KATHMANDU UNIVERSITY

SCHOOL OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

MINI PROJECT REPORT

"Health Analyzer"

A **Fourth year/ First Semester** Artificial Intelligence [COMP 472]
Mini Project Report submitted in partial fulfilment of the requirements
for the degree of Bachelor of Engineering.

By:

Ashish Pokhrel (38)
Dinesh Poudel (39)
Shusant Sapkota (43)

Supervised By:

Mr. Santosh Khanal
Department of Computer Science and Engineering

August 2021

ABSTRACT

Recently, many researchers have designed various automated prediction models using various supervised learning models. A disease prediction model helps prediction of disease which may control the death rate due to these diseases. It will also help the doctors and patient to easily study about the possible results in an efficient manner. In this project, we have designed a disease prediction model implementing various machine learning algorithms, which predicts the chances of getting various diseases like corona, heart attack, breast cancer. The model has been implemented on a beautifully designed website through which the user can enter their data. These data are then processed in our model and the result are displayed in real time. Also, our project has medicine recommender which checks the condition of the user and recommends the most feasible medicine. The main purpose of this project is to help doctors, patients and lab technicians to carry out checkups for possible disease in an efficient way.

Keywords: *Prediction, Disease, Machine Learning Algorithm, etc.*

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1 Background

Machine learning is used in various areas like education and healthcare. Machine learning is used in healthcare in vast areas. The healthcare sector produces large amounts of data in terms of images, patient data, and so on that helps to identify patterns and make predictions. Machine learning is used in healthcare to solve various problems. Heart disease is based on the individual, and the extent of heart disease can vary from person to person. Thus, making a machine learning model, training it on the dataset, and entering individual patient details can help in prediction. The prediction result will be according to the data entered and hence will be specific to that individual. Coronavirus is a disease that has no clearly defined treatment. The coronavirus 2019 (COVID-19) originated from China. There are different treatments that are going on for it but there are no clearly defined steps for treatment.

Artificial intelligence (AI) aims to mimic human cognitive functions. It is bringing a paradigm shift to healthcare, powered by the increasing availability of healthcare data and rapid progress of analytics techniques. Recently, many models have been developed for automated diagnosis of various diseases such as cancer, COVID-19, and diabetes. Recently, many researchers have started using machine learning models for real-time diagnosis of disease. However, efficient early stage diagnosis is still defined as an ill-posed problem. Recently, many researchers have started using deep-learning models to obtain significantly better performance as compared with the machine learning models.

In this study, the machine learning models are applied to the coronavirus, heart disease, and cancer dataset to predict the risk of these diseases in an individual. An end-to-end process is used where people must enter their details in the web application and submit the data. The real-time processing takes place, and the risk is predicted within a few seconds. The trained parameters of the model are stored and prediction is done in real-time.

1.2 Problem Statement

Visiting hospital for a regular checkup seem risky these days as the corona virus has taken the life of millions. There is no guarantee that the doctor or the workers working in the hospitals are negative to the virus. So, for a patient visiting the hospital for normal checkup, it is possible that he might return home carrying the virus along him. Also, the human errors in the report might cause false positive results. So, it Is important to find a way to carry out the normal checkups with least possible contact with the health personals along with minimizing the human errors.

1.3 Objectives

The project has multiple purposes. Some of its key objectives are:

1. To develop a model that helps doctors to easily obtain the results of a suspected disease.
2. To help the patients to analyze the disease they have.
3. To help the lab technicians to get a efficient report without any human error.

1.4 Motivation and Significance

In current pandemic situation, hospitals are crowded and the risk of getting infected with the virus comes along with it. This has raised a state of fear among the patients to visit the hospital for checkup of the disease they are infected with. The problem not only resides with the patients but has the equal impact to the doctors and the co-workers in the hospital. So, there seem to be a necessity, to create a way to make the interaction between the doctor and patients less risky. It can be done using a system which will be able to predict the chances of a patient getting the particular disease. Keeping this in mind, we have developed a model which can predict the chances of getting a disease based on the information that the patient or the doctor has obtained from the lab report.

CHAPTER 2
LITERATURE REVIEW

Symptomate

Symptomate is an advanced system for preliminary medical diagnosis. It helps to identify the disease; a patient has based on the symptoms he/she has faced. Symptomate asks a series of questions. These questions are taken as the dataset and then it is further processed in the model. The model evaluates the symptoms using various machine learning algorithms. Based on the processed data a disease that the patients might be suffering is displayed on the screen. Symptomate predicts the possible cause of the symptoms and even provides the options for the next step. It also suggests the lab test based on the symptoms.

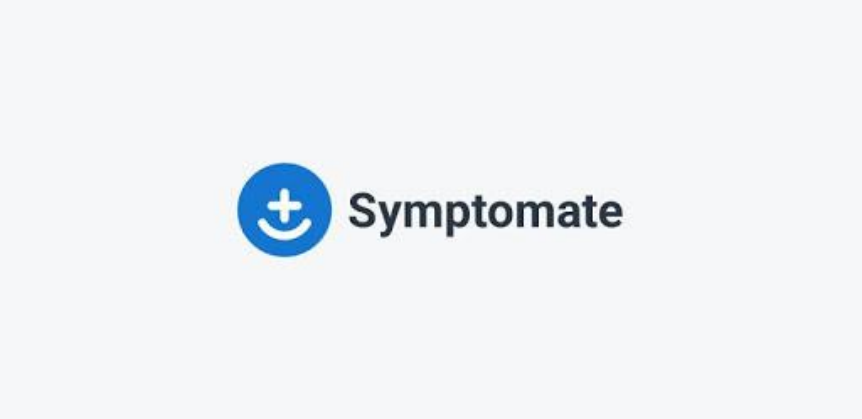


Figure 2.1: Symptomate

CHAPTER 3
METHODOLOGY

Design and Implementation

In order to complete a project successfully, certain steps and methods are to be followed strictly. While performing this project we strictly followed the design plans and implemented. Firstly, we made flow diagrams to analyses our project and work according to it.

3.1 Flow Diagram

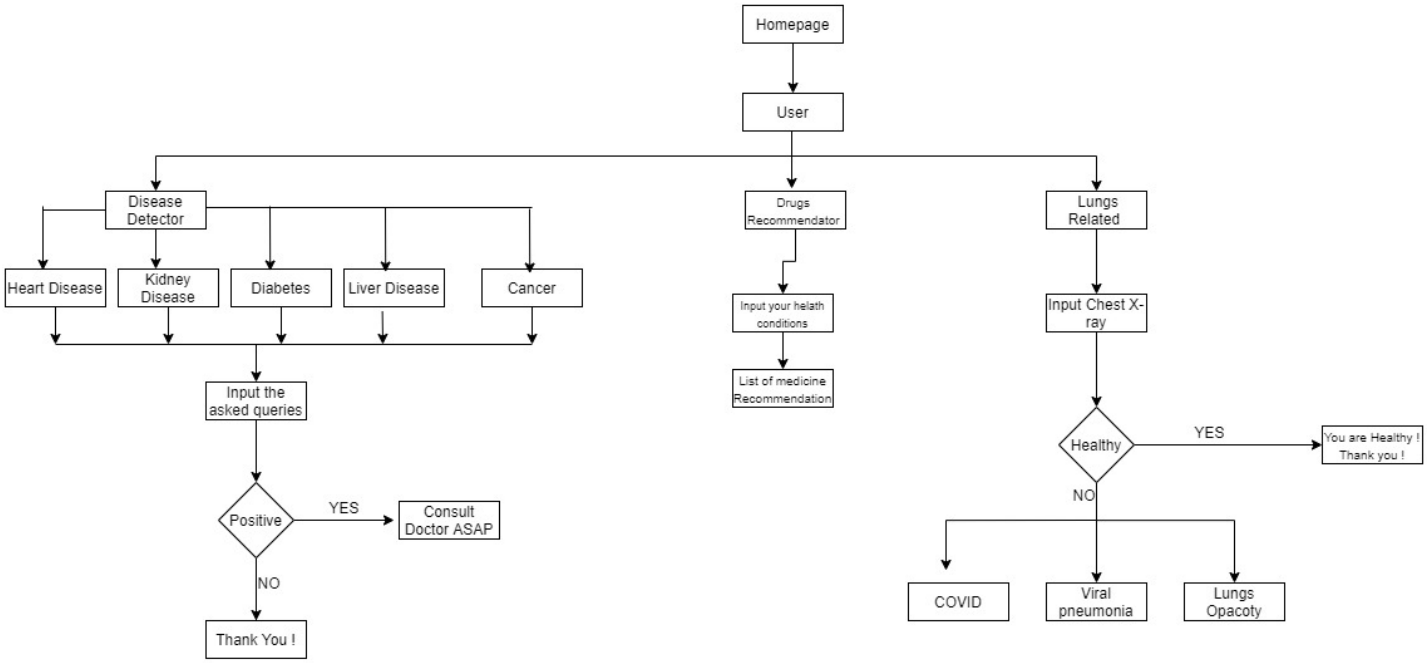


Figure 3.1 Flow Diagram

Users accessing our project will be able to know whether the specific patient is suffering from various types of disease or Not. Here the user needs to feel the queries form and based on the data our project will predict whether they have heart disease, kidney disease, diabetes, Liver Disease, Cancer or Not. Our Health Analyzer will also recommend the medicines based on the condition of the user. Similarly, users can also check various disease related to lings by entering their chest x-ray in our project.

3.2 Algorithms

- **Random Forest Algorithm:**

  Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.As the name suggests, Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.
  The below diagram explains the working of the Random Forest algorithm:

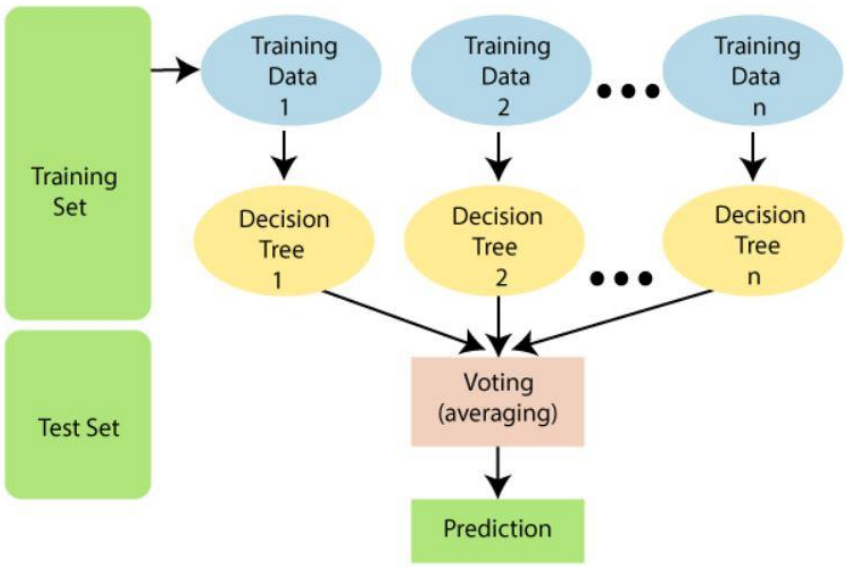


Figure 3.2.1: Random Forest Algorithm

- **Decision Tree:**

  Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.
  In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

The decisions or the test are performed on the basis of features of the given dataset. It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions. It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure. In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm. A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.

Below diagram explains the general structure of a decision tree:

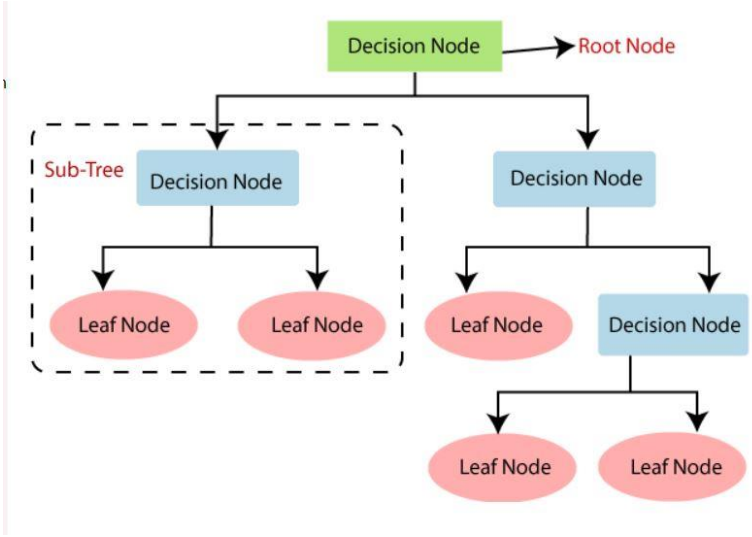
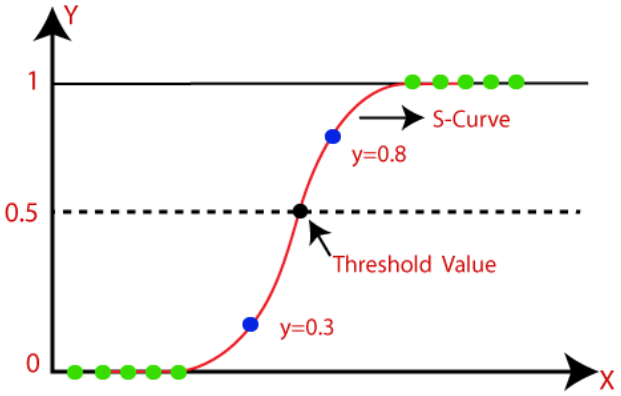


Figure 3.2.2: Decision Tree

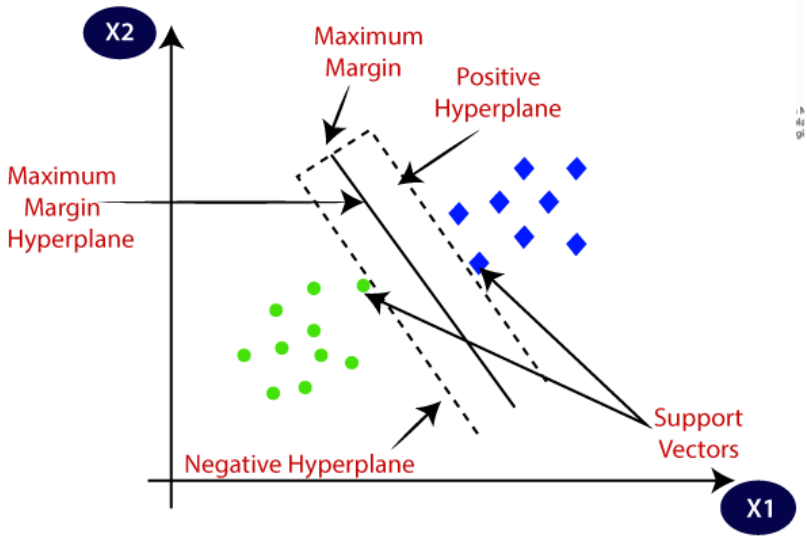
- **Logistic Regression:**

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes. In simple words, the dependent variable is binary in nature having data coded as either 1 (stands for success/yes) or 0 (stands for failure/no). Mathematically, a logistic regression model predicts P(Y=1) as a function of X. It is one of the simplest ML algorithms that can be used for various classification problems such as spam detection, Diabetes prediction, cancer detection etc.



- **SVM:**

  Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:



- **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.

3.3 Tools

These are the tools that were used while developing this Health Analyzer model, and these very tools must be available if one wishes to use or test the model prepared.

A. Numpy & Matplotlib
   NumPy is a library for Python to handle large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. Matplotlib is a plotting library for Python and its numerical mathematics extension NumPy. It is used here for visualizing the data.

B. Pandas
   Pandas is a library for Python to manipulate and analyse data. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is used here to work with data extracted from the videos.

C. Scikit-learn
   Scikit-learn (also known as sklearn) is a software machine learning library for Python. It features various classification, regression and clustering algorithms including support vector machines, random forests, k-means and many more. Sklearn integrates well with many other Python libraries, such as matplotlib for plotting, numpy for array vectorization and pandas dataframes.

D. Seaborn
   Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. It provides beautiful default styles and colour palettes to make statistical plots more attractive. It is built on the top of the matplotlib library and also closely integrated to the data structures from pandas.

E. Keras
   Keras is a powerful and easy-to-use free open source Python library for developing and evaluating deep learning models. It wraps the efficient numerical computation libraries Theano and TensorFlow and allows you to define and train neural network models in just a few lines of code.

F. Tensorflow

   It is an open source artificial intelligence library, using data flow graphs to build models. It allows developers to create large-scale neural networks with many layers. TensorFlow is mainly used for: classification, perception, understanding, discovering, prediction and creation.

G. Graphviz:

Graphviz (short for Graph Visualization Software) is a package of open-source tools initiated by AT&T Labs Research for drawing graphs specified in DOT language scripts having the file name extension "gv". It also provides libraries for software applications to use the tool.

3.4 Procedure

3.4.1 Data Collection

A. Breast Cancer Wisconsin (Diagnostic) Data Set

Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image.

Attribute Information:

- ID number
- Diagnosis (M = malignant, B = benign)

Ten real-valued features are computed for each cell nucleus:

- radius (mean of distances from center to points on the perimeter)
- texture (standard deviation of gray-scale values)
- perimeter
- area
- smoothness (local variation in radius lengths)
- compactness (perimeter^2 / area - 1.0)
- concavity (severity of concave portions of the contour)
- concave points (number of concave portions of the contour)
- symmetry
- fractal dimension ("coastline approximation" - 1)

The mean, standard error and "worst" or largest (mean of the three largest values) of these features were computed for each image resulting in 30 features. For instance, field 3 is Mean Radius, field 13 is Radius SE, field 23 is Worst Radius.

B. Pima Indians Diabetes Database (Diabetes Dataset)

This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from

a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage.
The dataset consists of several medical predictor variables and one target variable, Outcome. Predictor variables includes the number of pregnancies the patient has had, their BMI, insulin level, age, and so on.

- Pregnancies: Number of times pregnant
- Glucose: Plasma glucose concentration a 2 hours in an oral glucose tolerance test
- BloodPressure: Diastolic blood pressure (mm Hg)
- SkinThickness: Triceps skin fold thickness (mm)
- Insulin: 2-Hour serum insulin (mu U/ml)
- BMI: Body mass index (weight in kg/(height in m)^2)
- DiabetesPedigreeFunction: Diabetes pedigree function
- Age: Age (years)
- Outcome: Class variable (0 or 1)

C. Heart Dataset

This database contains 76 attributes, but all published experiments refer to using a subset of 14 of them. In particular, the Cleveland database is the only one that has been used by ML researchers to this date. The "target" field refers to the presence of heart disease in the patient.

- Age : Age of the patient
- Sex : Sex of the patient
- exang: exercise induced angina (1 = yes; 0 = no)
- ca: number of major vessels (0-3)
- cp : Chest Pain type chest pain type
- Value 1: typical angina
- Value 2: atypical angina
- Value 3: non-anginal pain
- Value 4: asymptomatic
- trtbps : resting blood pressure (in mm Hg)
- chol : cholestoral in mg/dl fetched via BMI sensor
- fbs : (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)
- rest_ecg : resting electrocardiographic results
- Value 0: normal
- Value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV)
- Value 2: showing probable or definite left ventricular hypertrophy by Estes' criteria
- thalach : maximum heart rate achieved
- target : 0= less chance of heart attack 1= more chance of heart attack

D. Kidney Dataset

The dataset was taken over 2-month period in India in Apollo Hospital. It has 400 rows with 25 features like red blood cells, pedal edema, sugar, etc. The aim is to classify whether a patient has chronic kidney disease or not. The classification is based on a attribute named 'classification' which is either 'ckd' (chronic kidney disease) or ‘notckd’. I've performed cleaning of the dataset which includes mapping the text to numbers and some other changes. After the cleaning I've done some EDA (Exploratory Data Analysis) and then I've divided the dataset int training and testing and applied the models on them. It is observed that the classification results are not much satisfying initially. So, instead of dropping the rows with Nan values I've used the lambda function to replace them with mode for each column. After that I've divided the dataset again into training and testing sets and applied models on them. This time the results are better and we see that the random forest and decision trees are the best performers with an accuracy of 1.0 and 0 misclassifications. The performance of the classification is measured by printing confusion matrix, classification report and accuracy.

- age - age
- bp - blood pressure
- sg - specific gravity
- al - albumin
- su - sugar
- rbc - red blood cells
- pc - pus cell
- pcc - pus cell clumps
- ba - bacteria
- bgr - blood glucose random
- bu - blood urea
- sc - serum creatinine
- sod - sodium
- pot - potassium
- hemo - hemoglobin
- pcv - packed cell volume
- wc - white blood cell count
- rc - red blood cell count
- htn - hypertension
- dm - diabetes mellitus
- cad - coronary artery disease
- appet - appetite
- pe - pedal edema
- ane - anemia
- class – class

E. Liver Dataset

The original data set contains 416 liver patient records and 167 non-liver patient records collected from North East of Andhra Pradesh, India. Any patient whose age exceeded 89 is listed as being of age "90".

The dataset has been preprocessed to remove samples with missing values and convert categorical features to numerical data. After this stage, the dataset contains 414 liver patient records and 165 records for healthy subjects.

Attribute information:

- Age: Age of the patient
- Female: Gender of the patient (1 if Female, 0 if Male)
- TB: Total Bilirubin
- DB: Direct Bilirubin
- Alkphos: Alkaline Phosphotase
- Sgpt: Alamine Aminotransferase
- Sgot: Aspartate Aminotransferase
- TP: Total Protiens
- ALB: Albumin
- A/R: Albumin and Globulin Ratio

F. Lungs Dataset

A team of researchers from Qatar University, Doha, Qatar, and the University of Dhaka, Bangladesh along with their collaborators from Pakistan and Malaysia in collaboration with medical doctors have created a database of chest X-ray images for COVID-19 positive cases along with Normal and Viral Pneumonia images. This COVID-19, normal, and other lung infection dataset is released in stages. In the first release, we have released 219 COVID-19, 1341 normal, and 1345 viral pneumonia chest X-ray (CXR) images. In the first update, we have increased the COVID-19 class to 1200 CXR images. In the 2nd update, we have increased the database to 3616 COVID-19 positive cases along with 10,192 Normal, 6012 Lung Opacity (Non-COVID lung infection), and 1345 Viral Pneumonia images. We will continue to update this database as soon as we have new x-ray images for COVID-19 pneumonia patients.

G. Drugs Review Dataset

The UCI ML Drug Review dataset provides patient reviews on specific drugs along with related conditions and a 10-star patient rating system reflecting overall patient satisfaction. The data was obtained by crawling online pharmaceutical review sites and stored in UCI site. It can be found in Kaggle . This data was published in a

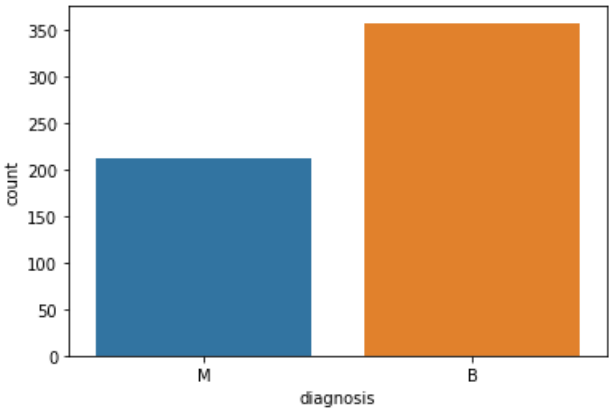
study on sentiment analysis of drug experience over multiple facets, ex. sentiments learned on specific aspects such as effectiveness and side effects (see the acknowledgments section to learn more).

3.4.2 Exploratory Data Analysis

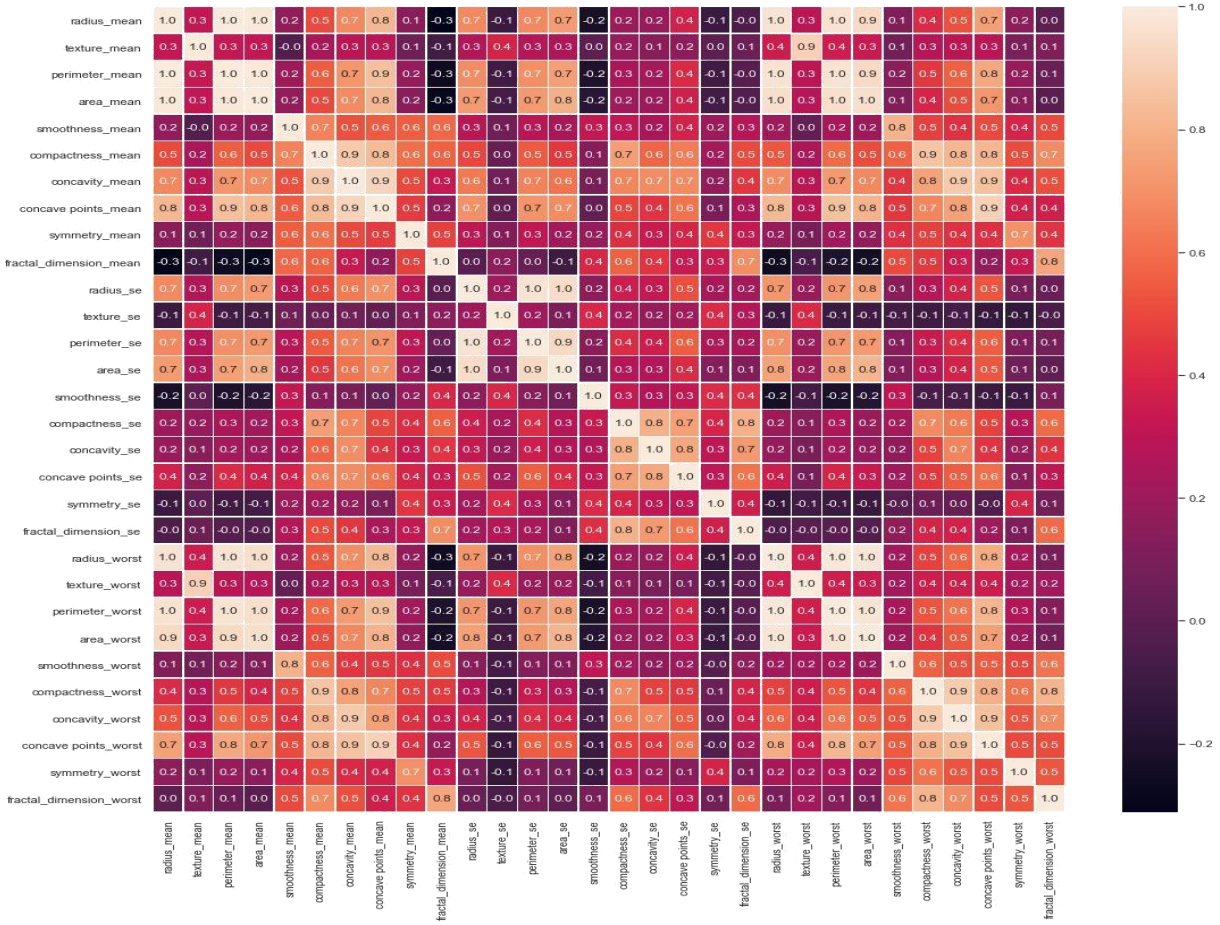
Breast Cancer

There are 32 feature variables in Breast Cancer Dataset and one target variable named as ‘diagnosis.

#	Column	Non-Null Count	Dtype
0	id	569 non-null	int64
1	diagnosis	569 non-null	object
2	radius mean	569 non-null	float64
3	texture_mean	569 non-null	float64
4	perimeter_mean	569 non-null	float64
5	area mean	569 non-null	float64
6	smoothness_mean	569 non-null	float64
7	compactness_mean	569 non-null	float64
8	concavity mean	569 non-null	float64
9	concave points_mean	569 non-null	float64
10	symmetry mean	569 non-null	float64
11	fractal_dimension_mean	569 non-null	float64
12	radius_se	569 non-null	float64
13	texture se	569 non-null	float64
14	perimeter_se	569 non-null	float64
15	area_se	569 non-null	float64
16	smoothness se	569 non-null	float64
17	compactness_se	569 non-null	float64
18	concavity se	569 non-null	float64
19	concave points_se	569 non-null	float64
20	symmetry_se	569 non-null	float64
21	fractal dimension se	569 non-null	float64
22	radius_worst	569 non-null	float64
23	texture_worst	569 non-null	float64
24	perimeter worst	569 non-null	float64
25	area_worst	569 non-null	float64
26	smoothness worst	569 non-null	float64
27	compactness_worst	569 non-null	float64
28	concavity_worst	569 non-null	float64
29	concave points worst	569 non-null	float64
30	symmetry_worst	569 non-null	float64
31	fractal_dimension_worst	569 non-null	float64
32	Unnamed: 32	0 non-null	float64



The number of people with Benign in the dataset is 357 and number of Malignant people is 212. The Correlation of numerical variables among each other can be viewed in the following heatmap.

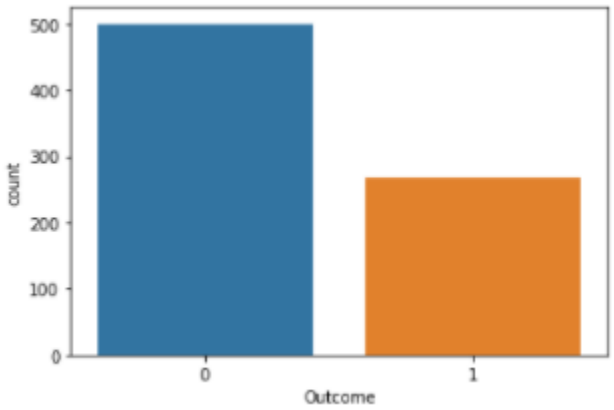


Diabetes

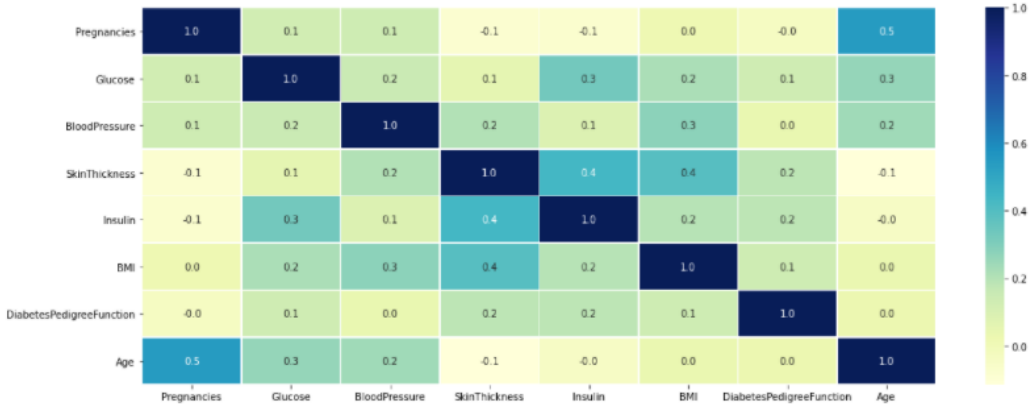
In diabetes dataset, there are eight feature variables and 'Outcome' as target variable. Out of nine variables all are numerical where all are integers except BMI and DiabetesPedigreeFunction which are floating point values.

```
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
```

The number of patients without diabetes are 500 and that without diabetes are 268

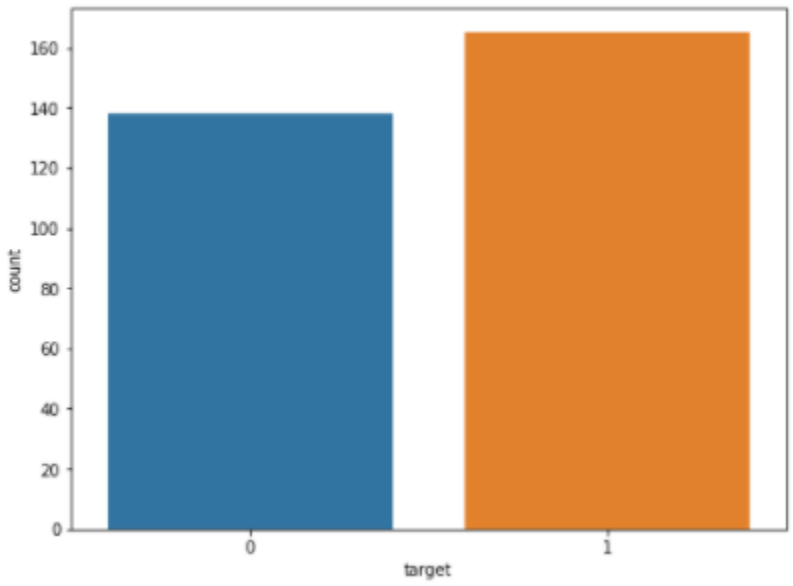


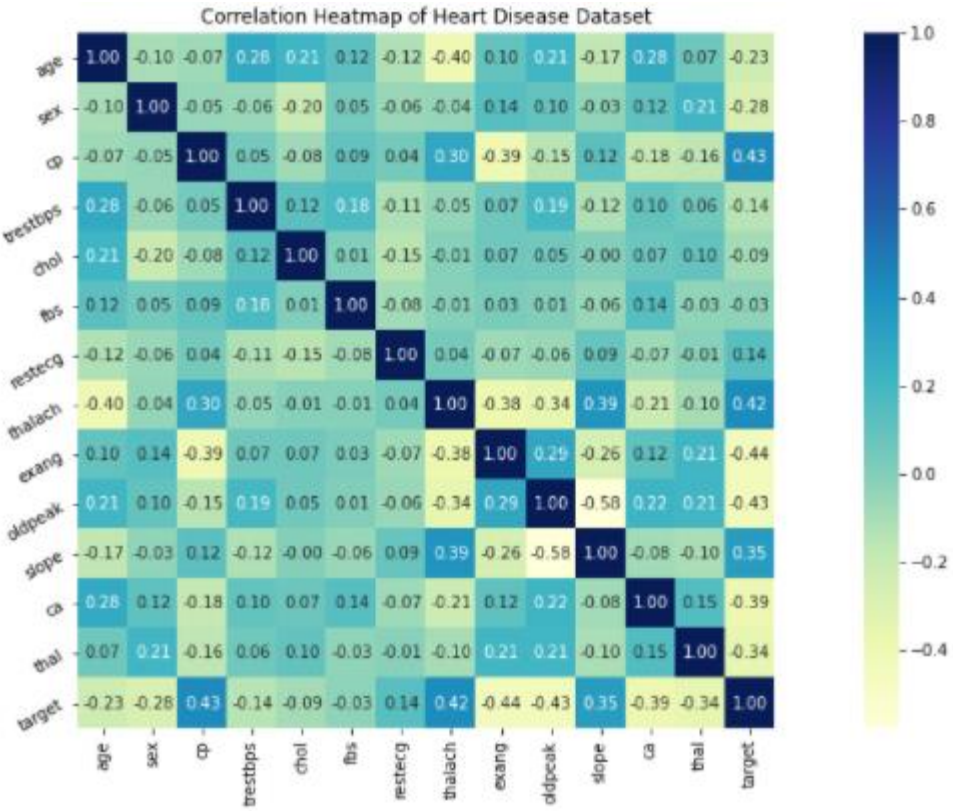
The intercorrelation between numerical variables can be viewed in the following heatmap diagram.



Heart Disease

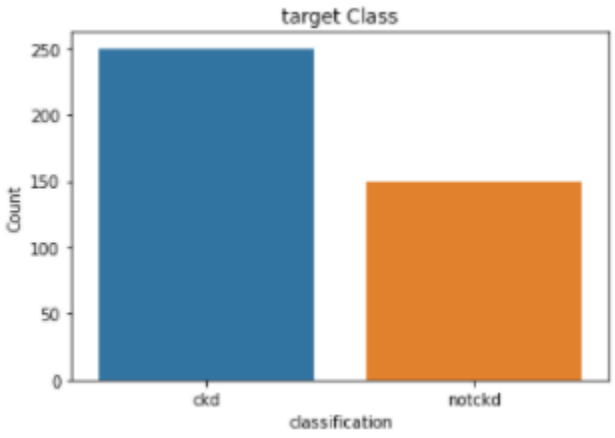
```
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       303 non-null    int64
 1   sex       303 non-null    int64
 2   cp        303 non-null    int64
 3   trestbps  303 non-null    int64
 4   chol      303 non-null    int64
 5   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
```




Correlation Heatmap of Heart Disease Dataset
age 1.00 -0.10 -0.07 0.28 0.21 0.12 -0.12 -0.40 0.10 0.21 -0.17 0.28 0.07 -0.23
sex -0.10 1.00 -0.05 -0.06 -0.20 0.05 -0.06 -0.04 0.14 0.10 -0.03 0.12 0.21 -0.28
cp -0.07 -0.05 1.00 0.05 -0.08 0.09 0.04 0.30 -0.39 -0.15 0.12 -0.18 -0.16 0.43
trestbps 0.28 -0.06 0.05 1.00 0.12 0.18 -0.11 -0.05 0.07 0.19 -0.12 0.10 0.06 -0.14
chol 0.21 -0.20 -0.08 0.12 1.00 0.01 -0.15 -0.01 0.07 0.05 -0.00 0.07 0.10 -0.09
fbs 0.12 0.05 0.09 0.18 0.01 1.00 -0.08 -0.01 0.03 0.01 -0.06 0.14 -0.03 -0.03
restecg -0.12 -0.06 0.04 -0.11 -0.15 -0.08 1.00 0.04 -0.07 -0.06 0.09 -0.07 -0.01 0.14
thalach -0.40 -0.04 0.30 -0.05 -0.01 -0.01 0.04 1.00 -0.38 -0.34 0.39 -0.21 -0.10 0.42
exang 0.10 0.14 -0.39 0.07 0.07 0.03 -0.07 -0.38 1.00 0.29 -0.26 0.12 0.21 -0.44
oldpeak 0.21 0.10 -0.15 0.19 0.05 0.01 -0.06 -0.34 0.29 1.00 -0.58 0.22 0.21 -0.43
slope -0.17 -0.03 0.12 -0.12 -0.00 -0.06 0.09 0.39 -0.26 -0.58 1.00 -0.08 -0.10 0.35
ca 0.28 0.12 -0.18 0.10 0.07 0.14 -0.07 -0.21 0.12 0.22 -0.08 1.00 0.15 -0.39
thal 0.07 0.21 -0.16 0.06 0.10 -0.03 -0.01 -0.10 0.21 0.21 -0.10 0.15 1.00 -0.34
target -0.23 -0.28 0.43 -0.14 -0.09 -0.03 0.14 0.42 -0.44 -0.43 0.35 -0.39 -0.34 1.00
age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal target
1.0 0.8 0.6 0.4 0.2 0.0 −0.2 −0.4

Kidney Disease

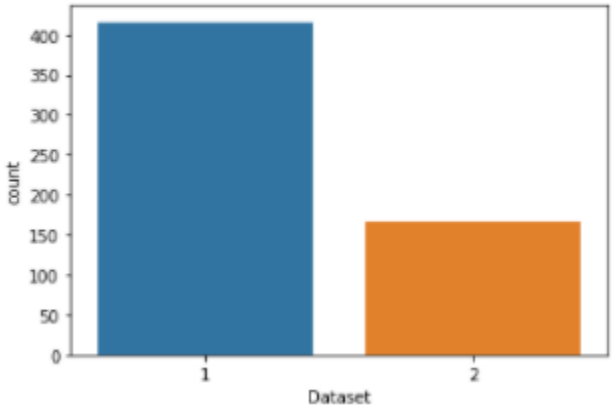
```
Data columns (total 26 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   id              400 non-null    int64
 1   age             391 non-null    float64
 2   bp              388 non-null    float64
 3   sg              353 non-null    float64
 4   al              354 non-null    float64
 5   su              351 non-null    float64
 6   rbc             248 non-null    object
 7   pc              335 non-null    object
 8   pcc             396 non-null    object
 9   ba              396 non-null    object
 10  bgr             356 non-null    float64
 11  bu              381 non-null    float64
 12  sc              383 non-null    float64
 13  sod             313 non-null    float64
 14  pot             312 non-null    float64
 15  hemo            348 non-null    float64
 16  pcv             330 non-null    object
 17  wc              295 non-null    object
 18  rc              270 non-null    object
 19  htn             398 non-null    object
 20  dm              398 non-null    object
 21  cad             398 non-null    object
 22  appet           399 non-null    object
 23  pe              399 non-null    object
 24  ane             399 non-null    object
 25  classification  400 non-null    object
```



Liver Disease

```
Data columns (total 11 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Age                         583 non-null    int64
 1   Gender                      583 non-null    object
 2   Total_Bilirubin             583 non-null    float64
 3   Direct_Bilirubin            583 non-null    float64
 4   Alkaline_Phosphotase        583 non-null    int64
 5   Alamine_Aminotransferase    583 non-null    int64
 6   Aspartate_Aminotransferase  583 non-null    int64
 7   Total_Protiens              583 non-null    float64
 8   Albumin                     583 non-null    float64
 9   Albumin_and_Globulin_Ratio  579 non-null    float64
 10  Dataset                     583 non-null    int64
```

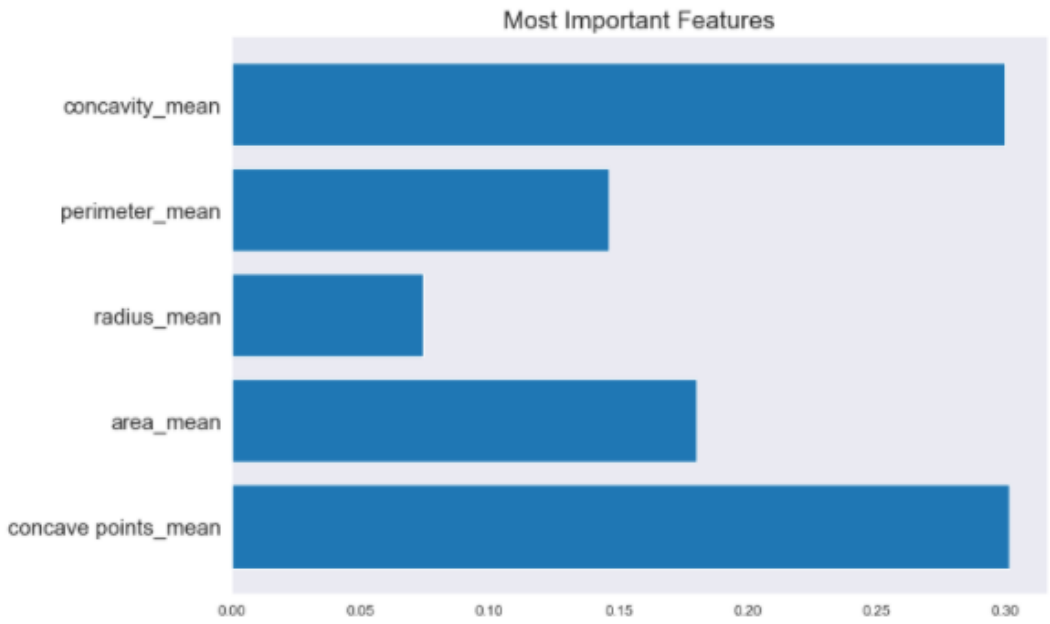
```
Number of patients diagnosed with liver disease: 416
Number of patients not diagnosed with liver disease: 167
```



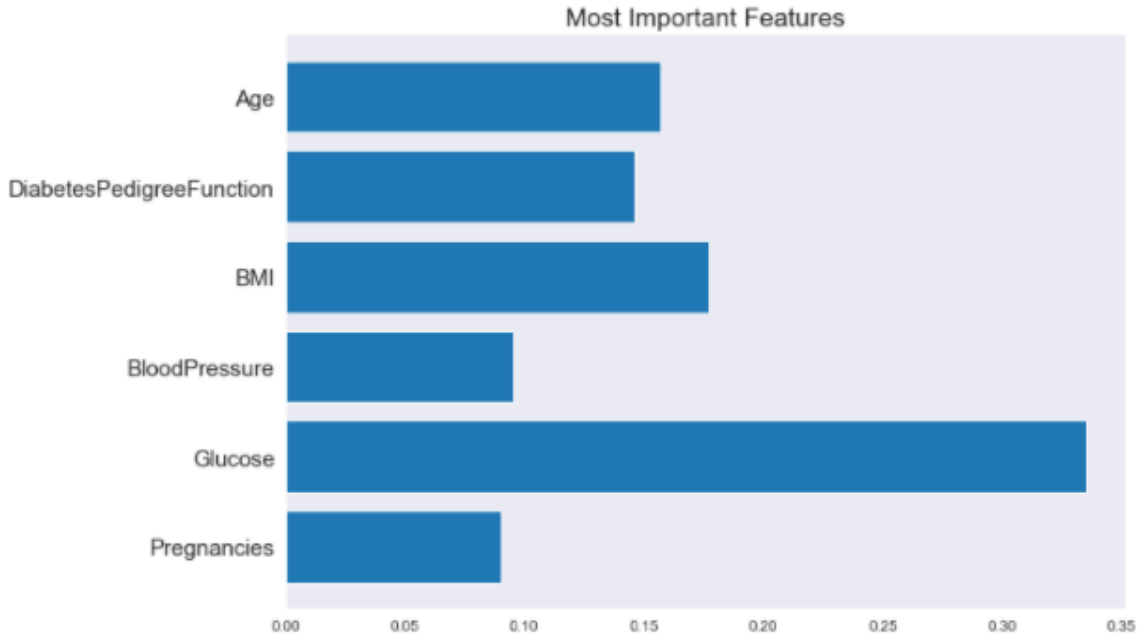
3.4.3 Feature Importance

After performing Exploratory Data Analysis and Visualization on these above datasets, we discover most important features for our different problems.

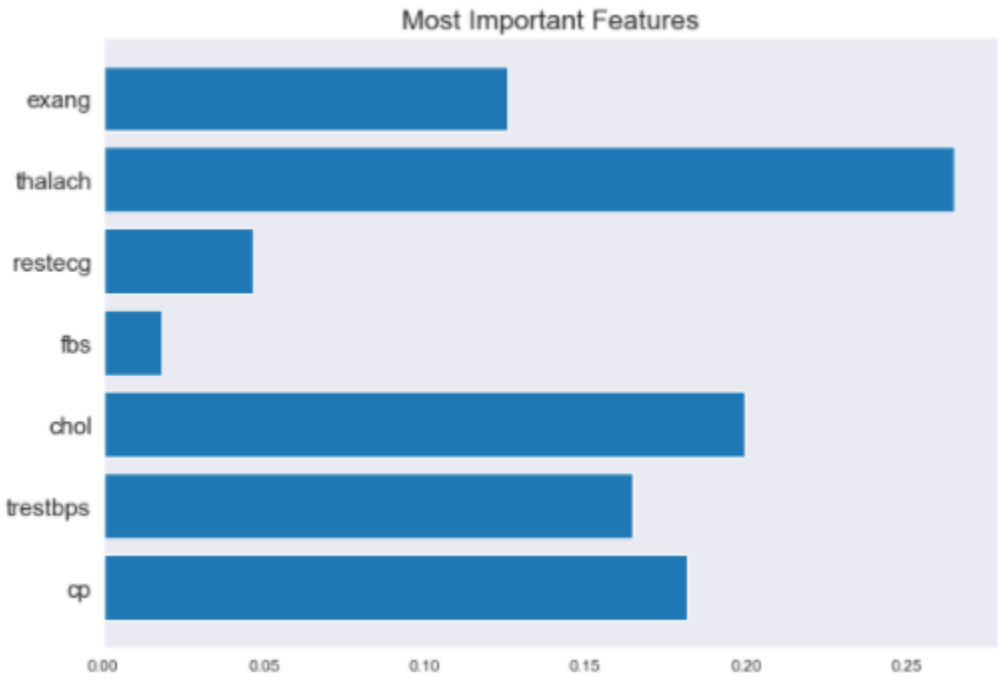
Breast Cancer



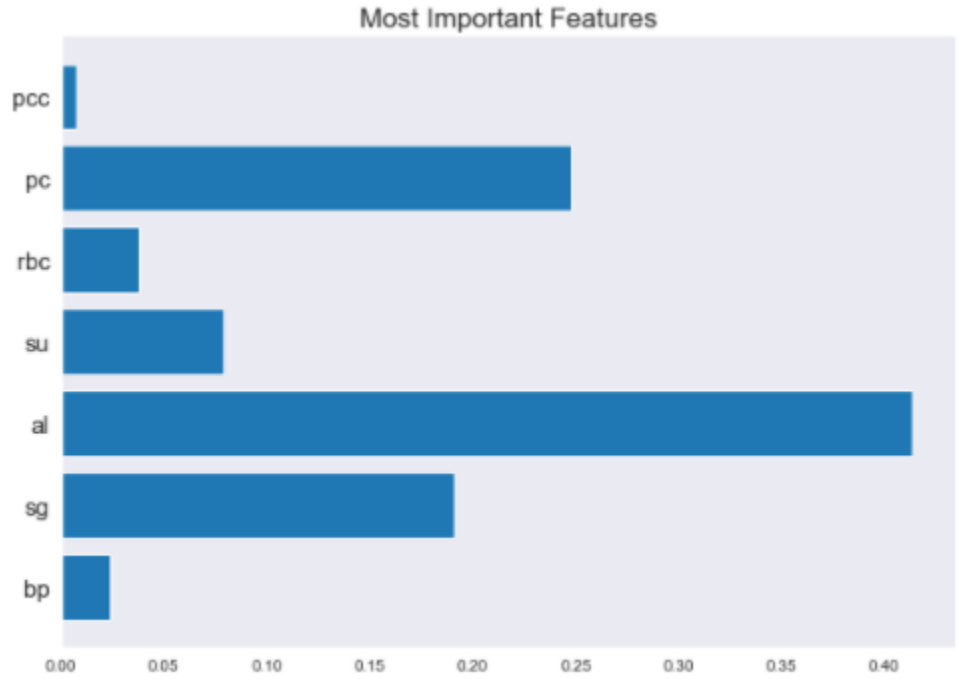
Diabetes



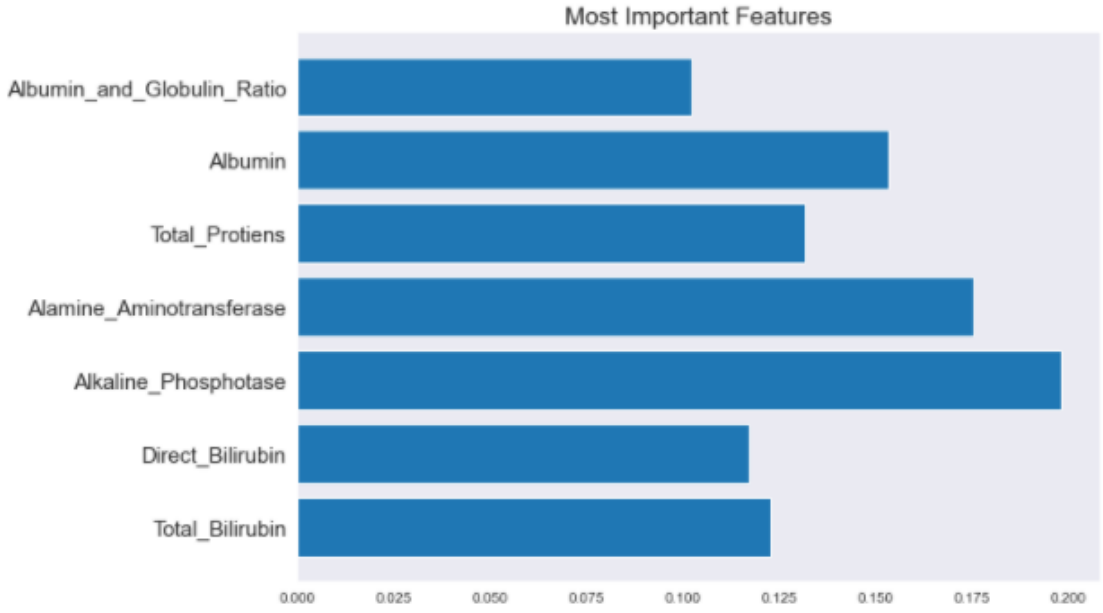
Heart Disease



Kidney Disease



Liver Disease



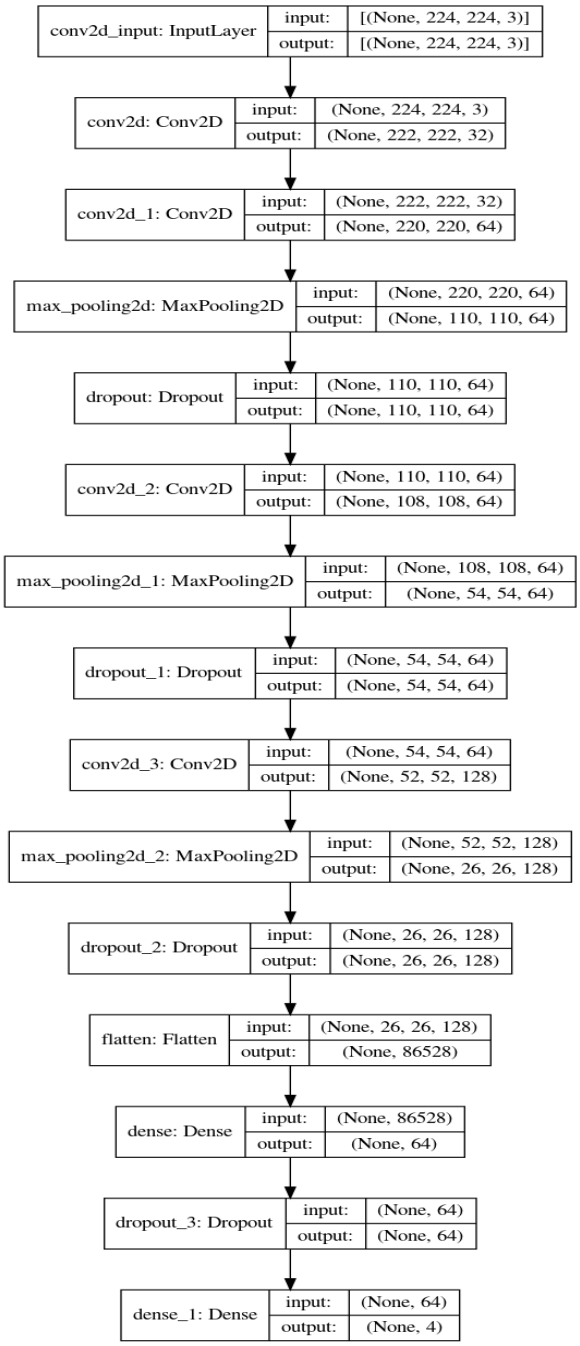
3.4.4 Model Preparation

```
training_accuracy = []

testing_accuracy = []

# Random Forest Classifier

model_rf = RandomForestClassifier(n_estimators=10)

model_rf.fit(X_train, y_train)

training_accuracy.append(model_rf.score(X_train,y_train))

testing_accuracy.append(model_rf.score(X_test, y_test))


# Logistic Regression

model_lr = LogisticRegression()

model_lr.fit(X_train, y_train)

training_accuracy.append(model_lr.score(X_train,y_train))
```

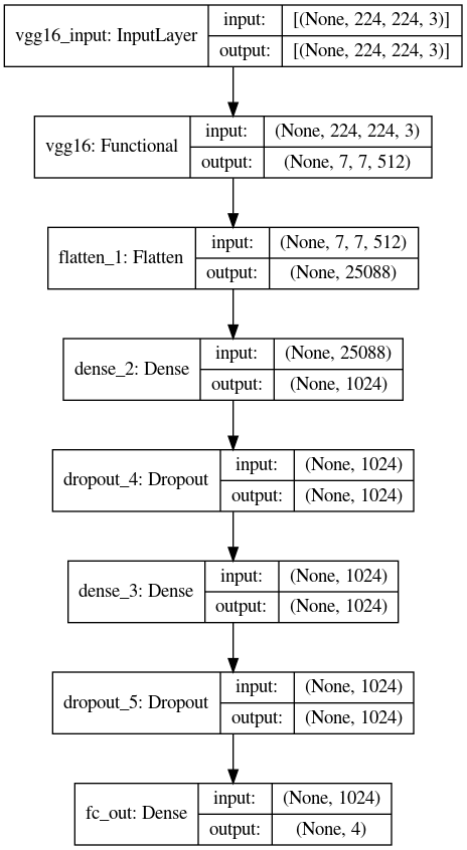
```
testing_accuracy.append(model_lr.score(X_test, y_test))


# Decision Tree Classifier
model_dt = DecisionTreeClassifier()
model_dt.fit(X_train, y_train)
training_accuracy.append(model_dt.score(X_train,y_train))
testing_accuracy.append(model_dt.score(X_test, y_test))


# K-Nearest Neighbor
model_knn = KNeighborsClassifier()
model_knn.fit(X_train, y_train)
training_accuracy.append(model_knn.score(X_train,y_train))
testing_accuracy.append(model_knn.score(X_test, y_test))


# Support Vector Machine
model_svc = SVC()
model_svc.fit(X_train, y_train)
training_accuracy.append(model_svc.score(X_train,y_train))
testing_accuracy.append(model_svc.score(X_test, y_test))
Algorithms = ['Random Forest', 'Logistic', 'Decision Tree', 'KNN', 'SVM']
```

Lungs Disease

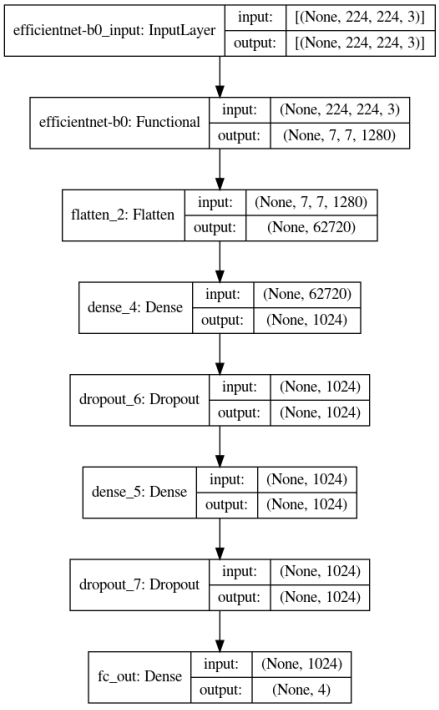
i. Simple Convolution Neural Network (CNN)



ii. Visual Geometry Group (VGG16) as Base



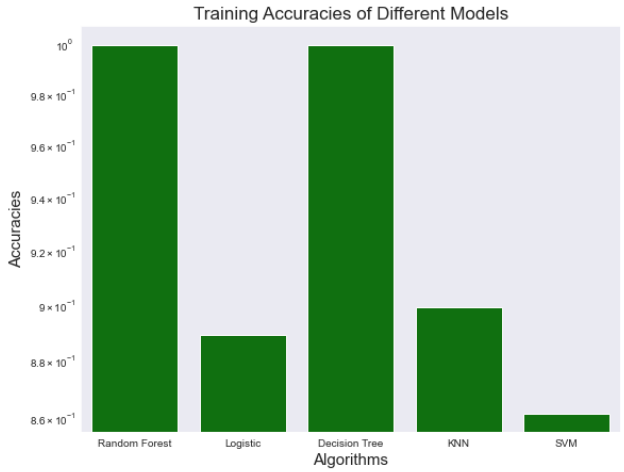
iii. EfficientNet as Base

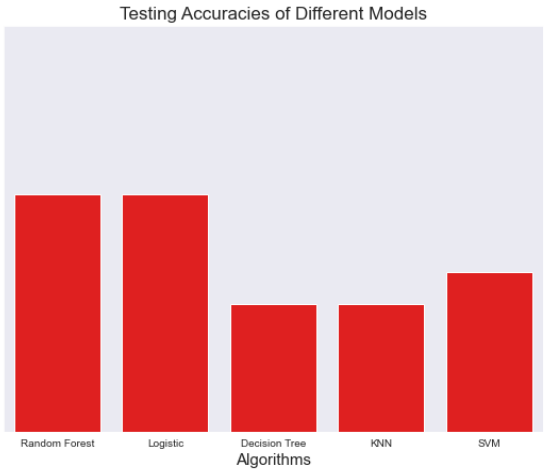


3.4.5 Model Testing

Breast Cancer

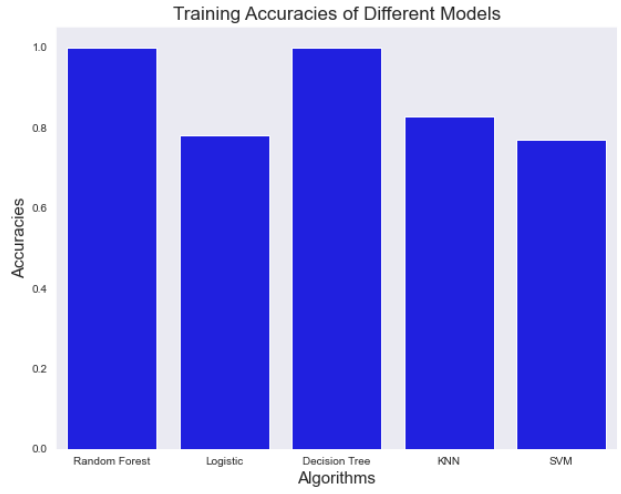
We used different machine learning algorithms for predicting the cancer. On applying the random forest along with other machine learning algorithm, the best result or the best accuracy among them was given by the random forest. Analyzing this data, we implemented the random forest to predict chances of getting cancer.

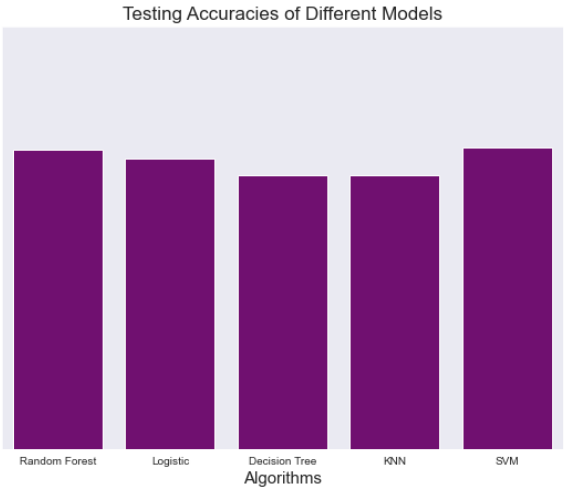




Diabetes

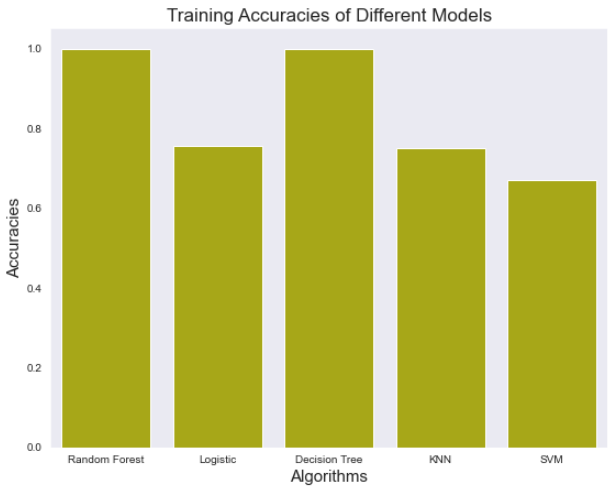
For diabetes, the dataset input obtained from the users were analyzed using machine learning algorithm. On analysis, it was found that the among the algorithm used, random forest gave the best accuracy on training the model. So for predicting the chances of getting diabetes, we used random forest algorithm.

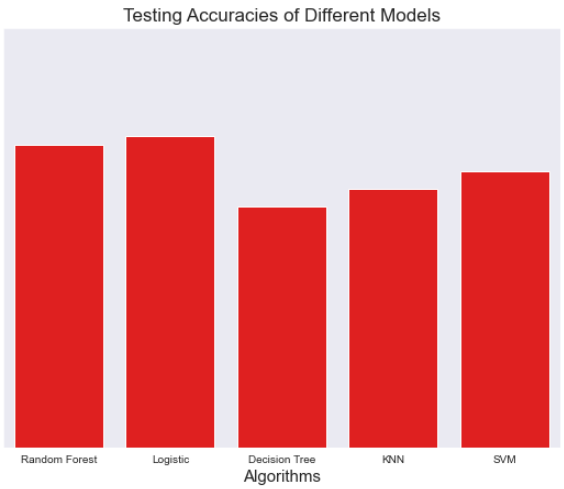




Heart Disease

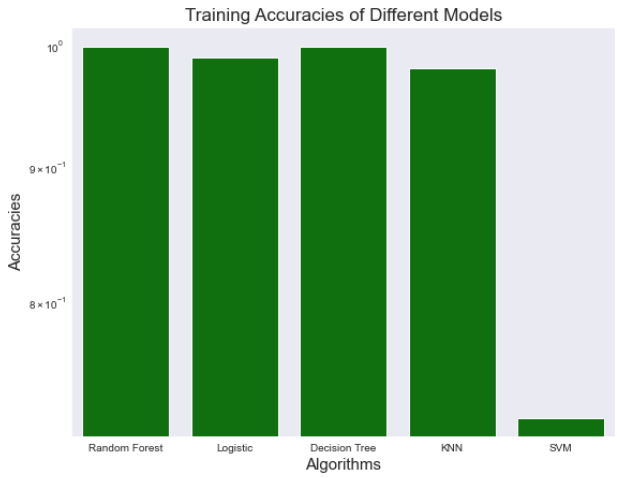
On applying the machine learning algorithm like KNN, SVC, etc and reviewing the testing and training accuracy we came to the conclusion that the best fit algorithm for predicting the chances of getting a heart attack is none other than random forest algorithm.

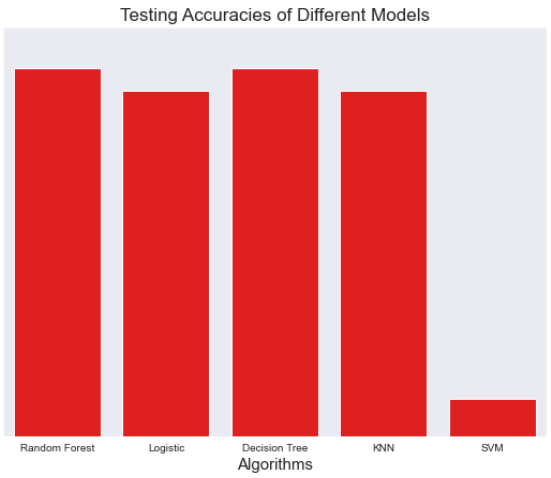




Kidney Disease

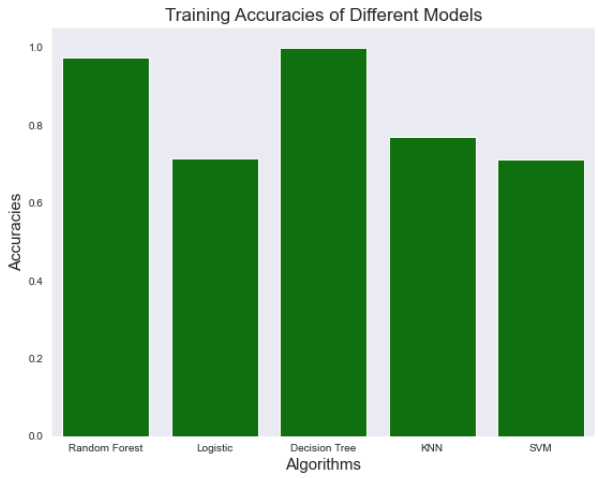
For predicting the kidney disease too, random forest algorithm gave the best accuracy while testing and training the model. So we decided to implement the kidney disease prediction using the same random forest algorithms as used in the disease above.

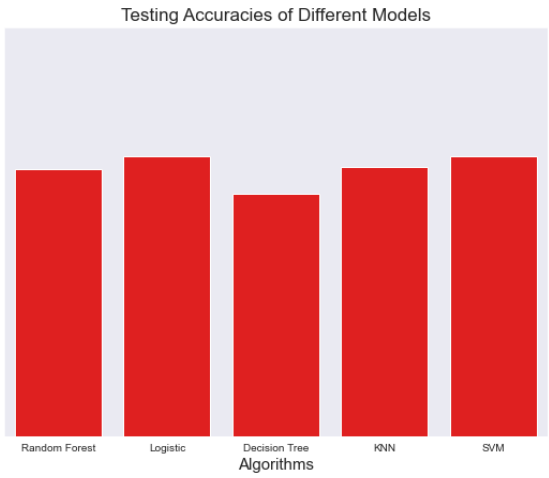




Liver Disease

For liver disease too, random forest gave the best accuracy on both testing and training. Hence, we decided to use the random forest on liver disease too, as the accuracy of other algorithms were low than the random forest algorithm.

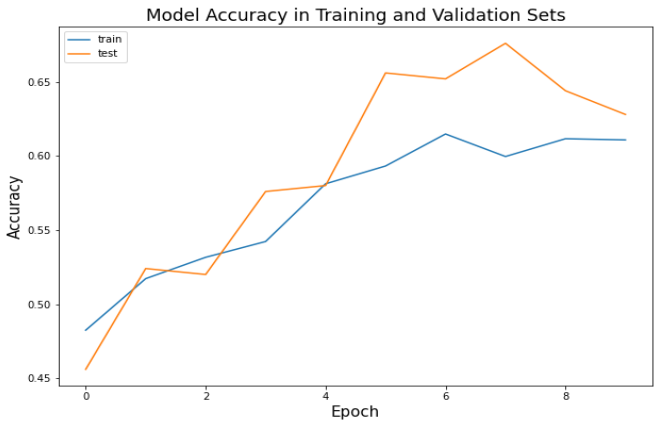




Lungs Disease

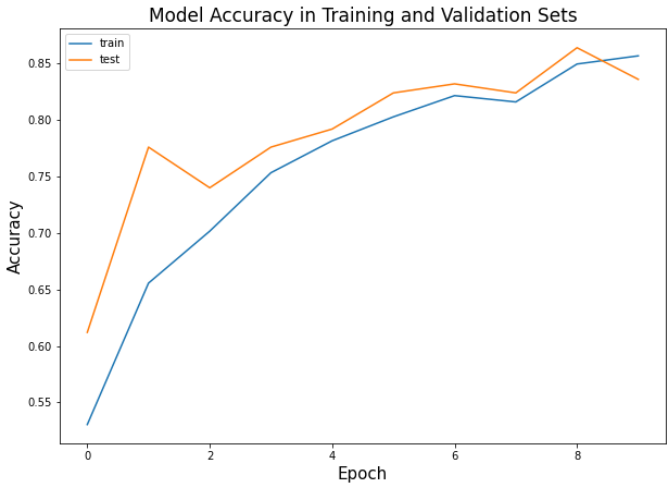
i. Simple CNN

In this model, we defined a convolutional layer. We limited this layer size due to computational complexity. This model is trained for ten epochs. The model progression with respect to training set and validation set is as shown below.



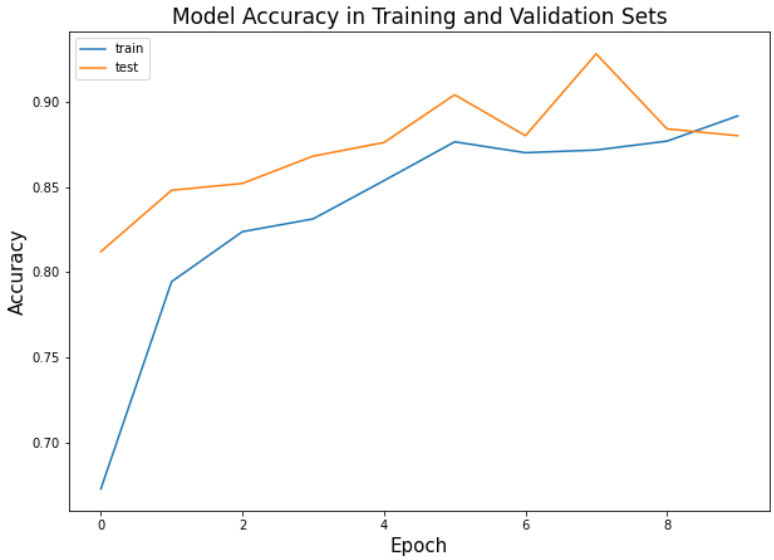
ii. VGG16 as Base Model

This model progression with respect to training set and validation set is as shown below:



iii. EfficientNet as Base Model

The performance of this model in different epochs with respect to training set and validation set is as shown below:



CHAPTER 4
RESULTS, CONCLUSION AND FUTURE ENHANCEMENT

4.1 Results

Our project Disease Detector was successfully designed using the result obtained from the various machine learning algorithm. The result of each individual disease given below:

a) Cancer

	Training_Accuracy	Testing_Accuracy
Random Forest	1.000000	0.941520
Logistic	0.889447	0.941520
Decision Tree	1.000000	0.900585
KNN	0.899497	0.900585
SVM	0.861809	0.912281

For detecting the cancer, the random forest algorithm was implemented on the data set. On training the model, 100% accuracy was achieved. Whereas, 94% accuracy was obtained while testing the model. Similarly, on applying the logistic regression on the same dataset, accuracy of 88% was achieved on training the model while 94% accuracy was achieved on testing the model. Then the decision tree was used which showed the accuracy of 100% on training the model while it gave 90% accuracy on testing the model.89% accuracy was gained while training while only 90% accuracy was gained while implementing the KNN algorithm. On applying SVM 86% and 91% accuracy was gained on training and testing respectively.

b) Diabetes

	Training_Accuracy	Testing_Accuracy
Random Forest	0.998273	0.744828
Logistic	0.780656	0.724138
Decision Tree	1.000000	0.682759
KNN	0.829016	0.682759
SVM	0.770294	0.751724

For detecting the diabetes, the random forest algorithm was implemented on the data set. On training the model, 99% accuracy was achieved. Whereas, 74% accuracy was obtained while testing the model. Similarly, on applying the logistic regression on the same dataset, accuracy of 78% was achieved on training the model while 72% accuracy

was achieved on testing the model. Then the decision tree was used which showed the accuracy of 100% on training the model while it gave 68% accuracy on testing the model.82% accuracy was gained while training while only 68% accuracy was gained while implementing the KNN algorithm. On applying SVM 77% and 75% accuracy was gained on training and testing respectively.

c) Heart Disease

	Training_Accuracy	Testing_Accuracy
Random Forest	1.000000	0.758242
Logistic	0.754717	0.780220
Decision Tree	1.000000	0.604396
KNN	0.750000	0.648352
SVM	0.669811	0.692308

For detecting the chances of getting heart attack, the random forest algorithm was implemented on the data set. On training the model, 100% accuracy was achieved. Whereas, 95% accuracy was obtained while testing the model. Similarly, on applying the logistic regression on the same dataset, accuracy of 75% was achieved on training the model while 78% accuracy was achieved on testing the model. Then the decision tree was used which showed the accuracy of 100% on training the model while it gave 60% accuracy on testing the model.75% accuracy was gained while training while only 64% accuracy was gained while implementing the KNN algorithm. On applying SVM 66% and 69% accuracy was gained on training and testing respectively.

d) Kidney disease

	Training_Accuracy	Testing_Accuracy
Random Forest	1.000000	0.981132
Logistic	0.990476	0.962264
Decision Tree	1.000000	0.981132
KNN	0.980952	0.962264
SVM	0.723810	0.735849

For detecting the kidney disease, the random forest algorithm was implemented on the data set. On training the model, 100% accuracy was achieved. Whereas, 98% accuracy was obtained while testing the model. Similarly, on applying the logistic regression on the same dataset, accuracy of 99% was achieved on training the model while 96% accuracy was achieved on testing the model. Then the decision tree was used which showed the accuracy of 100% on training the model while it gave 98% accuracy on testing the model.98% accuracy was gained while training while only 96% accuracy was gained while implementing the KNN algorithm. On applying SVM 72% and 73% accuracy was gained on training and testing respectively.

e) Liver

	Training_Accuracy	Testing_Accuracy
Random Forest	0.973039	0.685714
Logistic	0.715686	0.720000
Decision Tree	1.000000	0.622857
KNN	0.769608	0.691429
SVM	0.710784	0.720000

For detecting the cancer, the random forest algorithm was implemented on the data set. On training the model, 97% accuracy was achieved. Whereas, 68% accuracy was obtained while testing the model. Similarly, on applying the logistic regression on the same dataset, accuracy of 71% was achieved on training the model while 72% accuracy was achieved on testing the model. Then the decision tree was used which showed the accuracy of 100% on training the model while it gave 62% accuracy on testing the model.76% accuracy was gained while training while only 69% accuracy was gained while implementing the KNN algorithm. On applying SVM 71% and 72% accuracy was gained on training and testing respectively.

f) Lungs

- Simple CNN

```
Epoch 8/10
50/50 [==============================] - 11s 217ms/step - loss: 0.9382 - recall: 0.4092 - precision: 0.6890 - acc: 0.6057 - val
_loss: 1.0221 - val_recall: 0.1720 - val_precision: 0.8958 - val_acc: 0.6720
Epoch 9/10
50/50 [==============================] - 11s 221ms/step - loss: 0.9207 - recall: 0.4632 - precision: 0.7105 - acc: 0.6131 - val
_loss: 1.0629 - val_recall: 0.0920 - val_precision: 0.8519 - val_acc: 0.6360
Epoch 10/10
50/50 [==============================] - 11s 214ms/step - loss: 0.9159 - recall: 0.4602 - precision: 0.6959 - acc: 0.6164 - val
_loss: 1.0358 - val_recall: 0.1400 - val_precision: 0.9459 - val_acc: 0.6680
```

Due to computation complexity we are compelled to build a very simple convolutional neural network. When this model is trained and tested successively we got its respective training set accuracy and testing set accuracy to be 61.64 % and 66.8%.

- VGG 16 age base model

```
Epoch 8/10
50/50 [==============================] - 12s 242ms/step - loss: 0.4902 - recall_1: 0.7815 - precision_1: 0.8676 - acc: 0.8344 -
val_loss: 0.4123 - val_recall_1: 0.8240 - val_precision_1: 0.8957 - val_acc: 0.8600
Epoch 9/10
50/50 [==============================] - 12s 248ms/step - loss: 0.4386 - recall_1: 0.8059 - precision_1: 0.8804 - acc: 0.8479 -
val_loss: 0.3766 - val_recall_1: 0.8480 - val_precision_1: 0.8797 - val_acc: 0.8680
Epoch 10/10
50/50 [==============================] - 12s 240ms/step - loss: 0.4309 - recall_1: 0.8188 - precision_1: 0.8736 - acc: 0.8532 -
val_loss: 0.3676 - val_recall_1: 0.8400 - val_precision_1: 0.8714 - val_acc: 0.8600
```

We combined the VGG 16 with our fully connected layer and successively trained and tested the model. We got its respective training set accuracy and

validation set accuracy to be 85.32 % and 86.0%. In testing set the respective loss, recall, precision, accuracy, are 0.3811, 0.38462, 0.89, and 0.87.

- EfficientNet base model

```
Epoch 8/10
50/50 [==============================] - 13s 253ms/step - loss: 0.3662 - recall_2: 0.8376 - precision_2: 0.8648 - acc: 0.8534 -
val_loss: 0.3250 - val_recall_2: 0.8640 - val_precision_2: 0.8816 - val_acc: 0.8720
Epoch 9/10
50/50 [==============================] - 12s 249ms/step - loss: 0.3244 - recall_2: 0.8748 - precision_2: 0.8943 - acc: 0.8859 -
val_loss: 0.2857 - val_recall_2: 0.8800 - val_precision_2: 0.8907 - val_acc: 0.8880
Epoch 10/10
50/50 [==============================] - 13s 261ms/step - loss: 0.2953 - recall_2: 0.8785 - precision_2: 0.9033 - acc: 0.8928 -
val_loss: 0.2663 - val_recall_2: 0.8920 - val_precision_2: 0.9177 - val_acc: 0.9120
```

We combined the EfficientNet with our fully connected layer and successively trained and tested the model. We got its respective training set accuracy and validation set accuracy to be 89.28 % and 91.2%. %. In testing set the respective loss, recall, precision, accuracy, are 0.2604, 0.8957, 0.91, and 0.90.

4.2 Conclusion

According to report of John Hopkins Medicine, it was estimated that the number of patients suffering misdiagnosis-related, potentially preventable, significant permanent injury or death annually in the United States ranges from 80,000 to 160,000. Doctors or any other medical personnel after analyzing medical test results verified whether a diseased has been incurred or not. Humans are prone to error, so are the doctors or lab expert. Since the covid pandemic, the interactions between the patient and doctor are very minimal. This condition even worsens the mis-diagnosis scenarios. In such situation, creating a system which can assist and predict the possibility of getting the disease seem necessity. It would be much better if there some assistance in guiding the better drugs according to specific medical condition and we'd tried working on it. So, we successfully developed a model using various machine learning algorithms which can predict the possibility of the diseases like corona, breast cancer, diabetes, heart disease, liver disease, lungs diseases, and kidney diseases.

4.3 Future Enhancements

We are planning to add these features to our LMS:

- Improve the accuracy of the models
- Collecting more data for different models to support the first statement
- Widening the scope of this system to various other diseases
- We've developed models which are doing well in both the training and the testing data. But we do not know how these models actually perform in real world situations. Thus, testing this model is next important step.

REFERENCES

[1] "Decision tree," 2021. [Online]. Available: https://en.wikipedia.org/wiki/Decision_tree.

[2] "Machine Learning Techniques," 2020. [Online]. Available: https://www.sciencedirect.com/topics/computer-science/logistic-regression#:~:text=Logistic%20regression%20is%20a%20process,%2Fno%2C%20and%20so%20on..

[3] "Understanding Support Vector Machine(SVM)," 2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/.

[4] "A Complete Guide to the Random Forest Algorithm," 2019. [Online]. Available: https://builtin.com/data-science/random-forest-algorithm.