**SPRINT-3**

|  |  |
| --- | --- |
| DATE | 25 October 2022 |
| TEAM ID | PNT2022TMID30520 |
| PROJECT NAME | AI-powered Nutrition Analyzer for Fitness Enthusiasts |

**APPLICATION BUILDING**

Now that we have trained our model, let us build our flask application which will be running in our local browser with a user interface.

In the flask application, the input parameters are taken from the HTML page These factors are then given to the model  to predict the type of food and to know the nutrition content in it. In order to know the nutrition content we will be using an API in this project.

**Create HTML Pages**

* • We use HTML to create the front-end part of the web page.
* • Here, we have created 3 HTML pages- home.html, image.html,imageprediction.html, and 0.html.
* • home.html displays the home page.
* • image.html is used for uploading the image
* • imageprediction.html will showcase the output
* • 0.html is to showcase the result. It tells the action to be performed on imageprediction.html while showcasing the result.

For more information regarding HTML

<https://www.w3schools.com/html/>

* • We also use JavaScript-main.js and CSS-main.css to enhance our functionality and view of HTML pages.
* o Link :[CSS](https://www.w3schools.com/css/) , [JS](https://www.w3schools.com/js/DEFAULT.asp)

Home.html looks like this

https://lh4.googleusercontent.com/yWHq_pVmCV2_UWcd5cUjNyYxm6AblysudZvpHm5De_vxtwz0N2YCXWsu5J_qlB0li4Gtsiy64jCIBFiO-OQVwBryXrgR9YueJRlsl2w-EquON6Ale5CYXo93y9FfCK0YUezjP93aimage.html

https://lh5.googleusercontent.com/H4u7yR2rDVWZPvODOxBNIcBdEqNMgH56AU3mPdqe3eK6MAtjpU8BT1B7S-cr_UhJsG6WgxH9IVL8Fij7M-8xs79T0Hxuc6425ceL5JV_rF7ctgYcm2GYQMuhpeoBd6iJQSo4o3NH

Imageprediction.html

https://lh6.googleusercontent.com/hfi6IRnkKNPCDgTfIOjwOde5vAjFPSHBkakrVDqNjOqycPeoC_1OzA-SNw9AH-NH9Ski2CZivEdtyKSaOZzeW2zt-5hGRlNtWb6spXhybl2lfoMxGwrI91sDc_XybSrzr2vKu4d5

**Build Python Code**

**Importing Libraries**

The first step is usually importing the libraries that will be needed in the program.

https://lh4.googleusercontent.com/CDlluvm2pOZ_VXtwIZ9-Vah2wr2k6Qqv_iO5Mn7sipFbCov3sAsfYH7EGIF34N8w5TXEI9km5jlg88RzCXlntVV0eRHcG9CA4Sllc85ZXlqcq1fe7bxzeUQzfZzkJGvlilDQDFeg

Importing the flask module into the project is mandatory. An object of the Flask class is our WSGI application. Flask constructor takes the name of the current module (\_\_name\_\_) as an argument Pickle library to load the model file.

**Creating Our Flask Application And Loading Our Model By Using Load\_model Method**

**Creating our flask application and loading our model by using the load\_model method**

https://lh4.googleusercontent.com/nSgBQlcrhqY0ih-0ggR2qVq1Q-7gOZu8Js4IARrEDNd9lW_ixH1Qa8YBrTAUciDeSC8fb4ToMesheBuaUBVBtZseqXtatnJthVwOswH2aV5auRDn_FpoLeFf8N4xTrR9Xg3IVfFI

**Routing To The Html Page**

Here, the declared constructor is used to route to the HTML page created earlier.

In the above example, the ‘/’ URL is bound with the home.html function. Hence, when the home page of the webserver 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 the POST Method.

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

https://lh6.googleusercontent.com/ogn_bIr0r1YvCnZJiTgqv5kd0Jhv2gHIjlbPhEyC62KZJhyDVFh1wAYWVhUh8VBEFhUP3IlbjdnyDURwVAClqvZIJyptr3cmyhdTYeyVBm45rZrrkGGY6uJxsdejRMoe3DBSOjbj  
When “image is uploaded “on the UI, the launch function is executed

https://lh3.googleusercontent.com/1G7ddU17DgysQzaHQjVp_sZerzaeRI80qINL154_ac_UeMsRhuceWPplPiik32aR2GrmSTwds_57mOfthTCuAmlg73VWIqYHhVJ_cdHwSwmRRYSLZ7aX3EIXJt_5biIhvOlo2wf7

It will take the image request and we will be storing that image in our local system then we will convert the image into our required size and finally, we will be predicting the results with the help of our model which we trained and depending upon the class identified we will showcase the class name and its properties by rendering the respective html pages.

https://lh4.googleusercontent.com/WnCBBYwf4BWCXgF_Af-kX3GSBCFUsKGM8YMxw90B4S628lExRpPcpVSkvNFBf6knsIAztU4rpKHF1o1qcMRGi5J9DlU5zRY3coLhuZHy2K7T7P9fDkROi0BJ0VcmUUBX_heitE41

**API Integration:**

Here we will be using Rapid API

Using RapidAPI, developers can search and test the APIs, subscribe, and connect to the APIs — all with a single account, single API key and single SDK. Engineering teams also use RapidAPI to share internal APIs and microservice documentation.

API used: [Link](https://rapidapi.com/calorieninjas/api/calorieninjas)

The link above will allow us to test the food item and will result the nutrition content present in the food item.

NOTE: When we keep hitting the API the limit of it might expire. So making a smart use of it will be an efficient way.

How to access and use the API will be shown in this [video](https://github.com/Guided-Projects/Nutrition_Image_Analysis_Using_CNN_Rapid-API/tree/main/Rapid%20API%20Video)

https://lh5.googleusercontent.com/C1EBm0XirzyNJEmu_onYL34eDBf4gv1LChQ8vDK5SgDAbMmZ7BoWGUzSjnE5-TfhcIIAqXxALZ09aDImP5xglelVpyNVOMQC3x4nasibOew7_Ey4LpQVi9DpEDAT8bFOWNHkIw3b

Finally, Run the application

This is used to run the application in a localhost. The local host runs on port number 5000.(We can give different port numbers)

https://lh4.googleusercontent.com/tCpjppoVz1URdR6rWSBcbGXPkgc4Q-3pToEsv_G2lsMPr6woYWDPDPs_Li943u68Z7__risbNR8X9IB2uX9bDRegKdLaU3HPWYHnL2Vzpg-vXul0K4az3V2_d2yZeai44ze8vBhb  
**Run The Application**

* • Open the anaconda prompt from the start menu.
* • Navigate to the folder where your app.py resides.
* • Now type the “python app.py” command.
* • It will show the local host where your app is running on http://127.0.0.1.5000/
* • Copy that localhost URL and open that URL in the browser. It does navigate to where you can view your web page.
* • **Enter the values, click on the predict button and see the result/prediction on the web page.**https://lh6.googleusercontent.com/wNTIPe3fZqmCEaxgRyYHW4qeChuLRTXxNhxR09Z0_bDVTdWrjj6RhBvxHIZszGIbGFD9VXkdxN_a3wWD7cVbxMDavRsOIVK_A1Wn1F5LHZ51SWiDeJxyT0W2hv-0ewoy2TPSb9bV
* • Then it will run on localhost:5000

https://lh4.googleusercontent.com/bYUY4_1c0piE0mPZc7TRabveLBYHcseFlPDs6KrLW8-7DiK9OQkXTB3BduHSviSI_DwQxwfJfbHJyx7qanG2tlcKPH1rYaP9P1ZkptQbiXlk8yegEWqXGkVGSKfIijmOum3FNmOV

Navigate to the localhost (<http://127.0.0.1:5000/>)where you can view your web page.

Click on classify button to see the results.

**Output screenshots:**

https://lh4.googleusercontent.com/yWHq_pVmCV2_UWcd5cUjNyYxm6AblysudZvpHm5De_vxtwz0N2YCXWsu5J_qlB0li4Gtsiy64jCIBFiO-OQVwBryXrgR9YueJRlsl2w-EquON6Ale5CYXo93y9FfCK0YUezjP93a

https://lh4.googleusercontent.com/GSVzWYL02FIxkUHB66BrDqsoAYDbTV8cijM4fpNElYuCgCQmjNmlu3wEMV97b6N_ztfwzQtzbVSqsyqrc4yfcZ4HsD6PrirNvIQPpVQFmLq6LEf_70J2lpcApirc4Wd-DyZq_Qz3

https://lh3.googleusercontent.com/WC0qTE6HWOK0dmce6EOGhVo-iJMIJeFAh_hjX9jDt4_Q0BQzOal-3RDuBtizbrkK3tBmN158BgxFUPxDvjAMPiIIVz0VFbf8jg50swAKeRDhQjsG9CF6o28ZPkj_i9R-sPG5Nz1O

**TRAIN THE MODEL ON IBM**

In this milestone, you will learn how to build Deep Learning Model Using the IBM cloud.

**Register For IBM Cloud**

**IBM Account:**

* • Please click[**here**](https://www.ibm.com/academic/home) to register for IBM
* • Please click [**here**](https://cloud.ibm.com/login)to log in to IBM Account

**Watch the below video to register and login into your IBM account**

**Train Model On IBM**

Please watch the below video to train the model on IBM  and integrate it with the flask Application

**IDEATION PHASE**

In this milestone you are expected to get started with the Ideation process.

**Literature Survey On The Selected Project & Information Gathering**

In this activity you are expected to gather/collect the relevant information on project usecase, refer the existing solutions, technical papers, research publications etc.

**Prepare Empathy Map**

In this activity you are expected to prepare the empathy map canvas to capture the

user Pains & Gains, Prepare list of problem statements.

**Ideation**

In this activity you are expected to list the ideas (at least 4 per each team member) by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.