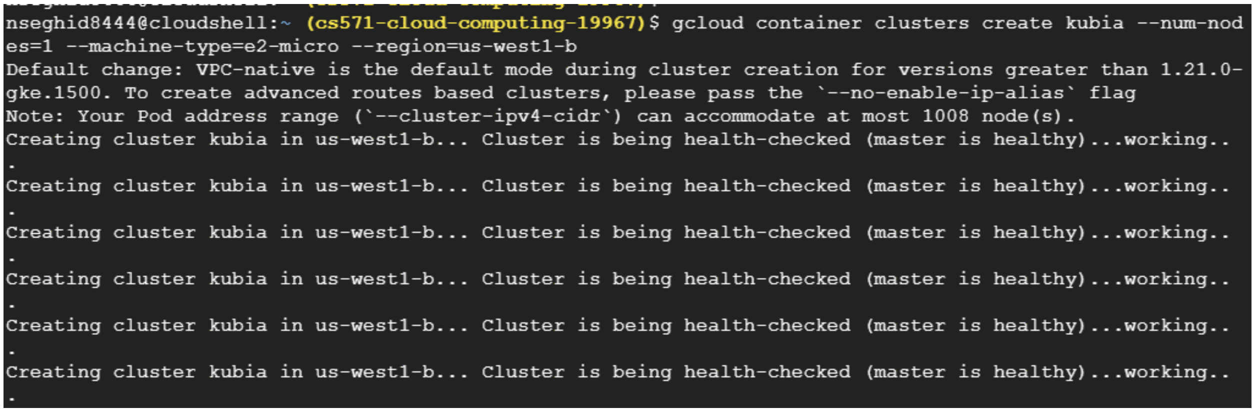
Niyat Habtom

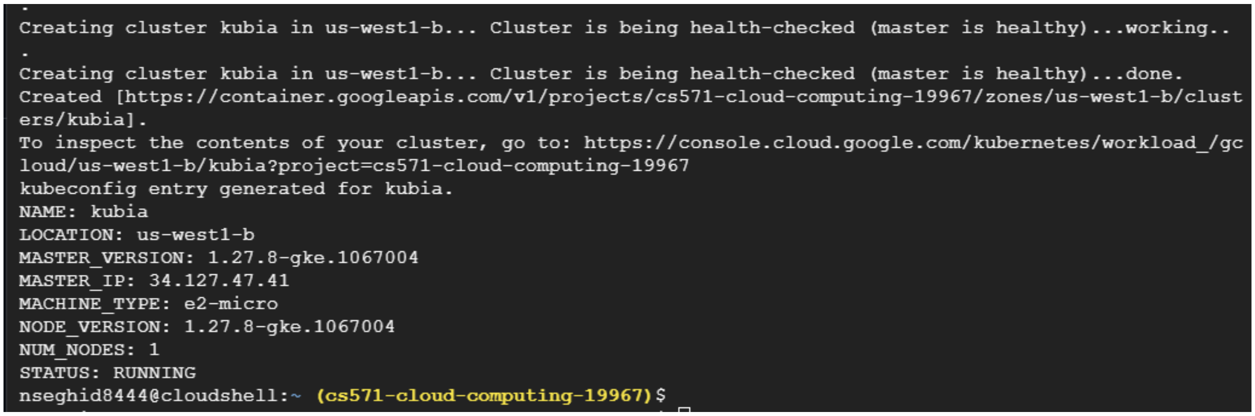
# Machine Learning on Kubernetes

## Setting up a functional Kubernetes cluster

1. First, we create a Kubernetes cluster with three nodes if we deleted the existing ones from the previous HomeWorks. So, we enable GKE first and create Kubernetes cluster:
   gcloud container clusters create kubia --num-nodes=1 --machine-type=e2-micro --region=us-west1

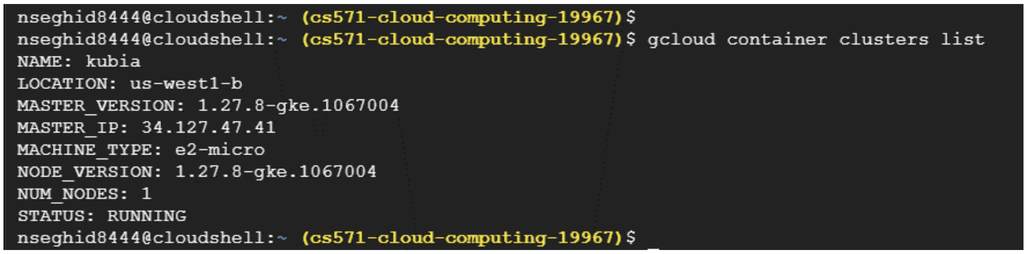


```
nseghid8444@cloudshell:~ (cs571-cloud-computing-19967)$ gcloud container clusters create kubia --num-nod
es=1 --machine-type=e2-micro --region=us-west1-b
Default change: VPC-native is the default mode during cluster creation for versions greater than 1.21.0-
gke.1500. To create advanced routes based clusters, please pass the `--no-enable-ip-alias` flag
Note: Your Pod address range (`--cluster-ipv4-cidr`) can accommodate at most 1008 node(s).
Creating cluster kubia in us-west1-b... Cluster is being health-checked (master is healthy)...working..
.
Creating cluster kubia in us-west1-b... Cluster is being health-checked (master is healthy)...working..
.
Creating cluster kubia in us-west1-b... Cluster is being health-checked (master is healthy)...working..
.
Creating cluster kubia in us-west1-b... Cluster is being health-checked (master is healthy)...working..
.
Creating cluster kubia in us-west1-b... Cluster is being health-checked (master is healthy)...working..
.
Creating cluster kubia in us-west1-b... Cluster is being health-checked (master is healthy)...working..
.
```



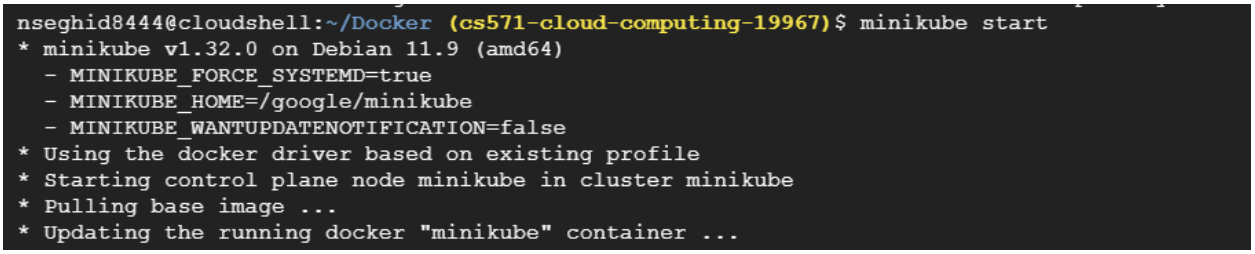
```
.
Creating cluster kubia in us-west1-b... Cluster is being health-checked (master is healthy)...working..
.
Creating cluster kubia in us-west1-b... Cluster is being health-checked (master is healthy)...done.
Created [https://container.googleapis.com/v1/projects/cs571-cloud-computing-19967/zones/us-west1-b/clust
ers/kubia].
To inspect the contents of your cluster, go to: https://console.cloud.google.com/kubernetes/workload_/gc
loud/us-west1-b/kubia?project=cs571-cloud-computing-19967
kubeconfig entry generated for kubia.
NAME: kubia
LOCATION: us-west1-b
MASTER_VERSION: 1.27.8-gke.1067004
MASTER_IP: 34.127.47.41
MACHINE_TYPE: e2-micro
NODE_VERSION: 1.27.8-gke.1067004
NUM_NODES: 1
STATUS: RUNNING
nseghid8444@cloudshell:~ (cs571-cloud-computing-19967)$
```

2. Check if the cluster and the nodes are correctly created and running.
   - Make sure your GKE cluster is running by: gcloud container clusters list



```
nseghid8444@cloudshell:~ (cs571-cloud-computing-19967)$
nseghid8444@cloudshell:~ (cs571-cloud-computing-19967)$ gcloud container clusters list
NAME: kubia
LOCATION: us-west1-b
MASTER_VERSION: 1.27.8-gke.1067004
MASTER_IP: 34.127.47.41
MACHINE_TYPE: e2-micro
NODE_VERSION: 1.27.8-gke.1067004
NUM_NODES: 1
STATUS: RUNNING
nseghid8444@cloudshell:~ (cs571-cloud-computing-19967)$ _
```

- Or we can use minikube to run a single node using: minikube start

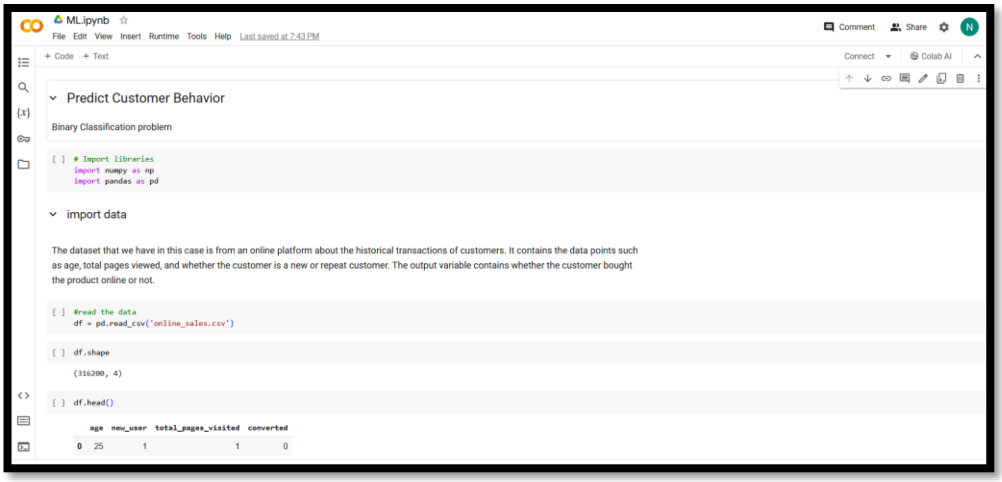


```
nseghid8444@cloudshell:~/Docker (cs571-cloud-computing-19967)$ minikube start
* minikube v1.32.0 on Debian 11.9 (amd64)
  - MINIKUBE_FORCE_SYSTEMD=true
  - MINIKUBE_HOME=/google/minikube
  - MINIKUBE_WANTUPDATENOTIFICATION=false
* Using the docker driver based on existing profile
* Starting control plane node minikube in cluster minikube
* Pulling base image ...
* Updating the running docker "minikube" container ...
```

## Implementing the procedures described on Chapter 4 Machine Learning Deployment Using Docker

### Step 1. Train a machine learning model

- Build, train, and test a machine learning model.



ML.ipynb
File Edit View Insert Runtime Tools Help Last saved at 7:43 PM
Comment
Share
+ Code + Text
Connect
Colab AI

Predict Customer Behavior

Binary Classification problem

```
# Import libraries
import numpy as np
import pandas as pd
```

import data

The dataset that we have in this case is from an online platform about the historical transactions of customers. It contains the data points such as age, total pages viewed, and whether the customer is a new or repeat customer. The output variable contains whether the customer bought the product online or not.

```
#read the data
df = pd.read_csv('online_sales.csv')
```

```
df.shape
```

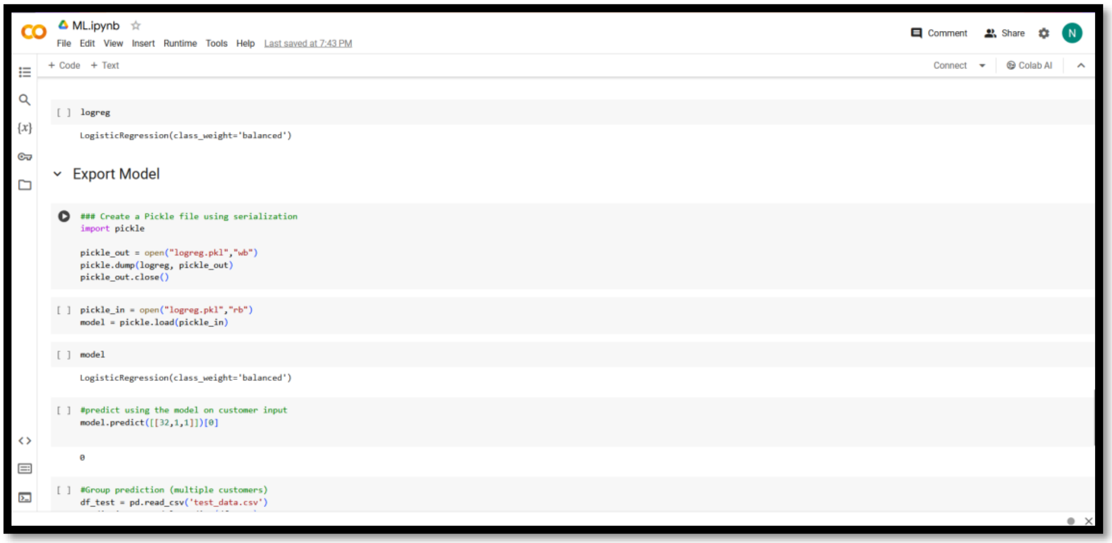
```
(316200, 4)
```

```
df.head()
```

|   | age | new_user | total_pages_visited | converted |
|---|---|---|---|---|
| 0 | 25 | 1 | 1 | 0 |

### Step 2: Save and export the trained ML model.

- Once The model is trained, we need to save it using pickle/joblib in order to reuse it later during predictions.


ML.ipynb
File Edit View Insert Runtime Tools Help Last saved at 7:43 PM
Comment
Share
N
+ Code + Text
Connect
Colab AI

```
[ ] logreg

    LogisticRegression(class_weight='balanced')
```

Export Model

```
### Create a Pickle file using serialization
import pickle

pickle_out = open("logreg.pkl","wb")
pickle.dump(logreg, pickle_out)
pickle_out.close()
```

```
[ ] pickle_in = open("logreg.pkl","rb")
    model = pickle.load(pickle_in)
```

```
[ ] model

    LogisticRegression(class_weight='balanced')
```

```
[ ] #predict using the model on customer input
    model.predict([[32,1,1]])[0]

    0
```

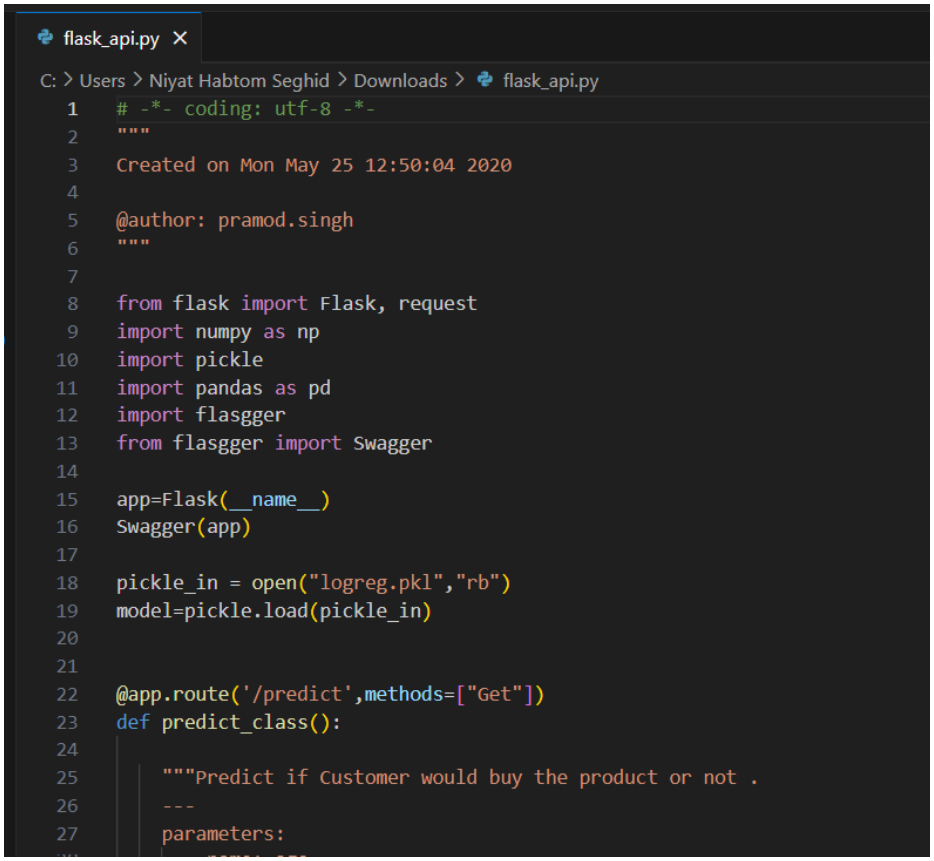
```
[ ] #Group prediction (multiple customers)
    df_test = pd.read_csv('test_data.csv')
```

## Step 3: Create a Flask app including the UI layer

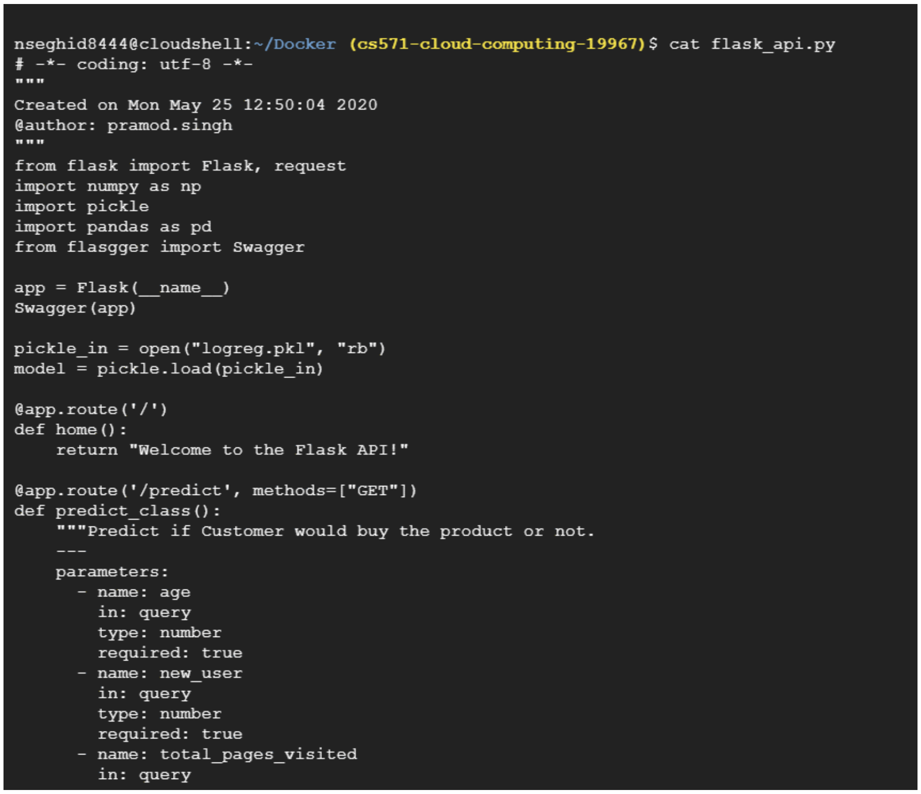
1. Here we will build a Flask app along with Flasgger to deploy an ML model easily. We will do this by creating a flask_api.py file with the following content.

```
# -*- coding: utf-8 -*-
"""
Created on Mon May 25 12:50:04 2020
@author: pramod.singh
"""
from flask import Flask, request
import numpy as np
import pickle
import pandas as pd
from flasgger import Swagger

app = Flask(__name__)
Swagger(app)

pickle_in = open("logreg.pkl", "rb")
model = pickle.load(pickle_in)

@app.route('/')
def home():
    return "Welcome to the Flask API!"

@app.route('/predict', methods=["GET"])
def predict_class():
    """Predict if Customer would buy the product or not.
    ---
    parameters:
      - name: age
        in: query
        type: number
        required: true
      - name: new_user
        in: query
        type: number
        required: true
      - name: total_pages_visited
        in: query
        type: number
        required: true
    responses:
      200:
        description: Prediction
    """
    age = int(request.args.get("age"))
    new_user = int(request.args.get("new_user"))
    total_pages_visited = int(request.args.get("total_pages_visited"))
    prediction = model.predict([[age, new_user, total_pages_visited]])
    return "Model prediction is " + str(prediction)

@app.route('/predict_file', methods=["POST"])
def prediction_test_file():
    """Prediction on multiple input test file.
    ---
    parameters:
      - name: file
        in: formData
        type: file
        required: true
    responses:
      200:
        description: Test file Prediction
    """
    df_test = pd.read_csv(request.files.get("file"))
    prediction = model.predict(df_test)
    return str(list(prediction))

if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0', port=5000)
```

Niyat Habtom



```python
# -*- coding: utf-8 -*-
"""
Created on Mon May 25 12:50:04 2020

@author: pramod.singh
"""

from flask import Flask, request
import numpy as np
import pickle
import pandas as pd
import flasgger
from flasgger import Swagger

app=Flask(__name__)
Swagger(app)

pickle_in = open("logreg.pkl","rb")
model=pickle.load(pickle_in)


@app.route('/predict',methods=["Get"])
def predict_class():

    """Predict if Customer would buy the product or not .
    ---
    parameters:
```

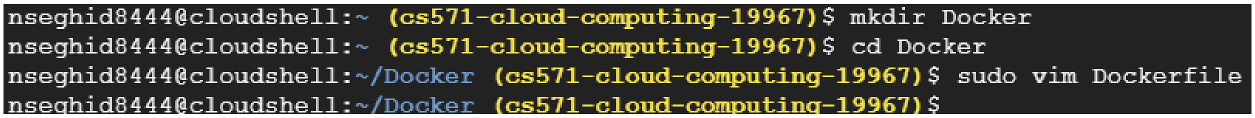


```
nseghid8444@cloudshell:~/Docker (cs571-cloud-computing-19967)$ cat flask_api.py
# -*- coding: utf-8 -*-
"""
Created on Mon May 25 12:50:04 2020
@author: pramod.singh
"""
from flask import Flask, request
import numpy as np
import pickle
import pandas as pd
from flasgger import Swagger

app = Flask(__name__)
Swagger(app)

pickle_in = open("logreg.pkl", "rb")
model = pickle.load(pickle_in)

@app.route('/')
def home():
    return "Welcome to the Flask API!"

@app.route('/predict', methods=["GET"])
def predict_class():
    """Predict if Customer would buy the product or not.
    ---
    parameters:
      - name: age
        in: query
        type: number
        required: true
      - name: new_user
        in: query
        type: number
        required: true
      - name: total_pages_visited
        in: query
```

## Step 4: Creating a custom Docker File for the app

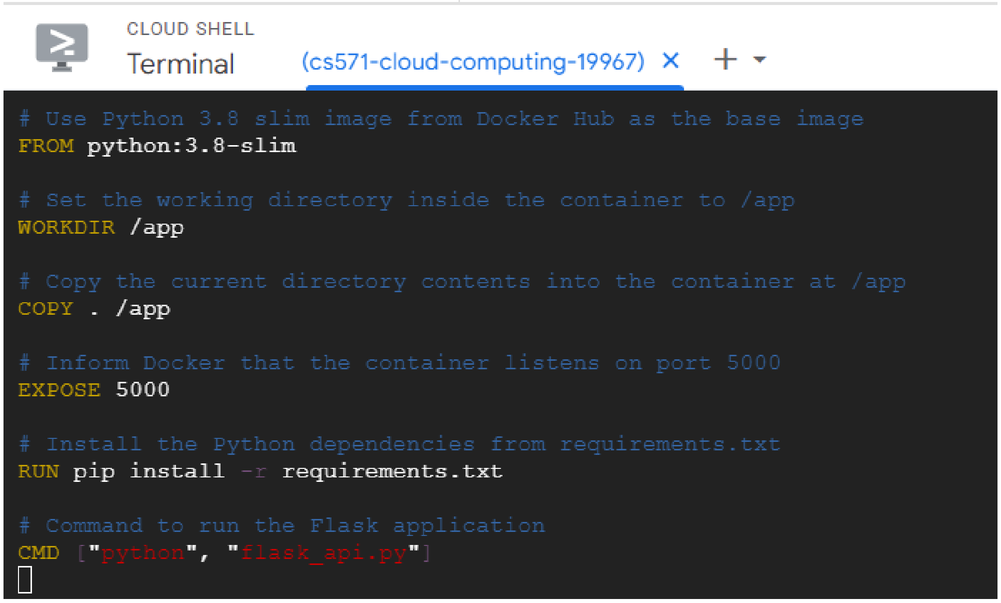
In this section, we will build a classification model from scratch and try to deploy it using a Flask app. The only difference from the previous chapter will be to Dockerize the entire application and run it on any platform. We will also make use of the library Flasgger to handle the UI part of the app to make it more intuitive to consume the results.

1. To start with, let's create a new folder in the local system called **Docker**.



```
nseghid8444@cloudshell:~ (cs571-cloud-computing-19967)$ mkdir Docker
nseghid8444@cloudshell:~ (cs571-cloud-computing-19967)$ cd Docker
nseghid8444@cloudshell:~/Docker (cs571-cloud-computing-19967)$ sudo vim Dockerfile
nseghid8444@cloudshell:~/Docker (cs571-cloud-computing-19967)$
```

2. We are now going to create a **Dockerfile** and mention all the steps to run this app using Docker. This Dockerfile creates a Docker image for a Flask web application: Edit Dockerfile to this new content using command: sudo vim Dockerfile

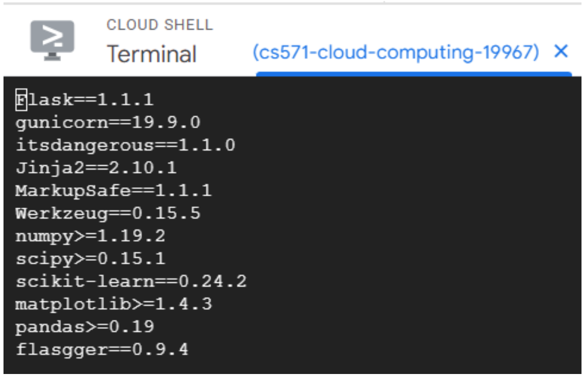


```
# Use Python 3.8 slim image from Docker Hub as the base image
FROM python:3.8-slim

# Set the working directory inside the container to /app
WORKDIR /app

# Copy the current directory contents into the container at /app
COPY . /app

# Inform Docker that the container listens on port 5000
EXPOSE 5000

# Install the Python dependencies from requirements.txt
RUN pip install -r requirements.txt

# Command to run the Flask application
CMD ["python", "flask_api.py"]
```

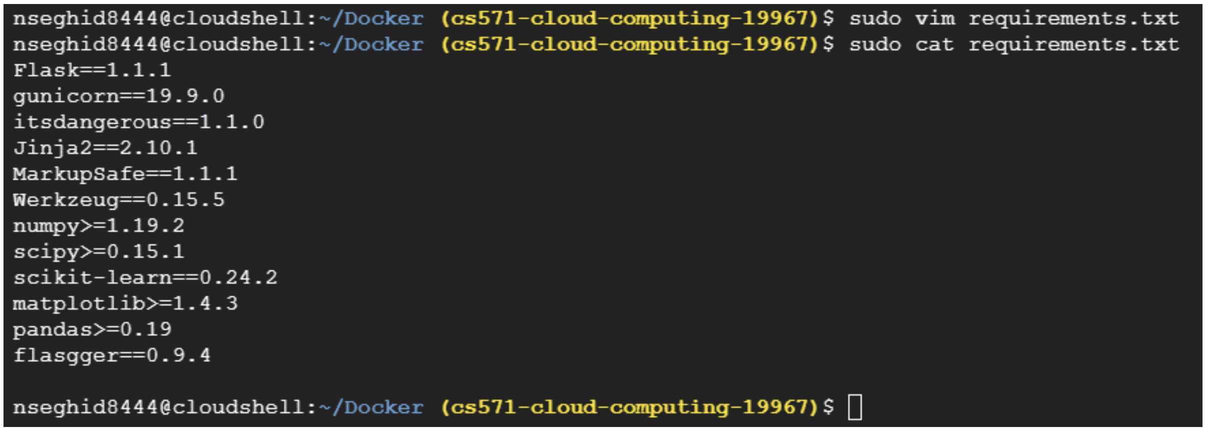
- We start with providing the base image first to the Docker server that needs to be pulled from Docker Hub.
- The next command is to change Docker's current working directory to the directory where we will copy all the files.
- In the next step, we copy all the files and content from the local directory to the Docker directory.
- The next command is to expose port 5000 of Docker to run this application.

- The next step is to install all the dependencies and required libraries.
- The last command in the Dockerfile is the startup command.

3. Create the **requirements.txt** file with the following contents. This file contains all the dependencies and libraries to be installed.



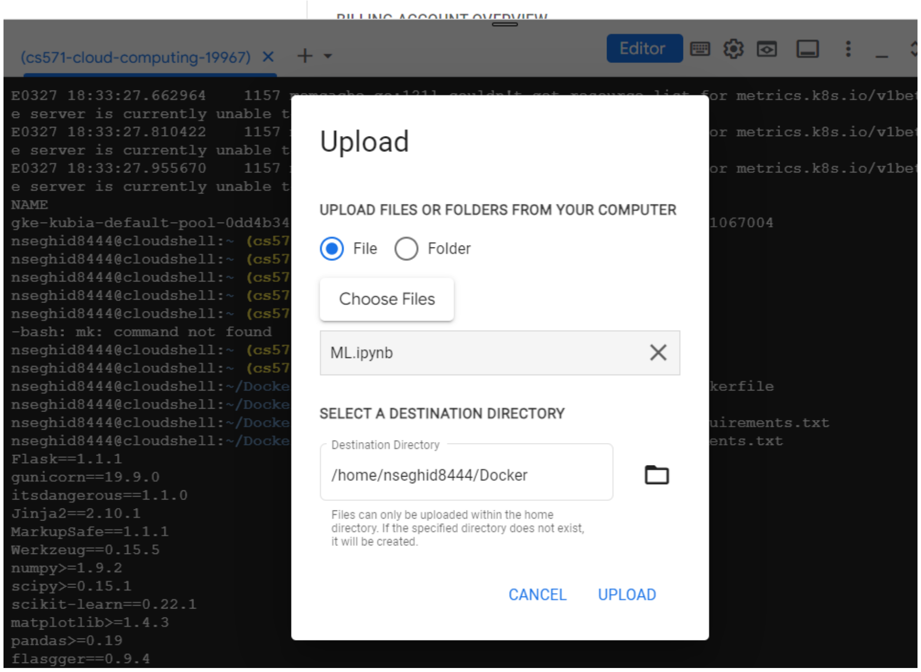
```
Flask==1.1.1
gunicorn==19.9.0
itsdangerous==1.1.0
Jinja2==2.10.1
MarkupSafe==1.1.1
Werkzeug==0.15.5
numpy>=1.19.2
scipy>=0.15.1
scikit-learn==0.24.2
matplotlib>=1.4.3
pandas>=0.19
flasgger==0.9.4
```



```
nseghid8444@cloudshell:~/Docker (cs571-cloud-computing-19967)$ sudo vim requirements.txt
nseghid8444@cloudshell:~/Docker (cs571-cloud-computing-19967)$ sudo cat requirements.txt
Flask==1.1.1
gunicorn==19.9.0
itsdangerous==1.1.0
Jinja2==2.10.1
MarkupSafe==1.1.1
Werkzeug==0.15.5
numpy>=1.19.2
scipy>=0.15.1
scikit-learn==0.24.2
matplotlib>=1.4.3
pandas>=0.19
flasgger==0.9.4

nseghid8444@cloudshell:~/Docker (cs571-cloud-computing-19967)$
```

4. Then we upload all the files we created before to the terminal. Upload logreg.pkl, ML.ipynb, and flask_api.py files by clicking the three dots at the top-right of the Cloud Shell Terminal and then choose upload.

Niyat Habtom



```
E0327 18:33:27.662964    1157                                          or metrics.k8s.io/v1bet
e server is currently unable t
E0327 18:33:27.810422    1157                                          or metrics.k8s.io/v1bet
e server is currently unable t
E0327 18:33:27.955670    1157                                          or metrics.k8s.io/v1bet
e server is currently unable t
NAME
gke-kubia-default-pool-0dd4b34                                          1067004
nseghid8444@cloudshell:~ (cs57
nseghid8444@cloudshell:~ (cs57
nseghid8444@cloudshell:~ (cs57
nseghid8444@cloudshell:~ (cs57
nseghid8444@cloudshell:~ (cs57
-bash: mk: command not found
nseghid8444@cloudshell:~ (cs57
nseghid8444@cloudshell:~ (cs57
nseghid8444@cloudshell:~/Docke                                          kerfile
nseghid8444@cloudshell:~/Docke
nseghid8444@cloudshell:~/Docke                                          uirements.txt
nseghid8444@cloudshell:~/Docke                                          ents.txt
Flask==1.1.1
gunicorn==19.9.0
itsdangerous==1.1.0
Jinja2==2.10.1
MarkupSafe==1.1.1
Werkzeug==0.15.5
numpy>=1.9.2
scipy>=0.15.1
scikit-learn==0.22.1
matplotlib>=1.4.3
pandas>=0.19
flasgger==0.9.4
```

Upload

UPLOAD FILES OR FOLDERS FROM YOUR COMPUTER

● File ○ Folder

Choose Files

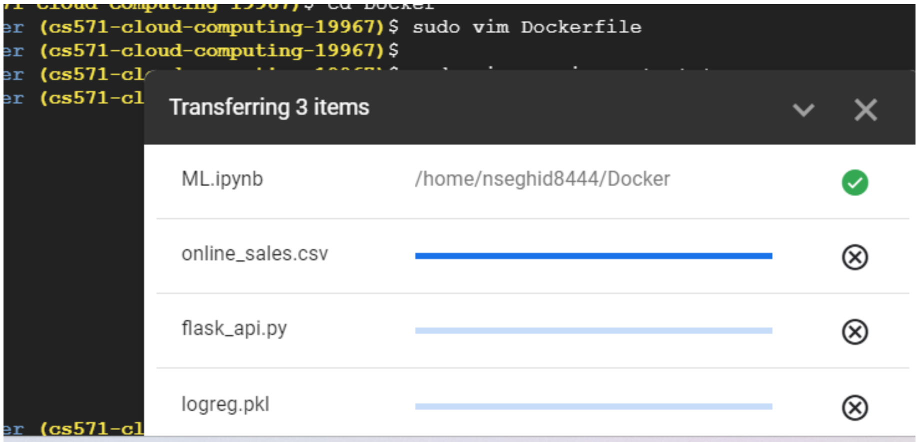
ML.ipynb

SELECT A DESTINATION DIRECTORY

Destination Directory: /home/nseghid8444/Docker

Files can only be uploaded within the home directory. If the specified directory does not exist, it will be created.

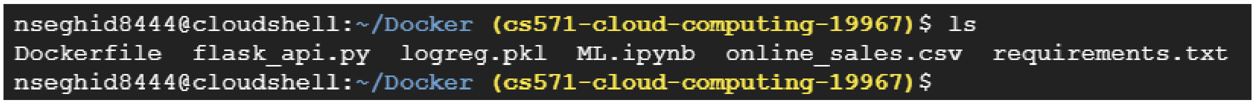
CANCEL UPLOAD



```
er (cs571-cloud-computing-19967)$ sudo vim Dockerfile
er (cs571-cloud-computing-19967)$
er (cs571-cl
er (cs571-cl
er (cs571-cl
```

Transferring 3 items

| | | |
|---|---|---|
| ML.ipynb | /home/nseghid8444/Docker | ✓ |
| online_sales.csv | | ⊗ |
| flask_api.py | | ⊗ |
| logreg.pkl | | ⊗ |



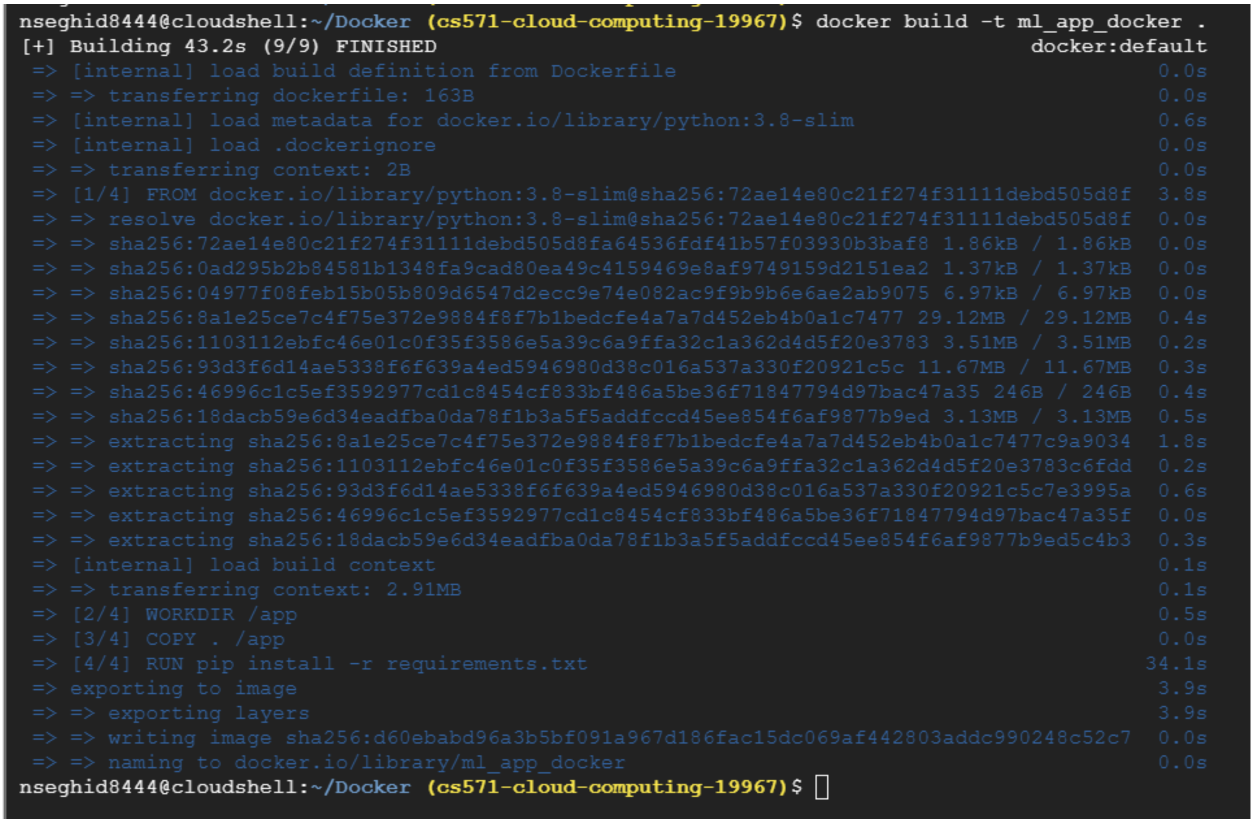
```
nseghid8444@cloudshell:~/Docker (cs571-cloud-computing-19967)$ ls
Dockerfile  flask_api.py  logreg.pkl  ML.ipynb  online_sales.csv  requirements.txt
nseghid8444@cloudshell:~/Docker (cs571-cloud-computing-19967)$
```

## Step 5. Run the app using a Docker container

In this step of the process, we build the Docker custom image from the Dockerfile in the previous step and run the container. Once all the commands in the Dockerfile get executed and we have the final image built, we can initiate the container to run our ML app.

1. We have to go to the terminal in the same directory where all the files are present and run the docker build command to build the Docker image from Dockerfile.

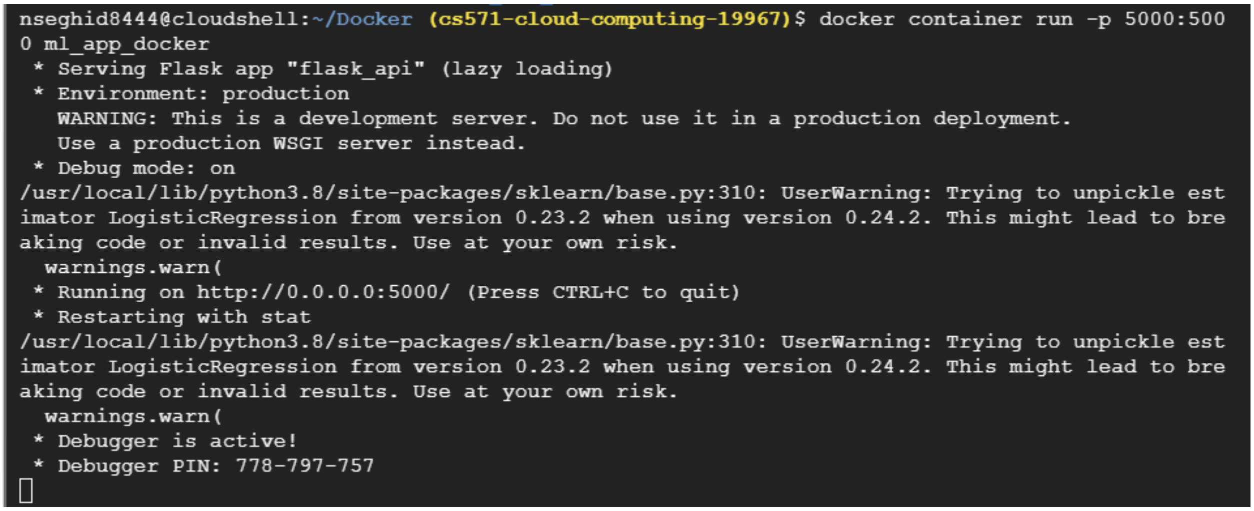
   ```
   sudo docker build -t ml_app_docker .
   ```



```
nseghid8444@cloudshell:~/Docker (cs571-cloud-computing-19967)$ docker build -t ml_app_docker .
[+] Building 43.2s (9/9) FINISHED                                                      docker:default
 => [internal] load build definition from Dockerfile                                             0.0s
 => => transferring dockerfile: 163B                                                             0.0s
 => [internal] load metadata for docker.io/library/python:3.8-slim                               0.6s
 => [internal] load .dockerignore                                                                0.0s
 => => transferring context: 2B                                                                  0.0s
 => [1/4] FROM docker.io/library/python:3.8-slim@sha256:72ae14e80c21f274f31111debd505d8f          3.8s
 => => resolve docker.io/library/python:3.8-slim@sha256:72ae14e80c21f274f31111debd505d8f          0.0s
 => => sha256:72ae14e80c21f274f31111debd505d8fa64536fdf41b57f03930b3baf8 1.86kB / 1.86kB          0.0s
 => => sha256:0ad295b2b84581b1348fa9cad80ea49c4159469e8af9749159d2151ea2 1.37kB / 1.37kB          0.0s
 => => sha256:04977f08feb15b05b809d6547d2ecc9e74e082ac9f9b9b6e6ae2ab9075 6.97kB / 6.97kB          0.0s
 => => sha256:8a1e25ce7c4f75e372e9884f8f7b1bedcfe4a7a7d452eb4b0a1c7477 29.12MB / 29.12MB          0.4s
 => => sha256:1103112ebfc46e01c0f35f3586e5a39c6a9ffa32c1a362d4d5f20e3783 3.51MB / 3.51MB          0.2s
 => => sha256:93d3f6d14ae5338f6f639a4ed5946980d38c016a537a330f20921c5c 11.67MB / 11.67MB          0.3s
 => => sha256:46996c1c5ef3592977cd1c8454cf833bf486a5be36f71847794d97bac47a35 246B / 246B          0.4s
 => => sha256:18dacb59e6d34eadfba0da78f1b3a5f5addfccd45ee854f6af9877b9ed 3.13MB / 3.13MB          0.5s
 => => extracting sha256:8a1e25ce7c4f75e372e9884f8f7b1bedcfe4a7a7d452eb4b0a1c7477c9a9034          1.8s
 => => extracting sha256:1103112ebfc46e01c0f35f3586e5a39c6a9ffa32c1a362d4d5f20e3783c6fdd          0.2s
 => => extracting sha256:93d3f6d14ae5338f6f639a4ed5946980d38c016a537a330f20921c5c7e3995a          0.6s
 => => extracting sha256:46996c1c5ef3592977cd1c8454cf833bf486a5be36f71847794d97bac47a35f          0.0s
 => => extracting sha256:18dacb59e6d34eadfba0da78f1b3a5f5addfccd45ee854f6af9877b9ed5c4b3          0.3s
 => [internal] load build context                                                                0.1s
 => => transferring context: 2.91MB                                                              0.1s
 => [2/4] WORKDIR /app                                                                           0.5s
 => [3/4] COPY . /app                                                                            0.0s
 => [4/4] RUN pip install -r requirements.txt                                                   34.1s
 => exporting to image                                                                           3.9s
 => => exporting layers                                                                          3.9s
 => => writing image sha256:d60ebabd96a3b5bf091a967d186fac15dc069af442803addc990248c52c7          0.0s
 => => naming to docker.io/library/ml_app_docker                                                 0.0s
nseghid8444@cloudshell:~/Docker (cs571-cloud-computing-19967)$ 
```

2. Once all the commands in the Dockerfile get executed and we have the final image built from the Dockerfile, we can initiate the container to run our ML app. This command runs a Docker container from the ml_app_docker image:

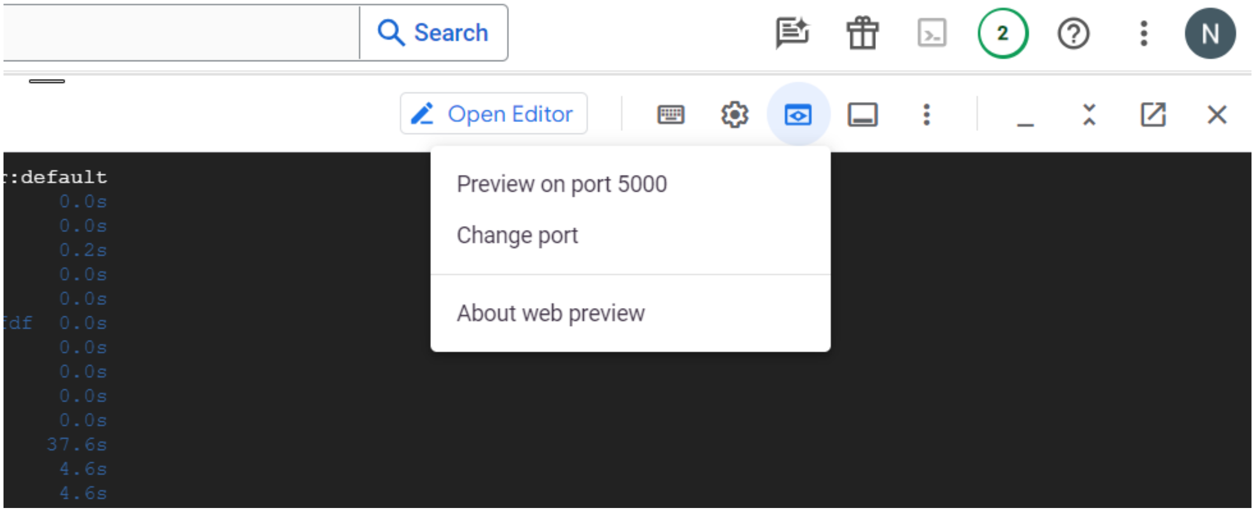
   ```
   sudo docker container run -p 5000:5000 ml_app_docker
   ```



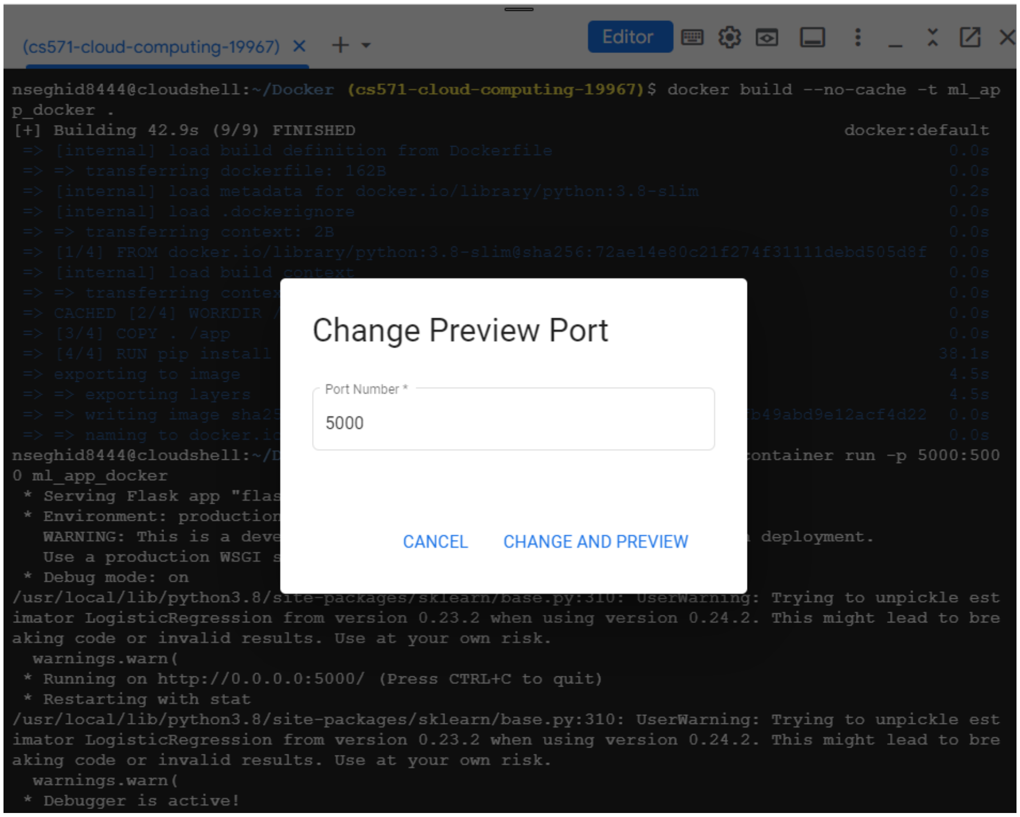
```
nseghid8444@cloudshell:~/Docker (cs571-cloud-computing-19967)$ docker container run -p 5000:500
0 ml_app_docker
 * Serving Flask app "flask_api" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
/usr/local/lib/python3.8/site-packages/sklearn/base.py:310: UserWarning: Trying to unpickle est
imator LogisticRegression from version 0.23.2 when using version 0.24.2. This might lead to bre
aking code or invalid results. Use at your own risk.
  warnings.warn(
 * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
 * Restarting with stat
/usr/local/lib/python3.8/site-packages/sklearn/base.py:310: UserWarning: Trying to unpickle est
imator LogisticRegression from version 0.23.2 when using version 0.24.2. This might lead to bre
aking code or invalid results. Use at your own risk.
  warnings.warn(
 * Debugger is active!
 * Debugger PIN: 778-797-757
```

- We can see that the app has started up successfully, and everything is running inside a Docker container.

3. In the right-upper side of the terminal, click the web preview option from the menu and then click Preview on port 5000. Change port if it is not 5000 by default.



```
:default
     0.0s
     0.0s
     0.2s
     0.0s
     0.0s
fdf  0.0s
     0.0s
     0.0s
     0.0s
     0.0s
    37.6s
     4.6s
     4.6s
```

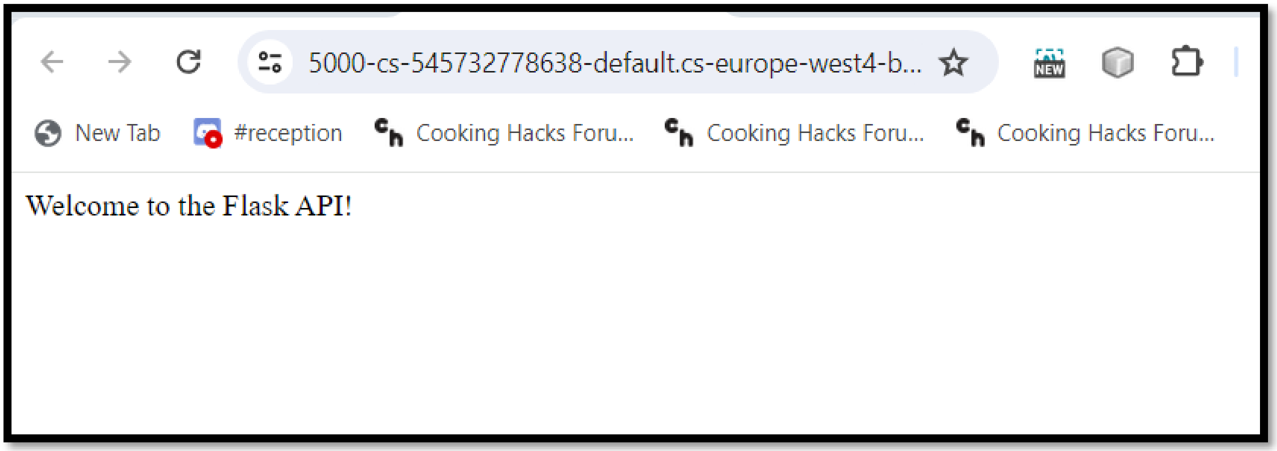


```
nseghid8444@cloudshell:~/Docker (cs571-cloud-computing-19967)$ docker build --no-cache -t ml_ap
p_docker .
[+] Building 42.9s (9/9) FINISHED                                                docker:default
 => [internal] load build definition from Dockerfile                                       0.0s
 => => transferring dockerfile: 162B                                                       0.0s
 => [internal] load metadata for docker.io/library/python:3.8-slim                         0.2s
 => [internal] load .dockerignore                                                          0.0s
 => => transferring context: 2B                                                            0.0s
 => [1/4] FROM docker.io/library/python:3.8-slim@sha256:72ae14e80c21f274f31111debd505d8f  0.0s
 => [internal] load build context                                                          0.0s
 => => transferring conte                                                                  0.0s
 => CACHED [2/4] WORKDIR /                                                                 0.0s
 => [3/4] COPY . /app                                                                      0.0s
 => [4/4] RUN pip install                                                                 38.1s
 => exporting to image                                                                     4.5s
 => => exporting layers                                                                    4.5s
 => => writing image sha25                                        b49abd9e12acf4d22  0.0s
 => => naming to docker.io                                                                 0.0s
nseghid8444@cloudshell:~/D                                        ontainer run -p 5000:500
0 ml_app_docker
 * Serving Flask app "flas
 * Environment: production
   WARNING: This is a deve                                        deployment.
   Use a production WSGI s
 * Debug mode: on
/usr/local/lib/python3.8/site-packages/sklearn/base.py:310: UserWarning: Trying to unpickle est
imator LogisticRegression from version 0.23.2 when using version 0.24.2. This might lead to bre
aking code or invalid results. Use at your own risk.
  warnings.warn(
 * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
 * Restarting with stat
/usr/local/lib/python3.8/site-packages/sklearn/base.py:310: UserWarning: Trying to unpickle est
imator LogisticRegression from version 0.23.2 when using version 0.24.2. This might lead to bre
aking code or invalid results. Use at your own risk.
  warnings.warn(
 * Debugger is active!
```

Change Preview Port

Port Number *: 5000

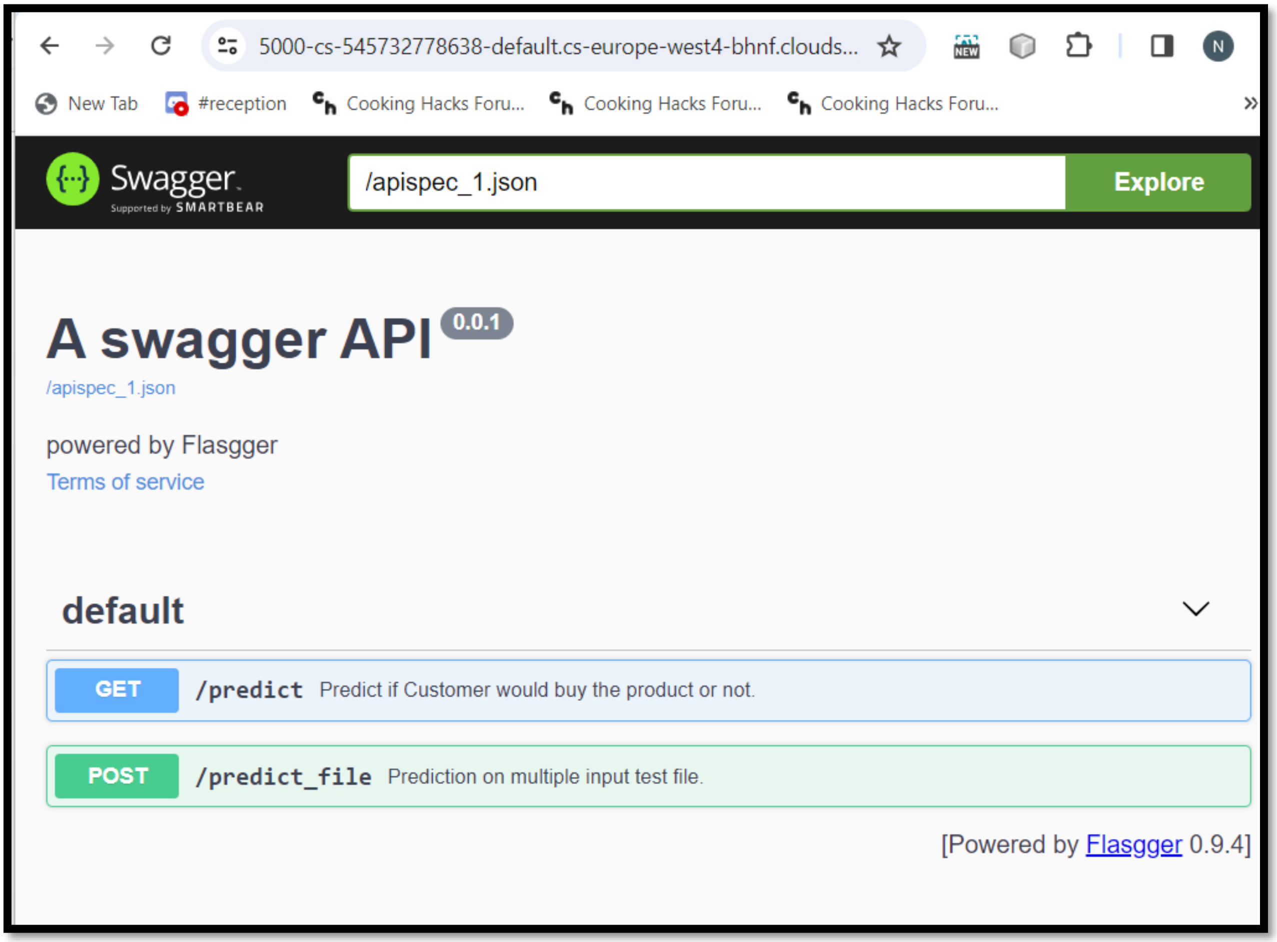
4. After clicking on Change and Preview you will see this message in the web preview.



5000-cs-545732778638-default.cs-europe-west4-b...

Welcome to the Flask API!

5. Add /apidocs/ at the end of the link to access to load the Swagger UI page.

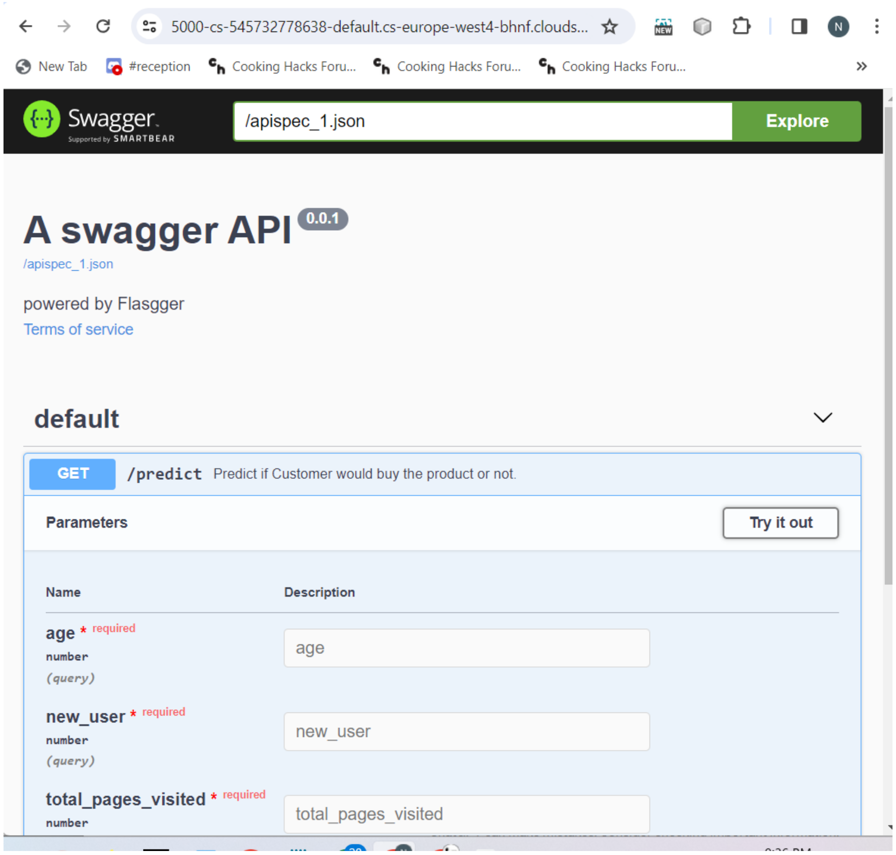


There are two tabs GET and POST. The prediction based on get requests is applicable for single-customer predictions, whereas the Post tab is for the test data prediction (customer group).

- Once we click the Get tab, we can see the options to provide input parameters on which the prediction needs to be made.
- The top-right corner contains a “Try it out” tab that allows us to fill in the values for the input parameters.

6. Click GET and then click Try it out in the top-right corner of the GET box.

Niyat Habtom



5000-cs-545732778638-default.cs-europe-west4-bhnf.clouds...

New Tab #reception Cooking Hacks Foru... Cooking Hacks Foru... Cooking Hacks Foru...

Swagger Supported by SMARTBEAR

/apispec_1.json Explore

# A swagger API 0.0.1

/apispec_1.json

powered by Flasgger

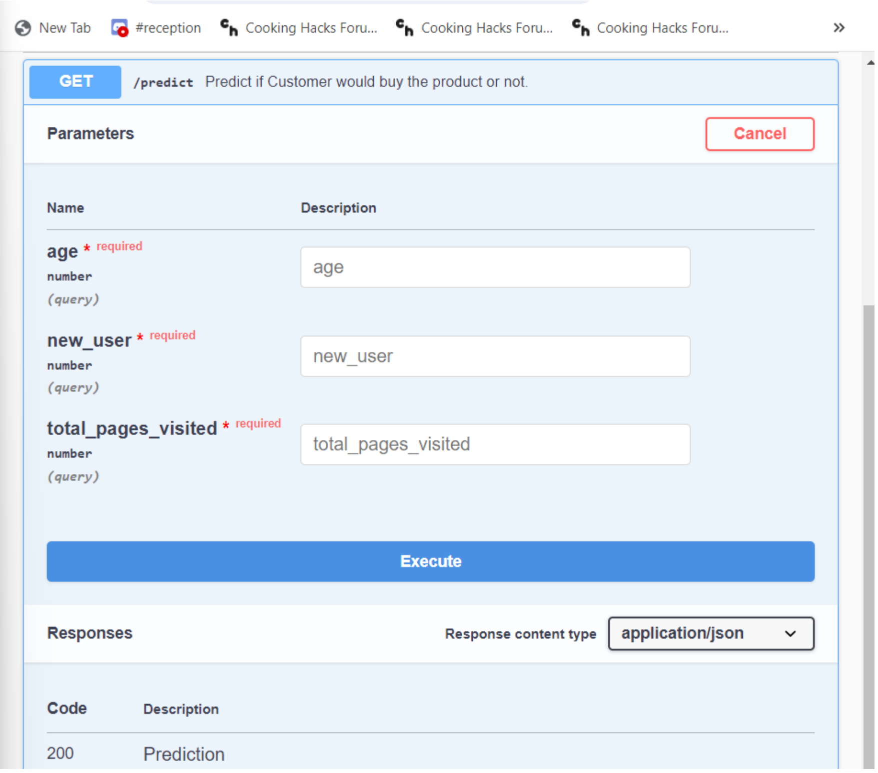
Terms of service

## default

GET /predict Predict if Customer would buy the product or not.

Parameters Try it out

| Name | Description |
|---|---|
| age * required<br>number<br>*(query)* | age |
| new_user * required<br>number<br>*(query)* | new_user |
| total_pages_visited * required<br>number | total_pages_visited |



New Tab #reception Cooking Hacks Foru... Cooking Hacks Foru... Cooking Hacks Foru...

GET /predict Predict if Customer would buy the product or not.

Parameters Cancel

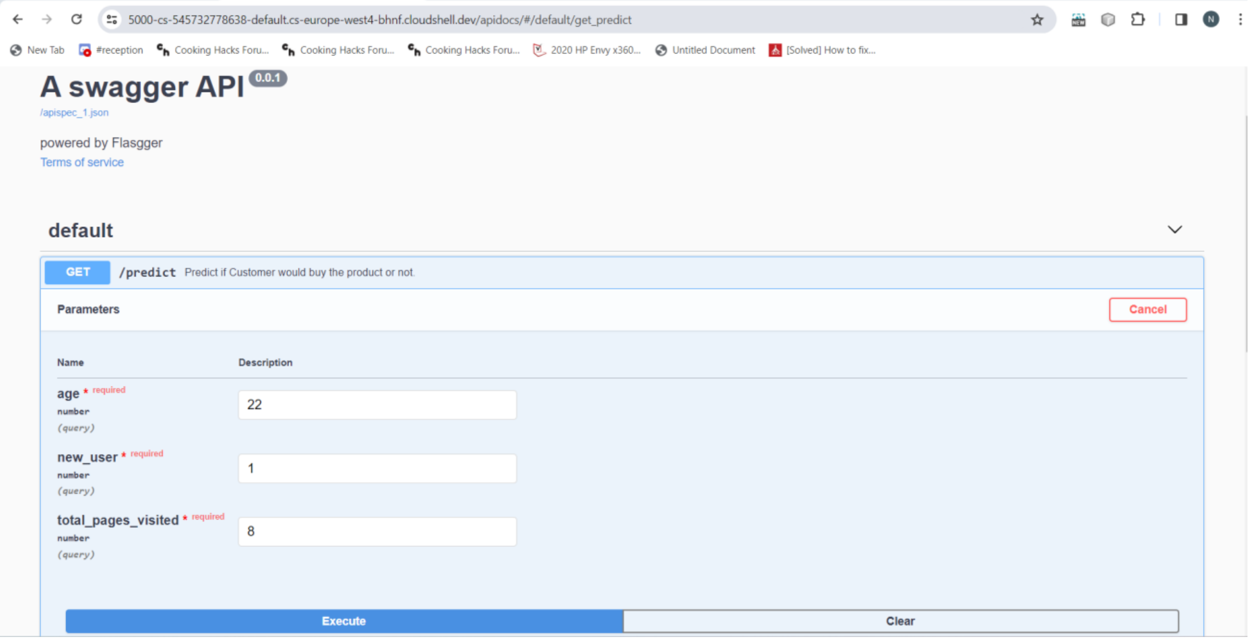
| Name | Description |
|---|---|
| age * required<br>number<br>*(query)* | age |
| new_user * required<br>number<br>*(query)* | new_user |
| total_pages_visited * required<br>number<br>*(query)* | total_pages_visited |

Execute

Responses Response content type application/json

| Code | Description |
|---|---|
| 200 | Prediction |

7. Fill values for the input parameters and then click Execute.



5000-cs-545732778638-default.cs-europe-west4-bhnf.cloudshell.dev/apidocs/#/default/get_predict

## A swagger API 0.0.1

/apispec_1.json

powered by Flasgger
Terms of service

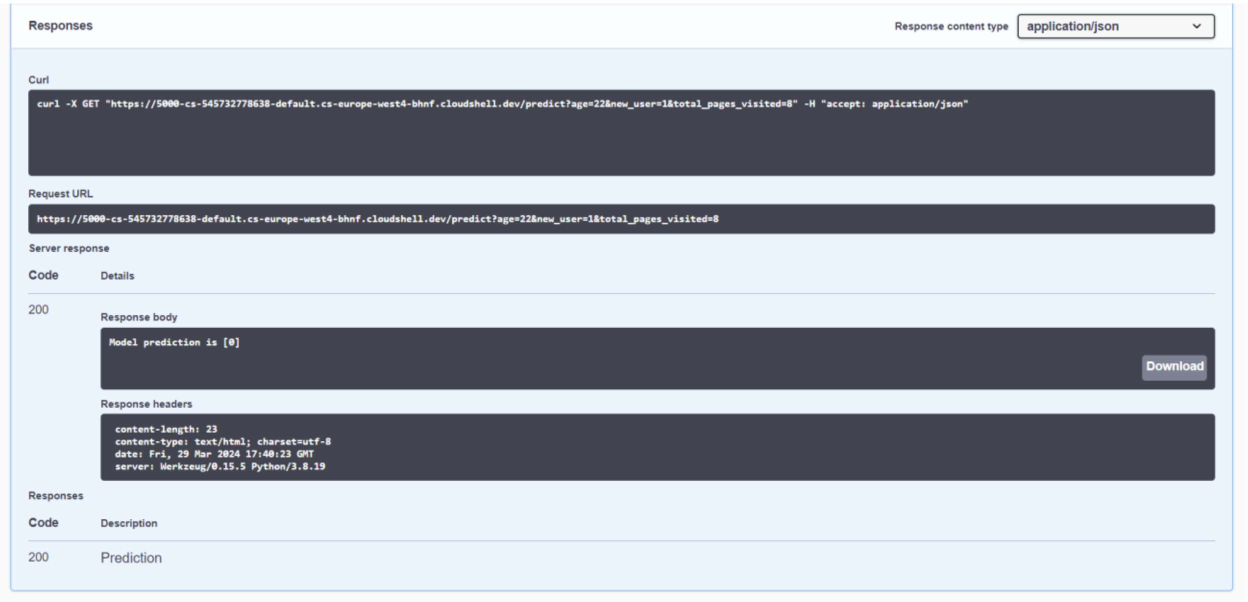
default

GET /predict Predict if Customer would buy the product or not

Parameters

| Name | Description |
|---|---|
| age * required<br>number<br>*(query)* | 22 |
| new_user * required<br>number<br>*(query)* | 1 |
| total_pages_visited * required<br>number<br>*(query)* | 8 |

8. Upon the execution call, the request goes to the app, and predictions are made by the model. The result of the model prediction is displayed in the Prediction section of the page as follows.



Responses

Response content type: application/json

Curl

```
curl -X GET "https://5000-cs-545732778638-default.cs-europe-west4-bhnf.cloudshell.dev/predict?age=22&new_user=1&total_pages_visited=8" -H "accept: application/json"
```

Request URL

```
https://5000-cs-545732778638-default.cs-europe-west4-bhnf.cloudshell.dev/predict?age=22&new_user=1&total_pages_visited=8
```

Server response

| Code | Details |
|---|---|
| 200 | Response body |

```
Model prediction is [0]
```

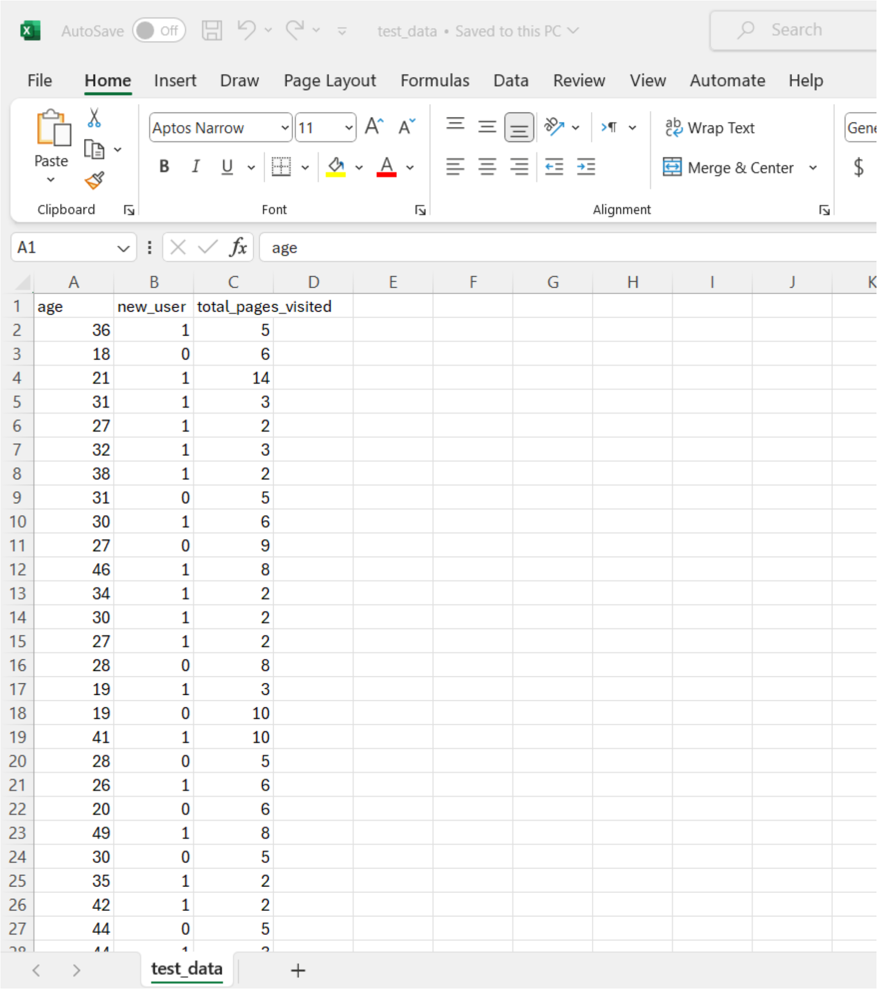
Response headers

```
content-length: 23
content-type: text/html; charset=utf-8
date: Fri, 29 Mar 2024 17:40:23 GMT
server: Werkzeug/0.15.5 Python/3.8.19
```

Responses

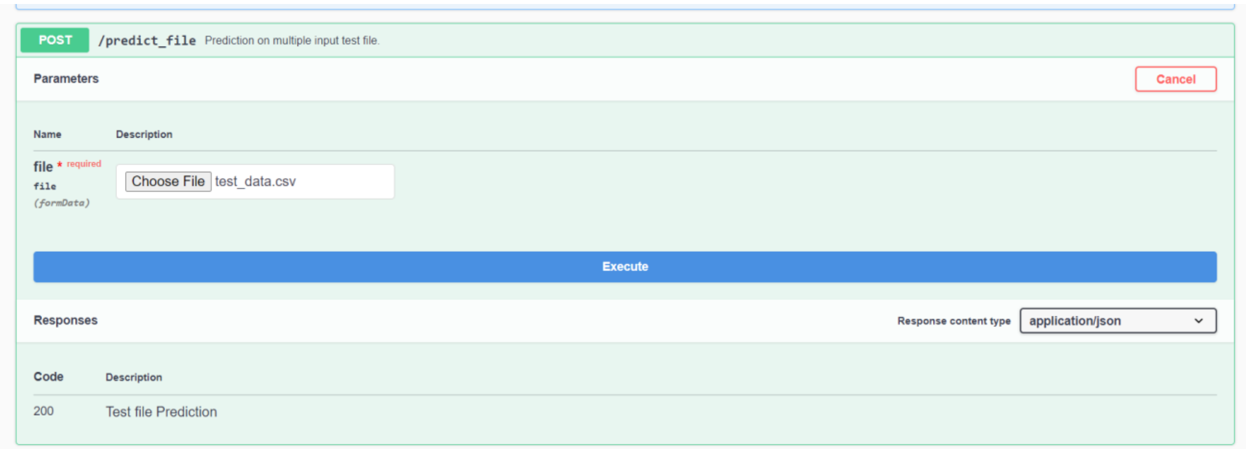
| Code | Description |
|---|---|
| 200 | Prediction |

9. Next we have a test_data csv file that we will use to make predictions for a group of customers through a post request.



| | A | B | C |
|---|---|---|---|
| 1 | age | new_user | total_pages_visited |
| 2 | 36 | 1 | 5 |
| 3 | 18 | 0 | 6 |
| 4 | 21 | 1 | 14 |
| 5 | 31 | 1 | 3 |
| 6 | 27 | 1 | 2 |
| 7 | 32 | 1 | 3 |
| 8 | 38 | 1 | 2 |
| 9 | 31 | 0 | 5 |
| 10 | 30 | 1 | 6 |
| 11 | 27 | 0 | 9 |
| 12 | 46 | 1 | 8 |
| 13 | 34 | 1 | 2 |
| 14 | 30 | 1 | 2 |
| 15 | 27 | 1 | 2 |
| 16 | 28 | 0 | 8 |
| 17 | 19 | 1 | 3 |
| 18 | 19 | 0 | 10 |
| 19 | 41 | 1 | 10 |
| 20 | 28 | 0 | 5 |
| 21 | 26 | 1 | 6 |
| 22 | 20 | 0 | 6 |
| 23 | 49 | 1 | 8 |
| 24 | 30 | 0 | 5 |
| 25 | 35 | 1 | 2 |
| 26 | 42 | 1 | 2 |
| 27 | 44 | 0 | 5 |

10. Upload the test data using the following order. Click → Post→Try it Out→Choose File.



POST /predict_file Prediction on multiple input test file.

Parameters

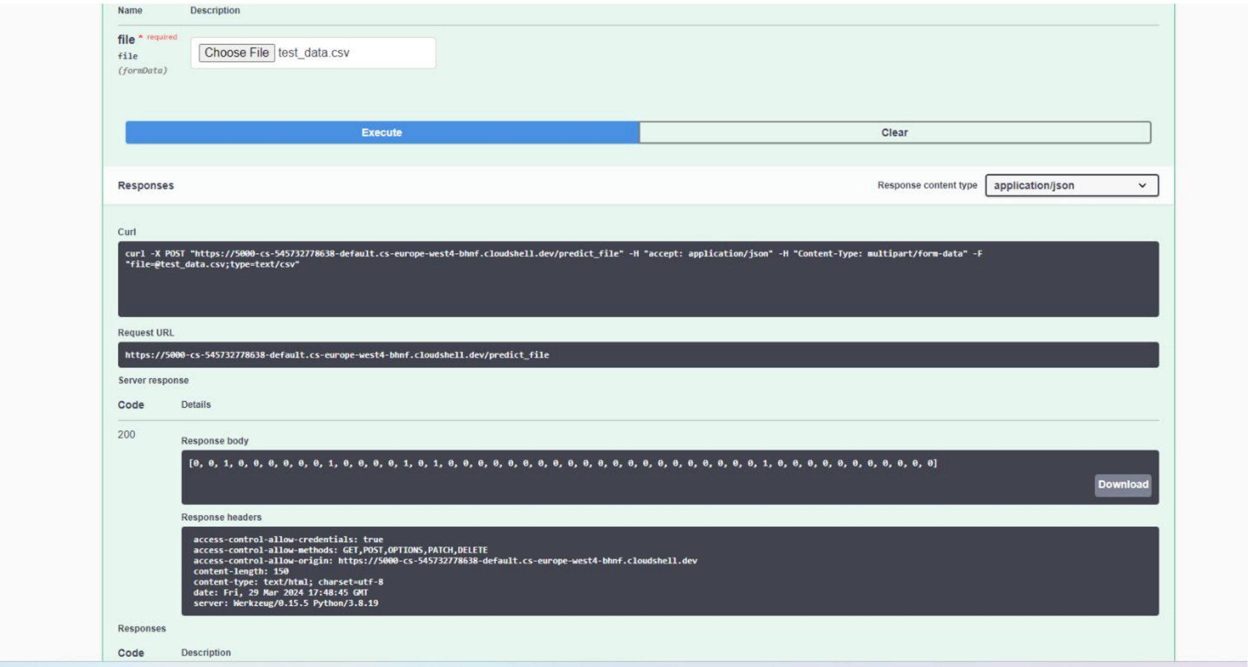
| Name | Description |
|---|---|
| file * required<br>file<br>(formData) | Choose File test_data.csv |

Responses

Response content type application/json

| Code | Description |
|---|---|
| 200 | Test file Prediction |

11. Then click Execute and the model would make the predictions. The results would be displayed upon execution as follows.



Curl

```
curl -X POST "https://5000-cs-545732778638-default.cs-europe-west4-bhnf.cloudshell.dev/predict_file" -H "accept: application/json" -H "Content-Type: multipart/form-data" -F
"file=@test_data.csv;type=text/csv"
```

Request URL

```
https://5000-cs-545732778638-default.cs-europe-west4-bhnf.cloudshell.dev/predict_file
```

Server response

Code Details

200

Response body

```
[0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

Response headers

```
access-control-allow-credentials: true
access-control-allow-methods: GET,POST,OPTIONS,PATCH,DELETE
access-control-allow-origin: https://5000-cs-545732778638-default.cs-europe-west4-bhnf.cloudshell.dev
content-length: 150
content-type: text/html; charset=utf-8
date: Fri, 29 Mar 2024 17:48:45 GMT
server: Werkzeug/0.15.5 Python/3.8.19
```

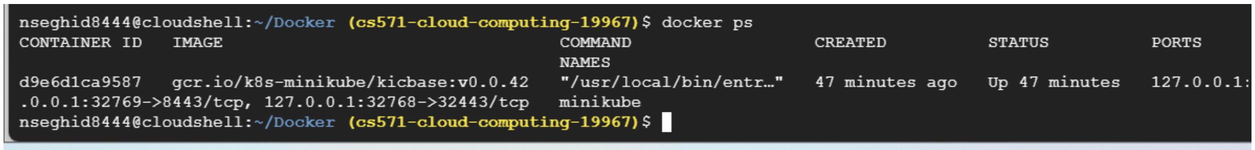
Responses

Code Description

## Step 6: Stopping/killing the running container.

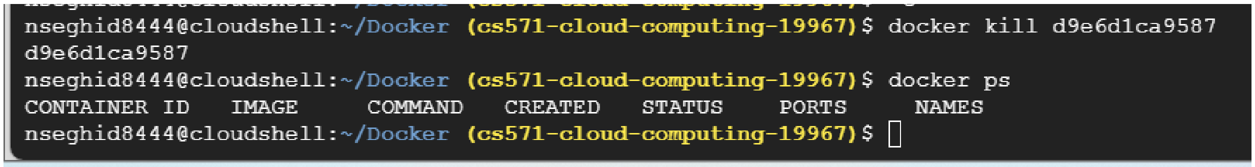
The last step left after running the application is to stop the running container. This can be done using the docker stop or kill command on the running container.

1. To see the list of running containers, use the docker ps command and select the running container ID to stop it.



```
nseghid8444@cloudshell:~/Docker (cs571-cloud-computing-19967)$ docker ps
CONTAINER ID   IMAGE                                 COMMAND                  CREATED          STATUS          PORTS
                                                      NAMES
d9e6d1ca9587   gcr.io/k8s-minikube/kicbase:v0.0.42   "/usr/local/bin/entr…"   47 minutes ago   Up 47 minutes   127.0.0.1:
.0.0.1:32769->8443/tcp, 127.0.0.1:32768->32443/tcp   minikube
nseghid8444@cloudshell:~/Docker (cs571-cloud-computing-19967)$ 
```

2. Use the command - docker kill to kill the running container as follows.
   docker kill <Container_ID>



```
nseghid8444@cloudshell:~/Docker (cs571-cloud-computing-19967)$ docker kill d9e6d1ca9587
d9e6d1ca9587
nseghid8444@cloudshell:~/Docker (cs571-cloud-computing-19967)$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
nseghid8444@cloudshell:~/Docker (cs571-cloud-computing-19967)$ 
```

## Updated Portfolio- Link to GitHub

https://github.com/niyat33/Cloud-Computing/blob/main/Kubernetes/Machine%20Learning/README.md