CS501- Week 10 Homework 1: Machine Learning on Kubernetes
Name: Feven Belay Araya
ID: 20027

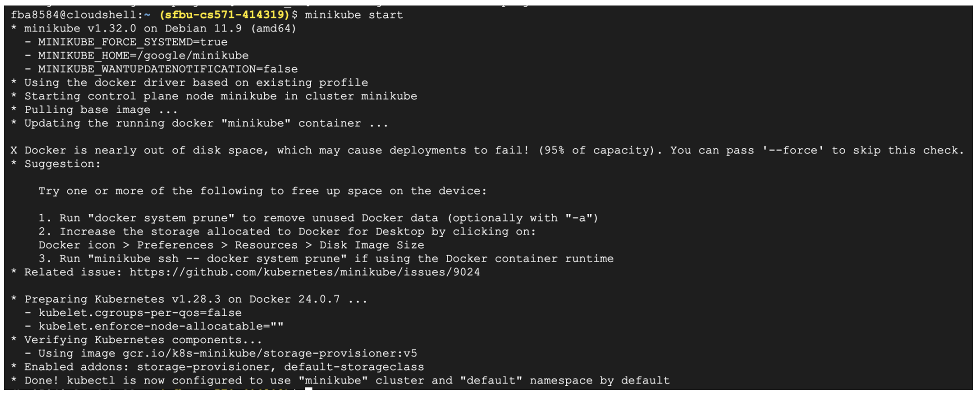
https://hc.labnet.sfbu.edu/~henry/sfbu/course/cloud_computing/genai/slide/exercise_kubernetes.html

Q2 ===> Machine Learning on Kubernetes

Machine Learning on Kubernetes

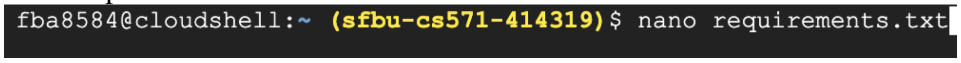
Creating and uploading necessary files in GCP- Cloud Shell Terminal

1. Start minikube in Google Cloud Platform



```
fba8584@cloudshell:~ (sfbu-cs571-414319)$ 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 ...

X Docker is nearly out of disk space, which may cause deployments to fail! (95% of capacity). You can pass '--force' to skip this check.
* Suggestion:

    Try one or more of the following to free up space on the device:

    1. Run "docker system prune" to remove unused Docker data (optionally with "-a")
    2. Increase the storage allocated to Docker for Desktop by clicking on:
    Docker icon > Preferences > Resources > Disk Image Size
    3. Run "minikube ssh -- docker system prune" if using the Docker container runtime
* Related issue: https://github.com/kubernetes/minikube/issues/9024

* Preparing Kubernetes v1.28.3 on Docker 24.0.7 ...
  - kubelet.cgroups-per-qos=false
  - kubelet.enforce-node-allocatable=""
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: storage-provisioner, default-storageclass
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

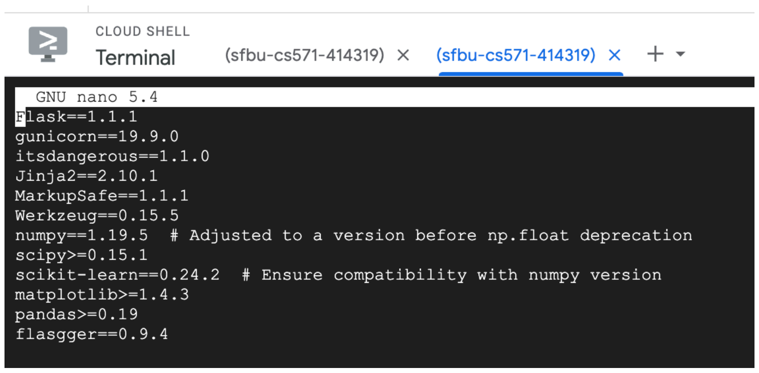
2. Create requirements.txt file using the following command
   - nano requirements.txt



```
fba8584@cloudshell:~ (sfbu-cs571-414319)$ nano requirements.txt
```

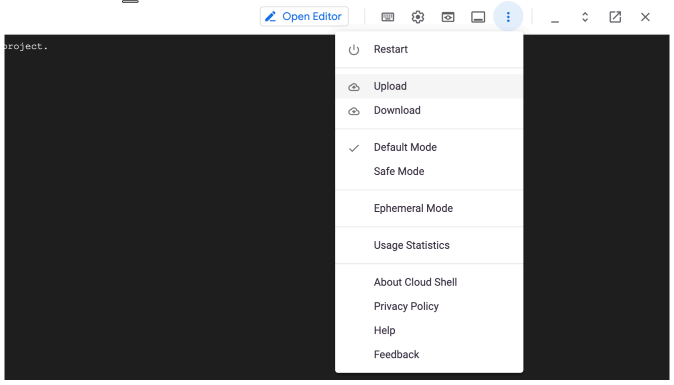
Then enter the following contents

```
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==1.19.5  # Adjusted to a version before np.float deprecation
scipy>=0.15.1
scikit-learn==0.24.2  # Ensure compatibility with numpy version
matplotlib>=1.4.3
pandas>=0.19
flasgger==0.9.4
```

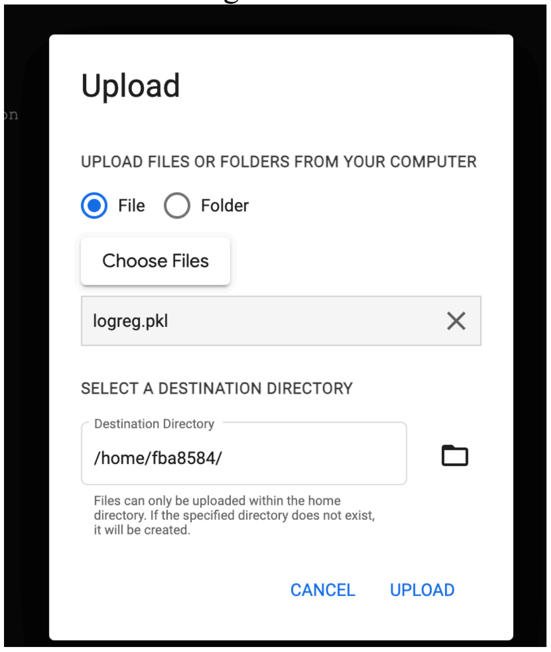


```
  GNU nano 5.4
Flask==1.1.1
gunicorn==19.9.0
itsdangerous==1.1.0
Jinja2==2.10.1
MarkupSafe==1.1.1
Werkzeug==0.15.5
numpy==1.19.5  # Adjusted to a version before np.float deprecation
scipy>=0.15.1
scikit-learn==0.24.2  # Ensure compatibility with numpy version
matplotlib>=1.4.3
pandas>=0.19
flasgger==0.9.4
```

3. Upload logreg.pkl file by clicking the three dots in the top-right part of the Cloud Shell Terminal and then choose upload



Then upload the logreg.pkl file as following


Upload
UPLOAD FILES OR FOLDERS FROM YOUR COMPUTER
File
Folder
Choose Files
SELECT A DESTINATION DIRECTORY
Destination Directory
CANCEL
UPLOAD

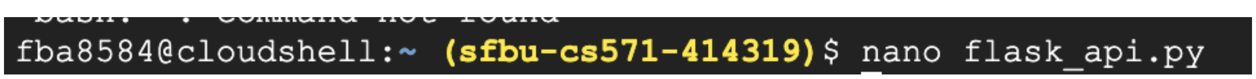
File [✓] Folder

logreg.pkl

Destination Directory: /home/fba8584/

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

4. Create flask_api.py file using the command
   - *nano flask_api.py*

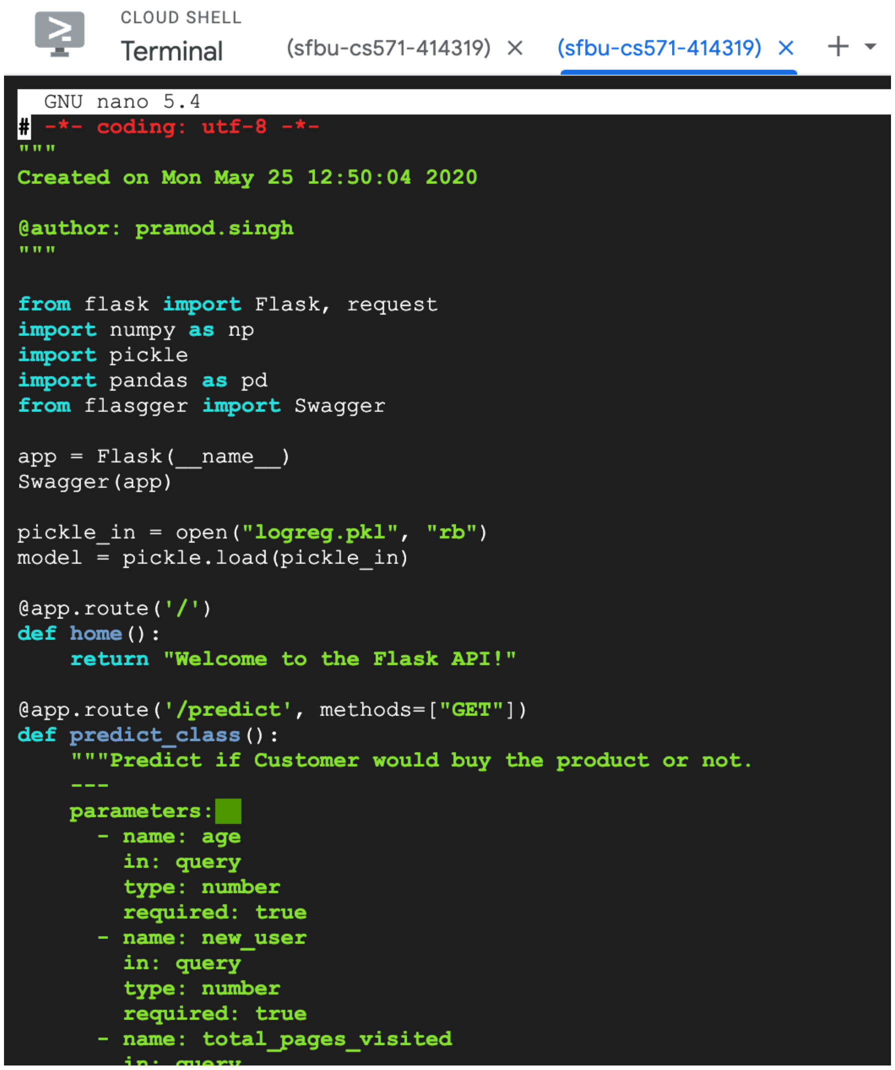


```
bash: : command not found
fba8584@cloudshell:~ (sfbu-cs571-414319)$ nano flask_api.py
```

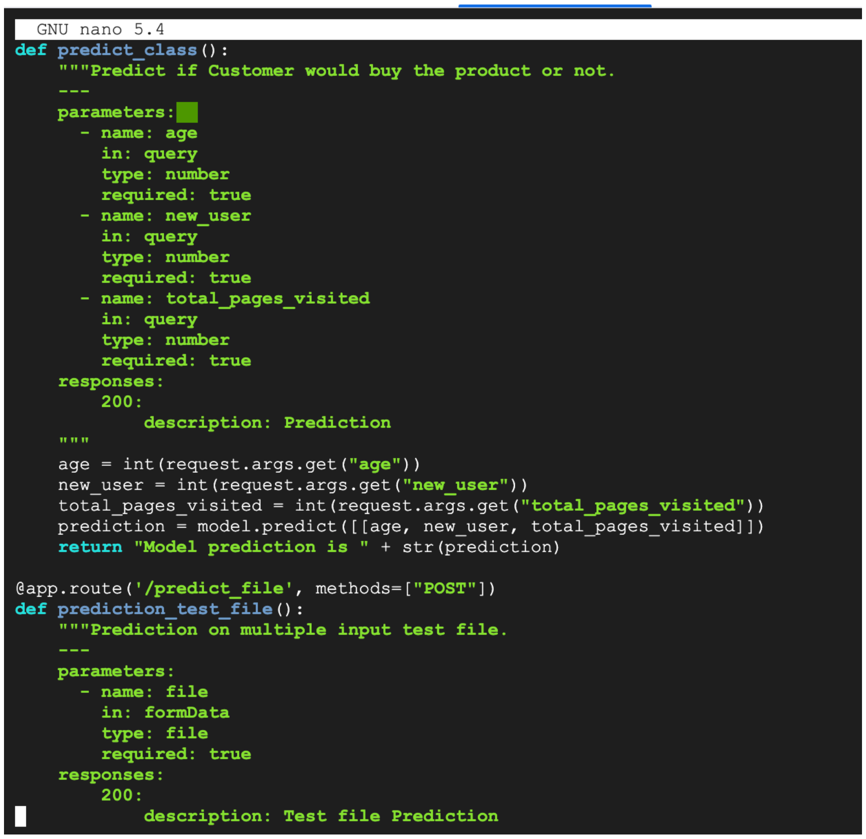
Then enter the following contents

```
# -*- 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.
```

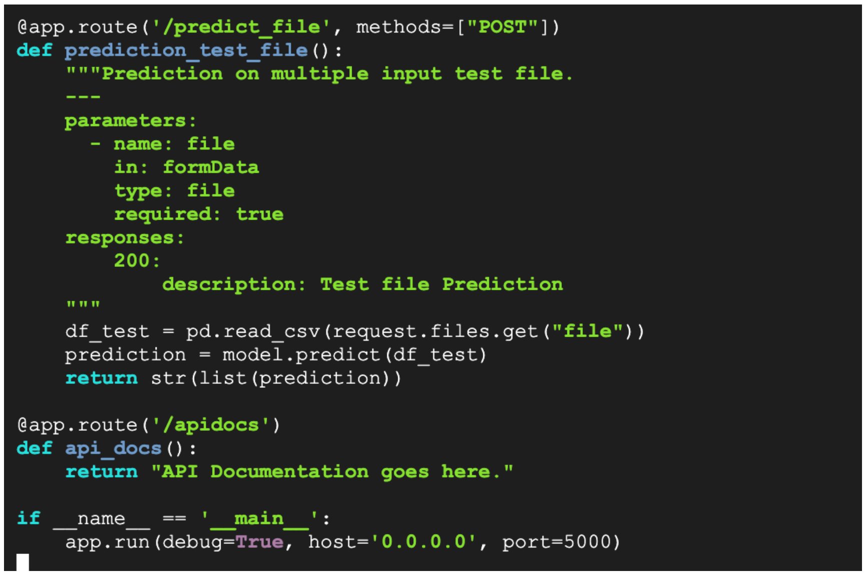
```
        ---
        parameters:
          - name: file
            in: formData
            type: file
            required: true
        responses:
            200:
                description: Test file Prediction
        """
        df_test = pd.read_csv(request.files.get("file"))
        prediction = model.predict(df_test)
        return str(list(prediction))

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



```
  GNU nano 5.4
# -*- 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
```



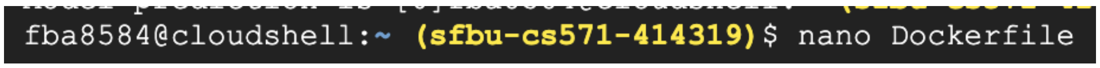
```
  GNU nano 5.4
def predict_class():
    """Predict if Customer would buy the product or not.
    ---
    parameters:
      - name: age
        in: query
        type: number
        required: true
      - name: new_user
        in: query
        type: number
        required: true
      - name: total_pages_visited
        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
```



```
@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))

@app.route('/apidocs')
def api_docs():
    return "API Documentation goes here."

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

Step 4: Dockerfile

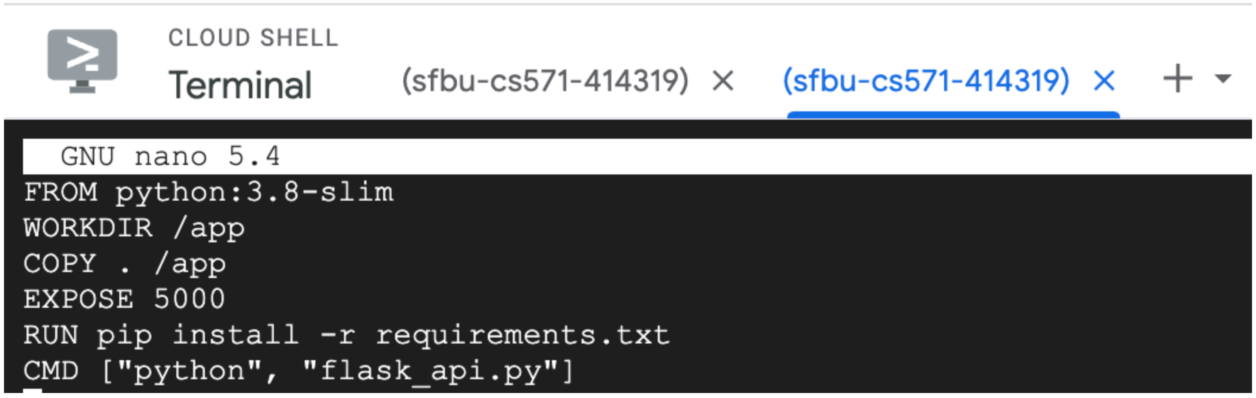
1. Create Dockerfile using command
   - *nano Dockerfile*



```
fba8584@cloudshell:~ (sfbu-cs571-414319)$ nano Dockerfile
```

Then enter the following content

FROM python:3.8-slim
WORKDIR /app
COPY . /app
EXPOSE 5000
RUN pip install -r requirements.txt
CMD ["python", "flask_api.py"]

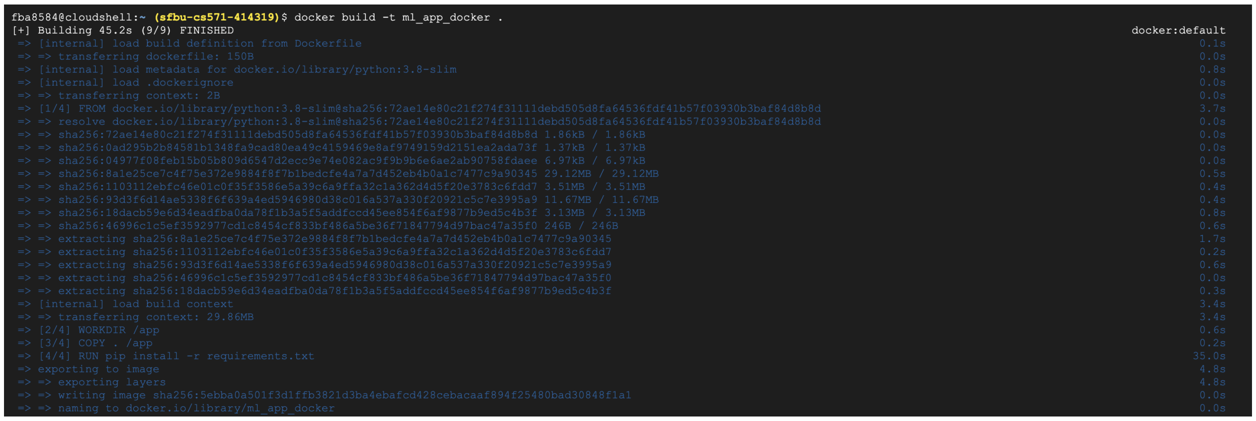


```
  GNU nano 5.4
FROM python:3.8-slim
WORKDIR /app
COPY . /app
EXPOSE 5000
RUN pip install -r requirements.txt
CMD ["python", "flask_api.py"]
```

1. 'FROM python:3.8-slim'
• This line sets the base image for the Docker image you are creating. It tells Docker to start with the 'python:3.8-slim' image, which is an official Python image with Python 3.8 installed on it. The 'slim' version is a smaller version of the image that has fewer packages pre-installed, making the image size smaller.
2. 'WORKDIR /app'
• This instruction sets the working directory within the Docker container to */app*. All subsequent commands will be executed in this directory within the container.
3. 'COPY . /app'
• This line copies everything from the current directory (on the host machine where you're running the Docker build command, indicated by the first**) into the */app directory inside the Docker image (the second ' /app*).
4. 'EXPOSE 5000'
• The 'EXPOSE' instruction informs Docker that the container listens on the specified network port at runtime. In this case, it tells Docker that the container will listen on port 5000. It's worth noting that this does not actually publish the port—it serves as documentation and is used by the 'docker run -p' command to map the container port to a port on the Docker host.
5. 'RUN pip install -r requirements.txt*
• This command tells Docker to run pip install' inside the container, which will install the Python dependencies listed in the requirements.tt file. These dependencies are necessary for the Flask application to run correctly.
6. CMD ["python", "flask_api.py"]'
• This is the command that will be executed by default when the Docker container starts. In this case, it's telling Docker to run 'flask_api.py using Python. This is the Flask application you want to run inside the container.

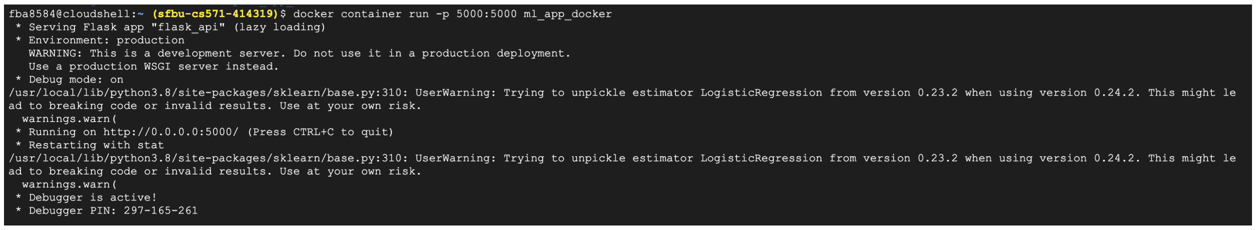
Step 5: Running the Docker Container

1. To build the docker image use the command
   - *sudo docker build -t ml_app_docker .*



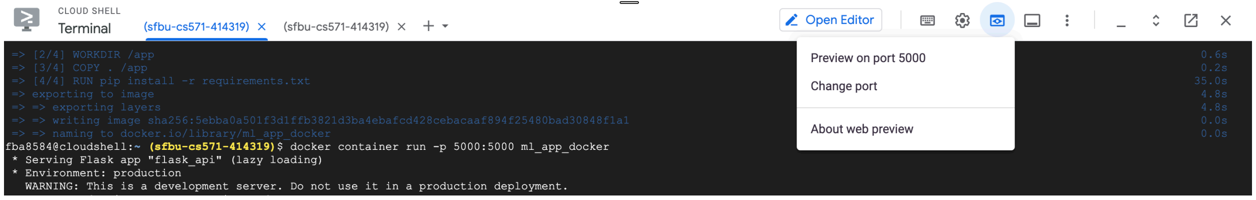
```
fba8584@cloudshell:~ (sfbu-cs571-414319)$ docker build -t ml_app_docker .
[+] Building 45.2s (9/9) FINISHED                                                                                                  docker:default
 => [internal] load build definition from Dockerfile                                                                                          0.1s
 => => transferring dockerfile: 150B                                                                                                          0.0s
 => [internal] load metadata for docker.io/library/python:3.8-slim                                                                            0.8s
 => [internal] load .dockerignore                                                                                                             0.0s
 => => transferring context: 2B                                                                                                               0.0s
 => [1/4] FROM docker.io/library/python:3.8-slim@sha256:72ae14e80c21f274f31111debd505d8fa64536fdf41b57f03930b3baf84d8b8d                      3.7s
 => => resolve docker.io/library/python:3.8-slim@sha256:72ae14e80c21f274f31111debd505d8fa64536fdf41b57f03930b3baf84d8b8d                      0.0s
 => => sha256:72ae14e80c21f274f31111debd505d8fa64536fdf41b57f03930b3baf84d8b8d 1.86kB / 1.86kB                                                0.0s
 => => sha256:0ad295b2b84581b1348fa9cad80ea49c4159469e8af9749159d2151ea2ada73f 1.37kB / 1.37kB                                                0.0s
 => => sha256:04977f08feb15b05b809d6547d2ecc9e74e082ac9f9b9b6e6ae2ab90758fdaee 6.97kB / 6.97kB                                                0.0s
 => => sha256:8a1e25ce7c4f75e372e9884f8f7b1bedcfe4a7a7d452eb4b0a1c7477c9a90345 29.12MB / 29.12MB                                              0.5s
 => => sha256:1103112ebfc46e01c0f35f3586e5a39c6a9ffa32c1a362d4d5f20e3783c6fdd7 3.51MB / 3.51MB                                                0.4s
 => => sha256:93d3f6d14ae5338f6f639a4ed5946980d38c016a537a330f20921c5c7e3995a9 11.67MB / 11.67MB                                              0.4s
 => => sha256:18dacb59e6d34eadfba0da78f1b3a5f5addfccd45ee854f6af9877b9ed5c4b3f 3.13MB / 3.13MB                                                0.8s
 => => sha256:46996c1c5ef3592977cd1c8454cf833bf486a5be36f71847794d97bac47a35f0 246B / 246B                                                    0.6s
 => => extracting sha256:8a1e25ce7c4f75e372e9884f8f7b1bedcfe4a7a7d452eb4b0a1c7477c9a90345                                                     1.7s
 => => extracting sha256:1103112ebfc46e01c0f35f3586e5a39c6a9ffa32c1a362d4d5f20e3783c6fdd7                                                     0.2s
 => => extracting sha256:93d3f6d14ae5338f6f639a4ed5946980d38c016a537a330f20921c5c7e3995a9                                                     0.6s
 => => extracting sha256:46996c1c5ef3592977cd1c8454cf833bf486a5be36f71847794d97bac47a35f0                                                     0.0s
 => => extracting sha256:18dacb59e6d34eadfba0da78f1b3a5f5addfccd45ee854f6af9877b9ed5c4b3f                                                     0.3s
 => [internal] load build context                                                                                                             3.4s
 => => transferring context: 29.86MB                                                                                                          3.4s
 => [2/4] WORKDIR /app                                                                                                                        0.6s
 => [3/4] COPY . /app                                                                                                                         0.2s
 => [4/4] RUN pip install -r requirements.txt                                                                                                35.0s
 => exporting to image                                                                                                                        4.8s
 => => exporting layers                                                                                                                       4.8s
 => => writing image sha256:5ebba0a501f3d1ffb3821d3ba4ebafcd428cebacaaf894f25480bad30848f1a1                                                  0.0s
 => => naming to docker.io/library/ml_app_docker                                                                                              0.0s
```

2. This command runs a Docker container from the ml_app_docker image:
   - *docker container run -p 5000:5000 ml_app_docker*



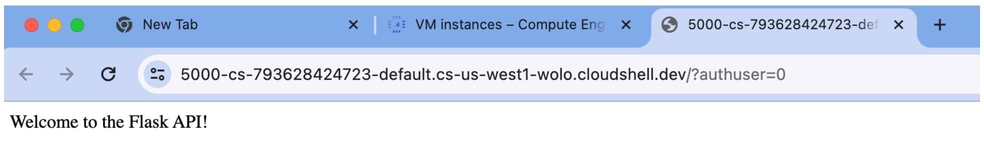
```
fba8584@cloudshell:~ (sfbu-cs571-414319)$ docker container run -p 5000:5000 ml_app_docker
 * Serving Flask app "flask_api" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
/usr/local/lib/python3.8/site-packages/sklearn/base.py:310: UserWarning: Trying to unpickle estimator LogisticRegression from version 0.23.2 when using version 0.24.2. This might le
ad to breaking 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 estimator LogisticRegression from version 0.23.2 when using version 0.24.2. This might le
ad to breaking code or invalid results. Use at your own risk.
  warnings.warn(
 * Debugger is active!
 * Debugger PIN: 297-165-261
```

3. In the right-upper side of the terminal click the eye shaped button and then click *Preview on port 5000*. Change port if it is not 5000 by default.



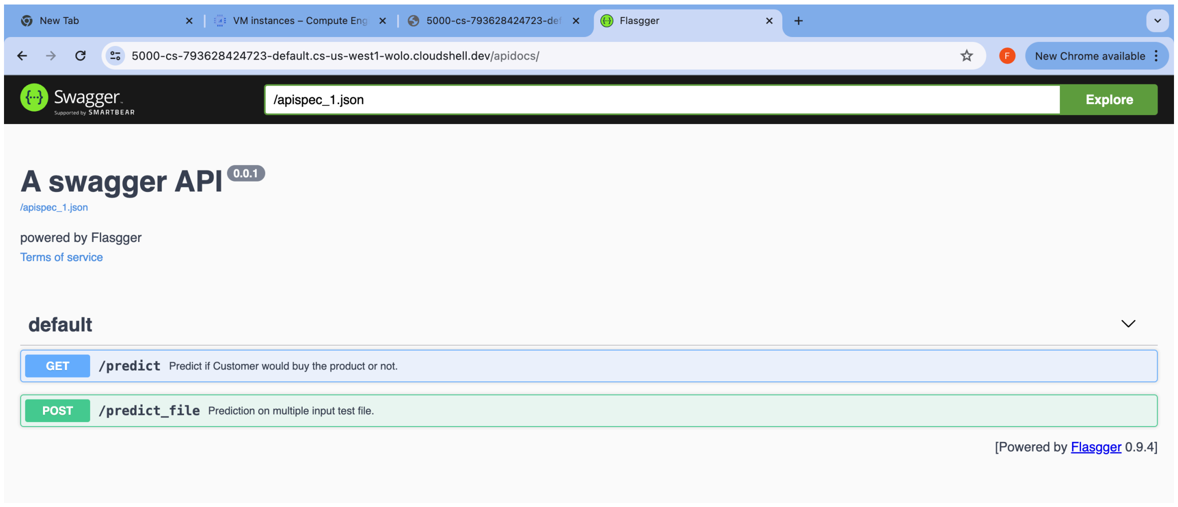
```
=> [2/4] WORKDIR /app                                                          0.6s
=> [3/4] COPY . /app                                                           0.2s
=> [4/4] RUN pip install -r requirements.txt                                  35.0s
=> exporting to image                                                          4.8s
=> => exporting layers                                                         4.8s
=> => writing image sha256:5ebba0a501f3d1ffb3821d3ba4ebafcd428cebacaaf894f25480bad30848f1a1  0.0s
=> => naming to docker.io/library/ml_app_docker                                0.0s
fba8584@cloudshell:~ (sfbu-cs571-414319)$ docker container run -p 5000:5000 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.
```

4. You will see this using the web preview.


New Tab
VM instances – Compute Eng
5000-cs-793628424723-def
5000-cs-793628424723-default.cs-us-west1-wolo.cloudshell.dev/?authuser=0

Welcome to the Flask API!

5. Add /apidocs/ at the end of the link to access the running ml- app as following
   - There are two tabs GET and POST.


New Tab
VM instances – Compute Eng
5000-cs-793628424723-def
Flagsger
5000-cs-793628424723-default.cs-us-west1-wolo.cloudshell.dev/apidocs/
New Chrome available
Swagger
/apispec_1.json
Explore

A swagger API 0.0.1
/apispec_1.json

powered by Flasgger
Terms of service

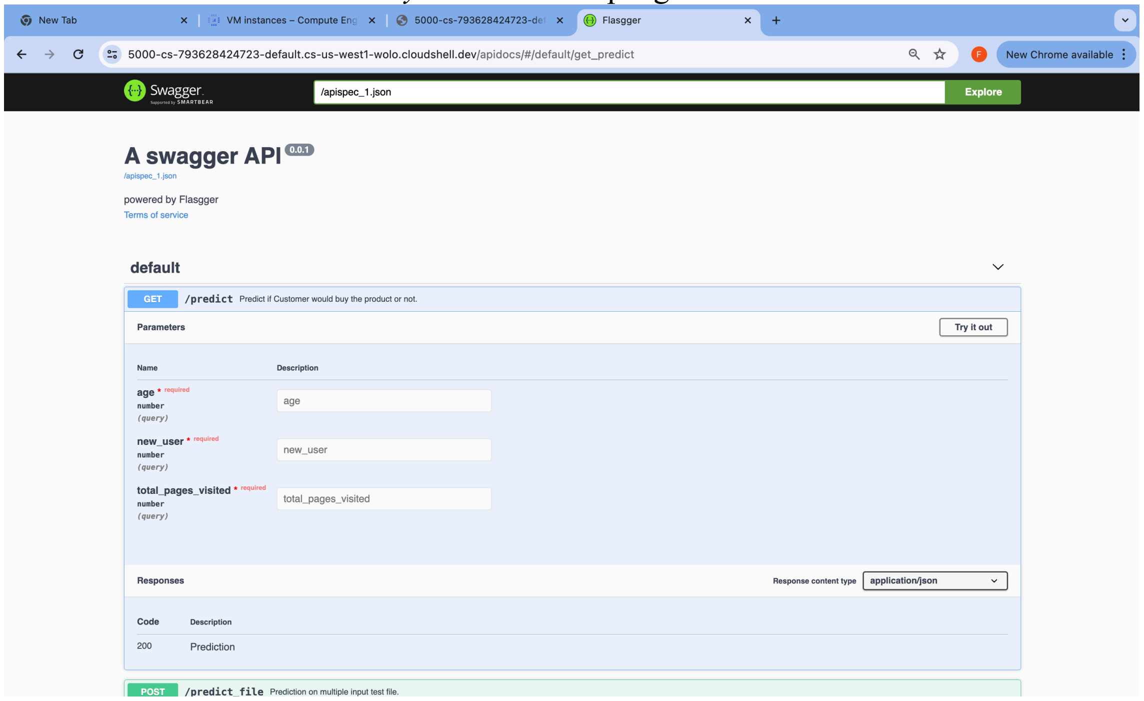
default

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

POST /predict_file Prediction on multiple input test file.

[Powered by Flasgger 0.9.4]

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


New Tab
VM instances – Compute Eng
5000-cs-793628424723-def
Flasgger
5000-cs-793628424723-default.cs-us-west1-wolo.cloudshell.dev/apidocs/#/default/get_predict
New Chrome available
Swagger
/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

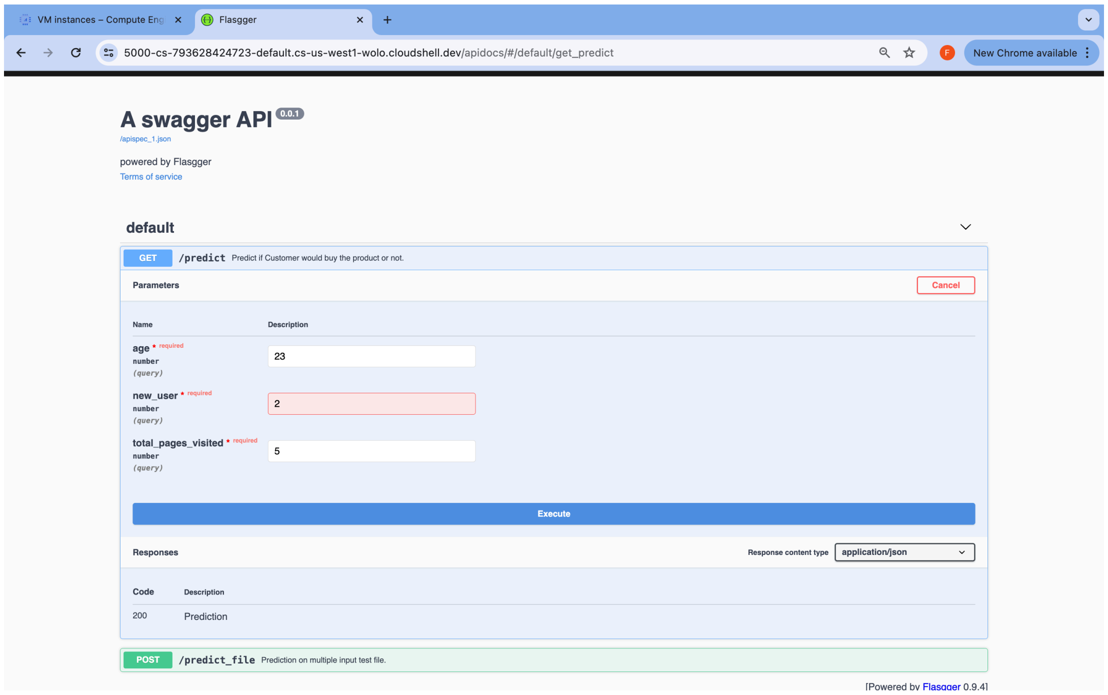
Name	Description
age * required number (query)	age
new_user * required number (query)	new_user
total_pages_visited * required number (query)	total_pages_visited

Responses — Response content type: application/json

Code	Description
200	Prediction

POST /predict_file Prediction on multiple input test file.

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


VM instances – Compute Eng
Flasgger
5000-cs-793628424723-default.cs-us-west1-wolo.cloudshell.dev/apidocs/#/default/get_predict
New Chrome available

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 number (query)	23
new_user * required number (query)	2
total_pages_visited * required number (query)	5

Responses — Response content type: application/json

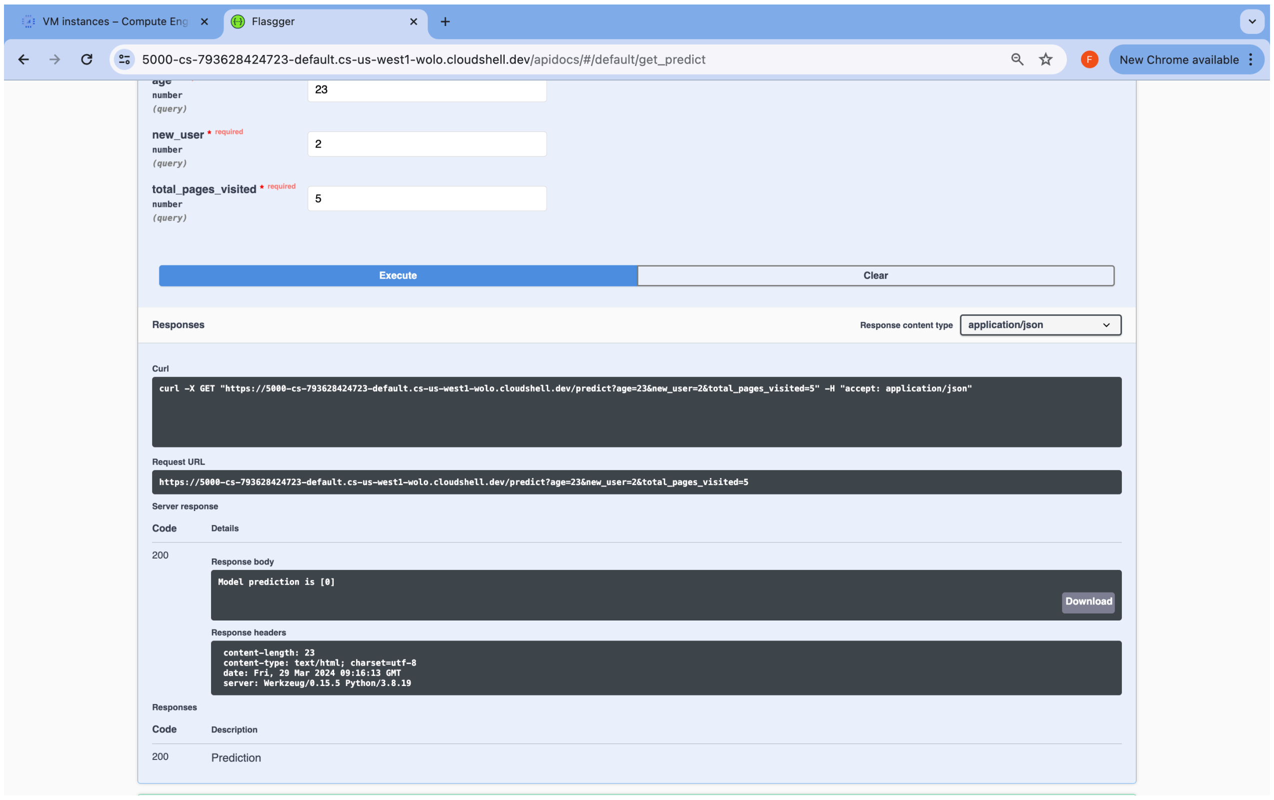
Code	Description
200	Prediction

POST /predict_file Prediction on multiple input test file.

[Powered by Flasgger 0.9.4]

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 following



5000-cs-793628424723-default.cs-us-west1-wolo.cloudshell.dev/apidocs/#/default/get_predict

age
number
(query)
23

new_user * required
number
(query)
2

total_pages_visited * required
number
(query)
5

Responses — Response content type: application/json

Curl
```
curl -X GET "https://5000-cs-793628424723-default.cs-us-west1-wolo.cloudshell.dev/predict?age=23&new_user=2&total_pages_visited=5" -H "accept: application/json"
```

Request URL
```
https://5000-cs-793628424723-default.cs-us-west1-wolo.cloudshell.dev/predict?age=23&new_user=2&total_pages_visited=5
```

Server response

Code	Details
200	Response body

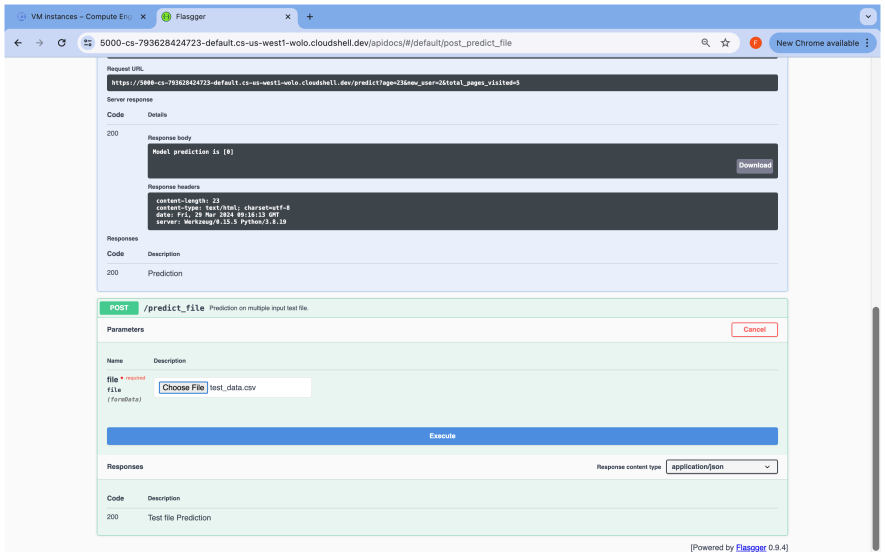
```
Model prediction is [0]
```

Response headers
```
content-length: 23
content-type: text/html; charset=utf-8
date: Fri, 29 Mar 2024 09:16:13 GMT
server: Werkzeug/0.15.5 Python/3.8.19
```

Responses

Code	Description
200	Prediction

9. The next prediction that can be done is for a group of customers (test data) via a post request.



5000-cs-793628424723-default.cs-us-west1-wolo.cloudshell.dev/apidocs/#/default/post_predict_file

Request URL
```
https://5000-cs-793628424723-default.cs-us-west1-wolo.cloudshell.dev/predict?age=23&new_user=2&total_pages_visited=5
```

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 09:16:13 GMT
server: Werkzeug/0.15.5 Python/3.8.19
```

Responses

Code	Description
200	Prediction

POST /predict_file Prediction on multiple input test file.

Parameters

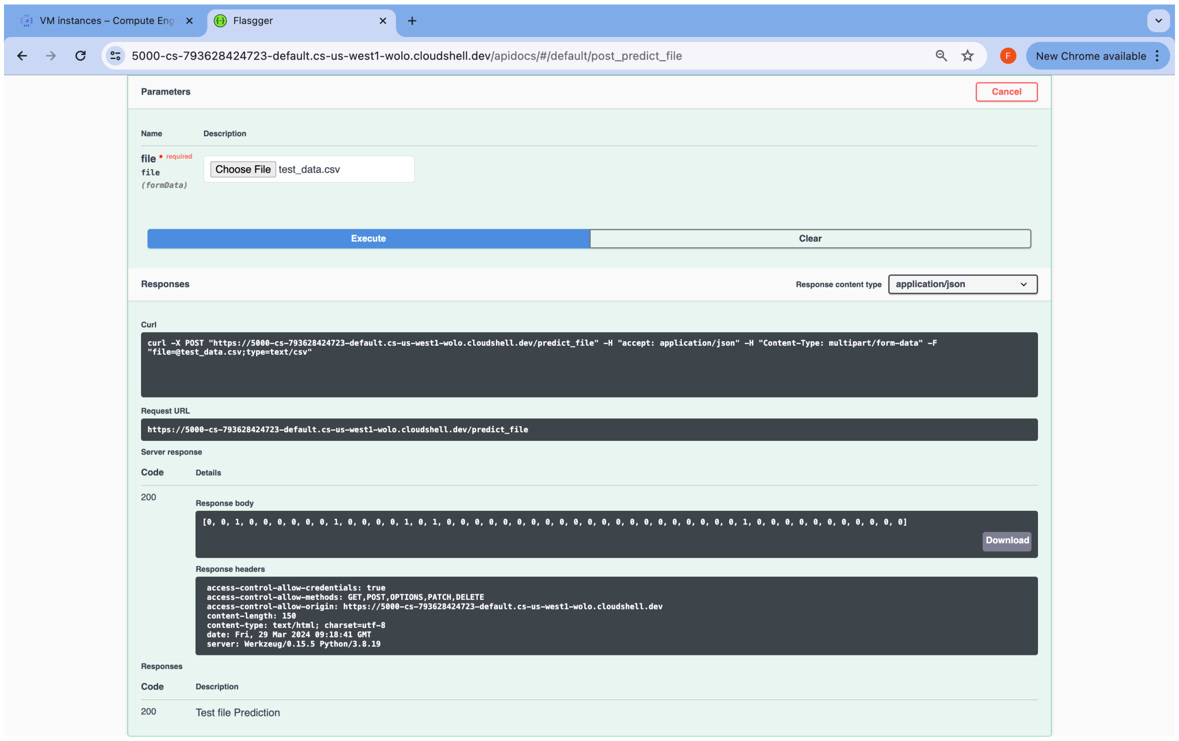
Name	Description
file * required file (formData)	test_data.csv

Responses — Response content type: application/json

Code	Description
200	Test file Prediction

[Powered by Flasgger 0.9.4]

10. Upload the test data file containing the same parameters in a similar order.
    The model would make the prediction, and the results would be displayed upon execute as following.



Curl

```
curl -X POST "https://5000-cs-793628424723-default.cs-us-west1-wolo.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-793628424723-default.cs-us-west1-wolo.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

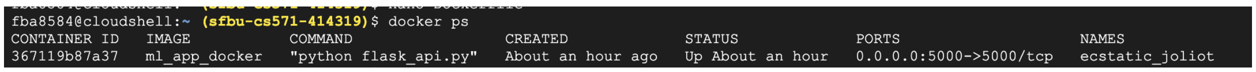
```
access-control-allow-credentials: true
access-control-allow-methods: GET,POST,OPTIONS,PATCH,DELETE
access-control-allow-origin: https://5000-cs-793628424723-default.cs-us-west1-wolo.cloudshell.dev
content-length: 150
content-type: text/html; charset=utf-8
date: Fri, 29 Mar 2024 09:18:41 GMT
server: Werkzeug/0.15.5 Python/3.8.19
```

Responses

Code	Description
200	Test file Prediction

Step 6: Stopping/killing the running container

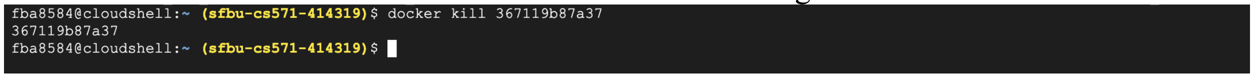
1. Use docker ps to list running Docker containers



```
fba8584@cloudshell:~ (sfbu-cs571-414319)$ docker ps
CONTAINER ID   IMAGE           COMMAND                 CREATED            STATUS            PORTS                      NAMES
367119b87a37   ml_app_docker   "python flask_api.py"   About an hour ago  Up About an hour  0.0.0.0:5000->5000/tcp     ecstatic_joliot
```

   - The CONTAINER_ID is given as 367119b87a37

2. Use the command
   - *docker kill <CONTAINER ID>* to kill the running container as follows.



```
fba8584@cloudshell:~ (sfbu-cs571-414319)$ docker kill 367119b87a37
367119b87a37
fba8584@cloudshell:~ (sfbu-cs571-414319)$ 
```