LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

CS6P05NI

Final Year Project Disease Prediction (E Check-up)

Year and Semester

2019-2020

Autumn

Submitted By:

Rishav Dahal

17031146(London Met ID)

Group: C8

Submission Date: 7th Jan 2020

Word Count: 2,932

Submitted To:

Internal Supervisor

Subekshya Shrestha

External Supervisor

Ishwor Shrestha

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Abstract:

This project is developed as an informative system which maps the possible symptoms that a person is facing into the most probable disease. It is a Machine Learning application which makes use of multiple algorithms to draw a conclusions based on the user symptoms. The user can select multiple symptoms and they are collectively directed towards one of each disease based on algorithms. Multiple predictions are made based on the data set by which the model is trained. Classification method is used and three different algorithms with different accuracy and complexity are used and a final prediction is obtained with acceptable error rate. Data is one of the major players in this project and data are accessed through a study conducted at Columbia University. This project aims to provide an open web based application to general people who can pre analyze their symptoms before visiting to a medical professional.

Contents

1) Introduction:

1.1 Overview And Motivation

With wide coverage of Internet technology today, people around the globe are familiar with search engines and internet surfing techniques. This upgrade in technology has many good aspects but some downsides also. People often tend to search every problem they have on the internet. Internet Technology have updated many fields and made the working easier.

Talking precisely about Medical field, technology is improving medical care with the use of Health wearable, Artificial organs, Telemedicines, etc. Talking about the other side of medical technology, we receive unnecessary search results in internet which makes us think once about that matter.

1.2 Problem Statement

When we type in the basic symptoms that we might be observing for few days in Google or Yahoo or any other search engines, we often get through unnecessary conclusions or predictions that mostly ends in death. Search engines are vague and provides both authorized and unauthorized results.

eCheckUp is intended to solve that particular problem by pre-analyzing the symptoms to conclude the possibility of one or more particular disease so that we can proceed into further treatment consulting medical professionals.

1.3 Objective Of The Project

1.3.1 General Objective

To implement different machine learning algorithms into a specific data set by training a model and compare the outputs of the algorithms with different accuracy rate and determine the best algorithm into the scenario.

1.3.2 Specific Objective

To develop a web application which maps your occurring symptoms into one or more possible disease this helps to minimize vague disease prediction. To know where your symptoms are leading to and proceed with the necessary checkups and meet the specific medical professionals.

1.4 Scope Of The Project

The scope of the project are the general people who have certain recognized symptoms and are searching for any possible disease prediction. Displaying them some particular disease based on their symptoms is the main working of the project.

CHAPTER 2: BACKGROUND RESEARCH

2.1) About the client/end-user

For further research purpose, I had generated a list of questions using Google forms. This made me clear what I am missing in the project. Following are the statists of form:

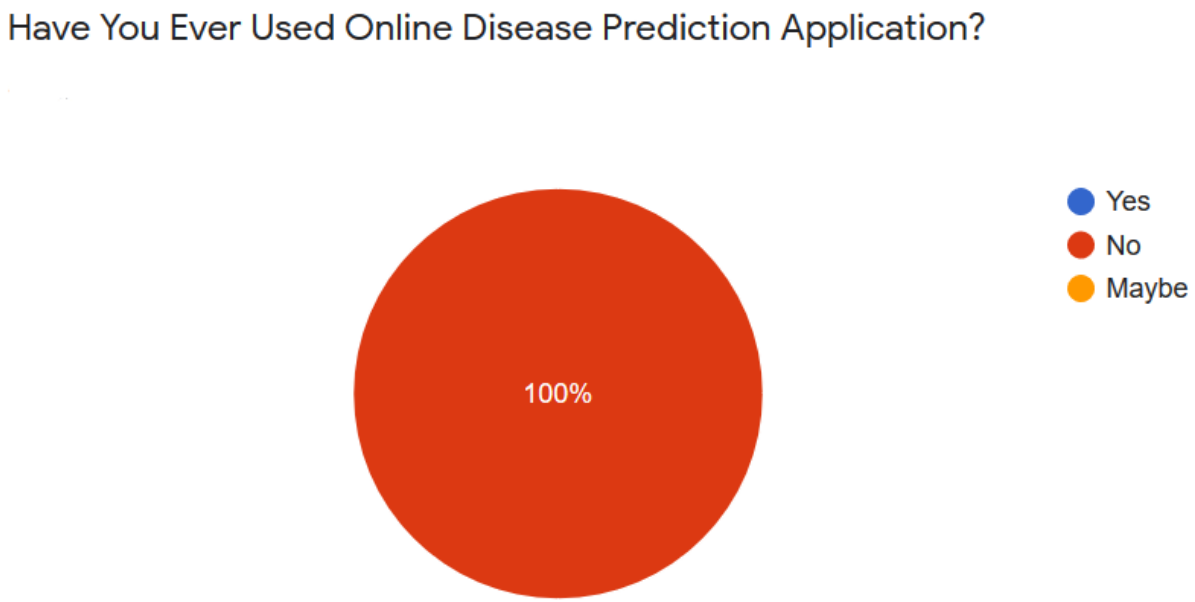


Figure 1 Survey 1

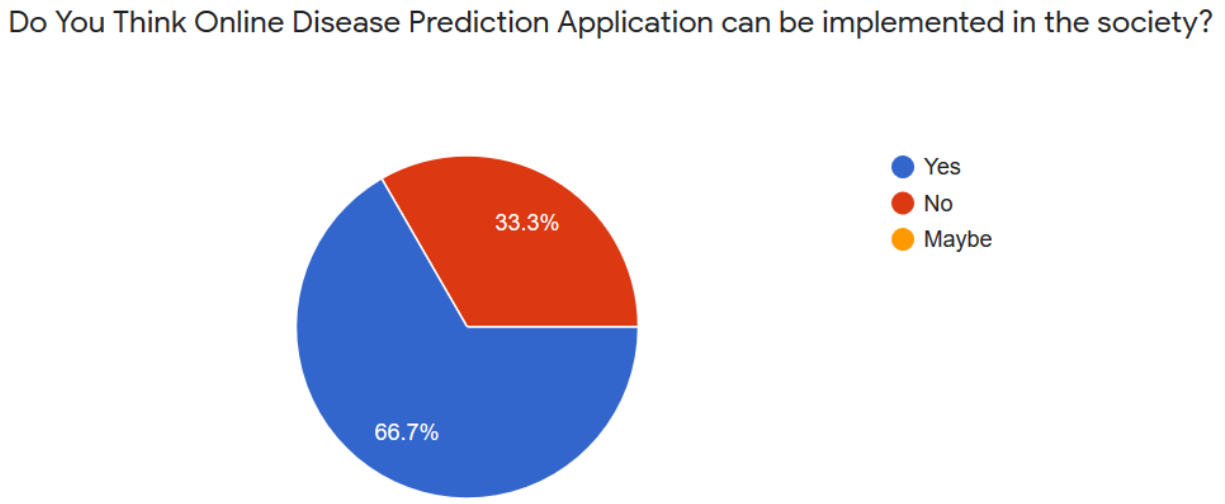


Figure 2 Survey 2

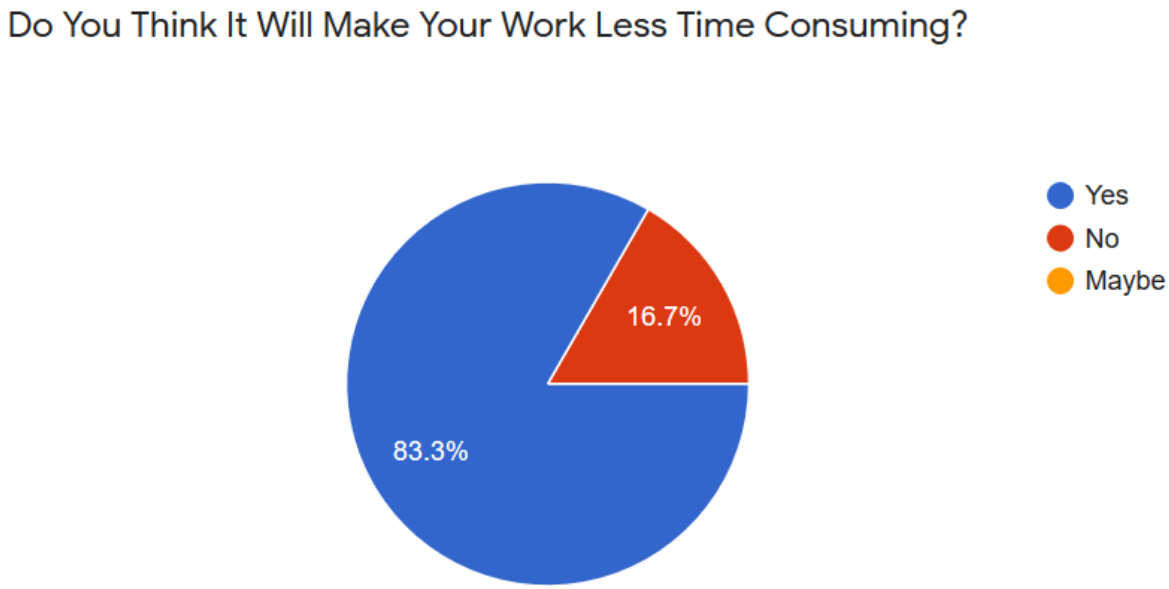


Figure 3 Survey 3

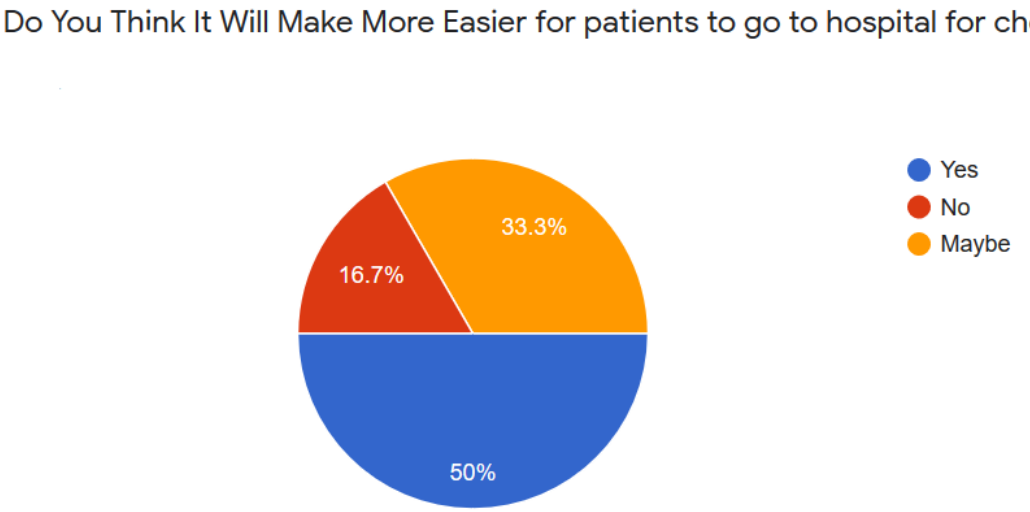


Figure 4 Survey 4

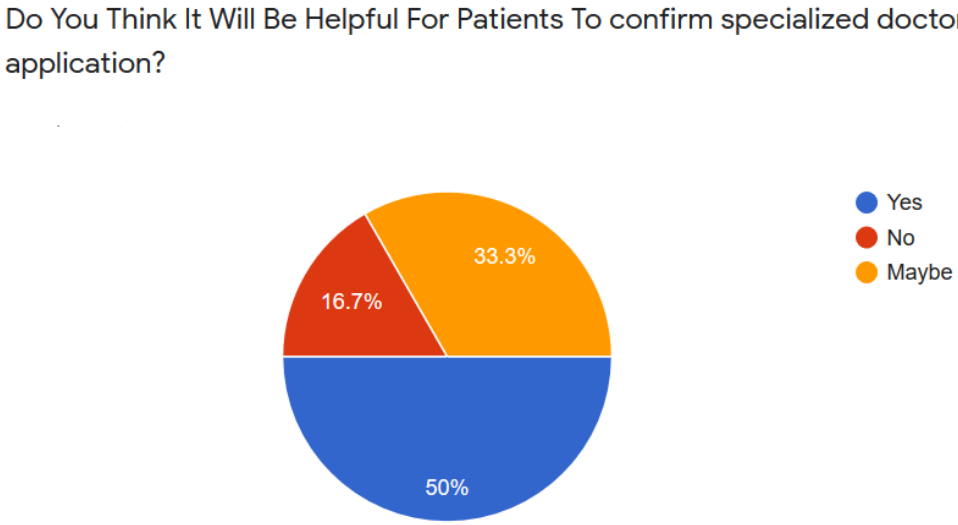


Figure 5Survey 5

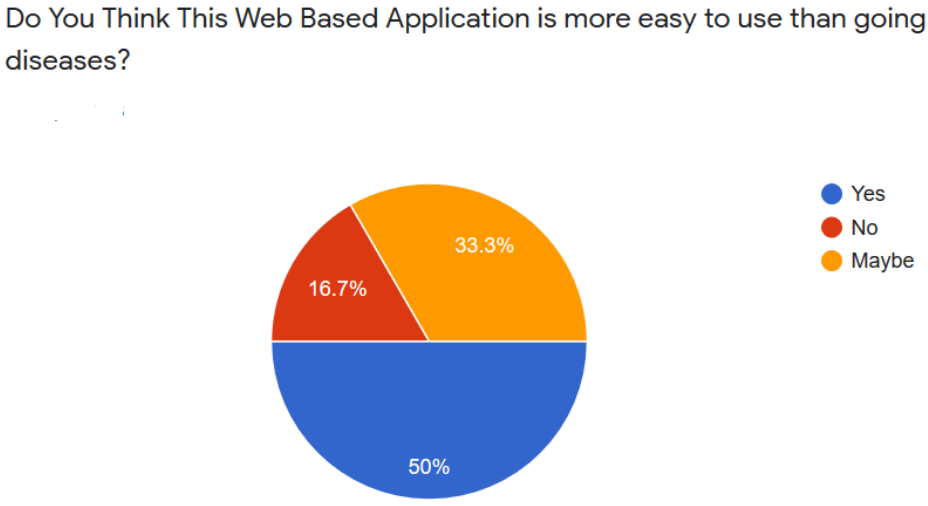


Figure 6 Survey 6

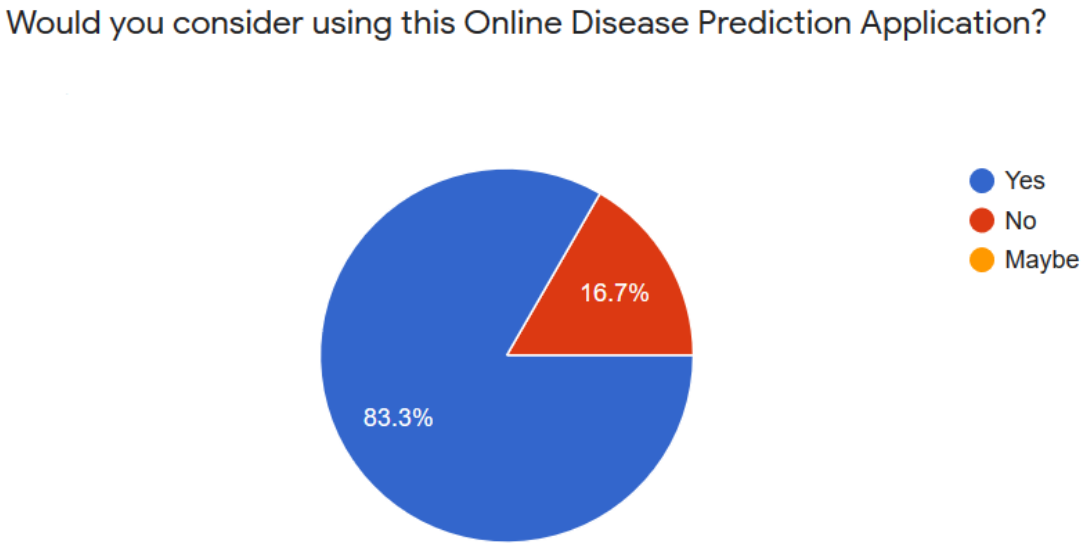


Figure 7 Survey 7

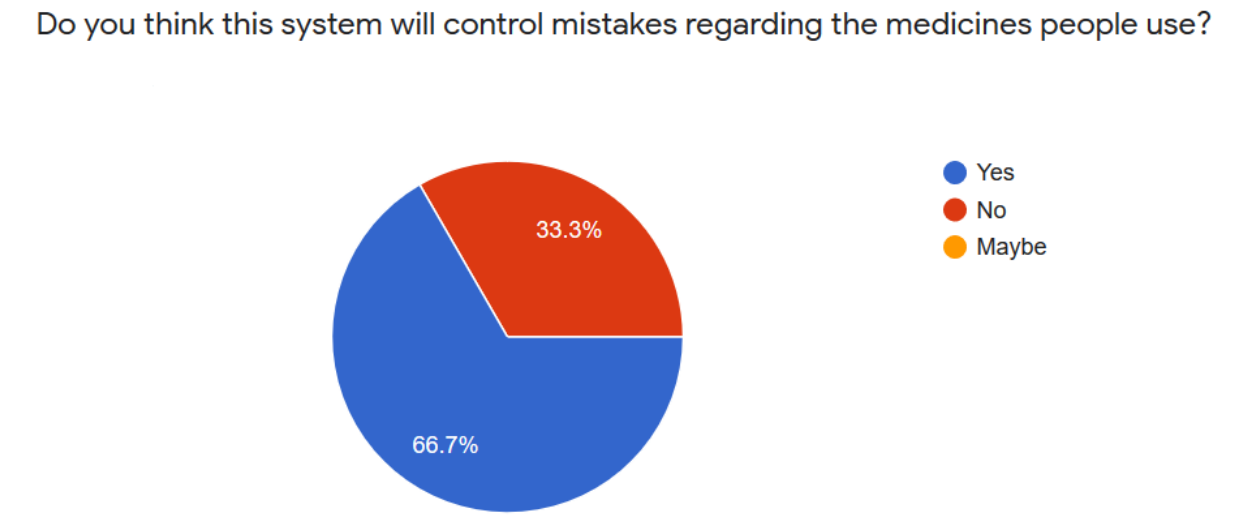


Figure 8 Survey 8

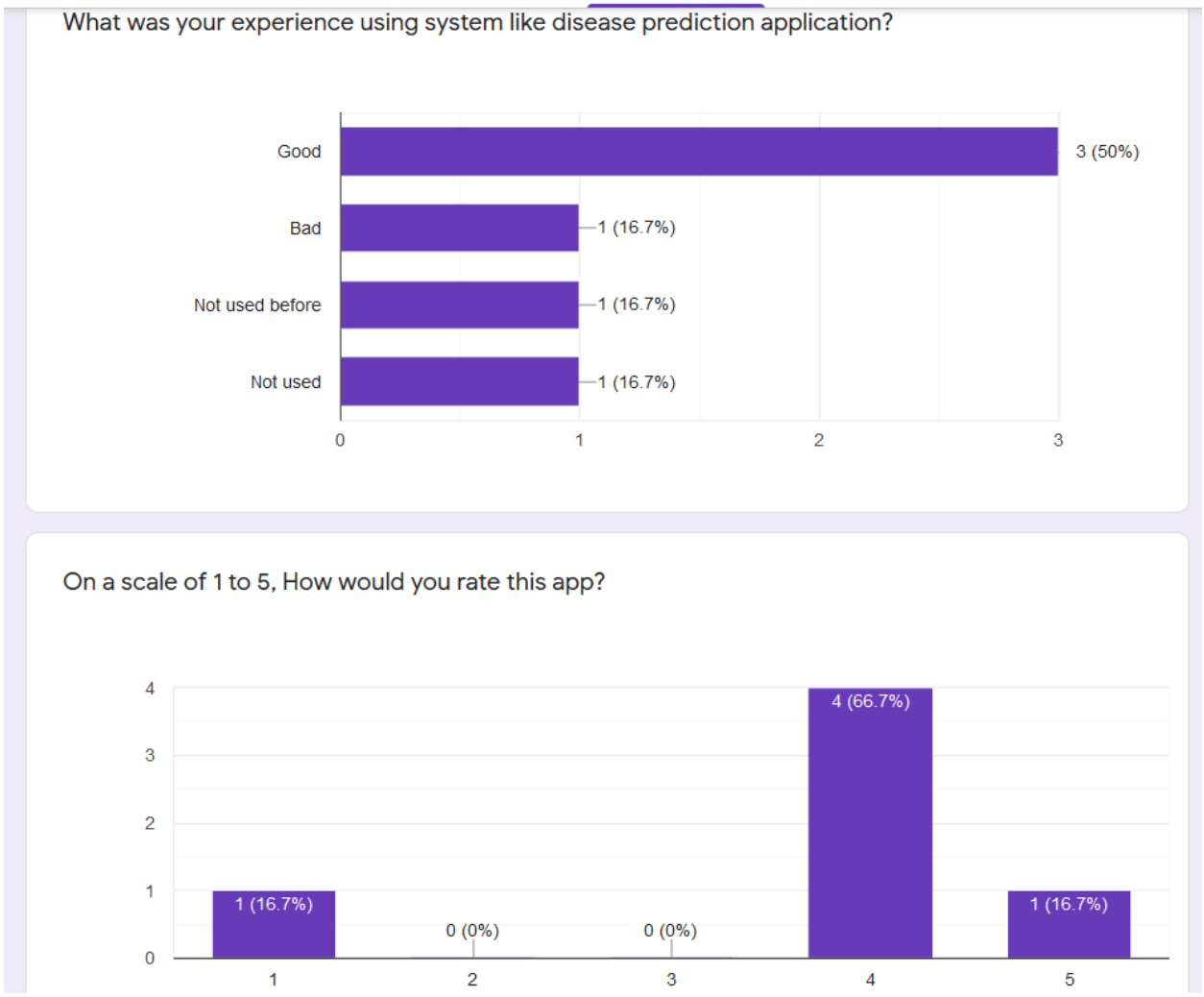


Figure 9 Survey 9&10

2.2. Literature Review

Web MD Symptom Checker (https://symptoms.webmd.com/default.htm#/info)

There is an online Symptom Checker application called WebMD Symptom Checker. It consists a full data set of symptoms and probable disease with conditions and treatments. eCheckUp is similar to WebMD Symptom Checker.

2.3. Current System

Talking about the current system i.e. Web MD Symptom Checker, it analyzes the symptoms of a user based on various factors as age, gender, symptoms, questions, conditions and details. It is a fully functioning application with big data set than eCheckUp. This project is a mini project inspired by the Web MD Symptom Checker.

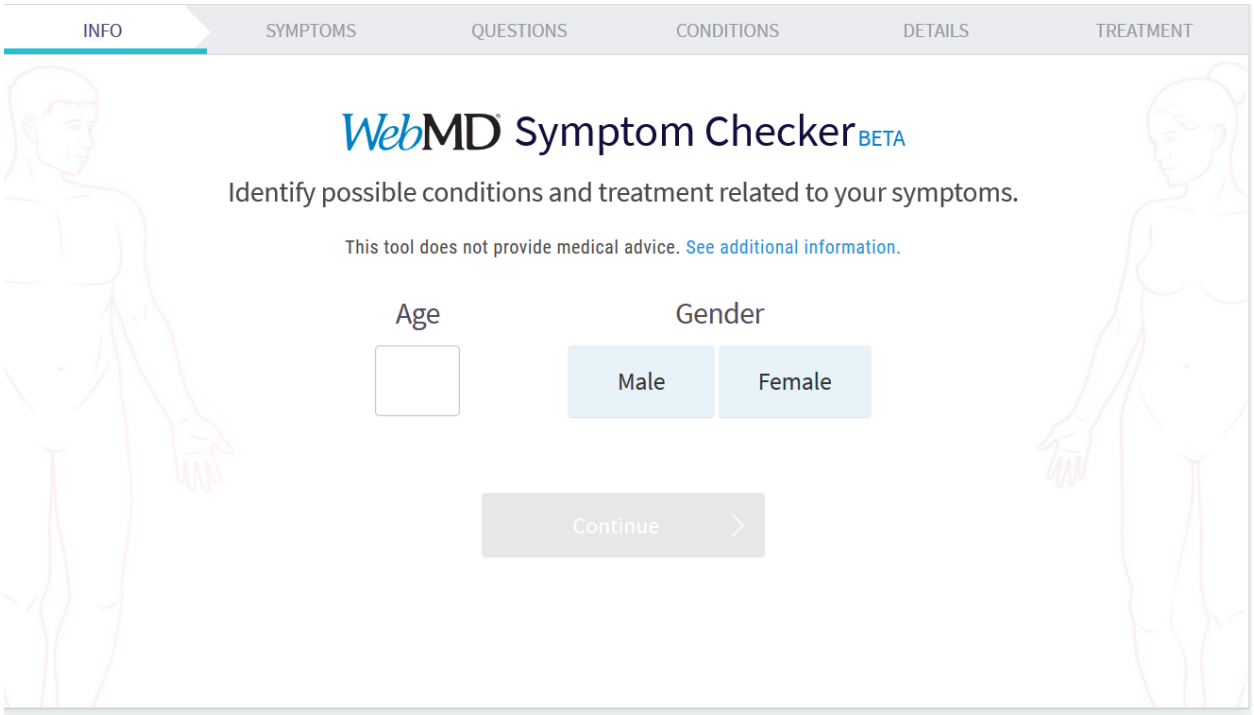


Figure 10 Symptom Checker UI (Web MD)

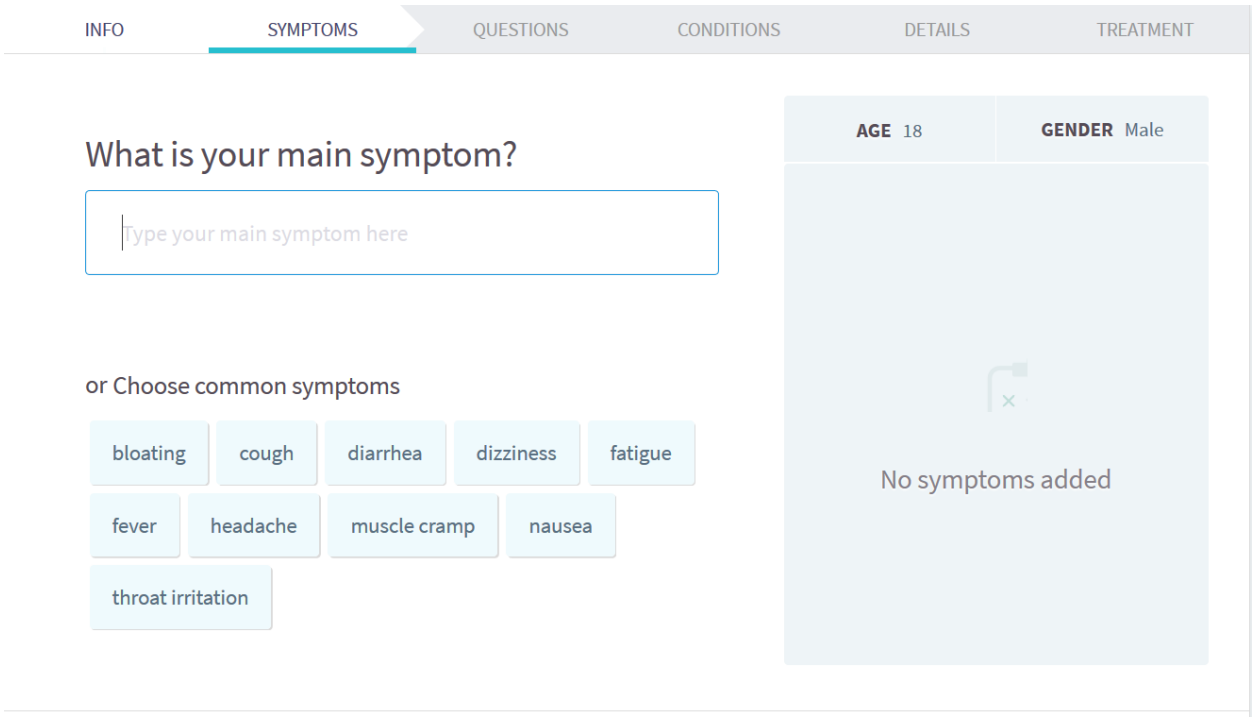


Figure 11 Choosing Symptom (WebMD)

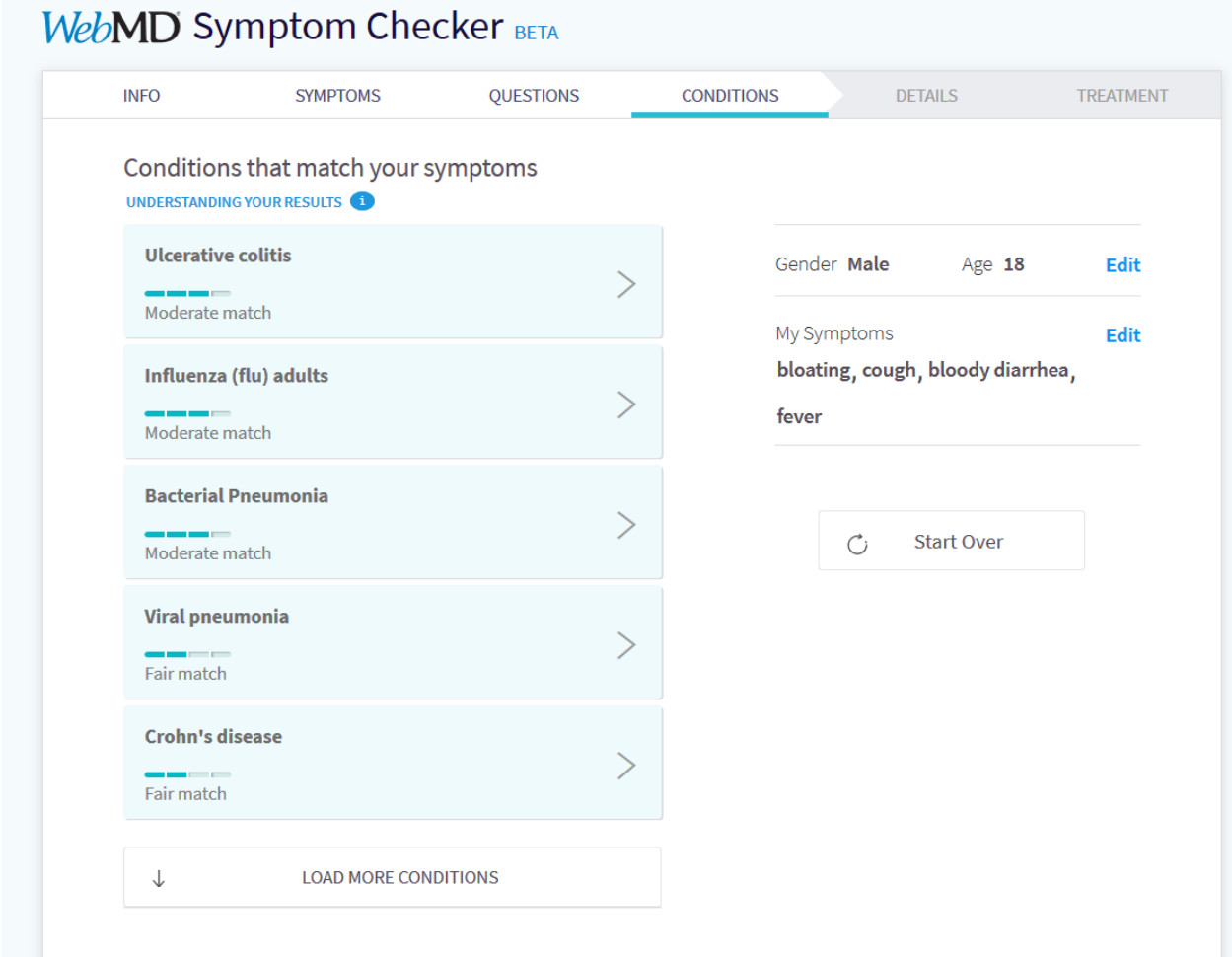


Figure 12 Result (Web MD)

2.4. The Problem With Current System

Since Web MD Symptom Checker is a complete working web application, there are no reported problem in it. It clearly states that it does not provide medical advice but shows some treatment methods. Web MD is in beta version and it does not show which data set it is making use of and accuracy. This is the problem with the current system but as a whole, it is a practical application with easy to use GUI and Treatment feature included.

CHAPTER 3: SPECIFICATION AND DESIGN

3.1. Requirement Elicitation And Analysis

The basic requirement of this project is a computer system which allows python applications to run. This project is based up on Python3, so the computer must support Python program to be executed successfully. Data set and training of the model is also major requirement of this project. The data set is divided into two parts: Training data set and Testing data set. The training data set is used to train the model while the testing data set is used to test the model. The accuracy obtained is approximately 95%.

3.1.1. Functional Requirement

- Functional requirements of eCheckUp can be:
  - The User should open the web application
  - The system should ask for the Name of the user and 5 symptoms as input
  - The user should click on predictions to create 3 predictions based on 3 different algorithms
  - The user should be able to understand the predictions and understand them for further processing and also save the data to save the user info in the system.
  - Python3 and libraries as numpy, tkinter and pandas.

3.2.2. Non-Functional Requirement

Similarly, some of the Non-functional requirements are:

- The data must be in CSV format to train the model
- The format of data must be in correct order as it is places
- The GUI must be designed correctly to provide correct output

3.3. System Design

The design of this system is easy compared to other system. This project do not contain any databases so diagrams like ER, Relational can be omitted. The system is designed using web markup languages and the processing part was done using python while GUI is made on a framework of python called as Tkinter.

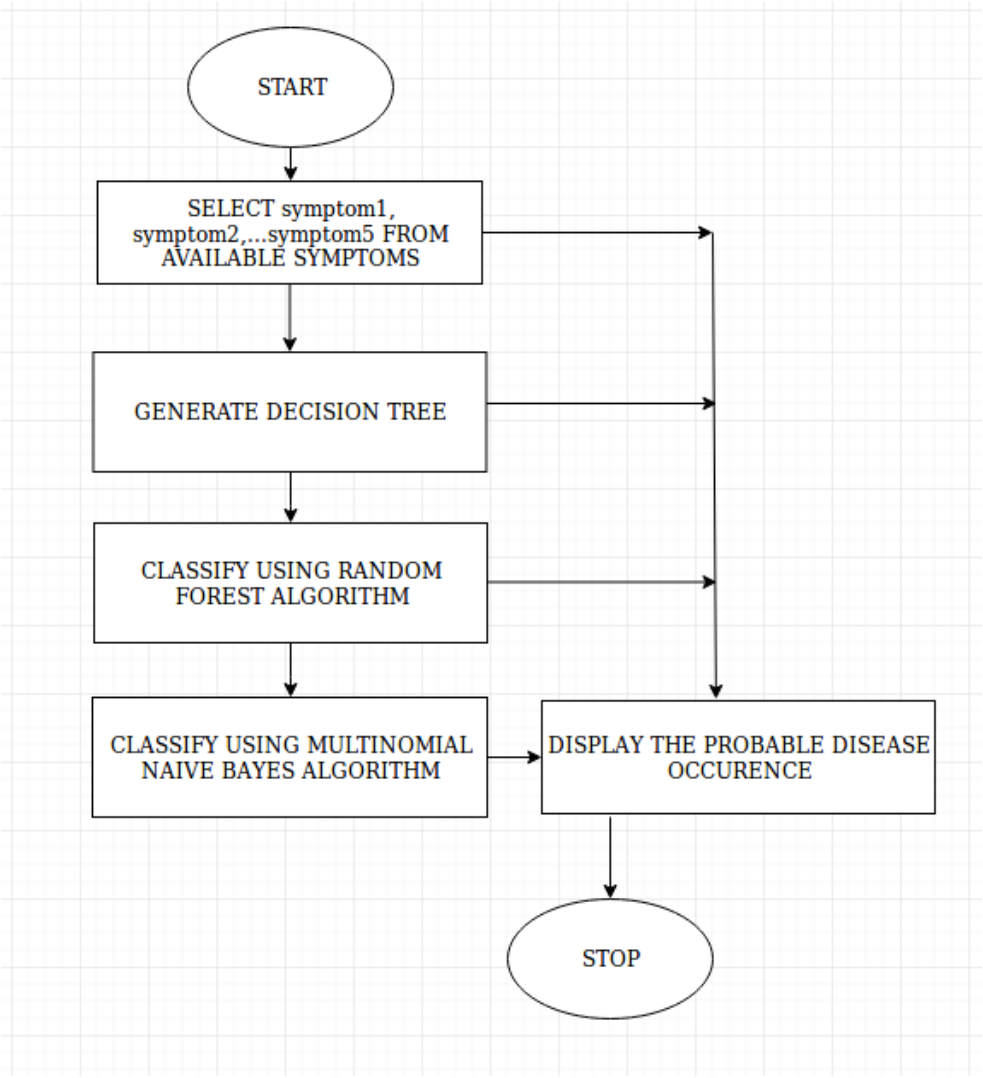


Figure 13 Work Flow Diagram

This is the work flow chart of the system. The user can select upto 5 symptoms from a given list of symptoms and can proceed to obtain the predictions. The program calls 3 functions: Decision Tree, Random Forest and Naive Bayes algorithms generate the outputs and the probable disease based on all of them individually is displayed. There is a save option which enters the user info into a file.

3.3.1. Use Case Diagram

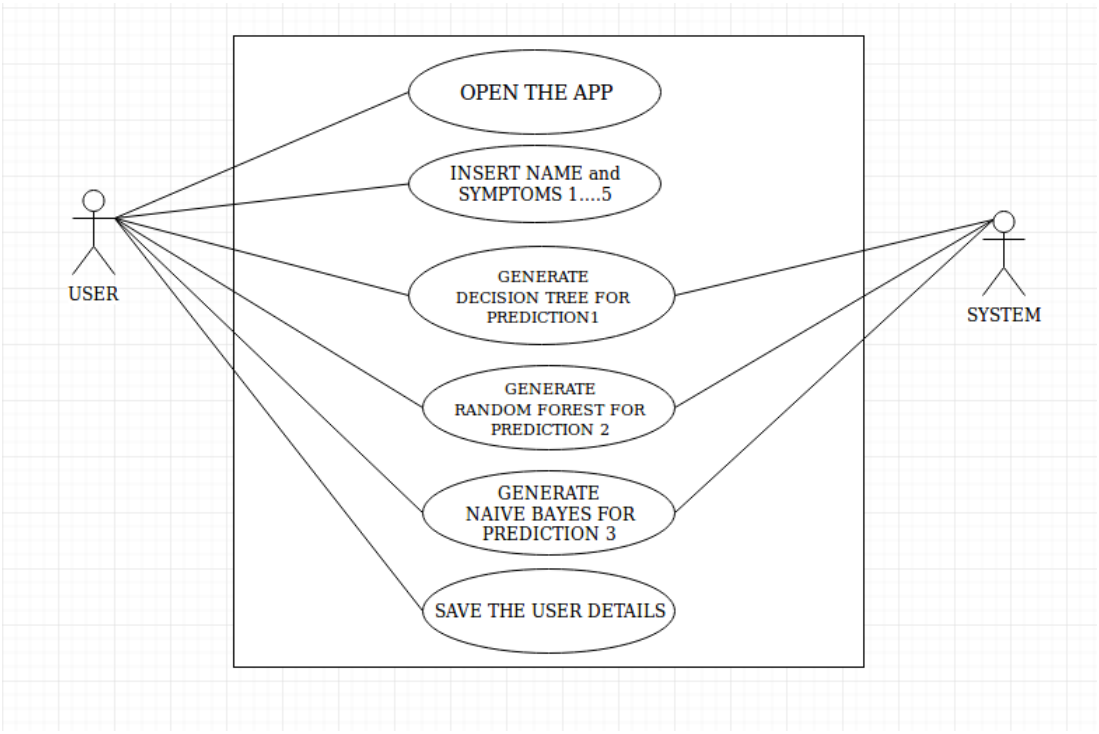


Figure 14 Use case Diagram

There are two actors in this project: User and System. User has access to the GUI part of the project while System can generate the predictions that can be viewed by the User. The above is the use case diagram of the project.

CHAPTER 4: IMPLEMENTATION AND EVALUATION

4.1. Tool And Technology

The tools and technology used in this project along with their function are listed below:

- Python3 is used as the prime coding language
- Python Libraries as numpy, pandas and tkinter is used to ease the working of the project
- CSV format is used to place the training data, testing data and save records of patients
- Any IDE that supports python is used to develop the project.

4.2. Implementation

The Implementation of this project is all to deal with mapping user inputs i.e. symptoms into probable disease with an acceptable error rate. The Classification method ML will used including three different algorithms: Decision Tree, Random Forest and Naive Bayes. These all algorithms have their different complexity measure and accuracy level. Working with three algorithms will help to compare the outputs on a given set of input. The main challenge is to obtain accurate result based on the trained model using the data set.

A **Decision Tree** is a map of the possible outcomes of a series of related choices. It allows an individual or organization to weigh possible actions against one another based on their costs, probabilities, and benefits. They can be used either to drive informal discussion or to map out an algorithm that predicts the best choice mathematically. (lucidchart, 2019)

The **Random Forest** algorithm combines multiple algorithm of the same type i.e. multiple decision trees, resulting in a forest of trees, hence the name "Random

Forest".The following are the basic steps involved in performing the random forest algorithm:

1. Pick N random records from the dataset.
2. Build a decision tree based on these N records.
3. Choose the number of trees you want in your algorithm and repeat steps 1 and 2.
4. In case of a regression problem, for a new record, each tree in the forest predicts a value for Y (output). The final value can be calculated by taking the average of all the values predicted by all the trees in forest. Or, in case of a classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote. (Synced, 2019)

A **Naive Bayes** classifier is an algorithm that uses Bayes' theorem to classify objects. Naive Bayes classifiers assume strong, or naive, independence between attributes of data points. Popular uses of naive Bayes classifiers include spam filters, text analysis and medical diagnosis.

A naive Bayes classifier is not a single algorithm, but a family of machine learning algorithms that make uses of statistical independence. These algorithms are relatively easy to write and run more efficiently than more complex Bayes algorithms. The formula used for naive Bayes is:

$$P(c|x) = (P(x|c) * P(c)) / P(x)$$ (Brownlee, 2019)

These three algorithms will be implemented in the project with their functions to predict the disease.

4.3. Evaluation

This application is intended to show the disease from symptoms and it was made so. It contain a simple and easy-to-use design which has no difficulty in using. Users after inserting up to 5 symptoms by selecting from the list are able to predict the probable disease they might be suffering from. This application provides no medical evidences but however it could be as a pre requisite before visiting to any medical organization or professionals. Talking about one of the predictions that could be made was if you are suffering from mild fever, belly pain, loss of smell, congestion and diarrhoea can be possibly typhoid and user can proceed to test for it by visiting a doctor.

5) Development

For the development phase, I have considered three methodologies i.e. Waterfall Model, Agile Methodology and Iterative Model.

5.1) Waterfall Model:

The Waterfall Model was the first Process Model to be introduced. It is very simple to understand and use. In a Waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases. The waterfall model is the earliest Software development life cycle approach that was used for software development.

In the waterfall approach, the whole process of software development is divided into separate phases. The outcome of one phase acts as the input for the next phase sequentially. This means that any phase in the development process begins only if the previous phase is complete. The waterfall model is a sequential design process in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Conception, Initiation, Analysis, Design, Construction, Testing, production/Implementation, and Maintenance. (toolsqa, 2018)

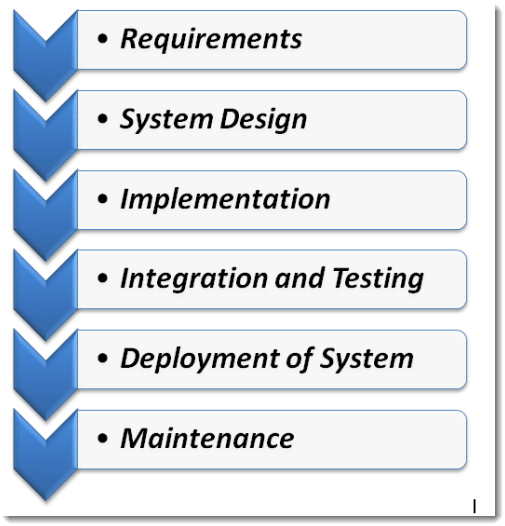


Figure 15 Waterfall Model

5.2) Scrum Methodology:

Scrum Methodology is a simple framework that facilitates team collaboration on complex projects. It is that scrum is easy to understand but it is hard to master.

Certainly, such type of a methodology is named after the cooperative rugby strategy in which teams gain distance by passing the ball back and forth. Although Scrum is actually an "inspect and adapt" framework for software development and not a project management methodology, per se, it has applications for other types of projects, and thus is considered by many to be an agile project management tool. Another good thing about it is that it minimizes peril by creating software in short time boxes, known as iterations, which happen to last from one week to one month. (projectmanager, 2020)

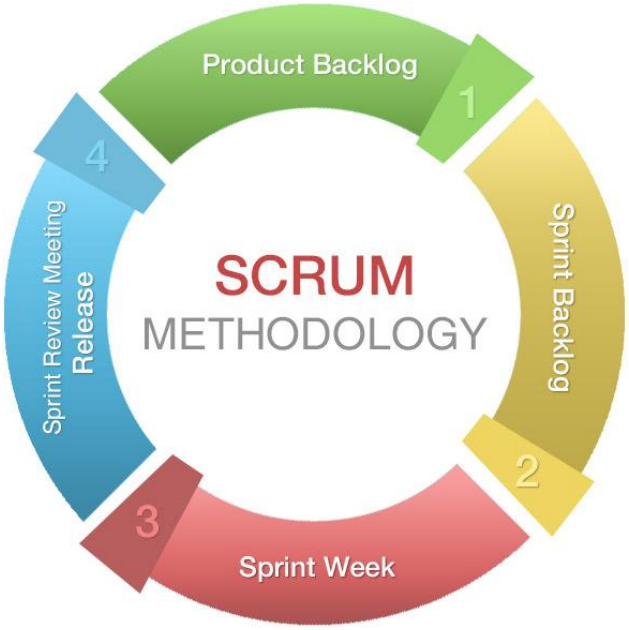
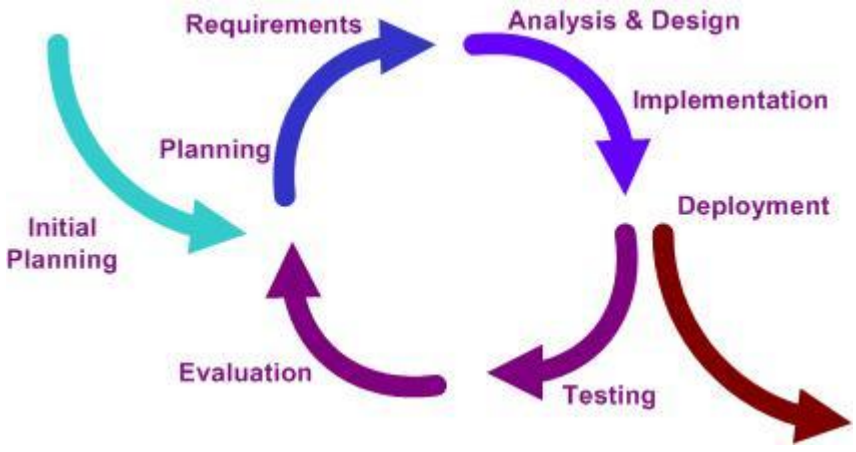


Figure 16 Scrum Method

5.3) Iterative Model:

I have chosen to utilize iterative model for my project. My tasks don't endeavour to begin with full particular of necessities. Rather improvement starts by determining and execute portions of the application. Every one of web customers will be assessed so as to improve and recognize further prerequisites in each stage. The procedure is then repeated, creating better server and the customers. My undertaking is evaluated to be done in 24 weeks.



(Klinger, n.d.)

Figure 17 Iterative Model

Considering an iterative life cycle model, it consists of repeating the following four phases:

- Requirements phase
- Design phase
- Implementation Phase
- Review Phase

Initial Planning

In this stage, all the arranging of the project is finished. Initially, I will make arrangements and guidelines to finish my activities in due time. I will set time period for each piece of the tasks. I will recognize my weakness in topic and dispense more opportunity to improve it. I will likewise investigate about the subjects required for the application advancement. I will also research about the algorithm used for analysis of symptoms.

Requirement Phase

In this stage, requirements for the software are accumulated and broke down. I will begin with essential requirements for web customer. I will begin by coding straightforward working code which can recognize diseases. With every Iteration requirement will help to grow bigger that give more highlights.

Design Phase

This stage starts with basic program plan and database structure. UI and UX design of server and customers is executed when the necessity is made. In this stage, requirements are outlined in configuration structure. This might be another program structure, or an augmentation of a prior plan. Every cycle stage will give reliable and straightforward plan.

Implementation

In this stage, execution of design and requirement are finished by coding. Most importantly, server is composed utilizing Python. Web customers are composed when the requirement and design is made. Testing is additionally performed is this stage. In this stage customers are incorporated with the server and tested. With every cycle, server and customers are made bug-less and stable.

Review Phase

A Review stage begins by assessing the server and customers written in execution stage. This phase also reviewed the current requirements that is weather the feature made in requirement phase is implemented or not. If the feature are advanced and is not possible to implement, then the requirements are changed. Nature of code is additionally assessed in this stage. Iteration makes the survey increasingly powerful and productive.

Deployment Phase

This is the final steps where the final output of a project can be find. In this phase, users can find the stable and final server and client that can be run in their devices.

Cycle Diagram:

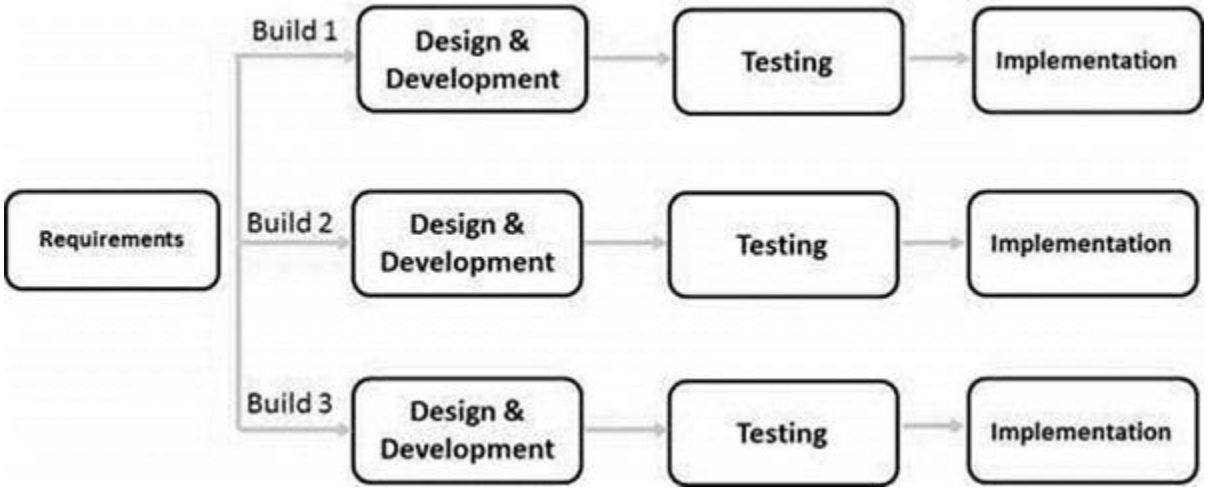


Figure 18 Cycle Diagram

6) Progress Review:

Progress Completed:

S.N	Tasks	Status
1	Research	Completed
2	Background Reading	Completed
3	Conceptual Design	Completed
4	Technical Design	Completed
5	Prototype	Completed
6	Complete Prototype	Completed
7	Interim Report	Completed
8	Analysis & Design(UI/UX)	Completed
9	Start Coding	Completed

7) Further work

S.N	Task	Status
10	Check Code	Incomplete
11	Report Writing	Incomplete
12	Final Test	Incomplete
13	Final Check (Code, Report)	Incomplete

APPENDIX I

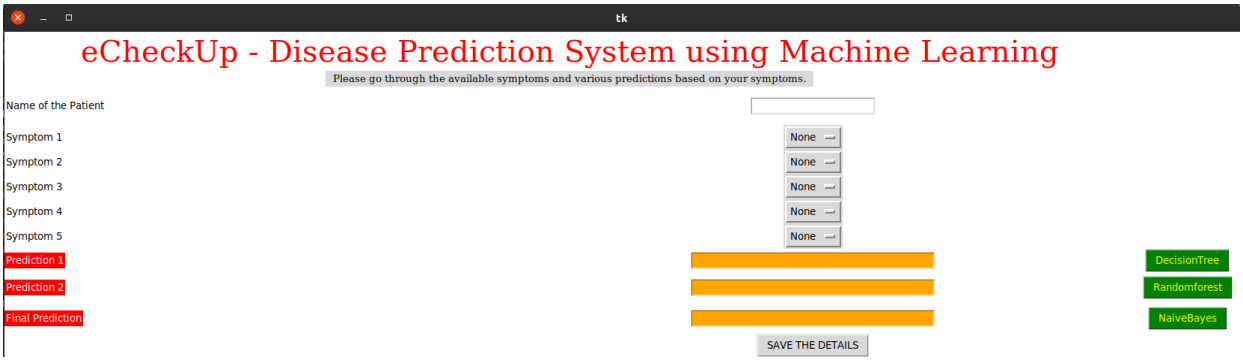


Figure 19 GUI of project

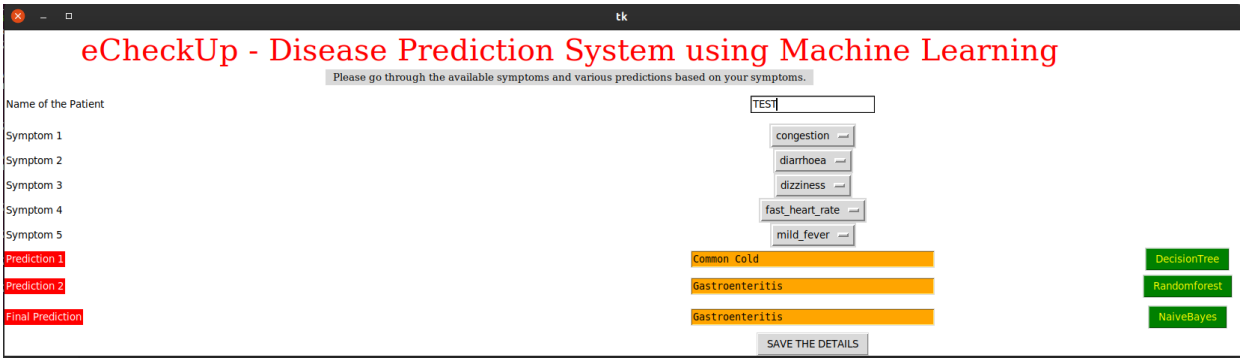


Figure 20 Inserting data

Backgroung Reading	T	19/09/2019	01/10/2019	9 days
Conceptual Design	T	02/10/2019	16/10/2019	11 days
Technical Design	T	17/10/2019	11/11/2019	18 days
Prototype	T	12/11/2019	18/12/2019	27 days
Complete Prototype	T	19/12/2019	30/12/2019	8 days
Complete Interim Report	T	31/12/2019	09/01/2020	8 days
Analysis & Design	T	10/01/2020	10/02/2020	22 days
Start Coding	T	11/02/2020	23/03/2020	30 days
Check Coding	T	23/03/2020	01/04/2020	8 days
Start Report Writing	T	01/04/2020	30/04/2020	22 days
Check, Test and Finalize	T	01/05/2020	12/05/2020	8 days
Final Check	T	13/05/2020	26/05/2020	10 days

Figure 21 Tasks

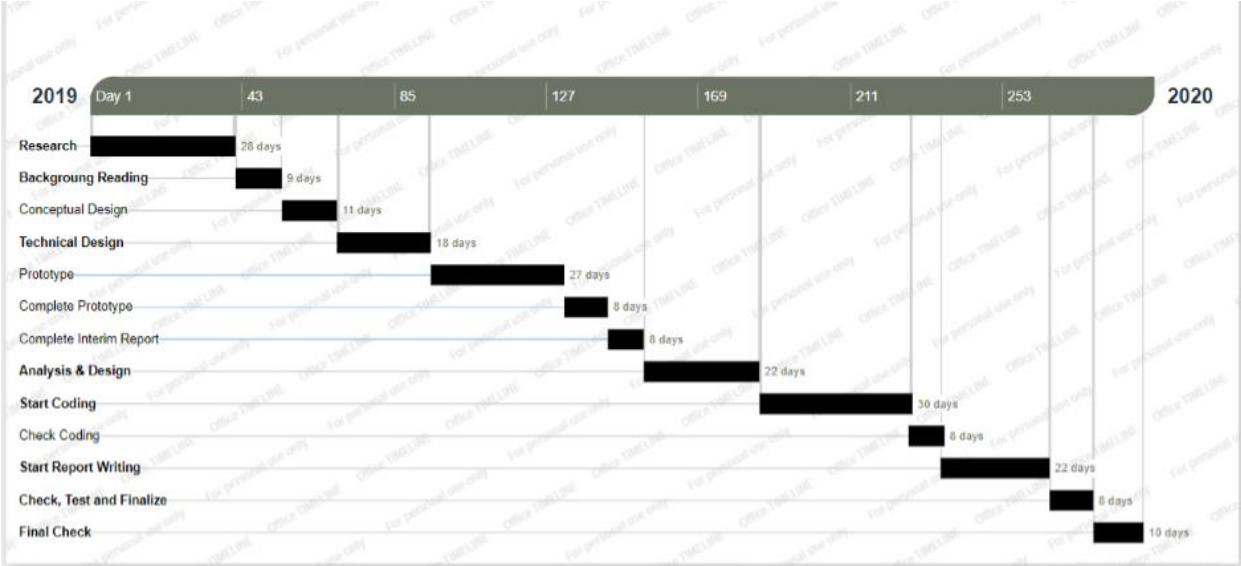


Figure 22 Gantt Chart

Bibliography

Brownlee, J. (2019) *machinelearningmastery* [Online]. Available from: https://machinelearningmastery.com/naive-bayes-classifier-scratch-python/ [Accessed 2020].

Klinger, C. (n.d.) *Iterative Software Development Approact* [Online]. Available from: https://wiki.nci.nih.gov/display/CommonProjects/Iterative+Software+Development+Approach [Accessed 05 April 2017].

lucidchart. (2019) *lucidchart* [Online]. Available from: https://www.lucidchart.com/pages/decision-tree [Accessed 2020].

projectmanager. (2020) *projectmanager* [Online]. Available from: https://www.projectmanager.com/blog/scrum-methodology [Accessed 2020].

Solutions, B. (n.d.) *What is a software development process* [Online]. Available from: http://www.selectbs.com/analysis-and-design/what-is-a-software-development-process [Accessed 05 April 2017].

Synced. (2019) *medium* [Online]. Available from: https://medium.com/@Synced/how-random-forest-algorithm-works-in-machine-learning-3c0fe15b6674 [Accessed 2020].

toolsqa. (2018) *toolsqa* [Online]. Available from: https://www.toolsqa.com/software-testing/waterfall-model/ [Accessed 2020].