COLLEGE OF ENGINEERING, PUNE-411005

2019-20

Prediction System for Heart-Disease-Diagnosis

A Report

Submitted by

Amruta S Jain (121942008)

Jayesh V Jawade (121942010)

INDEX

Prediction System for Heart-Disease-Diagnosis

Introduction

The total number of deaths due to Heart diseases reads in million in a year. Thus, how to predict heart diseases in real life is of great significance. In this project, we plan to develop a machine learning system that can classify a patient into different stages of heart disease classes. The diagnosis of heart disease can be classified into various classes based on the Electrocardiogram (ECG) readings and other attributes. First class will refer to the normal patient while other classes shall represent different classes of different stages of patient suffers by heart disease like as stage1, stage2, stage3 and stage4, according to other attributes such as fasting blood sugar, serum cholesterol, etc. This is a supervised learning problem.

Dataset

In the Heart disease data set consists of patient data from Cleveland, Hungary, Long Beach and Switzerland. The combined dataset consists of 14 features and 916 samples with many missing values. From that data is get pre-processed and following features are used.

The features used in here are as follows:

1. age: The patients age in years

2. sex: The patient gender (1=male; 0=female)

3. cp: Chest pain type,

- Value 1: typical angina
- Value 2: atypical angina
- Value 3: non-anginal pain
- Value 4: asymptomatic

4. trestbps: Resting blood pressure (in mm Hg on admission to the hospital)

5. chol: Serum cholestoral in mg/dl

6. fbs: Fasting blood sugar > 120 mg/dl? (1=true; 0=false)

7. restecg: 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

8. thalach: Maximum heart rate achieved

9. exang: Chest pain(angina) after exercise? (1=yes; 0=no)

10. thal: Not described

- Value 3=normal
- Value 6=treated defect
- Value 7=reversible defect

11. num: Target

- Value 0: less than 50% narrowing of coronary arteries (no heart disease)
- Value 1,2,3,4: >50% narrowing. The value indicates the stage of heart disease

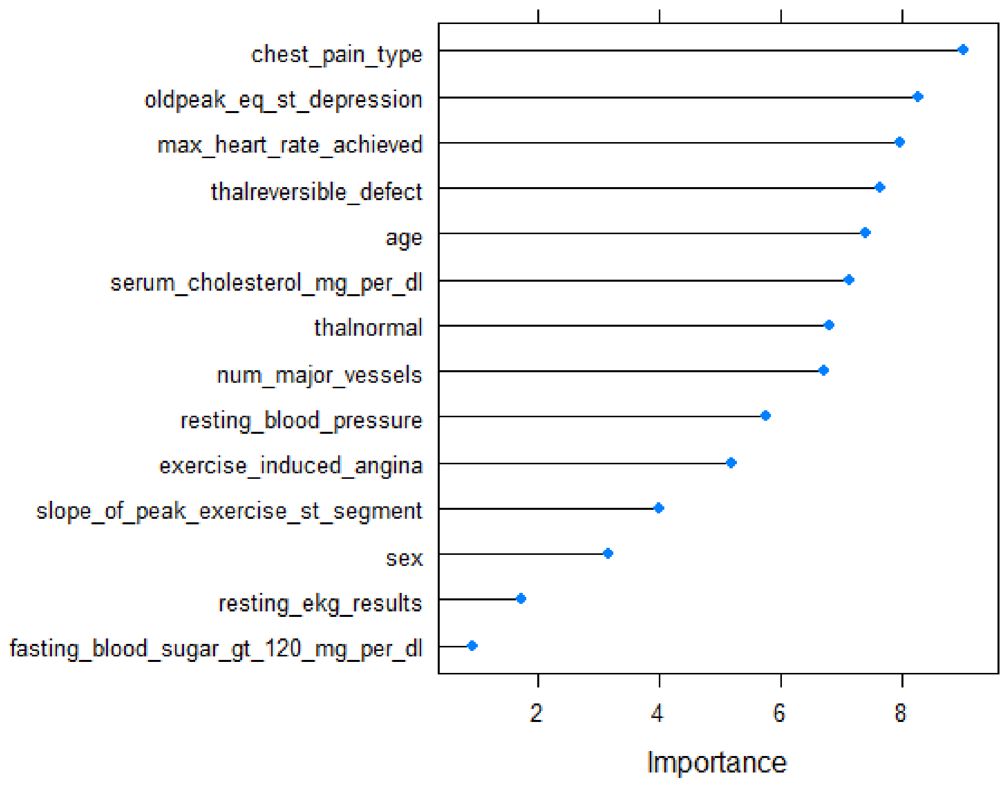


Fig: Variable Importance

Survey

The aim is to distinguish between the presence and absence of heart disease and to classify it in one of the 4 stages. For the time being, there exists a computer program that makes such a classification. However, there are differences between the cardiolog's and the programs classification. Taking the cardiolog's as a gold standard we aim to minimize this difference by means of machine learning tools.

Scope

These machine learning techniques can be deployed in hospitals where a large dataset is available and can help the doctors in making more precise decisions and to cut down the number of causalities due to heart diseases in the future. It is also helpful for all common peoples who get result of their ECG report in easy form as there heart disease in which stage and according to that, what precautions and medicine have to take for cure.

Methodology

1. Feature Selection:

   Firstly, we removed some of the categorical features that were not very essential in heart disease diagnosis. If any training instance has a missing value for a given attribute, we set it as sum of the value or for some we taken it as median, for that attribute related to the class it belongs to. If for a given attribute majority of values are missing, then we discard that attribute and remove or drop it from our dataset. As in this project dataset is used which is from four different datasets, and by doing pre-processing combined dataset is made.

   The features can be grouped into blocks –

   - features concerning biographical characteristics, i.e., age, sex, height, weight and heart rate.
   - features concerning average wave durations of each interval (PR interval, QRS complex, and ST intervals).
   - features concerning with fasting blood sugar, previous ECG reports, etc.
   - Some features are drop from all that datasets which is having no use or important. To reduce size as well as time of processing dataset as see in combined dataset.

2. Random Forests Classifier

This classifier is ensemble algorithm. Ensemble algorithm are those which combines more than one algorithm of same or different kind for classifying object. As it creates a set of decision trees from randomly selected subset of training dataset. It then aggregates the votes from different decision trees to decide the final class of the test object.

We implement a Random Forest classifier. The model works by continually sampling with replacement a portion of the training dataset, and fitting a decision tree to it. The number of trees refer to the number of times the dataset is randomly sampled. Moreover, in each sampling iteration, a random set of features are selected. In decision trees, each node refers to one of the input variables, which has edges to children for all possible values that the input can take. Each leaf corresponds to a value of the class label given the values of the input variables represented by the path from the root node to the leaf node. The number of trees and the number of leaves is learned via cross validation.

Execution of project

<u>Core code of project</u> :-

File - main_file.py :

```
from flask import Flask, render_template, url_for, request
from sklearn.externals import joblib
import os
import numpy as np
import pickle

app = Flask(__name__, static_folder='static')

@app.route("/")
def index():
    return render_template('home.html')

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

```
def result():
    age = int(request.form['age'])
    sex = int(request.form['sex'])
    trestbps = float(request.form['trestbps'])
    chol = float(request.form['chol'])
    restecg = float(request.form['restecg'])
    thalach = float(request.form['thalach'])
    exang = int(request.form['exang'])
    cp = int(request.form['cp'])
    fbs = float(request.form['fbs'])
    x = np.array([age, sex, cp, trestbps, chol, fbs, restecg,
                  thalach, exang]).reshape(1, -1)
    scaler_path = os.path.join(os.path.dirname(__file__), 'models/scaler.pkl')
    scaler = None
    with open(scaler_path, 'rb') as f:
        scaler = pickle.load(f)
    x = scaler.transform(x)
    model_path = os.path.join(os.path.dirname(__file__), 'models/rfc.sav')
    clf = joblib.load(model_path)
    y = clf.predict(x)
    print(y)
# No heart disease
    if y == 0:
        return render_template('nodisease.html')
    # y=1,2,4,4 are stages of heart disease
    else:
        return render_template('heartdisease.htm', stage=int(y))


@app.route('/about')
def about():
    return render_template('about.html')
if __name__ == "__main__":
    app.run(debug=True)
```

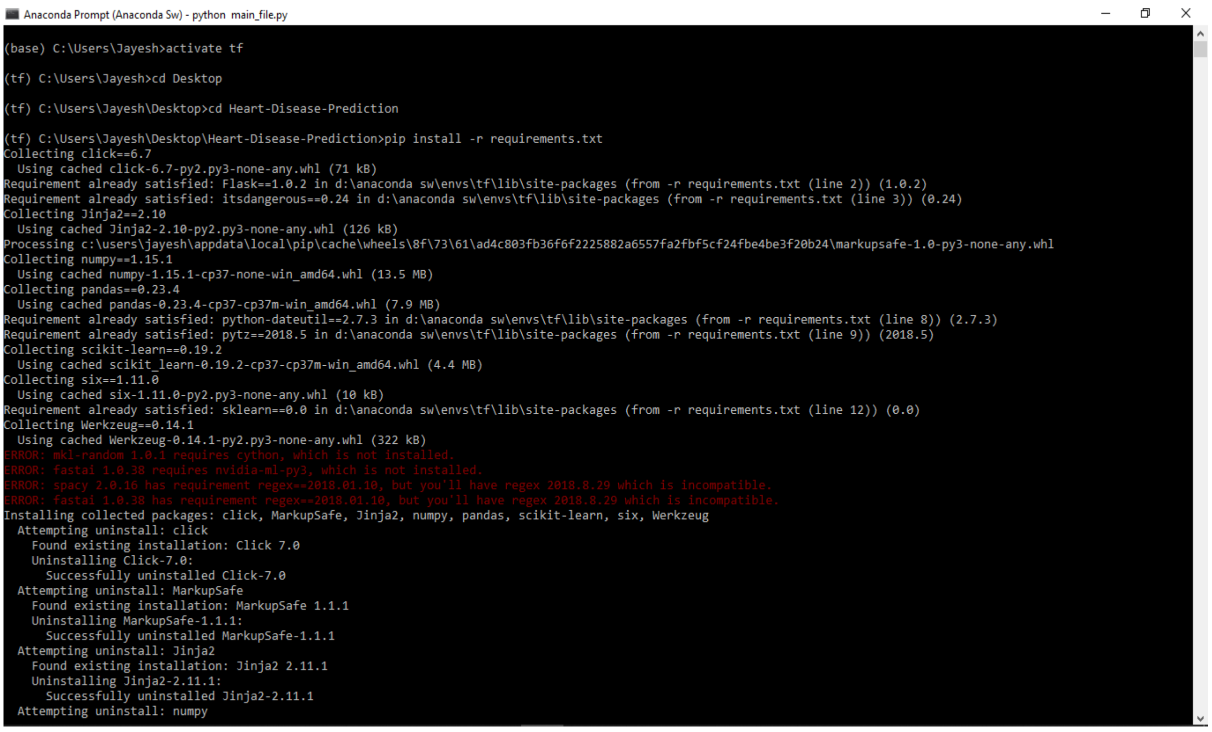
File - model.py :

```
from sklearn.svm import SVC, LinearSVC
from sklearn.preprocessing import MinMaxScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.externals import joblib
import numpy as np
import pandas as pd
import pickle
import os
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
root = os.path.dirname(__file__)
path_df = os.path.join(root, 'recons_dataset/combined_dataset.csv')
data = pd.read_csv(path_df)
scaler = MinMaxScaler()
train, test = train_test_split(data, test_size=0.25)
X_train = train.drop('num', axis=1)
Y_train = train['num']
X_test = test.drop('num', axis=1)
Y_test = test['num']
# We don't scale targets: Y_test, Y_train as SVC returns the class labels not probability
values
X_train = scaler.fit_transform(X_train)
X_test = scaler.fit_transform(X_test)
clf = RandomForestClassifier()
```

```
# Training the classifier
clf.fit(X_train, Y_train)



# Testing model accuracy. Average is taken as test set is very small hence accuracy varies a lot everytime the model is trained
acc = 0
acc_binary = 0
for i in range(0, 20):
    Y_hat = clf.predict(X_test)
    Y_hat_bin = Y_hat>0
    Y_test_bin = Y_test>0
    acc = acc + accuracy_score(Y_hat, Y_test)
    acc_binary = acc_binary +accuracy_score(Y_hat_bin, Y_test_bin)

print("Average test Accuracy:{}".format(acc/20))
print("Average binary accuracy:{}".format(acc_binary/20))

# Saving the trained model for inference
model_path = os.path.join(root, 'models/rfc.sav')
joblib.dump(clf, model_path)

# Saving the scaler object
scaler_path = os.path.join(root, 'models/scaler.pkl')
with open(scaler_path, 'wb') as scaler_file:
    pickle.dump(scaler, scaler_file)
```

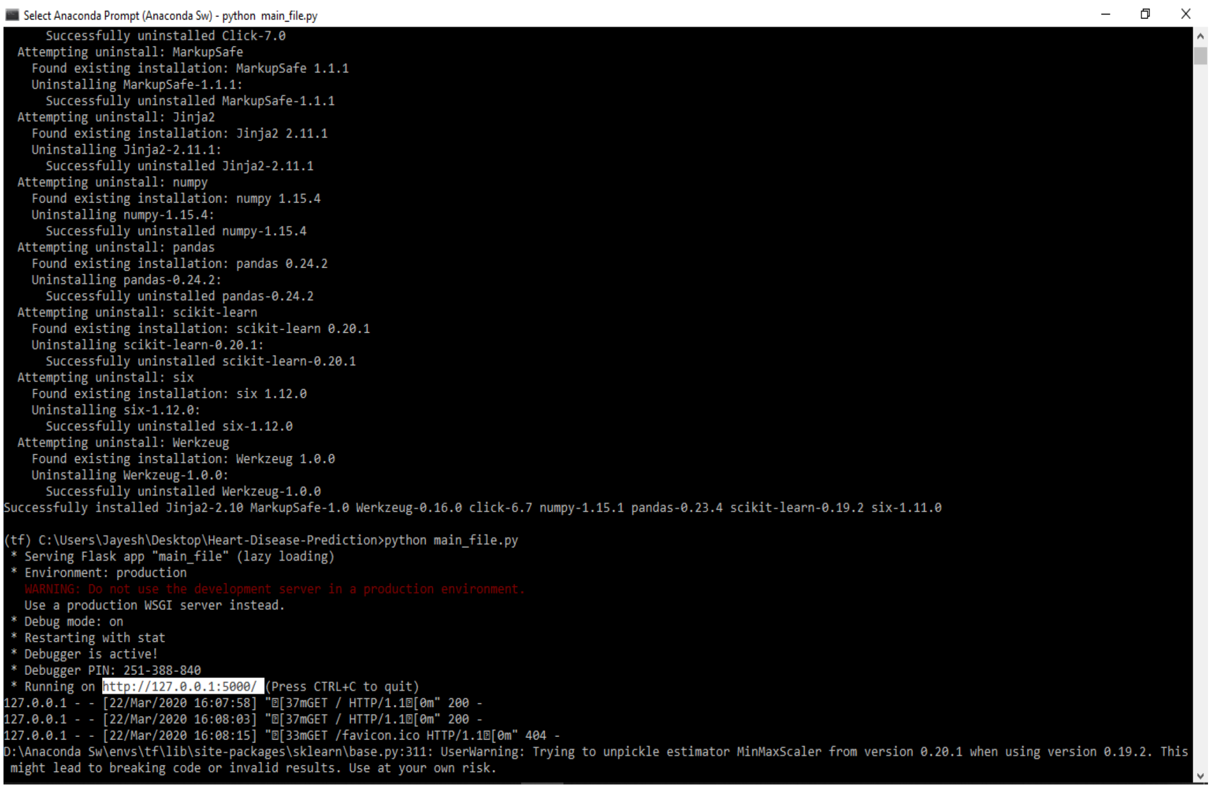
<u>Output of Project</u> :-

1. Compile and run project in anaconda prompt, as shown in below images.



Anaconda Prompt (Anaconda Sw) - python main_file.py

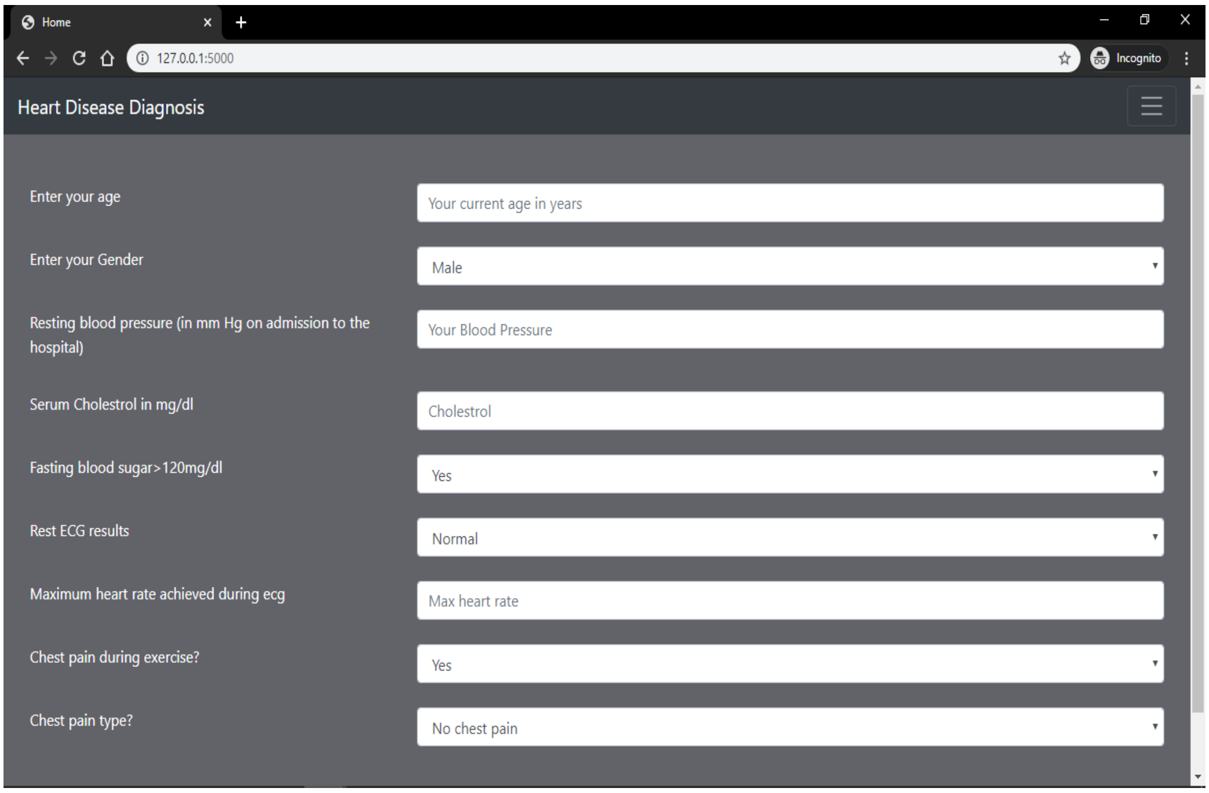
```
(base) C:\Users\Jayesh>activate tf

(tf) C:\Users\Jayesh>cd Desktop

(tf) C:\Users\Jayesh\Desktop>cd Heart-Disease-Prediction

(tf) C:\Users\Jayesh\Desktop\Heart-Disease-Prediction>pip install -r requirements.txt
Collecting click==6.7
  Using cached click-6.7-py2.py3-none-any.whl (71 kB)
Requirement already satisfied: Flask==1.0.2 in d:\anaconda sw\envs\tf\lib\site-packages (from -r requirements.txt (line 2)) (1.0.2)
Requirement already satisfied: itsdangerous==0.24 in d:\anaconda sw\envs\tf\lib\site-packages (from -r requirements.txt (line 3)) (0.24)
Collecting Jinja2==2.10
  Using cached Jinja2-2.10-py2.py3-none-any.whl (126 kB)
Processing c:\users\jayesh\appdata\local\pip\cache\wheels\8f\73\61\ad4c803fb36f6f2225882a6557fa2fbf5cf24fbe4be3f20b24\markupsafe-1.0-py3-none-any.whl
Collecting numpy==1.15.1
  Using cached numpy-1.15.1-cp37-none-win_amd64.whl (13.5 MB)
Collecting pandas==0.23.4
  Using cached pandas-0.23.4-cp37-cp37m-win_amd64.whl (7.9 MB)
Requirement already satisfied: python-dateutil==2.7.3 in d:\anaconda sw\envs\tf\lib\site-packages (from -r requirements.txt (line 8)) (2.7.3)
Requirement already satisfied: pytz==2018.5 in d:\anaconda sw\envs\tf\lib\site-packages (from -r requirements.txt (line 9)) (2018.5)
Collecting scikit-learn==0.19.2
  Using cached scikit_learn-0.19.2-cp37-cp37m-win_amd64.whl (4.4 MB)
Collecting six==1.11.0
  Using cached six-1.11.0-py2.py3-none-any.whl (10 kB)
Requirement already satisfied: sklearn==0.0 in d:\anaconda sw\envs\tf\lib\site-packages (from -r requirements.txt (line 12)) (0.0)
Collecting Werkzeug==0.14.1
  Using cached Werkzeug-0.14.1-py2.py3-none-any.whl (322 kB)
ERROR: mkl-random 1.0.1 requires cython, which is not installed.
ERROR: fastai 1.0.38 requires nvidia-ml-py3, which is not installed.
ERROR: spacy 2.0.16 has requirement regex==2018.01.10, but you'll have regex 2018.8.29 which is incompatible.
ERROR: fastai 1.0.38 has requirement regex==2018.01.10, but you'll have regex 2018.8.29 which is incompatible.
Installing collected packages: click, MarkupSafe, Jinja2, numpy, pandas, scikit-learn, six, Werkzeug
  Attempting uninstall: click
    Found existing installation: Click 7.0
    Uninstalling Click-7.0:
      Successfully uninstalled Click-7.0
  Attempting uninstall: MarkupSafe
    Found existing installation: MarkupSafe 1.1.1
    Uninstalling MarkupSafe-1.1.1:
      Successfully uninstalled MarkupSafe-1.1.1
  Attempting uninstall: Jinja2
    Found existing installation: Jinja2 2.11.1
    Uninstalling Jinja2-2.11.1:
      Successfully uninstalled Jinja2-2.11.1
  Attempting uninstall: numpy
```



Select Anaconda Prompt (Anaconda Sw) - python main_file.py

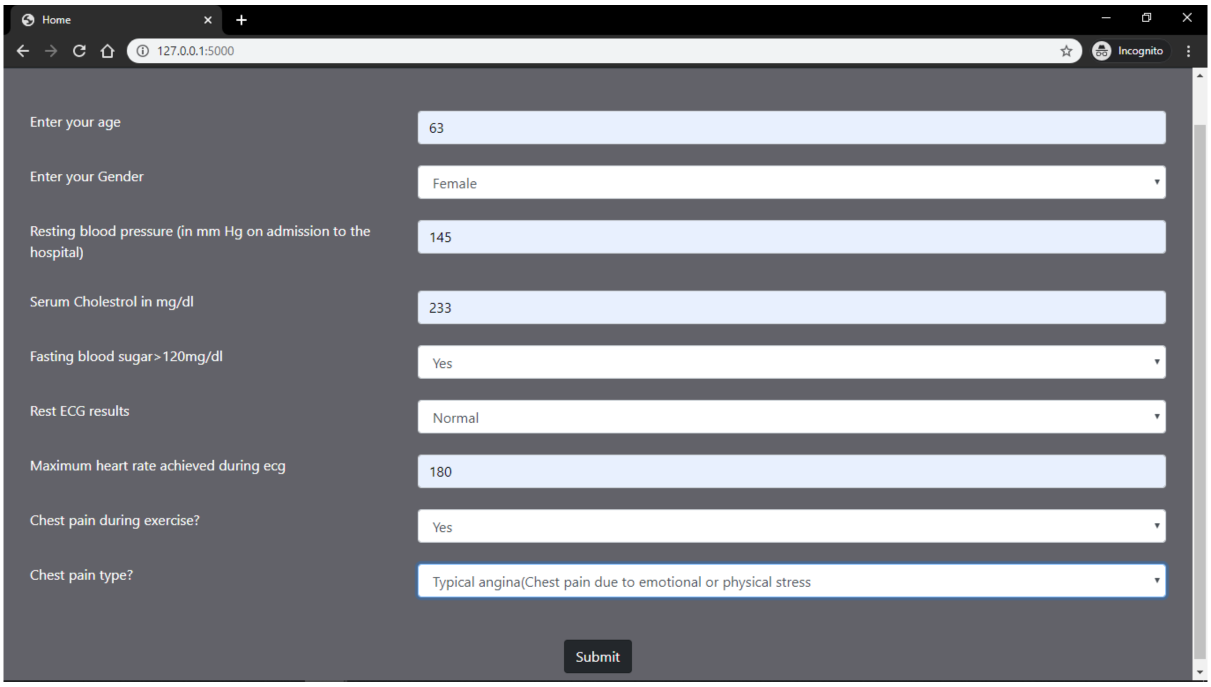
```
      Successfully uninstalled Click-7.0
  Attempting uninstall: MarkupSafe
    Found existing installation: MarkupSafe 1.1.1
    Uninstalling MarkupSafe-1.1.1:
      Successfully uninstalled MarkupSafe-1.1.1
  Attempting uninstall: Jinja2
    Found existing installation: Jinja2 2.11.1
    Uninstalling Jinja2-2.11.1:
      Successfully uninstalled Jinja2-2.11.1
  Attempting uninstall: numpy
    Found existing installation: numpy 1.15.4
    Uninstalling numpy-1.15.4:
      Successfully uninstalled numpy-1.15.4
  Attempting uninstall: pandas
    Found existing installation: pandas 0.24.2
    Uninstalling pandas-0.24.2:
      Successfully uninstalled pandas-0.24.2
  Attempting uninstall: scikit-learn
    Found existing installation: scikit-learn 0.20.1
    Uninstalling scikit-learn-0.20.1:
      Successfully uninstalled scikit-learn-0.20.1
  Attempting uninstall: six
    Found existing installation: six 1.12.0
    Uninstalling six-1.12.0:
      Successfully uninstalled six-1.12.0
  Attempting uninstall: Werkzeug
    Found existing installation: Werkzeug 1.0.0
    Uninstalling Werkzeug-1.0.0:
      Successfully uninstalled Werkzeug-1.0.0
Successfully installed Jinja2-2.10 MarkupSafe-1.0 Werkzeug-0.16.0 click-6.7 numpy-1.15.1 pandas-0.23.4 scikit-learn-0.19.2 six-1.11.0

(tf) C:\Users\Jayesh\Desktop\Heart-Disease-Prediction>python main_file.py
 * Serving Flask app "main_file" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 251-388-840
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [22/Mar/2020 16:07:58] "[37mGET / HTTP/1.1[0m" 200 -
127.0.0.1 - - [22/Mar/2020 16:08:03] "[37mGET / HTTP/1.1[0m" 200 -
127.0.0.1 - - [22/Mar/2020 16:08:15] "[33mGET /favicon.ico HTTP/1.1[0m" 404 -
D:\Anaconda Sw\envs\tf\lib\site-packages\sklearn\base.py:311: UserWarning: Trying to unpickle estimator MinMaxScaler from version 0.20.1 when using version 0.19.2. This
 might lead to breaking code or invalid results. Use at your own risk.
```

2. Run the flask web app (GUI) in browser, GUI is provided for better understanding of non-technical persons.

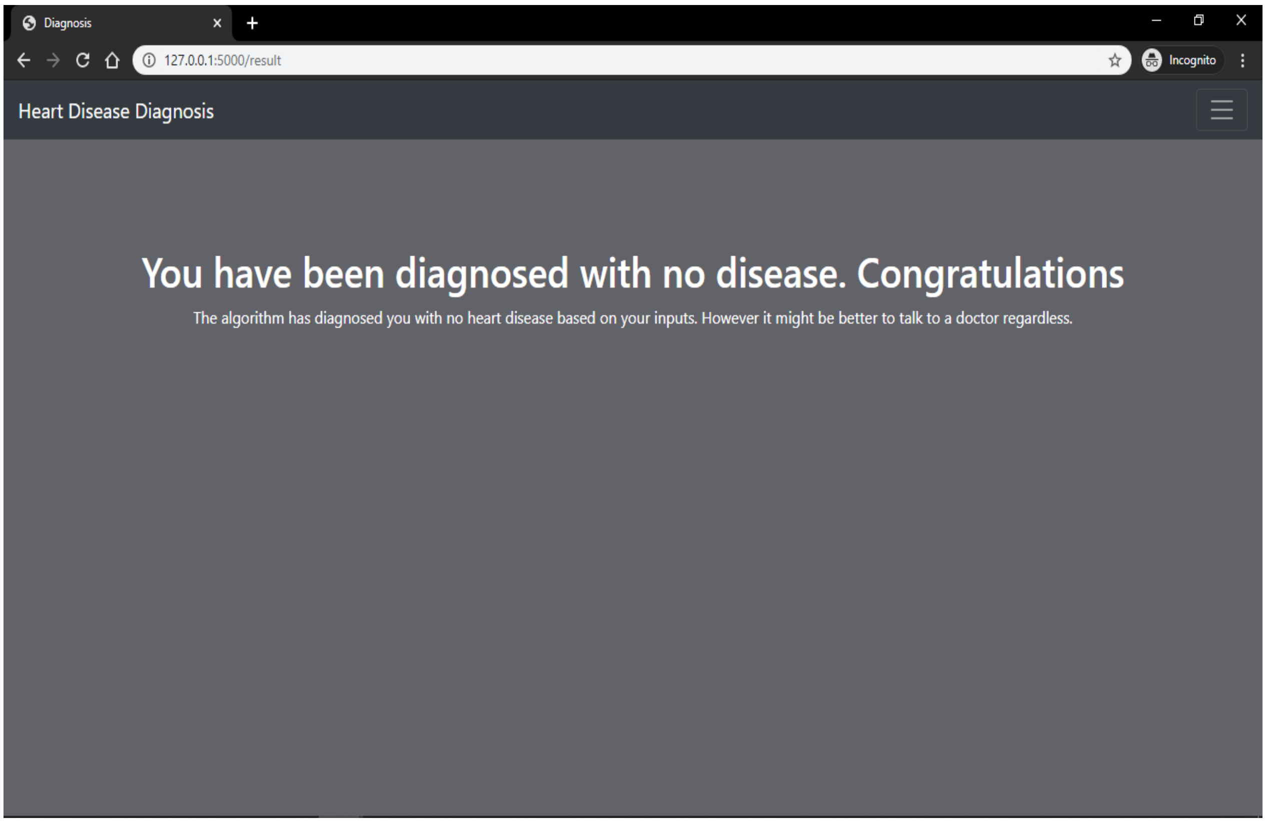


Heart Disease Diagnosis

Enter your age	Your current age in years
Enter your Gender	Male
Resting blood pressure (in mm Hg on admission to the hospital)	Your Blood Pressure
Serum Cholestrol in mg/dl	Cholestrol
Fasting blood sugar>120mg/dl	Yes
Rest ECG results	Normal
Maximum heart rate achieved during ecg	Max heart rate
Chest pain during exercise?	Yes
Chest pain type?	No chest pain



Enter your age	63
Enter your Gender	Female
Resting blood pressure (in mm Hg on admission to the hospital)	145
Serum Cholestrol in mg/dl	233
Fasting blood sugar>120mg/dl	Yes
Rest ECG results	Normal
Maximum heart rate achieved during ecg	180
Chest pain during exercise?	Yes
Chest pain type?	Typical angina(Chest pain due to emotional or physical stress



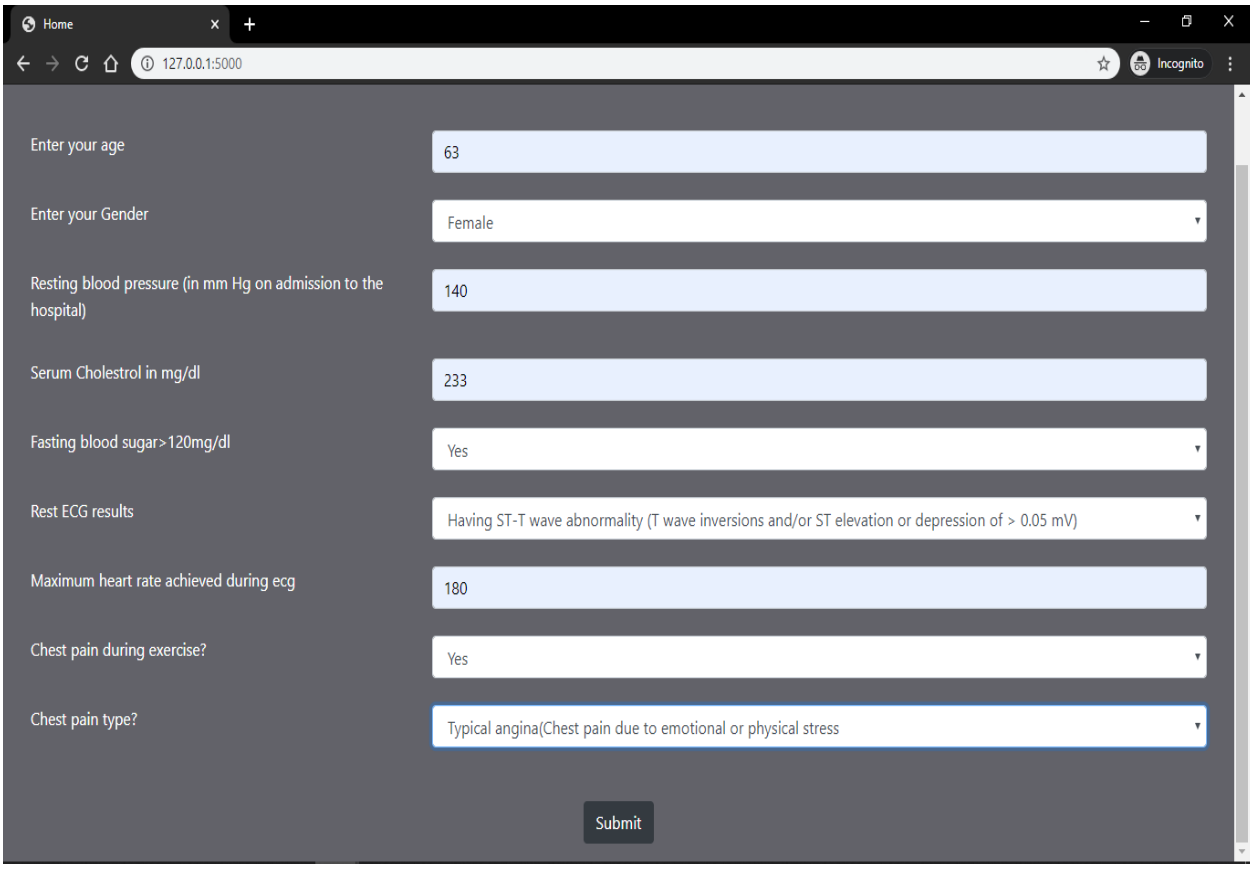
Diagnosis

127.0.0.1:5000/result

Heart Disease Diagnosis

You have been diagnosed with no disease. Congratulations

The algorithm has diagnosed you with no heart disease based on your inputs. However it might be better to talk to a doctor regardless.

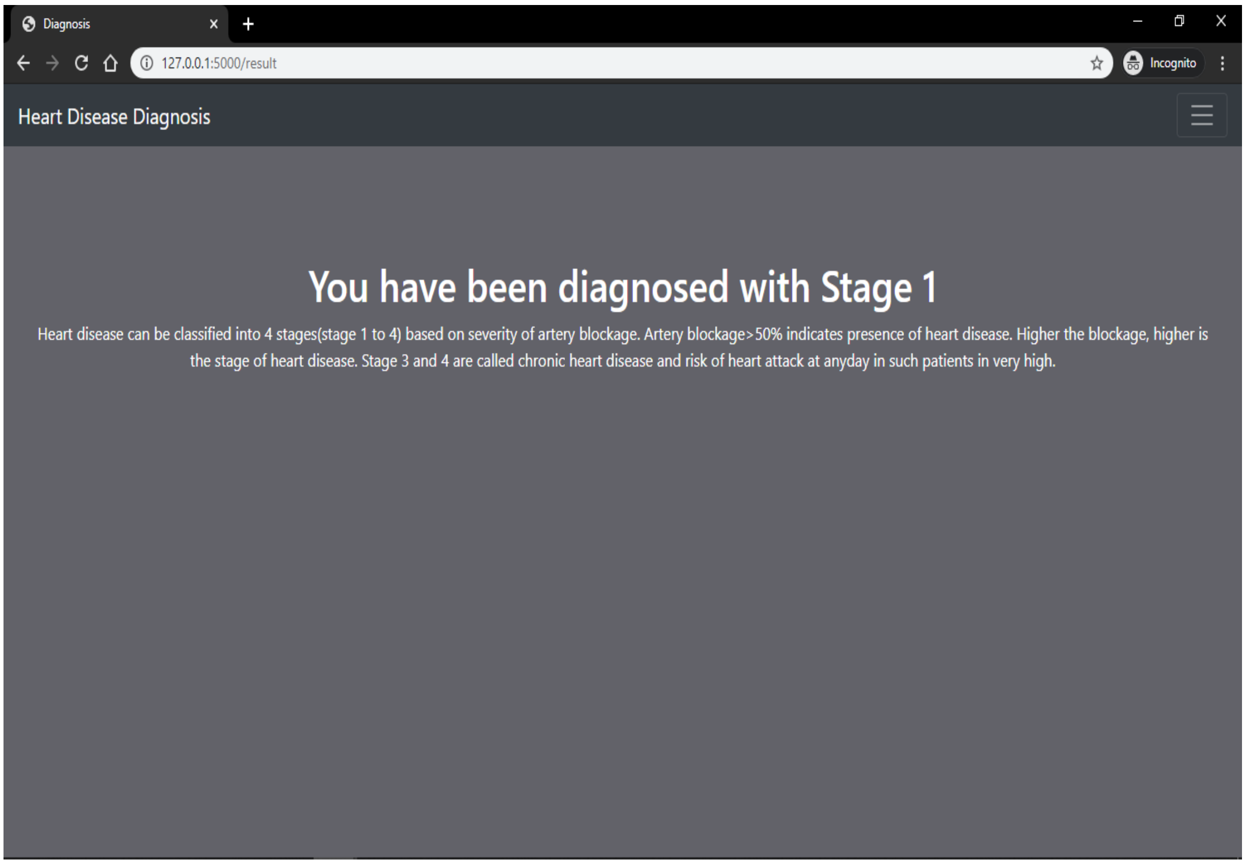


Home

127.0.0.1:5000

Enter your age	63
Enter your Gender	Female
Resting blood pressure (in mm Hg on admission to the hospital)	140
Serum Cholestrol in mg/dl	233
Fasting blood sugar>120mg/dl	Yes
Rest ECG results	Having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV)
Maximum heart rate achieved during ecg	180
Chest pain during exercise?	Yes
Chest pain type?	Typical angina(Chest pain due to emotional or physical stress

Submit



You have been diagnosed with Stage 1

Heart disease can be classified into 4 stages(stage 1 to 4) based on severity of artery blockage. Artery blockage>50% indicates presence of heart disease. Higher the blockage, higher is the stage of heart disease. Stage 3 and 4 are called chronic heart disease and risk of heart attack at anyday in such patients in very high.

Conclusion

In this project, we develop a system that could robustly detect a stage number that indicate how much severe pain or attack on heart is detected. Secondly in this project, we develop a method to robustly classify an ECG, trace or predict result on new object. We show results, as well as vary other parameters or apply some other classifier or technique, algorithm for better results. We finally use Random Forest algorithm model to implement our project with accuracy of 75-80 % for binary prediction of heart disease.

Models used and accuracy

A Random forest classifier achieves an average multi-class classification as heart pain varies from stage1 to stage4 accuracy of 56-60% (183 test samples). It gets 75-80% average binary classification accuracy (heart disease or no heart disease)

References

[1] Heart Disease Prediction Using Machine learning and Data Mining Technique by Jaymin Patel, Prof. Tejal Upadhyay, Dr. Samir Patel

[2] Aljanabi, Maryam & Qutqut, Mahmoud & Hijjawi, Mohammad. (2018). Machine Learning Classification Techniques for Heart Disease Prediction: A Review. International Journal of Engineering and Technology. 7. 5373-5379. 10.14419/ijet.v7i4.28646.

[3] Data Science for Good — Machine Learning for Heart Disease Prediction by Margaret Wanjiru.