MACHINE LEARNING FOR DISASTER MANAGEMENT

Mini Project Report

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Engineering (Computer Engineering)

by:

Ketki Matkar	TU3F1718135
Pratiksha Lohare	TU3S1819004
Vrushali Pawar	TU3S1819006

Under the Guidance of

Prof. Audumbar Umbare

Department of Computer Engineering

TERNA ENGINEERING COLLEGE

Nerul (W), Navi Mumbai 400706

(University of Mumbai)

(2019-2020)

Internal Approval Sheet

TERNA ENGINEERING COLLEGE, NERUL

Department of Computer Engineering

Academic Year 2019-20

CERTIFICATE

This is to certify that the major project entitled **"Machine Learning for Disaster Management"** is a bonafide work of

Ketki Matkar	TU3F1718135
Pratiksha Lohare	TU3S1819004
Vrushali Pawar	TU3S1819006

Submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the Bachelor of Engineering (Computer Engineering).

Guide **Head of Department** **Principal**

Approval Sheet

Project Report Approval

This Major Project Report – an entitled "**Machine Learning for Disaster Management**" by following students is approved for Mini Project of ***Third Year in "Computer Engineering"***.

Submitted by:

Ketki Matkar	TU3F1718135
Pratiksha Lohare	TU3S1819004
Vrushali Pawar	TU3S1819006

Examiners Name & Signature:

1.----------

2.----------

Date: ----------

Place: ----------

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Ketki Matkar	TU3F1718135
Pratiksha Lohare	TU3S1819004
Vrushali Pawar	TU3S1819006

Date: _____

Place: _____

Acknowledgement

We would like to express our sincere gratitude towards our guide **Prof. Audumbar Umbare**, Project Coordinators **Prof. Gaurav Deshmukh, Prof. Dnyaneshwar Thombre** for their help, guidance and encouragement, they provided during the project development. This work would have not been possible without their valuable time, patience and motivation. We thank them for making my stint thoroughly pleasant and enriching. It was great learning and an honor being their student.

We are deeply thankful to **Dr. Archana Mire (H.O.D Computer Department)** and entire team in the Computer Department. They supported us with scientific guidance, advice and encouragement, they were always helpful and enthusiastic and this inspired us in our work.

We take the privilege to express our sincere thanks **to Dr. L. K. Ragha** our Principal for providing the encouragement and much support throughout our work.

Ketki Matkar TU3F1718135

Pratiksha Lohare TU3S1819004

Vrushali Pawar TU3S1819006

Date: _____

Place: _____

ABSTRACT

The sinking of the Titanic is one of the most historic shipwrecks of all time. The tragedy killed thousands, 1502 out of 2224 passengers, and led many wondering what could have been done better. One of the most important reason is that there was not enough lifeboats, and although there was probably quite amount of luck involved, there were some groups of people that were more likely to survive than others. In this paper a data analytical study will be conducted with the passenger's data from the Titanic from a data platform Kaggle to find out about this survival likelihood. For the data analytical approach, we apply the theory of machine learning, in this case Random Forest, to come up with models that can best predict what kinds of passengers are more likely to survive. The models' prediction performance varied from 77% on the public leaderboard to around 82% internally in the data mining tool. Weka, the data mining tool, was used to conduct the analysis and prediction.The Titanic Kaggle competition involves around two split dataset. These two datasets are generally called "train" and "test". One is used to train your prediction model, the other one is used to make your model prediction and your submission. The test dataset has a private and a public component and further separated as such. The public component will be published real-time on a public leaderboard, the private component will be published later at the end of the competition. This is done to prevent 'overfitting' the dataset. By means of the real-time public leaderboard you can quickly evaluate your own model and compare that with others.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER - 1

INTRODUCTION

1.1 Aim and Objectives of Project

The Big Data trend is quite noticeable lately. New theories and tools become available for many unsolved old questions such as the Titanic that sunk in 1912. While many studies explained the human and hydraulic cause of the sinking, questions remained regarding the chances of survival for the passengers . This renewed interests is mainly derived from a Kaggle competition. Kaggle is a platform where the data demand from businesses and organizations meet the supply of data engineers through a crowd-sourcing concept. The most common form of bringing this demand and supply together on this platform is through a competition with price money. For example, at time of writing, one of the top competition, the California Health-care Foundation rewards $100.000 to the best team of data engineers that is able to improve the models for fighting a disease that causes blindness. Microsoft rewards $16.000 for a classification problem regarding malware. Being able to understand data is a key competence of many companies and analysts nowadays. Regression, classification and machine learning theories are identified as one of the (Chen, 2012) foundational technologies and emerging data analytics. This article is focused around the analysis of a historical dataset from the Titanic tragedy from the Kaggle platform. The chosen data analytical method is Random Forest which will be elaborated in the next paragraph. The main research question will revolve around what Random Forest modelling can do for predicting the survival rate.

1.2 Motivation of Project

The importance of Machine Learning is deep-rooted in our day to day lives over the past few years, that we fail to identify them.

1. The heavily hyped, self-driving Google car? The essence of machine learning.
2. Online recommendation offers such as those from Amazon and Netflix? Machine learning applications for everyday life.
3. Knowing what customers are saying about you on Twitter? Machine learning combined with linguistic rule creation.
4. Fraud detection? One of the more obvious, important uses in our world today.
5. The side boosters of Falcon Heavy which landed back on to the space station? Though there are several other parameters involved in making a side boosters land, a highly trained Machine Learning model has helped them navigate their way through.

Machine Learning in medicine? Stanford ML research lab under Andrew Ng is already under some fantastic research which helps classify where the fatigue in the muscles is by looking at the X-Ray. In 2/3 cases it proved to predict better than the radiologists themselves who had treated the patient.
Point being, moving forward the application of Machine Learning is going to increase and as the data becomes more and more available, computational processing becomes cheaper and data storage become even more affordable, the ability to produce sound and highly accurate Machine Learning models would also be very conducive.

1.3 Organization of the report

Chapter 1 gives a brief overview about the aim for developing this project. The problem definition tells us about the expected outcome of the project for the application.

Chapter 2 of the report includes the literature survey on the existing system.

Chapter 3 shows the software model used to design is described, along with this our proposed system is also described.

Chapter 4 shows the UML diagrams that give us an abstract view of the system. This chapter gives a detailed explanation about the technologies used and the techniques used in the development of the project. Along with this the Hardware and software requirements and software component are described.

Chapter 5 shows the project module that are designed and used in our proposed system.

Chapter 6 shows the overall working of the system.

Chapter 7 evaluates the performance of the project. We have stetted up an experiment setup to do the same.

Chapter 8 shows the Gantt chart of the project.

Chapter 9 describes the tasks that are achieved through this project. It describes the applications of the project.

Chapter 10 is conclusion. This chapter gives a summary of the entire project. It also gives the future scope for research and development in this project.

CHAPTER - 2

LITERATURE SURVEY

2.1 Existing System

MACHINE LEARNING

At a very high level, machine learning is the process of teaching a computer system how to make accurate predictions when fed data. Those predictions could be answering whether a piece of fruit in a photo is a banana or an apple, spotting people crossing the road in front of a self-driving car, whether the use of the word book in a sentence relates to a paperback or a hotel reservation, whether an email is spam, or recognizing speech accurately enough to generate captions for a YouTube video. The key difference from traditional computer software is that a human developer hasn't written code that instructs the system how to tell the difference between the banana and the apple. Instead a machine-learning model has been taught how to reliably discriminate between the fruits by being trained on a large amount of data.

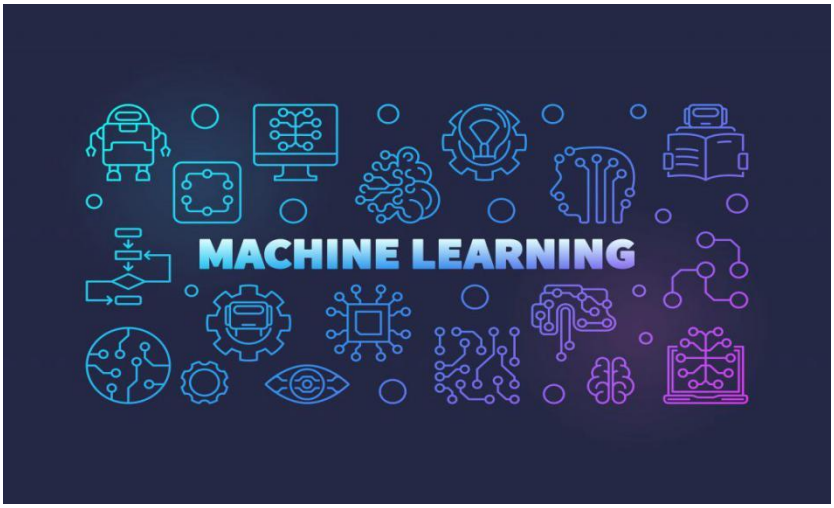

Figure 2.1: Machine Learning

TYPES OF MACHINE LEARNING

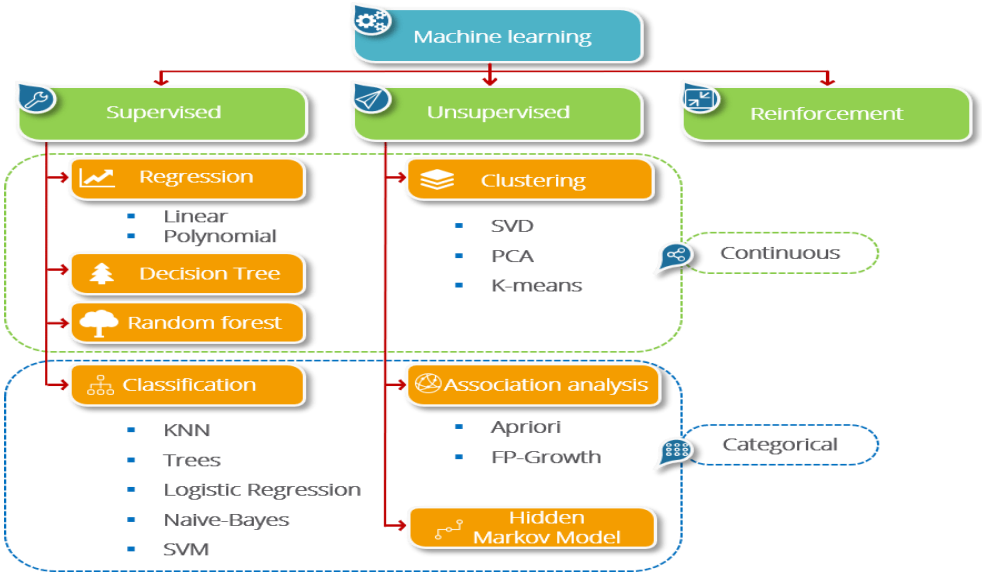

Figure 2.2 : Types of Machine Learning

Supervised Learning:

Supervised Learning is the one, where you can consider the learning is guided by a teacher. We have a dataset which acts as a teacher and its role is to train the model or the machine. Once the model gets trained it can start making a prediction or decision when new data is given to it.This is the learning type that you will most likely encounter, as it is exhibited in many of the following common applications:

1) **Advertisement Popularity**: Selecting advertisements that will perform well is often a supervised learning task. Many of the ads you see as you browse the internet are placed there because a learning algorithm said that they were of reasonable popularity (and clickability). Furthermore, its placement associated on a certain site or with a certain query (if you find yourself using a search engine) is largely due to a learned algorithm saying that the matching between ad and placement will be effective.

2) **Spam Classification**: If you use a modern email system, chances are you've encountered a spam filter. That spam filter is a supervised learning system. Fed email examples and labels (spam/not spam), these systems learn how to preemptively filter out malicious emails so that their user is not harassed by them. Many of these also behave in such a way that a user can provide new labels to the system and it can learn user preference.

3) **Face Recognition**: Do you use Facebook? Most likely your face has been used in a supervised learning algorithm that is trained to recognize your face. Having a system that

takes a photo, finds faces, and guesses who that is in the photo (suggesting a tag) is a supervised process. It has multiple layers to it, finding faces and then identifying them, but is still supervised nonetheless.

Unsupervised Learning:

The model learns through observation and finds structures in the data. Once the model is given a dataset, it automatically finds patterns and relationships in the dataset by creating clusters in it. What it cannot do is add labels to the cluster, like it cannot say this a group of apples or mangoes, but it will separate all the apples from mangoes.Suppose we presented images of apples, bananas and mangoes to the model, so what it does, based on some patterns and relationships it creates clusters and divides the dataset into those clusters. Now if a new data is fed to the model, it adds it to one of the created clusters.The outcomes from an unsupervised learning task are controlled by the data and the way its formatted. Some areas you might see unsupervised learning crop up are:

1) **Recommender Systems**: If you've ever used YouTube or Netflix, you've most likely encountered a video recommendation system. These systems are often times placed in the unsupervised domain. We know things about videos, maybe their length, their genre, etc. We also know the watch history of many users. Taking into account users that have watched similar videos as you and then enjoyed other videos that you have yet to see, a recommender system can see this relationship in the data and prompt you with such a suggestion.

2) **Buying Habits**: It is likely that your buying habits are contained in a database somewhere and that data is being bought and sold actively at this time. These buying habits can be used in unsupervised learning algorithms to group customers into similar purchasing segments. This helps companies market to these grouped segments and can even resemble recommender systems.

3) **Grouping User Logs**: Less user facing, but still very relevant, we can use unsupervised learning to group user logs and issues. This can help companies identify central themes to issues their customers face and rectify these issues, through improving a product or designing an FAQ to handle common issues. Either way, it is something that is actively done and if you've ever submitted an issue with a product or submitted a bug report, it is likely that it was fed to an unsupervised learning algorithm to cluster it with other similar issues.

Reinforcement Learning:

Reinforcement learning is fairly different when compared to supervised and unsupervised learning. Where we can easily see the relationship between supervised and unsupervised (the presence or absence of labels), the relationship to reinforcement learning is a bit murkier. Some people try to tie reinforcement learning closer to the two by describing it as a type of learning that relies on a time-dependent sequence of labels, however, my opinion is that that simply makes things more confusing. For any reinforcement learning problem, we need an agent and an environment as well as a way to connect the two through a feedback loop. To connect the agent to the environment, we give it a set of actions that it can take that affect the environment. To connect the environment to the agent, we have it continually issue two signals to the agent: an updated state and a reward (our reinforcement signal for behavior).

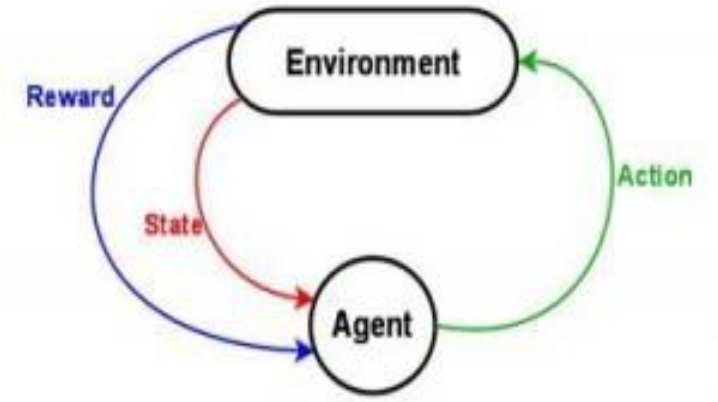

Figure 2.3 : Reinforcement Learning

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 SDLC Model

We can choose SDLC waterfall model due to nature of this model, it is hard to get back to the previous phase once completed. Although these is can be very rigid in some software projects which need some flexibility while this model can be essential or the most suitable model for other software project contexts. The usage of the waterfall model can fall under the projects which do not focus on changing the requirements, for example:

1) Projects initiated from a request for proposal (RFP), the customer has a very clear documented requirements
2) Mission Critical projects, for example, in a Space shuttle
3) Embedded systems

You can review the selection criteria for the SDLC models from this article, while this is the list of things to consider when using the waterfall:

- The requirements are clear and frozen.
- Technology is understood and used by the team in different projects.
- The project cannot be delivered in an iterative manner.
- Documentation is essential.
- Professional Project management skills.
- The project cost is defined.

3.1.1 Phases of SDLC Model

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

The following illustration is a representation of the different phases of the Waterfall Model.

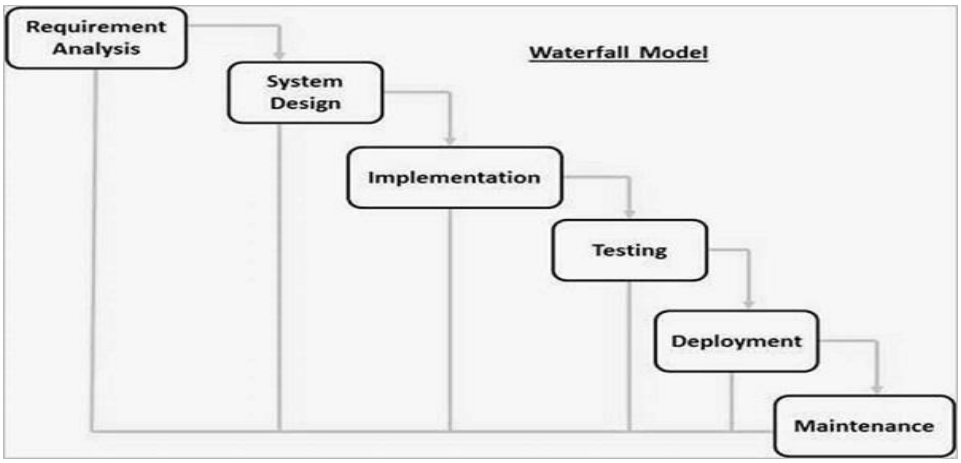

Figure 3.1 : Waterfall Model

The sequential phases in Waterfall model are −

1) **Requirement Gathering and analysis** −All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.

2) **System Design** −The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.

3) **Implementation** −With inputs from the system design; the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.

4) **Integration and Testing** − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

5) **Deployment of system** − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.

6) **Maintenance** − There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model, phases do not overlap.

3.1.2 Waterfall Model - Application

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are −
Requirements are very well documented, clear and fixed.

- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

3.1.3 Waterfall Model - Advantages

The advantages of waterfall development are that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one.
Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order.
Some of the major advantages of the Waterfall Model are as follows −

- Simple and easy to understand and use
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood.
- Clearly defined stages.
- Well understood milestones.
- Easy to arrange tasks.
- Process and results are well documented.

3.1.3 Waterfall Model - Disadvantages

The disadvantage of waterfall development is that it does not allow much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage.
The major disadvantages of the Waterfall Model are as follows −

- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long and ongoing projects.
- Not suitable for the projects where requirements are at a moderate to high risk of changing.
- It is difficult to measure progress within stages.
- Cannot accommodate changing requirements.
- Adjusting scope during the life cycle can end a project.
- Integration is done as a "big-bang at the very end, which doesn't allow identifying any technological or business bottleneck or challenges early.

CHAPTER 4

PROJECT SCHEDULING

Project Scheduling in a project refers to road-map of all activities to be done with specified order and within time slot allotted to each activity. Project managers tend to define various tasks, and project milestones and they arrange them keeping various factors in mind. They look for tasks lie in critical path in the schedule, which are necessary to complete in specific manner (because of task inter dependency) and strictly within the time allocated. Arrangements of tasks which lie out of critical path are less likely to impact over all schedule of the project.

For scheduling a project, it is necessary to -

1. Break down the project tasks into smaller, manageable form
2. Find out various tasks and correlate them
3. Estimate time frame required for each task
4. Divide time into work-units
5. Assign adequate number of work-units for each task
6. Calculate total time required for the project from start to finish

4.1 Time Line Chart

A gantt chart is a horizontal bar chart used to show a project plan and its progress over time. Gantt charts are incredibly useful in project management because they allow you to track the status of project tasks. They also help keep track of deadlines, milestones, and hours worked.

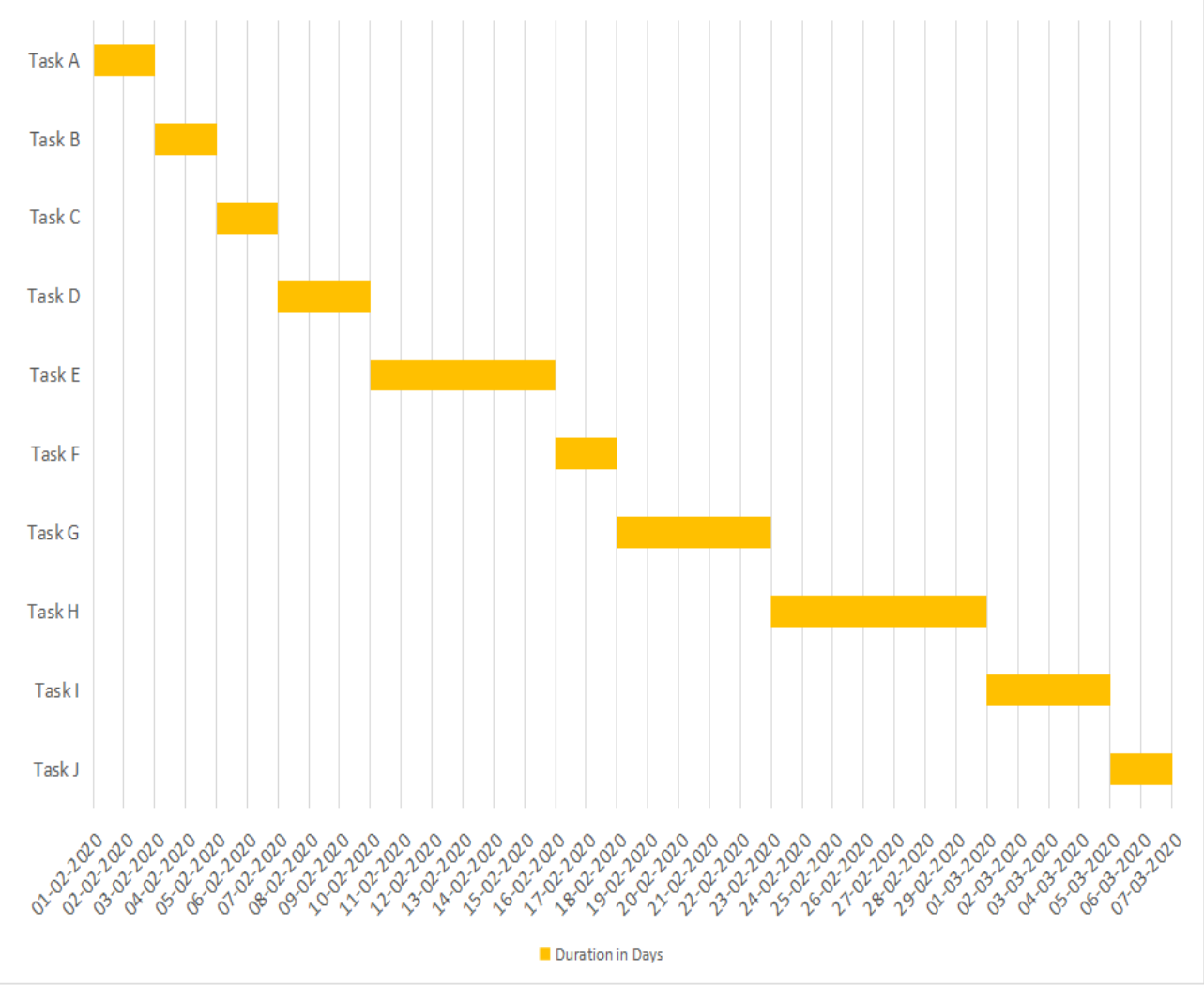

Figure 4.1 : Gantt Chart

CHAPTER 5

REQUIREMENT MODELING

5.1 Use Case Description

A use case model is a business analysis presentation of the steps defining the interactions between a user (called an actor) and a system (usually a computer system). It details the interactions and sets the expectations of how the user will work within the system.

The use case model consists of two artifacts: the use case diagram, which is a graphical representation showing which actors can operate which use cases, and the use case description (sometimes called the use case narrative), which is the text-based, detailed, step-by-step interactions and dialogue between the actor and the system.

The use case narrative is what people often mean when they say use case. Just remember there are multiple pieces that make up a use case model.

The use case description is a written account of the sequence of steps performed by an analyst to accomplish a complete business transaction. It's initiated by an actor, provides value to that actor, and is a goal of the actor working in that system. Below, you see a use case description that clearly documents how a student manager approves a training request from a student worker.

USE CASE DESCRIPTION FOR THE PROJECT:

Table 1:List of Actors

ACTOR	DESCRIPTION
End User	View the computation done by machine learning algorithm.
Developer	Develops machine learning model to do analysis of data set.
Database Administrator	Provides the updated data to developer.

Table 5.1 : List Of Actors

Table 2:Use Case Description	
Name	Prediction Of Passengers Survived in Titanic
ID	UC-001
Description	Client can view the computations made by machine learning algorithm which is developed by developer
Primary Actor	End User
Preconditions	Client should be logged into the system
Post-conditions	Client can view the predictions made by the trained model
Organizational Benefits	It will increase the awareness among the people about how the precautions should be taken. In case of worst scenarios,it tells how to control the every type of loss
Frequency Of Use	A few times every quarter
Main Course	1. Database Administrator will provide the latest data after committing all the corrections in the server to developer. 2. Admin will provide data only if developer has been given the permission by the higher authority. 3. Developer will convert the raw data into data set and differentiate them into various classes. 4. Developer will develop a user friendly model by applying machine learning algorithms. 5. Model will be trained by providing the data set. 6. It will analyse the information and according to the inputs given by end user the output will be displayed. 7. The output which is made visible can be in form of text or graphical format like picture,graph,charts etc. 8. Software resources cannot be used , every time user has to do the authentication.
Alternate Course	1. Developer can give access to particular clients so that they wont have to verify their identity during accessing the system. 2. Client can directly use the resources of the software. 3. Client can even add new data set for testing the system.
Extension	1. Database Administrator has the rights to update the database and manage the server. 2. Admin can backup and restore data for maintaining the integrity. 3. Admin can create new user and change their profile. 4. If any user violates the rules, admin can delete the user. 5. Developer can make new model for computation.

Table 5.2 : Use Case Description

5.2 UML Diagram

A cornerstone part of the system is the functional requirements that the system fulfills. Use Case diagrams are used to analyze the system's high-level requirements. These requirements are expressed through different use cases. We notice three main components of this UML diagram:

1) Functional requirements – represented as use cases; a verb describing an action
2) Actors – they interact with the system; an actor can be a human being, an organization or an internal or external application
3) Relationships between actors and use cases – represented using straight arrows

Within the circular containers, we express the actions that the actors perform. Such actions are: purchasing and paying for the stock, checking stock quality, returning the stock or distributing it. As you might have noticed, use case UML diagrams are good for showing dynamic behaviors between actors within a system, by simplifying the view of the system and not reflecting the details of implementation.

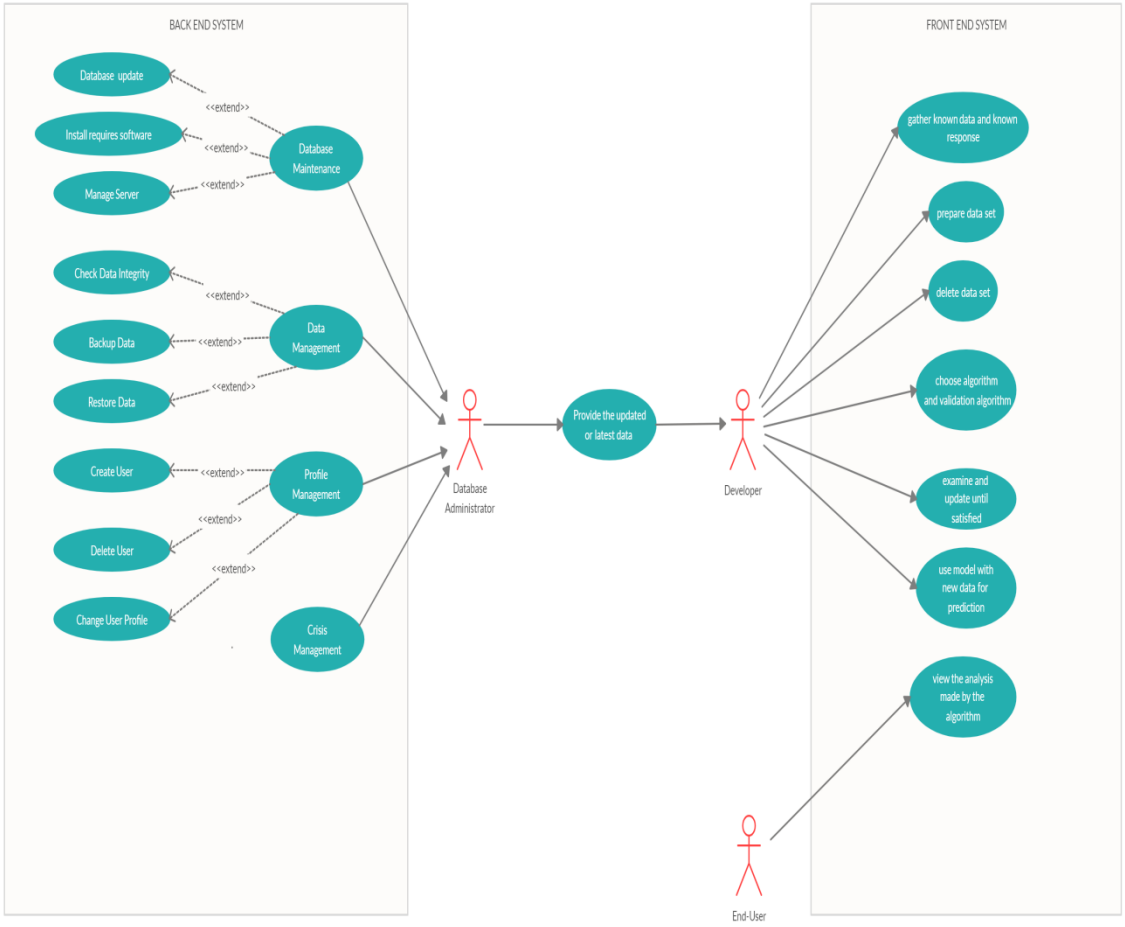

Figure 5.1 : UML Diagram

5.3 COCOMO Model Estimation

Cocomo (Constructive Cost Model) is a regression model based on LOC, i.e number of Lines of Code. It is a procedural cost estimate model for software projects and often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time and quality.
The key parameters which define the quality of any software products, which are also an outcome of the Cocomo are primarily Effort & Schedule:

- Effort: Amount of labor that will be required to complete a task. It is measured in person-months units.
- Schedule: Simply means the amount of time required for the completion of the job, which is, of course, proportional to the effort put. It is measured in the units of time such as weeks, months.

System	a_b	b_b	c_b	d_b
Organic	2.4	1.05	2.5	0.38
Semidetached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Table 5.3 : Basic Model

Effort=a * $(KLOC)^b$ PM
Development Time=c * $(Effort)^d$ Months
Average Staff Size=(Effort/Development Time) Persons
Productivity=(KLOC/Effort) KLOC per PM

Number of lines of code=80=0.08 KLOC

1. Organic Model:
Effort=2.4 * $(0.08)^{1.05}$ = 0.17 PM
Development Time=2.5 * $(0.17)^{0.38}$ =1.28 Months
Average Staff Size=(0.17/1.28) = 0.13 Persons
Productivity=(80/0.17) =470.59 KLOC per PM

2. Semidetached Model:
Effort=3.0 * $(0.08)^{1.12}$ = 0.18 PM
Development Time=2.5 * $(0.18)^{0.35}$ =1.37 Months
Average Staff Size=(0.18/1.37) = 0.13 Persons
Productivity=(80/0.18) = 444.44 KLOC per PM

3. Embedded Model:
Effort=3.6 * $(0.08)^{1.20}$ =0.17 PM
Development Time=2.5 * $(0.17)^{0.32}$ =1.42 Months
Average Staff Size=(0.17/1.42) =0.12 Persons
Productivity=(80/0.17) = 470.59 KLOC per PM

5.4 Activity Table

Activity Table is a tabular representation of the activities which are conducted according to the software development cycle. This table includes Task ID, Task Name, Start Date of the task, End Date of the task and Duration required to complete the respective task. It gives a brief idea of in whether the project will be finished in given deadline irrespective of the technical or other issues.

Task ID	Task Name	Start Date	End Date	Duration in Days
Task A	Making New Database	01-02-2020	02-02-2020	2
Task B	Upload Data in DB	03-02-2020	04-02-2020	2
Task C	Permission To Developer	05-02-2020	06-02-2020	2
Task D	Providing Data to DV	07-02-2020	09-02-2020	3
Task E	Preparation of Data Set	10-02-2020	15-02-2020	6
Task F	Installation of Software	16-02-2020	17-02-2020	2
Task G	Choosing Valid Algorithm	18-02-2020	22-02-2020	5
Task H	Developing ML Model	23-02-2020	29-02-2020	7
Task I	Training The Model	01-03-2020	04-03-2020	4
Task J	Giving Access to Client	05-03-2020	07-03-2020	3

Table 5.4 : Activity Table

5.5 Predecessor Table

Predecessor Table is a table that tells which tasks are dependent on each other and calculates the deadline for the project. It is very essential for development of flowchart of the software and network diagram.

Task	Immediate Predecessor	Duration
A	-	2
B	A	2
C	B	2
D	B,C	3
E	D	6
F	E	2
G	E	5
H	G,F	7
I	H,G	4
J	I,C	3

Table 5.5 : Predecessor Table

5.6 Flow Chart

A flowchart is a diagram that depicts a process, system or computer algorithm. They are widely used in multiple fields to document, study, plan, improve and communicate often complex processes in clear, easy-to-understand diagrams. Flowcharts, sometimes spelled as flow charts, use rectangles, ovals, diamonds and potentially numerous other shapes to define the type of step, along with connecting arrows to define flow and sequence. They can range from simple, hand-drawn charts to comprehensive computer-drawn diagrams depicting multiple steps and routes.

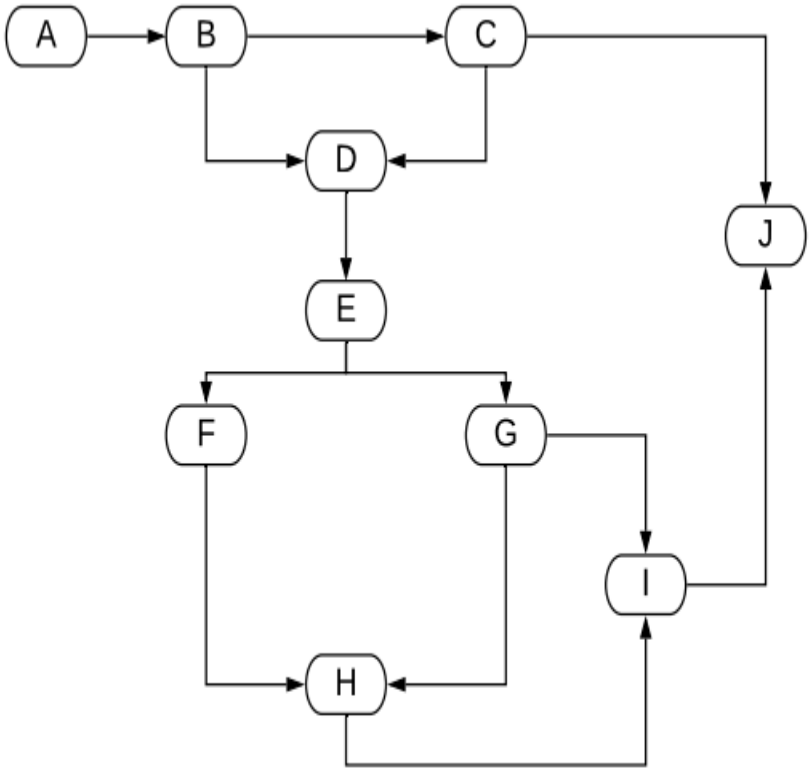

Figure 5.2 : Flow Chart

5.7 Network Diagram

A network diagram is a visual representation of network architecture. It maps out the structure of a network with a variety of different symbols and line connections. It is the ideal way to share the layout of a network because the visual presentation makes it easier for users to understand how items are connected.

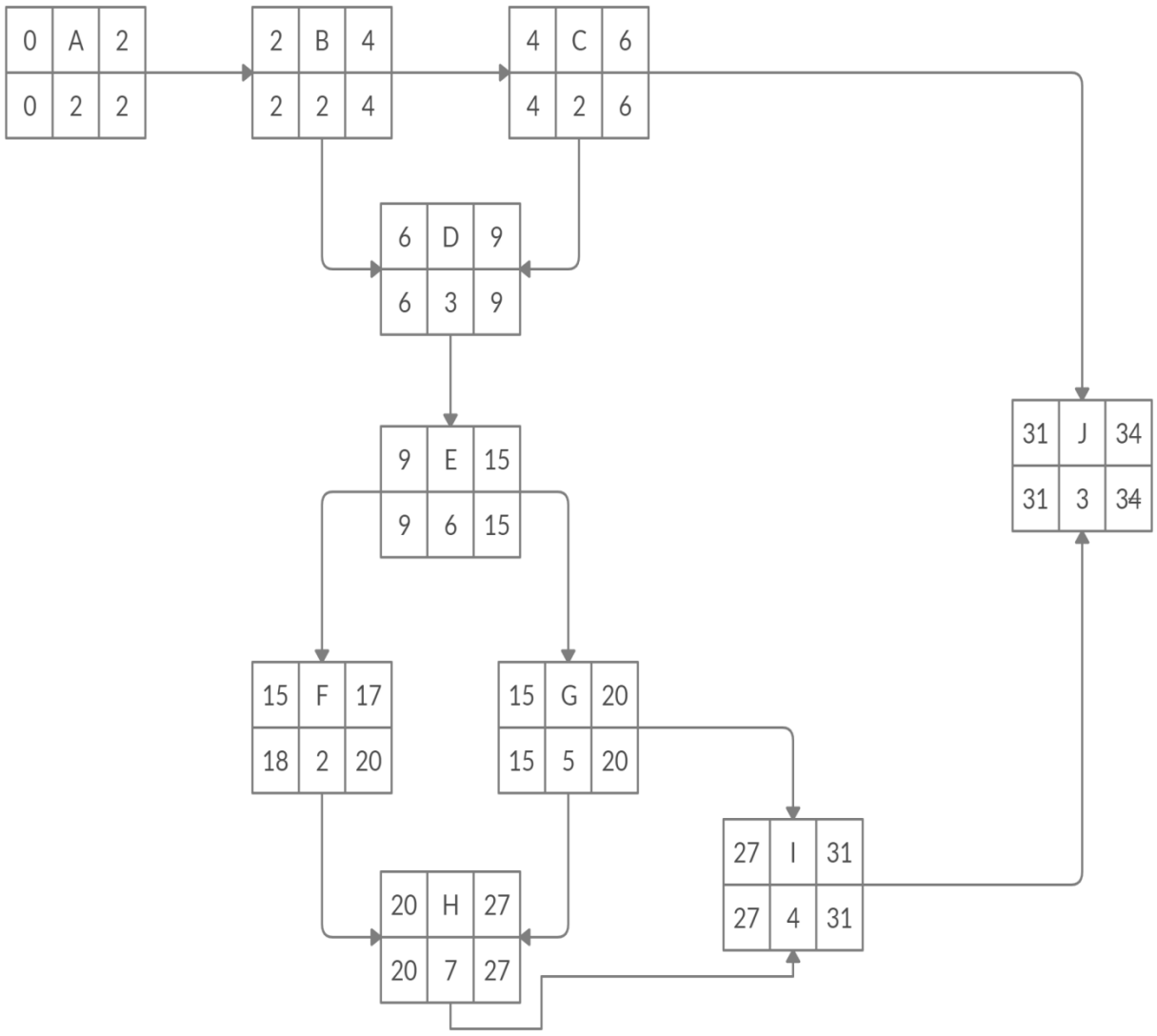

Figure 5.3 : Network Diagram

5.8 Class Diagram

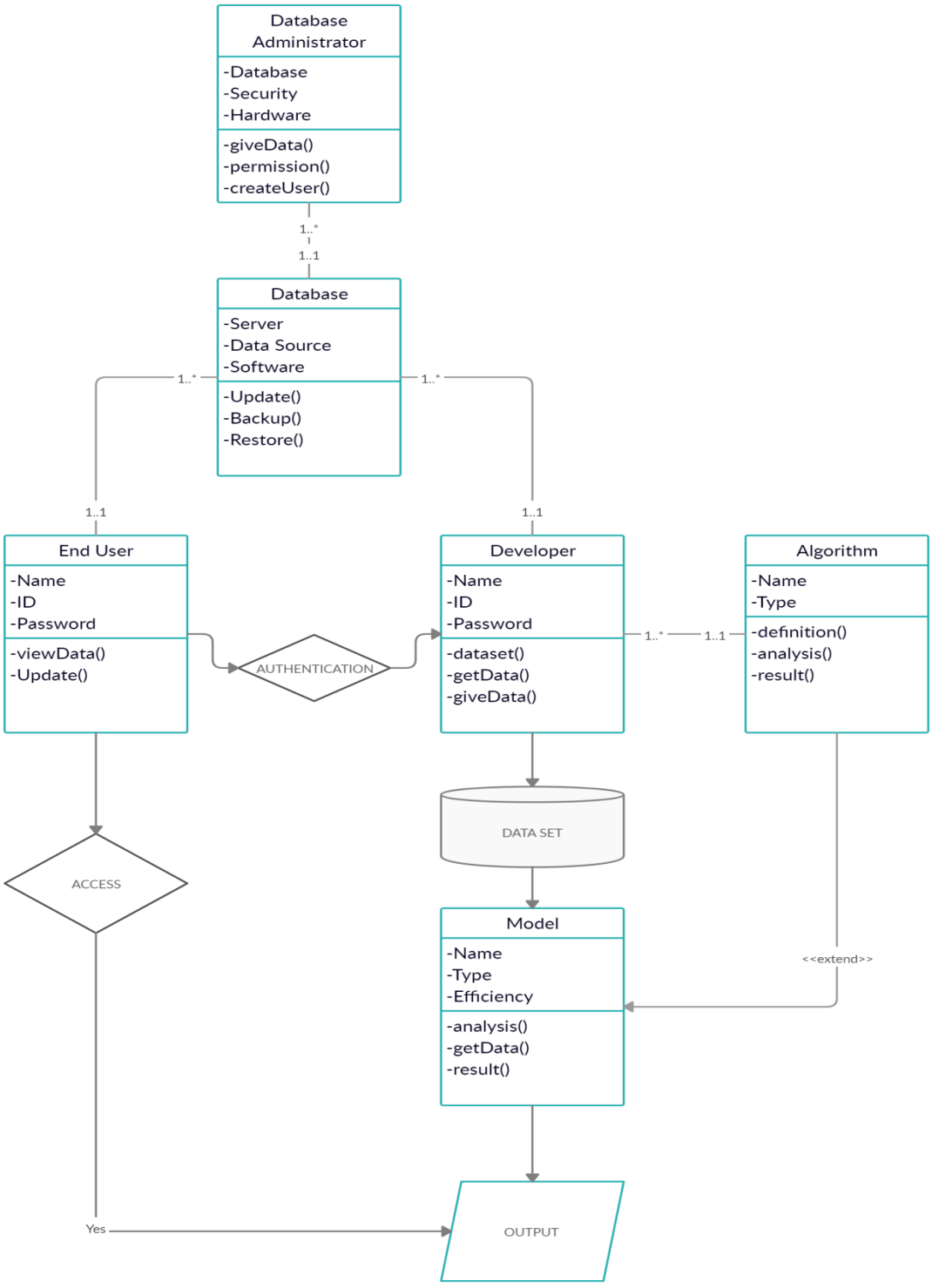

Figure 5.4 : Class Diagram

5.9 DFD Diagram

Also known as DFD, Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation.
Data flow diagrams can be divided into logical and physical. The logical data flow diagram describes flow of data through a system to perform certain functionality of a business. The physical data flow diagram describes the implementation of the logical data flow.
DFD graphically representing the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and between components of a system. The visual representation makes it a good communication tool between User and System designer. Structure of DFD allows starting from a broad overview and expand it to a hierarchy of detailed diagrams. DFD has often been used due to the following reasons:

- Logical information flow of the system
- Determination of physical system construction requirements
- Simplicity of notation
- Establishment of manual and automated systems requirements

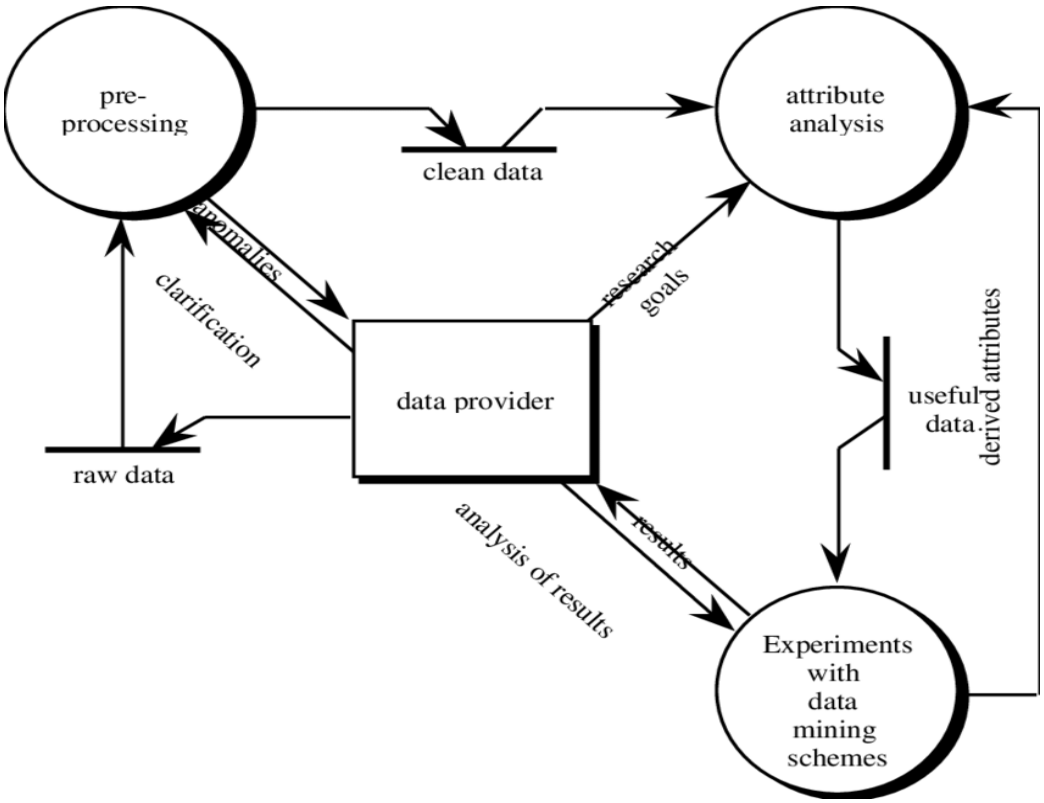

Figure 5.5 : DFD Diagram

5.10 Data Dictionary

Field Name	Field Size	Data Type	Description	Example
Database	1000	TEXT	Storage medium to store data	Titanic
Admin	50	TEXT	Person who controls and manages the system	Database Administrator
Developer	90	TEXT	Person who develops the software by using the data	Software Engineer
End User	20	TEXT	Person who uses the software made by developer	Customer or Client
Raw Data	1000	BINARY	It is the data which has been extracted by Data Source	record
Data Set	1000	BINARY	It is the converted data fro raw data	train,test
Input	10	INTEGER	It is the data which is given by the user operating the system	Data Set
Output	50	INTEGER	It is the data which has been computed by the software	Estimation, Figure,Graph
Algorithm	800	FUNCTION	It is a function which is defined to perform a specific task	getData, Update
Model	900	FUNCTION	It is physical representation of the algorithm	Naive Bayes
Profile	30	TEXT	It is a post which is assigned by the admin for the newly created users.	Admin, Developer
Permission	30	BOOLEAN	These are the rights given to existing users	Yes or No
Server	2000	TEXT	It is the storage device for Data which is very confidential	PHP

Table 5.6 : Data Dictionary

5.11 State Transition Diagram

These are used to model objects which have a finite number of possible states and whose interaction with the outside world can be described by its state changes in response to a finite number of events. A state transition diagram is a digraph whose nodes are states and whose directed arcs are transitions labelled by event names. A state is drawn as a rounded box containing an optional name. A transition is drawn as an arc with the arrow from the receiving state to the target state. The label on the arrow is the name of the event causing the transition. State transition diagrams have a number of applications. They are used in object-oriented modelling techniques to represent the life cycle of an object. They can be used as a finite state recognizer for a regular language, for instance, to describe regular expressions used as variables in computer languages.

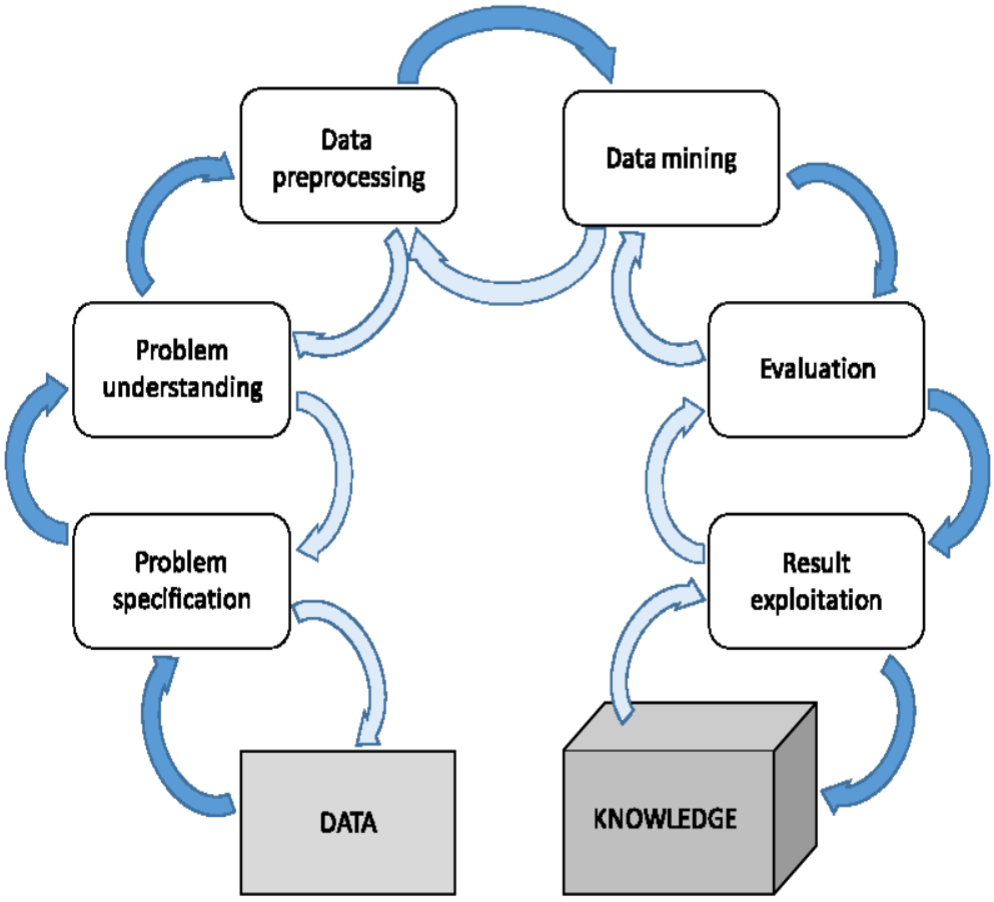

Figure 5.6 : State Transition Diagram

5.12 Activity Diagram

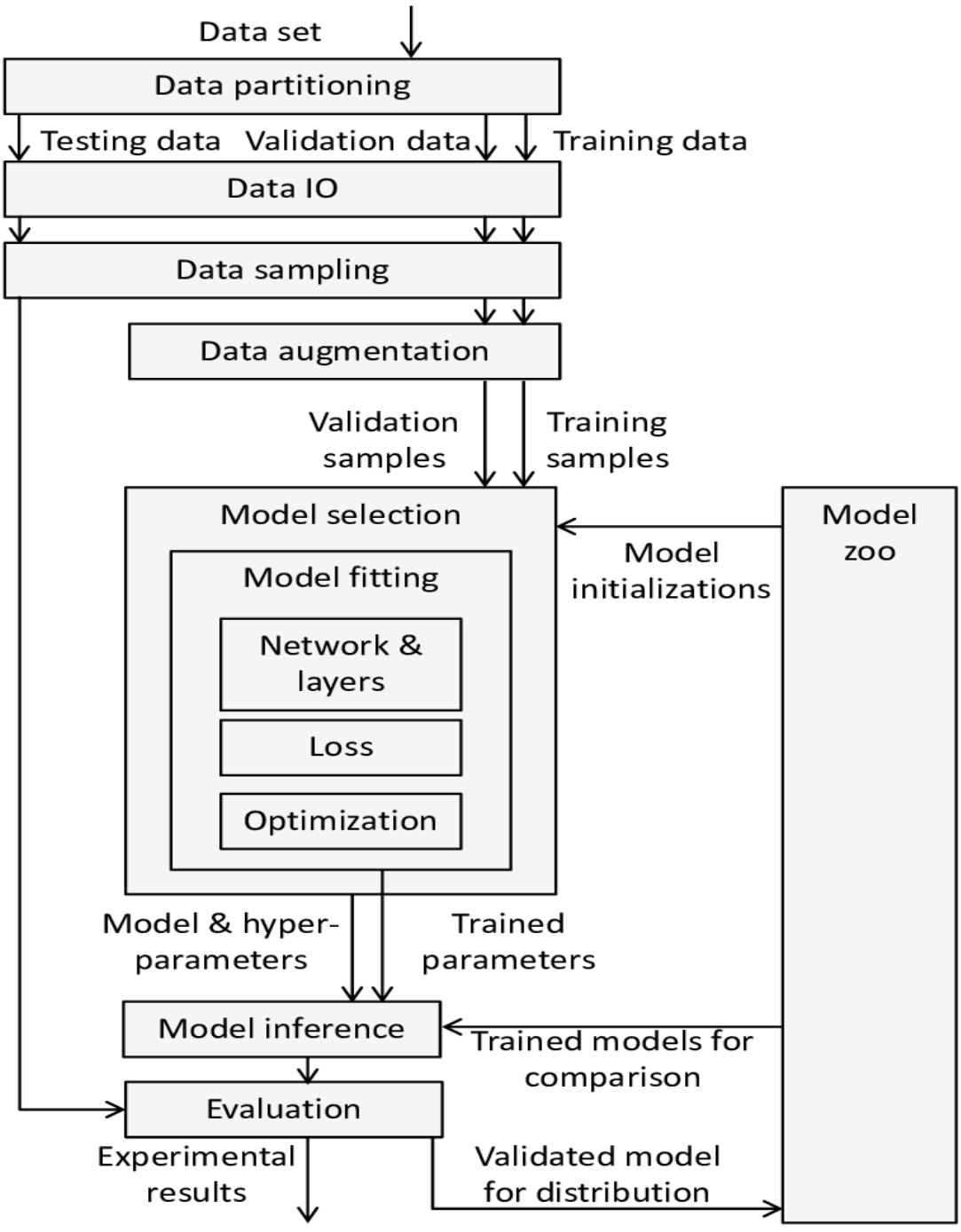

Figure 5.7 : Activity Table

CHAPTER 6

DESIGN

6.1 ER Diagram

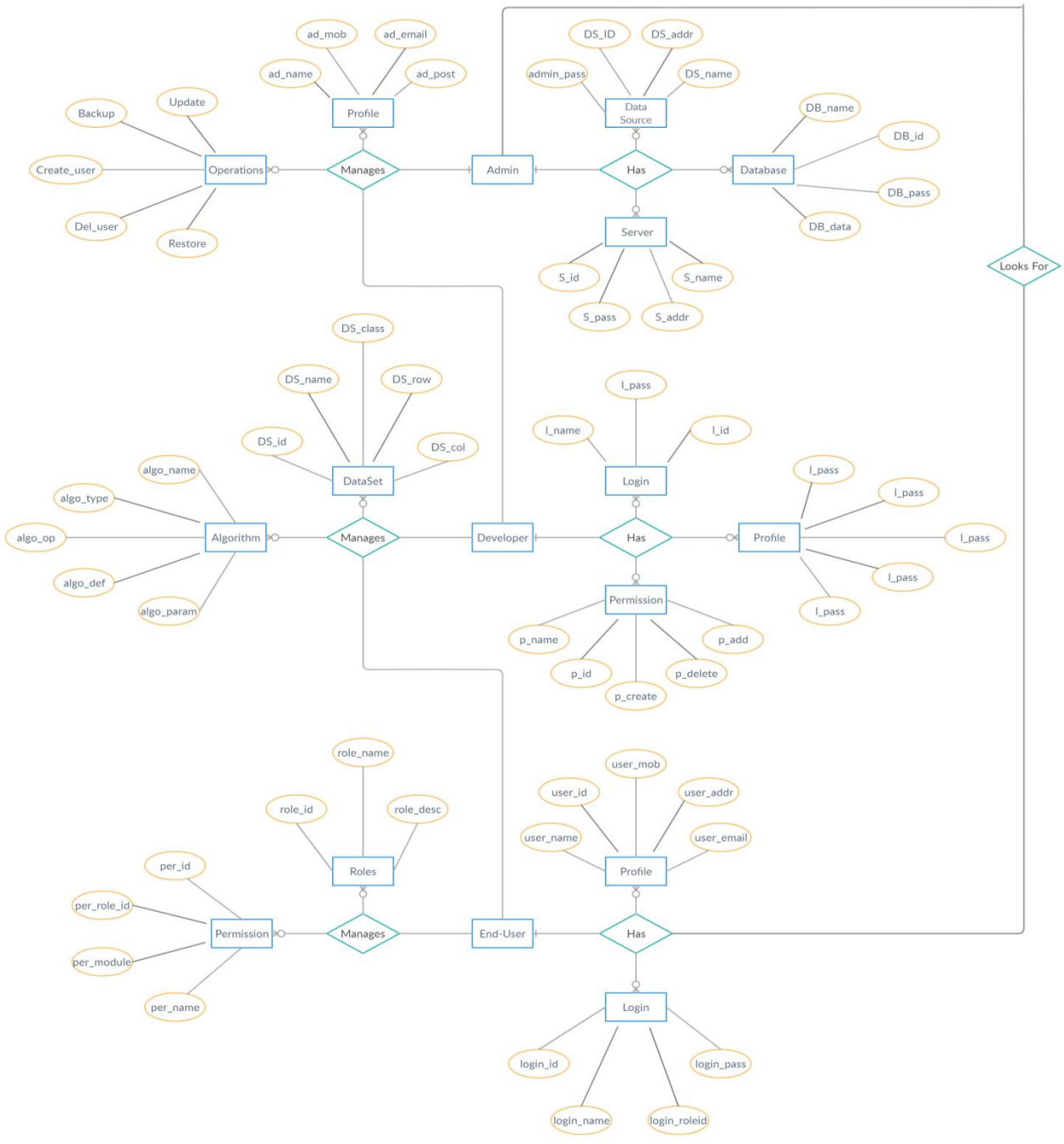

Figure 6.1 : ER Diagram

6.2 Software Requirement Specification (SRS)

A software requirements specification (SRS) is a detailed description of a software system to be developed with its functional and non-functional requirements. The SRS is developed based the agreement between customer and contractors. It may include the use cases of how user is going to interact with software system. The software requirement specification document consistent of all necessary requirements required for project development. To develop the software system we should have clear understanding of Software system. To achieve this we need to continuous communication with customers to gather all requirements.
A good SRS defines the how Software System will interact with all internal modules, hardware, communication with other programs and human user interactions with wide range of real life scenarios. Using the Software requirements specification (SRS) document on QA lead, managers creates test plan. It is very important that testers must be cleared with every detail specified in this document in order to avoid faults in test cases and its expected results.

SRS FOR THE PROJECT:

1. Introduction:

1.1. Purpose

Natural calamities cannot be avoid. Best efforts can be done to minimize the effect of these disaster. Mismanagement results into more damage to human life than the actual disaster. Disaster Management Systems help in managing post disaster so that another disaster doesn't occur after the natural disaster due to human mistakes. The design of the software discussed in this document follows the concept of providing the instant response to the causalities due to any kind of accident and maintaining the database about the casualties.

1.2. Scope

This documentation will help the disaster management team to analyse how the human mistakes converting into man made disaster can be prevented. Even if such calamities happens what should be best steps taken to minimize the loss i.e,death toll,damage to infrastructure etc.

1.3 Intended Audience

This software specification is intended for

- Researchers, developers in the field of the disaster management.
- Hospitals and welfare organizations, as it will enhance their way of accident and disaster management.
- End user’s, as it will improve their involvement in locating the place of the accident or disaster.

1.4 Benefits of the system:

The aim of the proposed system is to address the limitations of the current system. The requirements for the system have been gathered from the defects recorded in the past and also based on the feedback from users of previous metrics tools. Following are the objectives of the proposed system:

- Reducing time in activities
- Centralized data handling.
- Making it convenient and efficient for the users.

(Transfer the data smoothly to all the departments involved and handle the data centralized way).

1.5 Document Overview

This document provides the details of functional and non-functional requirements of the proposed system under the headings of overall Description, System Requirement and analysis, supplementary requirements, other non-functional requirements.

2. Overall description

The proposed system is supposed to replace the existing Disaster management System.

2.1 Product Perspective

The design of the software discussed in this document follows the concept of providing the instant response to the casualties due to any kind of accident,disaster and maintaining the database about the casualties. Therefore this software provides the layout by which a communication channel will be established between the end-user & the concerned body.
The End-user will view the product as a system which will provide it the functionality to inform/update the scenario regarding the accident or accident as it happened and the user can upload the pictures of the site of the accident or disaster.
Back-end refers to the point of operation or execution, in the sense that once the user has initiated the action of providing the information of the site of the accident or disaster, then the concerned body will provide the sources for rescue and relief. The version of system running at the back – end will be different from the end – user as it will have the geographical layout to identify the location of the user, which has sent the information, and maintained database regarding the casualties.

2.2 Product Function

The following are the product functions for proposed system:

1. Analyses the death toll on various factors
2. Gives idea about strategies to reduce the loss.
3. Uses machine learning techniques due to which human efforts are reduced and errors are nil.
4. Data set provided by organization are analyzed very efficiently by machine learning model.

5. Maintaining records of the missing person on website via database.
6. Find Injured and missing Persons.

2.3 User characteristics

The user should be familiar with the activities involved in the calamities and should have a basic idea to operate the system.

2.4 Design Constraints

The language requirements are PYTHON, PHP,SQL. The content will be made non editable so that the end user cannot edit the marks or any other contents.

2.5 User Documentation

This software is simple to use and easy to understand. Users having basic Knowledge of computers will be able to use this software.

2.6 Assumptions and dependencies

- The system is dependent on the end – user to initiate the action of providing the information of the site of the accident or disaster.
- At least 1 user must be at the site of the accident or disaster
- When the information is transferred at the back-end then it will be provided only to affiliated bodies, which may be placed far from the location of the incident.
- If the affiliated bodies are placed at a far distance from the site of the incident, then it should consume some time for the affiliated bodies to reach at the site of accident.

3. System Requirements and analysis

The system requirements and analysis section introduces the numerous requirements of the system from the user's point of view. It also introduces a number of decisions that have been taken regarding implementation of the system.

3.1User Interface:

The user interface provided by the system should be a user friendly. A web browser-based interface must be provided for all the Disaster Management functionality. This sections lists the screens required and associated UI requirements.

3.2 Hardware Requirements:

2 GB RAM is required. Touch screen monitors – To facilitate navigation of the categorized management when processing acquisitions and also for normal management maintenance tasks. Standard, color laser printers – For printing donation receipts and reports,Mobile devices,Digital Cameras.

3.3 Software Requirements:

The system can be executed on a computer system having any version of windows operating system, mac OS, Ubuntu. The language requirements are PYTHON, PHP,SQL.

3.4 Software Interfaces

The system should support the following software interfaces

- Operating System.
- Latest Web Browsers.
- Google API (Application Program Interface).

3.5 Communications Interfaces

The system should support the following software interfaces

- Internet Connection.
- Mobile Connectivity.

4. Other Non-functional Requirements

4.1 Performance Requirements:

The system should allow you to add multiple updation records to the central database.

4.2 Security Requirements:

- The system should provide only authorized access to critical data
- The system should check data integrity for critical variables
- All fields should be validated before data is sent to the database

4.3 Portability Requirements:

The system can be executed on a computer system having any version of windows operating system, mac OS, Ubuntu.

4.4 Maintainability Requirements:

The system should be maintainable and an authorized user should be able to reset all options to default settings.

4 .5 Reliability Requirements

- After proper verification and validation, the data is committed to the central database.
- Appropriate backup procedures should be applied to prevent corruption and loss of data.

4.6 Usability Requirements

- A logical and user friendly interface helps the users to access the system easily.
- Error prevention should be supported by providing proper validation mechanisms for the data before actual submission to the database.

4.7 Software system Attributes

- The system should be equipped with current and archive databases.
- The system should present the information to the user clearly and precisely
- The system should allow entry of numbers, operands, special symbols, and letters of the alphabets.
- The system should send notifications to the user before committing user actions.

5. Appendices

5.1 References

- The SRS document created in this appendix follows the IEEE Guide to Software Requirement Specifications.
- Internet

CHAPTER 7

METHODOLOGY

7.1 Project Modules

RANDOM FOREST IN MACHINE LEARNING

Random forests is a supervised learning algorithm. It can be used both for classification and regression. It is also the most flexible and easy to use algorithm. A forest is comprised of trees. It is said that the more trees it has, the more robust a forest is. Random forests creates decision trees on randomly selected data samples, gets prediction from each tree and selects the best solution by means of voting. It also provides a pretty good indicator of the feature importance.
Random forests has a variety of applications, such as recommendation engines, image classification and feature selection. It can be used to classify loyal loan applicants, identify fraudulent activity and predict diseases. It lies at the base of the Boruta algorithm, which selects important features in a data set.

How does the algorithm work?
It works in four steps:

1) Select random samples from a given data set.
2) Construct a decision tree for each sample and get a prediction result from each decision tree.
3) Perform a vote for each predicted result.
4) Select the prediction result with the most votes as the final prediction.

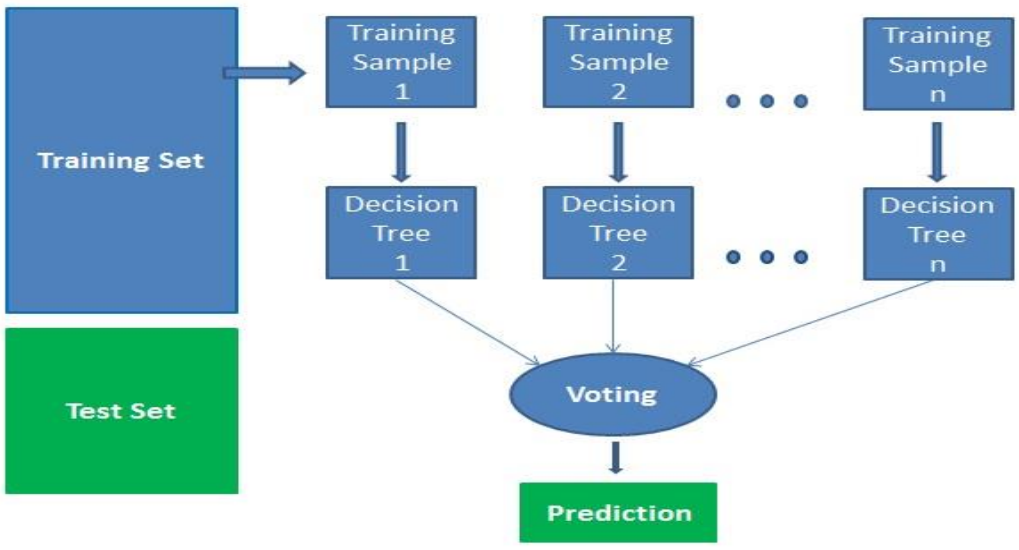

Figure 7.1 : Random Forest

DECISION TREE IN MACHINE LEARNING

A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.
A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules.
Tree based learning algorithms are considered to be one of the best and mostly used supervised learning methods. Tree based methods empower predictive models with high accuracy, stability and ease of interpretation. Unlike linear models, they map non-linear relationships quite well. They are adaptable at solving any kind of problem at hand (classification or regression). Decision Tree algorithms are referred to as CART (Classification and Regression Trees).

The strengths of decision tree methods are:

1) Decision trees are able to generate understandable rules.
2) Decision trees perform classification without requiring much computation.
3) Decision trees are able to handle both continuous and categorical variables.
4) Decision trees provide a clear indication of which fields are most important for prediction or classification.

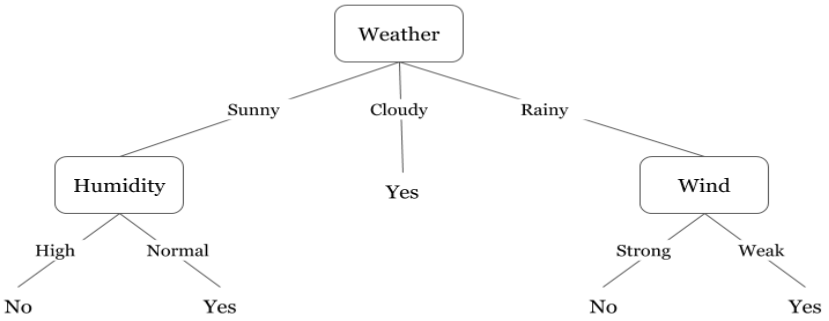

Figure 7.2 : Example of Decision Tree

7.2 Algorithm

Predicting Survival in the Titanic Data Set

Decision Tree Classifier in Python using Scikit-learn

The root node (the first decision node) partitions the data based on the most influential feature partitioning. There are 2 measures for this, Gini Impurity and Entropy.

Entropy The root node (the first decision node) partitions the data using the feature that provides the most information gain.

Information gain tells us how important a given attribute of the feature vectors is.

It is calculated as:

Information Gain=entropy(parent)–[average entropy(children)] Where entropy is a common measure of target class impurity, given as:

Entropy=Σi–pilog2pi where i is each of the target classes.

Gini Impurity Gini Impurity is another measure of impurity and is calculated as follows:

Gini=1–Σip2i Gini impurity is computationally faster as it doesn't require calculating logarithmic functions, though in reality which of the two methods is used rarely makes too much of a difference.

```
# data analysis and wrangling

import pandas as pd

import numpy as np

import random as rnd


# visualization

import seaborn as sns

import matplotlib.pyplot as plt

%matplotlib inline
```

```
# machine learning

from sklearn.linear_model import LogisticRegression

from sklearn.svm import SVC, LinearSVC

from sklearn.ensemble import RandomForestClassifier

from sklearn.neighbors import KNeighborsClassifier

from sklearn.naive_bayes import GaussianNB

from sklearn.linear_model import Perceptron

from sklearn.linear_model import SGDClassifier

from sklearn.tree import DecisionTreeClassifier


#index_col=('PassengerId' )
df = pd.read_csv('train.csv')
df.head()
df=df.set_index('PassengerId')
df.head()
```

We will be using Pclass, Sex, Age, SibSp (Siblings aboard), Parch (Parents/children aboard), and Fare to predict whether a passenger survived.

```
df = df[['Pclass', 'Sex', 'Age', 'SibSp', 'Parch', 'Fare', 'Survived']]
df.head(10)
```

We need to convert ‘Sex’ into an integer value of 0 or 1.

```
df['Sex'] = df['Sex'].map({'male': 0, 'female': 1})
df.head(4)
```

We will also drop any rows with missing values.You can use the isna() method then sum to count the NaN values.

```
df.info()
df.isna().sum()
df.shape
df.size
df.count()
df.describe()
df.boxplot(figsize=(15,15))
df.corr()
```

We will also drop any rows with missing values.

```
df = df.dropna()
df.count()
```

```
Create Target Varibale Dependent & Independent
X = df.drop('Survived', axis=1)
y = df['Survived']
X.head()
y.head()

To show Correlations Among Features
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
df = pd.read_csv('train.csv')
plt.subplots(figsize = (15,8))
sns.heatmap(df.corr(), annot=True,cmap="PiYG")
plt.title("Correlations Among Features", fontsize = 20);
plt.subplots(figsize=(10,8))
sns.kdeplot(df.loc[(df['Survived'] = = 0),'Pclass'],shade=True,color='r',label='Not Survived')
ax=sns.kdeplot(df.loc[(df['Survived'] = = 1),'Pclass'],shade=True,color='b',label='Survived' )
labels = ['First', 'Second', 'Third']
plt.xticks(sorted(df.Pclass.unique()),labels)
plt.subplots(figsize=(15,10))
ax=sns.kdeplot(df.loc[(df['Survived'] == 0),'Fare'],color='r',shade=True,label='Not Survived')
ax=sns.kdeplot(df.loc[(df['Survived'] == 1),'Fare'],color='b',shade=True,label='Survived' )
plt.title('Fare Distribution Survived vs Non Survived',fontsize=25)
plt.ylabel('Frequency of Passenger Survived',fontsize=20)
plt.xlabel('Fare',fontsize=20)
```

Analyze by visualizing data

Now we can continue confirming some of our assumptions using visualizations for analyzing the data. Let us start by understanding correlations between numerical features and our solution goal (Survived).

A histogram chart is useful for analyzing continous numerical variables like Age where banding or ranges will help identify useful patterns. The histogram can indicate distribution of samples using automatically defined bins or equally ranged bands. This helps us answer questions relating to specific bands (Did infants have better survival rate?)

Note that x-axis in histogram visualizations represents the count of samples or passengers.

```
g = sns.FacetGrid(df, col='Survived')
g.map(plt.hist, 'Age', bins=20)
```

Observations:

1.Infants (Age <=4) had high survival rate.

2.Oldest passengers (Age = 80) survived.

3.Large number of 15-25 year olds did not survive.

4. Most passengers are in 15-35 age range.

Correlating numerical and ordinal features
We can combine multiple features for identifying correlations using a single plot. This can be done with numerical and categorical features which have numeric values.

Observations

Pclass=3 had most passengers, however most did not survive. Confirms our classifying assumption.
Infant passengers in Pclass=2 and Pclass=3 mostly survived. Further qualifies our classifying assumption.
Most passengers in Pclass=1 survived. Confirms our classifying assumption.
Pclass varies in terms of Age distribution of passengers.

```
# grid = sns.FacetGrid(train_df, col='Pclass', hue='Survived')
grid = sns.FacetGrid(df, col='Survived', row='Pclass', height=2.2, aspect=1.6)
grid.map(plt.hist, 'Age', alpha=.5, bins=20)
grid.add_legend();
```

Train & Test Data

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=1)
from sklearn import tree
model = tree.DecisionTreeClassifier(criterion='gini')
model
```

Defining some of the attributes like max_depth, max_leaf_nodes, min_impurity_split, and min_samples_leaf can help prevent overfitting the model to the training data.

First we fit our model using our training data.

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

Then we score the predicted output from model on our test data against our ground truth test data.

```
y_predict = model.predict(X_test)
X_test.head()
y_predict
from sklearn.metrics import accuracy_score
accuracy_score(y_test, y_predict)
```

We see an accuracy score of ~83.2%, which is significantly better than 50/50 guessing.
Let's also take a look at our confusion matrix:

```
from sklearn.metrics import confusion_matrix
pd.DataFrame(
    confusion_matrix(y_test, y_predict),
    columns=['Predicted Not Survival', 'Predicted Survival'],
    index=['True Not Survival', 'True Survival']
)
```

Decision Tree

The root node, with the most information gain, tells us that the biggest factor in determining survival is Sex.
If we zoom in on some of the leaf nodes, we can follow some of the decisions down.
We have already zoomed into the part of the decision tree that describes males, with a ticket lower than first class, that are under the age of 10.
The impurity is the measure as given at the top by Gini, the samples are the number of observations remaining to classify and the value is the how many samples are in class 0 (Did not survive) and how many samples are in class 1 (Survived).

Let's follow this part of the tree down, the nodes to the left are True and the nodes to the right are False:

We see that we have 19 observations left to classify: 9 did not survive and 10 did. From this point the most information gain is how many siblings (SibSp) were aboard. A. 9 out of the 10 samples with less than 2.5 siblings survived. B. This leaves 10 observations left, 9 did not survive and 1 did. 6 of these children that only had one parent (Parch) aboard did not survive. None of the children aged > 3.5 survived Of the 2 remaining children, the one with > 4.5 siblings did not survive.

Decision trees are a great tool but they can often overfit the training set of data unless pruned effectively, hindering their predictive capabilities.

CHAPTER 8

RESULTS

8.1 Working Of The System (Screen Snapshots and details of the same)

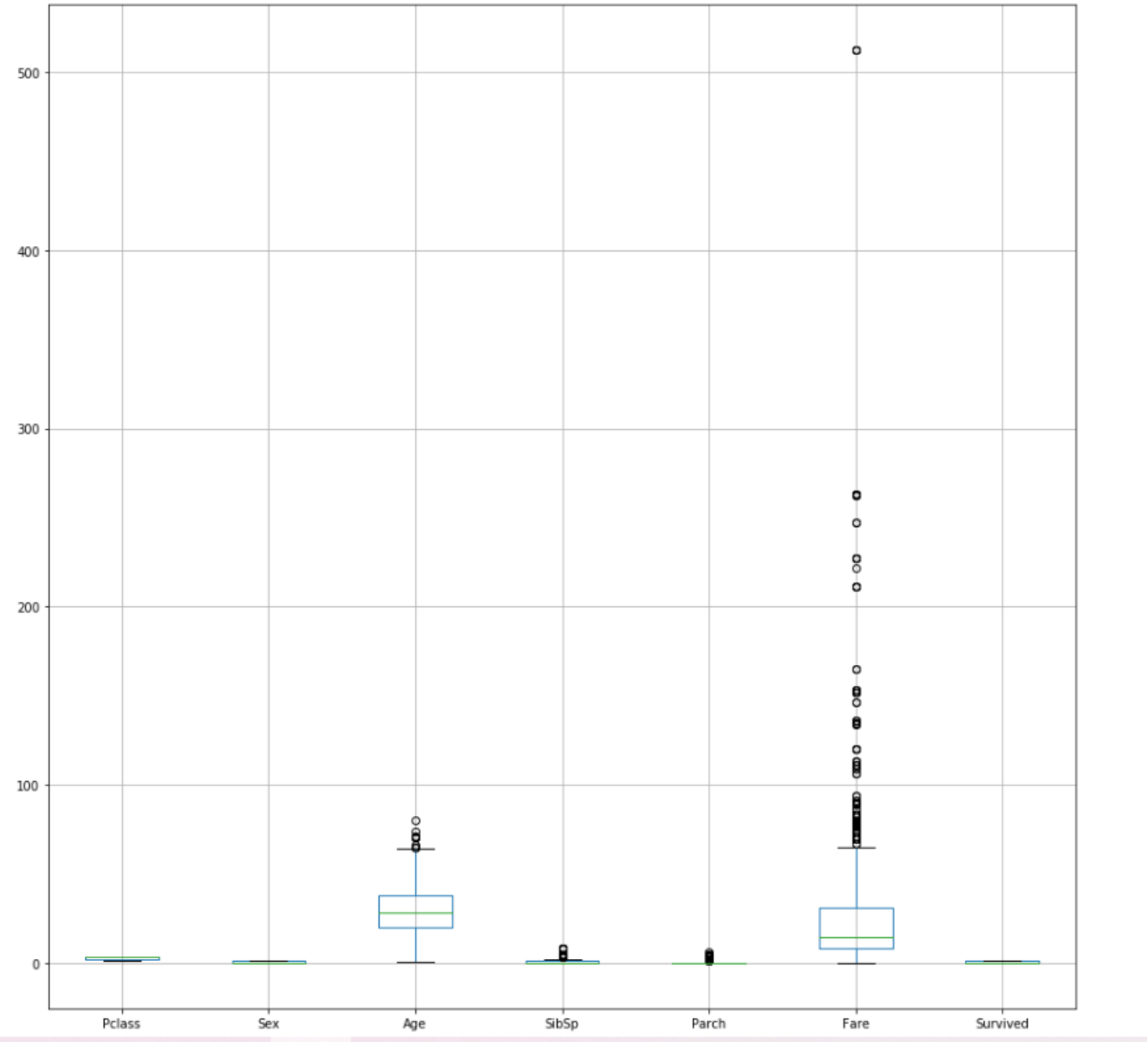

Figure 8.1 : Box-plot

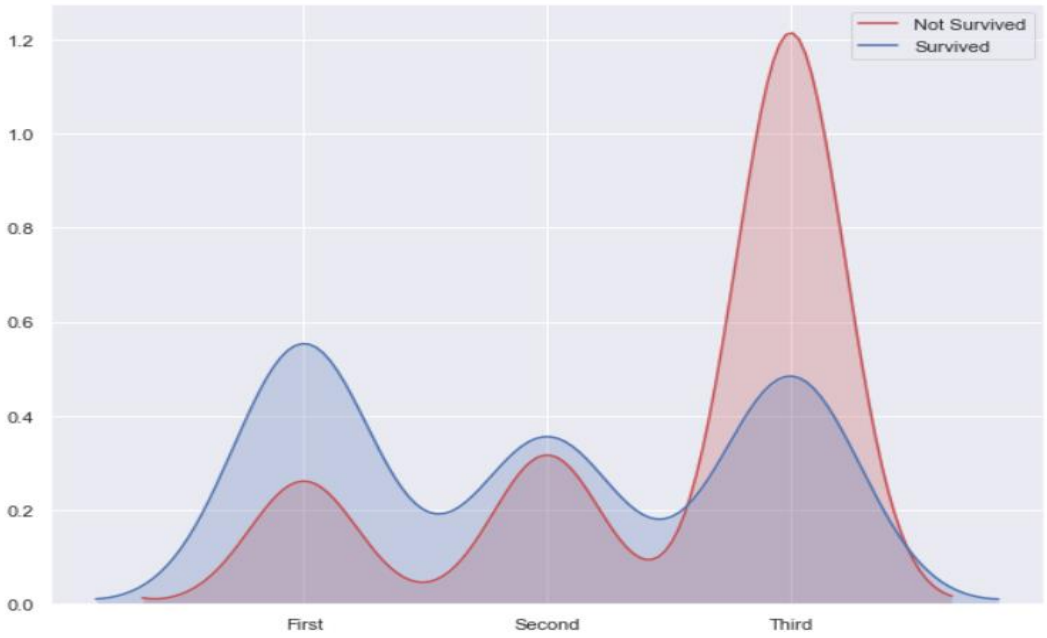

Figure 8.2 : Survival Rate of Class

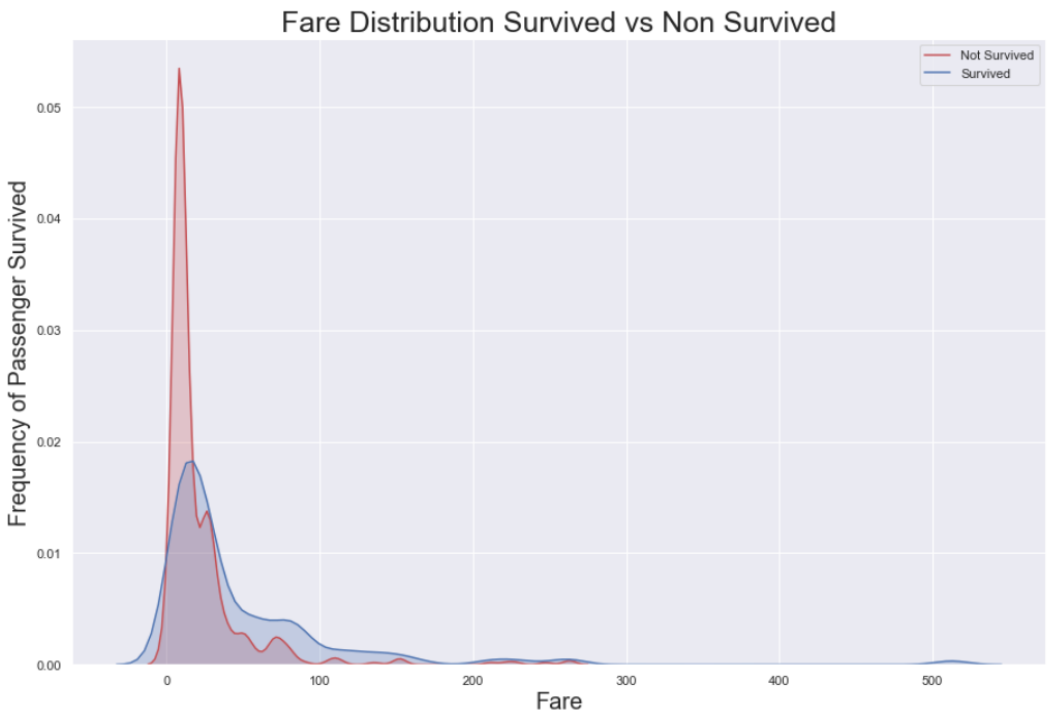

Figure 8.3 : Fare Distribution

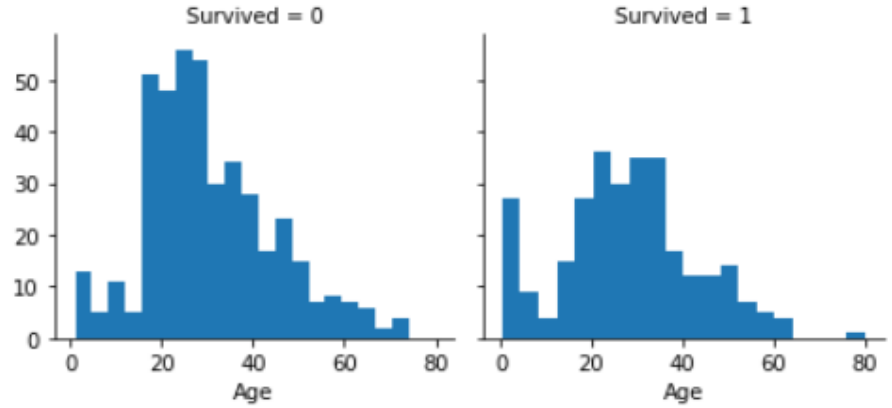

Figure 8.4 : Survival Rate of Age

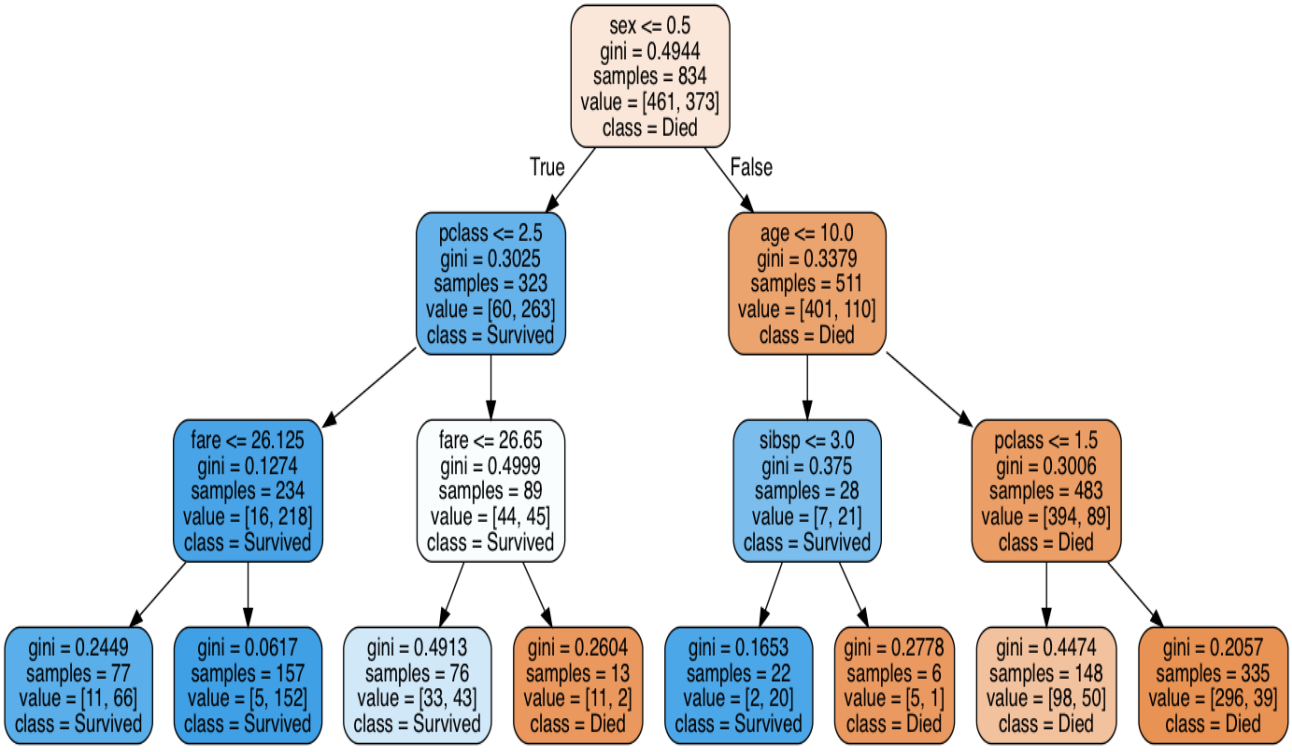

Figure 8.5 : Decision Tree

CHAPTER 9

CONCLUSION

In analyzing the Titanic passenger's data, we have used a predictive analytical methodology (Random Forest) that computed 5 models with each 100 trees in a few seconds to come up with different predictive models. We conclude that both the results in Weka and at the public leader-board are high (77% - 82%) and that the prediction error do not differ more than around 5% with each other. The differences in the results within Weka are about 3.5% and the differences in the results at the public leader-board are less than 1%. Further research can be done by either trying out more theories to divide the data set or by using other analytics to analyze the data set to validate the results, such as neural networks, Bayes functions or other decision tree models. The analytical tool Weka has been found useful in converting the data, preprocessing the data, exploring the data and generating high predictive models that can also be easily applied to the provided data sets from the Kaggle competition on the Titanic passengers data. The user friendly interface allows for rapid testing different models once the requirements of the data sets are met. However, the process of converting and preprocessing the data was the most problematic in the study. The issues in these stages were mostly caused by compatibility 9 issues in the train and test data set and syntax differences. In this study we have also noticed 3 general issues when conducting predictive analysis, namely: the bias-variance dilemma, the accuracy-interpret ability dilemma and the exploration-exploitation dilemma.

REFERENCES

- Ali, J., Khan, R., Ahmad, N., & Maqsood, I. (2012). *Random forests and decision trees*. International Journal of Computer Science Issues (IJCSI).
- British Trade Administration. (1911). *British Wreck Commissioner's Inquiry Report*. Retrieved from http://www.titanicinquiry.org/ BOTInq/BOTReport/botRepBOT.php
- Chen, H., Chiang, R. H., & Storey, V. C. (2012). Business Intelligence and Analytics: *From Big Data to Big Impact*. MIS quarterly, 36(4), 1165- 1188.
- Crepinsek, M., Liu, S.-H., and Mernik, M. 2013. *Exploration and exploitation in evolutionary algorithms: A survey*. ACM Comput. Surv. 45, 3, Article 35 (June 2013), 33 pages.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (Second ed.). Springer.
- James, G., Witten, D., Hastie, T., Tibshirani, R. (2013). *An Introduction to Statistical Learning*. Springer
- Van den Berg, J. (2014). *Lecture on: Combining fuzzy and statistical uncertainty: probabilistic fuzzy systems and their applications.* Delft University of Technology.
- Wikispace. (2015). *Making predictions.* Retrieved from http://weka.wikispaces.com/ Making+predictions
- Witten, I.H., Frank, E., Hall, M.A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques (Third Edition)*, Pages 3-38. Boston. ISBN 9780123748560.