**Polycystic Ovary Syndrome Classification Using Machine Learning**

In a world where women's health is of paramount importance, the Proactive Health project envisions a

future where machine learning plays a pivotal role in identifying PCOS risk factors before clinical

symptoms emerge. Imagine a scenario where individuals can undergo routine health assessments, and

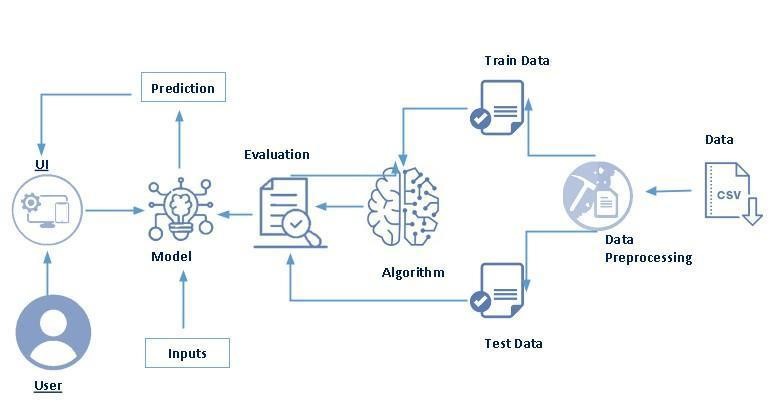
the predictive model provides timely insights, enabling healthcare professionals to take preventive

measures and provide personalized treatment plans. This proactive approach has the potential to

significantly improve the overall well-being of women and reduce the long-term health implications

associated with PCOS.

**Technical Architecture:**



**Project Flow:**

* User interacts with the UI to enter the input.
* Entered input is analysed by the model which is integrated.
* Once model analyses the input the prediction is showcased on the UI

To accomplish this, we have to complete all the activities listed below,

* Define Problem / Problem Understanding
  1. Specify the business problem

○ Business requirements ○ Literature Survey

○ Social or Business Impact.

● Data Collection & Preparation

○ Collect the dataset

○ Data Preparation

* Exploratory Data Analysis
  1. Descriptive statistical

○ Visual Analysis

* Model Building
  1. Training the model in multiple algorithms

○ Testing the model

* Performance Testing & Hyperparameter Tuning
  1. Testing model with multiple evaluation metrics

○ Comparing model accuracy before & after applying hyperparameter tuning

* Model Deployment
  1. Save the best model

○ Integrate with Web Framework

* Project Demonstration & Documentation
  1. Record explanation Video for project end to end solution

○ Project Documentation-Step by step project development procedure

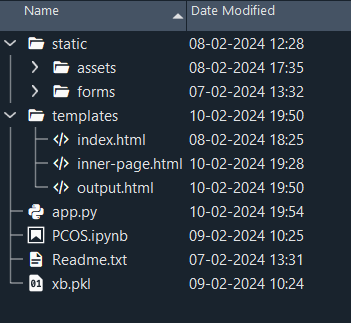
**Prior Knowledge:**

You must have prior knowledge of following topics to complete this project.

* ML Concepts
* Supervised learning: <https://www.javatpoint.com/supervised-machine-learning>
* Unsupervised learning: <https://www.javatpoint.com/unsupervised-machine-learning>
* Decision tree: [https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm](https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm%20)
* Random forest: <https://www.javatpoint.com/machine-learning-random-forest-algorithm>
* Logistic Regression: <https://www.javatpoint.com/logistic-regression-in-machine-learning>
* Xg boost: <https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/>
* Evaluation metrics: <https://www.analyticsvidhya.com/blog/2021/07/metrics-to-evaluate-your-classification-model-to-take-the-right-decisions/>
* Flask Basics : [https://www.youtube.com/watch?v=lj4I\_CvBnt0](https://www.youtube.com/watch?v=lj4I_CvBnt0%20)

**Project Structure:**

Create the Project folder which contains files as shown below



* We are building a flask application which needs HTML pages stored in the templates folder and a python script app.py for scripting.
* xb.pkl is our saved model. Further we will use this model for flask integration.
* Training folder contains a model training file.

# Milestone 1: Data Collection & Preparation

ML depends heavily on data. It is the most crucial aspect that makes algorithm training possible. So, this section allows you to download the required dataset.

## Activity 1: Collect the dataset

There are many popular open sources for collecting the data. Eg: kaggle.com, UCI repository, etc.

In this project we have used .csv data. This data is downloaded from kaggle.com. Please refer to the link given below to download the dataset.

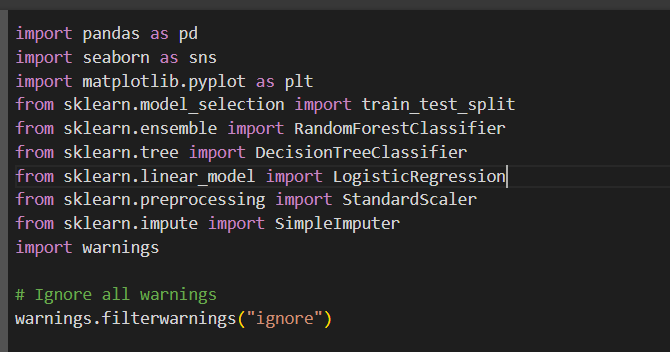
Link: <https://www.kaggle.com/datasets/prasoonkottarathil/polycystic-ovary-syndrome-pcos>

As the dataset is downloaded. Let us read and understand the data properly with the help of some visualisation techniques and some analysing techniques.

**Note:** There are a number of techniques for understanding the data. But here we have used some of it. In an additional way, you can use multiple techniques.

**Activity 1.1: Importing the libraries**

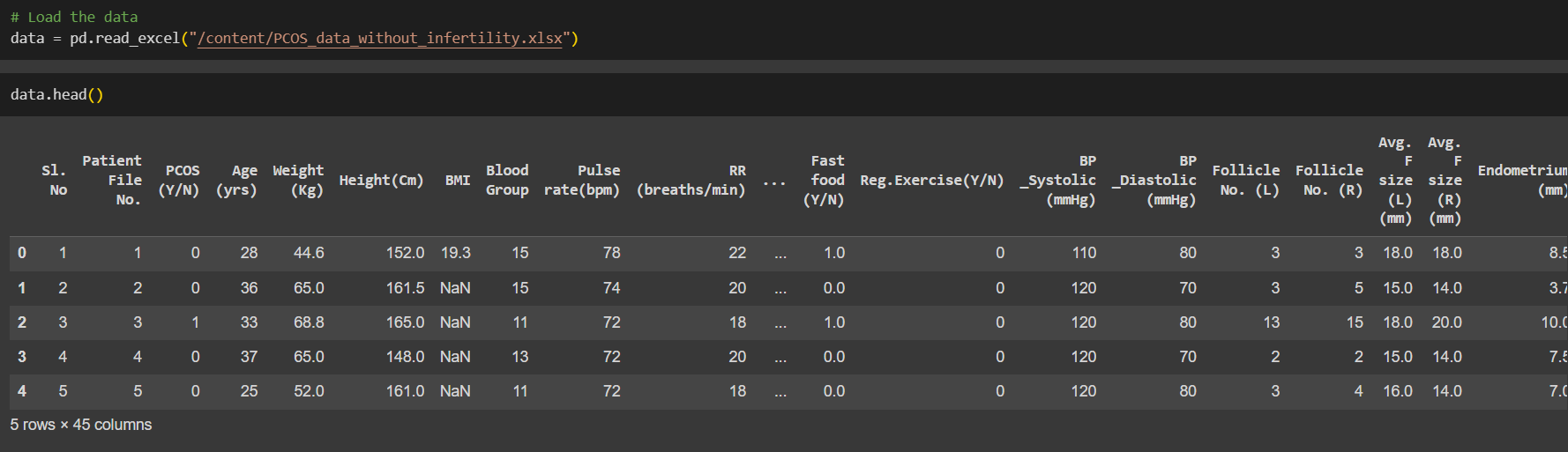
Import the necessary libraries as shown in the image.



## Activity 1.2: Read the Dataset

Our dataset format might be in .csv, excel files, .txt, .json, etc. We can read the dataset with the help of pandas.

In pandas we have a function called read\_csv() to read the dataset. As a parameter we have to give the directory of the csv file.



## Activity 2: Data Preparation

As we have understood how the data is, let's pre-process the collected data.

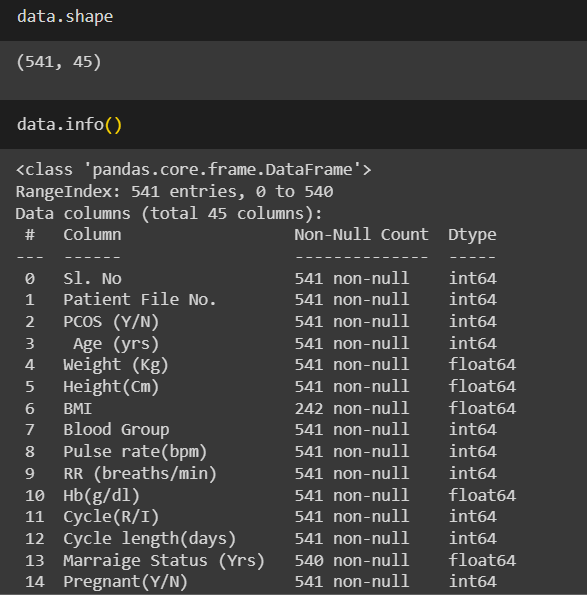
The download data set is not suitable for training the machine learning model as it might have so much randomness so we need to clean the dataset properly in order to fetch good results. This activity includes the following steps.

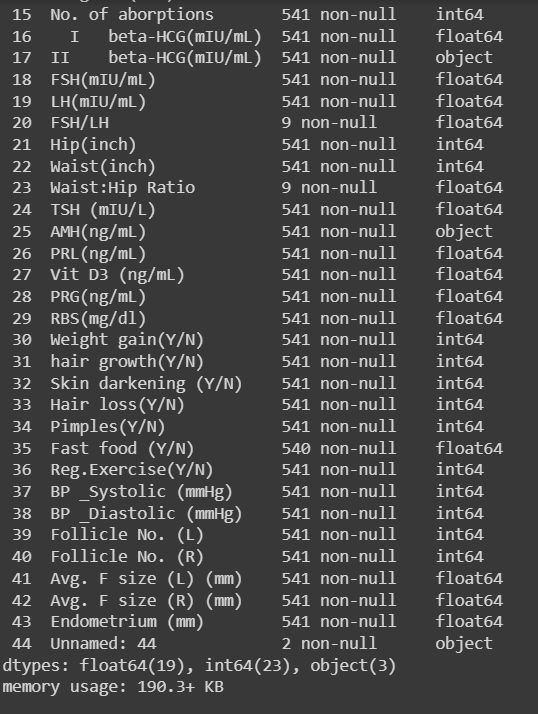
* Handling missing values
* Handling with unwanted columns
* Handling with important features
* Handling Outliers

Note: These are the general steps of pre-processing the data before using it for machine learning. Depending on the condition of your dataset, you may or may not have to go through all these steps.

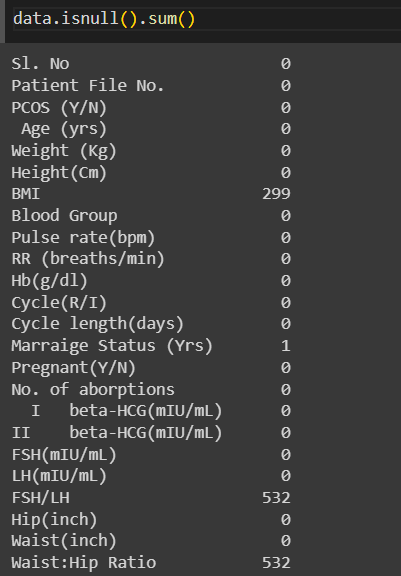
## Activity 2.1: Handling missing values

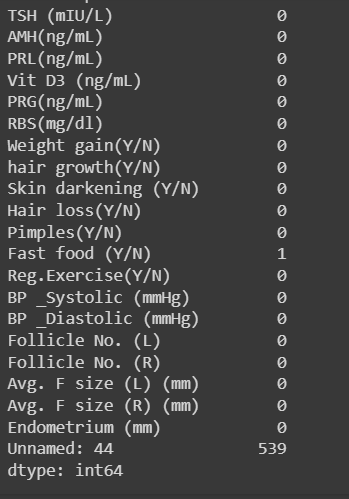
* Let’s find the shape of our dataset first. To find the shape of our data, the data.shape method is used. To find the data type, data.info() function is used.

****

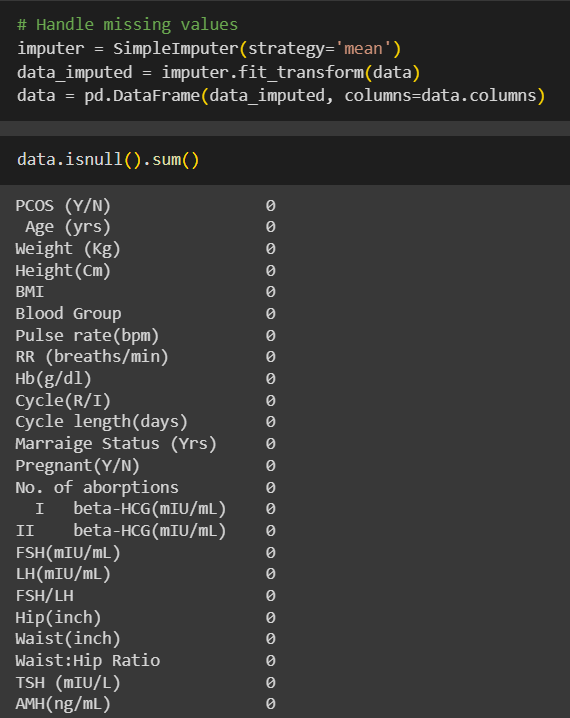


* For checking the null values, data.isnull() function is used. To sum those null values we use .sum() function. From the below image we found that there are no null values present in our dataset. So we can skip handling the missing values step.



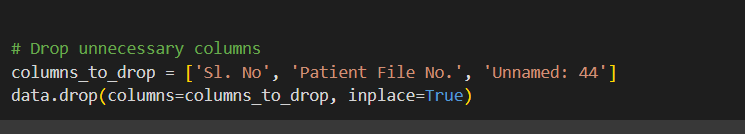


By using the Simple Imputer we have the replaced the null values with mean values for each and every columns that have the null values.



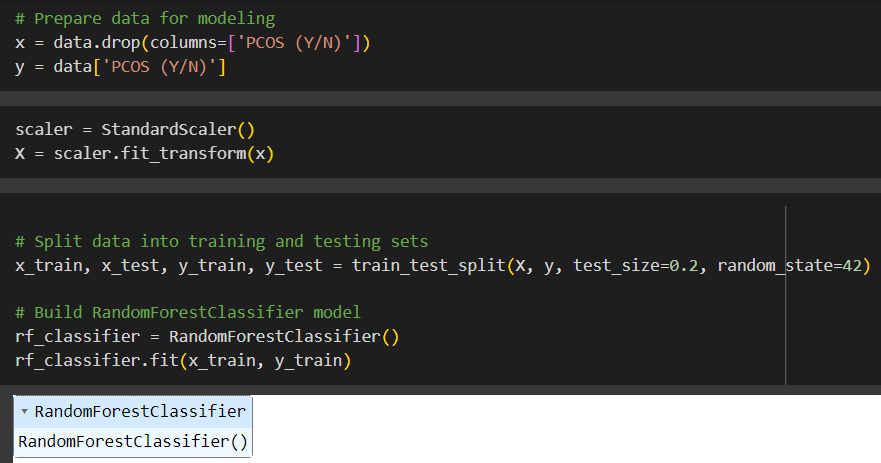
## Activity 2.2:Handling with Unwanted Columns Data

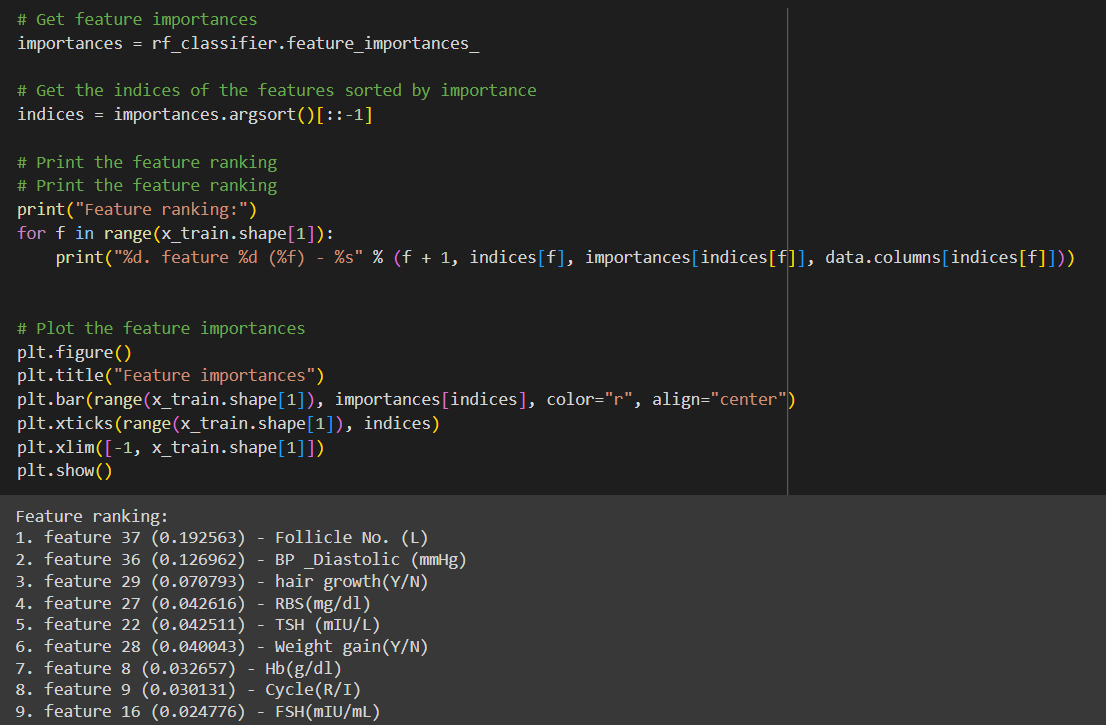
From the data we need to remove the columns as they are unwanted columns.



## Activity 2.3:Handling with the Important Features

Feature Importance: It refers to techniques that calculate a score for all the input features for a given model. The scores represent the “importance” of each feature. A higher score means that the specific feature will have a larger effect on the model that is being used to predict a certain variable.



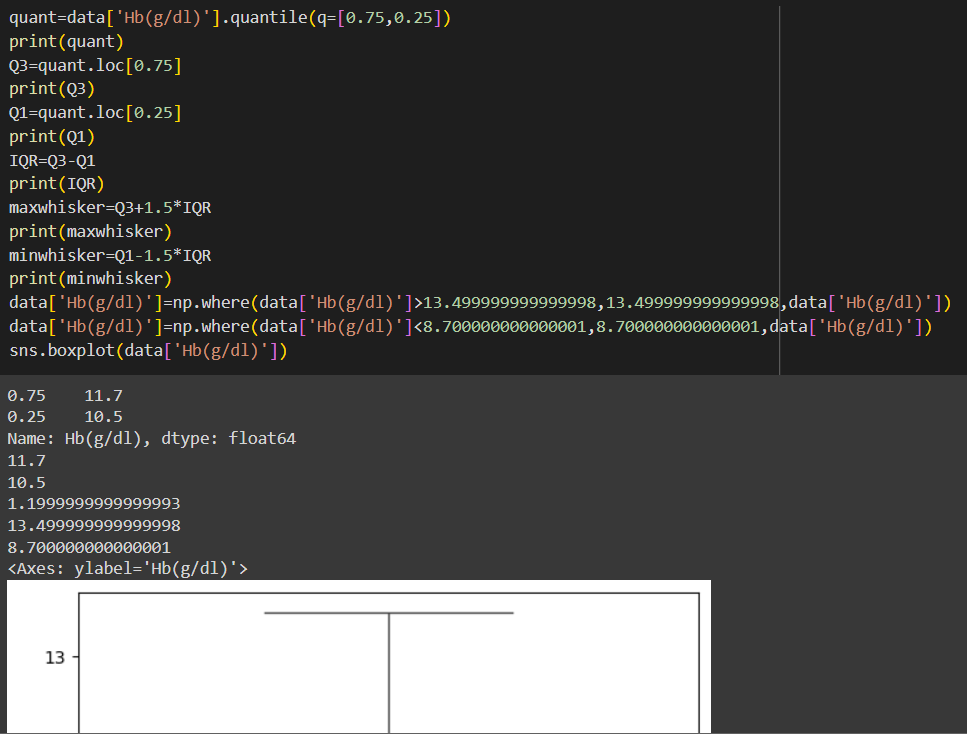


## Activity 2.4:Handling Imbalance Data

With the help of boxplot, outliers are visualized. And here we are going to find upper bound and lower bound of the columns.



* To find upper bound we have to multiply IQR (Interquartile range) with 1.5 and add it with 3rd quantile. To find lower bound instead of adding, subtract it with 1st quantile. Take image attached below as your reference.

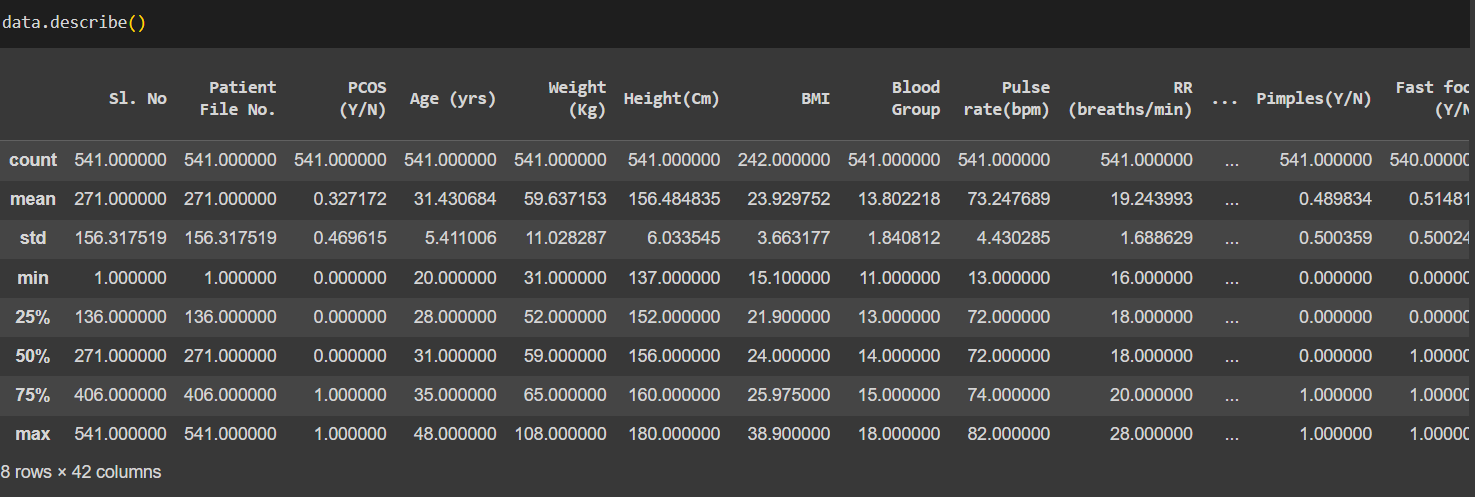


* To handle the outliers transformation technique is used. Here log transformation is used. We have created a function to visualize the distribution and probability plot of the features.

# Milestone 2: Exploratory Data Analysis

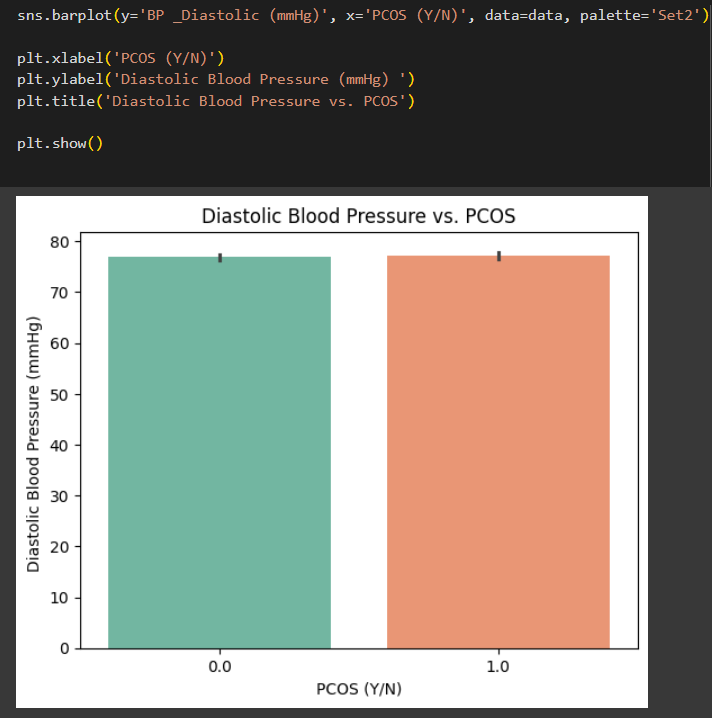
## Activity 1: Descriptive statistics

Descriptive analysis is to study the basic features of data with the statistical process. Here pandas has a worthy function called describe. With this describe function we can understand the unique, top and frequent values of features. And we can find mean, std, min, max and percentile values of continuous features.

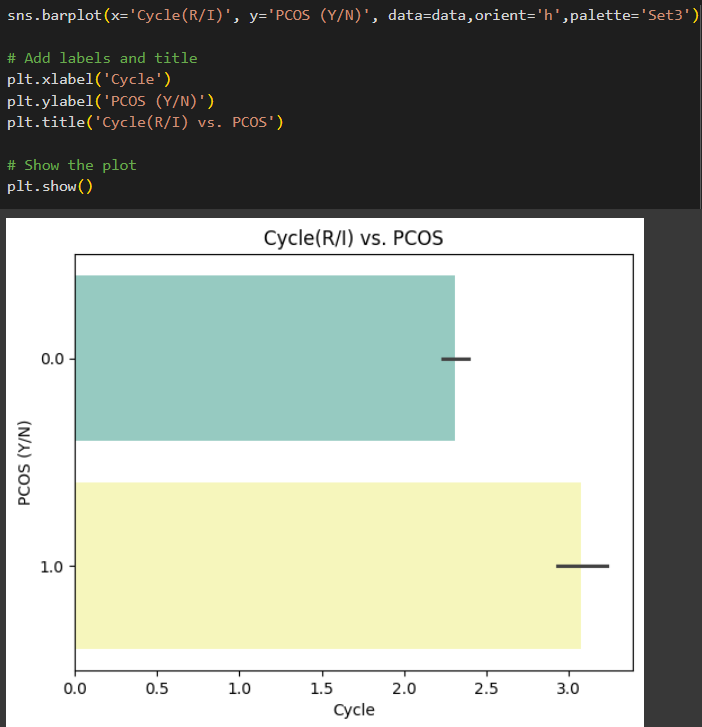


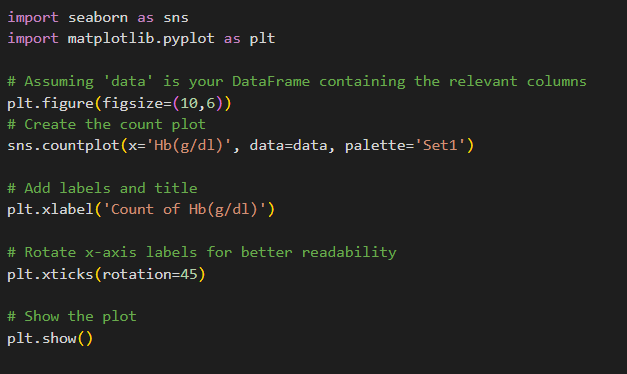
## Activity 2: Visual analysis

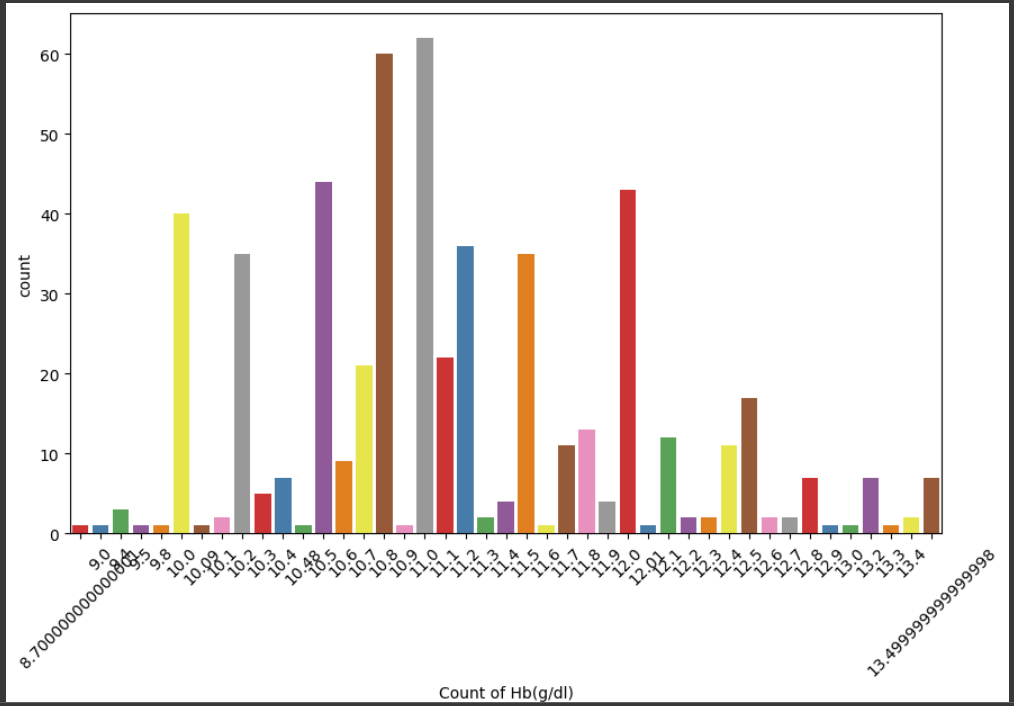
Visual analysis is the process of using visual representations, such as charts, plots, and graphs, to explore and understand data. It is a way to quickly identify patterns, trends, and outliers in the data, which can help to gain insights and make informed decisions.



It is a barplot draw for the Diastolic Blood Pressure upon PCOS (YES?/N0).From the graph we can explain that the when can say that negative and positive results are same when the Diastolic Blood Pressure is 70.



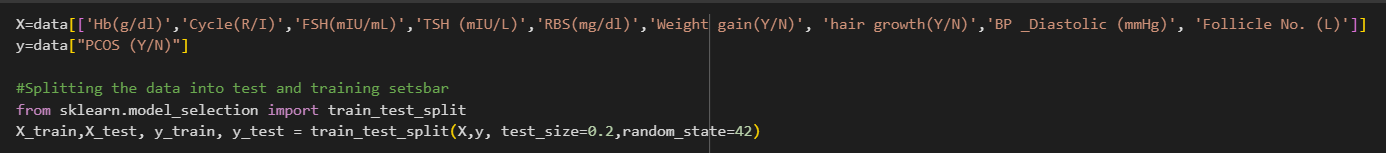




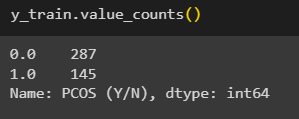
**Splitting data into train and test**

Now let’s split the Dataset into train and test sets. First split the dataset into x and y and then split the data set

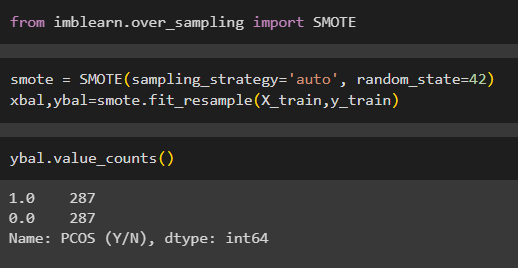
Here x and y variables are created. On x variable, data is passed with dropping the target variable. And on y target variable is passed. For splitting training and testing data we are using train\_test\_split() function from sklearn. As parameters, we are passing x, y, test\_size, random\_state.



**Checking for Balance of target variable**



As the target variable is imbalanced.so, we need to balance the target variable as shown in the below code.



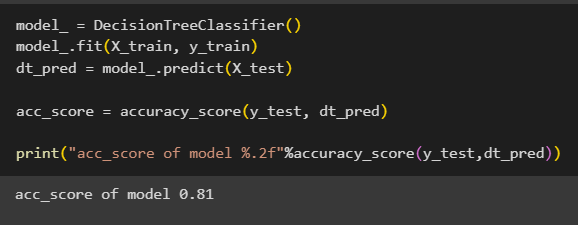
# Milestone 3: Model Building

## Activity 1: Training the model in multiple algorithms

Now our data is cleaned and it’s time to build the model. We can train our data on different algorithms. For this project we are applying three classification algorithms. The best model is saved based on its performance.

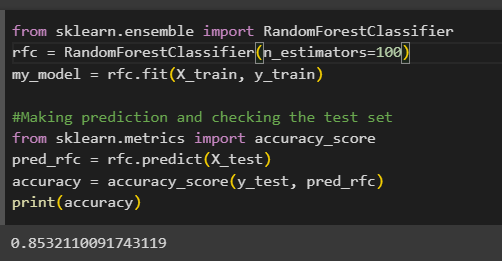
## Activity 1.1: Decision tree model

A function named “model\_” is created and train and test data are passed as the parameters. Inside the function, Decision Tree Classifier algorithm is initialized and training data is passed to the model with the .fit() function. Test data is predicted with .predict() function and saved in a new variable. For evaluating the model, overfitting and accuracy is calculated.



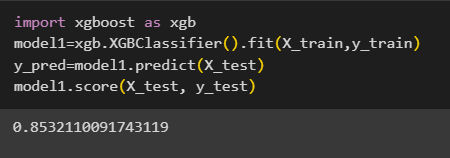
## Activity 1.2: Random forest model

A function named “model” is created and train and test data are passed as the parameters. Inside the function, RandomForestClassifier algorithm is initialised and training data is passed to the model with .fit() function. Test data is predicted with .predict() function and saved in a new variable. For evaluating the model accuracy is calculated.



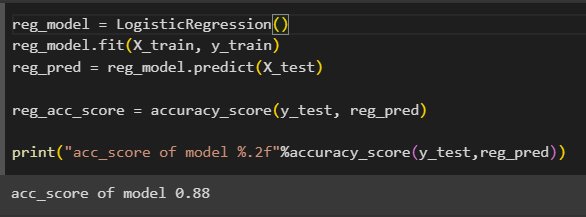
## Activity 1.3: XGBoost Model

A function named “model1” is created and train and test data are passed as the parameters. Inside the function, “XGBClassifier” algorithm is initialised and training data is passed to the model with .fit() function. Test data is predicted with .predict() function and saved in a new variable. For evaluating the model accuracy is calculated.



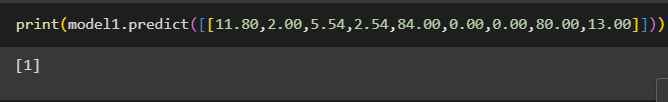
## Activity 1.4: Logistic Regression

A function named “reg\_model” is created and train and test data are passed as the parameters. Inside the function, “logistic regression ” algorithm is initialised and training data is passed to the model with .fit() function. Test data is predicted with .predict() function and saved in a new variable. For evaluating the model accuracy is calculated.



## Activity 2: Testing the model

Here we have tested with XGBoost algorithm. You can test with all algorithm. With the help of predict() function.



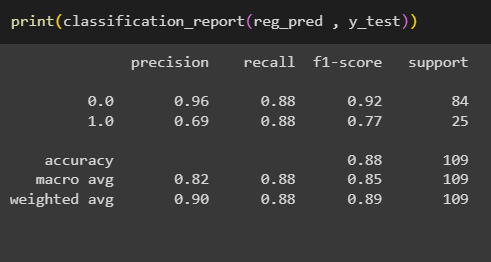
# Milestone 4: Performance Testing

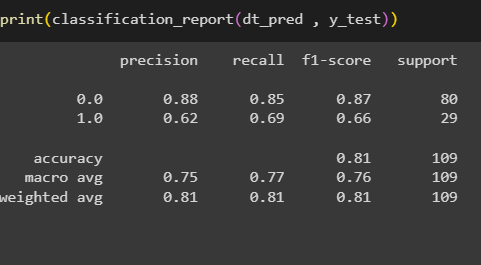
## Activity 1: Testing model with multiple evaluation metrics

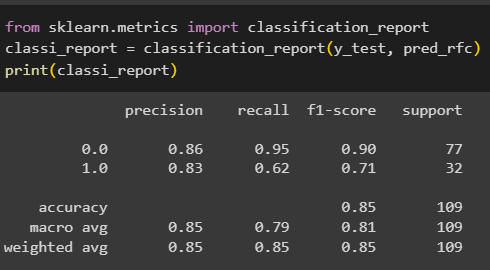
Multiple evaluation metrics means evaluating the model's performance on a test set using different performance measures. This can provide a more comprehensive understanding of the model's strengths and weaknesses. We are using evaluation metrics for classification tasks including accuracy, precision, recall, support and F1-score.

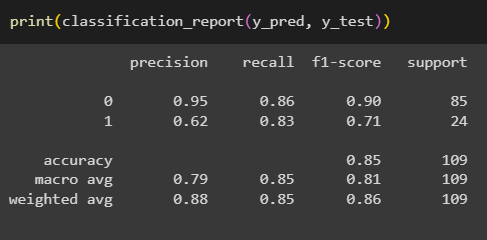
## Activity 1.1: Compare the model

For comparing the above four models, the compare Model function is defined.







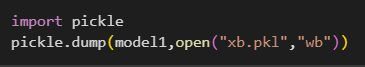


After calling the function, the results of models are displayed as output. From the three models xgboost is performing well

# Milestone 5: Model Deployment

## Activity 1: Save the best model

Saving the best model after comparing its performance using different evaluation metrics means selecting the model with the highest performance and saving its weights and configuration. This can be useful in avoiding the need to retrain the model every time it is needed and also to be able to use it in the future.



## Activity 2: Integrate with Web Framework

In this section, we will be building a web application that is integrated to the model we built. A UI is provided for the uses where he has to enter the values for predictions. The enter values are given to the saved model and prediction is showcased on the UI.

This section has the following tasks

* Building HTML Pages
* Building server-side script
* Run the web application

**Activity 2.1: Building Html Pages:**

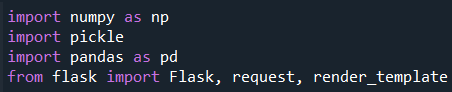
For this project create two HTML files namely

* index.html
* inner-page.html
* output.html

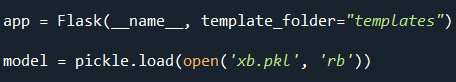
and save them in the templates folder.

**Activity 2.2: Build Python code:**

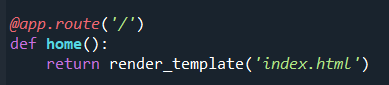
Import the libraries

****

Load the saved model. Importing the flask module in the project is mandatory. An object of Flask class is our WSGI application. Flask constructor takes the name of the current module (\_\_name\_\_) as argument.



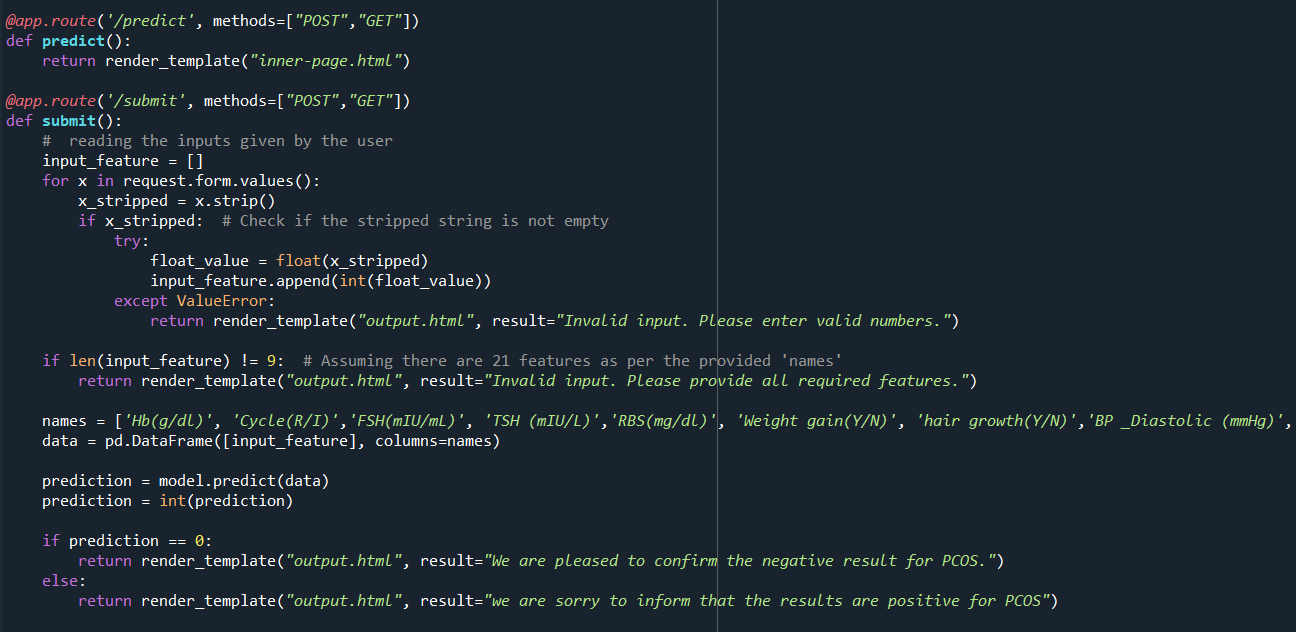
Render HTML page:



Here we will be using a declared constructor to route to the HTML page which we have created earlier.

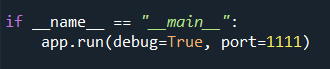
In the above example, ‘/’ URL is bound with the index.html function. Hence, when the home page of the web server is opened in the browser, the html page will be rendered.

Whenever you enter the values from the html page the values can be retrieved using POST Method.

Retrieves the value from UI: 

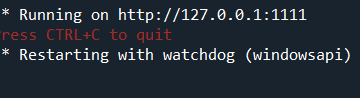
Here we are routing our app to predict() function. This function retrieves all the values from the HTML page using Post request. That is stored in an array. This array is passed to the model.predict() function. This function returns the prediction. And this prediction value will be rendered to the text that we have mentioned in the output.html page earlier.

Main Function:

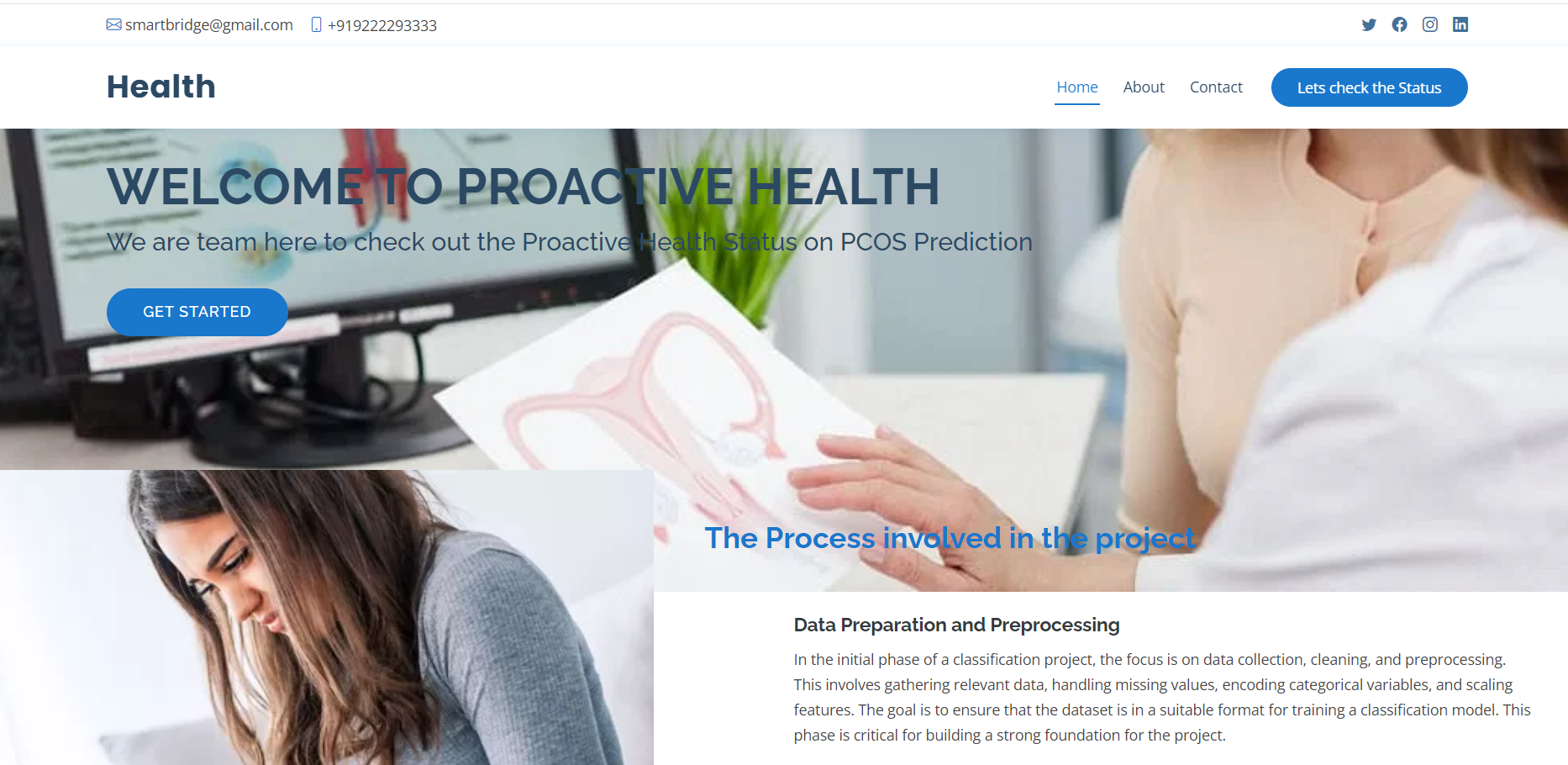


## Activity 2.3: Run the web application

* Open anaconda prompt from the start menu
* Navigate to the folder where your python script is.
* Now type “python app.py” command
* Navigate to the localhost where you can view your web page.
* Click on the predict button from the top left corner, enter the inputs, click on the submit button, and see the result/prediction on the web.

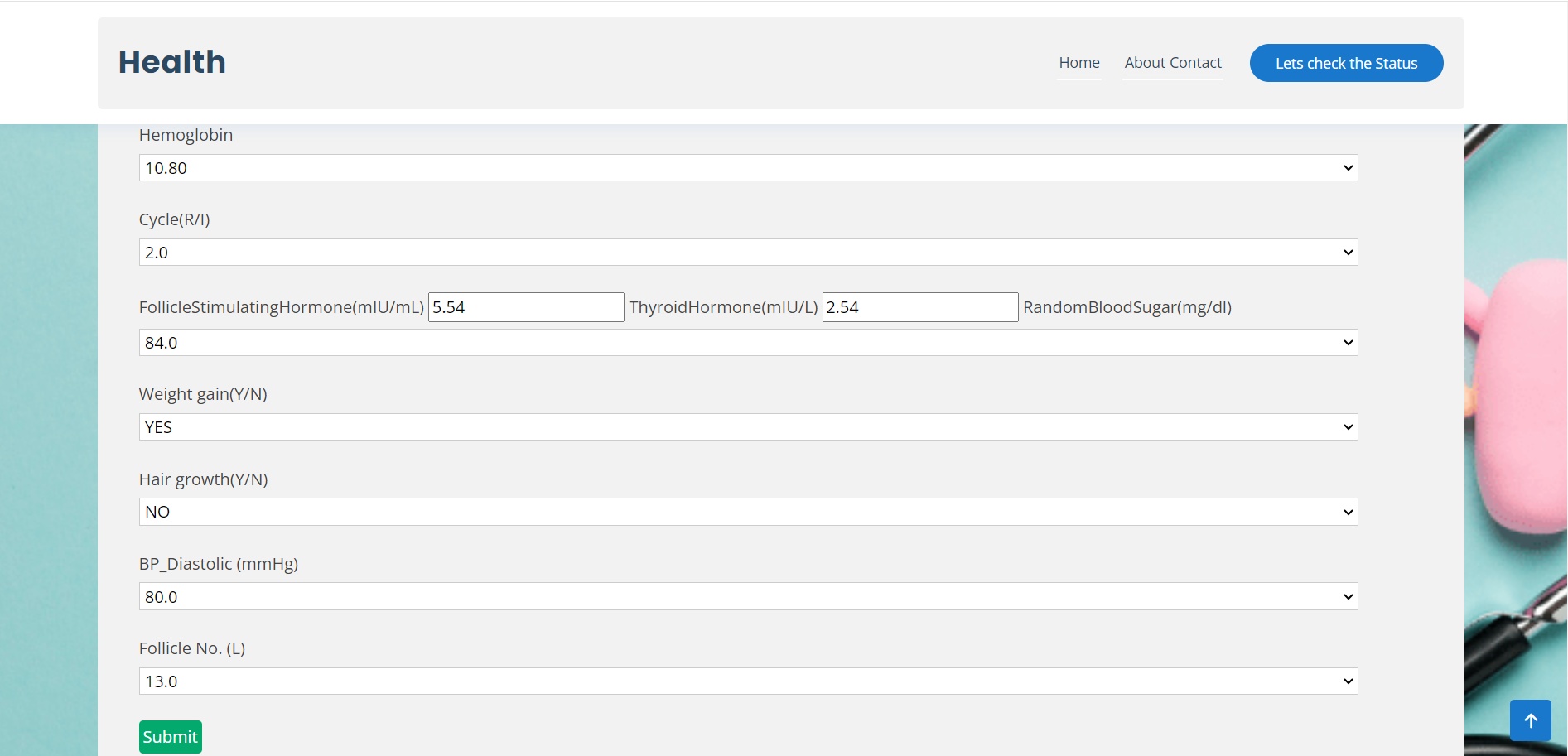


Now,Go the web browser and write the localhost url (http://127.0.0.1:1111) to get the below result

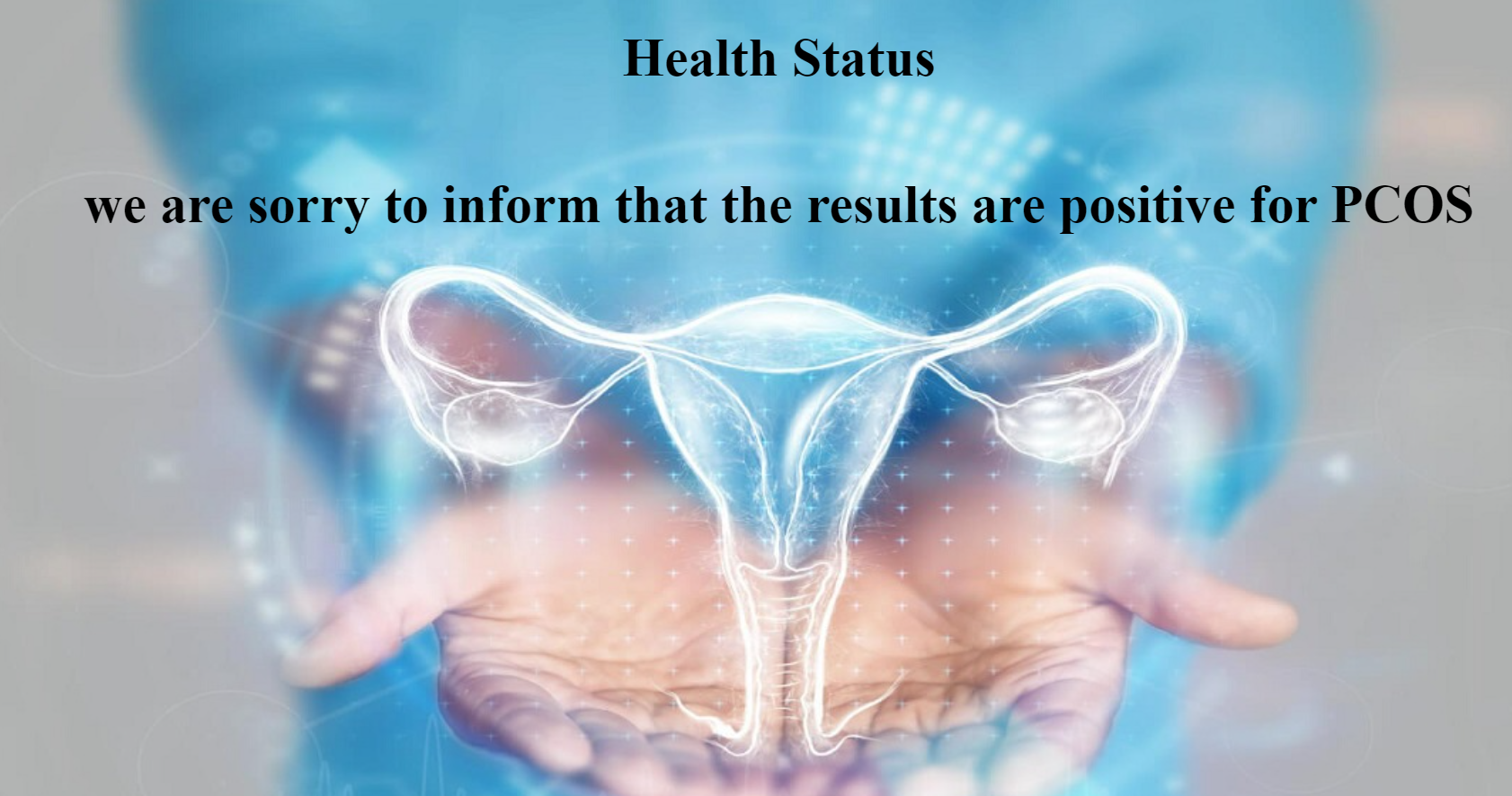


The above UI page is the index page. In this index page it explains the process involved in the project and the contact details in this page.

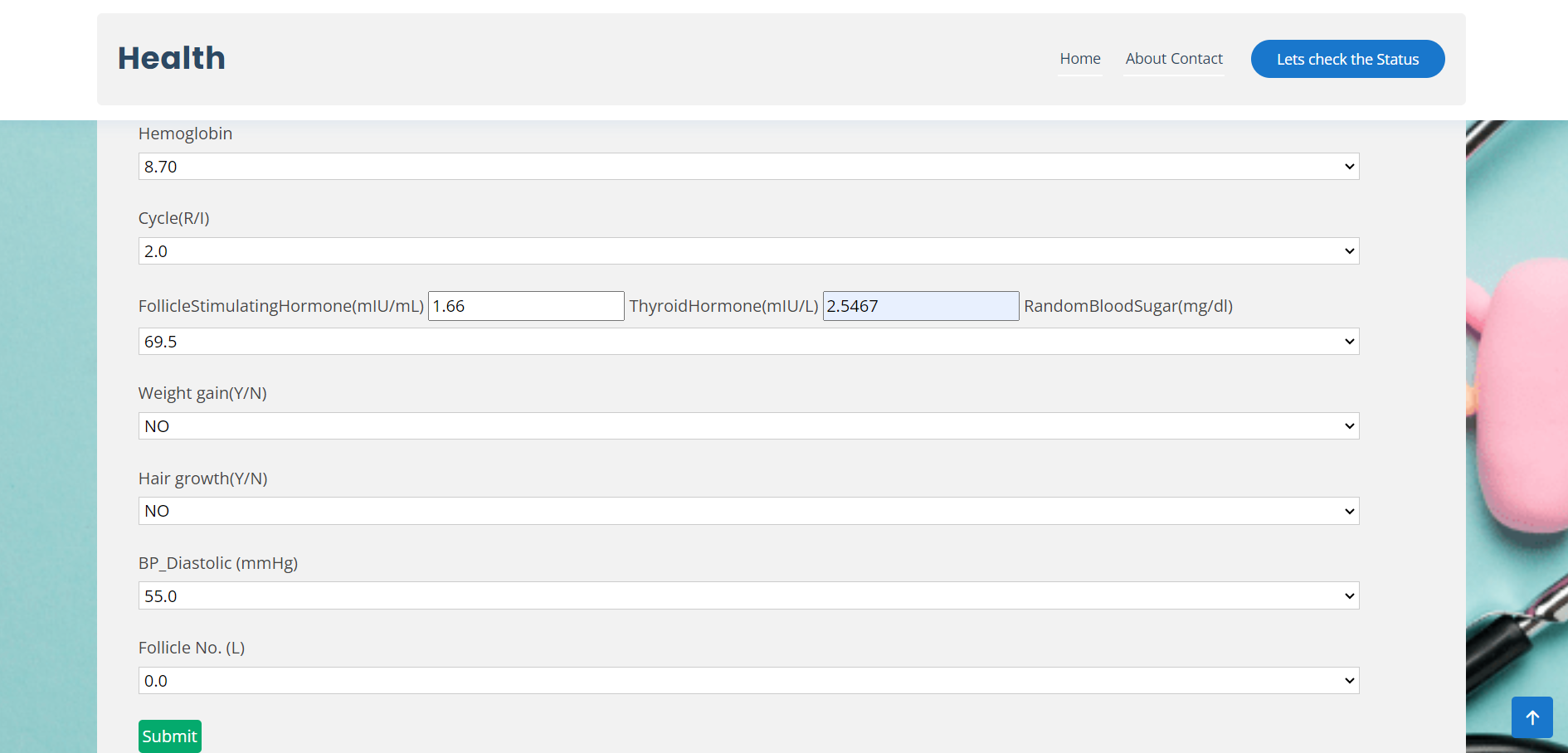
On the top right corner we have Lets Check the Status click on it. So ,that it will be redirect to the inner-page.

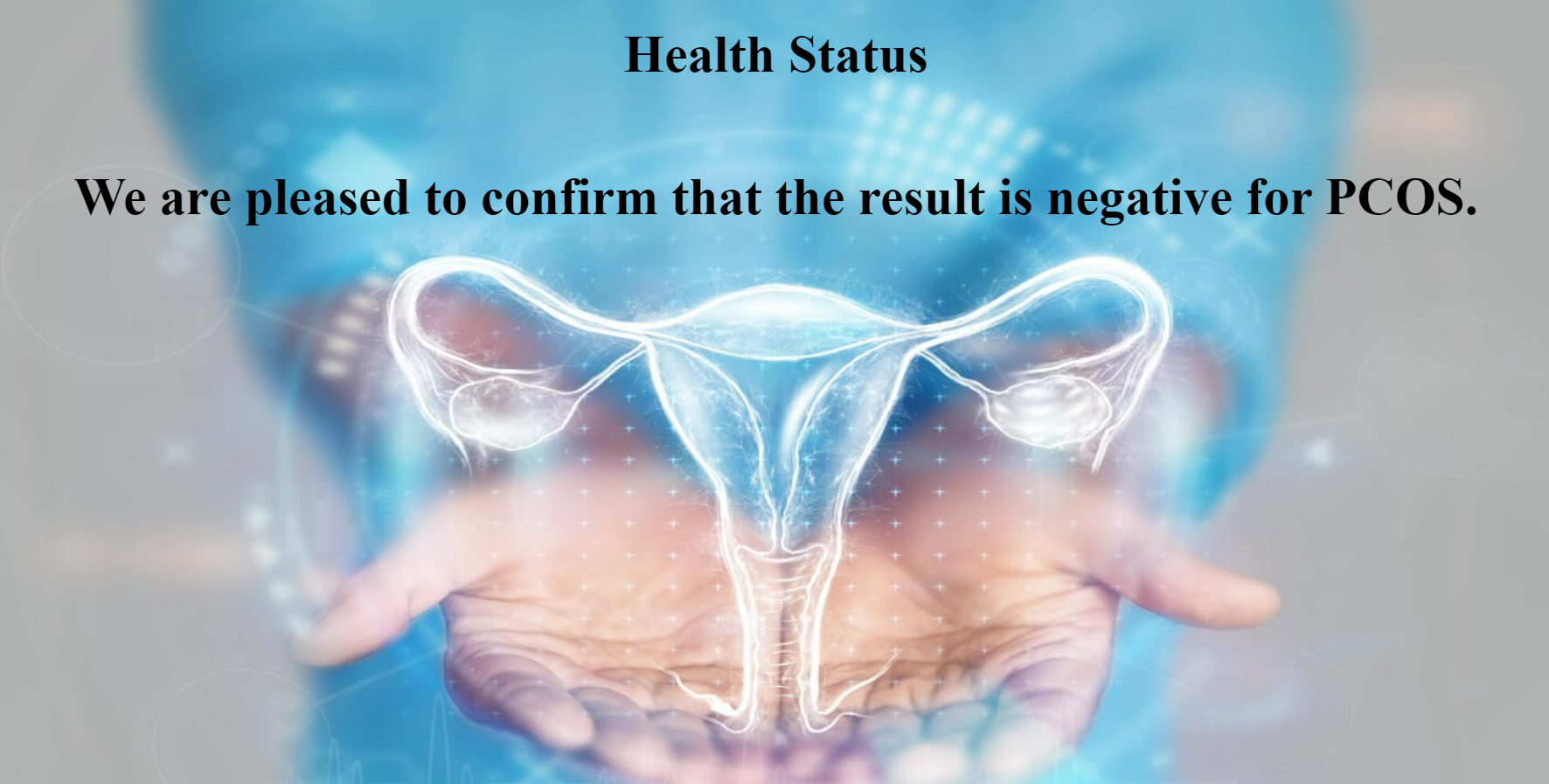


The above image is the inner page. The inner page have the features that are required for our PCOS prediction. So, this how we will be entering the required values for the features and click on the submit button.



As we have entered the values of the required features it predicted that the results are positive for PCOS prediction. This is how we build the user interface for the project.





**Milestone 6: Project Demonstration & Documentation**

Below mentioned deliverables to be submitted along with other deliverables

**Activity 1:- Record explanation Video for project end to end solution**

## Activity 2:- Project Documentation-Step by step project development procedure

Create document as per the template provided