use of machine learning methods and a dataset consisting of 891 rows in the train set and 418 rows in the test set, the research attempts to determine the correlation between factors such as age, sex, passenger class, fare etc. to the chance of survival of the passengers. These factors may or may not have impacted the survival rates of the passengers. In this research paper, various machine learning algorithms namely Logistic Regression, Naive Bayes, Decision Tree, Random Forest have been implemented to predict the survival of passengers. In particular, this research work compares the algorithm on the basis of the percentage of accuracy on a test dataset.

Advantages/Disadvantages

This research works compares the algorithm on the basis of the percentage of the test dataset.

[2]"Predicting the Survival Rate of Titanic Disaster Using Machine Learning Approaches" Jyothi Shetty, S Pallavi

The Titanic incident has led the scientist and investigators to comprehend what can have prompted the survival of a few travelers and death of the rest. Many machine learning algorithms contributed in predicting the survival rate of passengers. In addition to the this, a dataset of 891 rows which includes the attributes namely Age, Passenger ID, Sex, Name, Embarked, Fare etc. has been used. In this paper, survival of passengers is figured out using various machine learning techniques namely decision tree, logistic regression and linear SVM. The main focus of this work is to differentiate between the three different machine learning algorithms to analyze the survival rate of traveller based on the accuracy.

Advantages/Disadvantages

Limited to calculate accuracy of train set and test set using cross validation approaches.

[3]"Predicting the Likelihood of Survival of Titanic's Passengers by Machine Learning" Anasuya Dasgupta, Ved Prakash Mishra

The sinking of RMS Titanic is presumably one of the most infamous and disastrous shipwrecks in history. During her maiden voyage and early morning hours of April 15, 1912, the Titanic regrettably 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 till date. The entire international community was deeply shocked and saddened after hearing the news of this sensational disaster which resulted in improved ship safety legislation. Her architect, Thomas Andrews died in the disaster. An eye-opening observation that came forth from the sinking of Titanic is the fact that some individuals had a better chance at surviving than the others. Kids and women had been given foremost priority. Social classes were heavily stratified in the early twentieth century, this was especially implemented on the Titanic Firstly, the aim is used and apply exploratory data analytics (EDA) to uncover previously unknown or hidden facts in the data set available. Then the task is to later anoint various machine learning models to conclude the study of which types of individuals are more likely to live. The outcomes of application of the different machine learning models were then set side by side and analyzed based upon precision.

Advantages/Disadvantages

Different machine learning algorithms ae used to get maximum accuracy.

[4]" Exploratory Data Analysis and Machine Learning on Titanic Disaster Dataset. January 2021, IEEE" Yogesh kakade

The sinking of the RMS Titanic caused the death of thousands of passengers and crew is one of the deadliest maritime disasters in history. One of the reasons that the shipwreck led to such loss of life was that there were not enough lifeboats for the passengers and crew. The interesting observation which comes out from the sinking is that some people were more likely to survive than others, like women, children were the one who got the priority to rescue. The objective is to first explore hidden or previously unknown information by applying exploratory data analytics on available dataset and then apply different machine learning models to complete the analysis of what sorts of people were likely to survive.

After this the results of applying machine learning models are compared and analyzed on the basis of accuracy.

Advantages/Disadvantages

Result was inaccurate because absent of data. Need to compute missing values to get accurate result.

CHAPTER 3

ANALYSIS

3.1 Existing System

3.1.1 Description

Analysis of Disaster is essential in order to understand the historical data. The correlation between the independent and dependent features was observed in order to determine features that may have impact on passenger survival. Existing system has explored the disaster data and three machine learning algorithms namely XGBoost, Decision trees, Random forests were implemented to predict survival rate of passengers. Some factors such as "Age", "Gender", "Children" played a key role towards the survival of Passengers. A comparative analysis between these algorithms was conducted and the models were evaluated by some metrics. Based on the analysis and metrics evaluation, XGBoost outperformed other implemented algorithms in this work.

3.1.2 Drawback

- The number of classification classes was also around two to three. Many of the systems proposed earlier could offer an accuracy in the range of 75 to 85% only.
- Many systems used very small dataset for the classification which could not lead to accurate classification.
- Several models were developed but were not deployed in a way which could be used by the end user.

3.2 Proposed System

3.2.1 Description

- The proposed system is based on using Several machine learning algorithms. The proposed system makes use of comparatively larger dataset for the prediction of Disaster Survivors.
- Its domain includes uploading survivors messages on a disaster response web application and that will categorize messages into different categories such as medical ,food and direct these messages to the right organization in order to provide speedy recovery
- The proposed system will make use of various machine learning algorithms present in numpy, pandas in order to improve the accuracy and efficiency of the model.
- After developing an accurate and successful prediction model, the model will be further deployed into a web application so that its use becomes easy.

3.2.2 Advantages

- The proposed system offers a great score of accuracy in very less time.
- A user can easily upload the messages in it and can make the classification therefore it can be used by a naive person too and direct these messages to the right organization in order to provide speedy recovery.
- help for the elimination of superfluous hardware and reduce the risk of human error.
- A better understanding of scalability.
- Greater Survivors Retention.
- Increased Employee Productivity.

3.3 Requirement Specification

A System Requirement Specification (SRS) is basically an organization's understanding of a customer or potential client's system requirements and dependencies at a particular point prior to any actual design or development work. The information gathered during the analysis is translated into a document that defines a set of requirements. It gives a brief description of the services that the system should provide and also the constraints under which, the system should operate. Generally, SRS is a document that completely describes what the proposed software should do without describing how the software will do it. It's a two-way insurance policy that assures that both the client and the organization understand the other's requirements from that perspective at a given point in time.

3.3.1 Functional Requirements

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing, and other specific functionality. In this system following are the functional requirements:

- The control is provided to the end user (for example Survivors)
- The message is required to be the input in the model
- As soon as the person clicks on classify button, it classifies the message into different categories.
- After classification, the end user should take the required necessary measures.

3.3.2 Non-Functional Requirements

Non-functional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviours.

They may relate to emergent system properties such as reliability, response time and store occupancy. Non-functional requirements arise through the user needs, because of budget constraints, organizational policies, the need for interoperability with other software and hardware systems or because of external factors such as:

Reliability: The system should be trustworthy and reliable in providing the functionalities. Once a user made some change it could be reflected in the system.

- **Maintainability:** The system should be monitored, and maintenance should be simple and objective in its approach.

Following are some non-functional requirements:

- The message should be proper.
- The system should be easy for usability and self-descriptive for maintenance purposes.

3.3.3 Software Requirements

The software requirements are the description of the features and functionalities of the target system. Requirements convey the expectations of users from the software product. The requirements can be obvious or hidden, known or unknown, expected, or unexpected from the client's point of view. In this system following are the software requirements:

- Jupyter Notebook
- Windows 10 and above
- Python 3.8 and above
- Anaconda
- Spyder

3.3.4 Hardware Requirements

- Inspiron 15 3593 Series
- 10th Generation Intel® Core™ i5-1035G1 Processor (6MB Cache, up to 3.6 GHz)
- Central Processing Unit (CPU) . An AMD equivalent processor will also be optimal.
- RAM — 8 GB minimum, 16 GB or higher is recommended Operating System — Ubuntu or Microsoft Windows 10.

CHAPTER 4

DESIGN

Design is a meaningful engineering representation of something that is to be built. It is the most crucial phase in the development of a system. Software design is a process through which the requirements are translated into a representation of software. Design is a place where design is fostered in software Engineering. Based on the user requirements and the detailed analysis of the existing system, a new system must be designed. This is the phase of system designing.

4.1 System Architecture

A system architecture is the conceptual model that defines the structure, behaviour, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviours of the system.

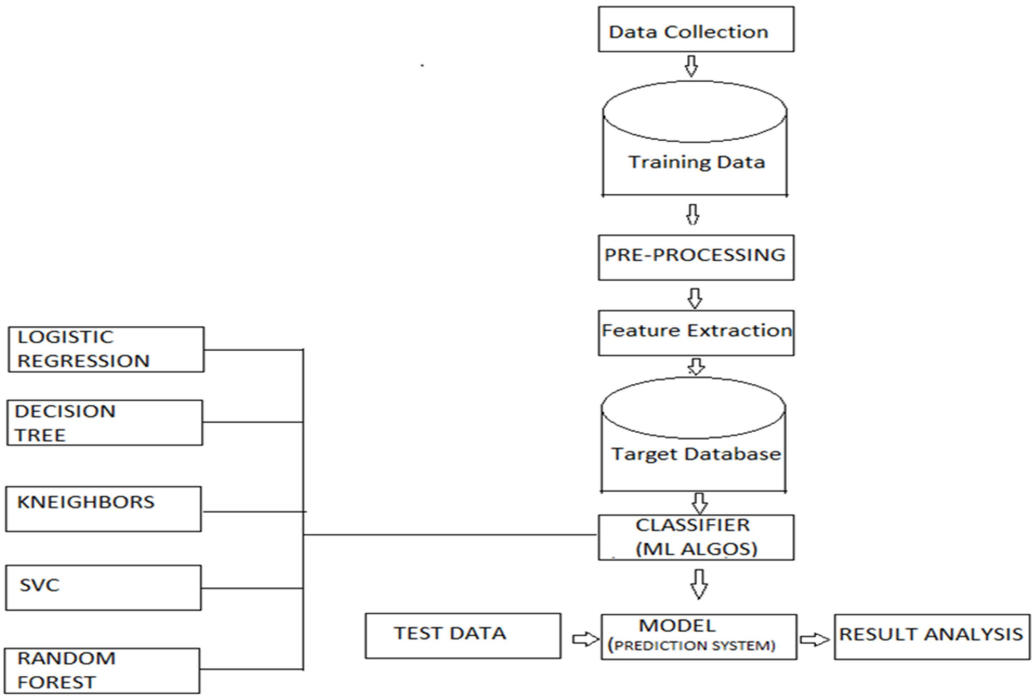


Fig 3.1.1: System Architecture of proposed system

This process requires the processing of plenty of data and studying it on the basis of various parameters available. This prediction varies from each individual because of the huge difference between the data provided by each and their capabilities. All of these can be easily and accurately managed by using Machine learning models and verifying the accuracies by it.

In this project we have used various Machine learning algorithms and techniques through which we are able to make accurate predictions.
Firstly, Data is loaded and then we cleaned and filled the null data provided to us followed by extraction of relationship between various parameters. Then we performed EDA followed by splitting the data and training the various models in order to achieve our objective. Predict the Disaster Survival when the parameters are specified accordingly. Measure the Accuracy Score or the Accuracy Rate of the Machine Learning Algorithm used to predict the data .Use Exploratory Data Analysis and plot at least 7 meaningful graphs and illustrate the scenario and relationship between various parameters involved in that graph. Compare the Accuracy scores of all the used Machine Learning Models and deduce which model has the highest Accuracy.

Random Forest Classifier Algorithm got the highest accuracy score out of all the used algorithms with an accuracy of 97.78% by using all the provided data.
Based on the performance of the four mentioned algorithms; XGBoost, Decision trees, Random Forests. Random Forest proved to be the best algorithm by outperforming other implemented algorithms for the Disaster classification problem since it achieved the highest accuracy.

CHAPTER 5

IMPLEMENTATION

The following illustrates the steps for calculating the accuracy by employing this learning model.

Step 1: Initially read the given dataset

Step 2: Filter out the columns such as Sex, Name, Pclass, Fare which leads to the survival prediction. Step 3: Preprocessing the data involves the removal of improper data like Cabin, Embarked.

Step 4: The attribute which contributes to the prediction which are null must be filled with the appropriatevalues using median such as age.

Step 5: Parse the categorical value to the integer type.

Step 6: Split the data

Step 7: Select the model.

Step 8: Train the model.

Step 9: Make predictions for the given training elements.

Step 10: Finally check the accuracy. For our dataset the accuracy obtained is 97%.

Module 1: Data Collection

Given Dataset consists of various information of Disaster Survivors like Name, age, fare, ticket, cabin etc. The first step is data collection process which can be obtained from many sources and then import the required modules.

Module 2: Data Pre-Processing and Feature Extraction

In this project we have used various Machine learning algorithms and techniques. Firstly, we cleaned and filled the null data provided to us followed by extraction of relationship between various parameters.

Data cleaning is one of the processes that increases prediction performance so it is important to set the data ready for further processes. Clean the given dataset for any unwanted values and pass the dataset through a specific Machine Learning Algorithm. This also involves the conversion of categorical columns into numeric values for more accuracy.

Module 3: Exploratory Data Analysis

Exploratory Data Analysis which depicts the relation between various parameters visually through Graphs.

Perform Exploratory Data Analysis with the use of various plots, graphs, heatmaps etc.

It illustrates the scenario and relationship between various parameters involved in that graph.

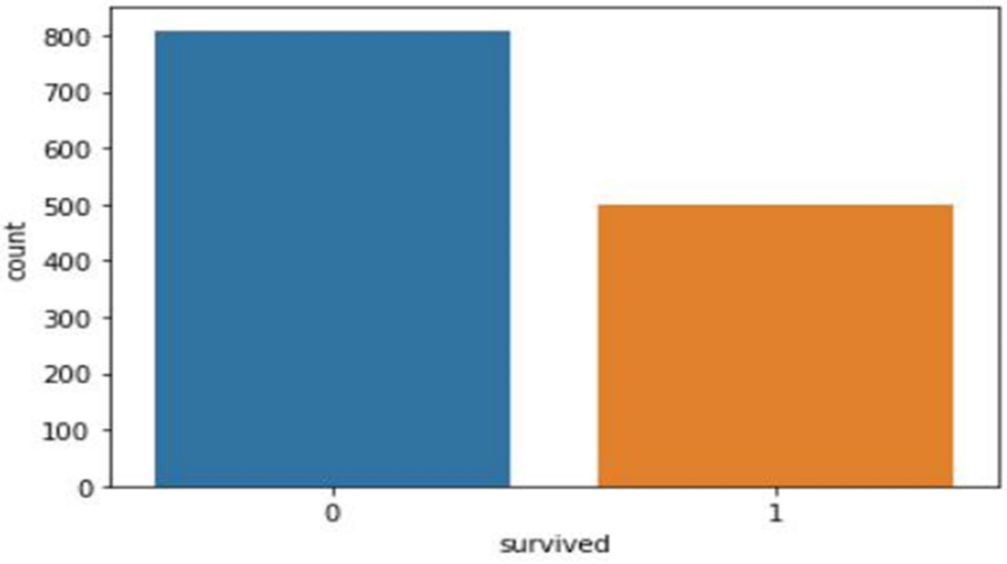


Fig 3.1.2: Plot between Survived and count

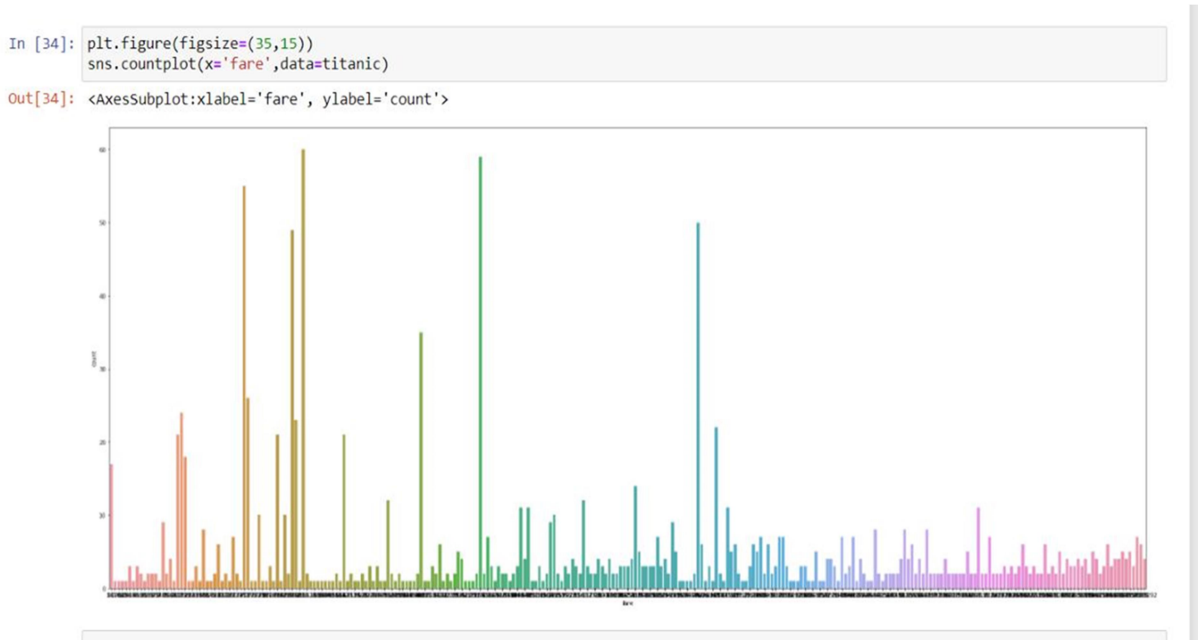


Fig 3.1.3: Graph

Module 4: Classifier Algorithm

Various machine learning models used are:

- LogisticRegression
- DecisionTreeClassifier
- KNeighborsClassifier
- SVC
- RandomForestClassifier
- XGBClassifier

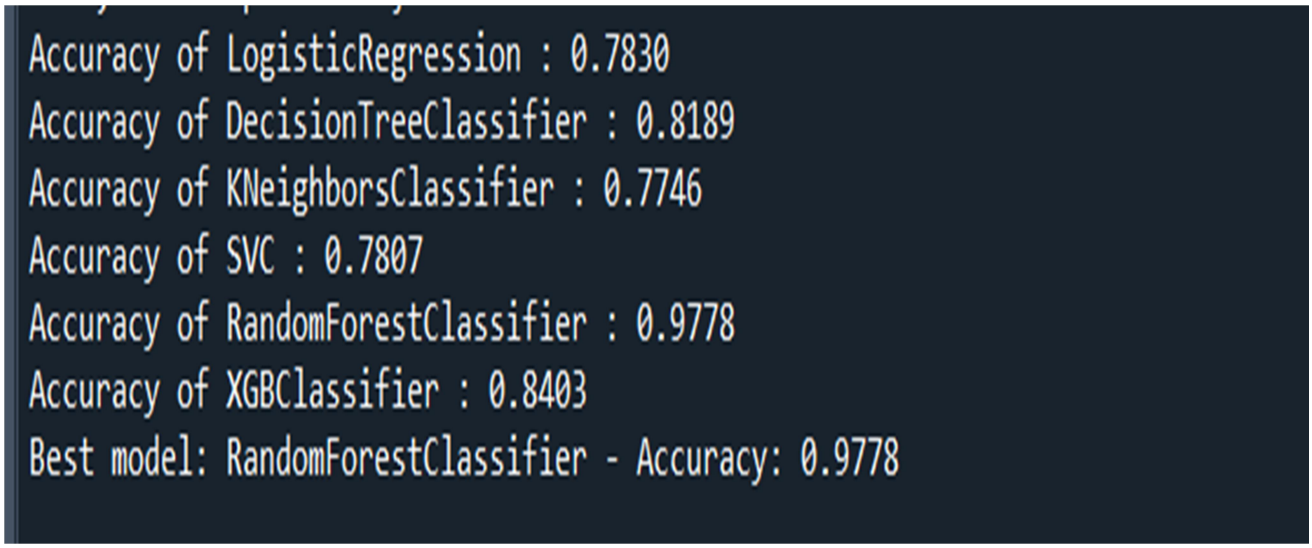


Fig 3.1.4: Results

Module 5: Result and Discussion

Split and train the data to get fit into the models by developing the useful Classification models. Repeat the process with few more Machine Learning Algorithms and note down the corresponding accuracy value of each model simultaneously**.** Compare the Accuracy scores of all the used Machine Learning Models and deduce which model has the highest Accuracy.

It is concluded that, Random Forest Classifier Algorithm got the highest accuracy score out of all the used algorithms with an accuracy of 97.78% by using all the provided data.

In this section, the analysis is done for the following categories: Gender by Survival, Age

group by Survival, Survival and Fare relationship, Passenger class by Survival, Survival rate of Gender based on Passenger class.

Gender by Survival- Here the statistical analysis is performed between the gender and survival. The significant result produced by the analysis is t-value is -19.28 and p-value is 0.0, which indicates that the rate of survival

Of women is more compared to men.

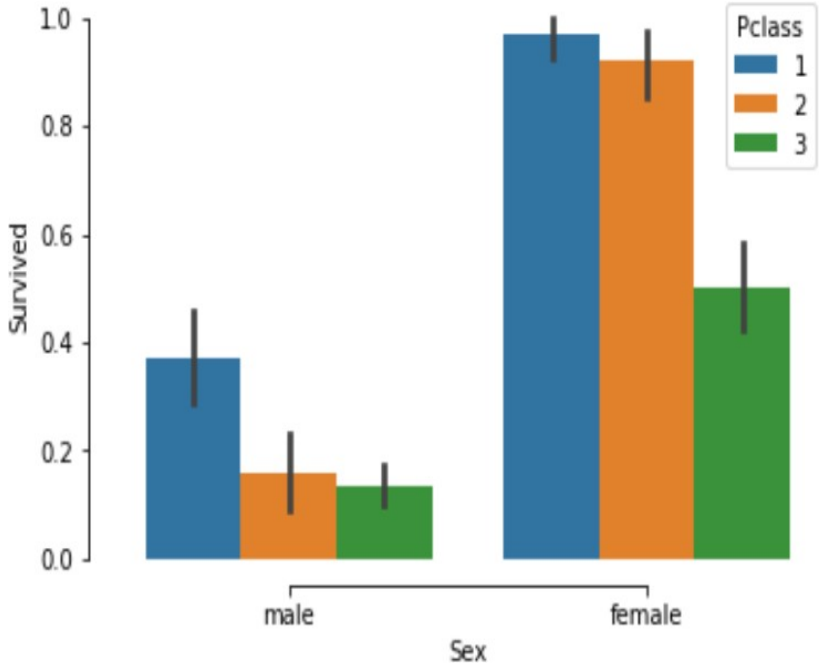


FIG.4 . Survival based on the gender

Age group by Survival- Here the statistical analysis is performed between the age group and survival. The t-test is conducted to calculate the mean age difference between the survival and non-survival. The significant result produced by the analysis is for t-value is -2.067 and p-value is 0.039, which indicates that the average age of no survivor is more compared to survivals.

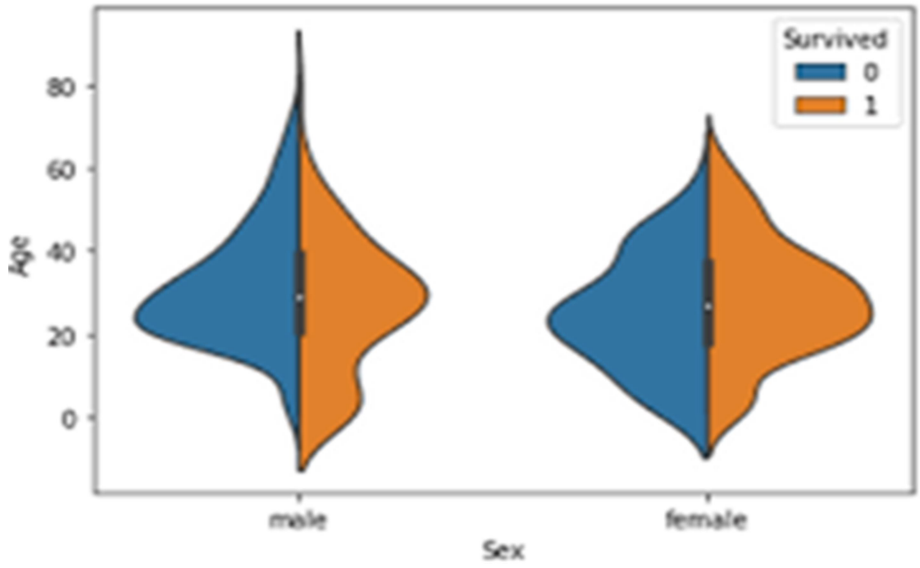


FIG.5 Survival based on the age group

Survival and Fare relationship- A t-test is directed to analyze the distinction in the mean of charge between non- survivors and survivors. The significant result produced by the analysis for t-value is -7.939 and p value is 0.000, which indicates that the amount paid by the survivor is greater compared to non-survivor.

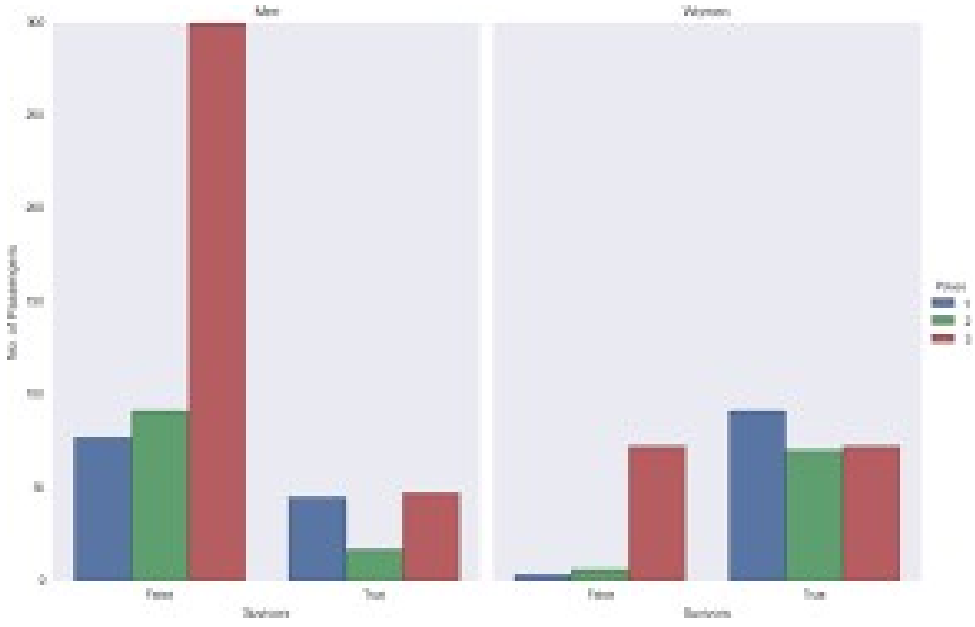


FIG.6 Survival based on the fare

Passenger class by Survival- A test is performed to check the survival rate based on the type of the class.

	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
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

Survival rate of Gender based on P class- A test is performed to check the survival rate of gender present in the different passenger classes.

The above Table IV depicts that the survival rate of female in 1st class is 96% whereas male is 36%. But, it is also observed that the chance of survival of females who belong to the 3rd class is 50% which is less than the 1st class females.

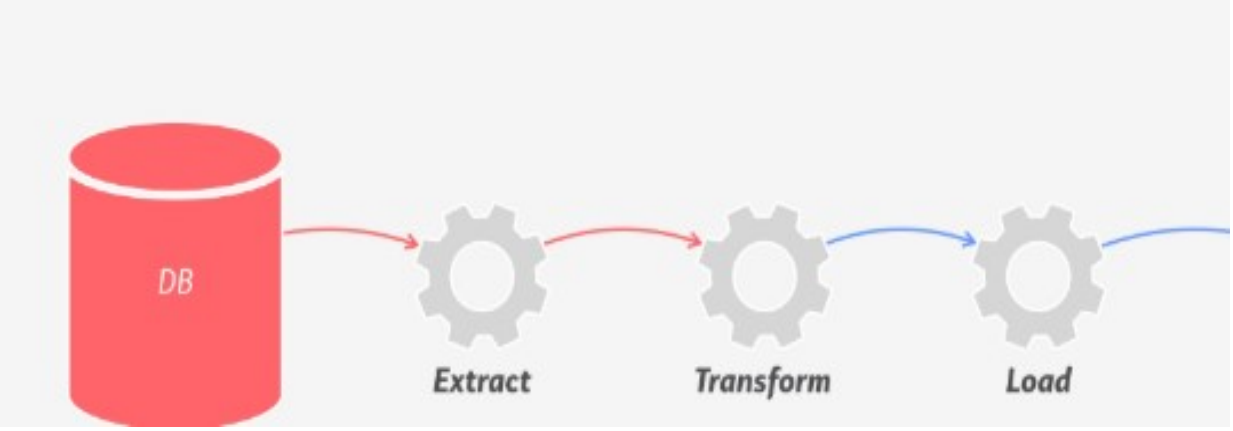
Data Overview

We will be analyzing real messages that were sent during disaster events. Let's look at the data description:

1. **messages.csv:** Contains the id, message that was sent and genre i.e the method (direct, tweet..) the message was sent.
2. **categories.csv:** Contains the id and the categories (related, offer, medical assistance..) the message belonged to.

ETL Pipeline

So, in this part, we will merge two datasets, **messages.csv, and categories.csv** on the common id column. The category column in the id **categories.csv** is in a string format so we need to create columns for each category. Then we will remove duplicates and load the dataset the transformed data into the database hosted using the SQLAlchemy library.



Flask Application

We will create a **train_classifer.py** to create functions that will transform, load, build, and save the model. Basically, we will use the ETL pipeline and ML pipeline. We will create a folder named app that will contain a master.html file which will be the front end and run.py that will run behind to perform computation.

The categories are of the form:

id	categories
2	related-1;request-0;offer-0;aid_related
7	related-1;request-0;offer-0;aid_related
8	related-1;request-0;offer-0;aid_related
9	related-1;request-1;offer-0;aid_related

And we finally load the transformed data into the database: **disaster.db**

```
engine = create_engine('sqlite:///disaster.db')
df.to_sql('disaster_response', engine, index=False)
```

ML Pipeline

Here, we will load the dataset from the disaster.db database. Our main task is to convert the messages into tokens so that they can be interpreted. So, we create a function that will remove punctuations, tokenize words, remove stop words, and perform lemmatization.

So, this is what our function will do:

These words make sense but they cannot be understood by the ML model. So, we will use countVectorizer and tfidf transformer to transform the tokens into features(integers) and we use a simple Random Forest Classifier to fit the training data.

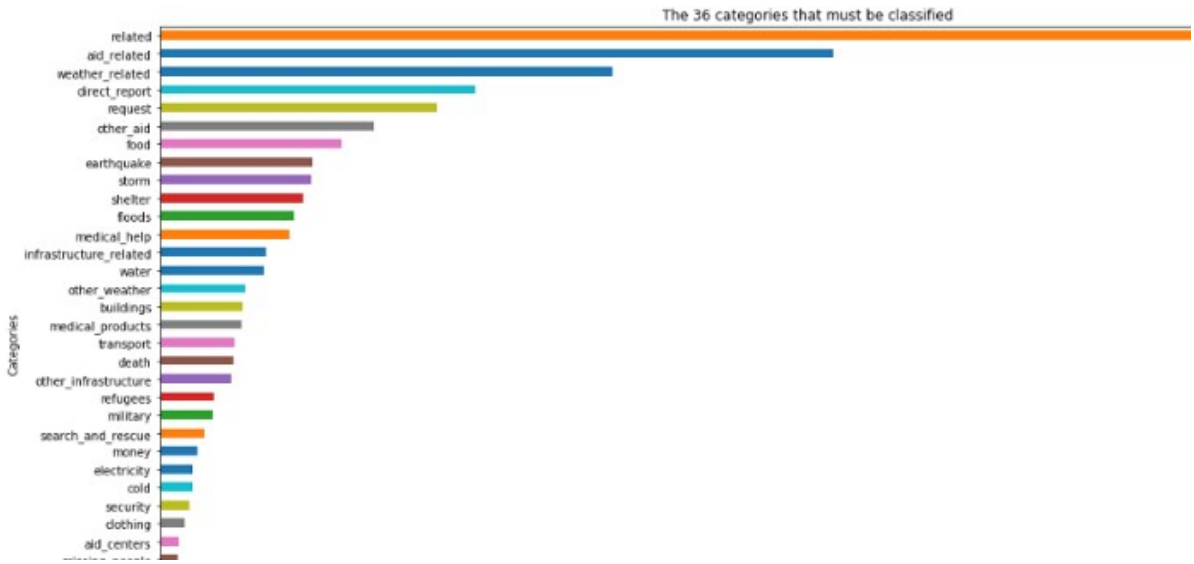
```
text='Weather update - a cold front from Cub

tokenize(text)
```

These words make sense but they cannot be understood by the ML model. So, we will use countVectorizer and tfidf transformer to transform the tokens into features(integers) and we use a simple Random Forest Classifier to fit the training data.

```
pipeline = Pipeline([('vect', CountVectorizer(tokenizer = tokenize)) ,
('tfidf',TfidfTransformer()),

('clf', MultiOutputClassifier(RandomForestClassifier()))])

pipeline.fit(X_train, Y_train)
```

For evaluating our model, we will be using the F-1 score as both False Negatives and False positives are important to us i.e. if we fail to predict the right category of the message then we won't be able to provide right assistance and if we wrongly predict the category of the message we will be wasting our time.

The Random Forest classifier gives us an F-1 score of **0.44.** The main reason behind the low score is that the categories are highly imbalanced. The distribution of the categories is as follows:



Let's improve the model using some different ML model and hyperparameter tuning. So, after doing a GridSearchCV to find the best parameter of the Random Forest model we were able to increase the F-1 score to **0.51.** Next, we train AdaBoost classifier and we were able to improve the F-1 score to **0.59**

```
#https://medium.com/swlh/the-hyperparameter-cheat-sheet-
770f1fed32ff
pipeline_ada = Pipeline([
    ('vect', CountVectorizer(tokenizer=tokenize)),
    ('tfidf', TfidfTransformer()),
    ('clf', MultiOutputClassifier(

AdaBoostClassifier(base_estimator=DecisionTreeClassifier(max_dep
th=1, class_weight='balanced'))
    ))
])parameters_ada = {
    'clf__estimator__learning_rate': [0.1, 0.3],
    'clf__estimator__n_estimators': [100, 200]
}cv_ada = GridSearchCV(estimator=pipeline_ada,
param_grid=parameters_ada, cv=3, scoring='f1_weighted',
verbose=3)
```

We save this model as a pickle file so that we do not need to train it again.

```
pickle.dump(cv_ada, open('disaster_ada_model.sav', 'wb'))
```

Flask Application

We will create a **train_classifer.py** to create functions that will transform, load, build, and save the model. Basically, we will use the ETL pipeline and ML pipeline. We will create a folder named app that will contain a master.html file which will be the front end and run.py that will run behind to perform computation.

Conclusion

Build a full-stack multi-output ML web application to classify messages sent during disasters into different categories and provide quick assistance from different disaster relief organizations

CHAPTER 6

TESTING AND RESULT

6.1 TESTING

Testing in general means the validation and verification. It shows that the system conforms to its specifications and the system meets all the expectations of the user.

6.1.1 IMPORT NECESSARY LIBRARIES

First off, we need to import several Python libraries such as numpy, pandas, matplotlib and seaborn.

In [1]:
```
#data analysis libraries
import numpy as np
import pandas as pd

#visualization libraries
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

#ignore warnings
import warnings
warnings.filterwarnings('ignore')
```

6.1.2 READ IN AND EXPLORE THE DATA

It's time to read in our training and testing data using pd.read_csv, and take a first look at the training data using the describe() function.

In [2]:
```
#import train and test CSV files
train = pd.read_csv("../input/train.csv")
test = pd.read_csv("../input/test.csv")

#take a look at the training data
train.describe(include="all")
```

Out[2]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cab
count	891.000000	891.000000	891.000000	891	891	714.000000	891.000000	891.000000	891	891.000000	204
unique	NaN	NaN	NaN	891	2	NaN	NaN	NaN	681	NaN	147
top	NaN	NaN	NaN	Carlsson, Mr. August Sigfrid	male	NaN	NaN	NaN	1601	NaN	G6
freq	NaN	NaN	NaN	1	577	NaN	NaN	NaN	7	NaN	4
mean	446.000000	0.383838	2.308642	NaN	NaN	29.699118	0.523008	0.381594	NaN	32.204208	NaN
std	257.353842	0.486592	0.836071	NaN	NaN	14.526497	1.102743	0.806057	NaN	49.693429	NaN
min	1.000000	0.000000	1.000000	NaN	NaN	0.420000	0.000000	0.000000	NaN	0.000000	NaN
25%	223.500000	0.000000	2.000000	NaN	NaN	20.125000	0.000000	0.000000	NaN	7.910400	NaN
50%	446.000000	0.000000	3.000000	NaN	NaN	28.000000	0.000000	0.000000	NaN	14.454200	NaN
75%	668.500000	1.000000	3.000000	NaN	NaN	38.000000	1.000000	0.000000	NaN	31.000000	NaN
max	891.000000	1.000000	3.000000	NaN	NaN	80.000000	8.000000	6.000000	NaN	512.329200	NaN

7.1.3 DATA ANALYSIS

We're going to consider the features in the dataset and how complete they are.

In [3]:
```
#get a list of the features within the dataset
print(train.columns)
```

```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

In [4]:
```
#see a sample of the dataset to get an idea of the variables
train.sample(5)
```

Out[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
572	573	1	1	Flynn, Mr. John Irwin ("Irving")	male	36.0	0	0	PC 17474	26.3875	E25	S
717	718	1	2	Troutt, Miss. Edwina Celia "Winnie"	female	27.0	0	0	34218	10.5000	E101	S
141	142	1	3	Nysten, Miss. Anna Sofia	female	22.0	0	0	347081	7.7500	NaN	S
226	227	1	2	Mellors, Mr. William John	male	19.0	0	0	SW/PP 751	10.5000	NaN	S
648	649	0	3	Willey, Mr. Edward	male	NaN	0	0	S.O./P.P. 751	7.5500	NaN	S

- **Numerical Features:** Age (Continuous), Fare (Continuous), SibSp (Discrete), Parch (Discrete)
- **Categorical Features:** Survived, Sex, Embarked, Pclass
- **Alphanumeric Features:** Ticket, Cabin

What are the data types for each feature?

- Survived: int
- Pclass: int
- Name: string
- Sex: string
- Age: float
- SibSp: int
- Parch: int
- Ticket: string
- Fare: float
- Cabin: string
- Embarked: string

Now that we have an idea of what kinds of features we're working with, we can see how much information we have about each of them.

In [5]:
```
#see a summary of the training dataset
train.describe(include = "all")
```

Out[5]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cab
count	891.000000	891.000000	891.000000	891	891	714.000000	891.000000	891.000000	891	891.000000	204
unique	NaN	NaN	NaN	891	2	NaN	NaN	NaN	681	NaN	147
top	NaN	NaN	NaN	Carlsson, Mr. August Sigfrid	male	NaN	NaN	NaN	1601	NaN	G6
freq	NaN	NaN	NaN	1	577	NaN	NaN	NaN	7	NaN	4
mean	446.000000	0.383838	2.308642	NaN	NaN	29.699118	0.523008	0.381594	NaN	32.204208	NaN
std	257.353842	0.486592	0.836071	NaN	NaN	14.526497	1.102743	0.806057	NaN	49.693429	NaN
min	1.000000	0.000000	1.000000	NaN	NaN	0.420000	0.000000	0.000000	NaN	0.000000	NaN
25%	223.500000	0.000000	2.000000	NaN	NaN	20.125000	0.000000	0.000000	NaN	7.910400	NaN
50%	446.000000	0.000000	3.000000	NaN	NaN	28.000000	0.000000	0.000000	NaN	14.454200	NaN
75%	668.500000	1.000000	3.000000	NaN	NaN	38.000000	1.000000	0.000000	NaN	31.000000	NaN
max	891.000000	1.000000	3.000000	NaN	NaN	80.000000	8.000000	6.000000	NaN	512.329200	NaN

Some Observations:

- There are a total of 891 passengers in our training set.
- The Age feature is missing approximately 19.8% of its values. I'm guessing that the Age feature is pretty important to survival, so we should probably attempt to fill these gaps.
- The Cabin feature is missing approximately 77.1% of its values. Since so much of the feature is missing, it would be hard to fill in the missing values. We'll probably drop these values from our dataset.
- The Embarked feature is missing 0.22% of its values, which should be relatively harmless.

In [6]:
```
#check for any other unusable values
print(pd.isnull(train).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 can see that except for the abovementioned missing values, no NaN values exist.

Some Predictions:

- Sex: Females are more likely to survive.
- SibSp/Parch: People traveling alone are more likely to survive.
- Age: Young children are more likely to survive.

- Pclass: People of higher socioeconomic class are more likely to survive.

7.1.4 DATA VISUALIZATION

It's time to visualize our data so we can see whether our predictions were accurate!

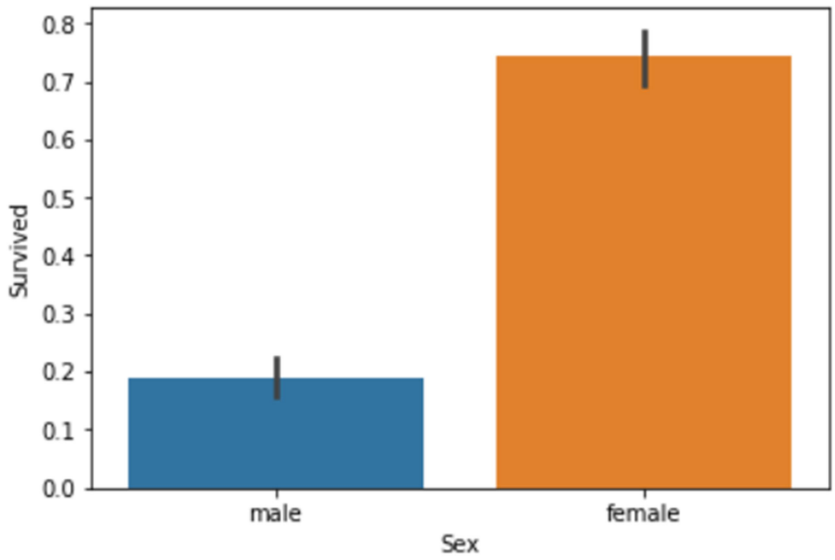
Sex Feature

In [7]:

```
#draw a bar plot of survival by sex
sns.barplot(x="Sex", y="Survived", data=train)

#print percentages of females vs. males that survive
print("Percentage of females who survived:", train["Survived"][train["Sex"] == 'female'].value_counts(nor
malize = True)[1]*100)

print("Percentage of males who survived:", train["Survived"][train["Sex"] == 'male'].value_counts(normali
ze = True)[1]*100)
```

Percentage of females who survived: 74.20382165605095

Percentage of males who survived: 18.890814558058924



As predicted, females have a much higher chance of survival than males. The Sex feature is essential in our predictions.

linkcode

Pclass Feature

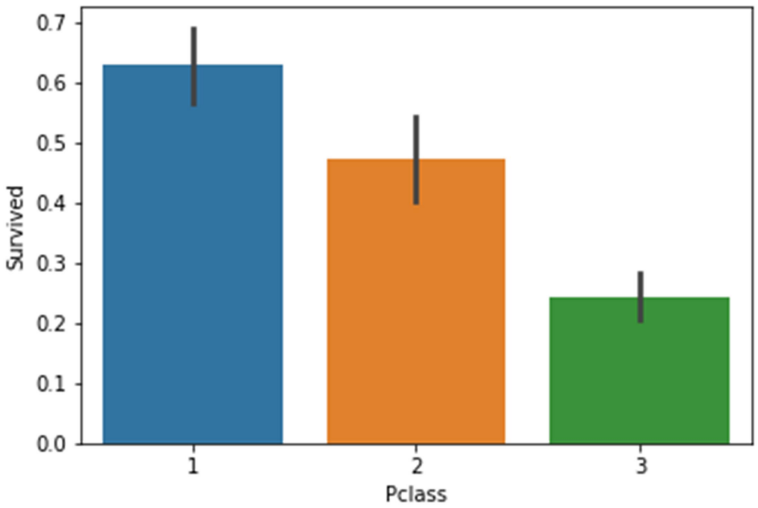
In [8]:

```
#draw a bar plot of survival by Pclass
sns.barplot(x="Pclass", y="Survived", data=train)

#print percentage of people by Pclass that survived
print("Percentage of Pclass = 1 who survived:", train["Survived"][train["Pclass"] == 1].value_counts(norm
alize = True)[1]*100)

print("Percentage of Pclass = 2 who survived:", train["Survived"][train["Pclass"] == 2].value_counts(norm
alize = True)[1]*100)

print("Percentage of Pclass = 3 who survived:", train["Survived"][train["Pclass"] == 3].value_counts(norm
alize = True)[1]*100)
```

Percentage of Pclass = 1 who survived: 62.96296296296296

Percentage of Pclass = 2 who survived: 47.28260869565217

Percentage of Pclass = 3 who survived: 24.236252545824847



As predicted, people with higher socioeconomic class had a higher rate of survival. (62.9% vs. 47.3% vs. 24.2%)

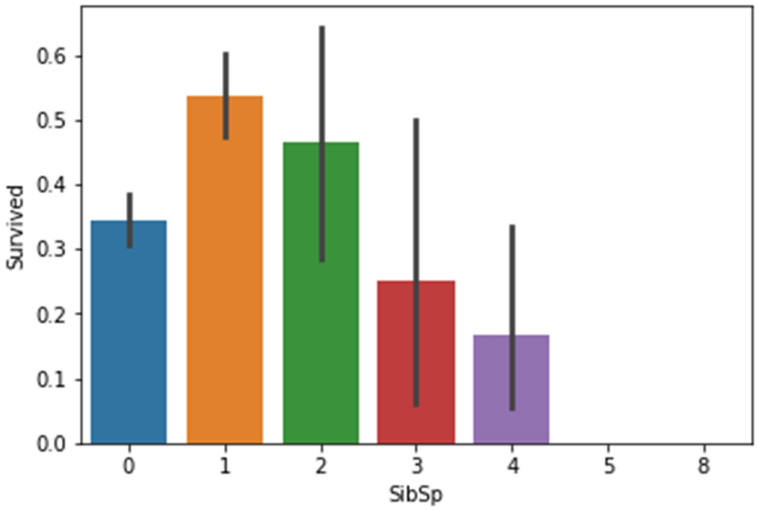
SibSp Feature

In [9]:
```
#draw a bar plot for SibSp vs. survival
sns.barplot(x="SibSp", y="Survived", data=train)

#I won't be printing individual percent values for all of these.
print("Percentage of SibSp = 0 who survived:", train["Survived"][train["SibSp"] == 0].value_counts(normal
ize = True)[1]*100)

print("Percentage of SibSp = 1 who survived:", train["Survived"][train["SibSp"] == 1].value_counts(normal
ize = True)[1]*100)

print("Percentage of SibSp = 2 who survived:", train["Survived"][train["SibSp"] == 2].value_counts(normal
ize = True)[1]*100)
```

Percentage of SibSp = 0 who survived: 34.53947368421053

Percentage of SibSp = 1 who survived: 53.588516746411486

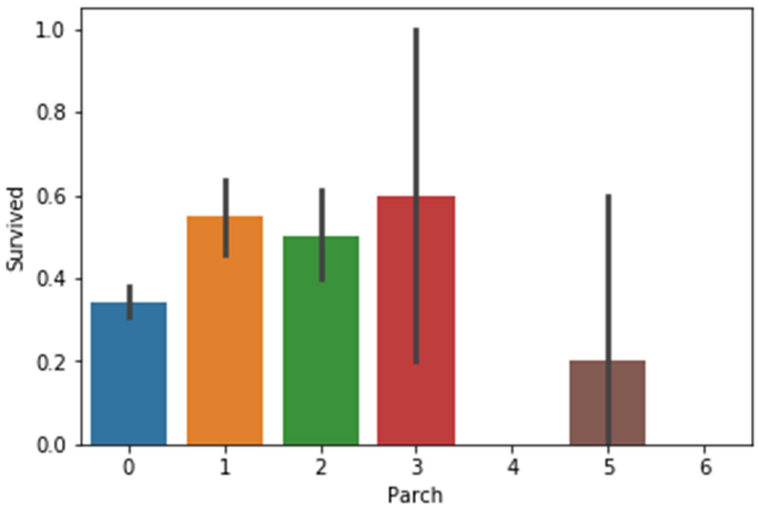
Percentage of SibSp = 2 who survived: 46.42857142857143



In general, it's clear that people with more siblings or spouses aboard were less likely to survive. However, contrary to expectations, people with no siblings or spouses were less to likely to survive than those with one or two. (34.5% vs 53.4% vs. 46.4%)

Parch Feature

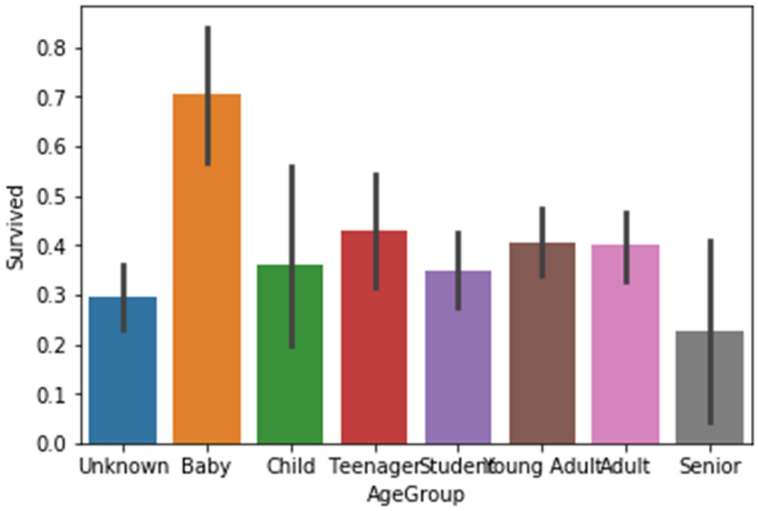
In [10]:
```
#draw a bar plot for Parch vs. survival
sns.barplot(x="Parch", y="Survived", data=train)
plt.show()
```



People with less than four parents or children aboard are more likely to survive than those with four or more. Again, people traveling alone are less likely to survive than those with 1-3 parents or children.

Age Feature

In [11]:
```
#sort the ages into logical categories
train["Age"] = train["Age"].fillna(-0.5)
test["Age"] = test["Age"].fillna(-0.5)
bins = [-1, 0, 5, 12, 18, 24, 35, 60, np.inf]
labels = ['Unknown', 'Baby', 'Child', 'Teenager', 'Student', 'Young Adult', 'Adult', 'Senior']
train['AgeGroup'] = pd.cut(train["Age"], bins, labels = labels)
test['AgeGroup'] = pd.cut(test["Age"], bins, labels = labels)

#draw a bar plot of Age vs. survival
sns.barplot(x="AgeGroup", y="Survived", data=train)
plt.show()
```



Babies are more likely to survive than any other age group.

Cabin Feature

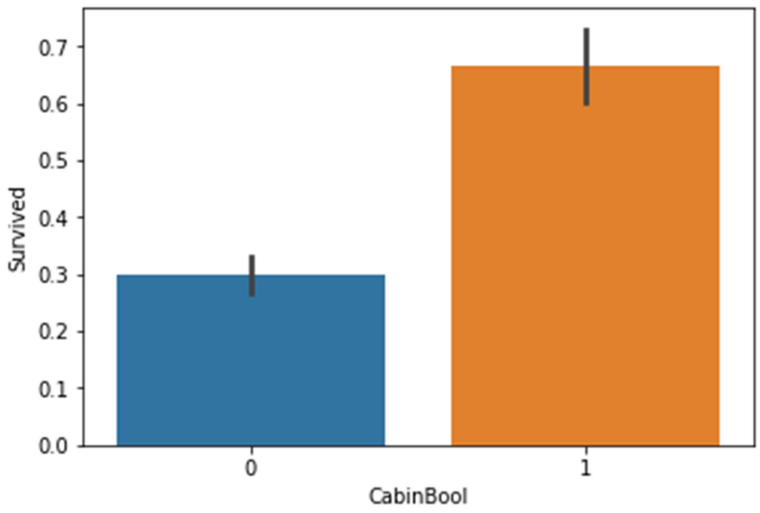
I think the idea here is that people with recorded cabin numbers are of higher socioeconomic class, and thus more likely to survive.

In [12]:

```
train["CabinBool"] = (train["Cabin"].notnull().astype('int'))
test["CabinBool"] = (test["Cabin"].notnull().astype('int'))

#calculate percentages of CabinBool vs. survived
print("Percentage of CabinBool = 1 who survived:", train["Survived"][train["CabinBool"] == 1].value_count
s(normalize = True)[1]*100)

print("Percentage of CabinBool = 0 who survived:", train["Survived"][train["CabinBool"] == 0].value_count
s(normalize = True)[1]*100)
#draw a bar plot of CabinBool vs. survival
sns.barplot(x="CabinBool", y="Survived", data=train)
plt.show()
```

Percentage of CabinBool = 1 who survived: 66.66666666666666

Percentage of CabinBool = 0 who survived: 29.985443959243085



People with a recorded Cabin number are, in fact, more likely to survive. (66.6% vs 29.9%)

7.1.5 CLEANING DATA

In [13]:
```
test.describe(include="all")
```

Out[13]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarke
count	418.000000	418.000000	418	418	418.000000	418.000000	418.000000	418	417.000000	91	418
unique	NaN	NaN	418	2	NaN	NaN	NaN	363	NaN	76	3
top	NaN	NaN	Davidson, Mrs. Thornton (Orian Hays)	male	NaN	NaN	NaN	PC 17608	NaN	B57 B59 B63 B66	S
freq	NaN	NaN	1	266	NaN	NaN	NaN	5	NaN	3	270
mean	1100.500000	2.265550	NaN	NaN	23.941388	0.447368	0.392344	NaN	35.627188	NaN	NaN
std	120.810458	0.841838	NaN	NaN	17.741080	0.896760	0.981429	NaN	55.907576	NaN	NaN
min	892.000000	1.000000	NaN	NaN	-0.500000	0.000000	0.000000	NaN	0.000000	NaN	NaN
25%	996.250000	1.000000	NaN	NaN	9.000000	0.000000	0.000000	NaN	7.895800	NaN	NaN
50%	1100.500000	3.000000	NaN	NaN	24.000000	0.000000	0.000000	NaN	14.454200	NaN	NaN
75%	1204.750000	3.000000	NaN	NaN	35.750000	1.000000	0.000000	NaN	31.500000	NaN	NaN
max	1309.000000	3.000000	NaN	NaN	76.000000	8.000000	9.000000	NaN	512.329200	NaN	NaN

- We have a total of 418 passengers.
- 1 value from the Fare feature is missing.
- Around 20.5% of the Age feature is missing, we will need to fill that in.

Cabin Feature

In [14]:
```
#we'll start off by dropping the Cabin feature since not a lot more useful information can be extracted fro
m it.
train = train.drop(['Cabin'], axis = 1)
test = test.drop(['Cabin'], axis = 1)
```

Ticket Feature

In [15]:
```
#we can also drop the Ticket feature since it's unlikely to yield any useful information
train = train.drop(['Ticket'], axis = 1)
test = test.drop(['Ticket'], axis = 1)
```

Embarked Feature

In [16]:
```
#now we need to fill in the missing values in the Embarked feature
print("Number of people embarking in Southampton (S):")
southampton = train[train["Embarked"] == "S"].shape[0]
print(southampton)

print("Number of people embarking in Cherbourg (C):")
cherbourg = train[train["Embarked"] == "C"].shape[0]
print(cherbourg)

print("Number of people embarking in Queenstown (Q):")
queenstown = train[train["Embarked"] == "Q"].shape[0]
print(queenstown)
```

Number of people embarking in Southampton (S):

644

Number of people embarking in Cherbourg (C):

168

Number of people embarking in Queenstown (Q):

77

It's clear that the majority of people embarked in Southampton (S). Let's go ahead and fill in the missing values with S.

In [17]:
```
#replacing the missing values in the Embarked feature with S
train = train.fillna({"Embarked": "S"})
```

Age Feature

Next we'll fill in the missing values in the Age feature. Since a higher percentage of values are missing, it would be illogical to fill all of them with the same value (as we did with Embarked). Instead, let's try to find a way to predict the missing ages.

In [18]:
```
#create a combined group of both datasets
combine = [train, test]

#extract a title for each Name in the train and test datasets
for dataset in combine:
    dataset['Title'] = dataset.Name.str.extract(' ([A-Za-z]+)\.', expand=False)

pd.crosstab(train['Title'], train['Sex'])
```

Out[18]:

Sex	female	male
Title		
Capt	0	1
Col	0	2
Countess	1	0
Don	0	1
Dr	1	6
Jonkheer	0	1
Lady	1	0
Major	0	2
Master	0	40
Miss	182	0
Mlle	2	0
Mme	1	0
Mr	0	517
Mrs	125	0
Ms	1	0
Rev	0	6
Sir	0	1

In [19]:

```
#replace various titles with more common names
for dataset in combine:
    dataset['Title'] = dataset['Title'].replace(['Lady', 'Capt', 'Col',
    'Don', 'Dr', 'Major', 'Rev', 'Jonkheer', 'Dona'], 'Rare')

    dataset['Title'] = dataset['Title'].replace(['Countess', 'Lady', 'Sir'], 'Royal')
    dataset['Title'] = dataset['Title'].replace('Mlle', 'Miss')
    dataset['Title'] = dataset['Title'].replace('Ms', 'Miss')
    dataset['Title'] = dataset['Title'].replace('Mme', 'Mrs')

train[['Title', 'Survived']].groupby(['Title'], as_index=False).mean()
```

Out[19]:

	Title	Survived
0	Master	0.575000
1	Miss	0.702703
2	Mr	0.156673
3	Mrs	0.793651
4	Rare	0.285714
5	Royal	1.000000

In [20]:

```
#map each of the title groups to a numerical value
title_mapping = {"Mr": 1, "Miss": 2, "Mrs": 3, "Master": 4, "Royal": 5, "Rare": 6}
for dataset in combine:
    dataset['Title'] = dataset['Title'].map(title_mapping)
    dataset['Title'] = dataset['Title'].fillna(0)

train.head()
```

Out[20]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Embarked	AgeGroup	CabinBool	Title
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	S	Student	0	1
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C	Adult	1	3
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	S	Young Adult	0	2
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	S	Young Adult	1	3
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	S	Young Adult	0	1

In [22]:
```
#map each Age value to a numerical value
age_mapping = {'Baby': 1, 'Child': 2, 'Teenager': 3, 'Student': 4, 'Young Adult': 5, 'Adult': 6, 'Senio
r': 7}
train['AgeGroup'] = train['AgeGroup'].map(age_mapping)
test['AgeGroup'] = test['AgeGroup'].map(age_mapping)

train.head()

#dropping the Age feature for now, might change
train = train.drop(['Age'], axis = 1)
test = test.drop(['Age'], axis = 1)
```

Name Feature

We can drop the name feature now that we've extracted the titles.

In [23]:
```
#drop the name feature since it contains no more useful information.
train = train.drop(['Name'], axis = 1)
test = test.drop(['Name'], axis = 1)
```

Sex Feature

In [24]:
```
#map each Sex value to a numerical value
sex_mapping = {"male": 0, "female": 1}
train['Sex'] = train['Sex'].map(sex_mapping)
test['Sex'] = test['Sex'].map(sex_mapping)

train.head()
```

Out[24]:

	PassengerId	Survived	Pclass	Sex	SibSp	Parch	Fare	Embarked	AgeGroup	CabinBool	Title
0	1	0	3	0	1	0	7.2500	S	4	0	1
1	2	1	1	1	1	0	71.2833	C	6	1	3
2	3	1	3	1	0	0	7.9250	S	5	0	2
3	4	1	1	1	1	0	53.1000	S	5	1	3
4	5	0	3	0	0	0	8.0500	S	5	0	1

Embarked Feature

In [25]:
```
#map each Embarked value to a numerical value
embarked_mapping = {"S": 1, "C": 2, "Q": 3}
train['Embarked'] = train['Embarked'].map(embarked_mapping)
test['Embarked'] = test['Embarked'].map(embarked_mapping)

train.head()
```

Out[25]:

	PassengerId	Survived	Pclass	Sex	SibSp	Parch	Fare	Embarked	AgeGroup	CabinBool	Title
0	1	0	3	0	1	0	7.2500	1	4	0	1
1	2	1	1	1	1	0	71.2833	2	6	1	3
2	3	1	3	1	0	0	7.9250	1	5	0	2
3	4	1	1	1	1	0	53.1000	1	5	1	3
4	5	0	3	0	0	0	8.0500	1	5	0	1

Fare Feature

It's time separate the fare values into some logical groups as well as filling in the single missing value in the test dataset.

In [26]:
```
#fill in missing Fare value in test set based on mean fare for that Pclass
for x in range(len(test["Fare"])):
    if pd.isnull(test["Fare"][x]):
        pclass = test["Pclass"][x] #Pclass = 3
        test["Fare"][x] = round(train[train["Pclass"] == pclass]["Fare"].mean(), 4)

#map Fare values into groups of numerical values
train['FareBand'] = pd.qcut(train['Fare'], 4, labels = [1, 2, 3, 4])
test['FareBand'] = pd.qcut(test['Fare'], 4, labels = [1, 2, 3, 4])

#drop Fare values
train = train.drop(['Fare'], axis = 1)
test = test.drop(['Fare'], axis = 1)
```

In [27]:
```
#check train data
train.head()
```

Out[27]:

	PassengerId	Survived	Pclass	Sex	SibSp	Parch	Embarked	AgeGroup	CabinBool	Title	FareBand
0	1	0	3	0	1	0	1	4	0	1	1
1	2	1	1	1	1	0	2	6	1	3	4
2	3	1	3	1	0	0	1	5	0	2	2
3	4	1	1	1	1	0	1	5	1	3	4
4	5	0	3	0	0	0	1	5	0	1	2

In [28]:

```
#check test data
test.head()
```

Out[28]:

	PassengerId	Pclass	Sex	SibSp	Parch	Embarked	AgeGroup	CabinBool	Title	FareBand
0	892	3	0	0	0	3	5	0	1	1
1	893	3	1	1	0	1	6	0	3	1
2	894	2	0	0	0	3	7	0	1	2
3	895	3	0	0	0	1	5	0	1	2
4	896	3	1	1	1	1	4	0	3	2

7.1.5 CHOOSING THE BEST MODEL

Splitting the Training Data

We will use part of our training data (22% in this case) to test the accuracy of our different models.

In [29]:

```
from sklearn.model_selection import train_test_split

predictors = train.drop(['Survived', 'PassengerId'], axis=1)
target = train["Survived"]
x_train, x_val, y_train, y_val = train_test_split(predictors, target, test_size = 0.22, random_state = 0)
```

Testing Different Models

I will be testing the following models with my training data (got the list from here):

- Gaussian Naive Bayes
- Logistic Regression
- Support Vector Machines
- Perceptron
- Decision Tree Classifier
- Random Forest Classifier
- KNN or k-Nearest Neighbors
- Stochastic Gradient Descent
- Gradient Boosting Classifier

For each model, we set the model, fit it with 80% of our training data, predict for 20% of the training data and check the accuracy.

In [51]:

```
## Print accuracy of the models with best parameters
best_models = [logit_best, tree_best, knn_best,
               svc_best, forest_best, xgb_best]
best_acc = ['model', 0]
for best_model in best_models:
  accuracy = best_model.score(x, y)
  model_name = type(best_model).__name__
  print('Accuracy of', type(best_model).__name__, ': {:.4f}'.
        format(accuracy))
  if accuracy > best_acc[1]:
    best_acc[0] = model_name
    best_acc[1] = accuracy
```

```
Accuracy of LogisticRegression : 0.7830
Accuracy of DecisionTreeClassifier : 0.8189
Accuracy of KNeighborsClassifier : 0.7746
Accuracy of SVC : 0.7807
Accuracy of RandomForestClassifier : 0.9778
Accuracy of XGBClassifier : 0.8403
```

In [52]:

```
## Print the model with the best accuracy
model_name = best_acc[0]
accuracy = best_acc[1]
print('Best model: {} - Accuracy: {:.4f}'.format(model_name, accuracy))
```

```
Best model: RandomForestClassifier - Accuracy: 0.9778
```

Let's compare the accuracies of each model!

In [40]:
```
models = pd.DataFrame({
    'Model': ['Support Vector Machines', 'KNN', 'Logistic Regression',
              'Random Forest', 'Naive Bayes', 'Perceptron', 'Linear SVC',
              'Decision Tree', 'Stochastic Gradient Descent', 'Gradient Boosting Classifier'],
    'Score': [acc_svc, acc_knn, acc_logreg,
              acc_randomforest, acc_gaussian, acc_perceptron,acc_linear_svc, acc_decisiontree,
              acc_sgd, acc_gbk]})
models.sort_values(by='Score', ascending=False)
```

```
Accuracy of LogisticRegression : 0.7830
Accuracy of DecisionTreeClassifier : 0.8189
Accuracy of KNeighborsClassifier : 0.7746
Accuracy of SVC : 0.7807
Accuracy of RandomForestClassifier : 0.9778
Accuracy of XGBClassifier : 0.8403
Best model: RandomForestClassifier - Accuracy: 0.9778
```

I decided to use the Random Forest Classifier model for the testing data.

7.1.6 CREATING SUBMISSION FILE

In [41]:
```
#set ids as PassengerId and predict survival
ids = test['PassengerId']
predictions = gbk.predict(test.drop('PassengerId', axis=1))

#set the output as a dataframe and convert to csv file named submission.csv
output = pd.DataFrame({ 'PassengerId' : ids, 'Survived': predictions })
output.to_csv('submission.csv', index=False)
```

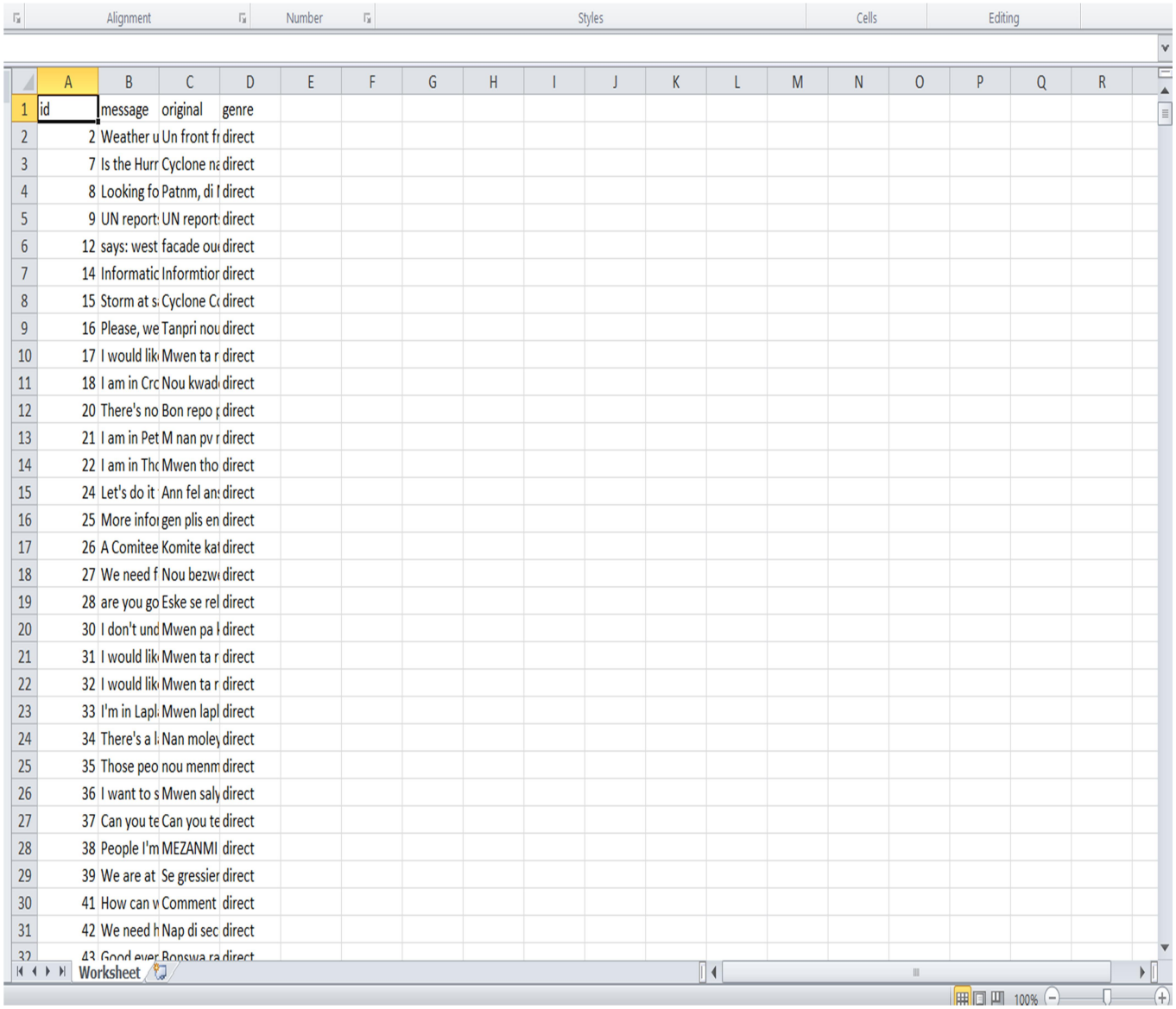
CHAPTER 7

CONCLUSION

It is concluded that, Random Forest Classifier Algorithm got the highest accuracy score out of all the used algorithms with an accuracy of 97.78% by using all the provided data.
Based on the performance of the four mentioned algorithms; XGBoost, Decision trees, Random Forests, in this paper, Random Forest proved to be the best algorithm by outperforming other implemented algorithms for the Disaster classification problem since it achieved the highest accuracy. This implies that the number of correctly classified classes in Random Forest is higher than that of other implemented algorithms. Also, the AUC and boxplots values for Random Forest appear to be the highest as compared with other implemented algorithms. Based on our data, Random Forest appears to be a very good classifier. Future work might consider cross validation. Cross validation could also be used to compute the model's accuracy based on different combinations of training and test samples. In addition, some other classifiers can also be applied.

Future work includes calculating the accuracy of the train set as well as test set using the cross- validation technique. Different approaches of machine learning such as K-NN classification, clustering can also be developed for finding the survival rate.

Snapshots



	A	B	C	D
1	id	message	original	genre
2	2	Weather u	Un front fr	direct
3	7	Is the Hurr	Cyclone na	direct
4	8	Looking fo	Patnm, di l	direct
5	9	UN report	UN report	direct
6	12	says: west	facade ou	direct
7	14	Informatic	Informtior	direct
8	15	Storm at s	Cyclone C	direct
9	16	Please, we	Tanpri nou	direct
10	17	I would lik	Mwen ta r	direct
11	18	I am in Cr	Nou kwad	direct
12	20	There's no	Bon repo p	direct
13	21	I am in Pet	M nan pv r	direct
14	22	I am in Th	Mwen tho	direct
15	24	Let's do it	Ann fel an	direct
16	25	More infor	gen plis en	direct
17	26	A Comitee	Komite kat	direct
18	27	We need f	Nou bezw	direct
19	28	are you go	Eske se rel	direct
20	30	I don't und	Mwen pa l	direct
21	31	I would lik	Mwen ta r	direct
22	32	I would lik	Mwen ta r	direct
23	33	I'm in Lapl	Mwen lapl	direct
24	34	There's a l	Nan moley	direct
25	35	Those peo	nou menm	direct
26	36	I want to s	Mwen saly	direct
27	37	Can you te	Can you te	direct
28	38	People I'm	MEZANMI	direct
29	39	We are at	Se gressier	direct
30	41	How can v	Comment	direct
31	42	We need h	Nap di sec	direct
32	43	Good ever	Bonswa ra	direct

Fig: disaster_messages.csv

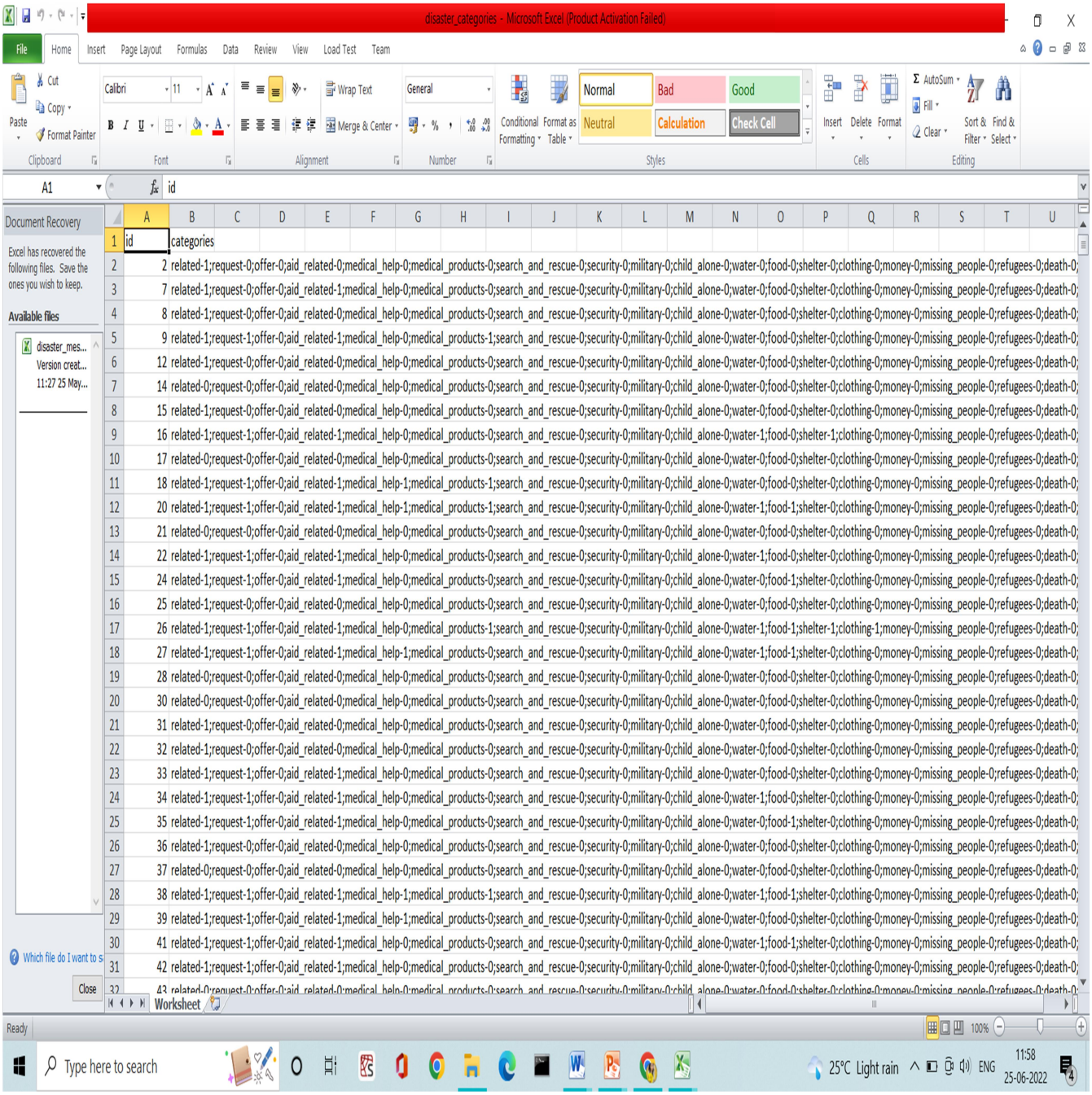
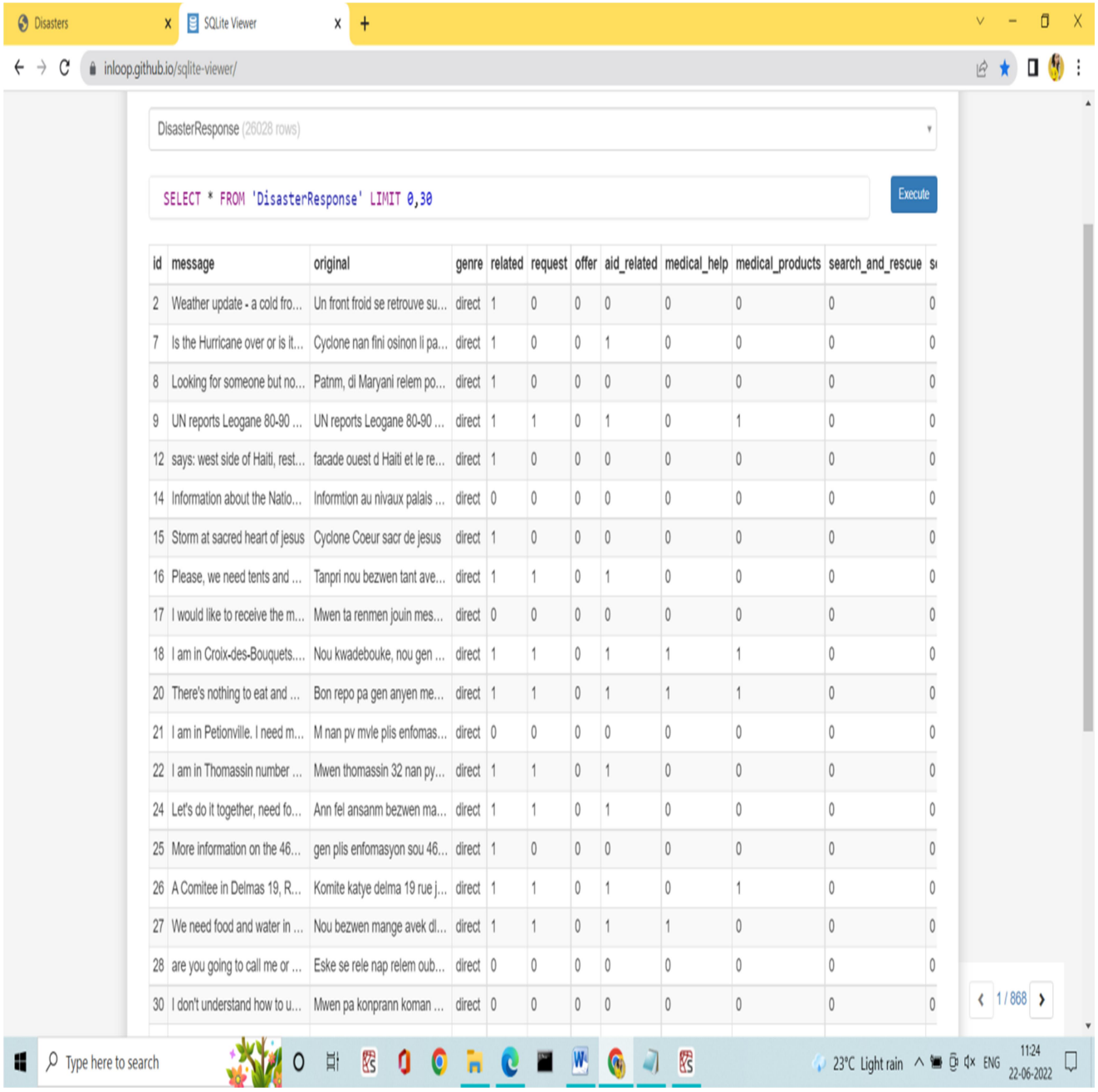


Fig: disaster_categories.csv



id	message	original	genre	related	request	offer	aid_related	medical_help	medical_products	search_and_rescue	s
2	Weather update - a cold fro...	Un front froid se retrouve su...	direct	1	0	0	0	0	0	0	0
7	Is the Hurricane over or is it...	Cyclone nan fini osinon li pa...	direct	1	0	0	1	0	0	0	0
8	Looking for someone but no...	Patnm, di Maryani relem po...	direct	1	0	0	0	0	0	0	0
9	UN reports Leogane 80-90 ...	UN reports Leogane 80-90 ...	direct	1	1	0	1	0	1	0	0
12	says: west side of Haiti, rest...	facade ouest d Haiti et le re...	direct	1	0	0	0	0	0	0	0
14	Information about the Natio...	Informtion au nivaux palais ...	direct	0	0	0	0	0	0	0	0
15	Storm at sacred heart of jesus	Cyclone Coeur sacr de jesus	direct	1	0	0	0	0	0	0	0
16	Please, we need tents and ...	Tanpri nou bezwen tant ave...	direct	1	1	0	1	0	0	0	0
17	I would like to receive the m...	Mwen ta renmen jouin mes...	direct	0	0	0	0	0	0	0	0
18	I am in Croix-des-Bouquets....	Nou kwadebouke, nou gen ...	direct	1	1	0	1	1	1	0	0
20	There's nothing to eat and ...	Bon repo pa gen anyen me...	direct	1	1	0	1	1	1	0	0
21	I am in Petionville. I need m...	M nan pv mvle plis enfomas...	direct	0	0	0	0	0	0	0	0
22	I am in Thomassin number ...	Mwen thomassin 32 nan py...	direct	1	1	0	1	0	0	0	0
24	Let's do it together, need fo...	Ann fel ansanm bezwen ma...	direct	1	1	0	1	0	0	0	0
25	More information on the 46...	gen plis enfomasyon sou 46...	direct	1	0	0	0	0	0	0	0
26	A Comitee in Delmas 19, R...	Komite katye delma 19 rue j...	direct	1	1	0	1	0	1	0	0
27	We need food and water in ...	Nou bezwen mange avek di...	direct	1	1	0	1	1	0	0	0
28	are you going to call me or ...	Eske se rele nap relem oub...	direct	0	0	0	0	0	0	0	0
30	I don't understand how to u...	Mwen pa konprann koman ...	direct	0	0	0	0	0	0	0	0

Fig 5- DatabaseResponse.db: Run this command for Database Creation

python data/process_data.py data/disaster_messages.csv data/disaster_categories.csv data/DisasterResponse.db

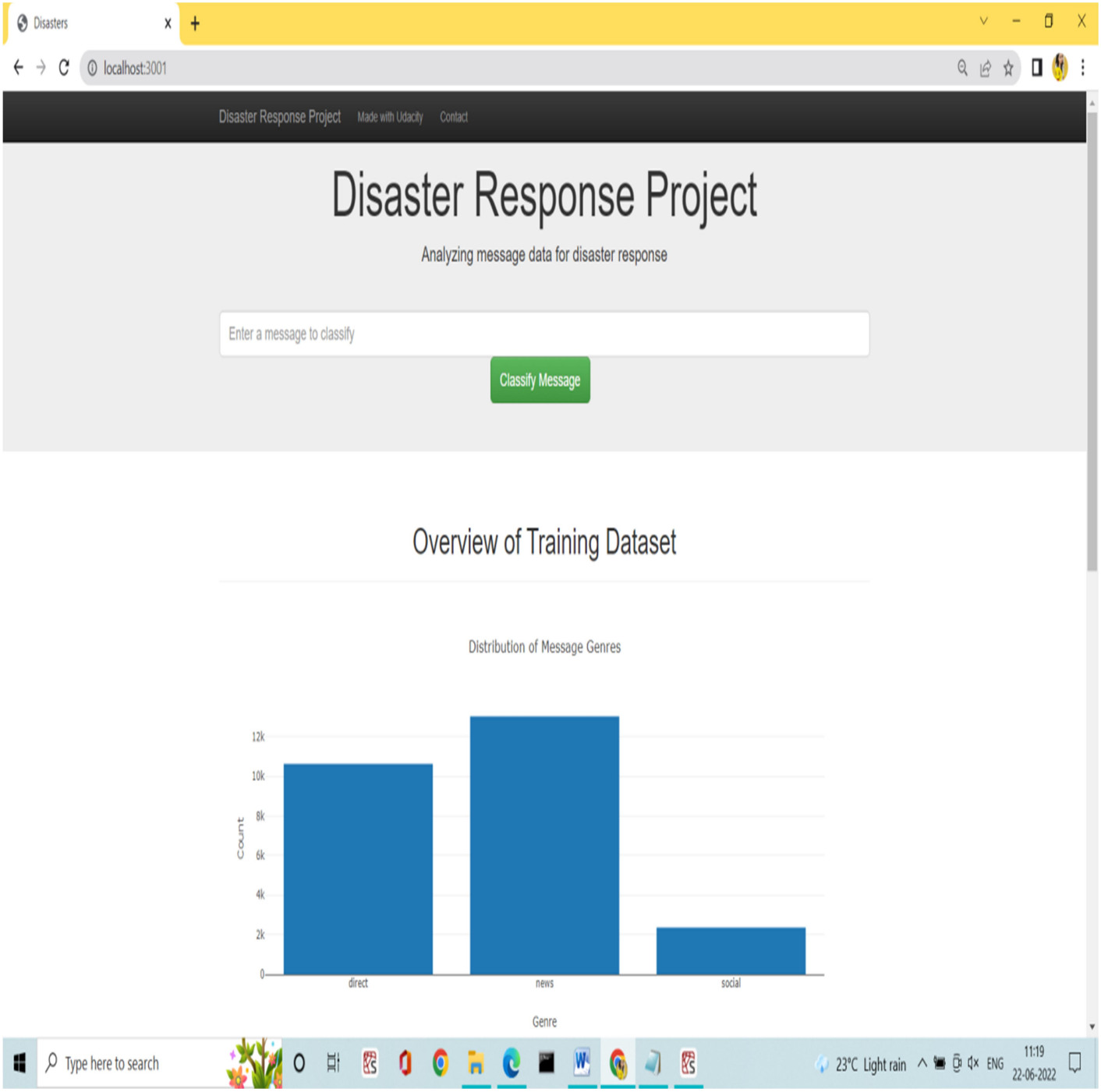
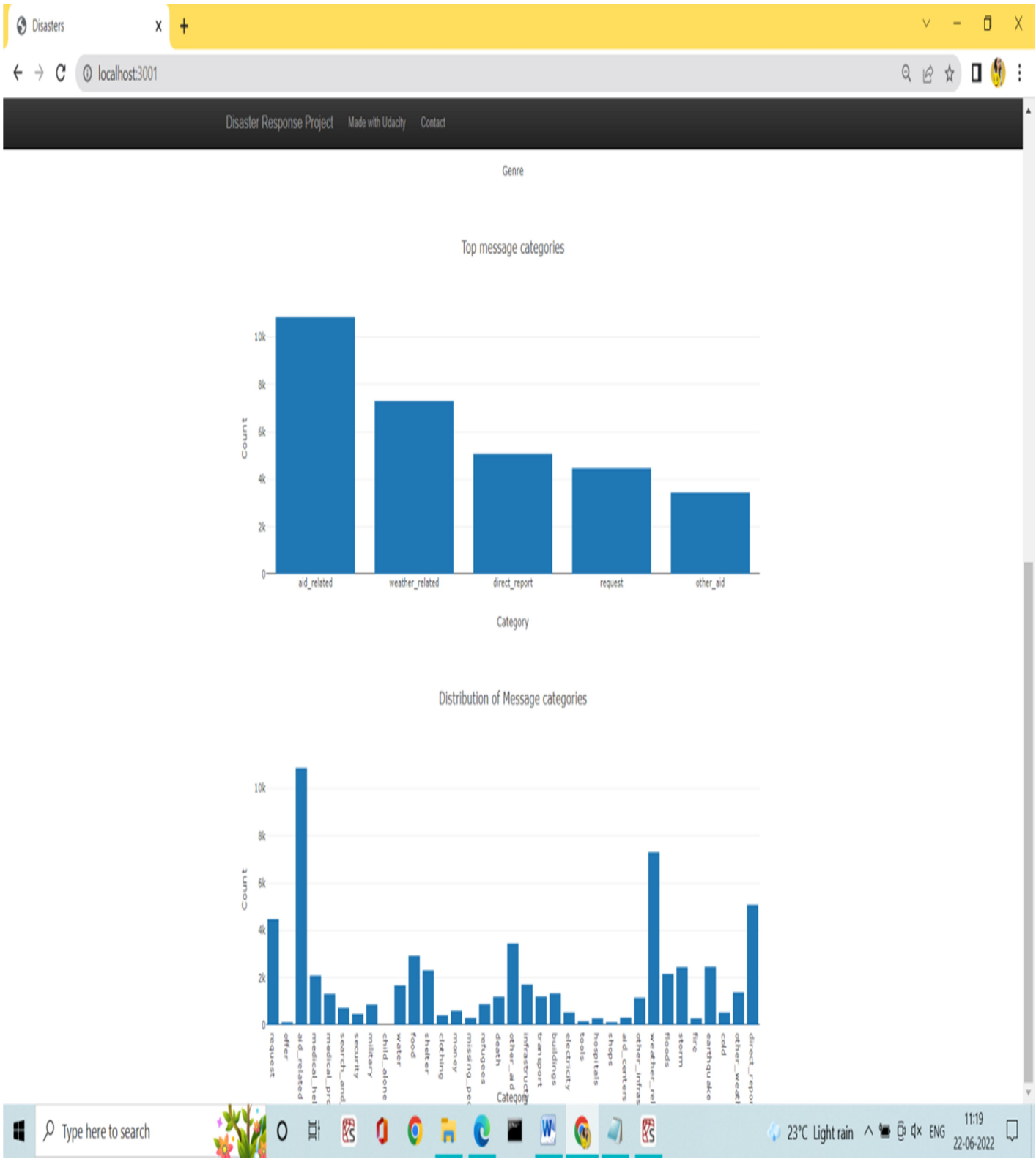


Fig 6- Disaster survival web application

Run this command : python app/run.py


Disasters
localhost:3001
Disaster Response Project
Made with Udacity
Contact
Genre
Top message categories
Count
10k
8k
6k
4k
2k
0
aid_related
weather_related
direct_report
request
other_aid
Category
Distribution of Message categories
Count
Category
Type here to search
23°C Light rain
ENG
11:19
22-06-2022

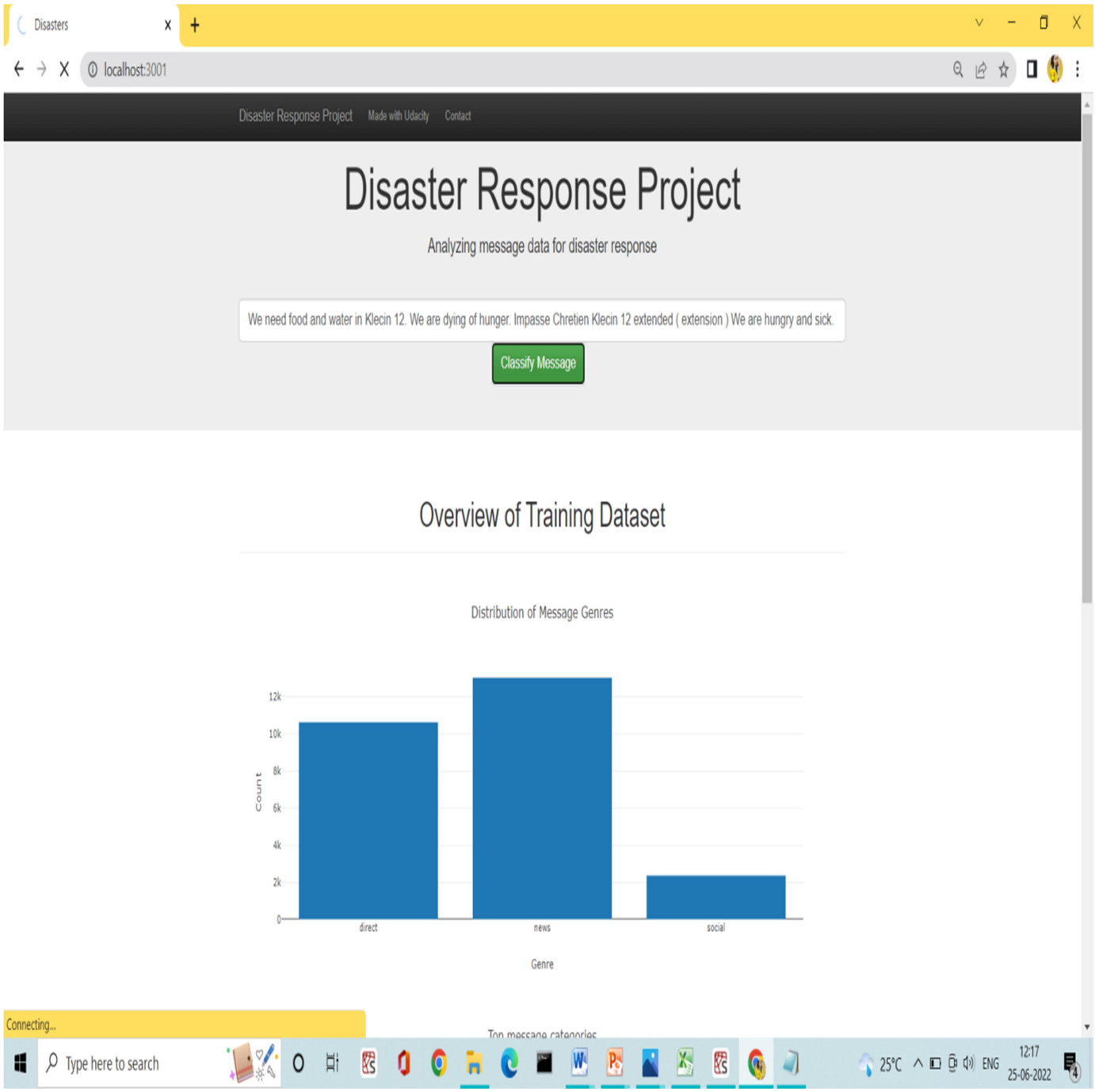

Disasters
localhost:3001
Disaster Response Project
Made with Udacity
Contact
Disaster Response Project
Analyzing message data for disaster response
We need food and water in Klecin 12. We are dying of hunger. Impasse Chretien Klecin 12 extended (extension) We are hungry and sick.
Classify Message
Overview of Training Dataset
Distribution of Message Genres
12k
10k
8k
6k
4k
2k
0
Count
direct
news
social
Genre
Connecting...
Type here to search
25°C
ENG
12:17
25-06-2022

Fig Result 1

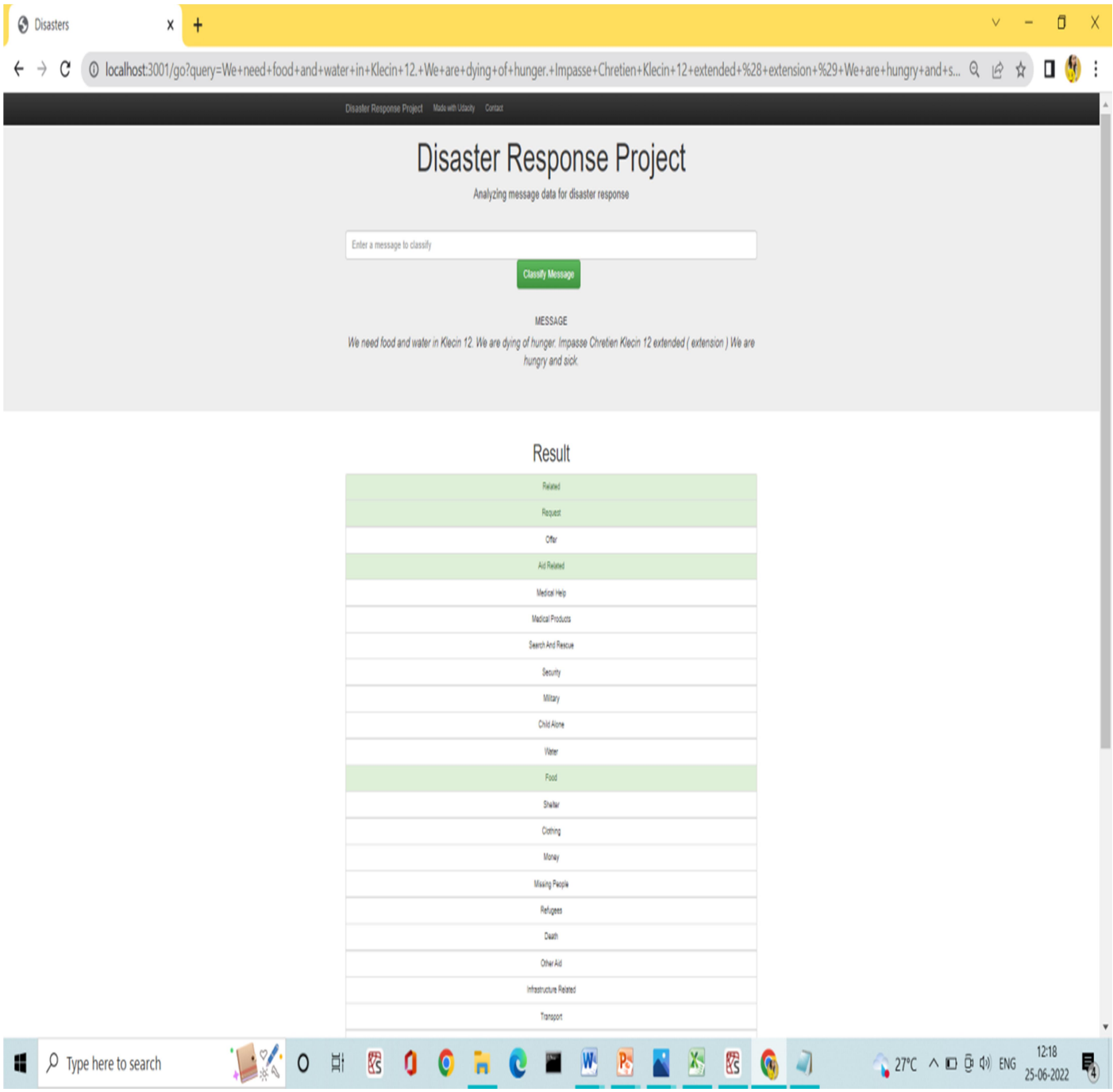

Disasters
localhost:3001/go?query=We+need+food+and+water+in+Klecin+12.+We+are+dying+of+hunger.+Impasse+Chretien+Klecin+12+extended+%28+extension+%29+We+are+hungry+and+s...
Disaster Response Project
Made with Udacity
Contact
Disaster Response Project
Analyzing message data for disaster response
Enter a message to classify
Classify Message
MESSAGE
We need food and water in Klecin 12. We are dying of hunger. Impasse Chretien Klecin 12 extended (extension) We are hungry and sick.
Result
Related
Request
Offer
Aid Related
Medical Help
Medical Products
Search And Rescue
Security
Military
Child Alone
Water
Food
Shelter
Clothing
Money
Missing People
Refugees
Death
Other Aid
Infrastructure Related
Transport
Type here to search
27°C
ENG
12:18
25-06-2022

Fig Result 2