**Build Python Code**

PROJECT NAME: statistical machine learning TEAM ID: PNT2022TMID04163 approaches to liver disease prediction

**PyPi is the official Python repository where all Python packages are stored**. You can think of it as the Github for Python Packages. To make your Python package available to people around the world, you'll need to have an account with PyPi

1.Upload to GitHub. Create a new GitHub repository and push all our code there. ...

2.Create an account in PyPI. We are going to publish the package in PyPI, we need an account. ...

3.Generating distributions. ...

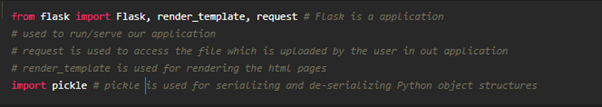
4.Deploy. ...

5.Output:

Let us build flask file ‘Liver\_Flask\_App.py’ which is a web framework written in python for server-side scripting.

Let’s see step by step procedure for building the backend application.

* App starts running when “\_\_name\_\_” constructor is called in main.
* render\_template is used to return html file.
* “GET” method is used to take input from the user.
* “POST” method is used to display the output to the user.



Now after all the libraries are imported, we will be creating our flask app. and the load our model into our flask app.

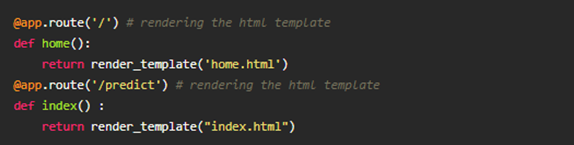
**Routing to the html Page:**

The simplest way to redirect to another URL is to **use an HTML <meta> tag with the http-equiv parameter set to “refresh”**. The content attribute sets the delay before the browser redirects the user to the new web page.

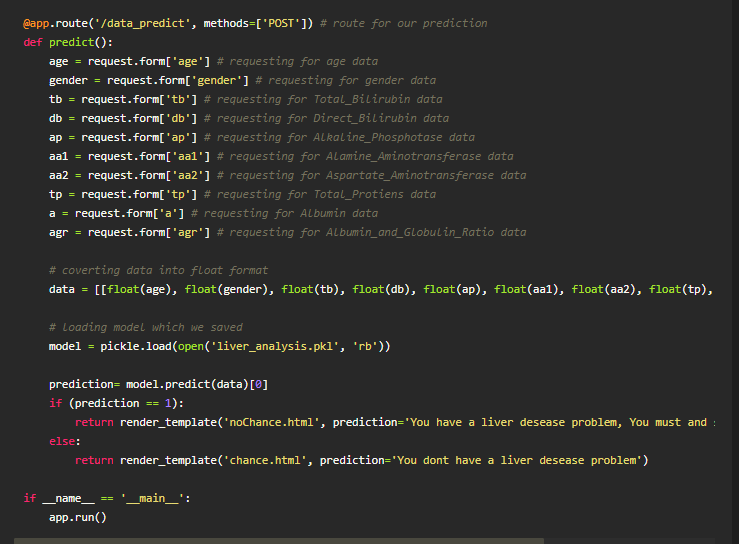
route is used to route the application where it should route to.

‘/’ URL is bound with the home.html function. Hence, when the home page of the web server is opened in the browser, the html page is rendered. Whenever you enter the values from the html page the values can be retrieved using POST Method.

Here, “home.html” is rendered when the home button is clicked on the UI and “index.html” is rendered when the predict button is clicked.



Firstly, we are rendering the home.html template and from there we are navigating to our prediction page that is upload.html. We enter input values here and these values are sent to the loaded model and the resultant output is displayed on index.html.



Here the route for prediction is given and necessary steps are performed in order to get the predicted output.

 Lastly, we run our app on the local host. Here we are running it on localhost:5000