Project Report

On

MULTIPLE DISEASES PREDICTION SYSTEM

Submitted By

U GOVARDHAN(R170856)

T BABJI(R170854)

S VIVEK(R170851)

Under the guidance of

P Santhosh Kumar

Department of Computer Science Engineering

Rajiv Gandhi University of Knowledge and Technologies(RGUKT),

R.K.Valley, Kadapa, Andhra Pradesh.

Rajiv Gandhi University of Knowledge and Technologies (RGUKT),

R.K.Valley, Kadapa, Andhra Pradesh.

CERTIFICATE

This is certify that the project work titled “**MULTIPLE DISEASE PREDICTION SYSTEM**” is a bonafide **project work** submitted by U Govardhan (R170856) in the department of COMPUTER SCIENCE ENGINEERING in partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering for the academic year 2022-2023 carried out the work under the supervision

P SANTHOSH KUMAR

Assistant professor,

Department of CSE,

RGUKT-RK VALLEY

N.SATYANANDARAM

Head of the Department,

Department of CSE,

RGUKT-RK VALLEY

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant guidance and encouragement crown all the efforts success.

I am extremely grateful to our respected Director, **Prof. K. SANDHYA RANI** for fostering an excellent academic climate in our institution.

I also express my sincere gratitude to our respected Head of the Department **Mr.N.SATYANANDARAM** for his encouragement, overall guidance in viewing this project a good asset and effort in bringing out this project.

I would like to convey thanks to our guide at college **Mr. P SANTHOSH KUMAR** for his guidance, encouragement, co-operation and kindness during the entire duration of the course and academics.

My sincere thanks to all the members who helped me directly and indirectly in the completion of project work. I express my profound gratitude to all our friends and family members for their encouragement.

INDEX

Abstract

With the increasing incidence of chronic diseases such as heart disease, cancer, diabetes, and COVID-19, there is a growing need for effective and accurate methods for disease prediction.

This project presents a multiple disease prediction system that uses machine learning algorithms to predict the likelihood of four major diseases: heart disease, breast cancer, diabetes, and COVID-19. The system is based on a dataset of patient records, including demographic and clinical data, as well as laboratory test results. The data is preprocessed and transformed into a suitable format for training and testing machine learning models.

For each disease, several classification models are evaluated, and the best performing models are selected based on their accuracy, sensitivity, and specificity. The heart disease prediction model is based on several risk factors such as age, gender, blood pressure, cholesterol levels.

The breast cancer prediction model uses features such as age, family history, and mammogram results. The diabetes prediction model takes into account factors such as age, body mass index, blood pressure, and glucose levels.

The COVID-19 prediction model uses clinical symptoms, laboratory test results, and demographic information. The results of the evaluation show that the multiple disease prediction system achieves high accuracy in predicting each of the four diseases.

The system can assist healthcare professionals in early diagnosis and intervention, potentially leading to improved patient outcomes and reduced healthcare costs. In conclusion, this project demonstrates the feasibility and effectiveness of a machine learning-based multiple disease prediction system. With further refinement and validation, the system has the potential to become a valuable tool in disease diagnosis and management.

INTRODUCTION

"Healthcare is an important aspect of our lives, and early detection of diseases can be crucial in providing effective treatment and improving health outcomes. With the advent of machine learning, there has been a significant increase in the development of predictive models that can aid in disease diagnosis and prognosis.

In this project, we aim to build a multiple disease prediction system that can predict the likelihood of four major diseases - diabetes, COVID-19, heart disease, and breast cancer - using machine learning algorithms and the test train split method. The system will utilize a dataset containing various health-related features, such as age, gender, blood pressure, and clinical data of patient,

We will employ different machine learning algorithms, such as logistic regression, Support vector machine to build predictive models for each disease. The test train split method will be used to train and evaluate the models, ensuring that they are accurate and generalizable.

By accurately predicting the likelihood of these diseases, healthcare professionals can take proactive measures to prevent or treat the diseases, ultimately improving patient outcomes. Overall, this project has the potential to revolutionize the healthcare industry by providing a reliable, efficient, and cost-effective means of predicting and preventing multiple diseases."

PURPOSE

The major purpose of this project is to develop a machine learning based multiple disease prediction system that can accurately predict the likelihood of four major diseases: Heart disease, Breast cancer, Diabetics, and Covid-19.The system aims to assist health care professionals in early diagnosis and intervention, potentially leading to improve patient outcomes and reduce health care costs

SCOPE

"The Multiple Disease Prediction Model currently predicts a few diseases, but in the upcoming days, we plan to expand it to include additional diseases such as lung cancer, stroke, or Alzheimer's disease.

However, it's important to note that achieving 100% accuracy may not be possible, particularly when dealing with complex health conditions that are influenced by many factors. Nonetheless, we will strive to improve the accuracy of our model by incorporating more data, refining our algorithms, and validating our predictions against clinical outcomes."

Requirement Specifications

Hardware Specification:

Processor	AMD RYZEN$^{(tm)}$ 5 5000 series or Intel 5i 10 gen or more Clock speed:3.0 GHz or more
RAM	8GB or more
Hard disk	512GB or More

Software Specifications:

Language:	Python
Tools:	Anaconda, Jupyter Notebook, Spyder, Github
GUI:	Streamlit
Modules:	▪ Numpy ▪ Pandas ▪ Matplotlib ▪ Pickle ▪ Sklearn ▪ Streamlit

Implementation

To implement the models of disease predictions we need Jupyter notebook from Anaconda IDE.

Anaconda is a distribution of the Python and R programming languages for scientific computing such as data science, machine learning applications etc. that aims to simply package management and deployment. It is suitable for Windows, Linux and macOS too, Anaconda comes with over 250 pre-installed packages.

Anaconda Distribution provides these applications by default:

- **JupyterLab**
- **Jupyter Notebook**
- **QtConsole**
- **Spyder**
- **Glue**
- **Orange**
- **RStudio**
- **Visual Studio Code….etc.**

Installation of Anaconda:

- Visit the Anaconda official website to download the anaconda distribution file
- It will detect automatically your system OS and gives the file to download
- Open the downloaded file and run with windows system administrator (if your OS is windows).
- Accept the License Agreement of Anaconda Distribution’s and follow the process mentioned in the Documentation of Anaconda official website for successful installation and setup.
- Launch the Jupyter Notebook in Anaconda Navigator to implement prediction models.

1.Heart Disease Prediction:

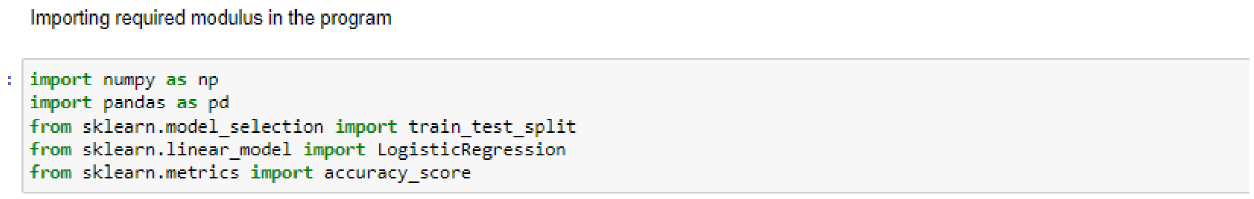
- Heart Disease Prediction is a machine learning model that can accurately predict whether a person has cardiovascular disease or not by considering certain attributes such as age, gender, cholesterol, chest pain etc.
- The process of developing a heart disease prediction model typically involves collecting data from many individuals who have been diagnosed with or without heart disease.
- This data is used to train a machine learning algorithm, which is then able to identify patterns and relationships between the various risk factors and the likelihood of heart disease.
- The collected data needs to be pre-processed so that it can be compatible for machine learning algorithms. The processed data is split into training data and testing data. The Training data is used to train the heart disease prediction model and the testing data is used to evaluate the performance of the model, this process is called as Train Test split.
- Logistic regression algorithm and training data are used to develop heart disease prediction model. Logistic regression is a Machine Learning classification algorithm that is used to predict categorical variables like Yes/No, True/False, 0/1, etc. Oncemodel is developed, it is used to predict whether a person has heart disease or not based on the newly collected attributes.

1.1 Heart Disease Prediction Model:

Here is step by step procedure how heart disease prediction model is developed.

1.1.1: Import required modules.

Import required modules into Jupyter Notebook that helps to develop heart disease prediction model.

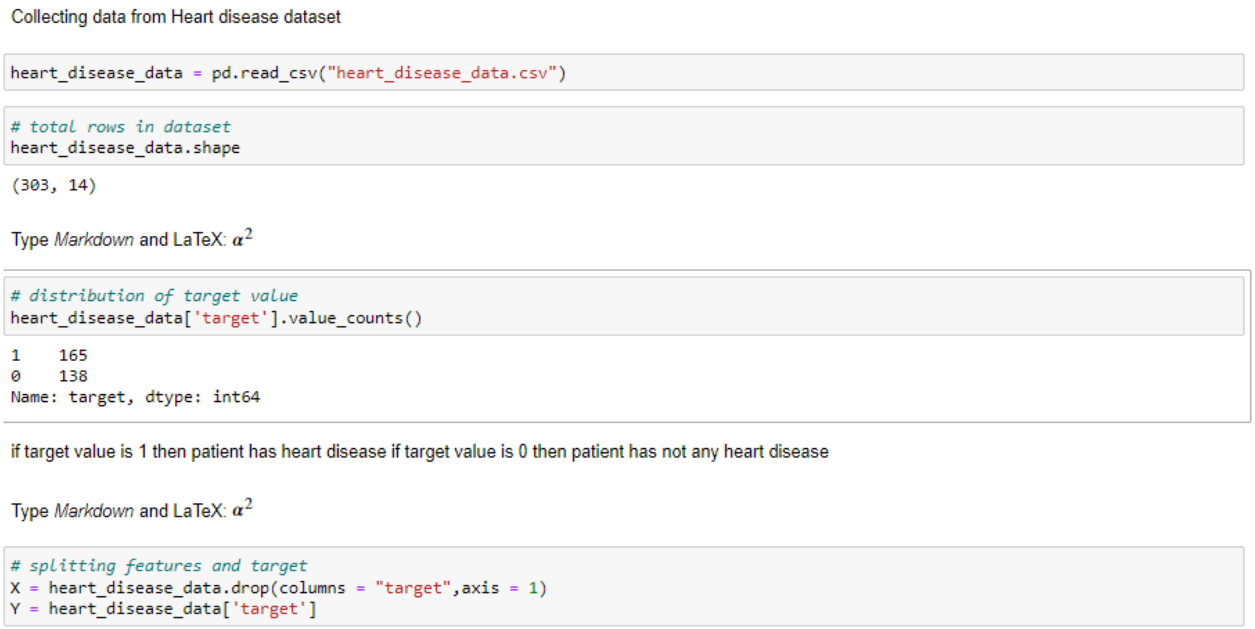


Importing required modulus in the program

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

1.1.2: Data collection and Processing

Read collected dataset into Jupyter Notebook and Split all attributes of heart disease into one variable and all results of patient into another variable as processing data.



Collecting data from Heart disease dataset

```
heart_disease_data = pd.read_csv("heart_disease_data.csv")
```

```
# total rows in dataset
heart_disease_data.shape
```

```
(303, 14)
```

Type *Markdown* and LaTeX: α^2

```
# distribution of target value
heart_disease_data['target'].value_counts()
```

```
1    165
0    138
Name: target, dtype: int64
```

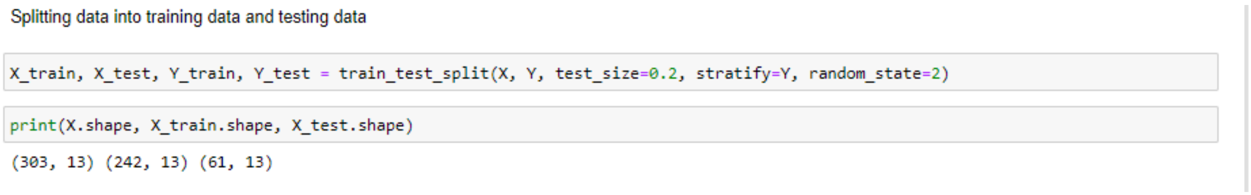
if target value is 1 then patient has heart disease if target value is 0 then patient has not any heart disease

Type *Markdown* and LaTeX: α^2

```
# splitting features and target
X = heart_disease_data.drop(columns = "target",axis = 1)
Y = heart_disease_data['target']
```

1.1.3: Splitting the processed data into training data and test data.

Split the attributes and results of the patients into training data of the attributes, training data of the result, testing data of the attributes and testing data of the results using train_test_split method.



Splitting data into training data and testing data

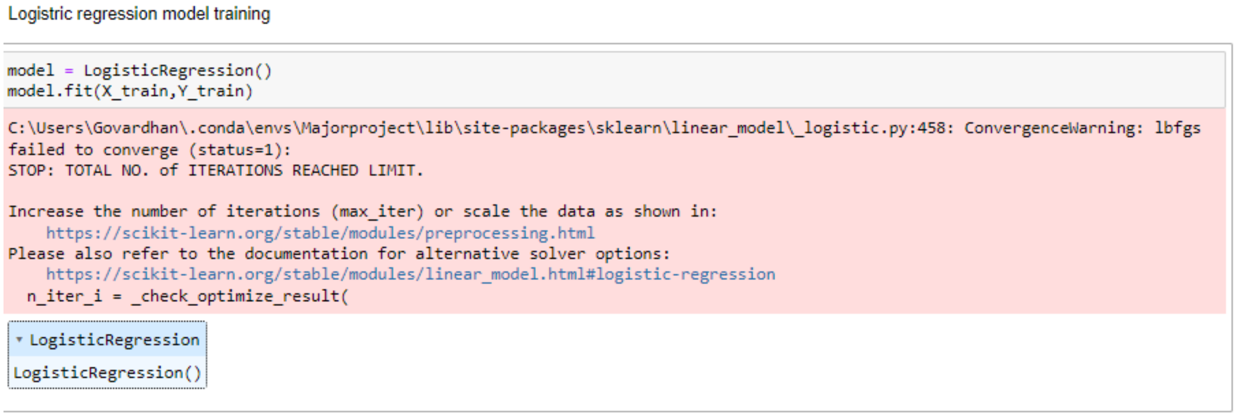
```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, stratify=Y, random_state=2)
```

```
print(X.shape, X_train.shape, X_test.shape)
```

```
(303, 13) (242, 13) (61, 13)
```

1.1.4: Train logistic regression algorithm

Train logistic regression with training data of attributes and results by using model.fit() method to build heart disease prediction model

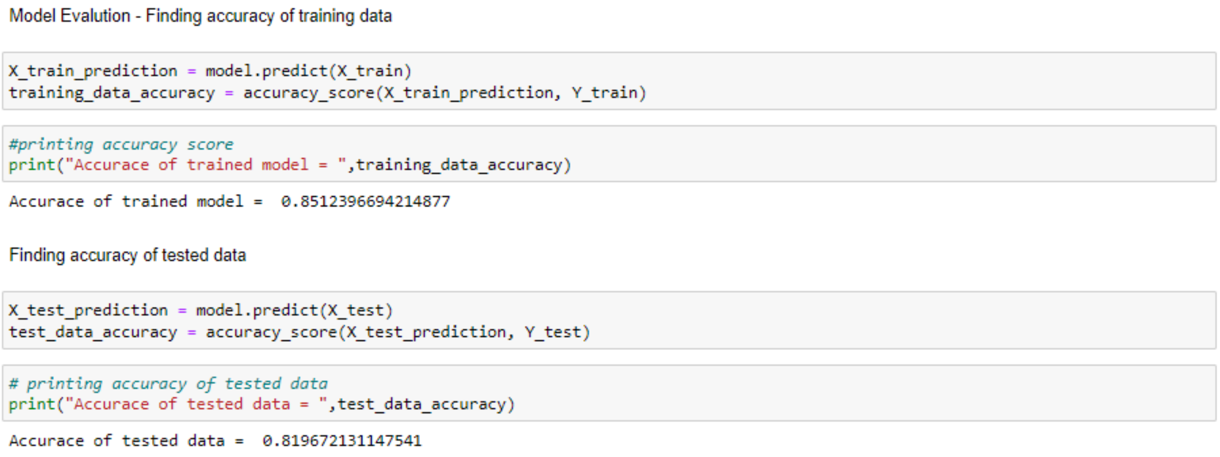


Logistic regression model training

```
model = LogisticRegression()
model.fit(X_train,Y_train)
```

```
C:\Users\Govardhan\.conda\envs\Majorproject\lib\site-packages\sklearn\linear_model\_logistic.py:458: ConvergenceWarning: lbfgs
failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

```
LogisticRegression()
```



Model Evalution - Finding accuracy of training data

```
X_train_prediction = model.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)
```

```
#printing accuracy score
print("Accurace of trained model = ",training_data_accuracy)
```

```
Accurace of trained model =  0.8512396694214877
```

Finding accuracy of tested data

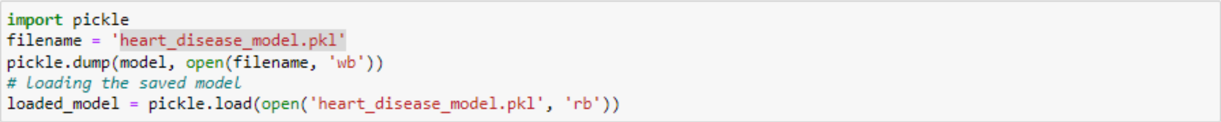
```
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
```

```
# printing accuracy of tested data
print("Accurace of tested data = ",test_data_accuracy)
```

```
Accurace of tested data =  0.819672131147541
```

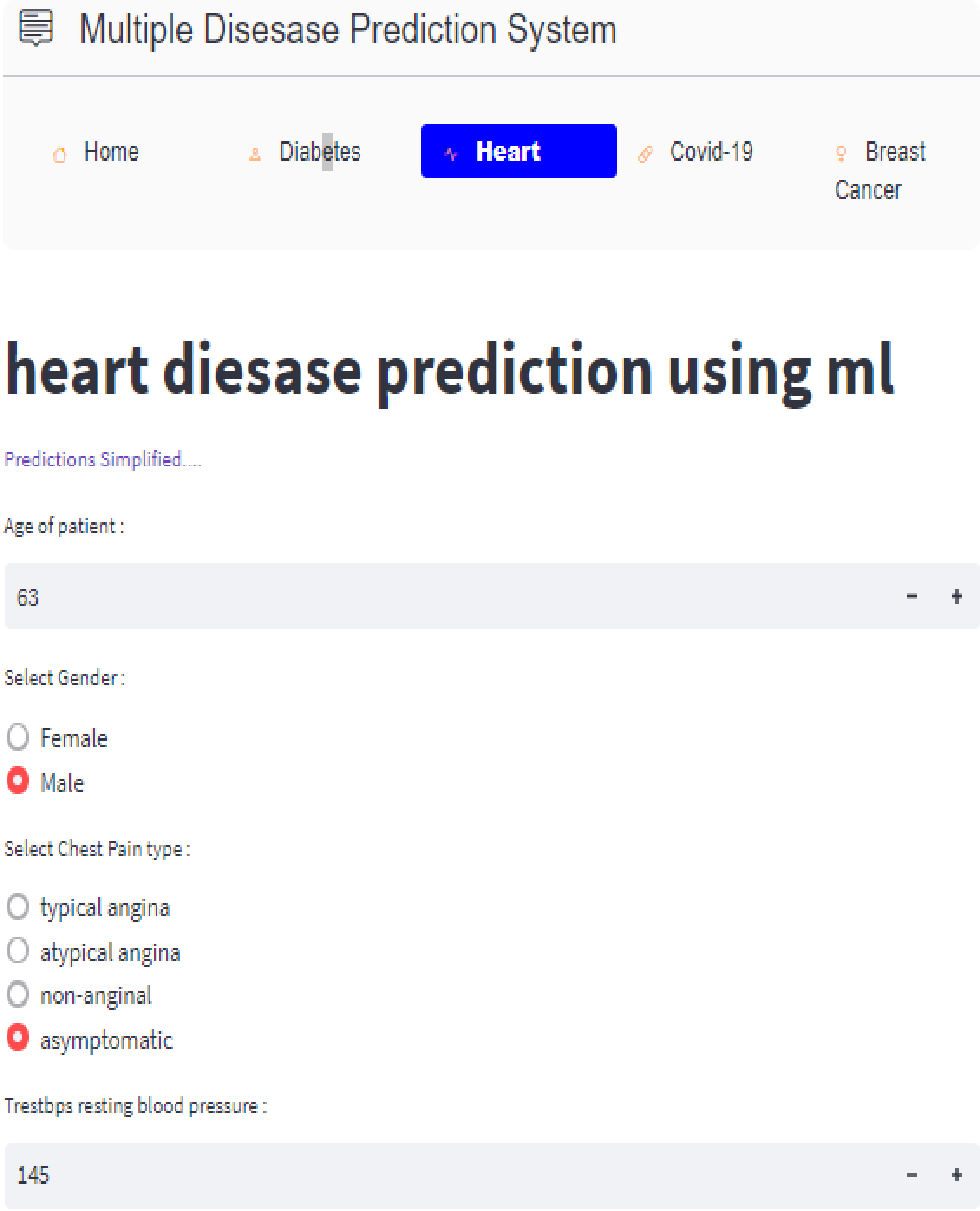
1.1.5 : Convert model into a pickle file.

Converting model into pickle module to integrate the model with python web application file.



```
import pickle
filename = 'heart_disease_model.pkl'
pickle.dump(model, open(filename, 'wb'))
# loading the saved model
loaded_model = pickle.load(open('heart_disease_model.pkl', 'rb'))
```

Input and output



heart diesease prediction using ml

Predictions Simplified....

Age of patient :

63

Select Gender :

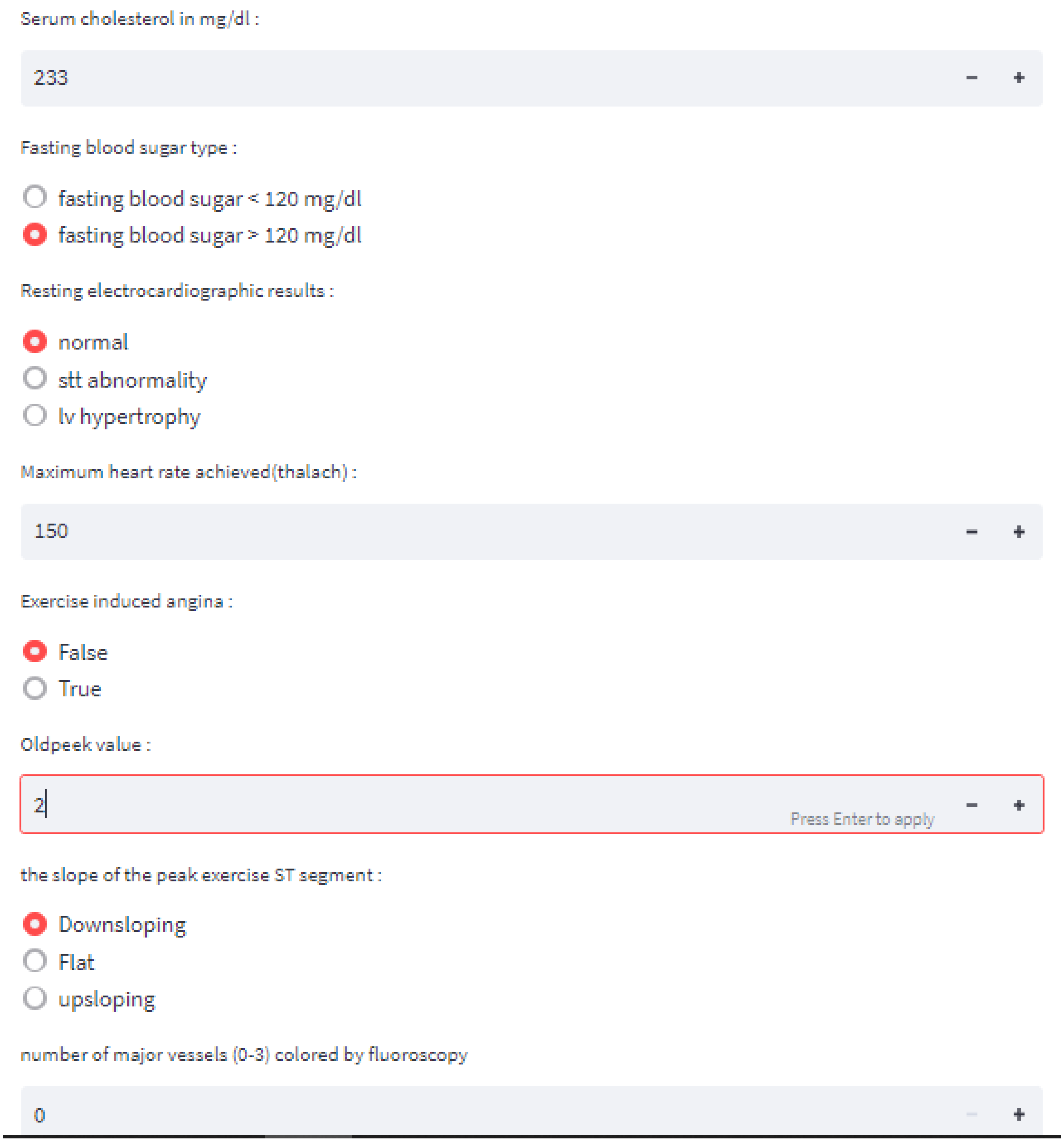
○ Female
● Male

Select Chest Pain type :

○ typical angina
○ atypical angina
○ non-anginal
● asymptomatic

Trestbps resting blood pressure :

145



Serum cholesterol in mg/dl :

233

Fasting blood sugar type :

○ fasting blood sugar < 120 mg/dl
● fasting blood sugar > 120 mg/dl

Resting electrocardiographic results :

● normal
○ stt abnormality
○ lv hypertrophy

Maximum heart rate achieved(thalach) :

150

Exercise induced angina :

● False
○ True

Oldpeek value :

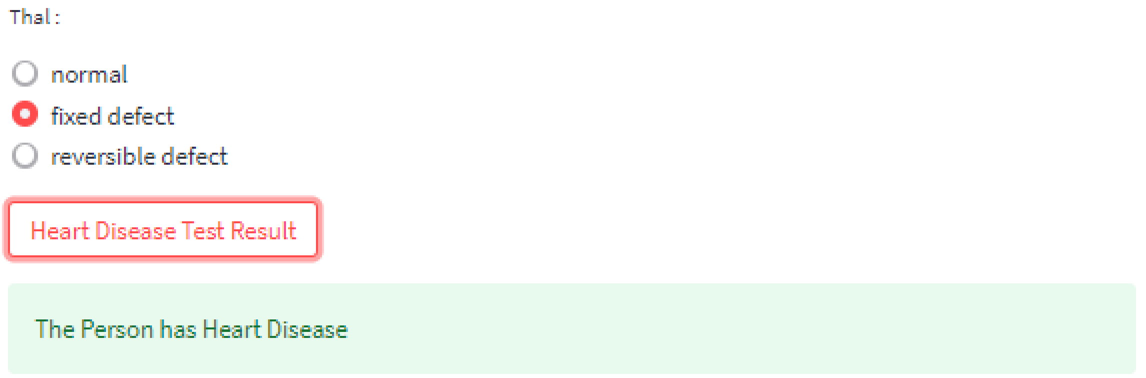
2

the slope of the peak exercise ST segment :

● Downsloping
○ Flat
○ upsloping

number of major vessels (0-3) colored by fluoroscopy

0



Thal :

○ normal
● fixed defect
○ reversible defect

The Person has Heart Disease

2.Breast cancer prediction:

Breast cancer prediction is using machine learning algorithms and statistical models to analyze large amounts of data and identify patterns that may be indicate whether the woman is suffering benign tumor(they do not invade nearby tissues or spread to other parts of the body.) or malignant tumor(they invade other sites).

Fine needle aspiration is a biopsy procedure that involves collecting a tissue sample for analysis. In this procedure, a thin needle is inserted into an area of abnormal-appearing tissue or body fluid. As with other types of biopsies, the sample collected during fine needle aspiration can help make a diagnosis or rule out conditions such as cancer.

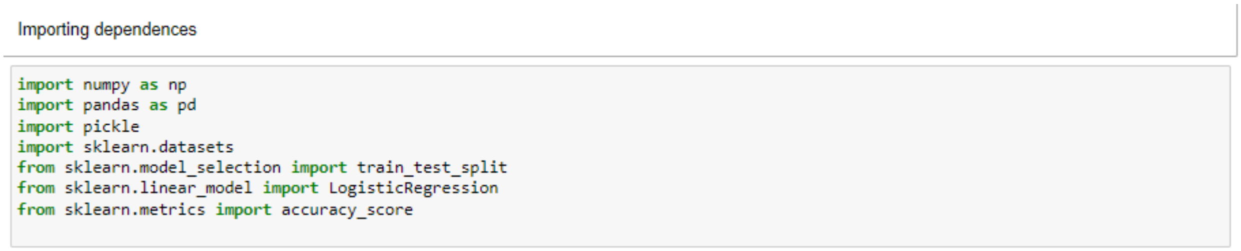
Logistic regression algorithm and training data are used to develop the breast cancer prediction model. Once model is developed , is used to predict whether breast cancer is benign tumor or malignant based on the newly collected cancer medical data.

2.1Breast Cancer Prediction Model :

Here is step by step procedure how breast cancer prediction model is developed.

2.1.1: Import required modules.

Import required modules into Jupyter Notebook that helps to develop breast cancer prediction model.

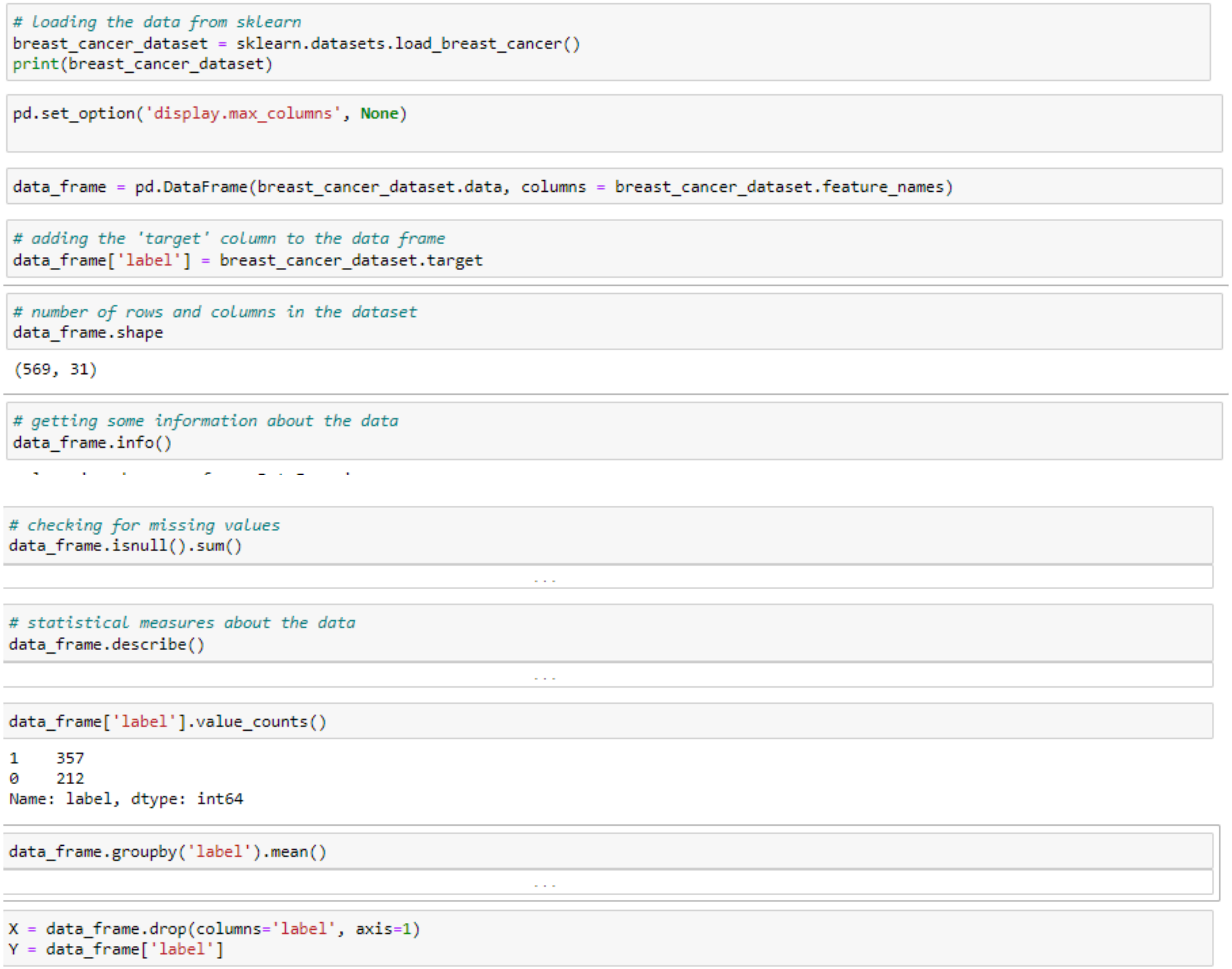


Importing dependences

```
import numpy as np
import pandas as pd
import pickle
import sklearn.datasets
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

2.1.2: Data collection and Processing.

Read collected dataset into Jupyter Notebook and Split all abnormal cell details of patient into one variable and all results of patient into another variable as processing data.



```
# loading the data from sklearn
breast_cancer_dataset = sklearn.datasets.load_breast_cancer()
print(breast_cancer_dataset)
```

```
pd.set_option('display.max_columns', None)
```

```
data_frame = pd.DataFrame(breast_cancer_dataset.data, columns = breast_cancer_dataset.feature_names)
```

```
# adding the 'target' column to the data frame
data_frame['label'] = breast_cancer_dataset.target
```

```
# number of rows and columns in the dataset
data_frame.shape
```

```
(569, 31)
```

```
# getting some information about the data
data_frame.info()
```

```
# checking for missing values
data_frame.isnull().sum()
```

```
...
```

```
# statistical measures about the data
data_frame.describe()
```

```
...
```

```
data_frame['label'].value_counts()
```

```
1    357
0    212
Name: label, dtype: int64
```

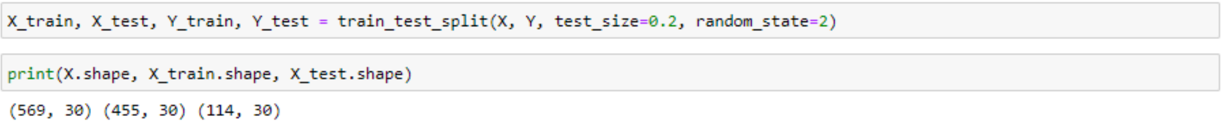
```
data_frame.groupby('label').mean()
```

```
...
```

```
X = data_frame.drop(columns='label', axis=1)
Y = data_frame['label']
```

2.1.3: Splitting the processed data into training data and test data.

Split the abnormal cell details and results of the patients into training data of the abnormal cell details, training data of the result, testing data of the abnormal cell details and testing data of the results using train_test_split method.



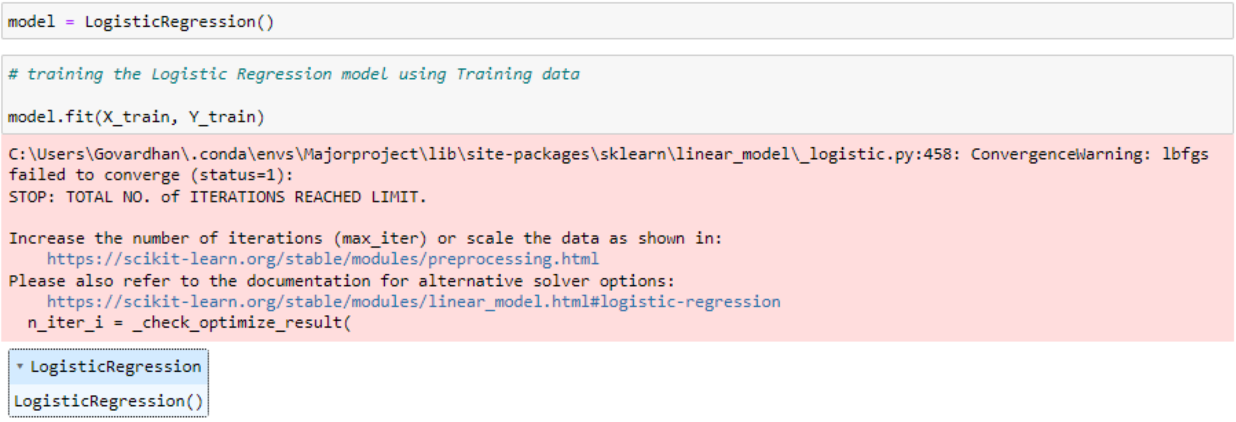
```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=2)
```

```
print(X.shape, X_train.shape, X_test.shape)
```

```
(569, 30) (455, 30) (114, 30)
```

2.1.4 : Train logistic regression model

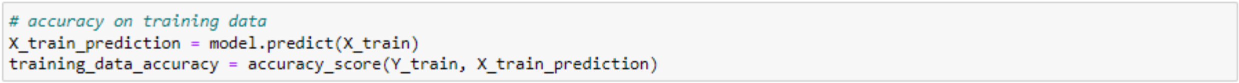
Training logistic regression with training data of abnormal cell's details and results by using model.fit() method to breast cancer prediction model.



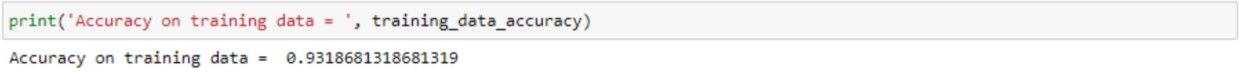
```
model = LogisticRegression()
```

```
# training the Logistic Regression model using Training data

model.fit(X_train, Y_train)
```

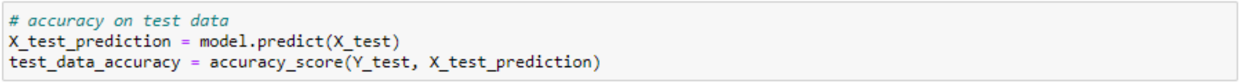
```
C:\Users\Govardhan\.conda\envs\Majorproject\lib\site-packages\sklearn\linear_model\_logistic.py:458: ConvergenceWarning: lbfgs
failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```



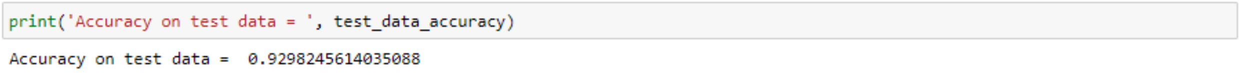
```
# accuracy on training data
X_train_prediction = model.predict(X_train)
training_data_accuracy = accuracy_score(Y_train, X_train_prediction)
```



```
print('Accuracy on training data = ', training_data_accuracy)

Accuracy on training data =  0.9318681318681319
```



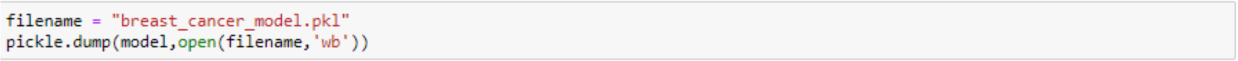
```
# accuracy on test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(Y_test, X_test_prediction)
```



```
print('Accuracy on test data = ', test_data_accuracy)

Accuracy on test data =  0.9298245614035088
```

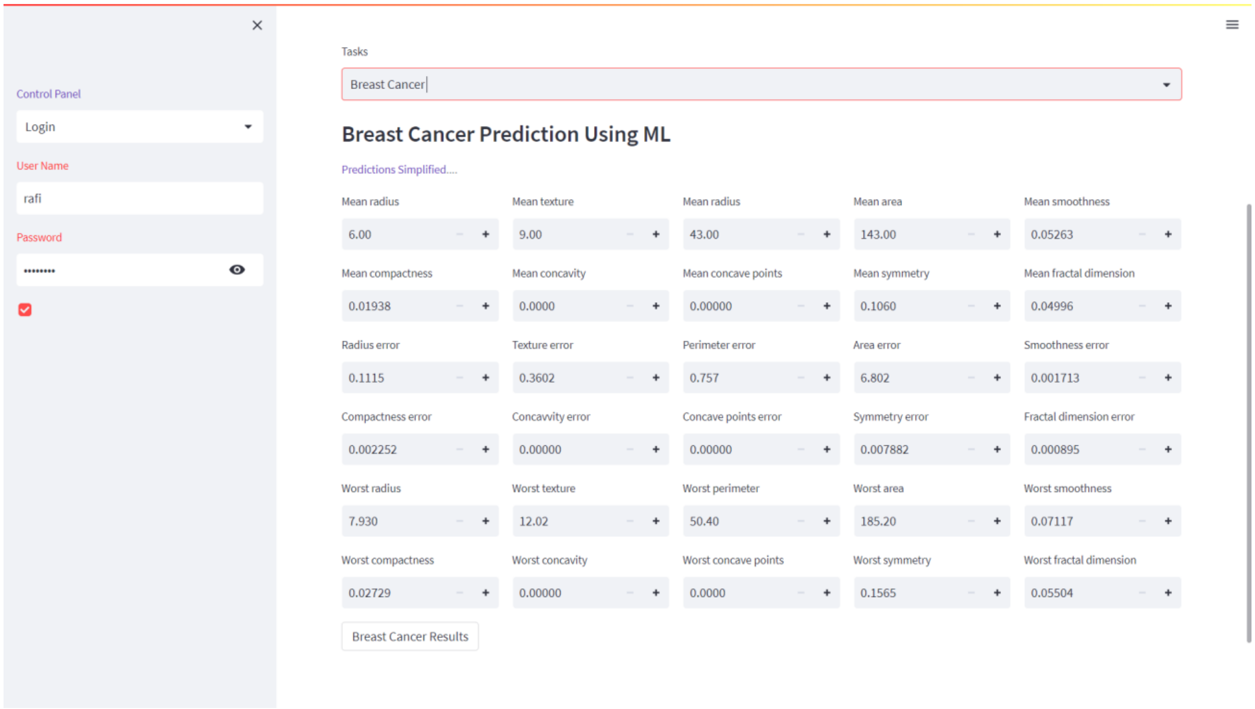
2.1.5 : Convert model into a picklefile.

Converting model into pickle module to integrate the model with python web application file.



```
filename = "breast_cancer_model.pkl"
pickle.dump(model,open(filename,'wb'))
```

Input and output:



Control Panel

Login

User Name

rafi

Password

••••••••

☑

Tasks

Breast Cancer

Breast Cancer Prediction Using ML

Predictions Simplified....

Mean radius	Mean texture	Mean radius	Mean area	Mean smoothness
6.00	9.00	43.00	143.00	0.05263
Mean compactness	Mean concavity	Mean concave points	Mean symmetry	Mean fractal dimension
0.01938	0.0000	0.00000	0.1060	0.04996
Radius error	Texture error	Perimeter error	Area error	Smoothness error
0.1115	0.3602	0.757	6.802	0.001713
Compactness error	Concavvity error	Concave points error	Symmetry error	Fractal dimension error
0.002252	0.00000	0.00000	0.007882	0.000895
Worst radius	Worst texture	Worst perimeter	Worst area	Worst smoothness
7.930	12.02	50.40	185.20	0.07117
Worst compactness	Worst concavity	Worst concave points	Worst symmetry	Worst fractal dimension
0.02729	0.00000	0.0000	0.1565	0.05504

3.Diabetes Prediction :

Diabetes prediction is using machine learning algorithms and statistical models to analyze large amounts of data and identify patterns that may be indicative of diabetes risk.

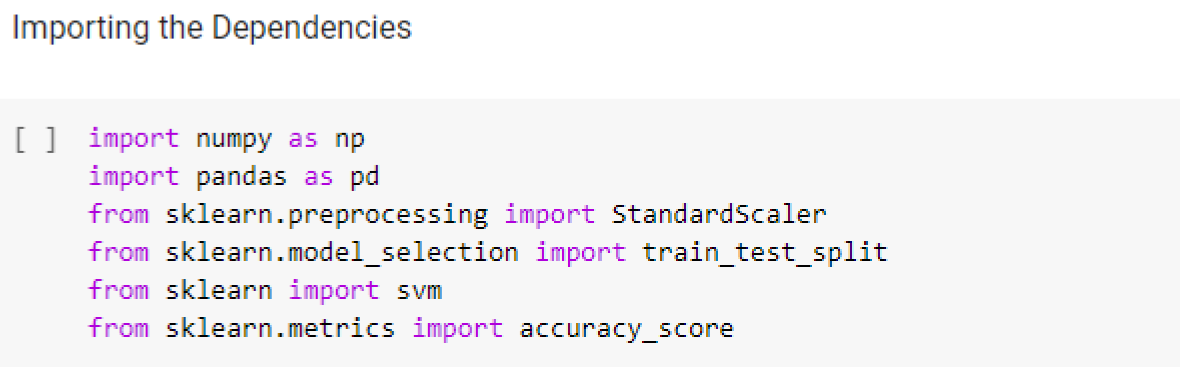
Support Vector Machine (SVM) algorithm and training data are used to develop diabetes prediction model. Once model is developed , it is used to predict whether person has diabetes or not based on the newly collected data.

Support Vector Machine (SVM) algorithm is a type of supervised machine learning algorithm used for classification and regression analysis.

3.1Diabetes Model :

Here is step by step procedure how diabetes prediction model is developed.

3.1.1 :Import required modules.



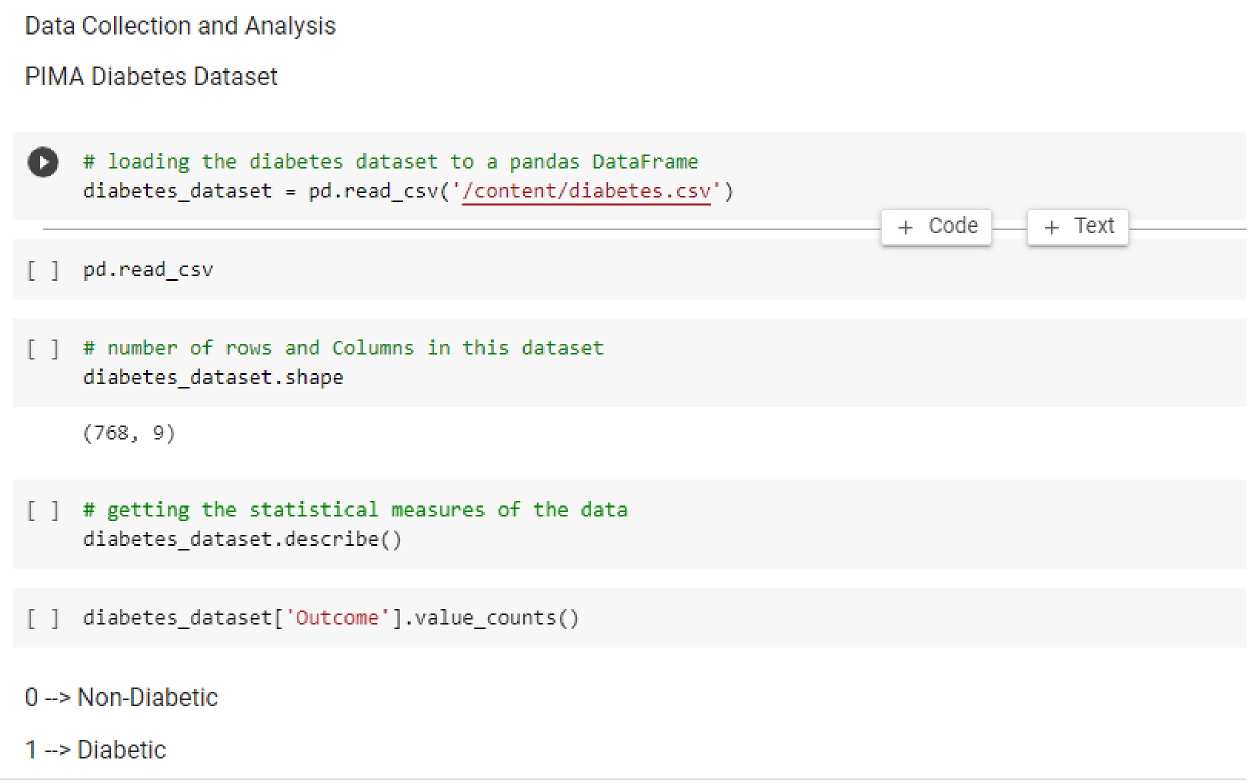
Importing the Dependencies

```
[ ]  import numpy as np
     import pandas as pd
     from sklearn.preprocessing import StandardScaler
     from sklearn.model_selection import train_test_split
     from sklearn import svm
     from sklearn.metrics import accuracy_score
```

Import required modules into Jupyter Notebook that helps to develop diabetes prediction model.

3.1.2 :Data collection and Processing

Read the collected dataset into Jupyter Notebook and Split all attributes of diabetes into one variable and all results of patient into another variable as processing data.



Data Collection and Analysis

PIMA Diabetes Dataset

```
# loading the diabetes dataset to a pandas DataFrame
diabetes_dataset = pd.read_csv('/content/diabetes.csv')
```

```
pd.read_csv
```

```
# number of rows and Columns in this dataset
diabetes_dataset.shape
```

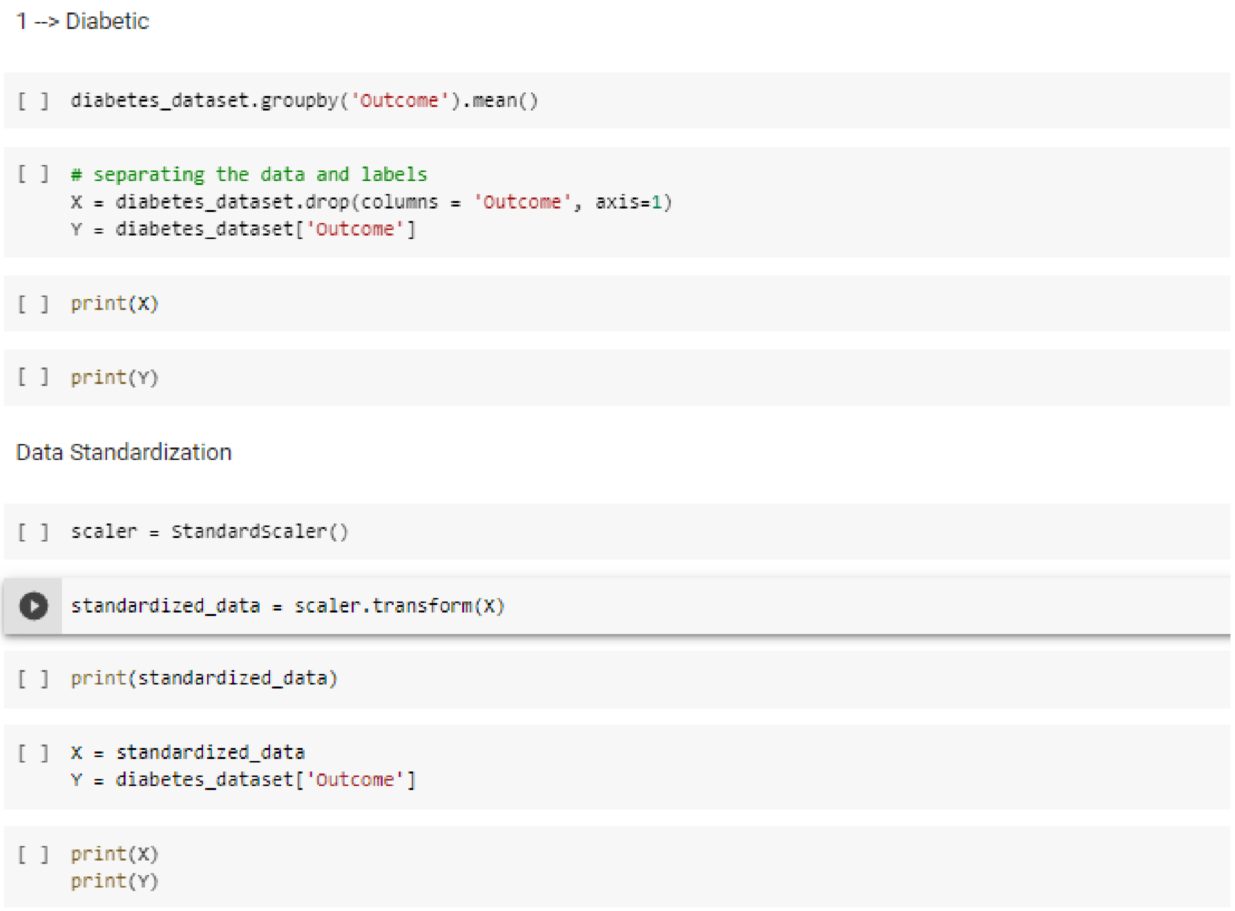
```
(768, 9)
```

```
# getting the statistical measures of the data
diabetes_dataset.describe()
```

```
diabetes_dataset['Outcome'].value_counts()
```

0 --> Non-Diabetic

1 --> Diabetic



1 --> Diabetic

```
diabetes_dataset.groupby('Outcome').mean()
```

```
# separating the data and labels
X = diabetes_dataset.drop(columns = 'Outcome', axis=1)
Y = diabetes_dataset['Outcome']
```

```
print(X)
```

```
print(Y)
```

Data Standardization

```
scaler = StandardScaler()
```

```
standardized_data = scaler.transform(X)
```

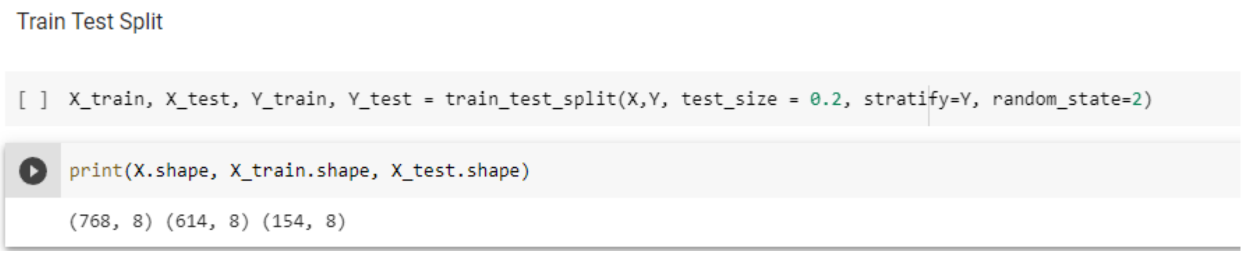
```
print(standardized_data)
```

```
X = standardized_data
Y = diabetes_dataset['Outcome']
```

```
print(X)
print(Y)
```

3.1.3:Splitting the processed data into training data and test data.

Split the attributes and results of the patients into training data of the attributes , training data of the result, testing data of the attributes and testing data of the results using train_test_split method.



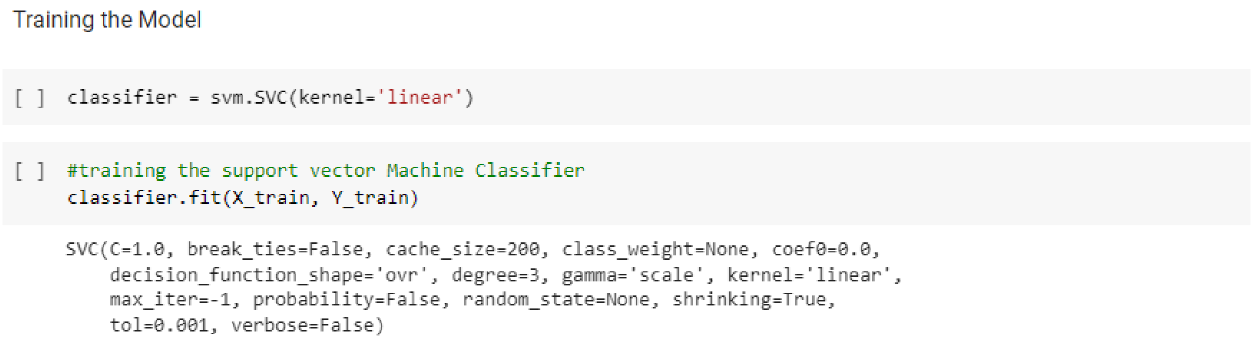
Train Test Split

```
[ ]  X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.2, stratify=Y, random_state=2)
```

```
print(X.shape, X_train.shape, X_test.shape)

(768, 8) (614, 8) (154, 8)
```

3.1.4 : Train Support Vector Machine algorithm

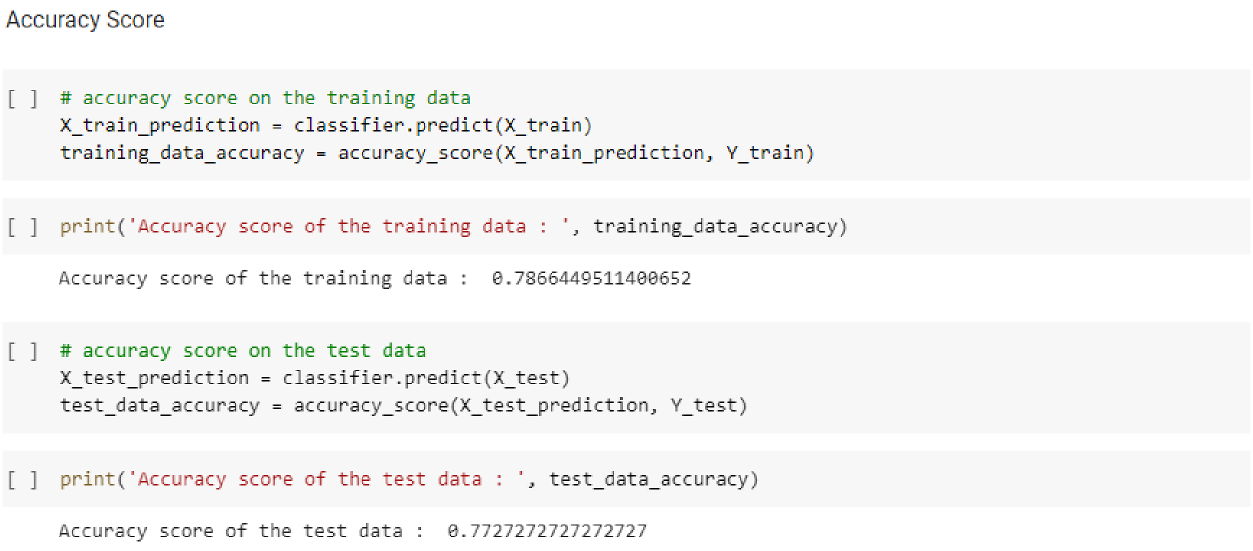
Train Support Vector Machine with training data of attributes and results by using model.fit() method to build diabetes prediction model.



Training the Model

```
[ ]  classifier = svm.SVC(kernel='linear')
```

```
[ ]  #training the support vector Machine Classifier
     classifier.fit(X_train, Y_train)

     SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
         decision_function_shape='ovr', degree=3, gamma='scale', kernel='linear',
         max_iter=-1, probability=False, random_state=None, shrinking=True,
         tol=0.001, verbose=False)
```



Accuracy Score

```
[ ]  # accuracy score on the training data
     X_train_prediction = classifier.predict(X_train)
     training_data_accuracy = accuracy_score(X_train_prediction, Y_train)
```

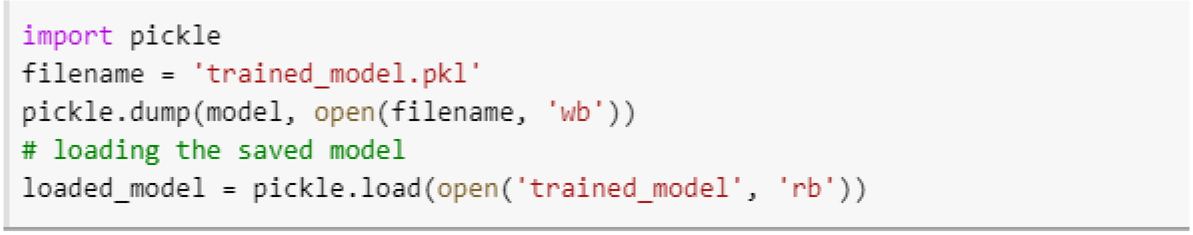
```
[ ]  print('Accuracy score of the training data : ', training_data_accuracy)

     Accuracy score of the training data :  0.7866449511400652
```

```
[ ]  # accuracy score on the test data
     X_test_prediction = classifier.predict(X_test)
     test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
```

```
[ ]  print('Accuracy score of the test data : ', test_data_accuracy)

     Accuracy score of the test data :  0.7727272727272727
```

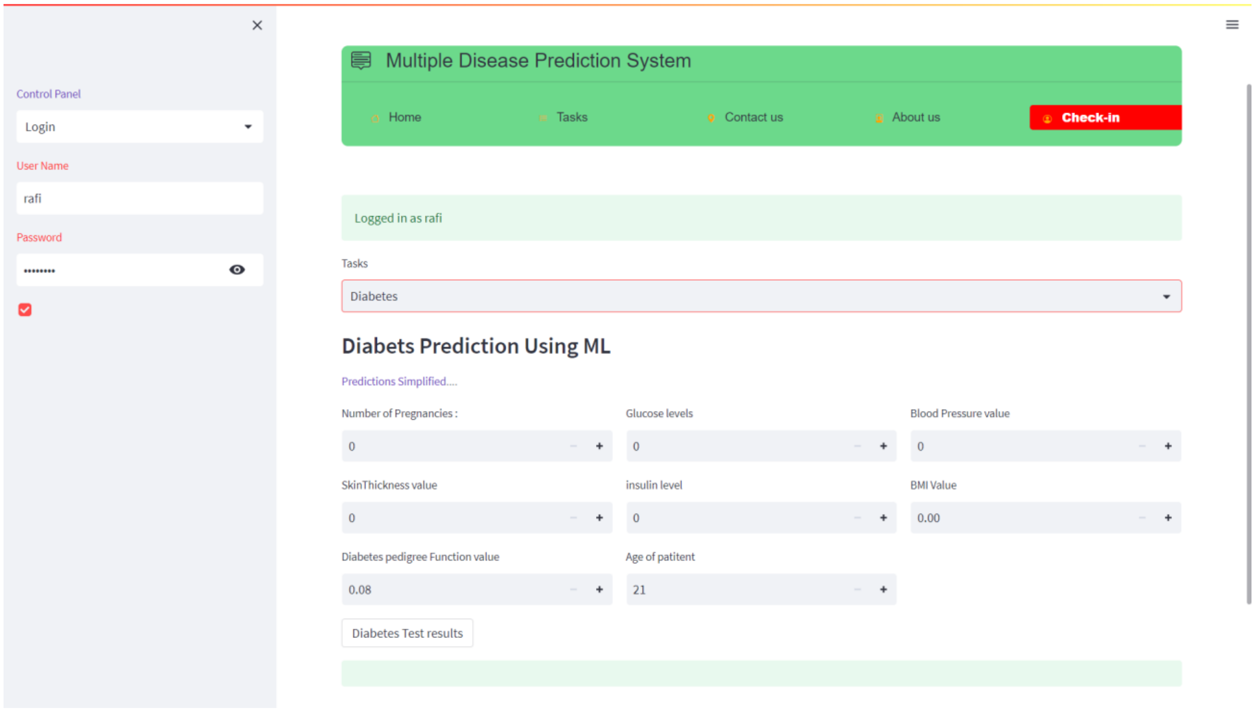
3.1.5 : Convert the model into a picklefile.

Converting model into pickle module to integrate the model with python web application file.



```
import pickle
filename = 'trained_model.pkl'
pickle.dump(model, open(filename, 'wb'))
# loading the saved model
loaded_model = pickle.load(open('trained_model', 'rb'))
```

INPUT AND OUTPUT:



Control Panel

Login

User Name

rafi

Password

••••••••

Logged in as rafi

Tasks

Diabetes

Diabets Prediction Using ML

Predictions Simplified....

Number of Pregnancies : 0

Glucose levels 0

Blood Pressure value 0

SkinThickness value 0

insulin level 0

BMI Value 0.00

Diabetes pedigree Function value 0.08

Age of patitent 21

4 Covid Prediction:

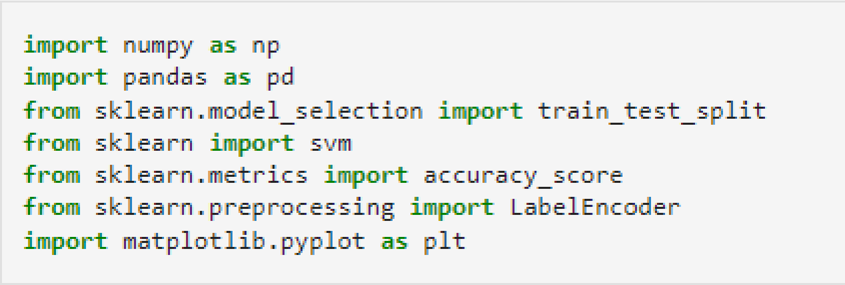
Covid prediction is a machine learning model that can accurately predict whether a person has effected with covid or not by considering whether person could effected with any diseases such as diabetes, cardio vascular diseases,sickle_cell_diesases,immune deficiency disease, down syndrome, cancer, chronic respiratory disease, hypertension vaccinated in the past or present.

4.1Covid Prediction Model :

Here is step by step procedure how covid predication model is developed

4.1.1: Import required modules

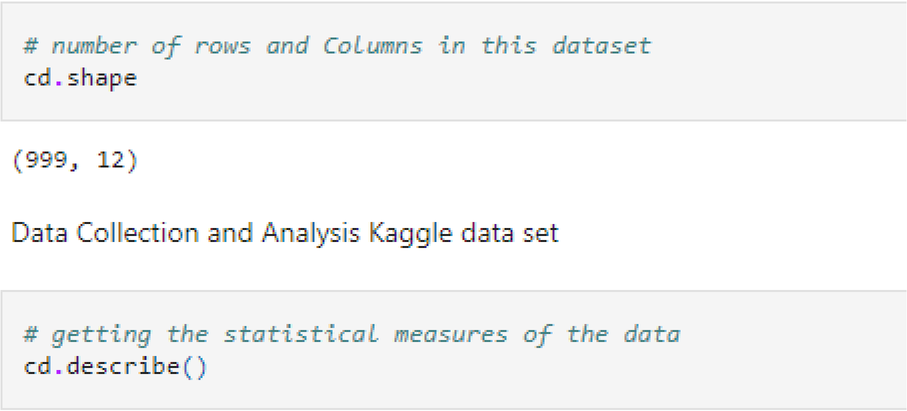
Import required modules into Jupyter Notebook that helps to develop covid prediction model



```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import LabelEncoder
import matplotlib.pyplot as plt
```

4.1.2: Data collection and Processing

Read collected dataset into Jupyter Notebook and Split all attribute diseases of covid into one variable and all results of patient into another variable as processing data

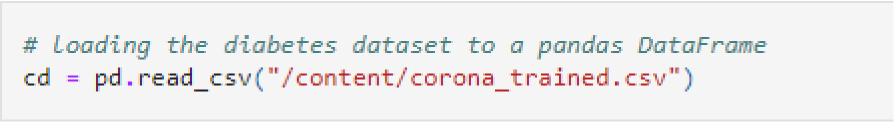


```
# number of rows and Columns in this dataset
cd.shape
```

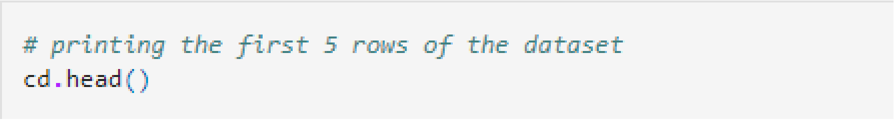
```
(999, 12)
```

Data Collection and Analysis Kaggle data set

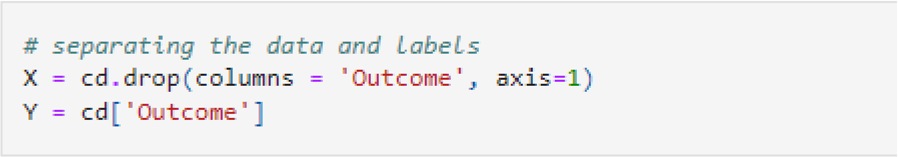
```
# getting the statistical measures of the data
cd.describe()
```



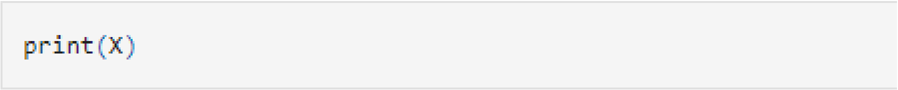
```
# Loading the diabetes dataset to a pandas DataFrame
cd = pd.read_csv("/content/corona_trained.csv")
```



```
# printing the first 5 rows of the dataset
cd.head()
```



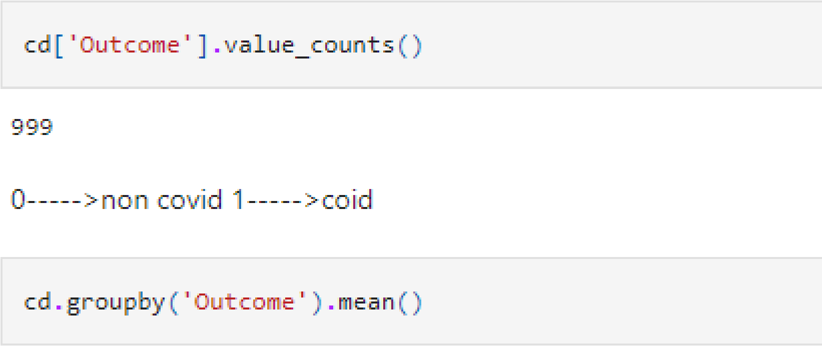
```
# separating the data and labels
X = cd.drop(columns = 'Outcome', axis=1)
Y = cd['Outcome']
```



```
print(X)
```

4.1.3: Splitting the processed data into training data and test data

Split the attributes and results of the patients into training data of the attributes , training data of the result, testing data of the attributes and testing data of the results using train_test_split method

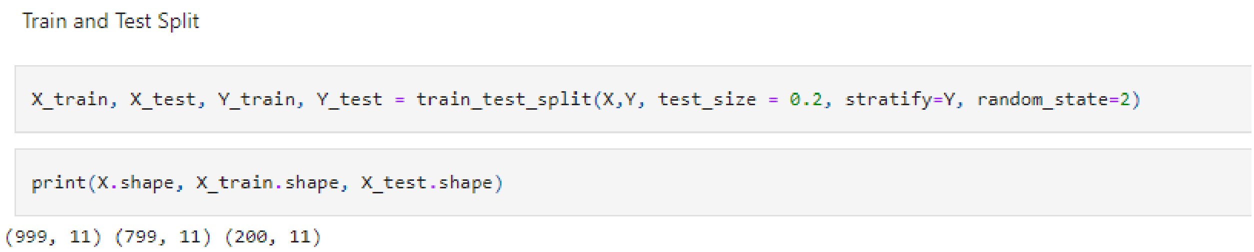


```
cd['Outcome'].value_counts()
```

```
999
```

0----->non covid 1----->coid

```
cd.groupby('Outcome').mean()
```



Train and Test Split

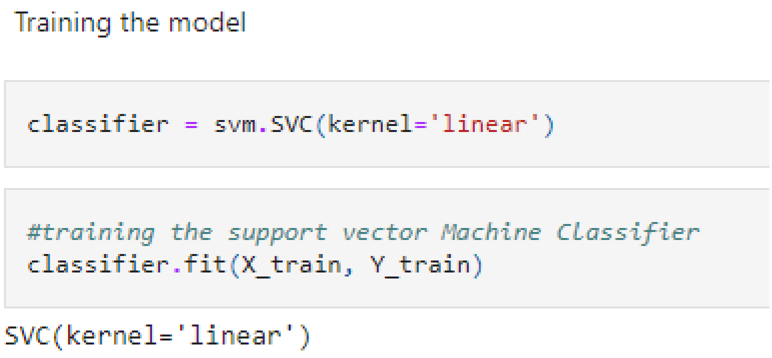
```
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.2, stratify=Y, random_state=2)
```

```
print(X.shape, X_train.shape, X_test.shape)
```

```
(999, 11) (799, 11) (200, 11)
```

4.1.4: Train Support Vector Machine (SVM)

Train Support Vector Machine (SVM) with training data of attributes and results by using model.fit() method to build covid prediction model

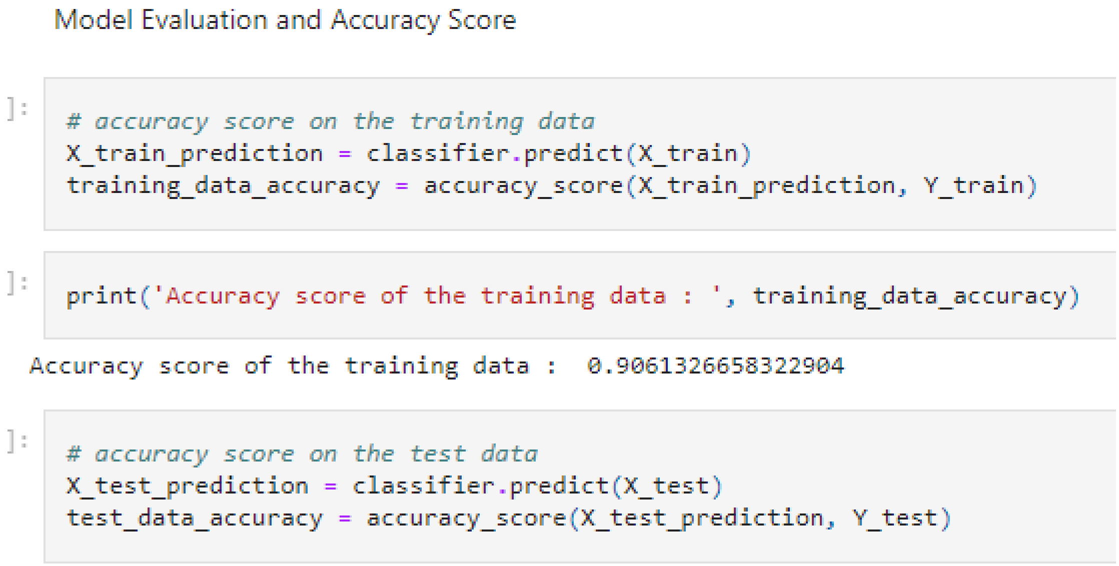


Training the model

```
classifier = svm.SVC(kernel='linear')
```

```
#training the support vector Machine Classifier
classifier.fit(X_train, Y_train)
```

```
SVC(kernel='linear')
```



Model Evaluation and Accuracy Score

```
# accuracy score on the training data
X_train_prediction = classifier.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)
```

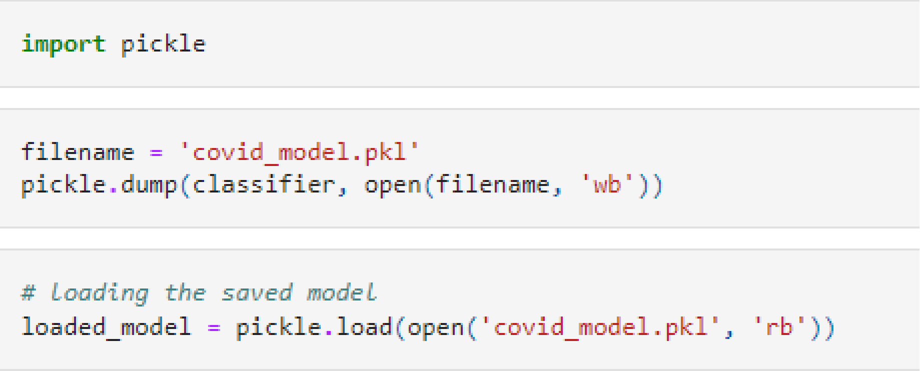
```
print('Accuracy score of the training data : ', training_data_accuracy)
```

```
Accuracy score of the training data :  0.9061326658322904
```

```
# accuracy score on the test data
X_test_prediction = classifier.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
```

4.1.5: Convert model into a pickle file

Converting model into pickle file to integrate the model with python web application file

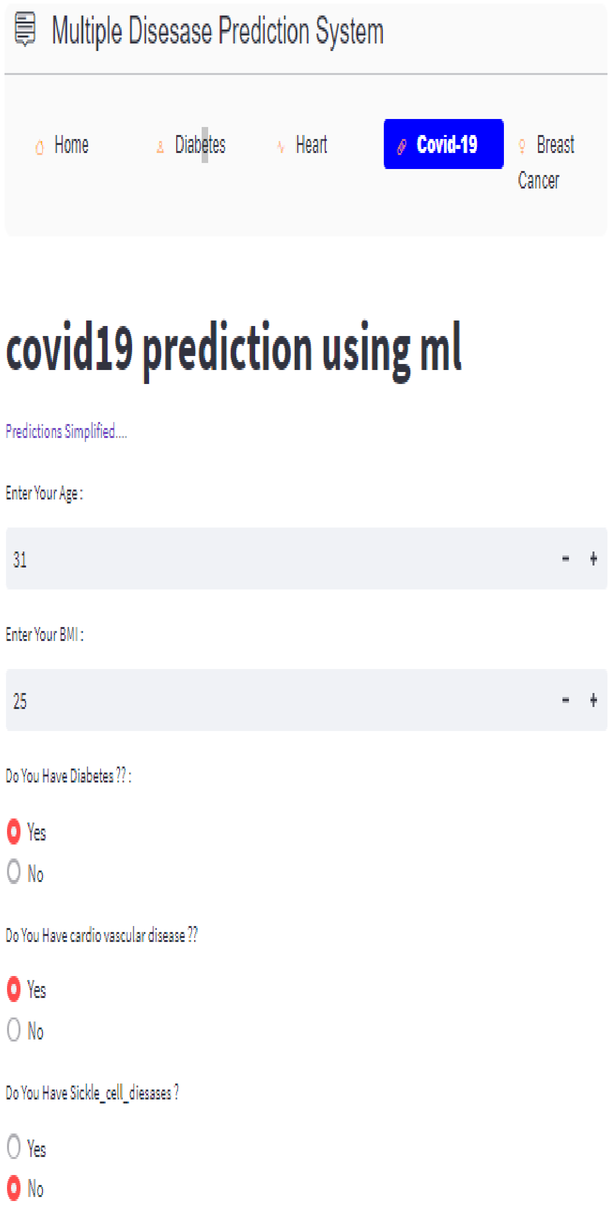


```
import pickle
```

```
filename = 'covid_model.pkl'
pickle.dump(classifier, open(filename, 'wb'))
```

```
# Loading the saved model
loaded_model = pickle.load(open('covid_model.pkl', 'rb'))
```

INPUT AND OUTPUT:



covid19 prediction using ml

Predictions Simplified....

Enter Your Age :

31

Enter Your BMI :

25

Do You Have Diabetes ?? :

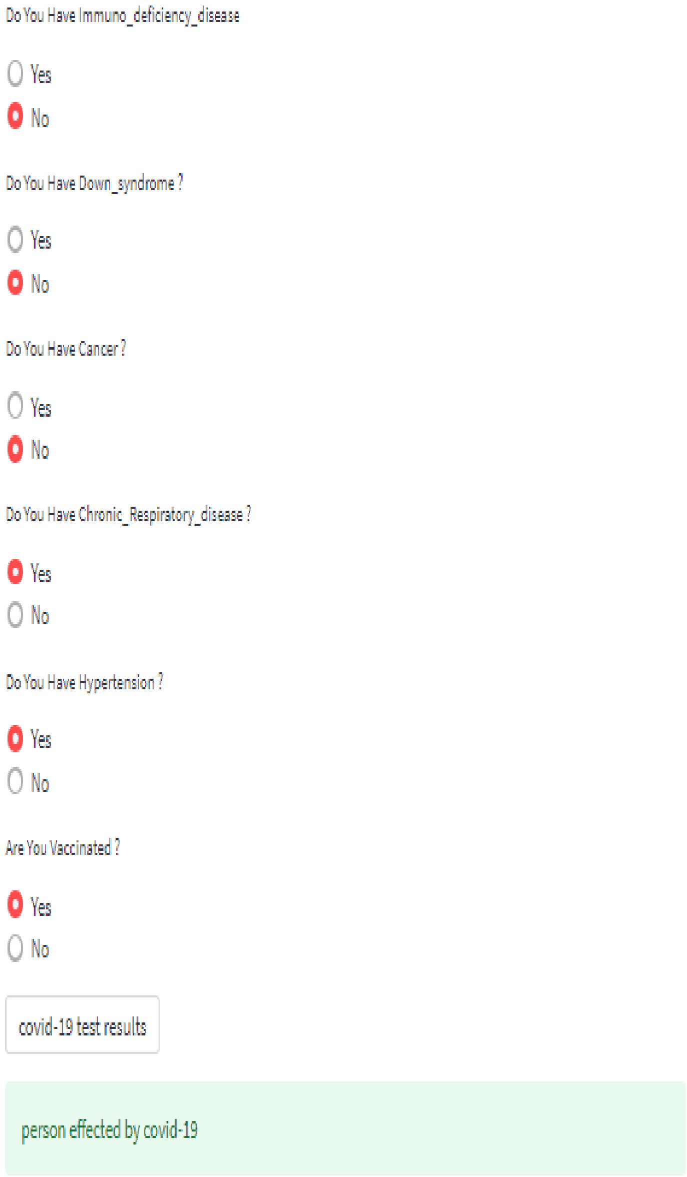
● Yes
○ No

Do You Have cardio vascular disease ??

● Yes
○ No

Do You Have Sickle_cell_diesases ?

○ Yes
● No



Do You Have Immuno_deficiency_disease

○ Yes
● No

Do You Have Down_syndrome ?

○ Yes
● No

Do You Have Cancer ?

○ Yes
● No

Do You Have Chronic_Respiratory_disease ?

● Yes
○ No

Do You Have Hypertension ?

● Yes
○ No

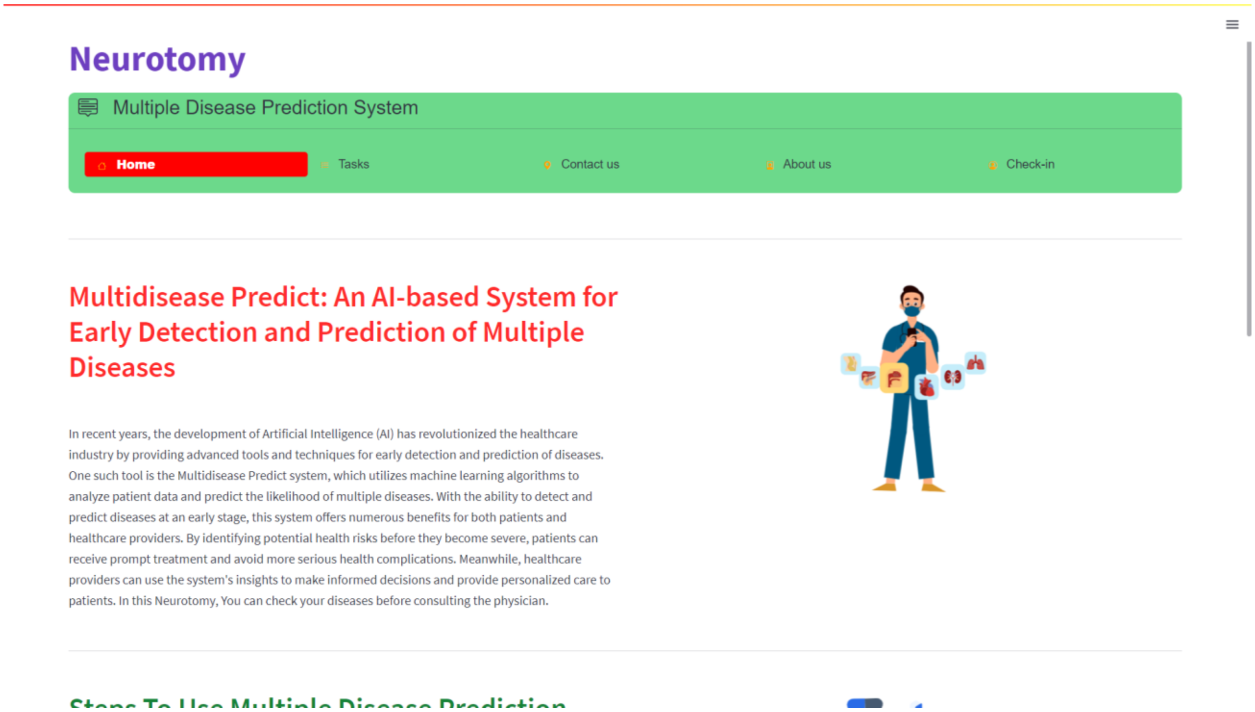
Are You Vaccinated ?

● Yes
○ No

person effected by covid-19

Interface

Home Page



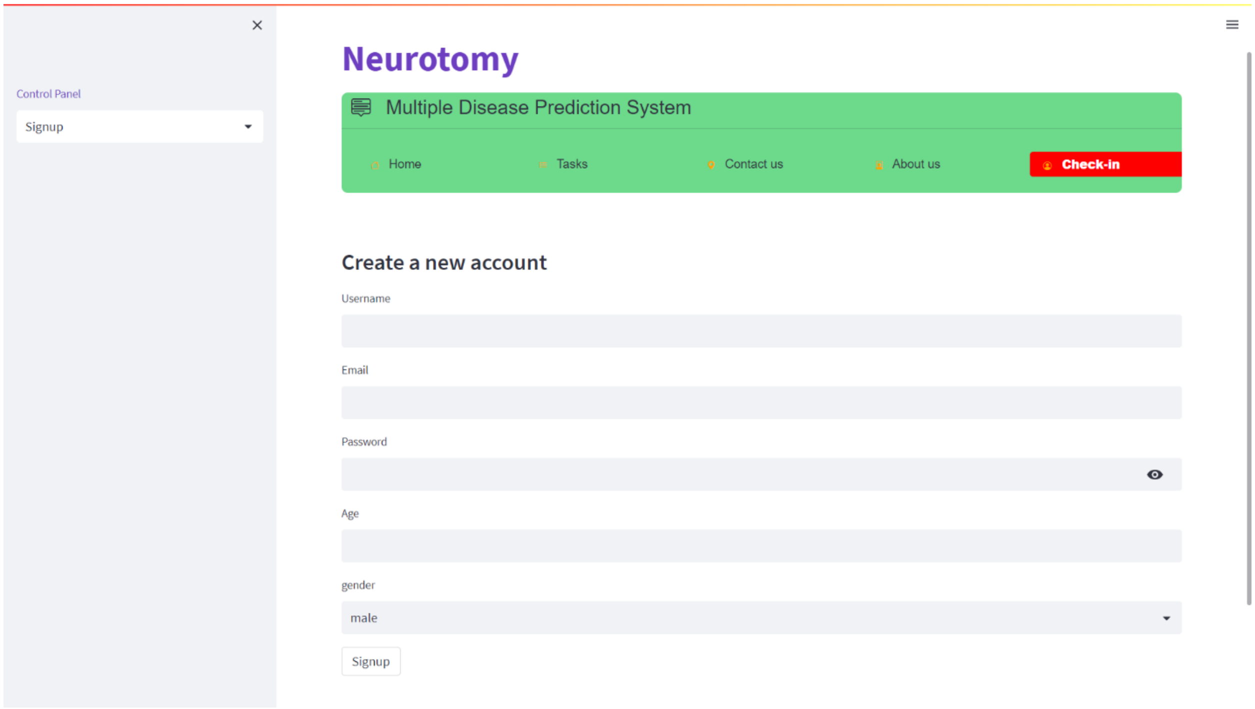
Multidisease Predict: An AI-based System for Early Detection and Prediction of Multiple Diseases

In recent years, the development of Artificial Intelligence (AI) has revolutionized the healthcare industry by providing advanced tools and techniques for early detection and prediction of diseases. One such tool is the Multidisease Predict system, which utilizes machine learning algorithms to analyze patient data and predict the likelihood of multiple diseases. With the ability to detect and predict diseases at an early stage, this system offers numerous benefits for both patients and healthcare providers. By identifying potential health risks before they become severe, patients can receive prompt treatment and avoid more serious health complications. Meanwhile, healthcare providers can use the system's insights to make informed decisions and provide personalized care to patients. In this Neurotomy, You can check your diseases before consulting the physician.

Steps To Use Multiple Disease Prediction

Check in page:

✓ Sign-Up Page:



Create a new account

Username

Email

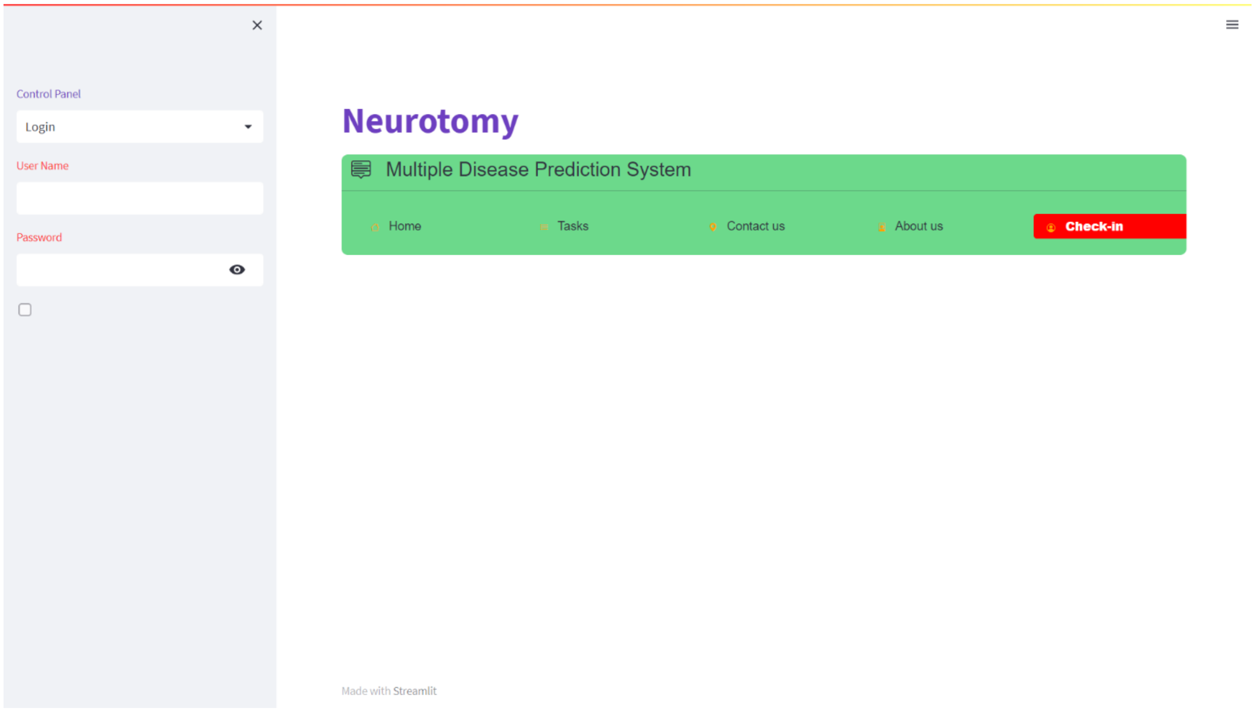
Password

Age

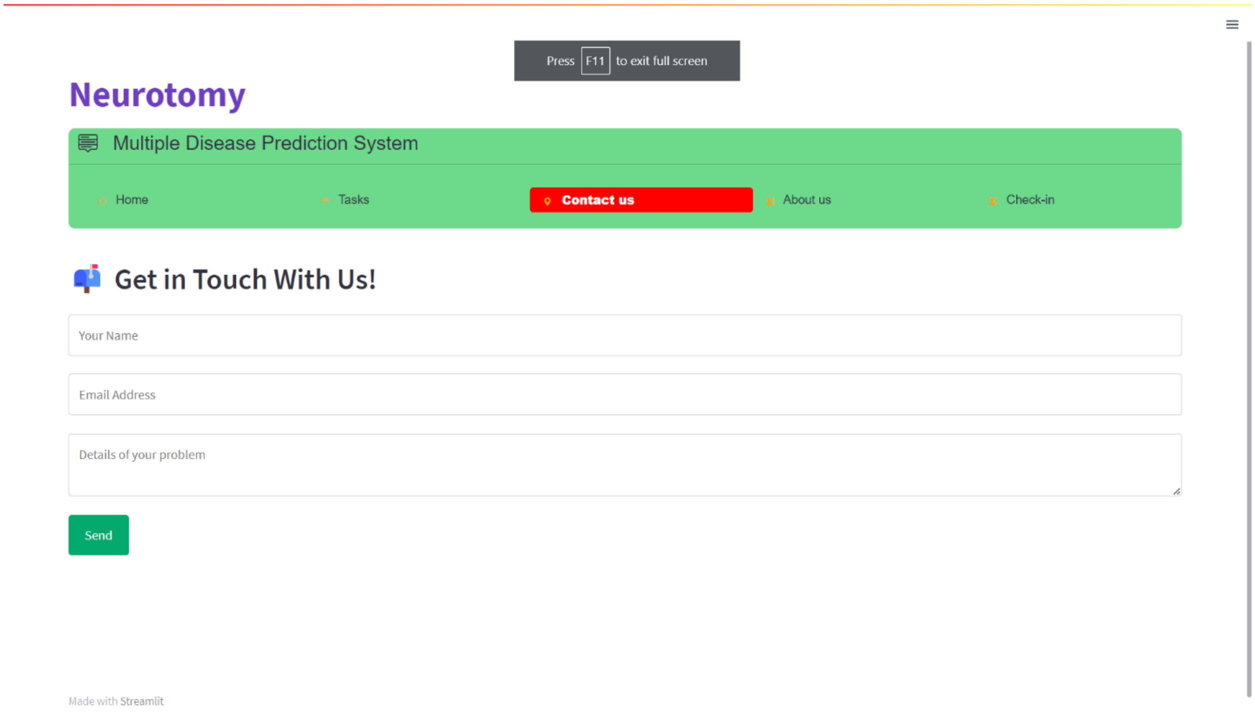
gender

male

✓ Login Page



Contact-Page:



CONCLUSION

In conclusion, this project has demonstrated the feasibility and effectiveness of a machine learning-based multiple disease prediction system. The system can accurately predict the likelihood of four major diseases: heart disease, breast cancer, diabetes, and COVID-19.

The system's high accuracy and sensitivity make it a potentially valuable tool for healthcare professionals in early diagnosis and intervention, which can improve patient outcomes and reduce healthcare costs.

The project's success is due to the use of machine learning algorithms, which are capable of identifying patterns and relationships in large and complex datasets.

The system leverages several features and risk factors that are commonly associated with each disease, such as age, gender, blood pressure, cholesterol levels, family history, mammogram results, glucose levels, and COVID-19 symptoms.

The project's limitations include the availability and quality of the dataset, which can affect the performance of the machine learning models. Additionally, the models may not generalize well to other populations or settings, and further validation and refinement are needed to ensure their reliability and generalizability. In summary, the multiple disease prediction system presented in this project has the potential to assist healthcare professionals in early disease diagnosis and intervention, which can improve patient outcomes and reduce healthcare costs.

Future research should focus on improving the system's accuracy, robustness, and scalability, and on integrating it with existing healthcare systems and practices.

REFERENCE

- https://www.researchgate.net/publication/353651472_Multiple_disease_prediction_using_Machine_learning_algorithms
- https://www.geeksforgeeks.org/ml-heart-disease-prediction-using-logistic-regression
- https://towardsdatascience.com/building-a-simple-machine-learning-model-on-breast-cancer-data-eca4b3b99fa3
- https://www.analyticsvidhya.com/blog/2022/01/diabetes-prediction-using-machine-learning/
- https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset
- https://www.kaggle.com/datasets/merishnasuwal/breast-cancer-prediction-dataset
- https://www.kaggle.com/mathchi/diabetes-data-set
- https://www.kaggle.com/code/athirags/corona-risk-prediction
- https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm
- https://www.javatpoint.com/logistic-regression-in-machine-learning