

Name: Fithi Ghebreamlak

ID: 20068

CS501- Week 10 Homework 1: Machine Learning on Kubernetes

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

Machine Learning on Kubernetes

Creating and uploading necessary files in GCP- Cloud Shell Terminal

1. Start minikube in Google Cloud Platform

```
fghebre408@cloudshell:~ (sfbu-cs571-429921) $ minikube start
* minikube v1.33.1 on Ubuntu 22.04 (amd64)
- MINIKUBE_FORCE_SYSTEMD=true
- MINIKUBE_HOME=/google/minikube
- MINIKUBE_WANTUPDATENOTIFICATION=false
* Automatically selected the docker driver. Other choices: none, ssh
* Using Docker driver with root privileges
* Starting "minikube" primary control-plane node in "minikube" cluster
* Pulling base image v0.0.44 ...
* Downloading Kubernetes v1.30.0 preload ...
  > preloaded-images-k8s-v18-v1...: 342.90 MiB / 342.90 MiB 100.00% 210.61
  > gcr.io/k8s-minikube/kicbase...: 481.58 MiB / 481.58 MiB 100.00% 108.94
* Creating docker container (CPUs=2, Memory=4000MB) ...
* Preparing Kubernetes v1.30.0 on Docker 26.1.1 ...
- kubelet.cgroups-per-qos=false
- kubelet.enforce-node-allocatable=""
- Generating certificates and keys ...
- Booting up control plane ...
- Configuring RBAC rules ...
* Configuring bridge CNI (Container Networking Interface) ...
* 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
fghebre408@cloudshell:~ (sfbu-cs571-429921) $
```

2. Create requirements.txt file using the following command

- sudo vim requirements.txt

Then enter the following contents

Flask==1.1.1

unicorn==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

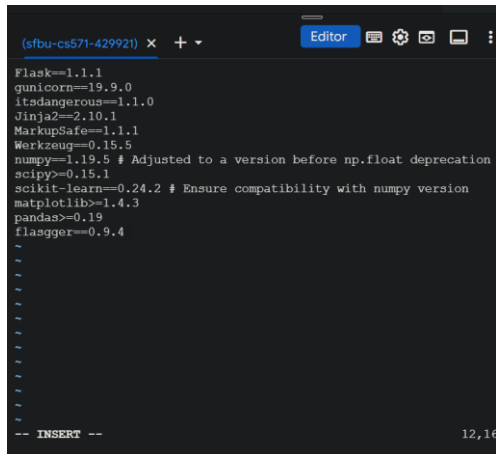
scipy>=0.15.1

scikit-learn==0.24.2 # Ensure compatibility with numpy version

matplotlib>=1.4.3

pandas>=0.19

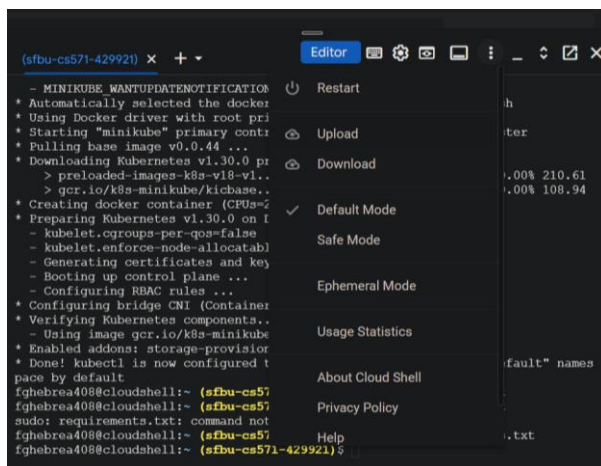
flask==0.9.4



```
(sfbu-cs571-429921) x + v Editor
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
flask==0.9.4
~
~
~
~
~
~
~
~
~
~
-- INSERT -- 12,16
```

3. Upload logreg.pkl file by clicking the three dots in the top-right part of the Cloud Shell

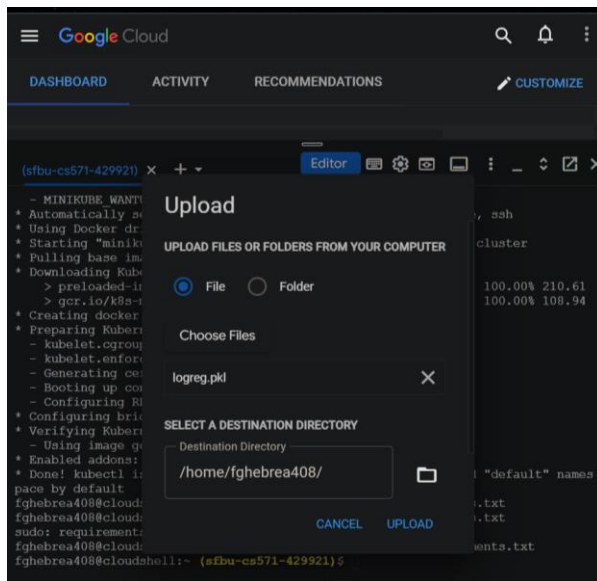
Terminal and then choose upload



```
(sfbu-cs571-429921) x + v Editor
- MINIKUBE_WANTUPDATENOTIFICATION
* Automatically selected the docker:
* Using Docker driver with root pri
* Starting "minikube" primary control plane node
* Pulling base image v0.0.44 ...
* Downloading Kubernetes v1.30.0 pr
  > preloaded-images-k8s-v18-v1...
  > gcr.io/k8s-minikube/kicbase...
* Creating docker container (CPUs=2
* Preparing Kubernetes v1.30.0 on f
  - kubelet.cgroups-per-qos=false
  - kubelet.enforce-node-allocatabl
  - Generating certificates and key
  - Booting up control plane ...
  - Configuring RBAC rules ...
* Configuring bridge CNI (Container
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube
* Enabled addons: storage-provisor
* Done! kubect1 is now configured t
pace by default
fghebre408@cloudshell:~ (sfbu-cs57
fghebre408@cloudshell:~ (sfbu-cs57
sudo: requirements.txt: command not
fghebre408@cloudshell:~ (sfbu-cs57
fghebre408@cloudshell:~ (sfbu-cs571-429921) $

Restart
Upload
Download
Default Mode
Safe Mode
Ephemeral Mode
Usage Statistics
About Cloud Shell
Privacy Policy
Help
```

Then upload the logreg.pkl file as following



4. Create flask_api.py file using the command
- sudo vim 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.

```

```

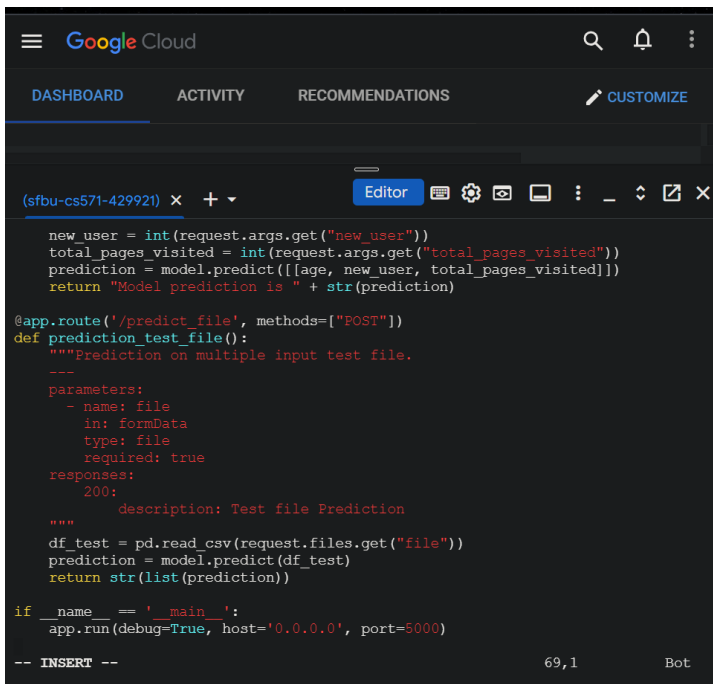
---

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)

```



```

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)

-- INSERT --

```

Step 4: Dockerfile

1. Create Dockerfile using command

- sudo vimi Dockerfile

Then enter the following content

FROM python:3.8-slim

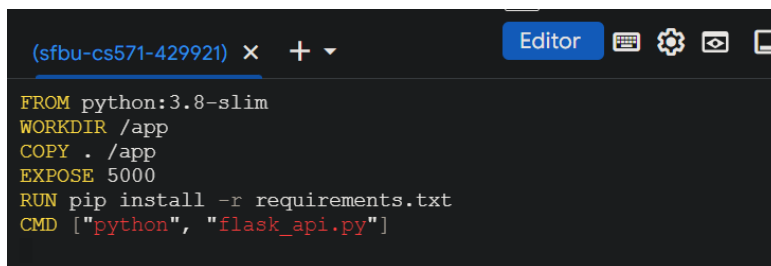
WORKDIR /app

COPY . /app

EXPOSE 5000

RUN pip install -r requirements.txt

CMD ["python", "flask_api.py"]

A screenshot of a code editor window titled '(sfbu-cs571-429921)'. The editor contains a Dockerfile with the following content: FROM python:3.8-slim, WORKDIR /app, COPY . /app, EXPOSE 5000, RUN pip install -r requirements.txt, and CMD ["python", "flask_api.py"]. The editor has a dark theme and a toolbar with icons for undo, redo, search, and other editing functions.

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.txt file. These dependencies are necessary for the Flask application to run correctly.

6. CMD ["python", "flask_api.py"]

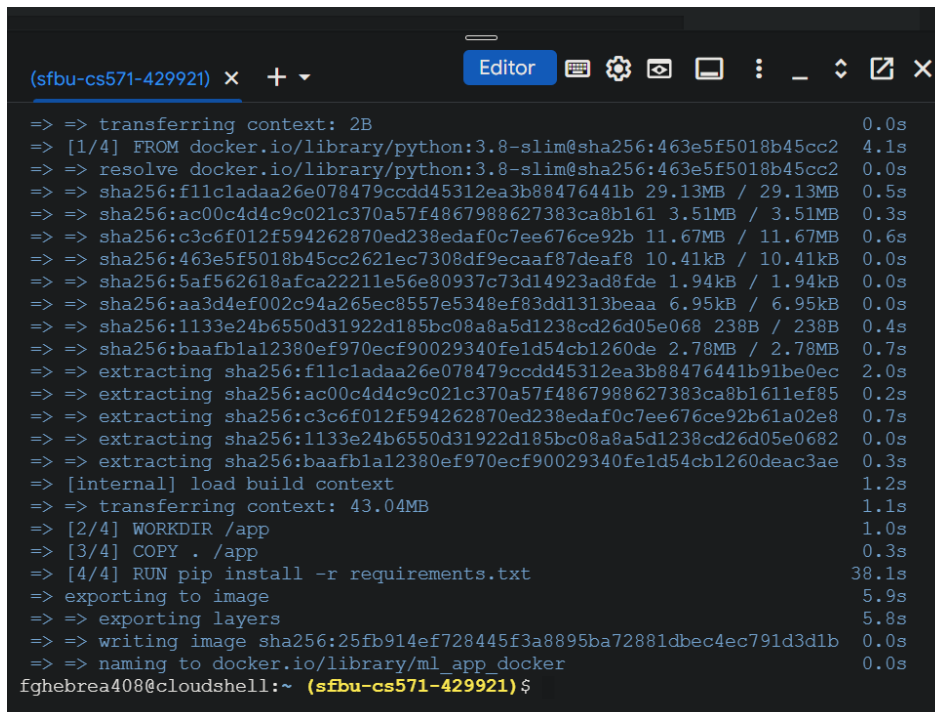
- This is the command that will be executed by default when the Docker container starts. In this case, it's telling

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 .



```
(sfbu-cs571-429921) x + v Editor
=> => transferring context: 2B 0.0s
=> [1/4] FROM docker.io/library/python:3.8-slim@sha256:463e5f5018b45cc2 4.1s
=> => resolve docker.io/library/python:3.8-slim@sha256:463e5f5018b45cc2 0.0s
=> => sha256:f11cladaa26e078479ccdd45312ea3b88476441b 29.13MB / 29.13MB 0.5s
=> => sha256:ac00c4d4c9c021c370a57f4867988627383ca8b161 3.51MB / 3.51MB 0.3s
=> => sha256:c3c6f012f594262870ed238edaf0c7ee676ce92b 11.67MB / 11.67MB 0.6s
=> => sha256:463e5f5018b45cc2621ec7308df9ecaaf87deaf8 10.41kB / 10.41kB 0.0s
=> => sha256:5af562618afca22211e56e80937c73d14923ad8fde 1.94kB / 1.94kB 0.0s
=> => sha256:aa3d4ef002c94a265ec8557e5348ef83dd1313beaa 6.95kB / 6.95kB 0.0s
=> => sha256:1133e24b6550d31922d185bc08a8a5d1238cd26d05e068 238B / 238B 0.4s
=> => sha256:baafb1a12380ef970ecf90029340fe1d54cb1260de 2.78MB / 2.78MB 0.7s
=> => extracting sha256:f11cladaa26e078479ccdd45312ea3b88476441b91be0ec 2.0s
=> => extracting sha256:ac00c4d4c9c021c370a57f4867988627383ca8b1611ef85 0.2s
=> => extracting sha256:c3c6f012f594262870ed238edaf0c7ee676ce92b61a02e8 0.7s
=> => extracting sha256:1133e24b6550d31922d185bc08a8a5d1238cd26d05e0682 0.0s
=> => extracting sha256:baafb1a12380ef970ecf90029340fe1d54cb1260deac3ae 0.3s
=> [internal] load build context 1.2s
=> => transferring context: 43.04MB 1.1s
=> [2/4] WORKDIR /app 1.0s
=> [3/4] COPY . /app 0.3s
=> [4/4] RUN pip install -r requirements.txt 38.1s
=> exporting to image 5.9s
=> => exporting layers 5.8s
=> => writing image sha256:25fb914ef728445f3a8895ba72881dbec4ec791d3d1b 0.0s
=> => naming to docker.io/library/ml_app_docker 0.0s
fghebre408@cloudshell:~ (sfbu-cs571-429921) $
```

2. This command runs a Docker container from the ml_app_docker image:

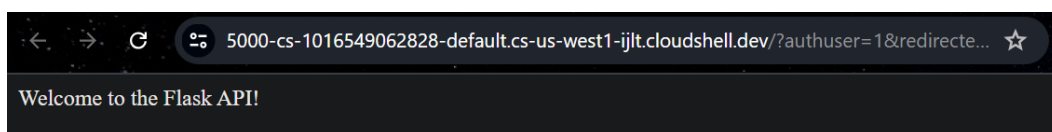
- docker container run -p 5000:5000 ml_app_docker

```
(sfbu-cs571-429921) x + - Editor
=> => exporting layers 5.8s
=> => writing image sha256:25fb914ef728445f3a8895ba72881dbec4ec791d3d1b 0.0s
=> => naming to docker.io/library/ml_app_docker 0.0s
fghebre408@cloudshell:~ (sfbu-cs571-429921)$ 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 1.2.2 when using version
0.24.2. This might lead 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 1.2.2 when using version
0.24.2. This might lead to breaking code or invalid results. Use at your own risk.
warnings.warn(
* Debugger is active!
* Debugger PIN: 501-503-373
```

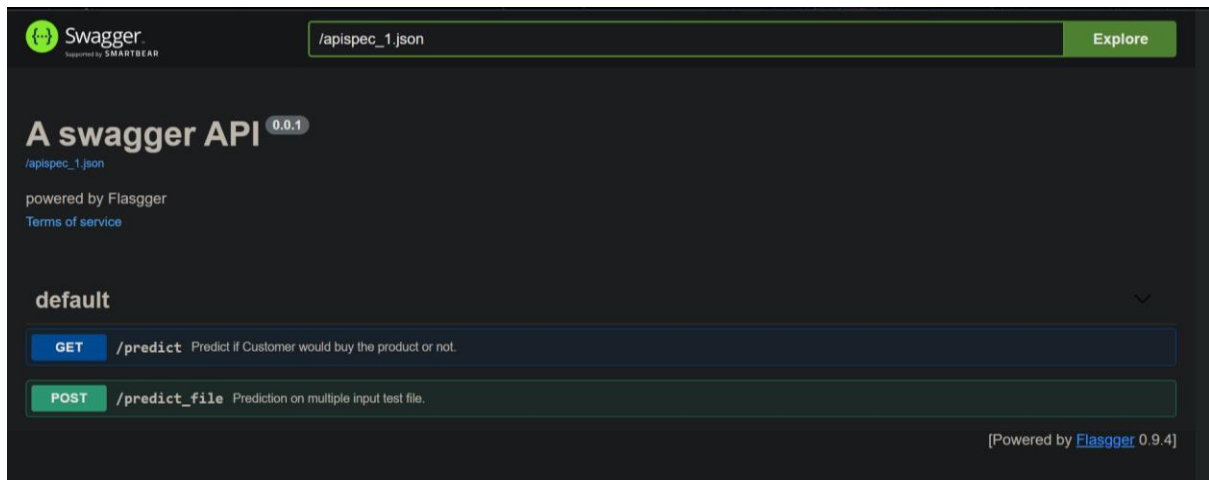
3. In the right-upper side of the terminal click the eye shaped button and then click Preview on port 5000. Change port if it is not 5000 by default.

```
(sfbu-cs571-429921) x + - Editor
fghebre408@cloudshell:~ (sfbu-cs571-429921)$ 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 1.2.2 when using version
0.24.2. This might lead 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 1.2.2 when using version
0.24.2. This might lead to breaking code or invalid results. Use at your own risk.
warnings.warn(
* Debugger is active!
* Debugger PIN: 501-503-373
```

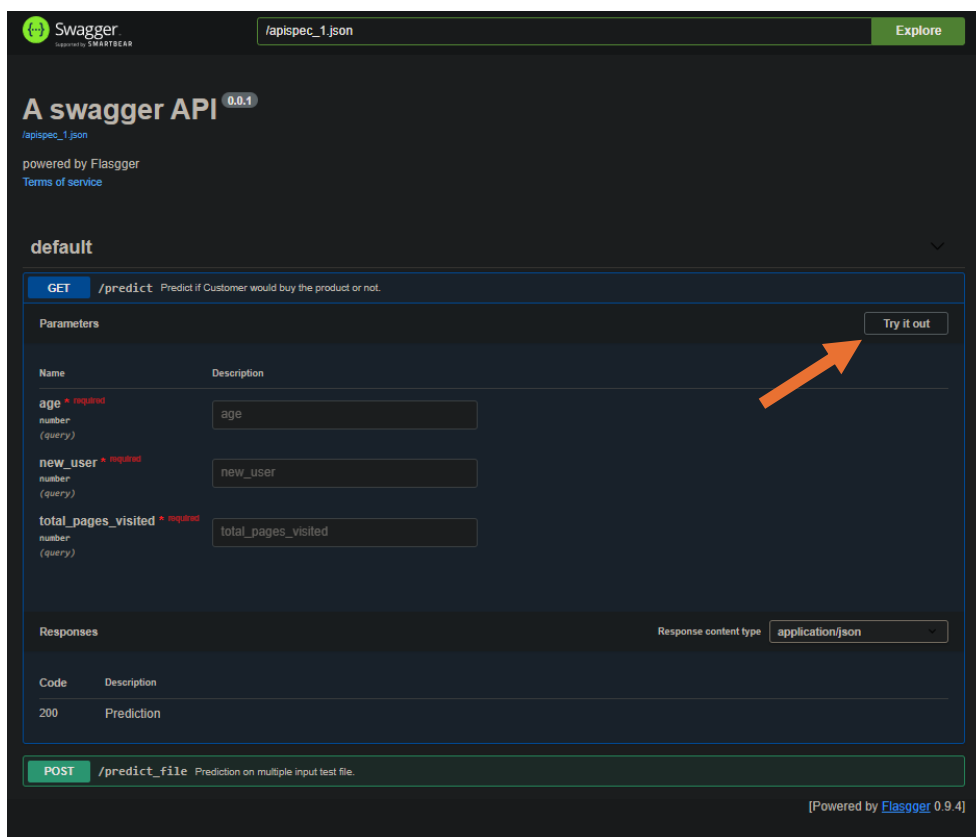
4. You will see this using the web preview.



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



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



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

default

GET

/predict

Predict if Customer would buy the product or not.

Parameters

Name	Description
age * required number (query)	<input type="text" value="24"/>
new_user * required number (query)	<input type="text" value="3"/>
total_pages_visited * required number (query)	<input type="text" value="5"/>

Execute

Responses

Code	Description	Response content type
200	Prediction	

POST

/predict_file

Prediction on multiple input test file.

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

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

Execute Clear

Responses Response content type: application/json

Curl

```
curl -X GET "https://9000-cs-101654062828-default.cs-us-west1-ijlt.cloudshell.dev/predict?age=23&new_user=2&total_pages_visited=5" -H "accept: application/json"
```

Request URL

```
https://9000-cs-101654062828-default.cs-us-west1-ijlt.cloudshell.dev/predict?age=23&new_user=2&total_pages_visited=5
```

Server response

Code	Details
200	<p>Response body</p> <pre>{ "prediction": 0 }</pre> <p>Response headers</p> <pre>content-length: 23 content-security-policy: frame-ancestors 'self' https://90-cs-101654062828-default.cs-us-west1-ijlt.cloudshell.dev https://cc-101654062828-default.cs-us-west1-ijlt.cloudshell.dev https://ids.cloud.google.com https://idm.cloud.google.com https://rds.cloud.google.com https://console.cloud.google.com date: Fri, 19 Jul 2024 22:31:15 GMT server: Werkzeug/0.15.1 Python/3.8.10</pre>

Responses

Code	Description
200	Prediction

POST /predict_file Prediction on multiple input test file.

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

POST /predict_file Prediction on multiple input test file.

Parameters

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

Execute

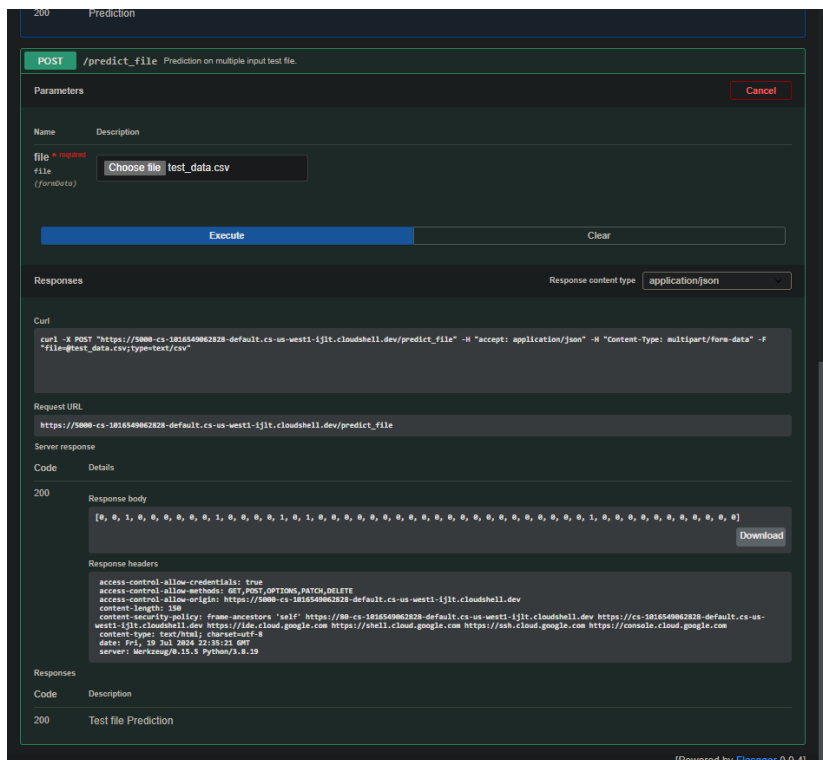
Responses Response content type: application/json

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.



Step 6: Stopping/killing the running container

1. Use docker ps to list running Docker containers

```
fghebre408@cloudshell:~ (sfbu-cs571-429921) $ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
ae59557c65ee   gcr.io/k8s-minikube/kicbase:v0.0.44 "/usr/local/bin/entr..." About an hour ago Up About an hour 127.0.0.1:32768->22/tcp, 127.0.0.1:32769->2376/tcp, 127.0.0.1:32770->5000/tcp, 127.0.0.1:32771->8443/tcp, 127.0.0.1:32772->32443/tcp minikube
fghebre408@cloudshell:~ (sfbu-cs571-429921) $
```

- The CONTAINER_ID is given as 367119b87a37

2. Use the command

- docker kill <CONTAINER ID> to kill the running container as follows.

```
fghebre408@cloudshell:~ (sfbu-cs571-429921) $ docker kill ae59557c65ee
fghebre408@cloudshell:~ (sfbu-cs571-429921) $
```

Updating Portfolio- [GitHub link](#)