ABSTRACT

Machine Learning is used across many ranges around the world. The healthcare industry is no exception. Machine Learning can play an essential role in predicting presence/absence of locomotors disorders, heart diseases and more. Such information, if predicted well in advance, can provide important intuitions to doctors who can then adapt their diagnosis and dealing per patient basis. We carried out different research on heart disease from a data analytics point of view. Prediction of heart disease is a very recent field as the data is becoming available. We use heart disease dataset available in the UCI machine learning repository to predict the heart disease by training the model with these datasets. Starting with a pre-processing phase, where we selected the most relevant features by the correlation matrix, then we applied three data analytics techniques Random Forest, SVM and Logistic Regression on the same dataset, in order to study the ac-curacy and stability of each of them. Then we apply these three algorithms to predict heart disease. The accuracy of KNN, SVM and Logistic Regression was 89%, 87% and 86% respectively. The accuracy can be increased by adding more training rows of dataset.

Keywords: diagnosis, Correlation, Random Forest, Logistic Regression, pre-processing, Repository, Training.

1.INTRODUCTION

Heart disease prediction System predicts whether patients have heart disease by giving some features of users. This is important to medical fields. If such a prediction is accurate enough, we can not only avoid wrong diagnosis but also save human resources. When a patient without a heart disease is diagnosed with heart disease, he will fall into unnecessary panic and when a patient with heart disease is not diagnosed with heart disease, he will miss the best chance to cure his disease. Such wrong diagnosis is painful to both patients and hospitals. With accurate predictions, we can solve the unnecessary trouble. Besides, if we can apply our machine learning tool into medical prediction, we will save human resources because we do not need complicated diagnosis processes in hospitals (though it is a very long way to go).

According to the World Health Organization, every year 12 million deaths occur worldwide due to Heart Disease. Heart disease is one of the biggest causes of morbidity and mortality among the population of the world. Prediction of cardiovascular disease is regarded as one of the most important subjects in the section of data analysis. The load of cardiovascular disease is rapidly increasing all over the world from the past few years. Many researches have been conducted in an attempt to pinpoint the most influential factors of heart disease as well as accurately predict the overall risk. Heart Disease is even highlighted as a silent killer which leads to the death of the person without obvious symptoms. The early diagnosis of heart disease plays a vital role in making decisions on lifestyle changes in high-risk patients and in turn reduces the complications.

The input to our algorithm is 13 features with number values . We use several algorithms such as Logistic Regression, SVM, KNN, Random Forest to output a binary number 1 or 0. 1 indicates the patient has heart disease and vice versa.

PROBLEM STATEMENT

Heart Disease is not that easy for doctors to detect, even doctors having good experience might not be able to detect. There is still a lack of access with almost two-thirds of the world's population lacking access hospital facility. The major challenge in heart disease is its detection. There are instruments available which can predict heart disease but either they are expensive or are not efficient to calculate the chance of heart disease in humans. Early detection of cardiac diseases can decrease the mortality rate and overall complications. However, it is not possible to monitor patients every day in all cases accurately and consultation of a patient for 24 hours by a doctor is not available since it requires more patience, time and expertise. Since we have a good amount of

data in today's world, we can use various machine learning algorithms to analyze the data

for hidden patterns. The hidden patterns can be used for health diagnosis in medicinal data.

In the context of Nepal, many of the rural government health posts lack basic equipment, and some have not been staffed for years. Rural areas of Nepal have one doctor for every 150,000 people so it is difficult for patients to be diagnosed properly. To eradicate such defects in heart disease detection, better technologies can be developed that can improvise detection and testing. Automating this detection task would greatly improve the efficiency of detection.

PROJECT OBJECTIVES

GENERAL OBJECTIVES

The main aim of this project is to determine whether a patient should be diagnosed with heart disease or not. The secondary aim is to develop a web application that allows users to predict heart disease utilizing the prediction engine.

SPECIFIC OBJECTIVES

- To develop machine learning model to predict future possibility of heart disease by implementing Logistic Regression, SVM and Random Forest.
- To determine significant risk factors based on medical dataset which may lead to heart disease.
- To analyze feature selection methods and understand their working principle.
- To develop user-friendly Web -application for heart disease prediction.
- To provide new approach to concealed patterns in the data.
- To reduce the cost of medical tests.
- To test the performance of built models and guarantee the acceptable.

SIGNIFICANCE OF THE STUDY

The ability of ML models to bring out the meaning from data and to prediction is used for early prediction of diseases. ML approach for disease detection is very important in the context of growing technology. With rapid growth of development, people are busy on their own work and do not have enough time for health checkup. Excessive work and busy schedule might lead people to heart disease problem. Therefore, disease prediction system made with machine learning algorithms help people to know about their heart condition any time on the basis of their symptoms. Even the people with heart disease can know about their heart condition. Sometimes the situation can occur when you need the doctor's help immediately, but they are not available due to some reason. At the time individual can enter the symptoms, then the system will process those symptoms for heart disease possibility.

The major significance of such system may include:

- Lower the cost of visiting doctor.
- Available for anyone and anywhere with internet access.
- Faster and reliable disease detection

1.4 SCOPE AND LIMITATIONS

The scope of the system is to provide a better heart disease prediction system in web-app with consideration of cross-platform operability. Our system provides following features:

Allows users to know about their heart issues through an intelligent health care system online

System uses intelligent supervised learning to guess the most accurate output

Can be easily implement for hospital to overcome patient load.

Provide service for with no necessary equipment.

Although our project tries to solve many problems in existing system and, it has certain limitations that must be considered. Some of the limitations of the project are:

- It consumes more time for processing the activities.
- Unavailability of medical resources.
- Patients must visit hospital for diagnosis.
- Must be access to internet to take service.
-

2. LITERATURE STUDY/REVIEW

This section consists of the literature study on the Heart Disease Prediction System. Our project is looking forward to define all the possible services so that there is an intelligent system to detect the heart disease.

2.1 REVIEW

With the ever-increasing in rapid development and busy life, it is also obvious that there will be impact on health of people. People with busy life will also have unhealthy food and irregular time to eat food. Therefore, there will be always chance of having health issues related to heart. So, Our system predict the people health issues. Even system provides service for the people on their heart t disease prediction with better accuracy but some services might seem overwhelming, and some services might seem insufficient, there is always some areas for improvement. Hence our project is looking forward to define all the possible features so that there is an intelligent prediction system.

With recent advances in machine learning paradigms, we have noticed a great interest in interpretation of machine learning models but before heading toward new generation of monitoring approach we must appreciate tasks such as by (Polaraju, Durga Prasad, & Tech Scholar, 2017) [1] made Prediction of Heart Disease using Multiple Regression Model and it proves that Multiple Linear Regression is appropriate for predicting heart disease chance. The work is performed using training data set consists of 3000 instances with 13 different attributes which has mentioned earlier. The data set is divided into two parts that is 70% of the data are used for training and 30% used for testing.

While another work form Purushottam, et, al [2] proposed a paper “Efficient Heart Disease Prediction System” using hill climbing and decision tree algorithms. They used Cleveland dataset and preprocessing of data is performed before using classification algorithms. The Knowledge Extraction is done based on Evolutionary Learning (KEEL), an opensource data mining tool that fills the missing values in the data set. A decision tree follows top-down order. For each actual node selected by hill-climbing algorithm a node is selected by a test at each level. The parameters and their values used are confidence. Its minimum confidence value is 0.25. The accuracy of the system is about 86.7%.

(Beyene & Kamat, 2018) [3] recommended different algorithms like Naive Bayes, Classification Tree, KNN, Logistic Regression, SVM and ANN. The Logistic Regression gives better accuracy compared to other algorithms. (Beyene & Kamat, 2018) suggested Heart Disease Prediction System using Data Mining Techniques. WEKA software used for automatic diagnosis of disease and to give qualities of services in healthcare centers. The paper used various algorithms like SVM, Naïve Bayes, Association rule, KNN, ANN, and Decision Tree. The paper recommended SVM is effective and provides more accuracy as compared with other data mining algorithm

(Soni, Ansari, & Sharma, 2011) [4] proposed to use non- linear classification algorithm for heart disease prediction. It is proposed to use bigdata tools such as Hadoop Distributed File System (HDFS), Map reduce along with SVM for prediction of heart disease with optimized attribute

set. This work made an investigation on the use of different data mining techniques for predicting heart diseases. It suggests to use HDFS for storing large data in different nodes and executing the prediction algorithm using SVM in more than one node simultaneously using SVM. SVM is used in parallel fashion which yielded better computation time than sequential SVM.

(Sai & Reddy, 2017) [5] proposed heart disease prediction using ANN algorithm in data mining. Due to increasing expenses of heart disease diagnosis disease, there was a need to develop new system which can predict heart disease. Prediction model is used to predict the condition of the patient after evaluation on the basis of various parameters like heart beat rate, blood pressure, cholesterol etc. The accuracy of the system is proved in java.

(A & Naik, 2016) [6] recommended to develop the prediction system which will diagnosis the heart disease from patient's medical data set. 13 risk factors of input attributes have considered to build the system. After analysis of the data from the dataset, data cleaning and data integration was performed. He used k-means and naïve Bayes to predict heart disease. This paper is to build the system using historical heart database that gives diagnosis. 13 attributes have considered for building the system. To extract knowledge from database, data mining techniques such as clustering, classification methods can be used. 13 attributes with total of 300 records were used from the Cleveland Heart Database. This model is to predict whether the patient have heart disease or not based on the values of 13 attributes.

2.2 DATASET

The major problem with using AI for the diagnosis of disease is the lack of data for training predictive models. Though there is vast amount of data including mammograms, genetic tests, and medical records, they are not open to the people who can make use of them for research. Some initiatives like “100,000 Genomes Project” in the UK, the U.S. Department of Veteran Affairs’ “Million Veteran Program”, and the NIH’s “The Cancer Genome Atlas” [7] will hopefully provide data to researchers and data scientists. In many countries’ health records are being digitized. The adoption of EMR is also increasing. According to a data brief by The Office of National Coordinator for Health Information Technology (ONC) [8], 3 out of 4 private or not-for-profit hospitals adopted at least a Basic EHR system in the US. In many other countries, different EHR systems exist.

2.3 EXISTING SYSTEM

In 2016, Dey et al. [10] used Principal Component Analysis (PCA) for selection of features and analyzed the performance of heart disease prediction using Naïve Bayes, Decision Trees and Support Vector Machines. After reducing the correlated variables to linearly uncorrelated variables using PCA, SVM outperformed Naïve Bayes and Decision Tree.

Methaila et. al [11] predicted heart disease using data mining Techniques. The main Methodology used for prediction is KNN Algorithms, Decision Trees like CART, C4.5, CHAID, J48, ID3 Algorithms, and Naive Bayes Techniques. This system uses 13 medical attributes as input and with that input, Data sets it to process the data mining techniques and shows the most accurate one.

Rairikar et. Al [12]. used three main data mining techniques in their work: namely Decision Tree, Neural Networks and Naïve Bayes Classifier are used. The main task of data Prediction is done using these three techniques

4. METHODOLOGY

4.1. SOFTWARE DEVELOPMENT LIFE CYCLE

The software model we used in this system was waterfall model. 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.

4.1.1 PRODUCT DEVELOPMENT LIFE CYCLE

4.1.2 HEART DISEASE DATASET

The dataset contains the records of people who do not have heart disease and the records of those who have heart disease or not. The dataset of people with or without heart disease with different attributes obtained from UCI repository. Attributes of heart disease is given as:

4.2.3 DATA PREPROCESSING

Data preprocessing in ML refers to the technique of preparing (cleaning and organizing) the raw data to make it suitable for building and training ML models. In simple words, data preprocessing in ML is a data mining technique that transforms raw data into an understandable, readable and required format. Data preprocessing is a crucial step to enhance the quality of data to promote the extraction of meaningful insights from the data. Data pre-processing includes:

IMPORTING ALL CRUCIAL LIBRARIES

In order to perform data preprocessing using Python, we imported some predefined Python libraries to perform some specific jobs. There are three specific libraries that we used for data preprocessing, which are:

NUMPY: NumPy Python library is the fundamental package for scientific calculation in Python. It also supports adding large, multidimensional arrays and matrices. It is used for including mathematical operation in the code.

MATPLOTLB: The second library is Matplotlib, which is a Python 2D plotting library, and with this library, we also imported a sub-library pyplot. This library is used to plot charts in Python for the code.

PANDAS: The last library is the Pandas library, which is one of the most famous Python libraries and used for importing and managing datasets. It is an open-source data manipulation and analysis library. The Sklearn Library is mainly used for modeling data and it provides efficient tools that are easy to use for any kind of predictive data analysis.

SKLEARN: Sklearn (scikit-learn) is a Python library that provides a wide range of unsupervised and supervised machine learning algorithms. It is also one of the most used machine learning libraries and is built on top of SciPy.

IMPORTING THE DATA

The dataset was imported which was collected for our machine learning project. Before importing a dataset, we set the current directory as a working directory.

HANDLING MISSING DATA

If dataset contains some missing data, then it may create a huge problem for machine learning model. It is necessary to handle missing values present in the dataset. The next Thus, next step of data preprocessing is to handle missing data in the datasets.

Regression analysis is used to systematically eliminate data: Regression is useful for handling missing data because it can be used to predict the null value using other information from the dataset. Regression methods can be successful in finding the missing data, but this largely depends on how well connected the remaining data is.

FEATURE SCALING

Feature scaling is the final step of data preprocessing in machine learning. This technique is used to standardize the independent variables of the dataset in a specific range. In feature scaling, we put our variables in the same range and in the same scale so that no variable dominates the other variable.

a) Standardization

$$x' = \frac{x - mean(x)}{standard\ deviation(x)} \quad \text{(i)}$$

4.1.3. TRAIN TEST SPLITTING

Dataset for Machine Learning model is split into two separate sets – training set and test set. Training set is the subset of a dataset that is used for training the machine learning model. A test set is the subset of the dataset that is used for testing the machine learning model. The ML model used the test set to predict outcomes.

The dataset is split into 75:25 ratios. This means that we took 75% of the data for training the model while leaving out the rest 25% for testing the model and output.

4.1.4. MODULE IMPLEMENTATION

Once we split the data, we then use Logistic Regression, SVM and KNN to train module with train data set. This algorithm has its own performance to train the module. So, each algorithm might have different output and accuracy

4.1.5. MODEL PERFORMANCE (ACCURACY) CALCULATIONS

Once we trained the model using the training data set, we then evaluated this model to find out the performance of the model using the test data. We look forward to generate 80-90% accuracy in our model during the performance calculation. However, each algorithm has different accuracy.

.

4.1.6. BUILDING A PREDICTIVE MODEL

In this step, we built a predictive system so that by entering all the input data, our model should predict whether a person has heart disease or not. This step is important so that if the new patient or a new user input the attributes and they can find out whether they have heart disease or not. In this step, we created a tuple of new input values and our already trained module should be able to predict the output correctly.

4.1.7 OUTPUT

This is the final step in which the output of the previously created input tuple is generated in the form of ‘does not have Heat Disease or ‘have heart disease, signifying if a person does not have heart disease or has heart disease respectively.

4.2 FEATURE OF CHOOSE WATERFALL MODEL

The reason to choose water fall model are as follows:

- Water model is easy to follow.
- It can be implemented any size of project
- Every stage has to done separately at the right time so you can’t jump stages.
- Documentation is produced at every stage of a water fall model allowing people to understand what has been done.
- Testing is done at every stage.

4.3. TOOLS AND TECHNOLOGY USED

4.3.1. TOOLS

JUPYTER LAB

Jupyter Lab is the latest web-based interactive development environment for notebooks, code, and data. Its flexible interface allows users to configure and arrange workflows in data science, scientific computing, computational journalism, and machine learning. A modular design invites extensions to expand and enrich functionality. Jupyter lab is mainly used for training and testing data of model in our project.

GITHUB

GitHub is a website and cloud-based service that helps developers store and manage their code, as well as track and control changes to their code.

VSCODE

Visual Studio Code is a code editor in layman’s terms. Visual Studio Code is “a free-editor that helps the programmer write code, helps in debugging and corrects the code using the intelli-sense method.

PYTHON3

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages. Python is a must for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. Python 3.0 was released in 2008. Although this version is supposed to be backward incompatibles, later on many of its important features have been reported to be compatible with version 2.7.

DJANGO

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source. Django was designed to help developers take applications from concept to completion as quickly as possible. Django takes security seriously and helps developers avoid many common securities mistake

SQLITE

SQLite is a relational database that's compatible with SQL. Unlike other SQL-based systems such as MySQL and PostgreSQL, SQLite doesn't use a client-server architecture. SQLite stores its data in a single cross-platform file. The database becomes an integral part of the program, eliminating resource-intensive standalone processes.

5. ALGORITHM

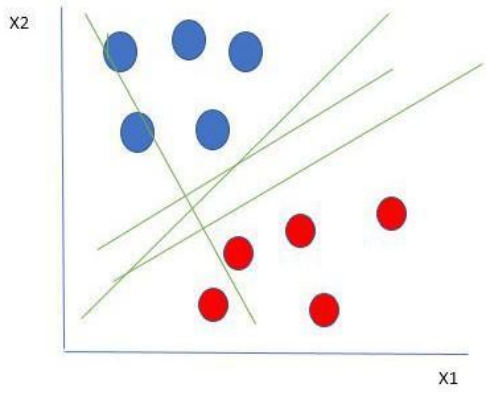
Three algorithms are implemented to train model for prediction system. Using of three algorithms help to find better accurate result.

5.1. SUPPORT VECTOR MACHINE(SVM)

Support Vector Machine (SVM) is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well its best suited for classification. The objective of SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points. The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line. If the number of input features is three, then the hyperplane becomes a 2-D plane. It becomes difficult to imagine when the number of features exceeds three.

5.1.1 WORKING PRINCIPLE OF SVM

Let's consider two independent variables x1, x2 and one dependent variable which is either a blue circle or a red circle.

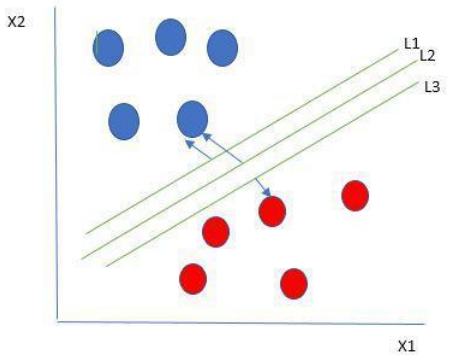


From the figure above its very clear that there are multiple lines (our hyperplane here is a line because we are considering only two input features x1, x2) that segregates our data points or

does a classification between red and blue circles. So how do we choose the best line or in general the best hyperplane that segregates our data points.

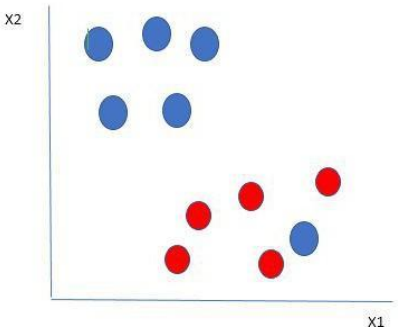
Selecting the best hyper-plane:

One reasonable choice as the best hyperplane is the one that represents the largest separation or margin between the two classes.



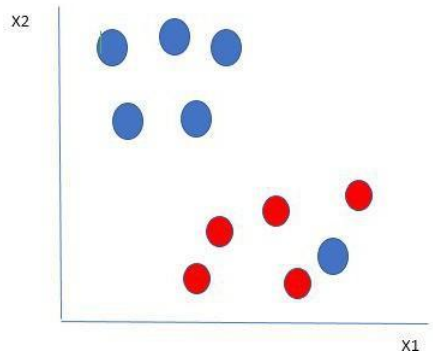
So, we choose the hyperplane whose distance from it to the nearest data point on each side is maximized. If such a hyperplane exists it is known as the maximum-margin hyperplane/hard margin. So, from the above figure, we choose L2.

Let's consider a scenario like shown below



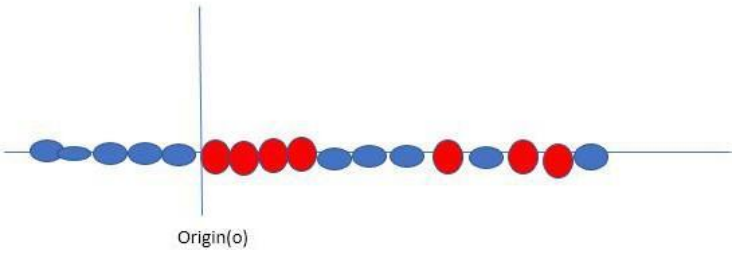
Here we have one blue ball in the boundary of the red ball. So how does SVM classify the data? It's simple! The blue ball in the boundary of red ones is an outlier of blue balls. The SVM

algorithm has the characteristics to ignore the outlier and finds the best hyperplane that maximizes the margin. SVM is robust to outliers.

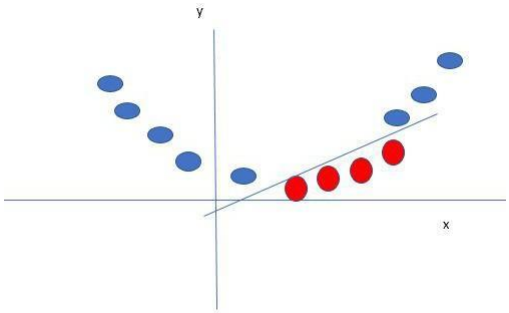


So, in this type of data points what SVM does is, it finds maximum margin as done with previous data sets along with that it adds a penalty each time a point crosses the margin. So, the margins in these types of cases are called soft margin. When there is a soft margin to the data set, the SVM tries to minimize (1/margin+∧(∑penalty)). Hinge loss is a commonly used penalty. If no violations no hinge loss. If violations hinge loss proportional to the distance of violation.

Till now, we were talking about linearly separable data (the group of blue balls and red balls are separable by a straight line/linear line). What to do if data are not linearly separable?



Say, our data is like shown in the figure above. SVM solves this by creating a new variable using a kernel. We call a point xi on the line and we create a new variable yi as a function of distance from origin o.so if we plot this, we get something like as shown below

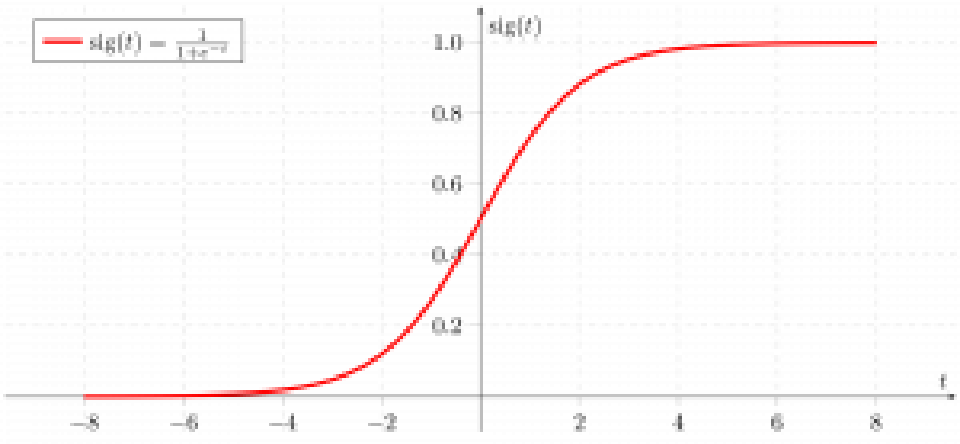


In this case, the new variable y is created as a function of distance from the origin. A non-linear function that creates a new variable is referred to as kernel.

5.2. LOGISTIC REGRESSION

Logistic regression is basically a supervised classification algorithm. In a classification problem, the target variable (or output), y, can take only discrete values for a given set of features (or inputs), X. Contrary to popular belief, logistic regression IS a regression model. Logistic regression models are a relationship between predictor variables and a categorical response variable. For example, we could use logistic regression to model the relationship between various measurements of a manufactured specimen (such as dimensions and chemical composition) to predict if a crack greater than 10 mils will occur (a binary variable: either yes or no). Logistic regression helps us estimate a probability of falling into a certain level of the categorical response given a set of predictors. We can choose from three types of logistic regression, depending on the nature of the categorical response variable: The model builds a regression model to predict the probability that a given data entry belongs to the category numbered as “1”. Just like Linear regression assumes that the data follows a linear function, Logistic regression models the data using the sigmoid function, which is given by

$$g(z) = \frac{1}{1+e^{-z}}$$



Logistic regression becomes a classification technique only when a decision threshold is brought into the picture above.

We can choose from three types of logistic regression, depending on the nature of the categorical response variable:

(i) Binary Logistic Regression:

Used when the response is binary (i.e., it has two possible outcomes). The cracking example given above would utilize binary logistic regression. Other examples of binary responses could include passing or failing a test, responding yes or no on a survey, and having high or low blood pressure.

(ii) Nominal Logistic Regression

Used when there are three or more categories with no natural ordering to the levels. Examples of nominal responses could include departments at a business (e.g., marketing, sales, HR), type of search engine used (e.g., Google, Yahoo!, MSN), and color (black, red, blue, orange).

(iii) Ordinal Logistic Regression

Used when there are three or more categories with a natural ordering to the levels, but the ranking of the levels does not necessarily mean the intervals between them are equal. Examples of ordinal responses could be how students rate the effectiveness of a college course (e.g., good, medium, poor), levels of flavors for hot wings, and medical condition (e.g., good, stable, serious, critical).

5.3. K-Nearest Neighbor (KNN)

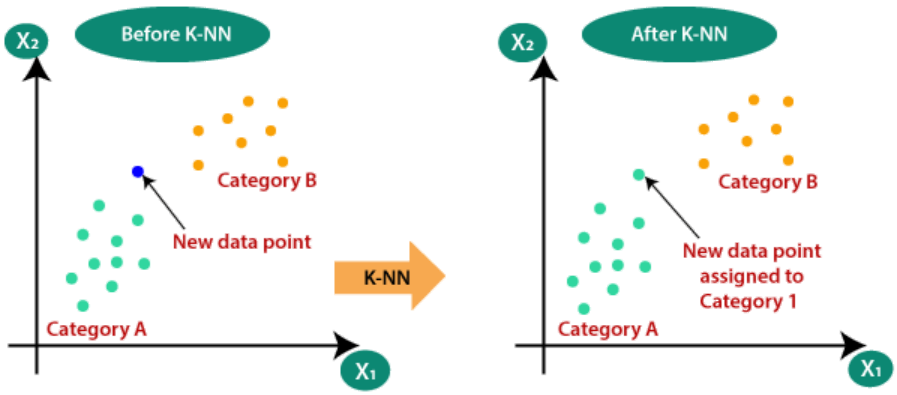
K-nearest neighbors (KNN) algorithm is a type of supervised ML algorithm which can be used for both classification as well as regression predictive problems. However, it is mainly used for classification predictive problems in industry. The following two properties would define KNN well –

(i) Lazy learning algorithm − KNN is a lazy learning algorithm because it does not have a specialized training phase and uses all the data for training while classification

(ii) Non-parametric learning algorithm − KNN is also a non-parametric learning algorithm because it doesn't assume anything about the underlying data.

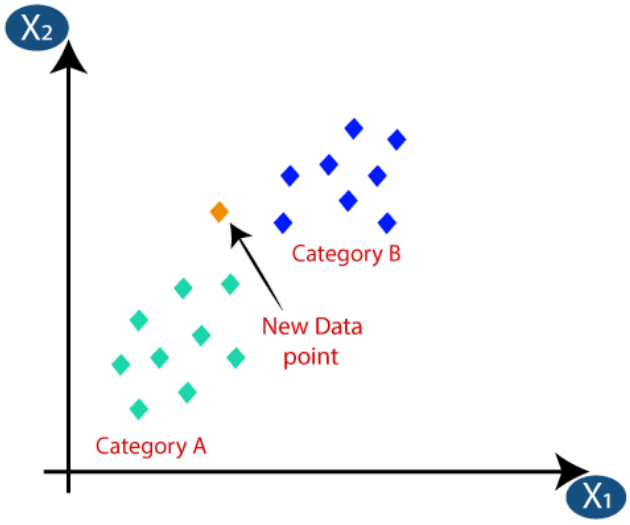
5.3.1. Working Principle of KNN

Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:



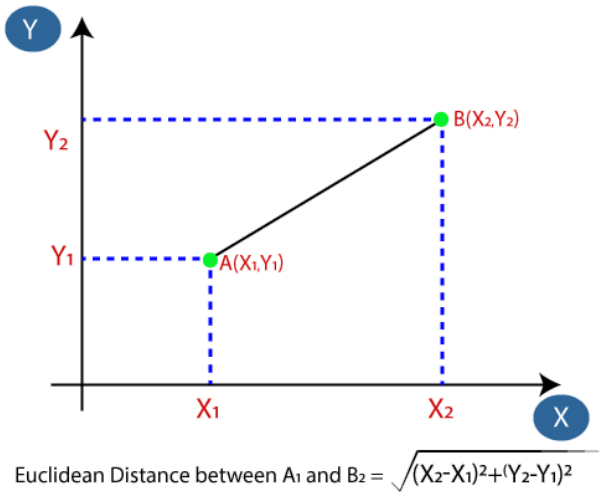
The K-NN working can be explained on the basis of the below algorithm:

Suppose we have a new data point and we need to put it in the required category. Consider the below image:

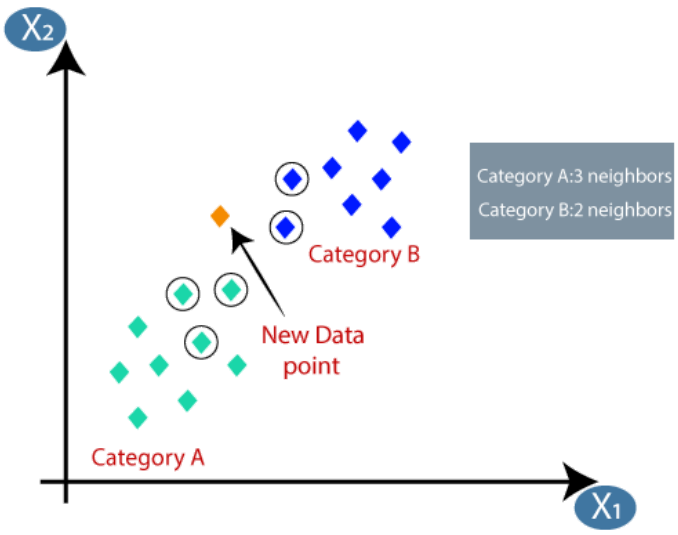


Step-1: Select the number K of the neighbors. Let's, say k=5. There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model. Large values for K are good, but it may find some difficulties.

Step-2: Take the K nearest neighbors as per the calculated Euclidean distance. The Euclidean Distance can be calculated as:



Step-3: By calculating the Euclidean distance we got the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B. Consider the below image:



Step-4: As we can see the 3 nearest neighbors are from category A, hence this new data point must belong to category A.

6.SOFTWARE REQUIREMENTS SPECIFICATIONS

A software requirement specification is a description of a software system to be developed. It lays out functional and non-functional requirements. It describes what the software product is expected to do and what not to do. It enlists enough and necessary requirements that are required for project development. It mainly aids to describe the scope of the work and provides software designers with a form of reference.

6.1. FUNCTIONAL REQUIREMENTS

Functional requirements describe the functionality that the system must perform. These capture the intended behavior of the system. This software system requires the following functionalities:

User

login Sign

up

Use dataset and build Classification model

Input parameters

Calculation using build Classification model

Predicted output

Logout

6.2. NON-FUNCTIONAL REQUIREMENTS

Performance Requirements

The performance of this software should be with as much less latency as possible. The system uses a lots of system resources, thus, is naturally slow than other systems. The software should perform in reasonable time, i.e., 2-5 seconds except while loading.

User Friendly

The UI ought to be as user friendly as possible with beautifully displayed GUI violations. Each action should be seamless to the user and navigating between program components should be as easy and intuitive to use as possible. Keyboard controls and mouse controls both are essential and makes it useful for users with different input preferences.

Extensibility

The system should be extensible. New functions should be able to be easily added without affecting the functions of existing functions. The system should be easily expandable without breaking.

Integrability

The system needs to be in a situation of accepting and integrating of new components written elsewhere. The process for integration of components should be as short as possible and as less tedious as well.

Maintainability

Along with extensibility and Integrability, the existing components should be maintainable. If any problem occurs in the existing modules, then it should be easy to fix and as less hassle should be involved as possible. There should exist at least cohesiveness and each module should be maintainable independently.

6.3. Feasibility Assessment

A feasibility assessment is done to analyze the viability of an idea. In the case of software development, it assesses the practicality of the project or system. The result from the feasibility assessment determines whether the project should go ahead, be redesigned, or be dropped. There are five areas of feasibility - Technical, Economic, Legal, Operational, and Scheduling

6.3.1. Operational Feasibility

The operational feasibility analysis describes how the system operates and what resources the system requires for performing its designated task. It is also a measure of how well a system solves the problems and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. The system runs in a website accessing data from the database server. The system is designed to be operated in the browser environment, and hence eliminates the difficulty of installation in every computer and the other devices from which the user is trying to use the system. The project is developed as a website allowing easy access to multiple users. The system is carefully designed to make it possible to be operable in most of the environment, hence the project can be considered operationally feasible.

6.3.2. Technical Feasibility

Technical Feasibility Assessment examines whether the proposed system can be designed to solve the desired problems and requirements using available technologies in the given problem domain. The system is said to be feasible technically if it can be deployable, operable, and manageable under the current technological context of the global market. Various factors are associated with the assessment of technical feasibility such as the right selection of technology

type, use of standard technology, and familiarity of project team members with the technology used. Since the development of single-page web applications is feasible and the dataset for training and building our ML model is also available, the project can be considered technically feasible.

6.3.3. Economic Feasibility

Economic Feasibility checks whether the cost required for complete system development is feasible using the available resources in hand. It should be noted that the cost of resources and overall cost of deployment of the system should be kept minimum while operational and maintenance costs for the system should be within the capacity of the organization. Since the system is hosted on servers that are easily available on the market, with the costs and performance being optimum, the system can be considered economically feasible for development.

6.3.4. Legal Feasibility

Legal Feasibility assessment checks the system for any conflicts with a legal requirement, regulations that are to be followed as per the standard maintained by the governing body. As such, the system that is being developed must comply with all the legal boundaries such as copyright violation, authorize the use of licenses, and others. This prevents any future conflicts for the system and also provides a legal basis for the system in the future if any other party tries to use part of our full system without necessary permits and documents. Since the data obtained from the user is being consented to and does not violate any other obligation of law and privacy, the project can be considered legally feasible.

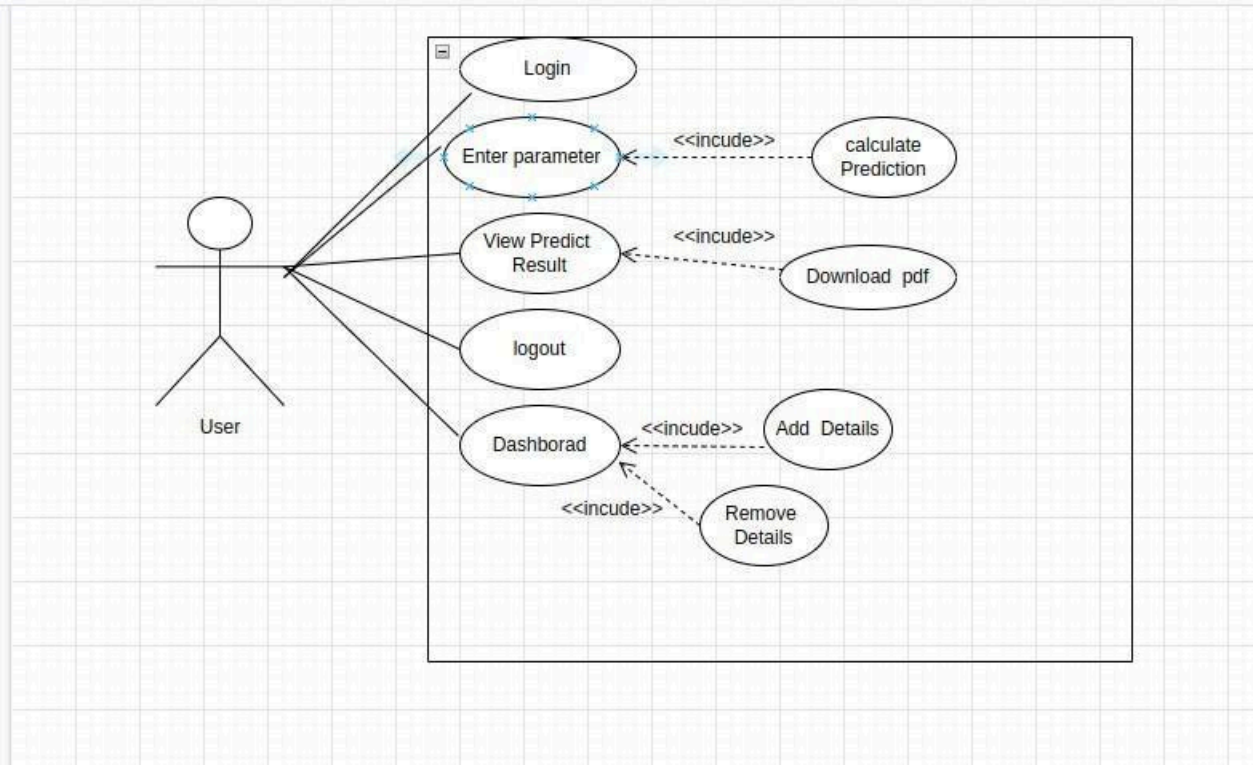
6.3.5. Scheduling Feasibility

Any project is considered to fail if it is not completed on time. Scheduling Feasibility estimates the time required for the system to fully develop and whether that time is feasible or not according to the current trend in the market. If the project takes a long time to complete, it may be outdated or some others may launch a similar system before our system is complete. So, it is required to fix the deadline for any project and the system should be out and operative before the specified deadline. As the scheduling of the project is consistent with the available time of the project, the project can be considered scheduling feasible.

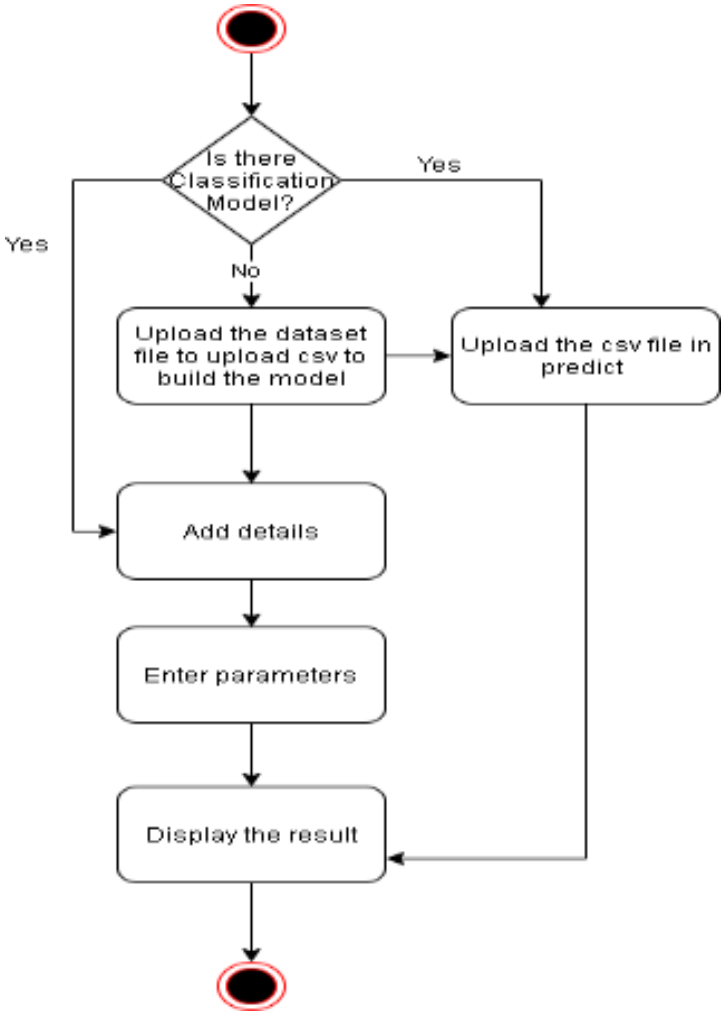
7. SYSTEM DESIGN AND UML MODELS

Designing according to the functional and non-functional requirements, we have tried to make sure that the ML pipeline confirms the user requirements of the system.

7.1. USE CASE DIAGRAM



7.2Activity Diagram



The figure shows the activity of the main steps involved in the system. If user is already logged in, should have to follow following activity in the system

Use an already generated model or upload dataset file for estimation.

Go to the estimation page and enter parameters.

Click estimate to predict parameter.

7.3. Sequence Diagram

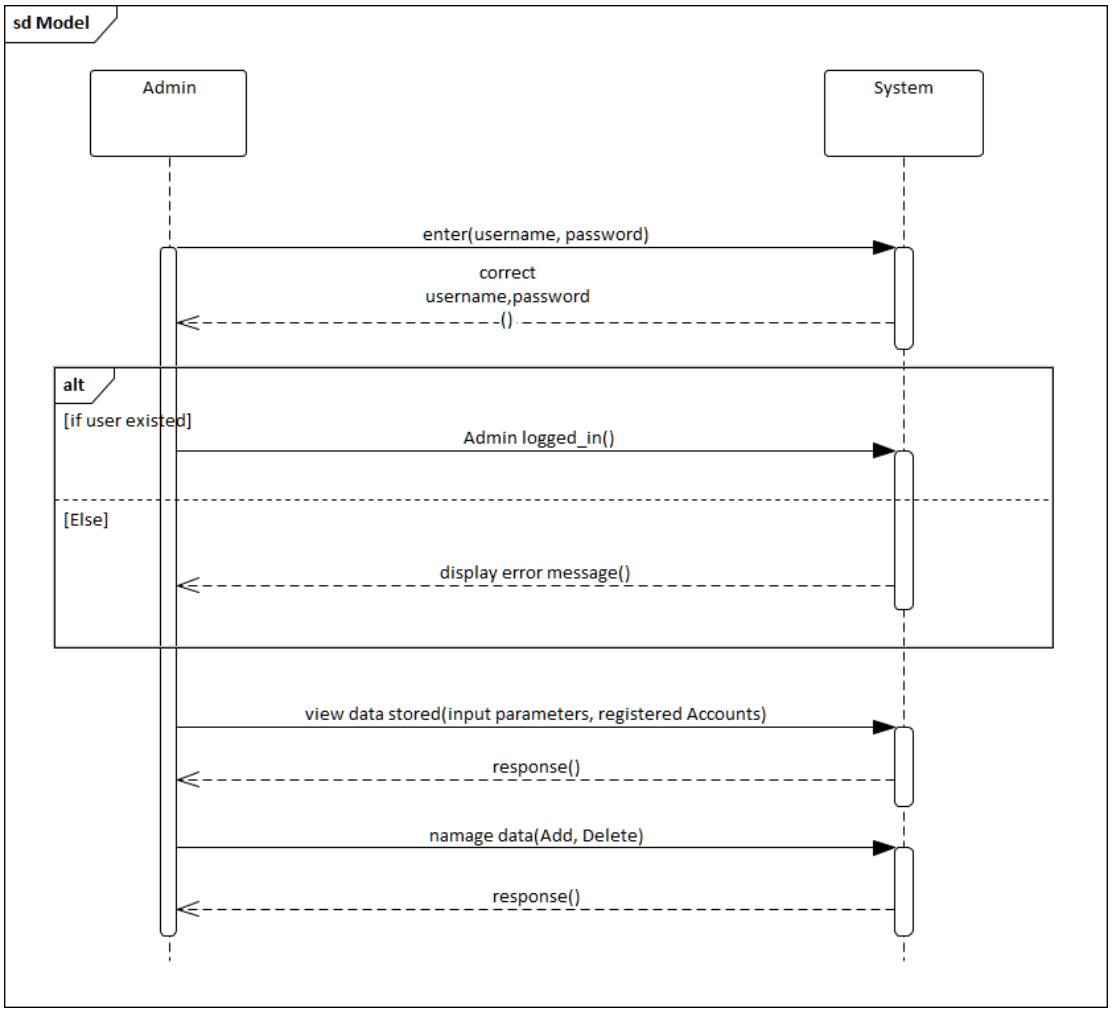


Figure shows the sequential activity for admin and system interaction.

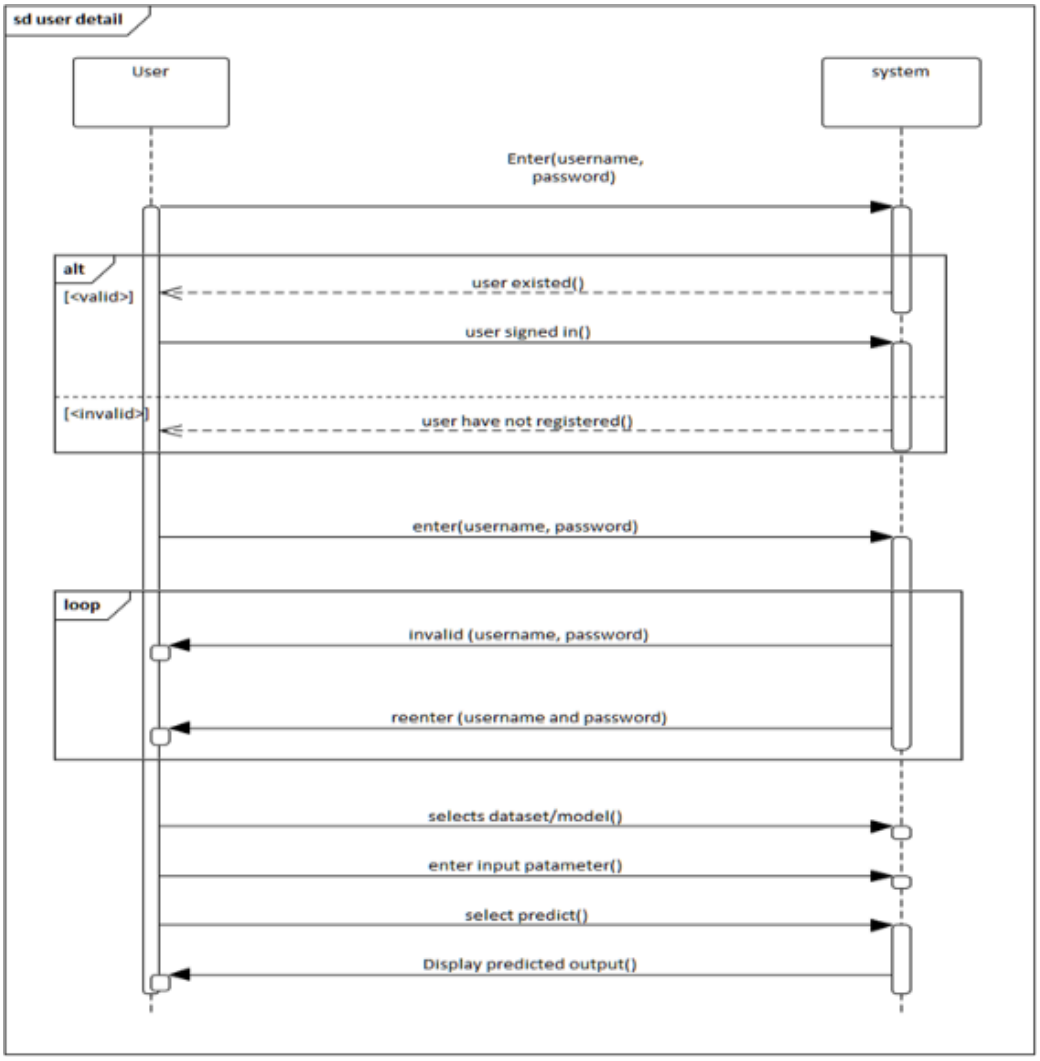


Figure shows the sequential activity for user and system interaction.

User first enters name and password then system validates it. If user existed, system allows to login else displays error message.

User can sign up in the system to register.

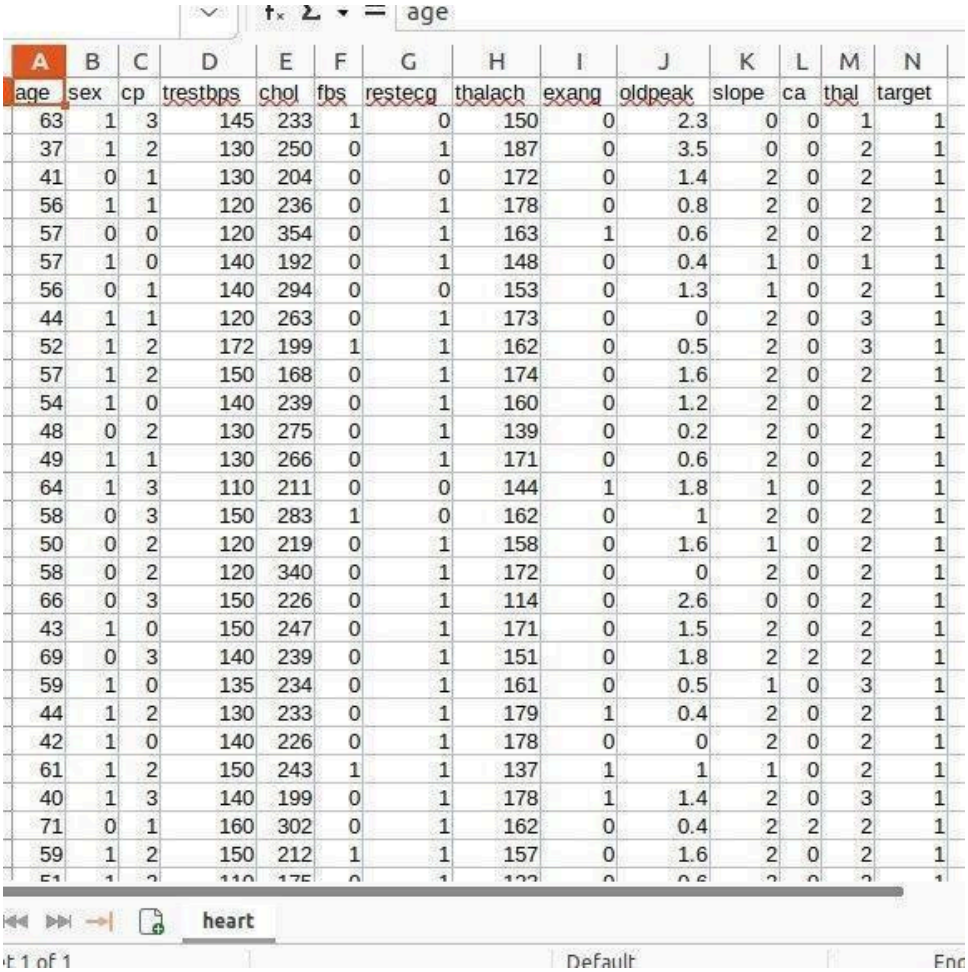
If user is already logged in, user can use the system to predict data.

User can enter parameter for input and predict data according to it.

6. ML PIPELINE

6.1 DATA GATHERING

Artificial intelligence requires data to learn and make decision making process. Data collected from different sources are in raw form which are transformed into different format to train the model.



age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
57	1	0	140	192	0	1	148	0	0.4	1	0	1	1
56	0	1	140	294	0	0	153	0	1.3	1	0	2	1
44	1	1	120	263	0	1	173	0	0	2	0	3	1
52	1	2	172	199	1	1	162	0	0.5	2	0	3	1
57	1	2	150	168	0	1	174	0	1.6	2	0	2	1
54	1	0	140	239	0	1	160	0	1.2	2	0	2	1
48	0	2	130	275	0	1	139	0	0.2	2	0	2	1
49	1	1	130	266	0	1	171	0	0.6	2	0	2	1
64	1	3	110	211	0	0	144	1	1.8	1	0	2	1
58	0	3	150	283	1	0	162	0	1	2	0	2	1
50	0	2	120	219	0	1	158	0	1.6	1	0	2	1
58	0	2	120	340	0	1	172	0	0	2	0	2	1
66	0	3	150	226	0	1	114	0	2.6	0	0	2	1
43	1	0	150	247	0	1	171	0	1.5	2	0	2	1
69	0	3	140	239	0	1	151	0	1.8	2	2	2	1
59	1	0	135	234	0	1	161	0	0.5	1	0	3	1
44	1	2	130	233	0	1	179	1	0.4	2	0	2	1
42	1	0	140	226	0	1	178	0	0	2	0	2	1
61	1	2	150	243	1	1	137	1	1	1	0	2	1
40	1	3	140	199	0	1	178	1	1.4	2	0	3	1
71	0	1	160	302	0	1	162	0	0.4	2	2	2	1
59	1	2	150	212	1	1	157	0	1.6	2	0	2	1

heart

6.2 DATA PREPROCESSING

Data preprocessing is technique that involves transferring raw data into an understandable format. Real-world data is usually incomplete, inconsistent, and lacks certain behaviors or trends, most likely to contain many inaccuracies. The process of getting usable data for a Machine Learning algorithm follows steps such as Feature Extraction and Scaling, Feature Selection,

Dimensionality reduction, and sampling. The product of Data Pre-processing is the final dataset used for training the model and testing purposes.

```
[8]: #Correlation
     correlation = df.corr()
     correlation
```

:[8]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	targe
age	1.000000	-0.098447	-0.068653	0.279351	0.213678	0.121308	-0.116211	-0.398522	0.096801	0.210013	-0.168814	0.276326	0.068001	-0.22543
sex	-0.098447	1.000000	-0.049353	-0.056769	-0.197912	0.045032	-0.058196	-0.044020	0.141664	0.096093	-0.030711	0.118261	0.210041	-0.28093
cp	-0.068653	-0.049353	1.000000	0.047608	-0.076904	0.094444	0.044421	0.295762	-0.394280	-0.149230	0.119717	-0.181053	-0.161736	0.43379
trestbps	0.279351	-0.056769	0.047608	1.000000	0.123174	0.177531	-0.114103	-0.046698	0.067616	0.193216	-0.121475	0.101389	0.062210	-0.14493
chol	0.213678	-0.197912	-0.076904	0.123174	1.000000	0.013294	-0.151040	-0.009940	0.067023	0.053952	-0.004038	0.070511	0.098803	-0.08523
fbs	0.121308	0.045032	0.094444	0.177531	0.013294	1.000000	-0.084189	-0.008567	0.025665	0.005747	-0.059894	0.137979	-0.032019	-0.02804
restecg	-0.116211	-0.058196	0.044421	-0.114103	-0.151040	-0.084189	1.000000	0.044123	-0.070733	-0.058770	0.093045	-0.072042	-0.011981	0.13723
thalach	-0.398522	-0.044020	0.295762	-0.046698	-0.009940	-0.008567	0.044123	1.000000	-0.378812	-0.344187	0.386784	-0.213177	-0.096439	0.42174
exang	0.096801	0.141664	-0.394280	0.067616	0.067023	0.025665	-0.070733	-0.378812	1.000000	0.288223	-0.257748	0.115739	0.206754	-0.43675
oldpeak	0.210013	0.096093	-0.149230	0.193216	0.053952	0.005747	-0.058770	-0.344187	0.288223	1.000000	-0.577537	0.222682	0.210244	-0.43069
slope	-0.168814	-0.030711	0.119717	-0.121475	-0.004038	-0.059894	0.093045	0.386784	-0.257748	-0.577537	1.000000	-0.080155	-0.104764	0.34587
ca	0.276326	0.118261	-0.181053	0.101389	0.070511	0.137979	-0.072042	-0.213177	0.115739	0.222682	-0.080155	1.000000	0.151832	-0.39172
thal	0.068001	0.210041	-0.161736	0.062210	0.098803	-0.032019	-0.011981	-0.096439	0.206754	0.210244	-0.104764	0.151832	1.000000	-0.34402
target	-0.225439	-0.280937	0.433798	-0.144931	-0.085239	-0.028046	0.137230	0.421741	-0.436757	-0.430696	0.345877	-0.391724	-0.344029	1.00000

6.3 FEATURE EXTRACTION AND ENGINEERING

The preprocessed data contain different features which are not useful for training the model. From the available labelled data unwanted features are removed and important features are extracted. Unwanted features in the labelled data reduce accuracy of model. So, it must be removed before applying for training model.

6.4. MODEL TRAINING

Feature engineering leaves data in format which then can be fed into algorithms to get the output. For better accuracy and options, multiple algorithms have been used. Logistic Regression, SVM and KNN algorithms has been used.

6.5. MODEL TESTING

After fitting the models, they were then subjected to the testing dataset and the supervised models then gave predictions. The accuracy is then calculated and made ready for deployment.

6.5.1 CONFUSION MATRIX

Once the model evaluation is complete, the pipeline selects the best model and deploys it. The pipeline can deploy multiple machine learning models to ensure a smooth transition between old and new models; the pipeline services continue to work on new prediction requests while deploying a new model.

K-nearest-neighbor classifier

```
# creating Knn Model
Knn_model= KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2)
Knn_model.fit(x_train_scaler, y_train)
y_pred_knn= Knn_model.predict(x_test_scaler)
Knn_model.score(x_test_scaler,y_test)
```

```
0.9016393442622951
```

```
print('Classification Report\n', classification_report(y_test, y_pred_knn))
print('Accuracy: {}%\n'.format(round((accuracy_score(y_test, y_pred_knn)*100),2)))
```

```
Classification Report
              precision    recall  f1-score   support

           0       0.90      0.90      0.90        29
           1       0.91      0.91      0.91        32

    accuracy                           0.90        61
   macro avg       0.90      0.90      0.90        61
weighted avg       0.90      0.90      0.90        61

Accuracy: 90.16%
```

```
cm = confusion_matrix(y_test, y_pred_knn)
cm
```

```
array([[26,  3],
       [ 3, 29]])
```

Fig: KNN confusion matrix

From above matrix, we conclude that there is only 3 False negative and 3 False Positive Value.

Support Vector Classifier

```
SVC_model= SVC()
SVC_model.fit(x_train_scaler, y_train)
y_pred_SVC= SVC_model.predict(x_test_scaler)
SVC_model.score(x_test_scaler,y_test)
```

```
0.8688524590163934
```

```
print('Classification Report\n', classification_report(y_test, y_pred_SVC))
print('Accuracy: {}%\n'.format(round((accuracy_score(y_test, y_pred_SVC)*100),2)))
```

```
Classification Report
              precision    recall  f1-score   support

           0       0.89      0.83      0.86        29
           1       0.85      0.91      0.88        32

    accuracy                           0.87        61
   macro avg       0.87      0.87      0.87        61
weighted avg       0.87      0.87      0.87        61

Accuracy: 86.89%
```

```
cm = confusion_matrix(y_test, y_pred_SVC)
cm
```

```
array([[24,  5],
       [ 3, 29]])
```

Fig: SVM confusion matrix

From above confusion matrix of SVM of we conclude that algorithm gave 3 False Negative and 5 False Positive Values.

Logistic Regression Model

```
# creating Logistic Regression Model
LR_model= LogisticRegression()
LR_model.fit(x_train_scaler, y_train)
y_pred_LR= LR_model.predict(x_test_scaler)
LR_model.score(x_test_scaler,y_test)
```

```
0.8852459016393442
```

```
print('Classification Report\n', classification_report(y_test, y_pred_LR))
print('Accuracy: {}%\n'.format(round((accuracy_score(y_test, y_pred_LR)*100),2)))
```

```
Classification Report
              precision    recall  f1-score   support

           0       0.89      0.86      0.88        29
           1       0.88      0.91      0.89        32

    accuracy                           0.89        61
   macro avg       0.89      0.88      0.88        61
weighted avg       0.89      0.89      0.89        61

Accuracy: 88.52%
```

```
# confusion matrix
cm = confusion_matrix(y_test, y_pred_LR)
cm
```

```
array([[25,  4],
       [ 3, 29]])
```

Fig: Logistic Regression

From Logistic Regression model confusion matrix, we conclude that model gave 3 False Negatives and 4 False Positive value.

7. WORK BREAKDOWN

8.GANT CHART

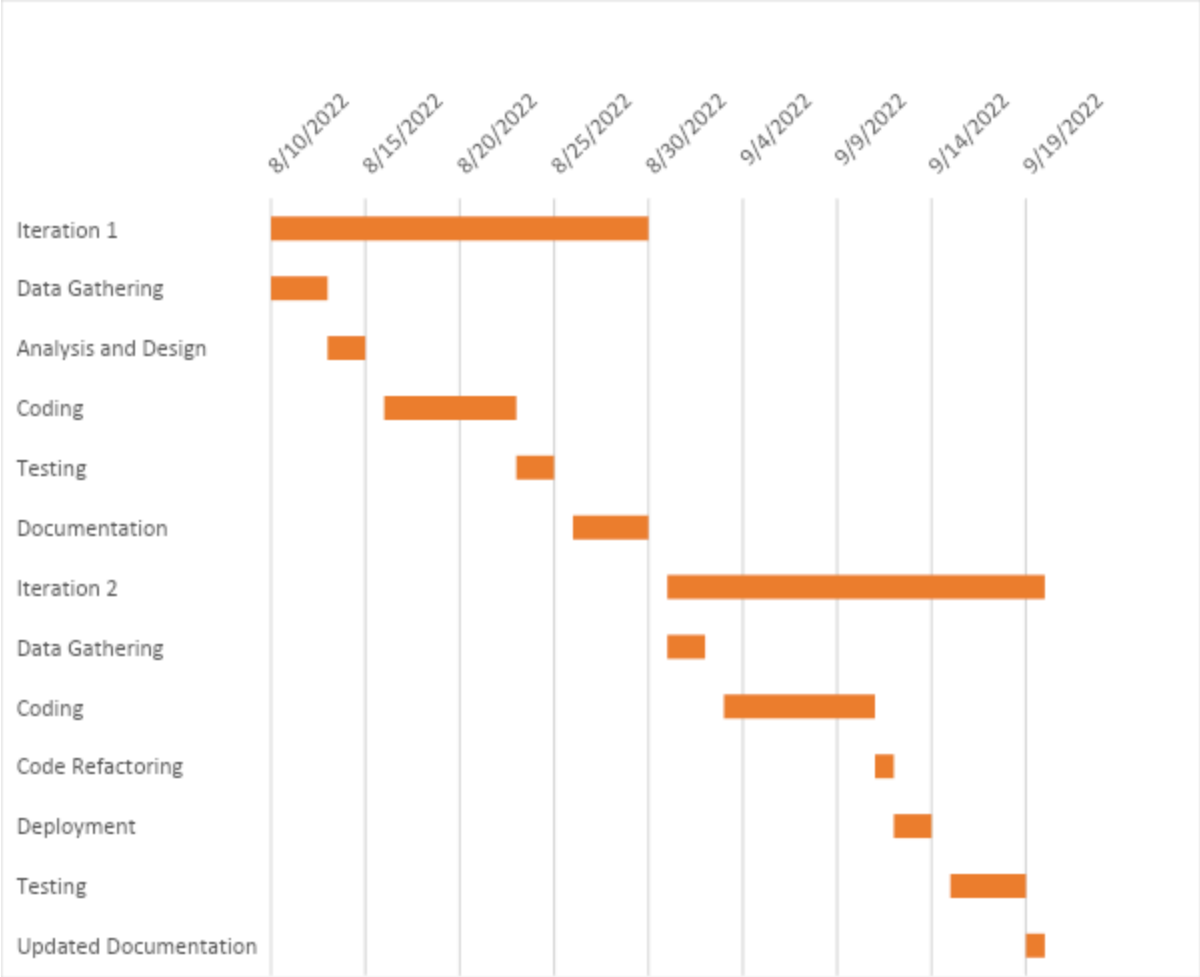


Fig:Gant Chart

9.CONCLUSION

We developed a Prediction Engine which enables the user to check whether he/she has heart disease or not. The user interacts with the Prediction Engine by filling a form which holds the parameter set provided as an input to the trained models. The Prediction engine provides an optimal performance compared to other state of art approaches. The Prediction Engine makes use of three algorithms to predict the presence of a disease namely: Support Vector Machine (SVM), K-Nearest Neighbours (KNN) and Random Forrest.The reason to choose these three algorithms are:

They are effective, if the training data is large.

A single dataset can be provided as an input to all these 3 algorithms with minimal or no modification.

A common scalar can be used to normalize the input provided to these 3 algorithms.

10. FUTURE EXTENSIONS

The project can be further improved by introducing the big datasets. We had to work with a very limited dataset available to us and hence the accuracy of the model was affected. We visited some hospitals that diagnose Parkinson's disease in Nepal, but due to the unavailability of voice recording device, we weren't able to collect the required dataset locally. As the disease is not talked about much in our country, we couldn't conduct a local survey. As our software is completely related to medical science, it is a sensitive topic. Hence, we cannot currently commercialize our product due to the accuracy not being up to the mark.

The major thing we must work on in the future is improving the accuracy of the model. So, we need to find a more official and actual dataset. The more dataset we can train, the more accurate our model will be. We can work with different hospitals in Nepal for the enhancement of our software's performance in real life scenario.

The future work that can be done to enhance project can be listed as:

Prediction of more number of diseases.

Improve accuracy by taking large number of datasets.

Develop Mobile Application for ease of use.

Provide a user account which allows the user to keep track of their medical test data and get suggestions or support to meet the right specialists or the tests to be taken.

To enhance the functionality of the prediction engine providing the details of 5 nearest hospitals or medical facilities to the user input location.

11. REFERENCES

[1] Polaraju, K., Durga Prasad, D., & Tech Scholar, M. (2017). Prediction of Heart Disease using Multiple Linear Regression Model. International Journal of Engineering Development and Research, 5(4), 2321–9939. Retrieved from www.ijedr.org

[2] Purushottam, Saxena, K., & Sharma, R. (2016). Efficient Heart Disease Prediction System. In Procedia Computer Science (Vol. 85, pp. 962–969). https://doi.org/10.1016/j.procs.2016.05.288

[3] Beyene, C., & Kamat, P. (2018). Survey on prediction and analysis the occurrence of heart disease using data mining techniques. International Journal of Pure and Applied Mathematics, 118(Special Issue 8), 165–173. Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.085041895038&partnerID=40&md5=2f0b0 c5191a82bc0c3f0daf67d73bc81

[4] Soni, J., Ansari, U., & Sharma, D. (2011). Intelligent and Effective Heart Disease Prediction System using Weighted Associative Classifiers. Heart Disease, 3(6), 2385–2392.

[5] Sai, P. P., & Reddy, C. (2017). International Journal of Computer Science and Mobile Computing HEART DISEASE PREDICTION USING ANN ALGORITHM IN DATA MINING.
International Journal of Computer Science & Mobile Computing, 6(4), 168–172. Retrieved from www.ijcsmc.com

[6] A, A. S., & Naik, C. (2016). Different Data Mining Approaches for Predicting Heart Disease, 277– 281. https://doi.org/10.15680/IJIRSET.2016.0505545

[7] Over 100,000 whole genome sequences now available for approved researchers | Genomics England". 31 July 2019.

[8] HealthIT.gov. Office of the National Coordinator for Health Information Technology. 25 April 2016.

[9] Methaila et. al, “Prediction of Heart Disease Using Data Mining Techniques”, IEEE 2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT), 2019.