**Improving the accuracy of predicting the risk of heart diseases using ensemble learning techniques**

A Mini project submitted in the partial fulfillment of the requirements for the award of the degree of

**Master of**

**Computer Applications**

By

**SAHIL GOEL (205118062)**

****

**DEPARTMENT OF COMPUTER APPLICATIONS NATIONAL INSTITUTE OF TECHNOLOGY TIRUCHIRAPPALLI – 620015**

**DECEMBER-2020 (5th Semester)**

**CERTIFICATE**

****

This is to certify that “**Improving the accuracy of predicting the risk of heart diseases using ensemble learning techniques.”** is a Mini project work successfully done by

SAHIL GOEL (205118062)

In partial fulfillment of the requirements for the awards of the degree of **Master of Computer Applications** of **National Institute of Technology, Tiruchirappalli** during the year 2020 (5th Semester).

**Dr. B. Balaji Dr. P. J. A. Alphonse**

**Project Guide Head of the Department**

**BONAFIDE CERTIFICATE**

This is to certify that “**Improving the accuracy of predicting the risk of heart diseases using ensemble learning techniques.”** is a Mini project work successfully done by

SAHIL GOEL (205118062)

In partial fulfillment of the requirements for the awards of the degree of **Master of Computer Applications** of **National Institute of Technology, Tiruchirappalli** during the year 2020 (5th Semester).

**Dr. B. Balaji Dr. P. J. A. Alphonse**

**Project Guide Head of the Department**

Project viva-voce held on ……………………..

**Internal Examiner External Examiner**

**ABSTRACT**

Machine learning involves artificial intelligence, and it is used in solving many problems in data science. One common application of machine learning is the prediction of an outcome based upon existing data. The machine learns patterns from the existing dataset, and then applies them to an unknown dataset in order to predict the outcome.

Classification is a powerful machine learning technique that is commonly used for prediction. Some classification algorithms predict with satisfactory accuracy, whereas others exhibit a limited accuracy. This project investigates a method termed ensemble classification, which is used for improving the accuracy of weak algorithms by combining multiple classifiers. Experiments with this tool were performed using a heart disease dataset.

A comparative analytical approach was done to determine how the ensemble technique can be applied for improving prediction accuracy in heart disease. The focus of this project is not only on increasing the accuracy of weak classification algorithms, but also on the implementation of the algorithm with a medical dataset, to show its utility to predict disease at an early stage. The results of the project indicate that with ensemble techniques such as stacking and by using feature selection we can improve the prediction accuracy of weak classifiers and exhibit satisfactory performance in identifying risk of heart disease.

# ACKNOWLEDGEMENTS

We express our deep sense of gratitude to **Dr. Mini Shaji Thomas**, Director, National Institute of Technology, Tiruchirappalli for giving us an opportunity to do this Mini project.

I wish to extend my sincere gratitude to **Dr. P. J. A. Alphonse**, Professor and Head of the Department, Department of Computer Applications, National Institute of Technology, Tiruchirappalli for supporting and encouraging me to carry out this project work and duly evaluating my progress.

I express my heartfelt gratitude to my internal guide **Dr. B. Balaji** professor, Department of Computer Applications, National Institute of Technology, Tiruchirappalli for his immense support and guidance during the study.

I also thank the review panel for helping me identifying the shortages of the project and helping me to overcome those shortages by guiding me.

Finally, I would like to express my regards for all the faculty members of the Department of Computer Applications and others who have helped me develop this project directly or indirectly.

# TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **Serial No** | **Title** | **Page No** |
| **1** | **CERTIFICATE** | **2** |
| **2** | **BONAFIDE CERTIFIACTE** | **3** |
| **3** | **ABSTRACT** | **4** |
| **4** | **ACKNOWLEDGEMENTS** | **5** |
| **5** | **TABLE OF CONTENTS** | **6** |
| **6** | **CHAPTER-1 INTRODUCTION** | **7** |
| **7** | **CHAPTER-2 PROBLEM STATEMENT** | **8** |
| **8** | **CHAPTER-3 PLATFORM** | **9** |
| **9** | **CHAPTER-4 RELATED WORK** | **10** |
| **10** | **CHAPTER-5 METHODOLOGY USED** | **11** |
| **11** | **CHAPTER-6 DESIGN AND DEVELOPMENT** | **12-14** |
| **12** | **CHAPTER-7 OUTPUT AND RESULT** | **15-17** |
| **13** | **CHAPTER-8 CONCLUSION AND FUTURE WORK** | **18** |
| **14** | **CHAPTER-9 CODE IMPLEMENTATION** | **19-23** |

**CHAPTER – 1 INTRODUCTION**

Cardiovascular Diseases (CVDs) are the number 1 cause of death globally: more people die annually from CVDs than from any other cause. An estimated 17.9 million people died from CVDs in 2016, representing 31% of all global deaths. Of these deaths, 85% are due to heart attack and stroke. Over three quarters of CVD deaths take place in low- and middle-income countries. Out of the 17 million premature deaths (under the age of 70) due to noncommunicable diseases in 2015, 82% are in low- and middle-income countries, and 37% are caused by CVDs. Most cardiovascular diseases can be prevented by addressing behavioural risk factors such as tobacco use, unhealthy diet and obesity, physical inactivity and harmful use of alcohol using population-wide strategies. People with cardiovascular disease or who are at high cardiovascular risk (due to the presence of one or more risk factors such as hypertension, diabetes, hyperlipidaemia or already established disease) need early detection and management using counselling and medicines, as appropriate.

It is difficult to manually determine the odds of getting heart disease based on risk factors. However, machine learning techniques are useful to predict the output from existing data. Hence, this paper applies one such machine learning technique called classification for predicting heart disease risk from the risk factors. It also tries to improve the accuracy of predicting

heart disease risk using a strategy termed ensemble.

# 

# CHAPTER – 2

# PROBLEM STATEMENT

Early detection of cardiovascular disease can be the difference between life and death. By being cognizant of the early signs of CVD, you’ll have a better chance of catching threats early on. When a person has risk factors, their doctor can refer them to a cardiologist for further testing. Some metrics like Blood Pressure, glucose level, age, BMI value, physical exercise, smoking and alcohol history can be used to predict the risk of heart diseases for an individual.

# CHAPTER – 3 PLATFORM

**Hardware:**

* Processor : intel core i5/i7
* Operating system : windows 10 or linux
* Space : 2GB

**Software:**

* Pycharm
* Web Browser

**Web Framework:**

* Flask

**Library:**

* Sklearn
* Matplotlib
* Numpy
* Pandas
* Pic

**CHAPTER – 4**

**RELATED WORK**

Analytical studies on data mining techniques for heart disease prediction reveal that neural networks, decision trees, Naïve Bayes and associative classification are powerful in predicting heart disease. Associative classification produces a high accuracy and strong flexibility as compared with traditional classifiers, even in handling unstructured data.

A comparative analysis of classification techniques has shown that decision tree classifiers are simple and accurate. Naïve Bayes was found to be the best algorithm, followed by neural networks and decision trees. Artificial neural networks are also employed for the prediction of diseases. Supervised networks have been used for diagnosis and they can be trained using the Back-Propagation Algorithm.

Most of the state-of-art research that produces high accuracy employs a hybrid method which include classification algorithms.

**CHAPTER – 5**

**METHODOLGY USED**

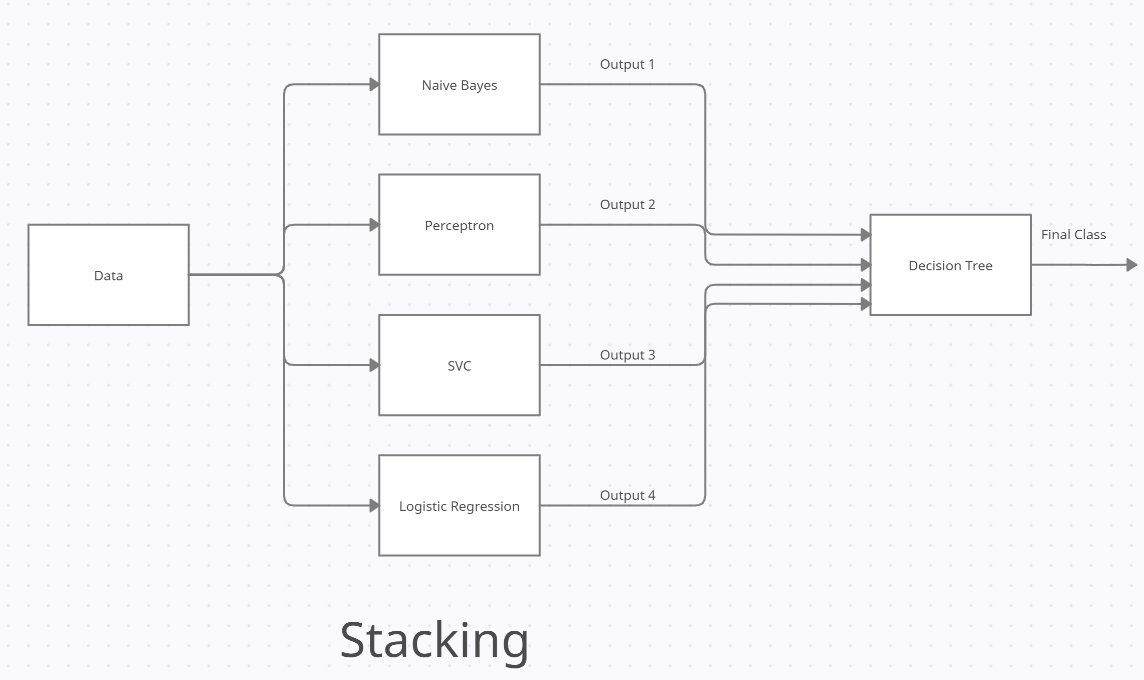
Methodologies used in this project are:

* **Stacking –** Stacking is an ensembling technique in which the training data is passed through various classifiers and the output of these classifiers is passed to another classifier which predicts the final class based on the outputs of original classifiers.
* **Bagging –** Bagging is a technique in which all the classifiers are trained on a subset of original data and the result is calculated based on the class predicted by the classifiers based on majority count.
* **Boosting –** Boosting is a technique in which weights are given to each record and it is increased for those records which are classified incorrectly. The classifier will adjust the parameters to classify the records which have more weight.

**CHAPTER - 6**

**DESIGN AND DEVELOPMENT**

* **Stacking**

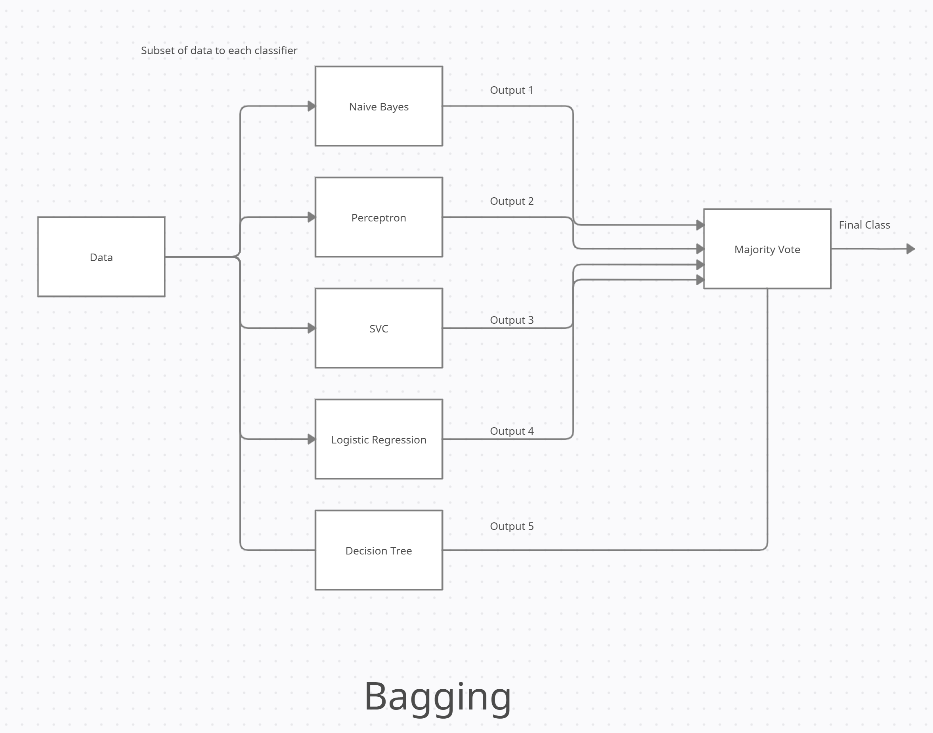


Classifiers used as base classifiers are:

* Naïve Bayes
* Multi-layer Perceptron
* Support Vector Classifier
* Logistic Regression

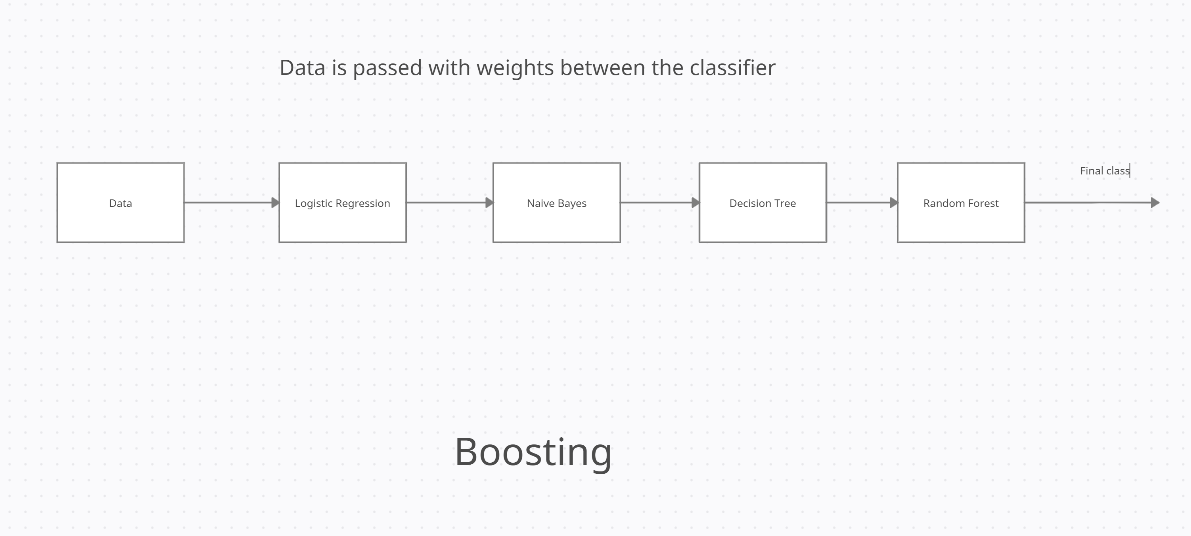
Final Classifier used:

* Decision Tree
* **Bagging**



Classifiers used:

* Naïve Bayes
* Multi-layer Perceptron
* Support Vector Classifier
* Logistic Regression
* Decision Tree
* **Boosting**



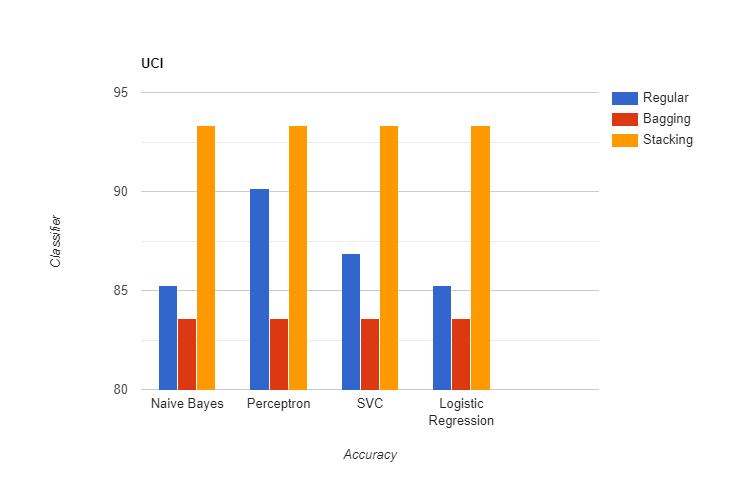
Classifiers used are:

* Logistic Regression
* Naïve Bayes
* Decision Tree
* Random Forest

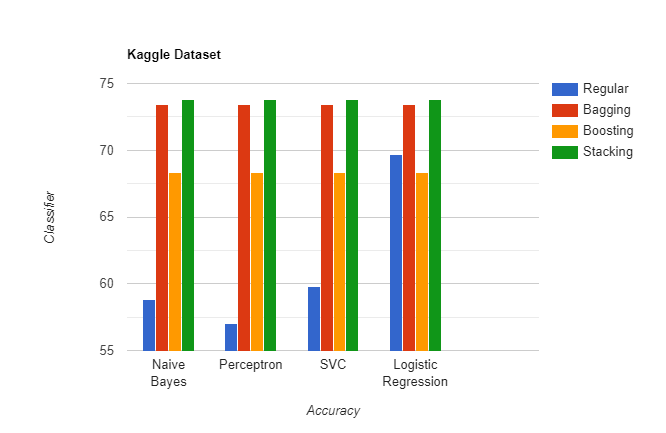
**CHAPTER-7**

**OUTPUT AND RESULT**

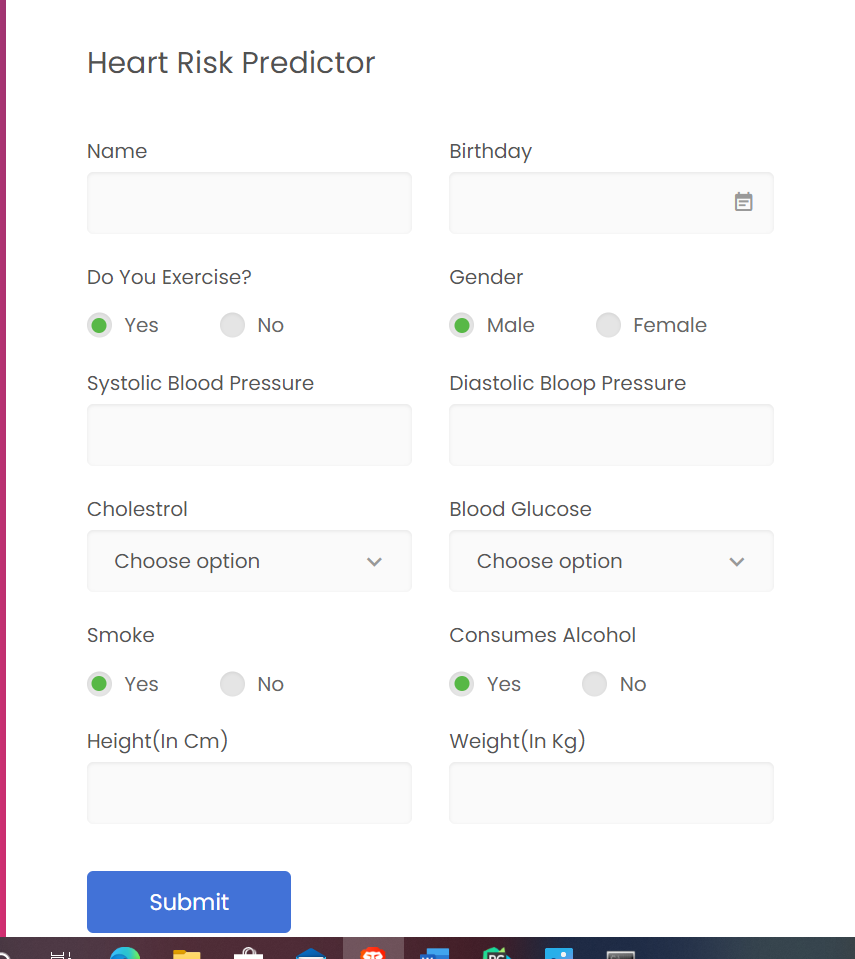
* **UCI Dataset Accuracy:**

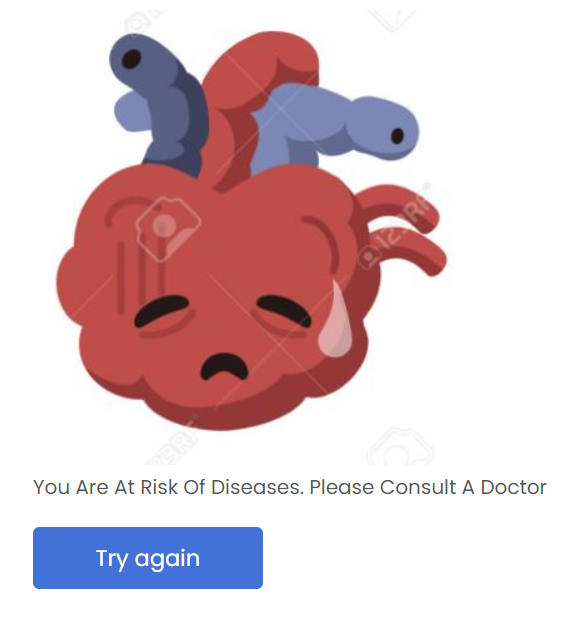
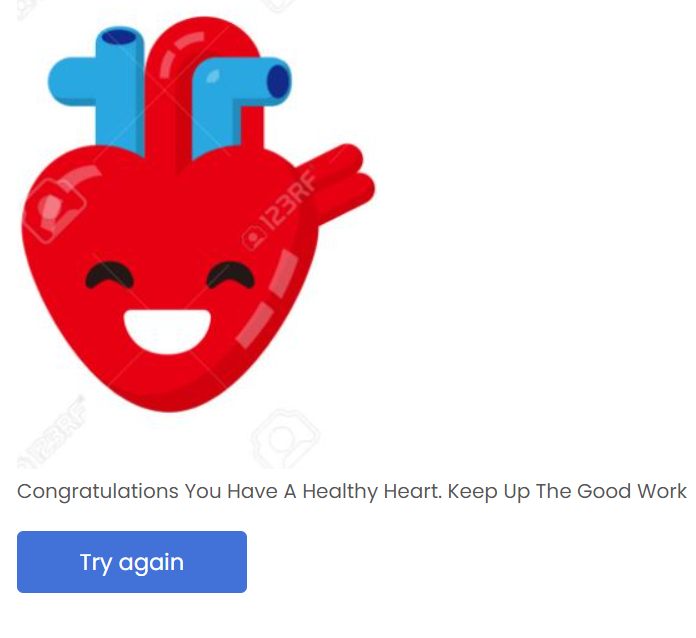


* **Kaggle Dataset**



* **Webpage**



**CHAPTER-8**

**CONCLUSION AND FUTURE WORK**

**Conclusion:**

From the result obtained it is clear that ensemble techniques do help in increasing the accuracy. With the help of this project, an individual can easily know whether he is at risk of disease. Although it is not a substitute for professional medical opinion, it can act as an indicator.

**Future Work:**

* Live learning can be used in which the model can be trained with new data as it is entered. It will help in adjusting the model with new trends in data.
* An application with database can be made through which the user can keep track of the previous records and keep track of the changes.

**CHAPTER-9**

**Code Implementation**

**Form.py**

from flask import Flask, render\_template, flash, request, redirect, url\_for,session

from wtforms import Form, TextField, TextAreaField, validators, StringField, SubmitField, DateField, RadioField

import numpy as np

import pandas as pd

import pickle

import datetime

from os import path

import matplotlib.pyplot as plt

# App config.

DEBUG = True

app = Flask(\_\_name\_\_)

app.config.from\_object(\_\_name\_\_)

app.config['SECRET\_KEY'] = '7d441f27d441f21567d441f2b6176a'

class ReusableForm(Form):

@app.route("/", methods=['GET', 'POST'])

def hello():

form = ReusableForm(request.form)

if session.get('result') is None:

session['result']=-1

return render\_template('index.html',results=int(session['result']), form=form)

@app.route("/check",methods=['POST'])

def check():

gender = request.form['gender']

dateofBirth = request.form['birthday']

systolic = request.form['sysbp']

diastolic = request.form['diabp']

cholestrol = request.form['cholestrol']

smoke = request.form['smoker']

alcohol = request.form['alcohol']

glucose = request.form['glucose']

exercise = request.form['phe']

height = request.form['height']

weight = request.form['weight']

session['result'] = predictor(dateofBirth,gender,systolic,glucose,diastolic,cholestrol,smoke,alcohol,exercise,

height,weight)

return redirect('/')

@app.route("/clear", methods=['POST'])

def clear():

session.clear()

return redirect('/')

if \_\_name\_\_ == "\_\_main\_\_":

app.run()

def predictor(date\_entry,gender,systolic,glucose,diastolic,cholestrol,smoke,alcohol,exercise,height,weight):

columns = ['age', 'gender', 'ap\_hi', 'ap\_lo', 'cholesterol', 'gluc', 'smoke', 'alcohol', 'active', 'bmi']

user\_data = []

day, month, year = map(int, date\_entry.split('/'))

birth\_date = datetime.date(year, month, day)

curr\_date = datetime.date.today()

user\_data.append((curr\_date - birth\_date).days)

user\_data.append(int(gender))

user\_data.append(int(systolic))

user\_data.append(int(diastolic))

user\_data.append(int(cholestrol))

user\_data.append(int(glucose))

user\_data.append(int(smoke))

user\_data.append(int(alcohol))

user\_data.append(int(exercise))

height = int(height)

weight = int(weight)

user\_data.append(weight / (height / 100) \*\* 2)

individual\_predictions = pd.DataFrame()

data = pd.DataFrame([user\_data], columns=columns)

fileName = 'Logistic Regression Stacking.sav'

logClassifier = pickle.load(open(fileName, 'rb'))

individual\_predictions['Logistic'] = logClassifier.predict(data)

fileName = 'Naive bayes Stacking.sav'

nb = pickle.load(open(fileName, 'rb'))

individual\_predictions['Naive Bayes'] = nb.predict(data)

fileName = 'Perceptron Stacking.sav'

perc = pickle.load(open(fileName, 'rb'))

individual\_predictions['Perceptron'] = perc.predict(data)

fileName = 'SVC Stacking.sav'

svc = pickle.load(open(fileName, 'rb'))

individual\_predictions['SVC'] = svc.predict(data)

fileName = 'Stacking.sav'

stacking = pickle.load(open(fileName, 'rb'))

output = pd.DataFrame()

output['answer'] = stacking.predict(individual\_predictions)

return int(output.at[0,'answer'])