Smart Bridge Internship

Health Monitoring using Machine Learning

Project Report

Team Members:

- Panchami Dilip Nayak
- Rishabh Hegde
- Saiprasad Rao
- Sukshith

CONTENTS

1. Introduction

1.1 Overview

Health monitoring systems continue to evolve in this digital age. The benefit of utilizing these calculations over manual location is that they can bolster human specialists to analyse wellbeing condition, perform examination of complex information in a more productive way. A large number of machine learning algorithms are available that can perform pattern recognition, classification and regression. Most algorithms are support vector machine (SVM), logistic regression, decision tree, random forest, k-NN etc. These algorithms are used in different domains applications and each domain uses different performances matrices. In this project, k-NN algorithms are applied on four parameters i.e. SBP, DSP, Pulse rate and Temperature to classify the patient is normal or abnormal.

1.2 Purpose

The purpose of this project is to provide alternative to the traditional management of patients and reducing the expenditure of patients by providing accurate health status based on certain parameters which can be entered by the user.A user can check his/her health status sitting at home and this reduces the burden of going to clinic to get a checkup.The project also includes a BMI calculator which calculates the BMI and determines whether the user is under weight/over weight/normal.This is a robust and flexible project which tries to consider all aspects of human healtha and the main idea of providing health monitoring for the patients is data integrity.

2. Literature Survey

2.1 Existing problem

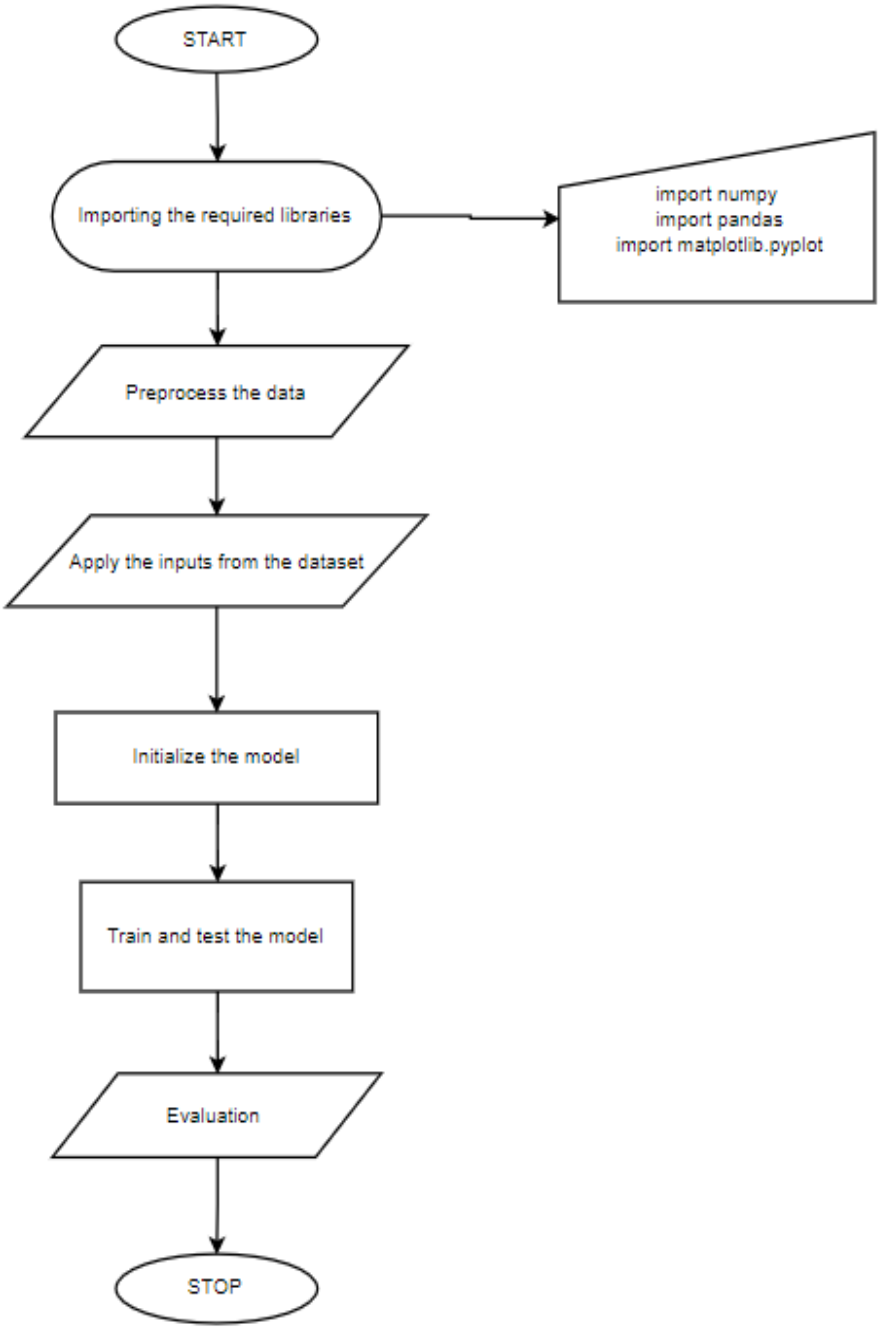
Better health is central to human happiness and well-being.Many factors influence health status and a country's ability to provide quality health services for its people. In order to maintain and lead a good life, it is essential that we take care of our health. At least half the world’s population is unable to access essential health services and many others are forced into extreme poverty by having to pay for healthcare they cannot afford. Also, doctors need to address many patients and patients are required to schedule appointments with a doctor, sometimes this is not very successful. A time/cost efficient and accurate alternative to predict the health status of a person is required to address this kind of problem.

2.2 Proposed Solution

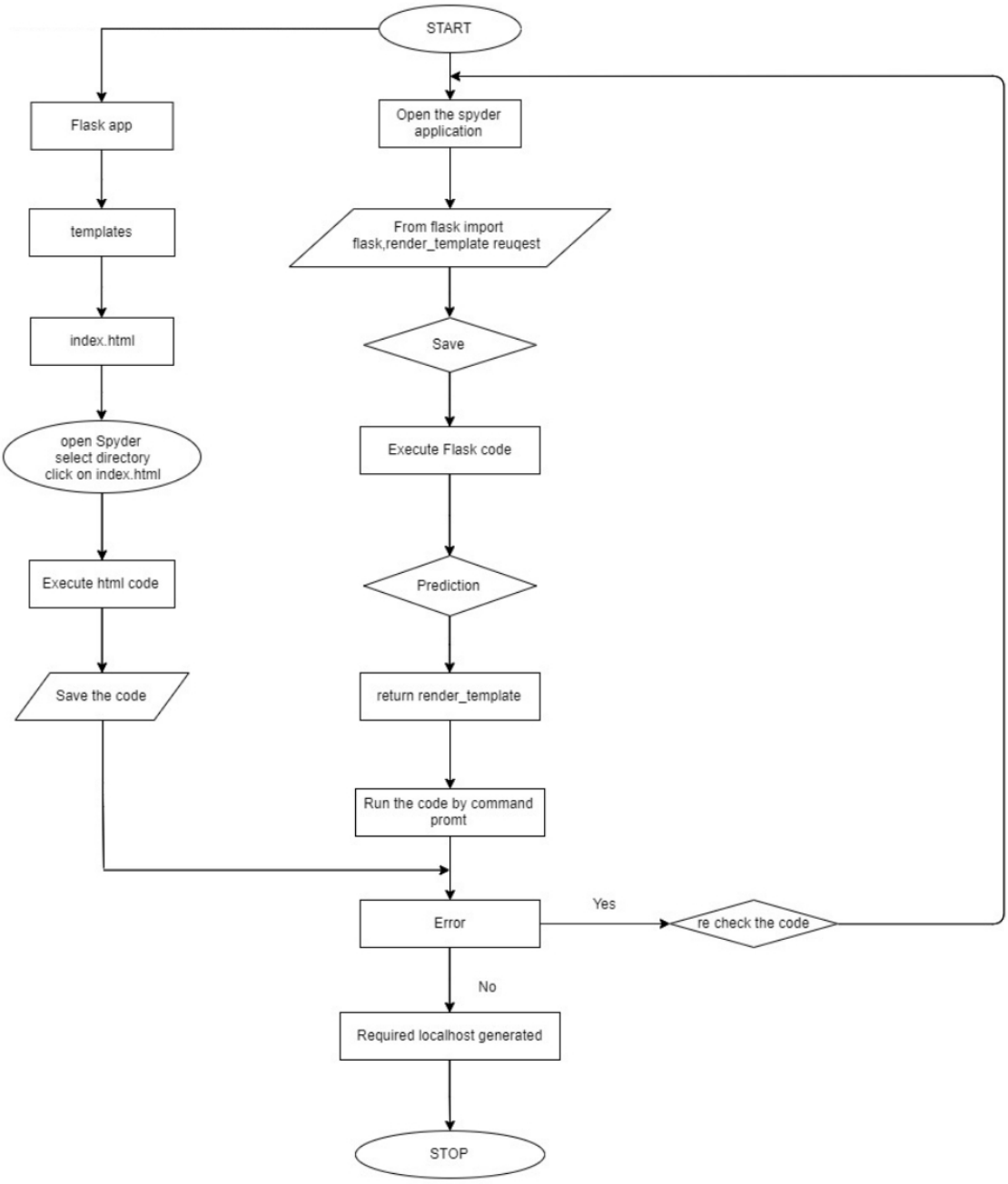
We propose to make a web application to predict the health status of a patient. The solution is majorly aimed at addressing the issues mentioned in Section 2.1.and making the process automated and faster. The proposed solution makes use of KNN algorithm to predict the health status of the patient on the basis of levels(0,1,2), where in 0 indicates that the person is healthy, 1 indicates that he/she is not completely healthy and needs to monitor his/her symptoms if any, and 2 indicates that the person is sick and needs immediate medical attention.

3. Theoretical Analysis

3.1 Block Diagram



Block Diagram 1



Block diagram 2

3.2 Hardware/Software Design

Design flow:

1. Data Collection and preprocessing
    - Collection of dataset.
    - Removing null values, if present.
    - Checking for the outliers.

2. Model Building
    - Train the model with dataset.
    - Test the model.

3. Model Deployment
    - Making webpages using Flask.
    - Creating static HTML templates.
    - Adding necessary CSS.

Requirements:

Software:

- Python 3
- numpy
- pandas
- matplotlib.pyplot
- KNeighborsClassifier
- Spyder
- Jupyter Notebook

Hardware:

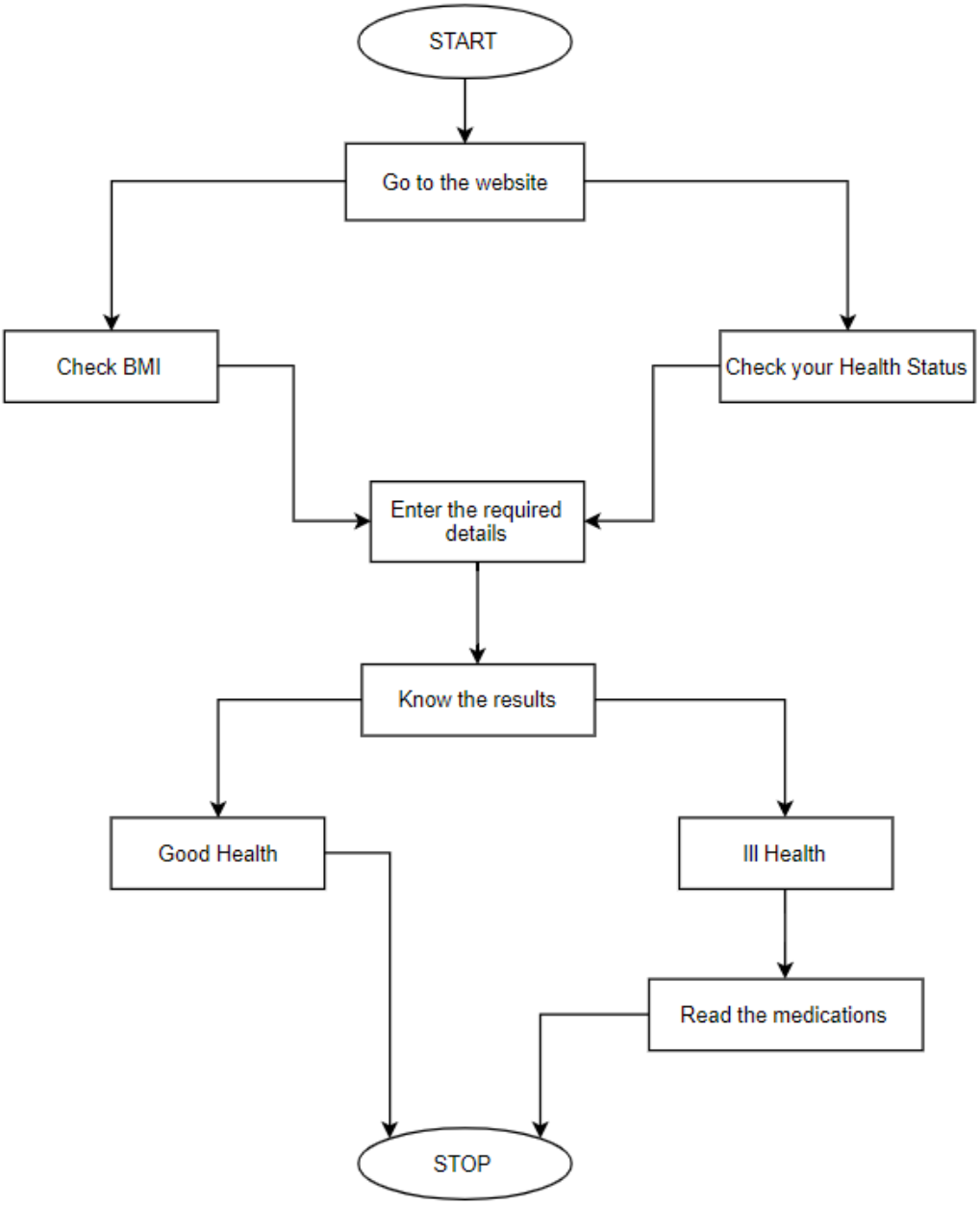
- Laptop/Desktop

4. Experimental Investigation

Throughout the development of the project there were a lot of learnings, following were our findings.

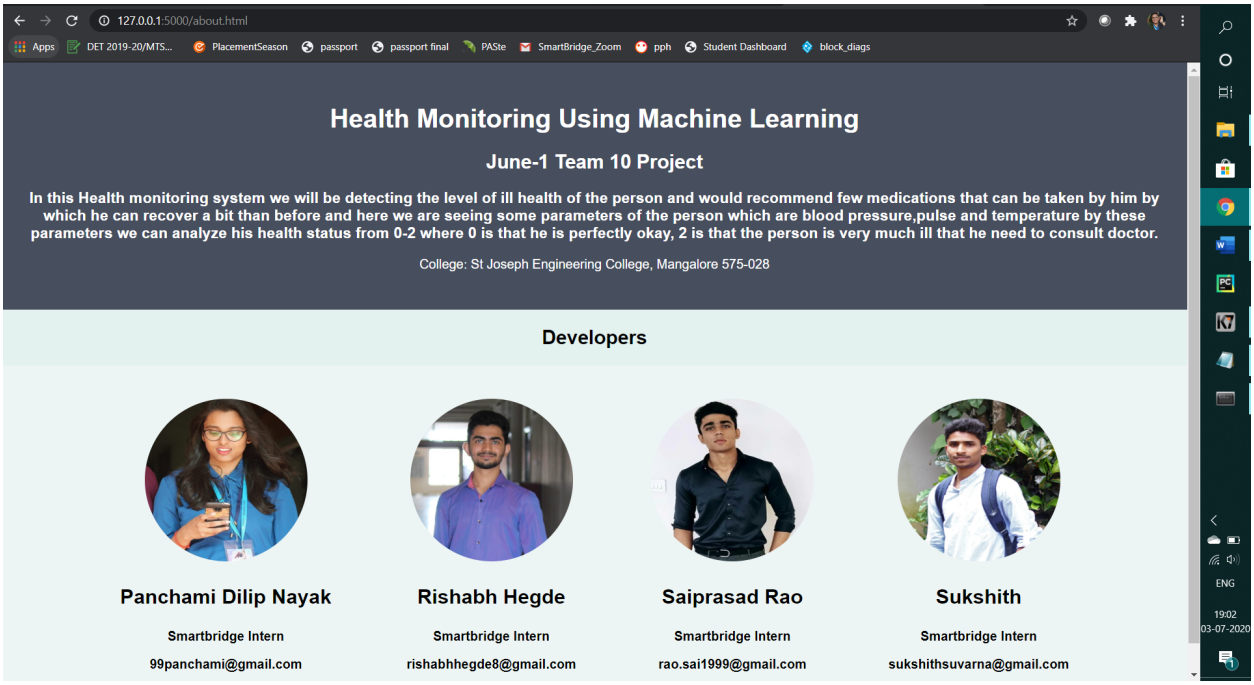
- Large datsets are required for such applications.
- Hence if such datasets would be given, the model would give better accuracy.

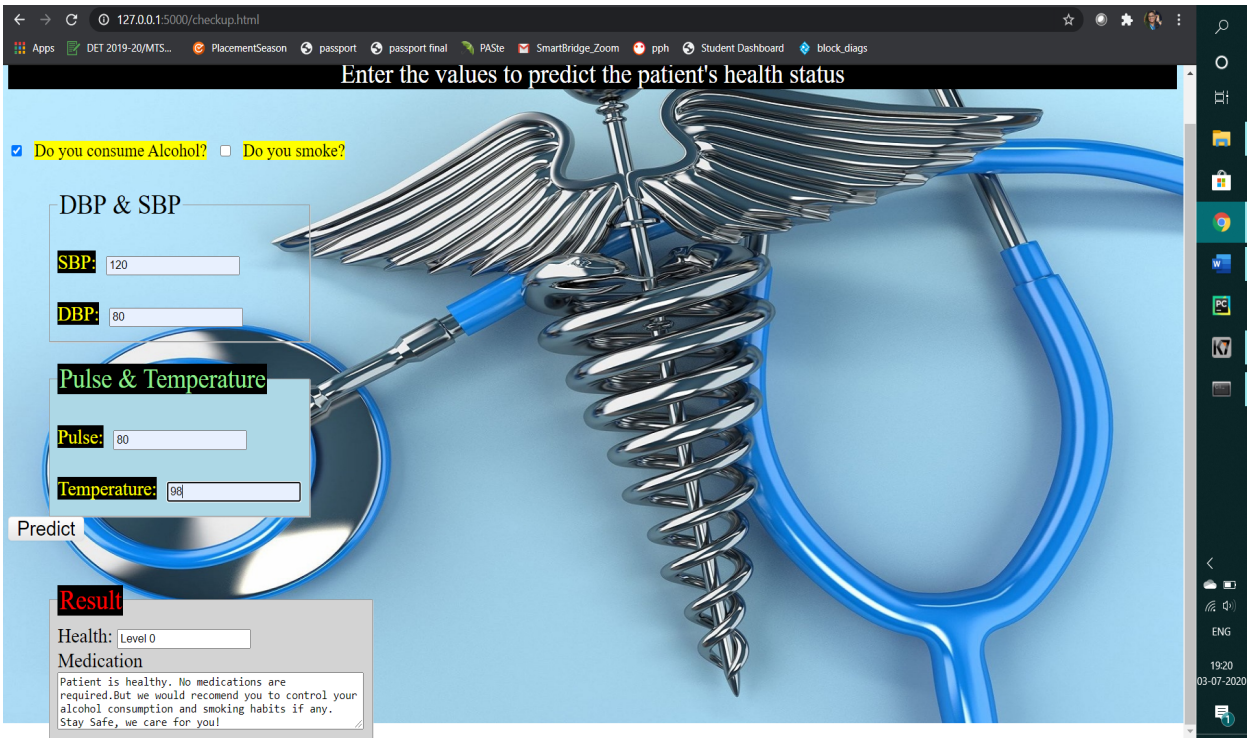
5. Flowchart

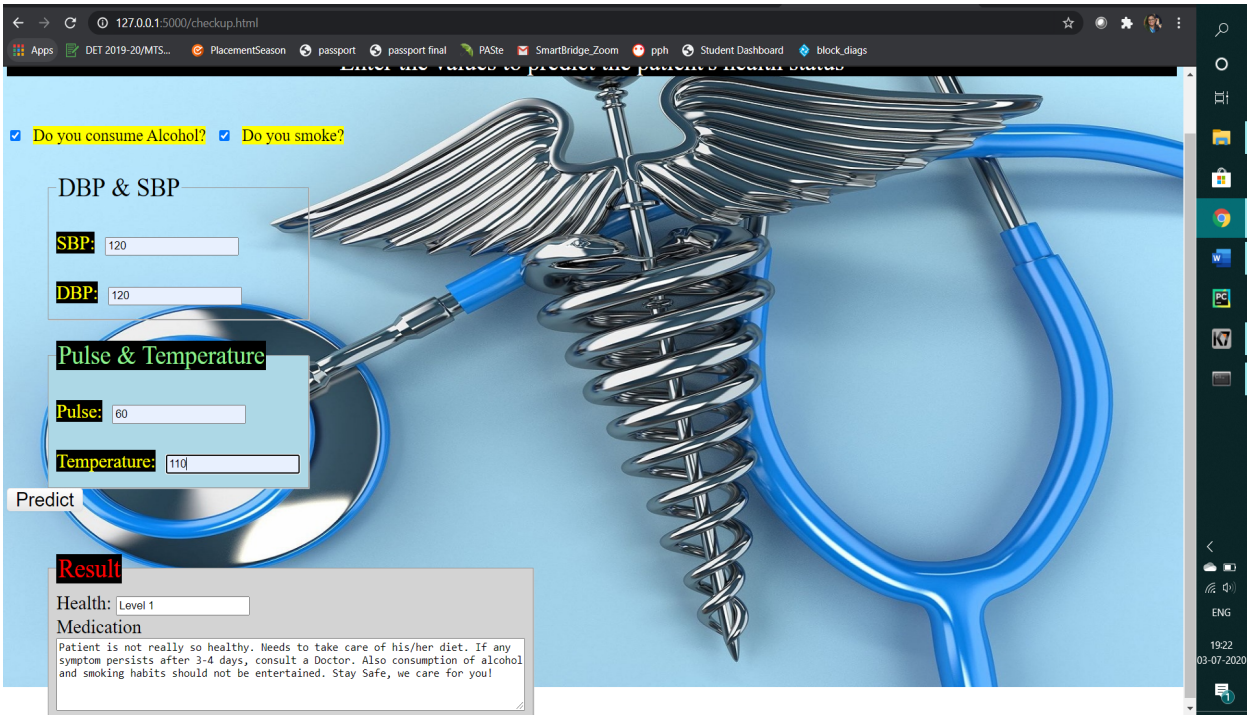


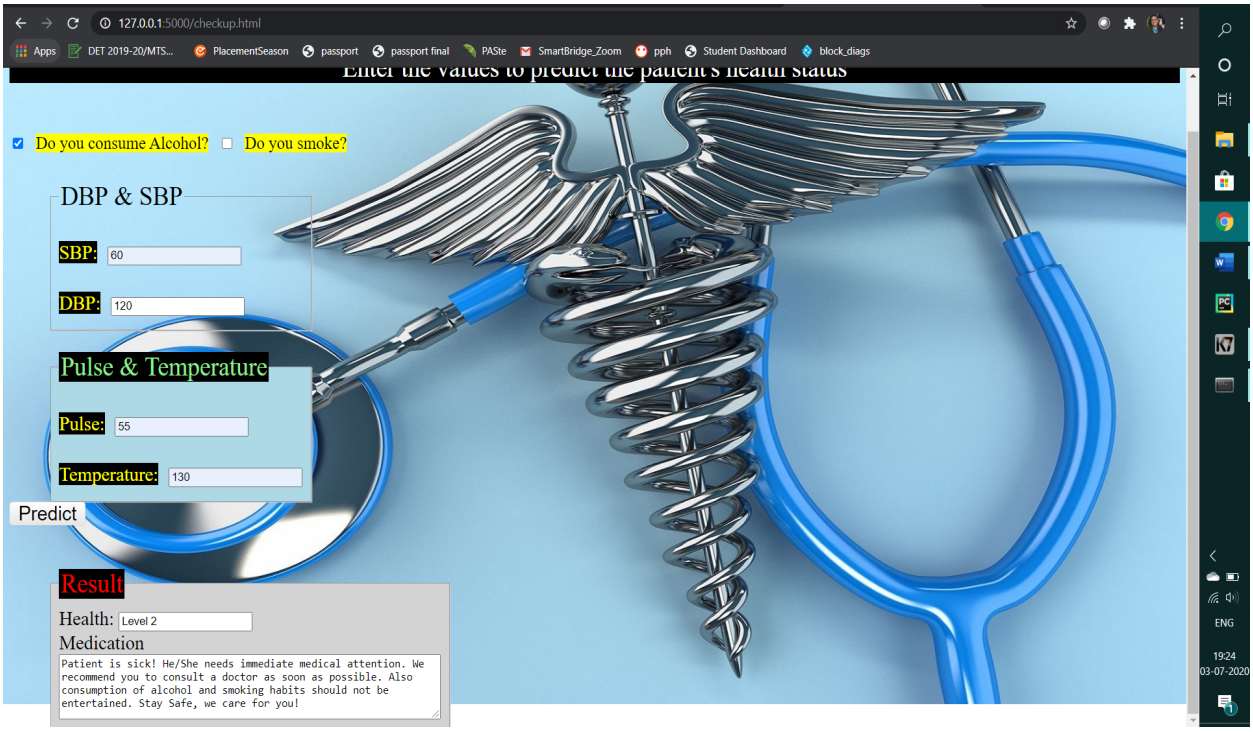
6. Results

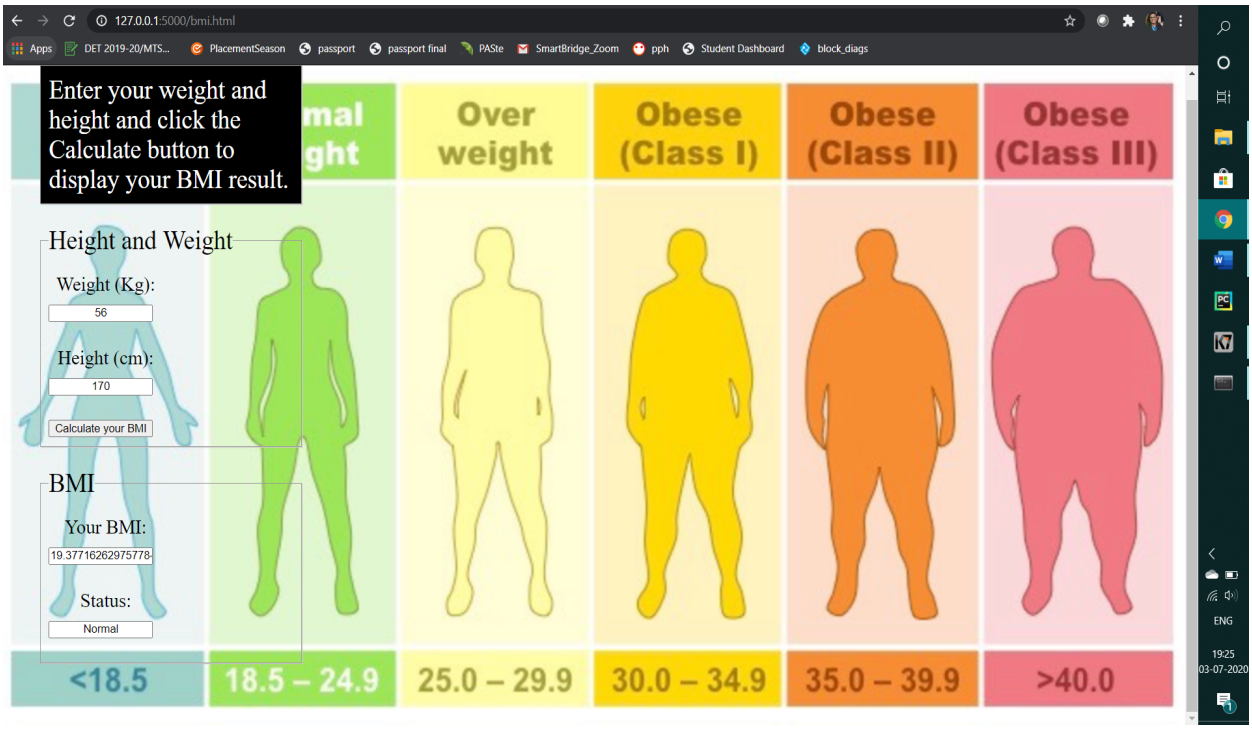
Successfully developed a web application that can predict the level of ill health of a person and would recommend few medications that can be taken by him by which he can recover a bit then before. Following are the scrrenshots:

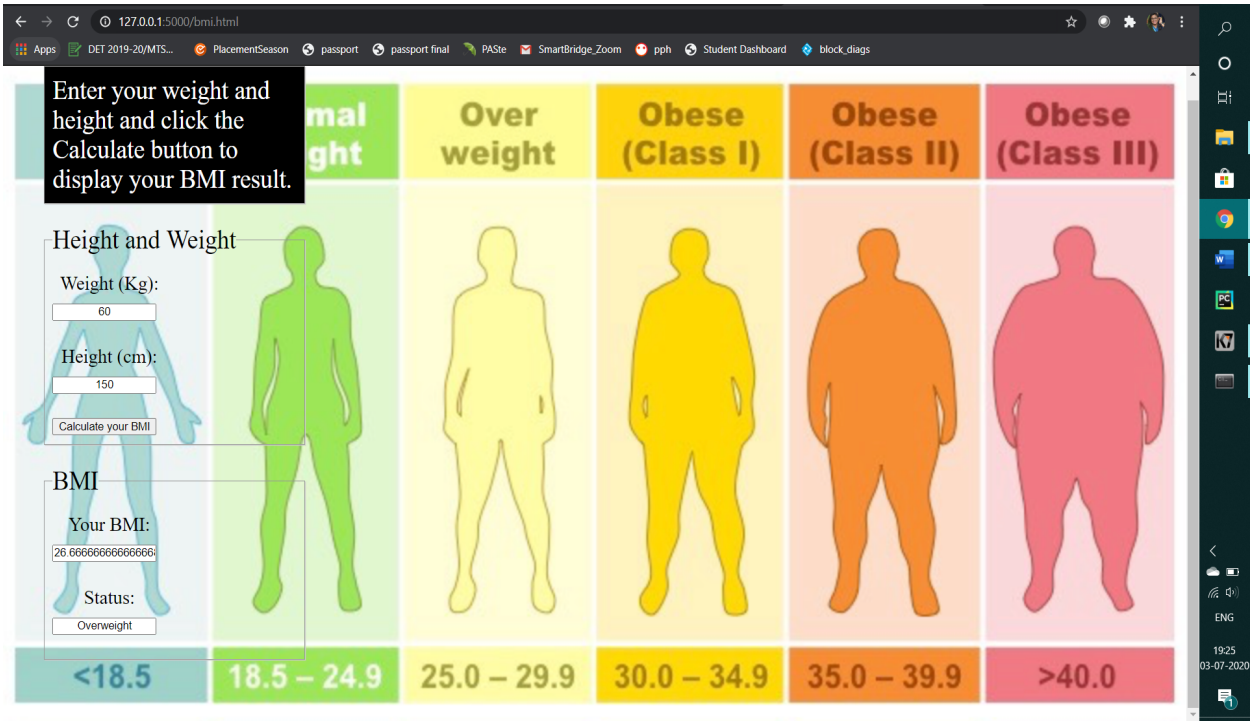


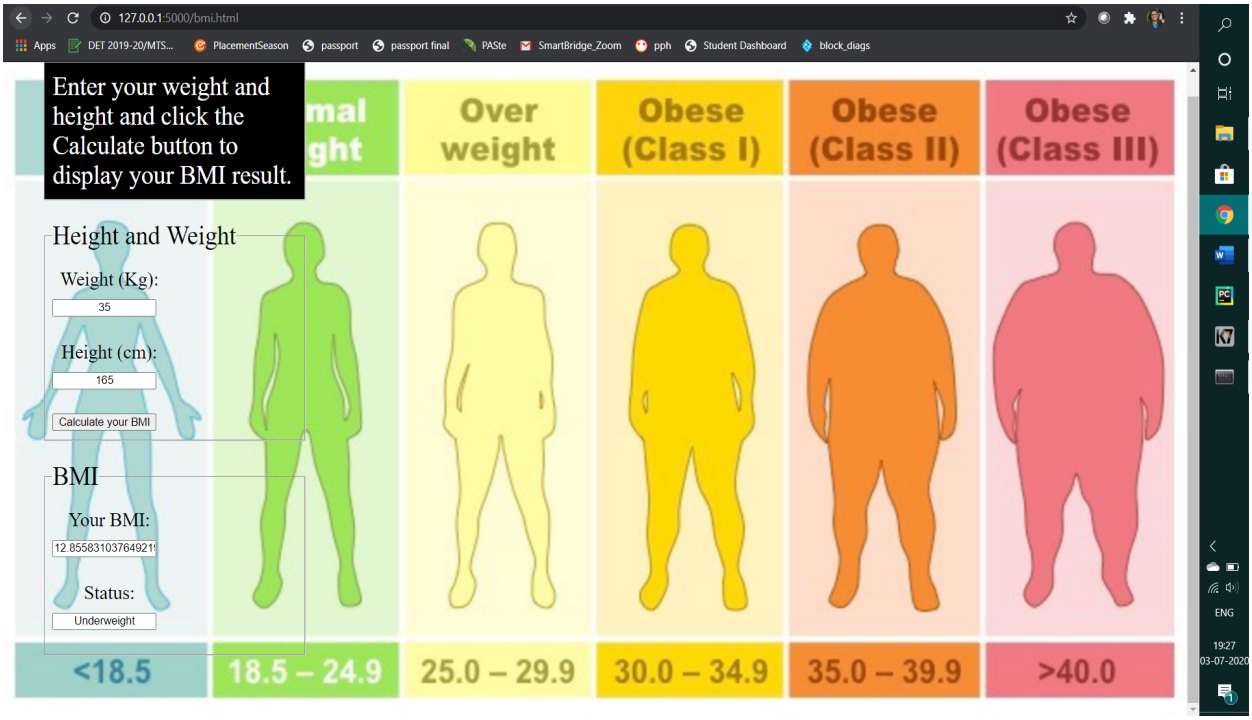

127.0.0.1:5000/checkup.html
Enter the values to predict the patient's health status
Do you consume Alcohol?
Do you smoke?
DBP & SBP
SBP: 120
DBP: 80
Pulse & Temperature
Pulse: 80
Temperature: 98
Predict
Result
Health: Level 0
Medication
Patient is healthy. No medications are required.But we would recomend you to control your alcohol consumption and smoking habits if any. Stay Safe, we care for you!


127.0.0.1:5000/checkup.html
Do you consume Alcohol?
Do you smoke?
DBP & SBP
SBP: 120
DBP: 120
Pulse & Temperature
Pulse: 60
Temperature: 110
Predict
Result
Health: Level 1
Medication
Patient is not really so healthy. Needs to take care of his/her diet. If any symptom persists after 3-4 days, consult a Doctor. Also consumption of alcohol and smoking habits should not be entertained. Stay Safe, we care for you!


127.0.0.1:5000/checkup.html
Enter the values to predict the patient's health status
Do you consume Alcohol?
Do you smoke?
DBP & SBP
SBP: 60
DBP: 120
Pulse & Temperature
Pulse: 55
Temperature: 130
Predict
Result
Health: Level 2
Medication
Patient is sick! He/She needs immediate medical attention. We recommend you to consult a doctor as soon as possible. Also consumption of alcohol and smoking habits should not be entertained. Stay Safe, we care for you!


127.0.0.1:5000/bmi.html
Enter your weight and height and click the Calculate button to display your BMI result.
Height and Weight
Weight (Kg):
56
Height (cm):
170
Calculate your BMI
BMI
Your BMI:
19.37716262975778
Status:
Normal
Over weight
Obese (Class I)
Obese (Class II)
Obese (Class III)
<18.5
18.5 – 24.9
25.0 – 29.9
30.0 – 34.9
35.0 – 39.9
>40.0


127.0.0.1:5000/bmi.html
Enter your weight and height and click the Calculate button to display your BMI result.
Height and Weight
Weight (Kg):
60
Height (cm):
150
Calculate your BMI
BMI
Your BMI:
26.666666666666
Status:
Overweight
Over weight
Obese (Class I)
Obese (Class II)
Obese (Class III)
<18.5
18.5 – 24.9
25.0 – 29.9
30.0 – 34.9
35.0 – 39.9
>40.0


127.0.0.1:5000/bmi.html
Enter your weight and height and click the Calculate button to display your BMI result.
Height and Weight
Weight (Kg):
35
Height (cm):
165
Calculate your BMI
BMI
Your BMI:
12.855831037649215
Status:
Underweight
Over weight
Obese (Class I)
Obese (Class II)
Obese (Class III)
<18.5
18.5 – 24.9
25.0 – 29.9
30.0 – 34.9
35.0 – 39.9
>40.0

7. Advantages and Disadvantages

Advantages	Disadvantages
Quick medical illness detection	Require a large dataset for good accuracy
Can be treated as a second opinion	

8. Applications

The appliication can be used by either patient or doctore to get a second opinion. If the train dataset given to the model is acurate then the accuracy of the model is increased and this application can be used as the main determnant to check the illness of a person.

9. Conclusions

The day is not far away when much of the health-related encounters will be conducted “virtually” using Machine Learning and Artificial Intelligence technology, ultimately culminating in a situation where this mode is one of the choice rather than of exception. Machine Learning is an exciting technology and continues to hold the promise of being truly transformational in terms of everyone’s benefit. As the population continues to grow at a rate faster than the concomitant growth in the number of available qualified clinicians and facilities (institutional beds, investigation laboratories, day-care centres, etc.), this technology will need to be optimally harnessed to ensure that all those who need care are catered to at least at acceptable levels, if not the best possible.

10. Future Scope

- Train and test the model with various algoriths to get better accuracy.
- Train the model with increased dataset.
- Build a mobile application, rather than web application to make it user friendly.

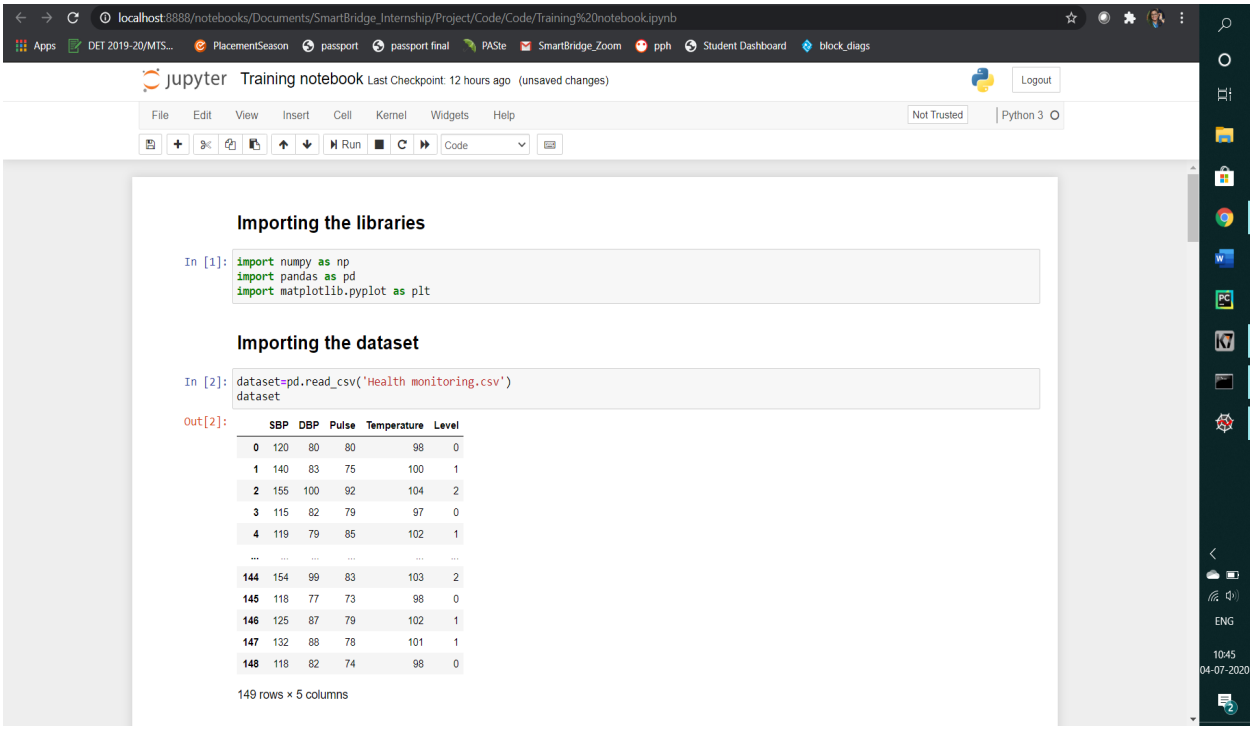
11. Bibiliography

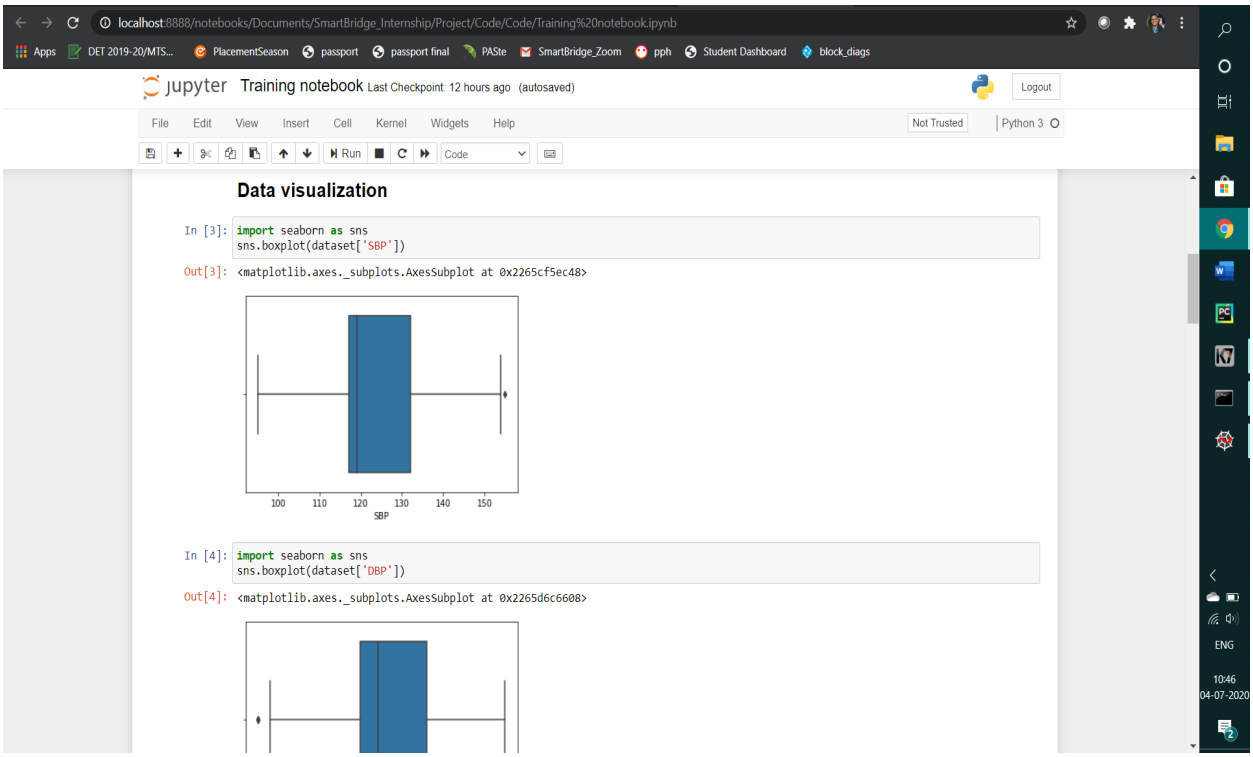
1. https://www.ijariit.com/manuscripts/v5i2/V5I2-1243.pdf
2. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8911389

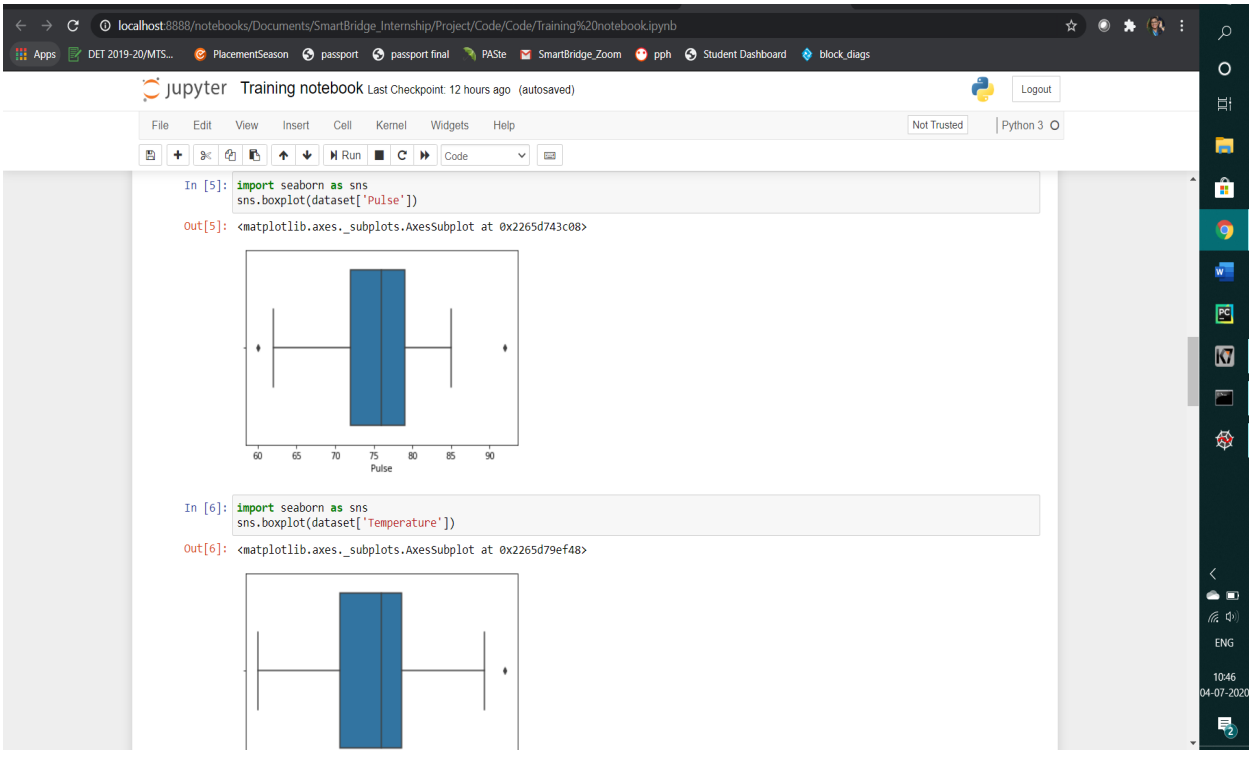
Appendix

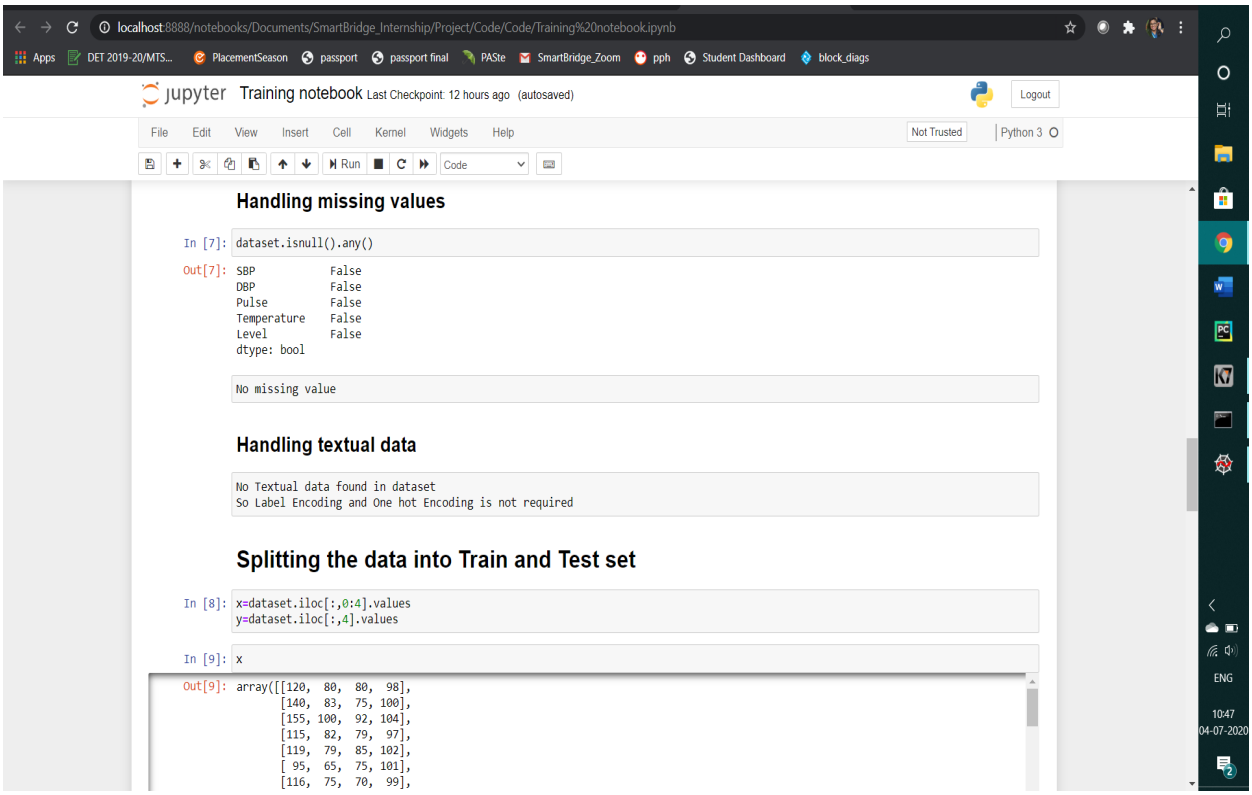
Model building:

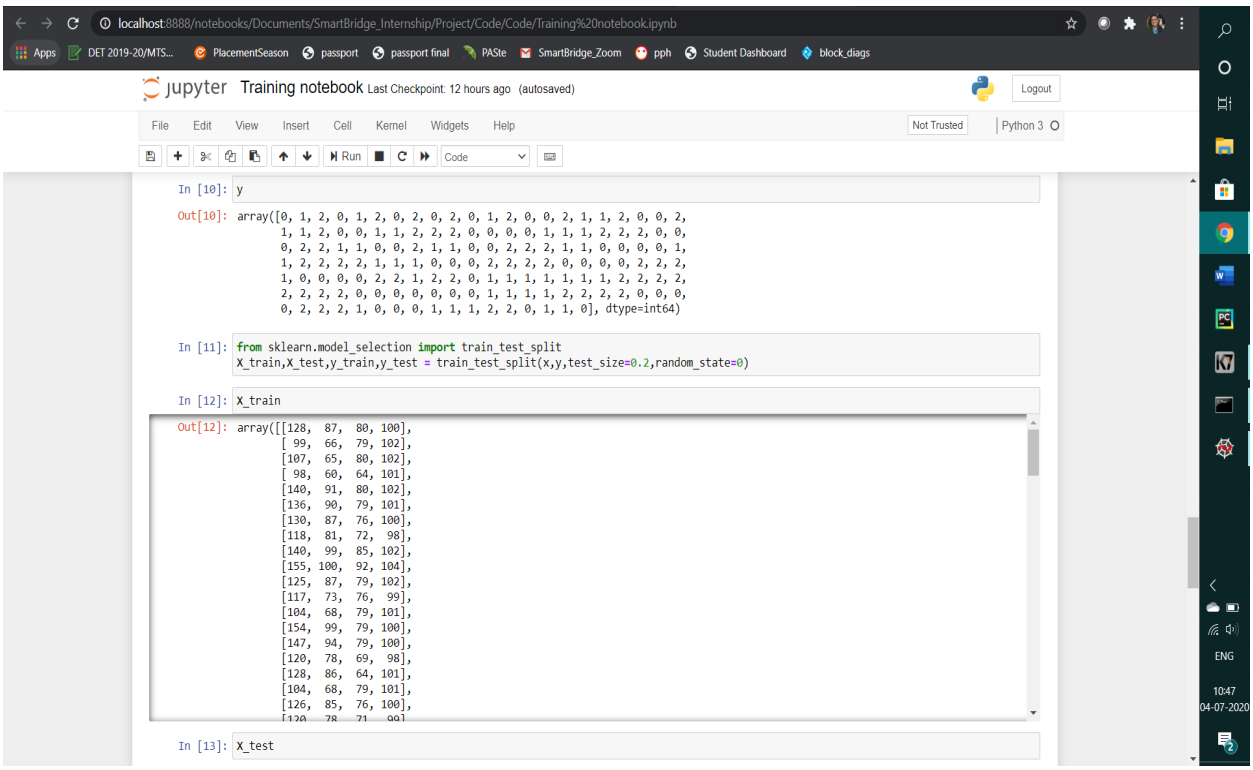
Training the model:

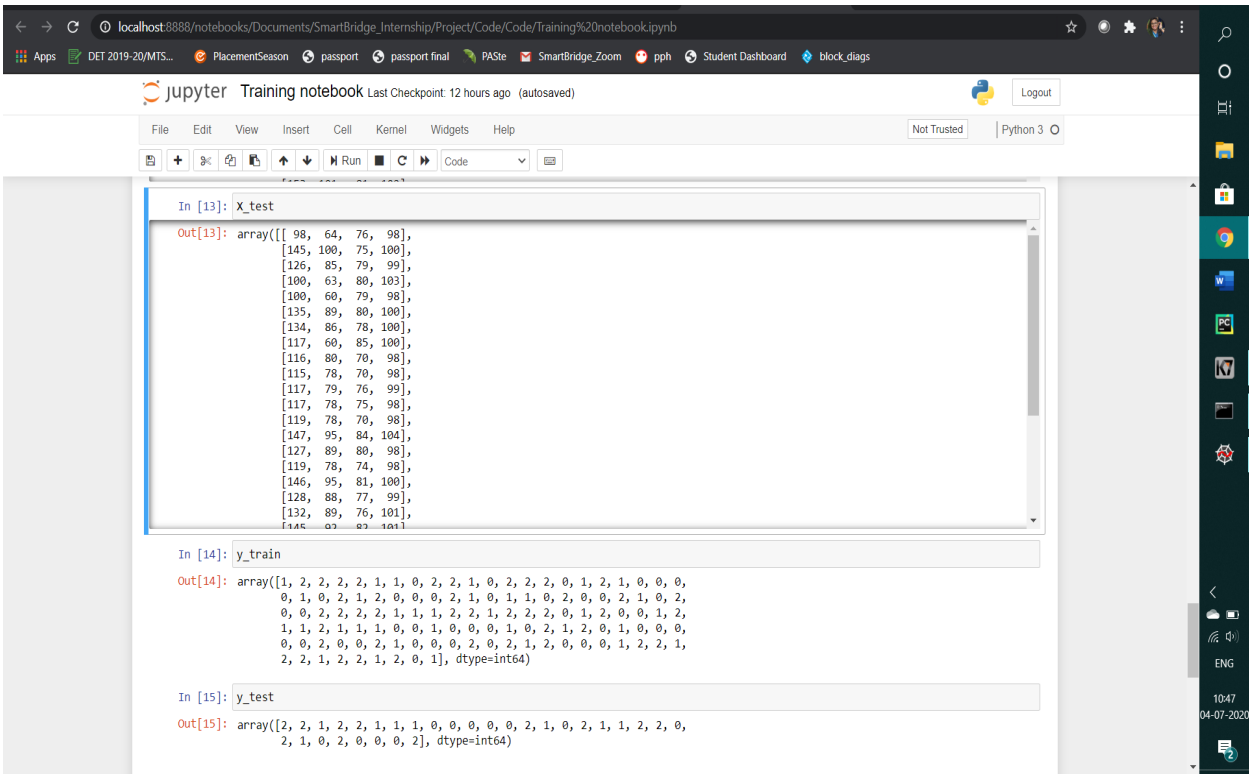


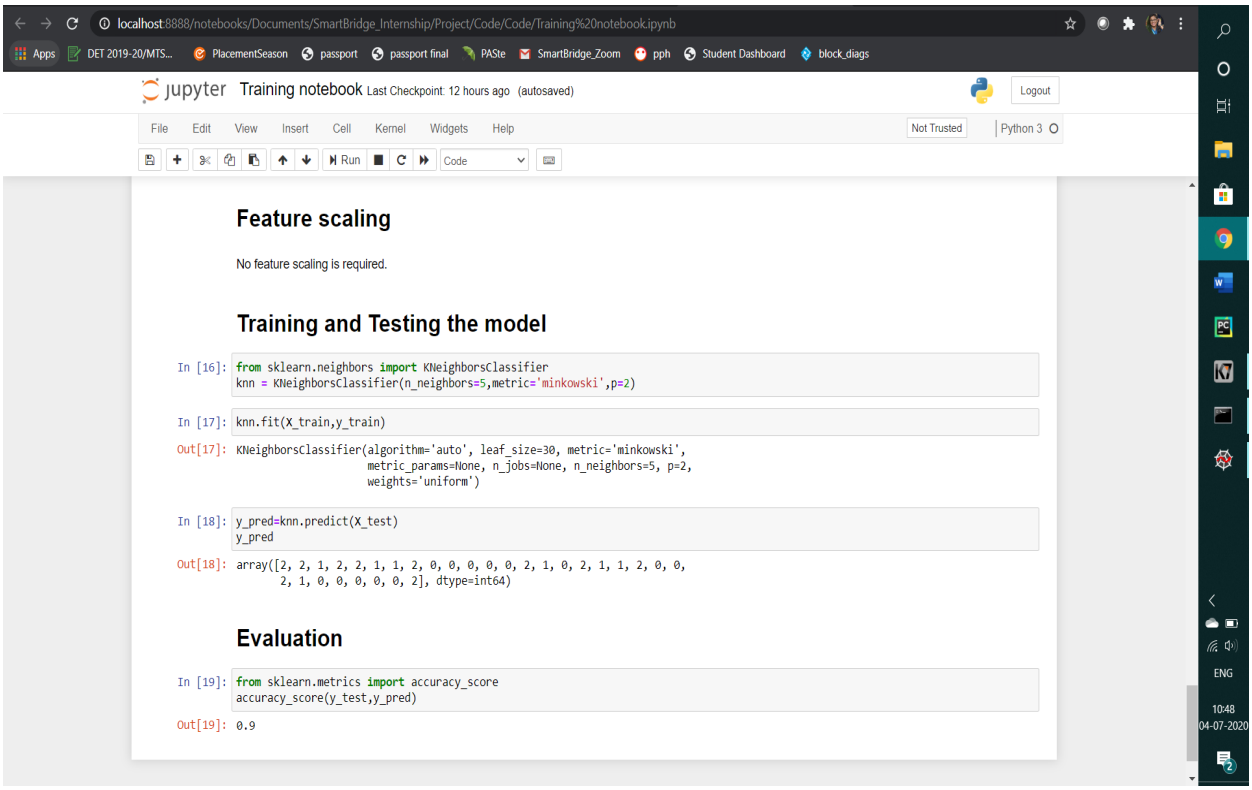

localhost:8888/notebooks/Documents/SmartBridge_Internship/Project/Code/Code/Training%20notebook.ipynb
jupyter Training notebook Last Checkpoint: 12 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code
Data visualization
In [3]:
import seaborn as sns
sns.boxplot(dataset['SBP'])
Out[3]:
<matplotlib.axes._subplots.AxesSubplot at 0x2265cf5ec48>
100 110 120 130 140 150
SBP
In [4]:
import seaborn as sns
sns.boxplot(dataset['DBP'])
Out[4]:
<matplotlib.axes._subplots.AxesSubplot at 0x2265d6c6608>
ENG
10:46
04-07-2020


localhost:8888/notebooks/Documents/SmartBridge_Internship/Project/Code/Code/Training%20notebook.ipynb
jupyter Training notebook Last Checkpoint: 12 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code
In [5]:
import seaborn as sns
sns.boxplot(dataset['Pulse'])
Out[5]:
<matplotlib.axes._subplots.AxesSubplot at 0x2265d743c08>
60 65 70 75 80 85 90
Pulse
In [6]:
import seaborn as sns
sns.boxplot(dataset['Temperature'])
Out[6]:
<matplotlib.axes._subplots.AxesSubplot at 0x2265d79ef48>
ENG
10:46
04-07-2020

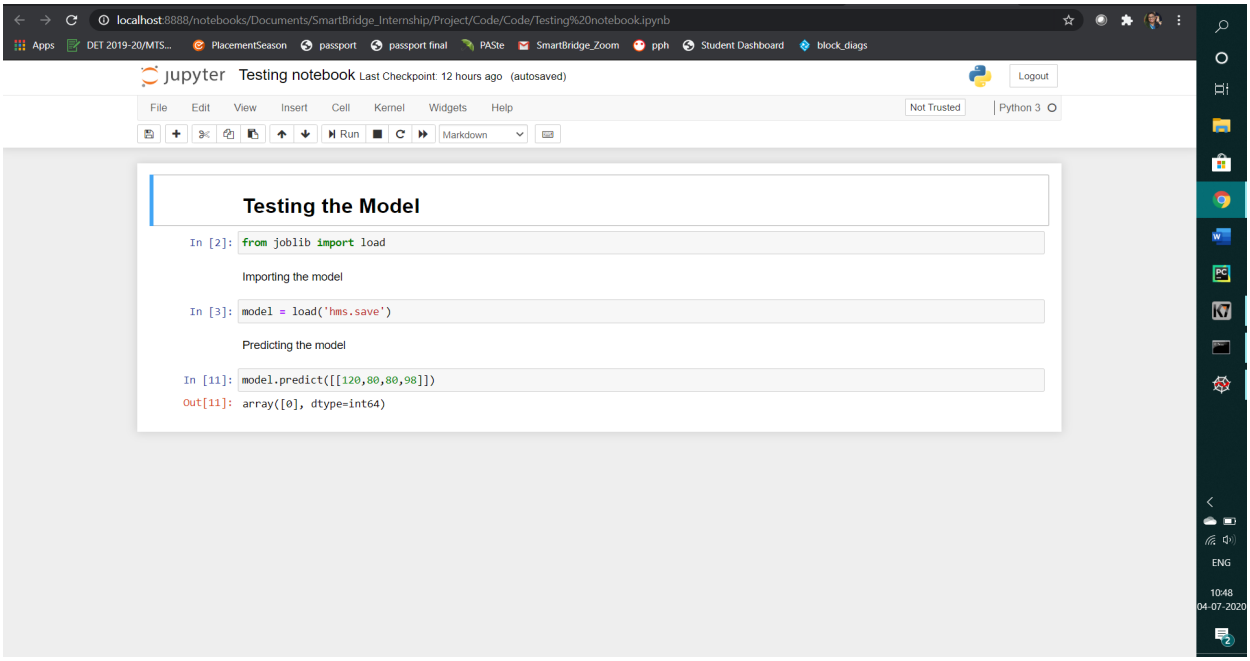

localhost:8888/notebooks/Documents/SmartBridge_Internship/Project/Code/Code/Training%20notebook.ipynb
jupyter Training notebook Last Checkpoint: 12 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Handling missing values
In [7]: dataset.isnull().any()
Out[7]: SBP False
DBP False
Pulse False
Temperature False
Level False
dtype: bool
No missing value
Handling textual data
No Textual data found in dataset
So Label Encoding and One hot Encoding is not required
Splitting the data into Train and Test set
In [8]: x=dataset.iloc[:,0:4].values
y=dataset.iloc[:,4].values
In [9]: x
Out[9]: array([[120, 80, 80, 98],
[140, 83, 75, 100],
[155, 100, 92, 104],
[115, 82, 79, 97],
[119, 79, 85, 102],
[95, 65, 75, 101],
[116, 75, 70, 99],


localhost:8888/notebooks/Documents/SmartBridge_Internship/Project/Code/Code/Training%20notebook.ipynb
jupyter Training notebook Last Checkpoint: 12 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
In [10]: y
Out[10]: array([0, 1, 2, 0, 1, 2, 0, 2, 0, 2, 0, 1, 2, 0, 0, 2, 1, 1, 2, 0, 0, 2,
1, 1, 2, 0, 0, 1, 1, 2, 2, 2, 0, 0, 0, 0, 1, 1, 1, 2, 2, 2, 0, 0,
0, 2, 2, 1, 1, 0, 0, 2, 1, 1, 0, 0, 2, 2, 2, 1, 1, 0, 0, 0, 0, 1,
1, 2, 2, 2, 2, 1, 1, 1, 0, 0, 0, 2, 2, 2, 2, 0, 0, 0, 0, 2, 2, 2,
1, 0, 0, 0, 0, 2, 2, 1, 2, 2, 0, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2,
2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 0, 0, 0,
0, 2, 2, 2, 1, 0, 0, 0, 1, 1, 1, 2, 2, 0, 1, 1, 0], dtype=int64)
In [11]: from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=0)
In [12]: X_train
Out[12]: array([[128, 87, 80, 100],
[99, 66, 79, 102],
[107, 65, 80, 102],
[98, 60, 64, 101],
[140, 91, 80, 102],
[136, 90, 79, 101],
[130, 87, 76, 100],
[118, 81, 72, 98],
[140, 99, 85, 102],
[155, 100, 92, 104],
[125, 87, 79, 102],
[117, 73, 76, 99],
[104, 68, 79, 101],
[154, 99, 79, 100],
[147, 94, 79, 100],
[120, 78, 69, 98],
[128, 86, 64, 101],
[104, 68, 79, 101],
[126, 85, 76, 100],
In [13]: X_test


localhost:8888/notebooks/Documents/SmartBridge_Internship/Project/Code/Code/Training%20notebook.ipynb
jupyter Training notebook Last Checkpoint: 12 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code
In [13]: X_test
Out[13]: array([[98, 64, 76, 98],
[145, 100, 75, 100],
[126, 85, 79, 99],
[100, 63, 80, 103],
[100, 60, 79, 98],
[135, 89, 80, 100],
[134, 86, 78, 100],
[117, 60, 85, 100],
[116, 80, 70, 98],
[115, 78, 70, 98],
[117, 79, 76, 99],
[117, 78, 75, 98],
[119, 78, 70, 98],
[147, 95, 84, 104],
[127, 89, 80, 98],
[119, 78, 74, 98],
[146, 95, 81, 100],
[128, 88, 77, 99],
[132, 89, 76, 101],
In [14]: y_train
Out[14]: array([1, 2, 2, 2, 2, 1, 1, 0, 2, 2, 1, 0, 2, 2, 2, 0, 1, 2, 1, 0, 0, 0,
0, 1, 0, 2, 1, 2, 0, 0, 0, 2, 1, 0, 1, 1, 0, 2, 0, 0, 2, 1, 0, 2,
0, 0, 2, 2, 2, 2, 2, 1, 1, 1, 2, 2, 1, 2, 2, 2, 0, 1, 2, 0, 0, 1, 2,
1, 1, 2, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 2, 1, 2, 0, 1, 0, 0, 0,
0, 0, 2, 0, 0, 2, 1, 0, 0, 0, 2, 0, 2, 1, 2, 0, 0, 0, 1, 2, 2, 1,
2, 2, 1, 2, 2, 1, 2, 0, 1], dtype=int64)
In [15]: y_test
Out[15]: array([2, 2, 1, 2, 2, 1, 1, 1, 0, 0, 0, 0, 0, 2, 1, 0, 2, 1, 1, 2, 2, 0,
2, 1, 0, 2, 0, 0, 0, 2], dtype=int64)
ENG
10:47
04-07-2020


localhost:8888/notebooks/Documents/SmartBridge_Internship/Project/Code/Code/Training%20notebook.ipynb
jupyter Training notebook Last Checkpoint: 12 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code
Feature scaling
No feature scaling is required.
Training and Testing the model
In [16]: from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=5,metric='minkowski',p=2)
In [17]: knn.fit(X_train,y_train)
Out[17]: KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
metric_params=None, n_jobs=None, n_neighbors=5, p=2,
weights='uniform')
In [18]: y_pred=knn.predict(X_test)
y_pred
Out[18]: array([2, 2, 1, 2, 2, 1, 1, 2, 0, 0, 0, 0, 0, 2, 1, 0, 2, 1, 1, 2, 0, 0,
2, 1, 0, 0, 0, 0, 0, 2], dtype=int64)
Evaluation
In [19]: from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)
Out[19]: 0.9
ENG
10:48
04-07-2020

testing the model:



Flask Code: