A Project Report

On

Cardiovascular Disease Prediction using KNN Algorithm

Submitted in partial fulfillment of the
Requirement for the award of the degree of

Bachelor of Technology

Computer Science and Engineering

Under the supervision of

Dr. Dileep Kr. Yadav
Professor

Submitted By:

Shyam Sagar Singh Choudhary(18SCSE1010709)

Priyam Singh (18SCSE1010038)

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

We hereby certify that the work which is being presented in the project, entitled **"Cardiovascular Disease Prediction using KNN Algorithm"** in partial fulfillment of the requirements for the award of the submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of month, **JULY-2021 to DECEMBER-2021**, under the supervision of**, Dr. Dileep Kr.Yadav Professor Department of Computer Science and Engineering**, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the project has not been submitted by us for the award of any other degree of this or any other places.

Shyam sagar singh choudhary
Priyam singh

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Dr. Dileep Kr. Yadav

Professor

CERTIFICATE

The Final Project examination of **Shyam sagar singh chodhary(18SCSE1010709)** & **Priyam singh (18SCSE1010038)** has been held on December 2021 and the work is recommended for the award **of BACHELOR OF TECHNOLGY IN COMPUTER SCIENCE AND ENGINEERING.**

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date: December, 2021
Place: Greater Noida

Acknowledgment

The success and result outcome of this project “Cardiovascular Disease Prediction using KNN Algorithm” required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

I respect and thank our guide: Mr. Dileep Kr Yadav, for providing me an opportunity to do the project work under their guidance and giving us all support and guidance which made me complete the project duly. I am extremely thankful to my college Galgotias University for providing such a nice support and guidance through our teachers.

Table of Contents

List of Figures

Abstract

Healthcare is one of the field which needs to be advanced technologically as well because in covid outbreak everything which we had was less as compared to the level of outbreak. Data mining techniques have been widely used to mine knowledgeable information from medical data bases. In data mining classification is a supervised learning that can be used to design models describing important data classes, where class attribute is involved in the construction of the classifier. Nearest neighbour (KNN) is very simple, most popular, highly efficient and effective algorithm for pattern recognition. KNN is a straight forward classifier, where samples are classified based on the class of their nearest neighbour. Medical data bases are high volume in nature.

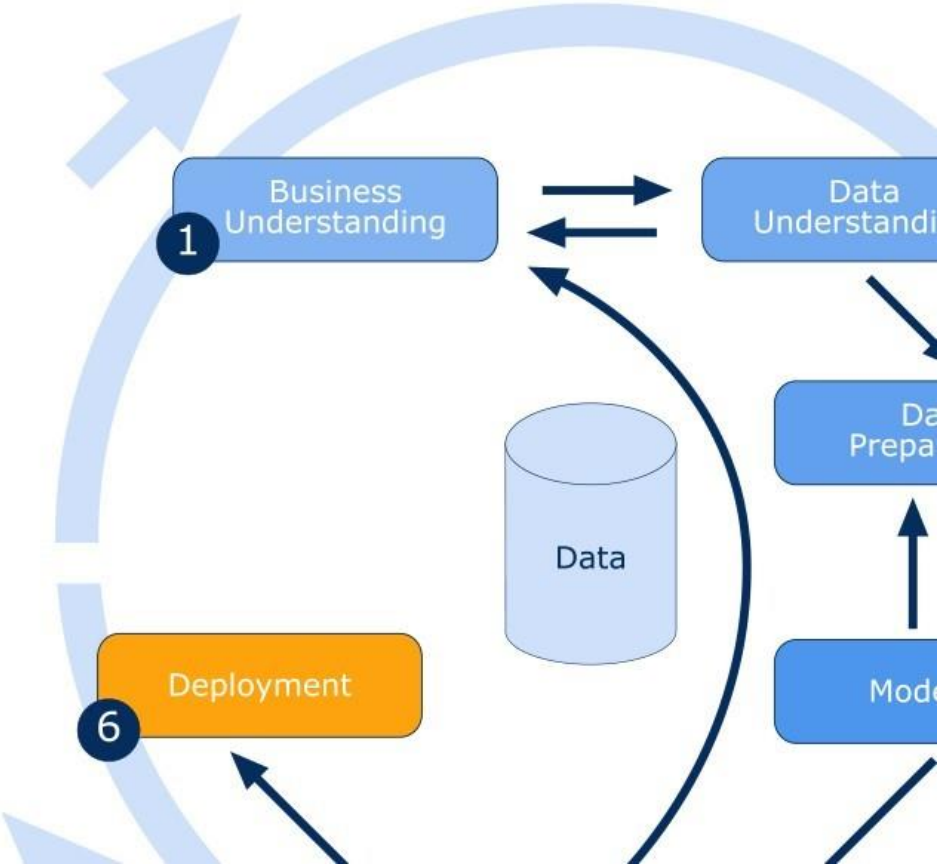
If the data set contains redundant and irrelevant attributes, classification may produce less accurate result. Heart disease is the leading cause of death in INDIA. In Andhra Pradesh heart disease was the leading cause of mortality accounting for 32%of all deaths, a rate as high as Canada (35%) and USA. Hence there is a need to define a decision support system that helps clinicians decide to take precautionary steps. In this paper we propose a new algorithm which combines KNN with genetic algorithm for effective classification. Genetic algorithms perform global search in complex large and multimodal landscapes and provide optimal solution. Experimental results shows that our algorithm enhance the accuracy in diagnosis of heart disease

The result of this introduction in Cardiovascular disease prediction improves it by making it accesible for ease prediction. Our aim in this project is to come up with a ML model for disease prediction that removes the demerits of running to the hospital for regular checkup.

Chapter 1

1.1 Introduction to Data Mining

Data mining is one of the most useful techniques that help entrepreneurs, researchers, and individuals to extract valuable information from huge sets of data. Data mining is also called Knowledge Discovery in Database (KDD).



Process of extracting information to identify patterns, trends, and useful data that would allow the business to take the data-driven decision from huge sets of data is called Data Mining.

Applications of Data Mining :-

- **Financial firms, banks, and their analysis :-** Many data mining techniques are involved in critical banking and financial data providing and keeping firms whose data is of utmost importance. One such method is distributed data mining, which is researched, modelled, crafted, and developed to help track suspicious activities or any mischievous or fraudulent transactions related to the credit card, net banking, or any other banking service.

- **Health care domain and insurance domain :-** The data mining-related applications can efficiently track and monitor a patient's health condition and help in efficient diagnosis based on the past sickness record. Similarly, the insurance industry's growth depends on the ability to convert the data into knowledge form or by providing various details about the customers, markets, and prospective competitors.

- **Application in the domain of transportation :-** The historic or batch form of data will help identify the mode of transport a particular customer generally opts for going to a particular place, say his home town, thereby providing him alluring offers and heavy discounts on new products and launched services.

- **Applications of data mining in the field of medicine:-** In the case of medical analysis, a patient's case can be analyzed by making a tab of his clinic visits and the season of his holidays. It also helps in the identification of patterns that have successful medical therapies for various kinds of illnesses. Researchers are using multi-dimensional data to reduce costs and improve the quality of services being provided today with extensive and better care.

- **Education:-** In education, the application of data mining has been prevalent, where the emerging field of educational data mining focuses mainly on the ways and methods by which the data can be extracted from age-old processes and systems of educational institutions.

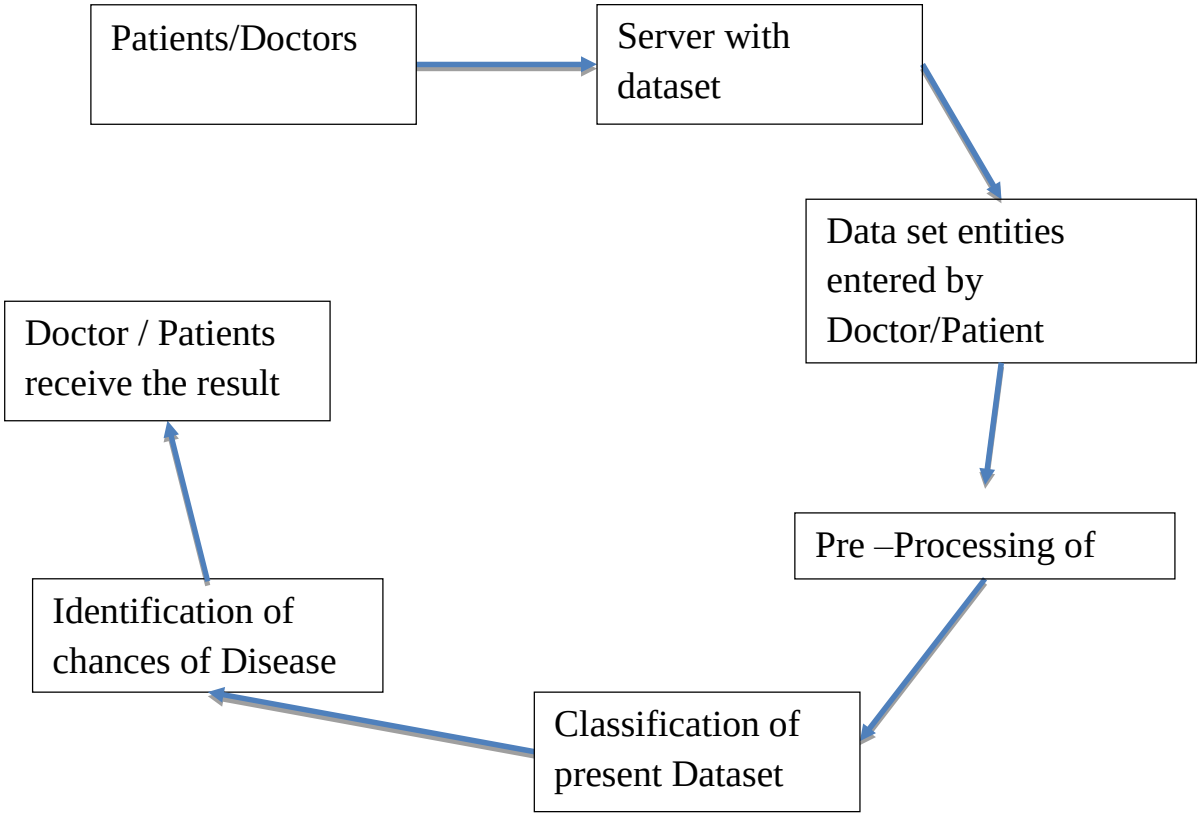
- **Manufacturing Engineering:-** The data can be assessed by ensuring that the manufacturing enterprise possesses the right set of knowledge as its asset lies in identifying the right set of product portfolios, product architecture, and the customer needs and requirements.

As the people are undergoing digital movement, the health sector has adopted new and secured technology to ease out the disease prediction processes. Until recently, people only preferred visting to hospitals,clinic or medical science books but we have started building e-prediction systems. With advancement in technology we started moving towards disease prediction via classification of data. It helps us to facilitate and process and most importantly know the disease at earliest.

Smart contract have helped enormously by:

- Reducing time taken for identification of disease
- Instant precautions for disease
- Automation of risks of disease

Most medical departments have shifted their traditional data from hard files to dataset on internet with new and improved technology. Medical Science dataset with the symptoms and risks are available on internet these days.



Data mining is a comprehensive method for a ML model, using algorithms like KNN helps us in classification of dataset with high accuracy.

KNN is one of the simplest and strong supervised learning algorithms used for classification and for regression in data mining. K- NN algorithm is based on the principle that, “the similar things or objects exist closer to each other.”

CHAPTER 2

2.1 LITERATURE SURVEY

This paper is a combination of the correlative application and detailed examination of different ML Algorithms in Python software which results in an immediate mechanism for the user to use the machine learning algorithms in Python software for estimating the cardiovascular diseases. Future enhancement comprises the work of different groups of methods to analyze these algorithms for better performance with more framework settings for these algorithms.

By examining the test results, it is concluded that J48 tree technique to be a classifier to predict heart problems because it contains more accuracy and less time to build and also observed that applying reduced error pruning to J48 results in higher performance. In summary, as identified through the literature review, we believe only a marginal success is attained in the design of a predictive model for patients having heart issues and the necessity of combinational and more complex models to increase the accuracy of analyzing the early onset of heart issues. Prediction of heart problems with distinct Decision Tree methods using classification. Heart disease is a mortal disease by its nature. This disease is a threat to life such as heart issues and may cause death. Data mining plays an important role in medical field and relevant actions are taken for prediction of disease. Many Classification Algorithms used for Disease Prediction, Decision Tree is taken because of its simplicity and accuracy.ML methods and Data Mining methods are selected for prediction of Heart Disease and diagnosing it. The disadvantage mainly depends on the applications related to classification techniques for prediction of heart disease, apart from this taking many data cleaning and pruning techniques that make a dataset suitable for mining. By analyzing the correct classification techniques will lead to the implementation of prediction systems that give more accuracy. Cardiovascular problems Prediction using ML Algorithms, predicting heart problems uses ML Algorithm provides prediction results for users. I this method Random Forest Algorithm was selected for its efficiency and accuracy and to find out prediction of heart problem percentage by knowing the correlation details between heart disease and other diseases. For the best performance and to increase accuracy new algorithms are used. Cardiovascular problems

predicting using ML Algorithms, we introduced a heart disease prediction system with different classifier techniques. The techniques are Naive Bayes and decision tree classification methods. We selected Decision Tree classifier for its performance, accuracy is more compared to others. Naive Bayes accuracy is more in some cases and small in other cases.

Several data-mining models have been embedded in the clinical environment to improve decision making and patient safety. Consequently, it is crucial to survey the principal data-mining strategies currently used in clinical decision making and to determine the disadvantages and advantages of using these strategies in data mining in clinical decision making. A literature review was conducted, which identified 21 relevant articles. The article findings showed that multiple models of data mining were used in clinical decision making. Although data mining is efficient and accurate, the models are limited with respect to disease and condition.

Enterprise data mining applications often involve complex data such as multiple large heterogeneous data sources, user preferences, and business impact. In such situations, a single method or one-step mining is often limited in discovering informative knowledge. It would also be very time and space consuming, if not impossible, to join relevant large data sources for mining patterns consisting of multiple aspects of information. It is crucial to develop effective approaches for mining patterns combining necessary information from multiple relevant business lines, catering for real business settings and decision-making actions rather than just providing a single line of patterns. The recent years have seen increasing efforts on mining more informative patterns, e.g., integrating frequent pattern mining with classifications to generate frequent pattern-based classifiers.

Rather than presenting a specific algorithm, this paper builds on our existing works and proposes combined mining as a general approach to mining for informative patterns combining components from either multiple data sets or multiple features or by multiple methods on demand. We summarize general frameworks, paradigms, and basic processes for multifeature combined mining, multisource combined mining, and multimethod combined mining. Novel types of combined patterns, such as incremental cluster patterns, can result from such frameworks, which cannot be directly produced by the existing methods. A set of real-world case studies has been conducted to test the frameworks, with some of them briefed in this paper. They identify combined patterns for informing government debt prevention and improving government service objectives, which show the flexibility and instantiation capability of combined mining in discovering informative knowledge in complex data.

Chapter 3

3.1 Functionality

Overview of Data mining advantages and their Disadvantages

Data Mining is similar to Data Science carried out by a person, in a specific Situation, on a particular data set, with an objective. This Process includes various types of services such as text mining, web mining, audio and video mining, pictorial data mining, and social media mining. It is done through Software that is simple or highly specific. By outsourcing data mining, all the work can be done faster with low operation costs. Specialized firms can also use new technologies to collect data that is impossible to locate manually. There are tonnes of information available on various platforms, but very little knowledge is accessible. The biggest challenge is to analyze the data to extract important information that can be used to solve a problem or for company development. There are many powerful instruments and techniques available to mine data and find better insight from it.

Mining:-

- The Data Mining technique enables organizations to obtain knowledge-based data.

- Data mining enables organizations to make lucrative modifications in operation and production.
- Compared with other statistical data applications, data mining is a cost efficient.
- Data Mining helps the decision-making process of an organization.

- It Facilitates the automated discovery of hidden patterns as well as the prediction of trends and behaviors.

- It can be induced in the new system as well as the existing platforms.

- It is a quick process that makes it easy for new users to analyze enormous amounts of data in a short time.

Disadvantages of Data Mining:-

- There is a probability that the organizations may sell useful data of customers to other organizations for money. As per the report, American Express has sold credit card purchases of their customers to other organizations.

- Many data mining analytics software is difficult to operate and needs advance training to work on.

- Different data mining instruments operate in distinct ways due to the different algorithms used in their design. Therefore, the selection of the right data mining tools is a very challenging task.

- The data mining techniques are not precise, so that it may lead to severe consequences in certain condition.

3.2 PROPOSED SYSTEM

The objective of the proposed system technique is to use ensemble techniques to improve the performance of predicting heart disease. Figure 1 describes the architecture of the proposed system. It is structured into six stages, including data collection, data preprocessing, feature selection, data splitting, training models, and evaluating models. The steps of the proposed approach are explained in detail as follows.

1.1. Data Collection :- The heart disease dataset [26] is utilized for training and evaluating models. It consists of 1025 records, 13 features, and one target column. The target column includes two classes: 1 indicates heart diseases, and 0 indicates nonheart disease. Table 1 describes the details of the features.

1.2. Data Preprocessing :-The features are scaled to be in the interval [0, 1]. It is worth noting that missing values are deleted from the dataset.

1.3. Feature Extraction (FE) :- The extraction of the best features is a crucial phase because irrelevant features often affect the classification efficiency of the machine learning classifier. In this phase, linear discriminant analysis (LDA) [27] and principal component analysis (PCA) [28, 29] are used to select essential features from the dataset.

1.4. Data Splitting :- In this step, the heart disease dataset is divided into a 75% training set and a 25% as the testing set. The training set is utilized for training the models, and the testing set is utilized to evaluate the models. Also, ninefold cross-validation is utilized in the training set.

1.5. Training Models :- Different types of machine learning algorithms: KNN, DT, RF, and NB are applied to classify heart disease. Also, two types of ensemble techniques: boosting and bagging are applied to classify heart disease:

- KNN is a nonparametric technique of lazy learning to enable the prediction of the new sample classification. It is utilized in several groups. It can be utilized in both the forecast problems of regression and classification. However, it is often utilized in classification when it applies to industrial problems as it fairs across all criteria examined when assessing a technique's functionality, but it is utilized mostly because of its ease of understanding and lower computation time [8–25, 27–30].

1.6. Evaluating Models. Evaluation of the proposed model is performed focusing on some criteria, namely, accuracy, recall, precision, F-score, ROC, and AUC. Accuracy is one of the most important performance metrics for classification. It is defined as the proportion between the correct classification and the total sample.

Proposed algorithm.

Our proposed algorithm Steps:

Step 1) load the data set

Step 2) Apply genetic search on the data set

Step 3) attributes are ranked based on their value

Step 4) select the subset of higher ranked attributes

Step 5) Apply (KNN+GA) on the subset of attributes that maximizes

Step 6) calculate accuracy of the classifier, which measures the ability of the

KNN Algorithm

KNN is one of the simplest and strong supervised learning algorithms used for classification and for regression in data mining.

K-Nearest algorithm is based on the principle that, “the similar things or objects exist closer to each other.”

- KNN is most commonly used to classify the data points that are separated into several classes, in order to make prediction for new sample data points.
- KNN is a non-parametric learning algorithm.
- KNN is a lazy learning algorithm.
- KNN classifies the data points based on the different kind of similarity measures (e.g. Euclidean distance etc).

In KNN algorithm ‘K’ refers to the number of neighbors to consider for classification. It should be odd value.

The value of ‘K’ in KNN algorithm must be selected carefully otherwise it may cause defects in our model.

Types of KNN Algorithm: -

Low Bias KNN algorithm
High Bias KNN algorithm

Low Bias KNN Algorithm: If the value of 'K' is small then it causes Low Bias.

High Bias KNN Algorithm: High variance. For example, over fitting of model.

In case if 'K' is very large in KNN algorithm:-

In the same way if 'K' is very large then it leads to High Bias, Low variance i.e. under fitting of model. There are many researches done on selection of right value of K, however in most of the cases taking 'K' = {square-root of (total number of data 'n')} gives pretty good result. If the value 'K' comes to be odd then it's all right else we make it odd either by adding or subtracting 1 from it.

K-NN works nicely with a small number of input variables (p), but when the number of inputs becomes very large, then there are more chances of error in prediction.

It is based on the simple concept of mathematics to measure the distance between two data points in graph. We can use different distance measuring techniques for K-NN and some of the distance measuring techniques are mentioned below: -

- Euclidean Distance
- Minkowski Distance
- Manhattan Distance

Euclidean Distance method is widely used as compared to the Minkowski Distance and Manhattan Distance.In our project we have used Euclidean Distance to calculate the distance between two points.

$$d = \sqrt{[(x2 - x1)2 + (y2 - y1)2]}$$

Working of the Project

Requirements

Common Requirements:

1. UNIX/WINDOWS Shell
2. GIT
3. Anaconda Prompt
4. Jupyter Notebook

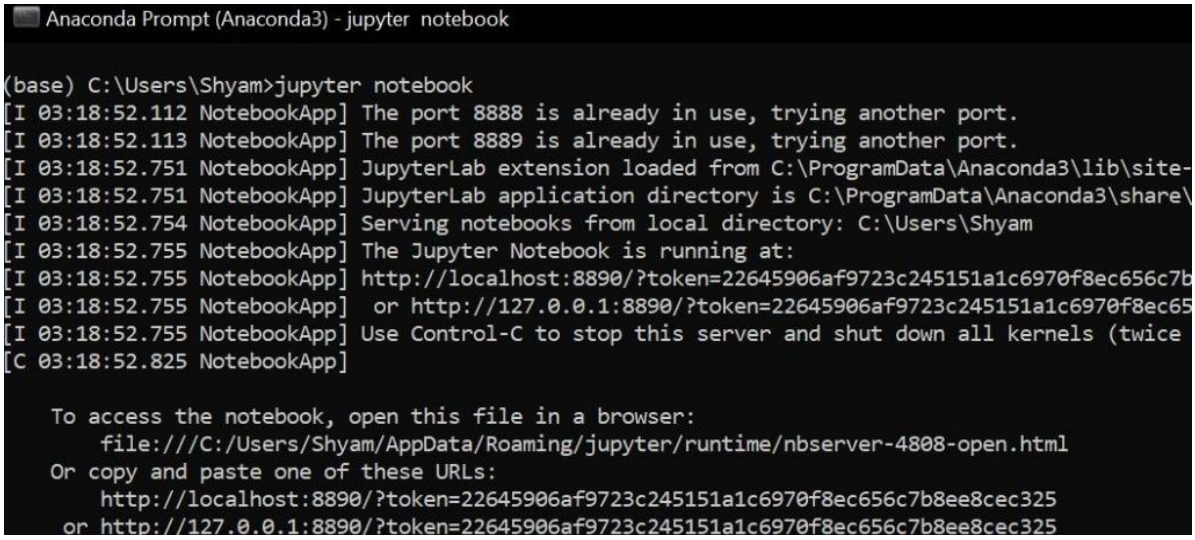
Backend Requirements:(for dataset)

1. MS- Excel
2. SQL
3. Google Cloud (In case of deployment)

Frontend Requirements:

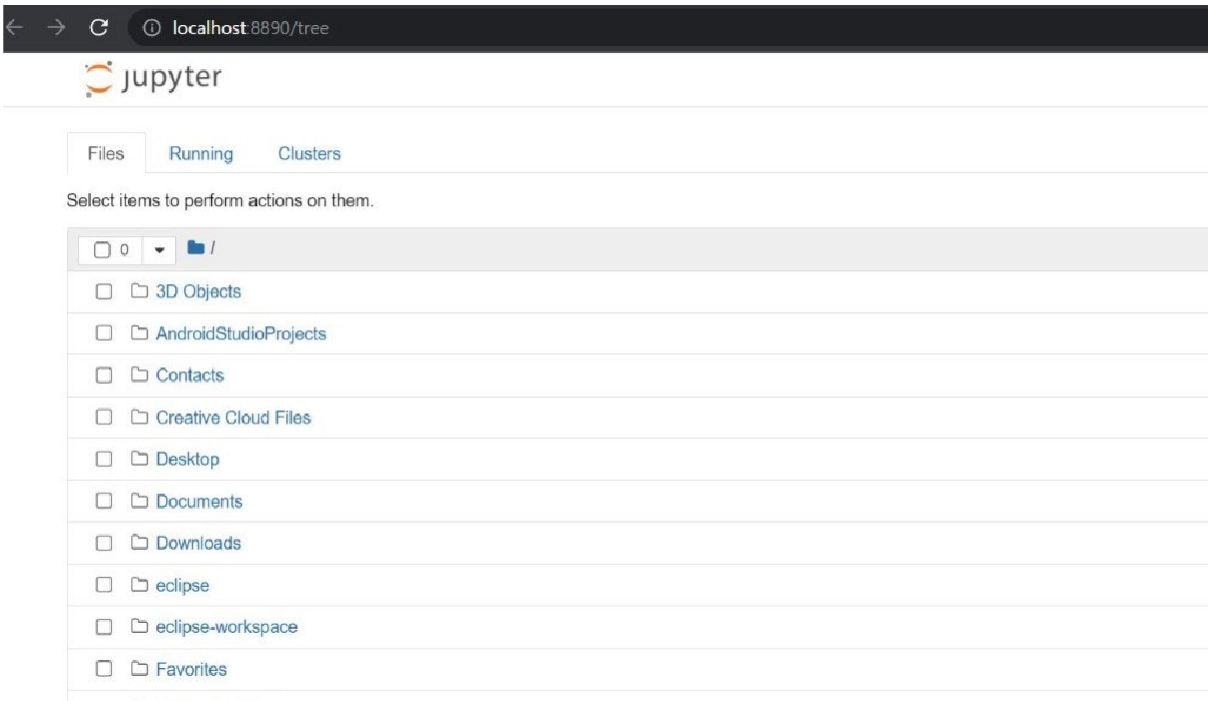
1. Python
2. Mockitt

Running the jupyter notebook using Anaconda



(jupyter notebook command in Anaconda)

After giving the jupyter notebook cmd , Anaconda redirects us to a locally hosted environment in our web browser.



(jupyter notebook window in our Web browser tab)

Dataset

```
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       303 non-null    int64
 1   sex       303 non-null    int64
 2   cp        303 non-null    int64
 3   trestbps  303 non-null    int64
 4   chol      303 non-null    int64
 5   fbs       303 non-null    int64
 6   restecg   303 non-null    int64
 7   thalach   303 non-null    int64
 8   exang     303 non-null    int64
 9   oldpeak   303 non-null    float64
 10  slope     303 non-null    int64
 11  ca        303 non-null    int64
 12  thal      303 non-null    int64
 13  target    303 non-null    int64
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
```

To implement the network based multiple-patient monitoring and alert mechanism, we use the following technologies and methodologies which will provide an active and user- friendly environment for the working of the system. Each technology we used are discussed in detail below :-

1) **Age:** The person's age in years

2) **Sex:** The person's sex(1= male, 0=female)

3) **Cp:** Chest pain type
   Value 0:asymptomatic
   -Value 1:atypical angina
   -Value 2:non aginal pain
   -Value 3:typical angina

4) **Trestbps:** The person's resting blood pressure(mm Hg on admission to the hospital)

5) **Chol:** The person's cholesterol measurement in mg/dl

6) **Fbs:** The person's fasting blood sugar(>120 mg/dl,1=true;0=false)

7) **Restecg:** resting electrocardiographic results
   -Value 0 :showing definite left ventricular hypertrophy
   -Value 1:normal
   -Value 2: having ST-T wave abnormality
8) **Thalach:** The person's amximun heart rate achieved

9) **Exang:** Excersise unduced agina (1=yes; 0=no)

10) **Oldpeak:** ST depression induced by exercise relative to rest ('ST' relates to positions on the ECG plot. See more here)

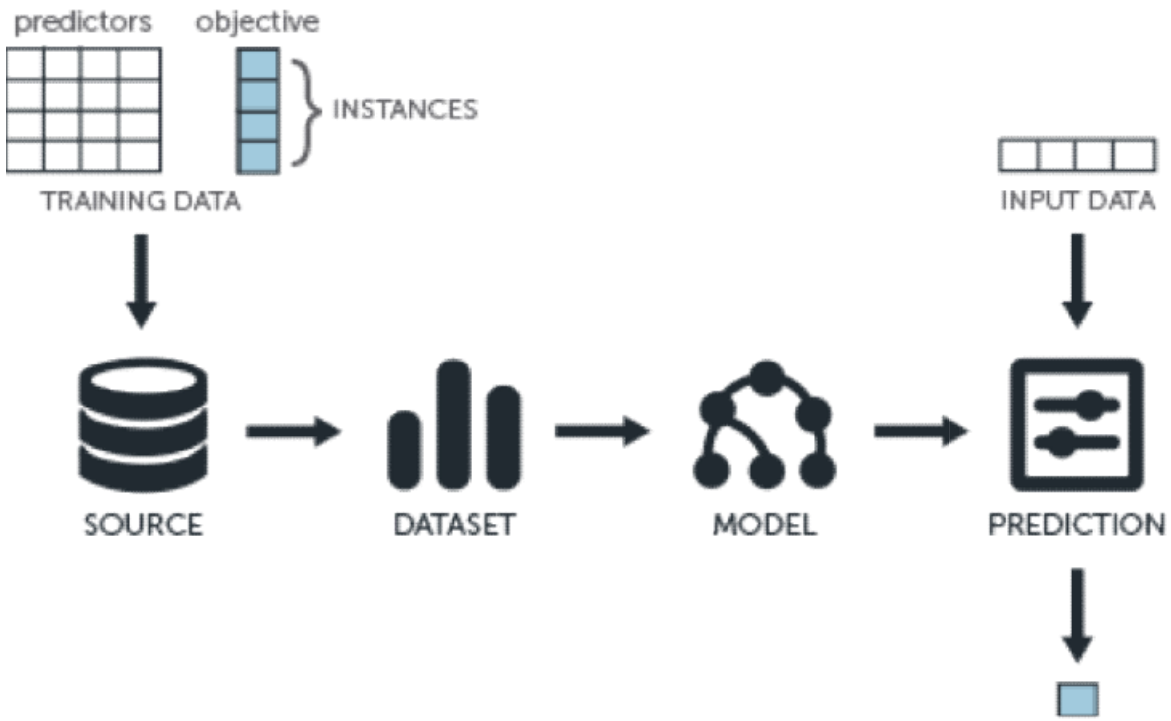
11) **Slope:** The slope of the peak exercise ST segment - 0:downsloping;1:flat;2:upsloping

12) **Ca:**The no of major vessels(0-3)

13) **Thal:** A blood disorder called thalassemia Value 0:NULL(dropped from the dataset previously)
   Value 1:fixed defect(no blood flow in some part of the heart)
   Value 2:normal blood flow
   Value 3:reversible defect(a blood flow is observed but it is normal)

14) **Target:**Heat Disease(1=no; 0=yes)

Proposed Model Block Diagram :-



Implementation of K-Nearest Neighbour on Heart disease dataset:-

1) Importing all Libraries :-

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
```

We can see here we have imported KNeighorsClassifier for our classification task. We import this from sklearn library. Sklearn has almost all the machine learning classifiers defined and we can call them and use them for our problem.

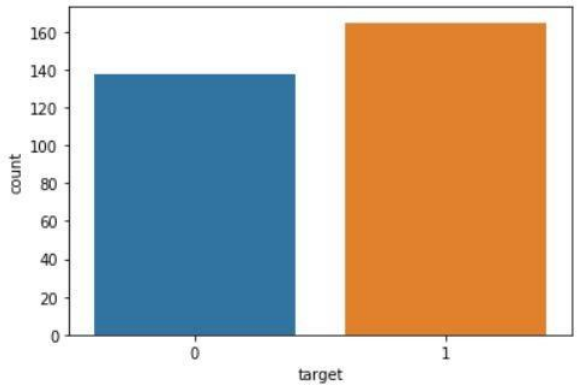
2) Read the heart disease dataset :-

```
df = pd.read_csv('heart.csv')
df.head()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

As we can see target tells us if the person is suffering from heart disease or not.

```
sns.countplot(df['target'])
```



We will proceed with this as there isn't much unbalance in target data.

3) Performing KNN by splitting to train and test set :-

```
x= df.iloc[:,0:13].values
y= df['target'].values
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test= train_test_split(x, y, test_size= 0.25, random_state=0)
from sklearn.preprocessing import StandardScaler
st_x= StandardScaler()
x_train= st_x.fit_transform(x_train)
x_test= st_x.transform(x_test)
```

This step is common for all ML tasks and here I have just split the dataset and scaled it for further processing.

4) The KNN Algorithm(Best value of K):

```
error = []
# Calculating error for K values between 1 and 30
for i in range(1, 30):
    knn = KNeighborsClassifier(n_neighbors=i)
    knn.fit(x_train, y_train)
    pred_i = knn.predict(x_test)
    error.append(np.mean(pred_i != y_test))
plt.figure(figsize=(12, 6))
plt.plot(range(1, 30), error, color='red', linestyle='dashed', marker='o',
         markerfacecolor='blue', markersize=10)
plt.title('Error Rate K Value')
plt.xlabel('K Value')
plt.ylabel('Mean Error')
print("Minimum error:-",min(error),"at K =",error.index(min(error))+1)
```

```
# Output => Minimum error:- 0.13157894736842105 at K = 7
```

- Initialize K number of neighbors that our model need to take into consideration to predict the outcome

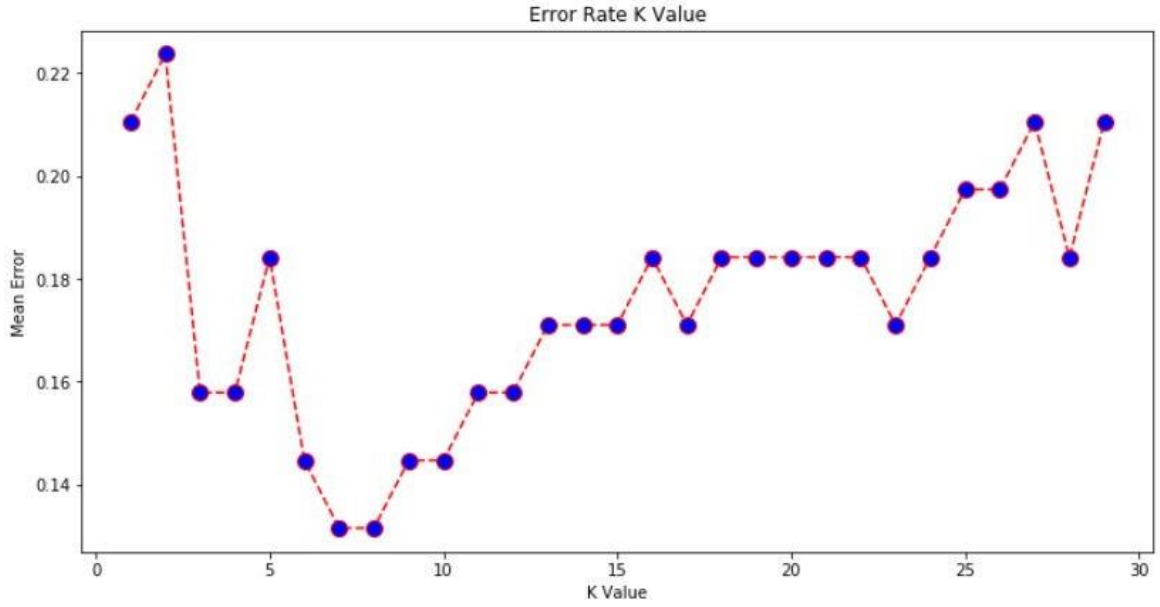
- When the model gets an input (real-world data) as an query, it will predict whether the individual has the heart disease or not by calculating the distance between this new unseen data and all other records in which our model is trained.

- After calculating the distance/similarity, sort the examples from shortest to longest (ascending order) by the distances.

- Pick the first K entries from the sorted collection and get the labels of the selected K entries.

- For regression, the algorithm returns the “mean” and if classification, the algorithm returns the “mode” of the K labels.

- Calculate the model accuracy score, to check how well the model performs in the real world.



5) Applying KNN :-

```
classifier= KNeighborsClassifier(n_neighbors=7)
classifier.fit(x_train, y_train)
y_pred= classifier.predict(x_test)
from sklearn.metrics import confusion_matrix
cm= confusion_matrix(y_test, y_pred)
```

```
# Output =>array([[26,  7],
                  [ 3, 40]], dtype=int64)
```

This way we can see our confusion matrix. Here I specified the k value as 7 as we got the lowest mean error at 7.

6) Accuracy :-

```
accuracy_score(y_test, y_pred)
```

```
# Output => 0.868421052631579
```

We got 86% accuracy on 25% of the dataset and this is a good sign. We could improve them by performing more hyperparameter tuning.

Applications of KNN:

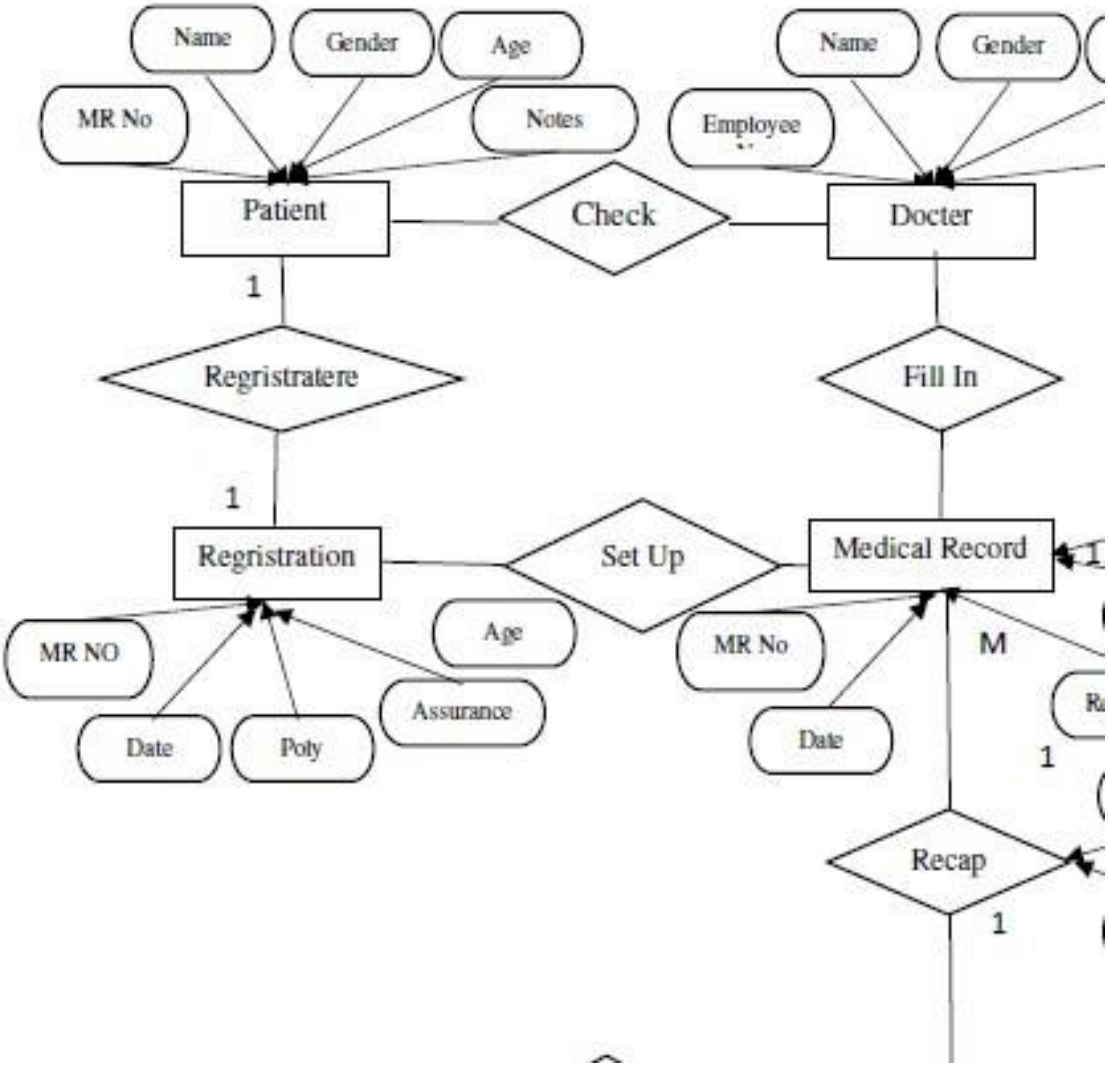
Now we know about KNN and how to implement them. Let's see some scenarios where KNN is used.

1. Music Recommendation System: Probably any recommendation system. But, in the case of music systems, we have a large amount of music coming and there is a high chance that we are getting the same music with different versions being recommended, These could be analyzed using KNN. We could even use it to see which music is of the person's liking.

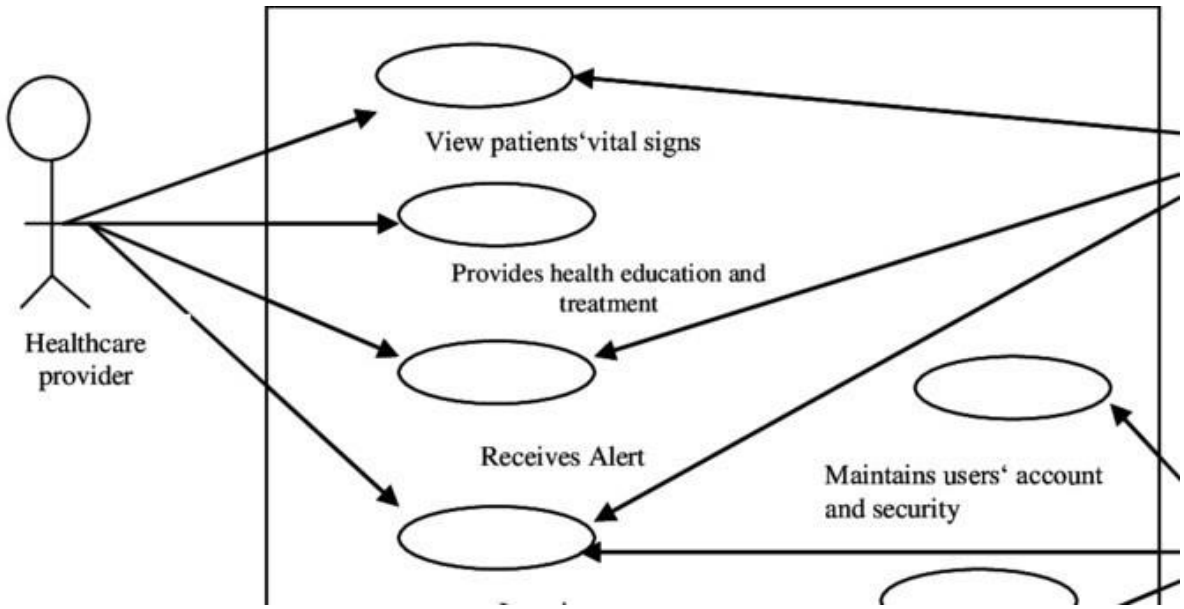
2. Outlier Detection: KNN has the ability to identify outliers.

3. Similar documents can be identified using KNN Algorithm.

ER Diagram:-

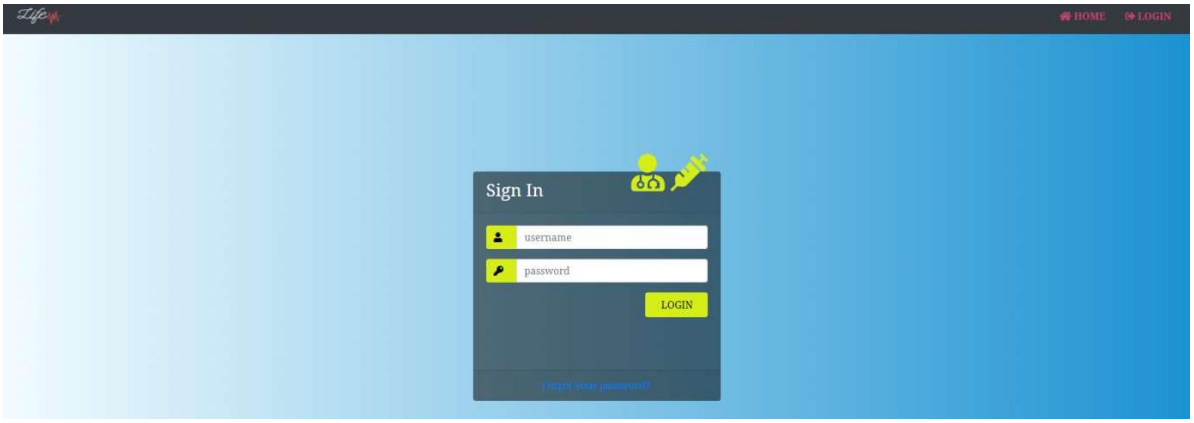

Name
Gender
Age
MR No
Notes
Patient
Check
Name
Gender
Employee
Docter
1
Regristratere
Fill In
1
Regristration
Set Up
Medical Record
1
MR NO
Age
Assurance
Date
Poly
MR No
M
Date
1
Recap
1

UML Diagram:-


View patients'vital signs
Provides health education and treatment
Healthcare provider
Receives Alert
Maintains users' account and security

The Project UI

The project UI will be made after the deployment of our ML model in clouds. Though a sample of UI is available.



Code Snippet

Importing required libraries and loading the dataset heart.csv

In [1]:
```
import numpy as np
import pandas as pd
import os
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
os.chdir("C:/Users/Shyam/Desktop")
```

Importing Data and having a brief look using pandas

In [2]:
```
dataset = pd.read_csv("heart.csv")
dataset.info()
dataset.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       303 non-null    int64
 1   sex       303 non-null    int64
 2   cp        303 non-null    int64
 3   trestbps  303 non-null    int64
 4   chol      303 non-null    int64
 5   fbs       303 non-null    int64
 6   restecg   303 non-null    int64
 7   thalach   303 non-null    int64
 8   exang     303 non-null    int64
 9   oldpeak   303 non-null    float64
 10  slope     303 non-null    int64
 11  ca        303 non-null    int64
 12  thal      303 non-null    int64
 13  target    303 non-null    int64
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
```

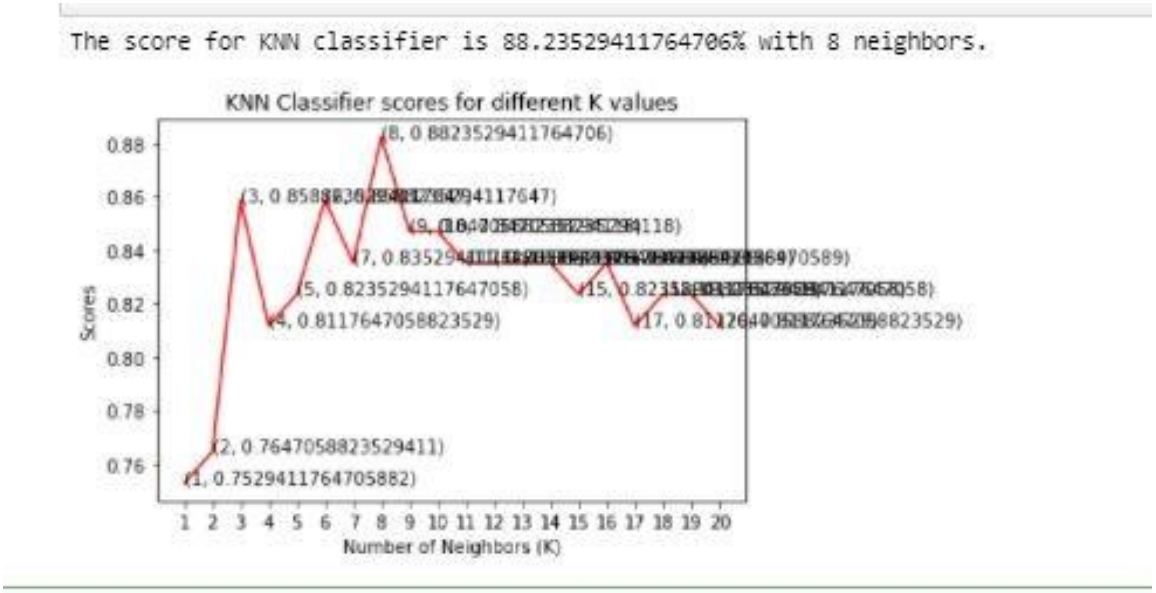
```
dataset = pd.get_dummies(dataset, columns = ['sex', 'cp', 'fbs', 'restecg', 'exang', 'slope', 'ca', 'thal'])
standardScaler = StandardScaler()
columns_to_scale = ['age', 'trestbps', 'chol', 'thalach', 'oldpeak']
dataset[columns_to_scale] = standardScaler.fit_transform(dataset[columns_to_scale])
y = dataset['target']
X = dataset.drop(['target'], axis = 1)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.28, random_state = 0)
knn_scores = []
for k in range(1,21):
    knn_classifier = KNeighborsClassifier(n_neighbors = k)
    knn_classifier.fit(X_train, y_train)
    knn_scores.append(knn_classifier.score(X_test, y_test))
plt.plot([k for k in range(1, 21)], knn_scores, color = 'red')
for i in range(1,21):
    plt.text(i, knn_scores[i-1], (i, knn_scores[i-1]))
plt.xticks([i for i in range(1, 21)])
plt.xlabel('Number of Neighbors (K)')
plt.ylabel('Scores')
plt.title('KNN Classifier scores for different K values')
print("The score for KNN classifier is {}% with {} neighbors.".format(knn_scores[7]*100, 8))
```

```
The score for KNN classifier is 88.23529411764706% with 8 neighbors.
```

Code for our ML model is written and implemented on jupyter notebook but user can use other software's like VS Code, etc.

Chapter 4

4.1Result and Discussion



After importing libraries and loading the dataset heart.csv , we check the dataset using pandas , we get the result that the dataset is a perfect dataset with no null values. Now using standard scaler we will scale the data set and after that will apply machine learning to it.

We then calculate the score of KNN Classifier (for 8 neighbours i.e. k=8) which comes 88.235 %. Using KNN classifier which scored best compared to other classifiers , I have acquired the scored of 88.23 after splitting the dataset in 72-28 which was the highest score till now.

Deploying a ML Model on Web or Device

Working with data is one thing, but deploying a machine learning model to production can be another.Data engineers are always looking for new ways to deploy their machine learning models to production. They want the best performance, and they care about how much it costs.Well, now you can have both! Let's take a look at the deployment process and see how we can do it successfully!

Most data science projects deploy machine learning models as an **on-demand prediction service** or in **batch prediction** mode. Some modern applications deploy **embedded models** in edge and mobile devices. Each model has its own merits. For example, in the batch scenario, optimizations are done to minimize model compute cost. There are fewer dependencies on external data sources and cloud services. The local processing power is sometimes sufficient for computing algorithmically complex models.It is also easy to debug an offline model *when* failures *occur or* tune hyperparameters since it runs on powerful servers.

On the other hand, web services can provide *cheaper* and near real-time predictions. Availability of CPU power is less of an issue if the model runs on a cluster or cloud service. The model can be easily made available to other applications through API calls and so on. One of the main benefits of embedded machine learning is that we can customize it to the requirements of a specific device.We can easily deploy the model to a device, and its runtime environment cannot be tampered with by an external party. A clear drawback is that the device needs to have enough computing power and storage space.

Deploying machine learning models as web services.

The simplest way to deploy a machine learning model is to create a web service for prediction. In this example, we use the Flask web framework to wrap a simple random forest classifier built with scikit-learn.

To create a machine learning web service, you need at least three steps.

The first step is to create a machine learning model, train it and validate its performance. The following script will train a random forest classifier. Model testing and validation are not included here to keep it simple. But do remember those are an integral part of any machine learning project.

Deploying machine learning models for batch prediction.

While online models can serve prediction, on-demand batch predictions are sometimes preferable. Offline models can be optimized to handle a high volume of job instances and run more complex models. In batch production mode, you don't need to worry about scaling or managing servers either.

Batch prediction can be as simple as calling the predict function with a data set of input variables. The following command does it.

prediction = classifier.predict(UNSEEN_DATASET)

Deploying machine learning models on edge devices as embedded models.

Computing on edge devices such as mobile and IoT has become very popular in recent years. The benefits of deploying a machine learning model on edge devices include, but are not limited to: Reduced latency as the device is likely to be close to the user than a server far away.

Reduce data bandwidth consumption as we ship processed results back to the cloud instead of raw data that requires big size and eventually more bandwidth. Edge devices such as mobile and IoT devices have limited computation power and storage capacity due to the nature of their hardware. We cannot simply deploy machine learning models to these devices directly, especially if our model is big or requires extensive computation to run inference on them.

6.2 Software Requirements

A major element in building a system is the section of compatible software since the software in the market is experiencing in geometric progression. Selected software should be acceptable by the firm and one user as well as it should be feasible for the system.

This document gives a detailed description of the software requirement specification. The study of requirement specification is focused specially on the functioning of the system. It allows the developer or analyst to understand the system, function to be carried out the performance level to be obtained and corresponding interfaces to be established.

6.2.1 Python

Python Language Introduction

Python is a general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently. 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 Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

- **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

- **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands. Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL). Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.
Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.
Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol- 68, SmallTalk, anUnix shell and other scripting languages.Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Python Features

Python's features include −

- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.

- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.

- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.

- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

- **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.

- **Databases** − Python provides interfaces to all major commercial databases.

- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

- **Scalable** − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has other good features, few are listed below:

- It supports functional and structured programming methods as well as OOP.

- It can be used as a scripting language or can be compiled to byte-code for building large applications.

- It provides very high-level dynamic data types and supports dynamic type checking.

- IT supports automatic garbage collection.

It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

VISUAL STUDIO CODE

Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging.

First and foremost, it is an editor that gets out of your way. The delightfully frictionless edit-build-debug cycle means less time fiddling with your environment, and more time executing on your ideas.

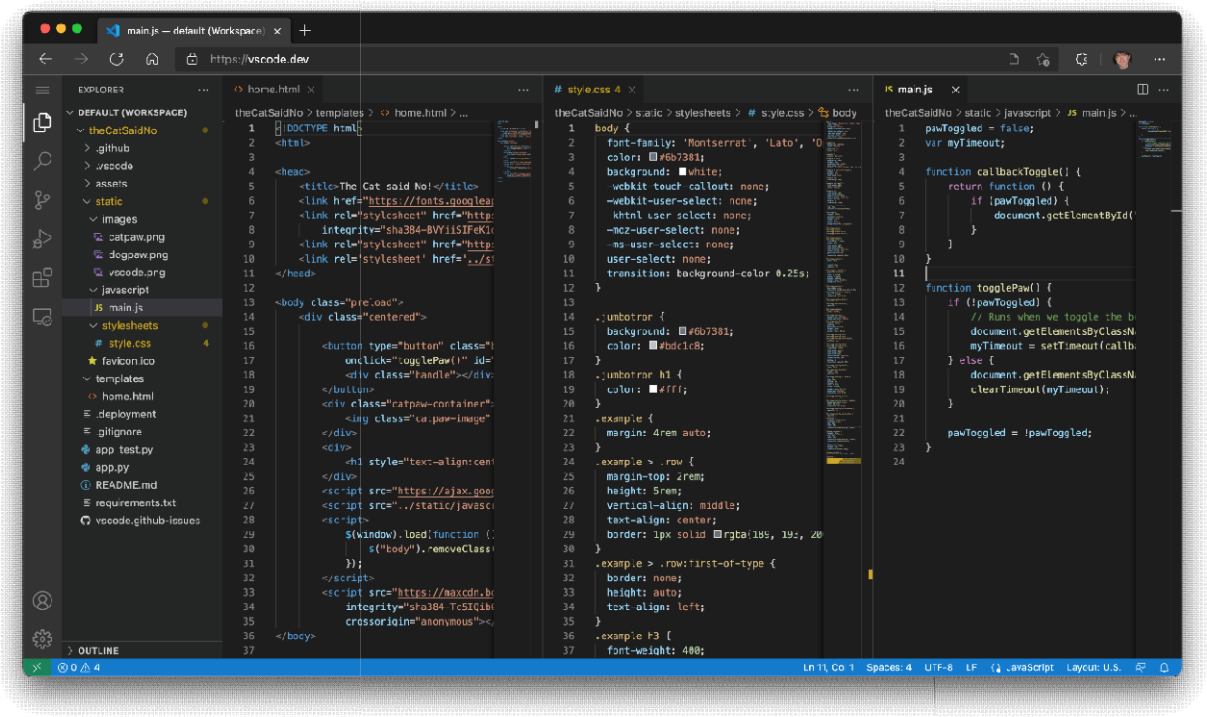


FIG 19 : VSCODE IDE

Mockitt

Mockitt is an online prototyping and collaboration tool to empower your design journey, preset your idea, and validate your concept.

The product is suitable for any user group, from freelancers interested in prototyping to product managers, ux/ui designers, etc., we provide convenient, fast and powerful tools to help every designer.

DEPLOYMENT

- Cloud, SaaS, Web-Based
- Desktop - Mac
- Desktop - Windows

Mockitt Features :-

- Animation
- Collaboration Tools
- Drag & Drop
- Software Prototyping
- Templates
- UI Prototyping
- Usability Testing
- UX Prototyping
- Version Control

Chapter 5

Conclusion

We successfully completed this project "Cardiovascular Disease Prediction using KNN Algorithm" under the guidance of our respected supervisor and group mates. We assure that all the methods used by us are 100% working.

Manually determining the odds of cardiovascular disease based on risk factors can be hard. Using Machine learning `techniques, we can predict the outcome with the help of existing data. But still, we can't trust the machine always.

In this paper, we developed the proposed system to predict heart disease. Ensemble methods (boosting and bagging) with feature extraction algorithms (PCA and LDA) are used to improve predicting heart disease performance. -e feature extraction algorithms are used to extract essential features from the Cleveland heart disease dataset. Comparison between ensemble methods (boosting and bagging) and five classifiers (KNN, SVM, NB, DT, and RF) is applied to selected features.The experimental results showed that the bagging ensemble learning algorithm with DT and PCA feature extraction method had achieved the best performance.

As you can see from this prediction, we got some percentage of "False positives and False negatives". The only way to prevent cardiovascular disease is to **stay healthy.**

FUTURE SCOPE

For the 13 features which were in the dataset, K Neighbors classifier performed better in the ML approach when data preprocessing is applied.

The computational time was also reduced which is helpful when deploying a model. It was also found out that the dataset should be normalized; otherwise, the training model gets overfitted sometimes and the accuracy achieved is not sufficient when a model is evaluated for real-world data problems which can vary drastically to the dataset on which the model was trained. It was also found out that the statistical analysis is also important when a dataset is analyzed and it should have a Gaussian distribution, and then the outlier's detection is also important and a technique known as Isolation Forest is used for handling this. The difficulty which came here is that the sample size of the dataset is not large. If a large dataset is present, the results can increase very much in deep learning and ML as well. The algorithm applied by us in ANN architecture increased the accuracy which we compared with the different researchers. The dataset size can be increased and then deep learning with various other optimizations can be used and more promising results can be achieved. Machine learning and various other optimization techniques can also be used so that the evaluation results can again be increased. More different ways of normalizing the data can be used and the results can be compared. And more ways could be found where we could integrate heart-disease-trained ML and DL models with certain multimedia for the ease of patients and doctors.

References

World Health Organization, *Cardiovascular Diseases*, WHO, Geneva, Switzerland, 2020, https://www.who.int/health-topics/cardiovascular- diseases/#tab=tab_1.

1. American Heart Association, *Classes of Heart Failure*, American Heart Association, Chicago, IL, USA, 2020, https://www.heart.org/en/health-topics/heart-failure/what-is-heart-failure/classes-of-heart-failure.
2. American Heart Association, *Heart Failure*, American Heart Association, Chicago, IL, USA, 2020, https://www.heart.org/en/health-topics/heart-failure.
3. S. Shalev-Shwartz and S. Ben-David, "Understanding machine learning," *From Theory to Algorithms*, Cambridge University Press, Cambridge, UK, 2020.View at: Google Scholar
4. T. Hastie, R. Tibshirani, and J. Friedman, "The elements of statistical learning," *Data Mining, Inference, and Prediction*, Springer, Cham, Switzerland, 2020.View at: Google Scholar
5. S. Marsland, "Machine learning," *An Algorithmic Perspective*, CRC Press, Boca Raton, FL, USA, 2020.View at: Google Scholar
6. P. Melillo, N. De Luca, M. Bracale, and L. Pecchia, "Classification tree for risk assessment in patients suffering from congestive heart failure via long-term heart rate variability," *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 3, pp. 727–733, 2013.View at: Publisher Site | Google Scholar
7. M. M. A. Rahhal, Y. Bazi, H. Alhichri, N. Alajlan, F. Melgani, and R. R. Yager, "Deep learning approach for active classification of electrocardiogram signals," *Information Sciences*, vol. 345, pp. 340–354, 2016.View at: Publisher Site | Google Scholar
8. G. Guidi, M. C. Pettenati, P. Melillo, and E. Iadanza, "A machine learning system to improve heart failure patient assistance," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 6, pp. 1750–1756, 2014.View at: Publisher Site | Google Scholar

Bioinformatics and Biomedicine (BIBM), pp. 1296–1299, IEEE, Kansas City,