Machine Learning on Kubernetes

CS571 - Cloud Computing Infrastructure Project

Saron Haile 20069

Table of contents

01

Introduction

In this project, we implemented a machine learning application on Kubernetes. The application uses a logistic regression model to predict whether a customer would buy a product based on given parameters. The entire setup is containerized using Docker, and the model is served using a Flask API.

Design

Architecture Overview:

1. **Flask API** : Provides endpoints to make predictions.
2. **Docker** : Containerised the application for consistency across different environments.
3. **Kubernetes (Minikube)** : Manages the deployment, scaling, and operations of the application.

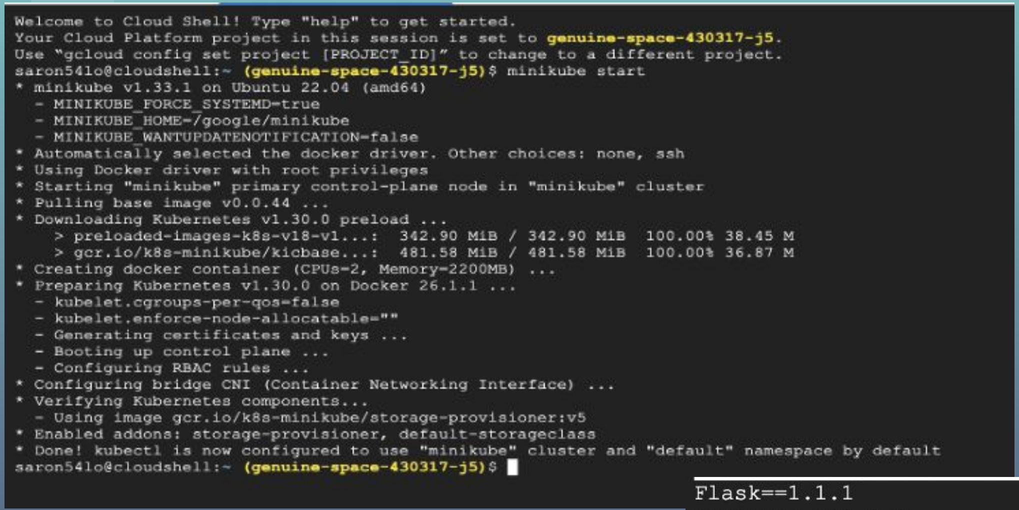
Key Components:

- requirements.txt: Lists the dependencies.
- flask_api.py: Contains the Flask application code.
- Dockerfile: Instructions to create the Docker image.
- logreg.pkl: Pre-trained logistic regression model.

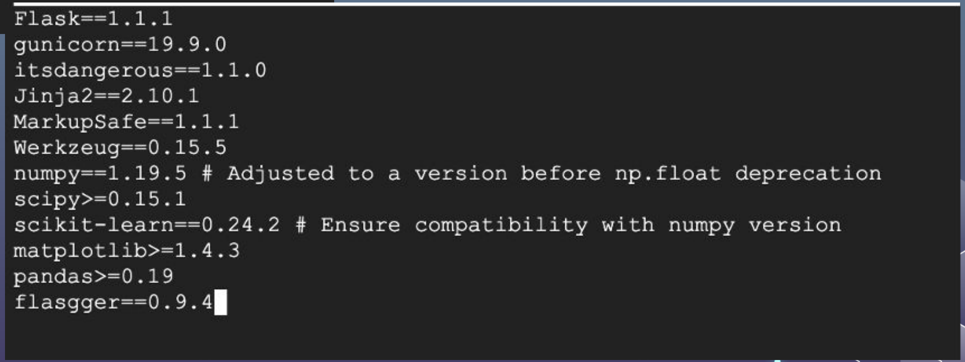
03

Implementation and Testing

Step 1: Start Minicube and create requirements.txt file

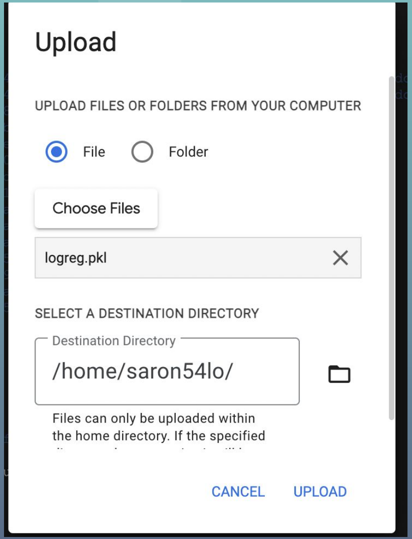


```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to genuine-space-430317-j5.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
saron54lo@cloudshell:~ (genuine-space-430317-j5)$ minikube start
* minikube v1.33.1 on 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% 38.45 M
    > gcr.io/k8s-minikube/kicbase...:  481.58 MiB / 481.58 MiB  100.00% 36.87 M
* Creating docker container (CPUs=2, Memory=2200MB) ...
* 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
saron54lo@cloudshell:~ (genuine-space-430317-j5)$
```



```
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
```

Step 2: Upload the logreg.pkl file and create flask_api.py file



Upload

UPLOAD FILES OR FOLDERS FROM YOUR COMPUTER

[✓ File] Folder

Choose Files

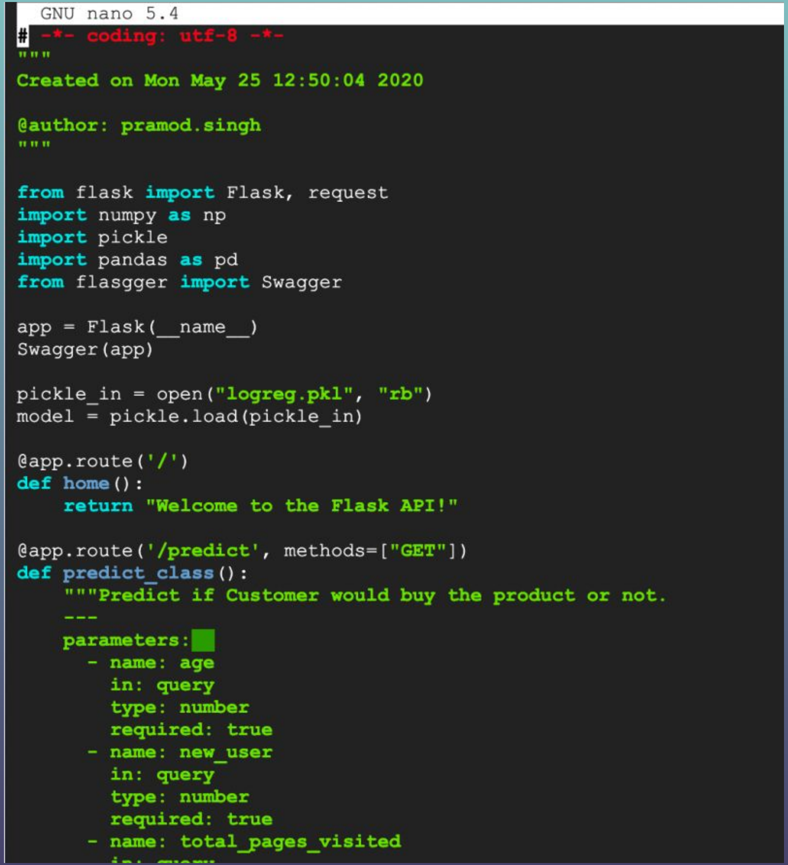
logreg.pkl

SELECT A DESTINATION DIRECTORY

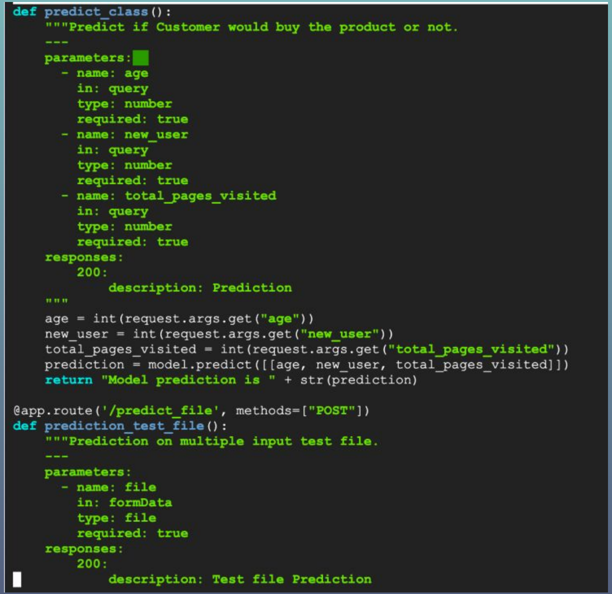
Destination Directory: /home/saron54lo/

Files can only be uploaded within the home directory. If the specified

CANCEL UPLOAD

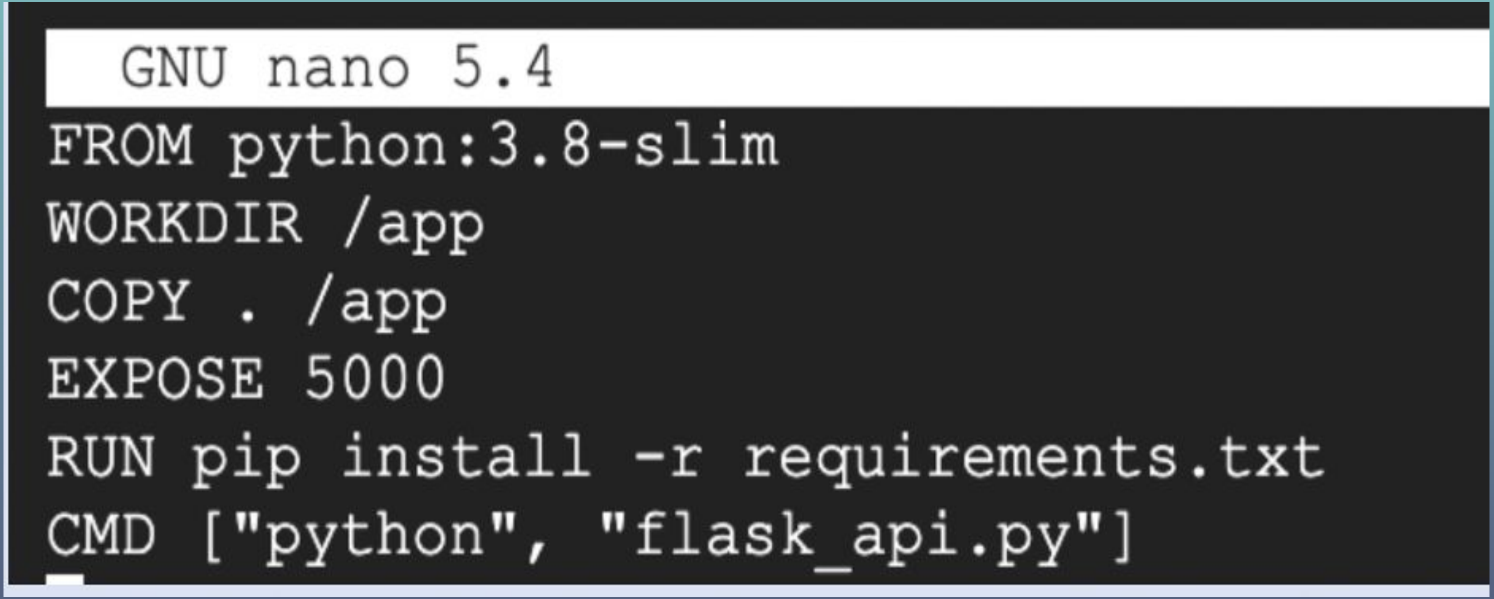


```
GNU nano 5.4
# -*- coding: utf-8 -*-
"""
Created on Mon May 25 12:50:04 2020

@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
```



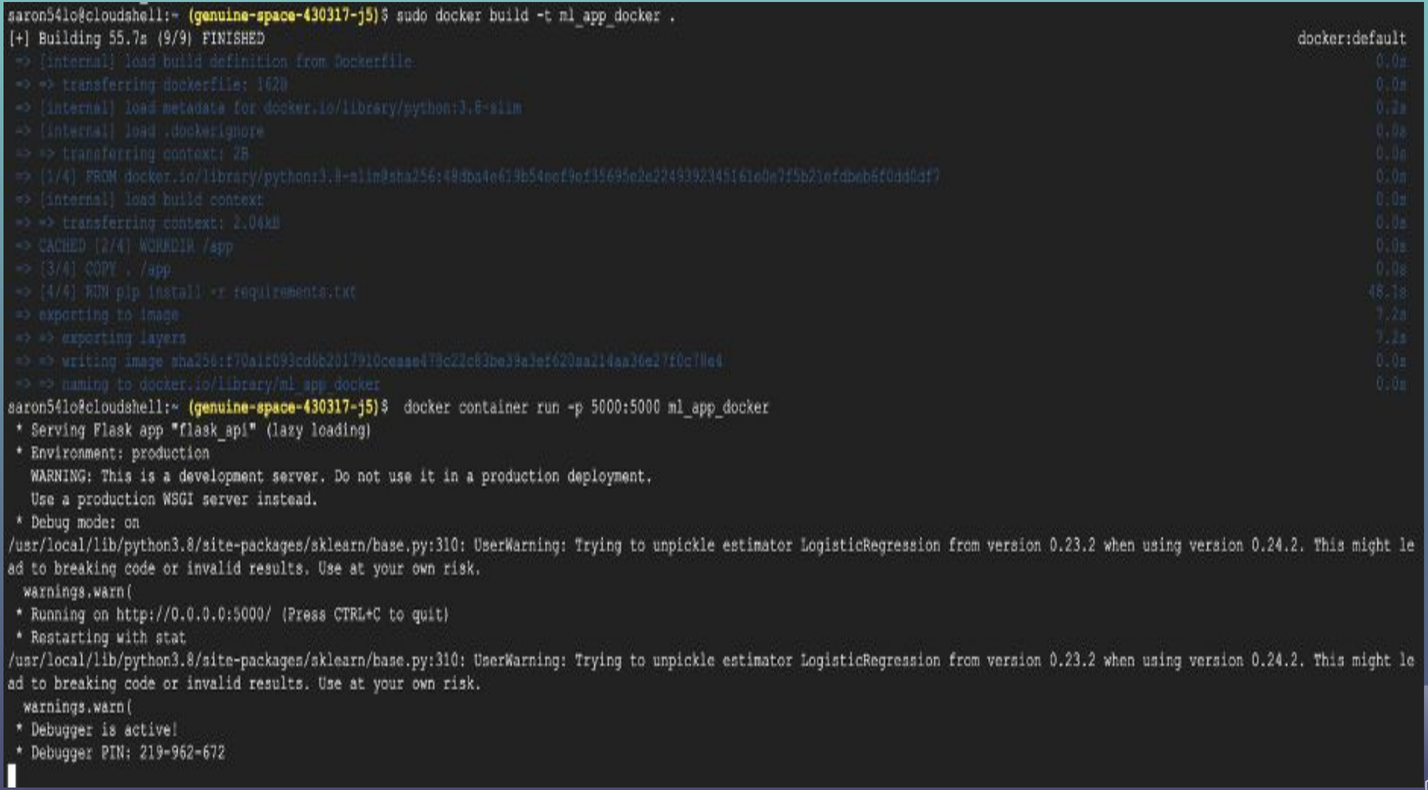
```
def predict_class():
    """Predict if Customer would buy the product or not.
    ---
    parameters:
      - name: age
        in: query
        type: number
        required: true
      - name: new_user
        in: query
        type: number
        required: true
      - name: total_pages_visited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

Step 3: Create Dockerfile



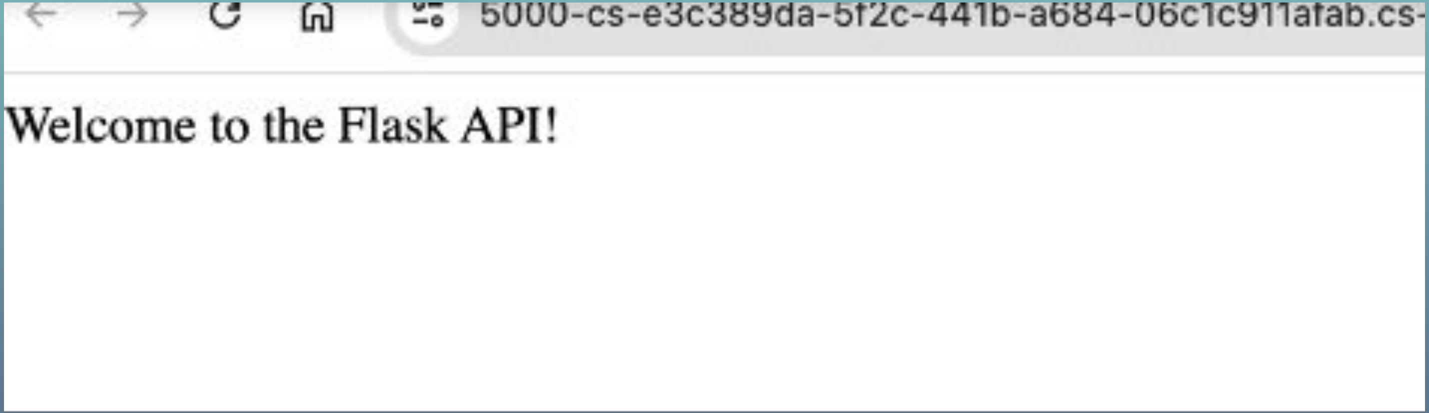
```
  GNU nano 5.4
FROM python:3.8-slim
WORKDIR /app
COPY . /app
EXPOSE 5000
RUN pip install -r requirements.txt
CMD ["python", "flask_api.py"]
```

Step 4: Run the docker container



```
saron54lo@cloudshell:~ (genuine-space-430317-j5)$ sudo docker build -t ml_app_docker .
[+] Building 55.7s (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@sha256:48dba4e619b54ecf9ef35695e2e2249392345161e0e7f5b21efdbeb6f0dd0df7        0.0s
 => [internal] load build context                                                                                               0.0s
 => => transferring context: 2.06kB                                                                                             0.0s
 => CACHED [2/4] WORKDIR /app                                                                                                   0.0s
 => [3/4] COPY . /app                                                                                                           0.0s
 => [4/4] RUN pip install -r requirements.txt                                                                                  48.1s
 => exporting to image                                                                                                          7.2s
 => => exporting layers                                                                                                         7.2s
 => => writing image sha256:f70a1f093cd6b2017910ceeae478c22c83be39a3ef620aa214aa36e27f0c78e4                                    0.0s
 => => naming to docker.io/library/ml_app_docker                                                                                0.0s
saron54lo@cloudshell:~ (genuine-space-430317-j5)$  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: 219-962-672
```

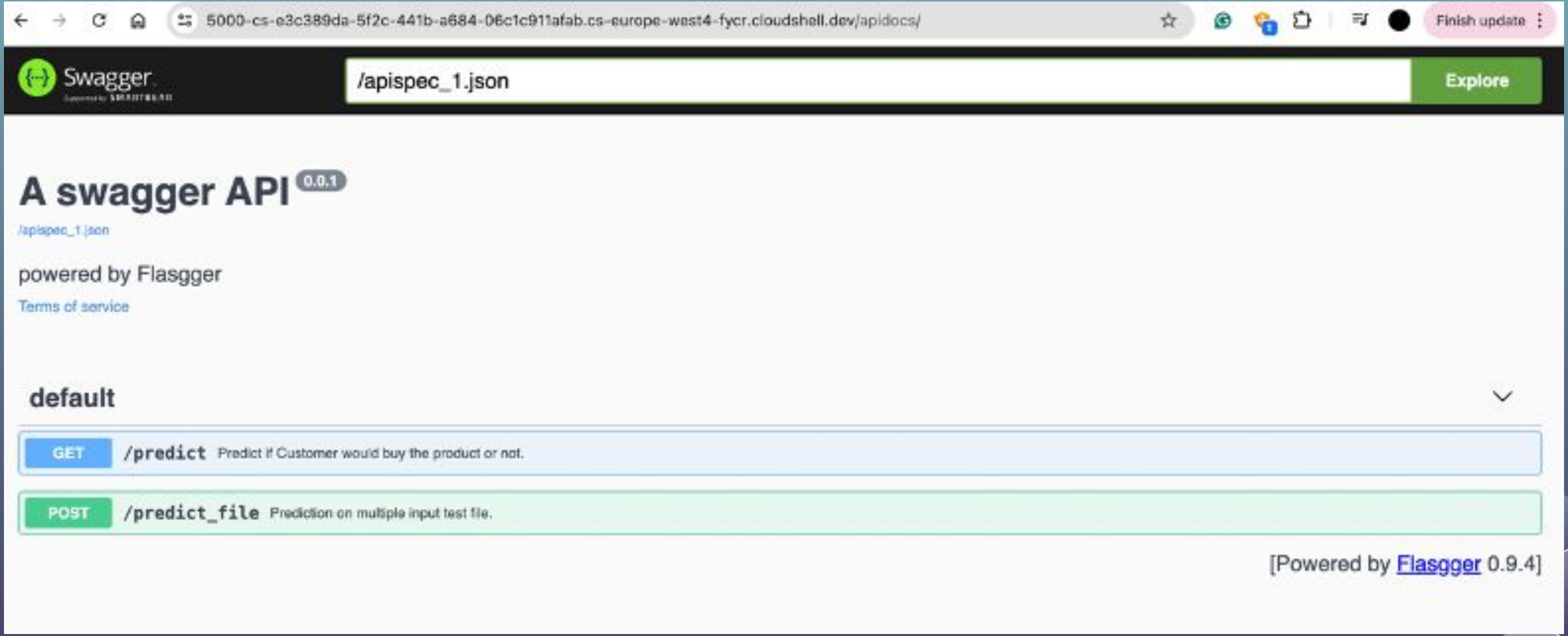
Step 5: Change port to 5000 and open the website



Welcome to the Flask API!

Step 6: Add /apidocs/ to the end of link

https://5000-cs-e3c389da-5f2c-441b-a684-06c1c911afab.cs-europe-west4-fycr.cloudshell.dev/apidocs/#/default/post_predict_file


5000-cs-e3c389da-5f2c-441b-a684-06c1c911afab.cs-europe-west4-fycr.cloudshell.dev/apidocs/
Finish update
Swagger
Explore

/apispec_1.json

A swagger API 0.0.1

/apispec_1.json

powered by Flasgger

Terms of service

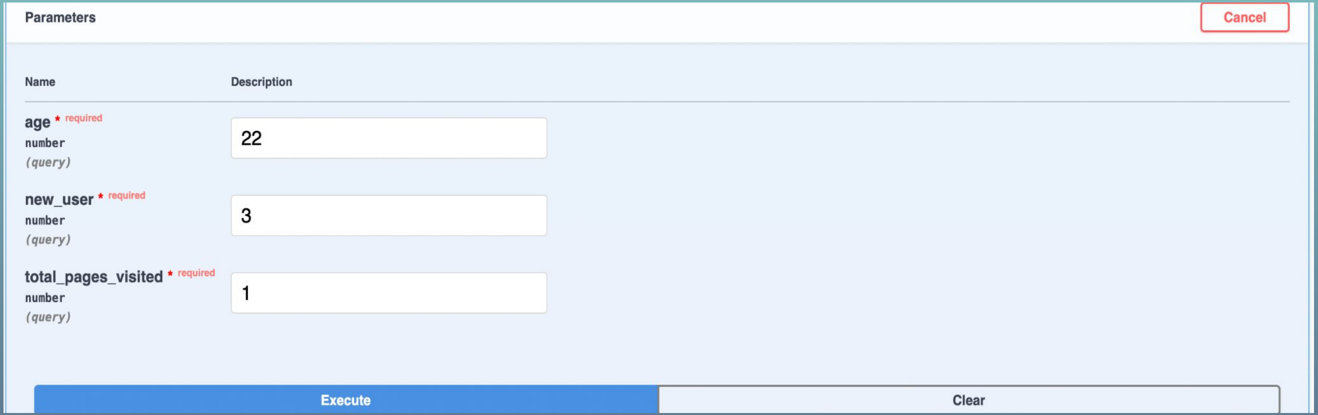
default

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]

Step 7: Try out the GET button



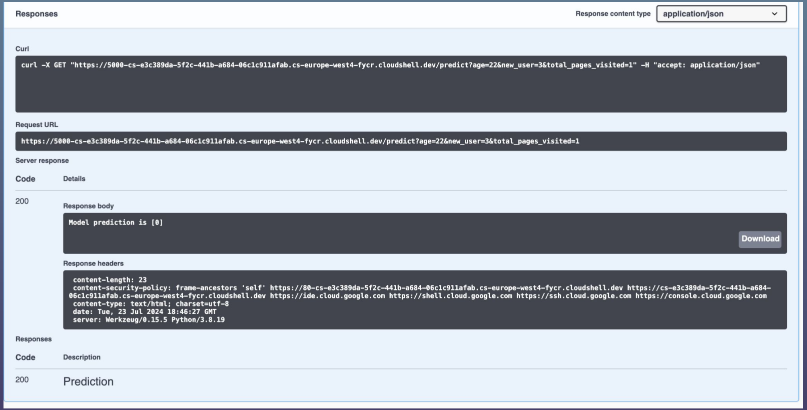
Parameters

Cancel

Name	Description
age * required number *(query)*	22
new_user * required number *(query)*	3
total_pages_visited * required number *(query)*	1

Execute

Clear



Responses

Response content type: application/json

Curl

```
curl -X GET "https://5000-cs-e3c389da-5f2c-441b-a684-06c1c911afab.cs-europe-west4-fycr.cloudshell.dev/predict?age=22&new_user=3&total_pages_visited=1" -H "accept: application/json"
```

Request URL

```
https://5000-cs-e3c389da-5f2c-441b-a684-06c1c911afab.cs-europe-west4-fycr.cloudshell.dev/predict?age=22&new_user=3&total_pages_visited=1
```

Server response

Code | Details

200

Response body

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

Download

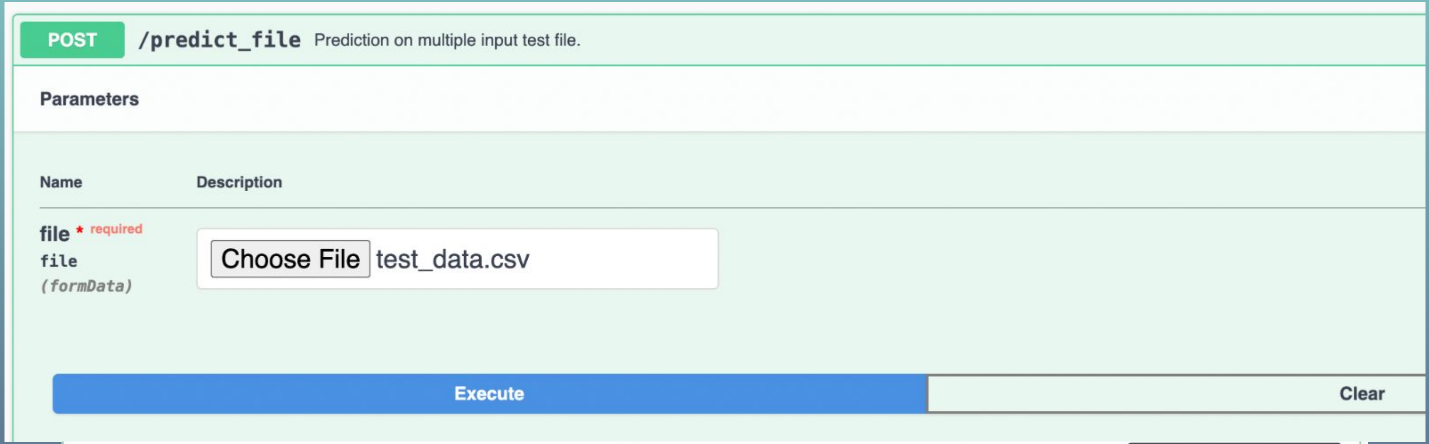
Response headers

```
 content-length: 23
 content-security-policy: frame-ancestors 'self' https://80-cs-e3c389da-5f2c-441b-a684-06c1c911afab.cs-europe-west4-fycr.cloudshell.dev https://cs-e3c389da-5f2c-441b-a684-
06c1c911afab.cs-europe-west4-fycr.cloudshell.dev https://ide.cloud.google.com https://shell.cloud.google.com https://ssh.cloud.google.com https://console.cloud.google.com
 content-type: text/html; charset=utf-8
 date: Tue, 23 Jul 2024 18:46:27 GMT
 server: Werkzeug/0.15.5 Python/3.8.19
```

Responses

Code	Description
200	Prediction

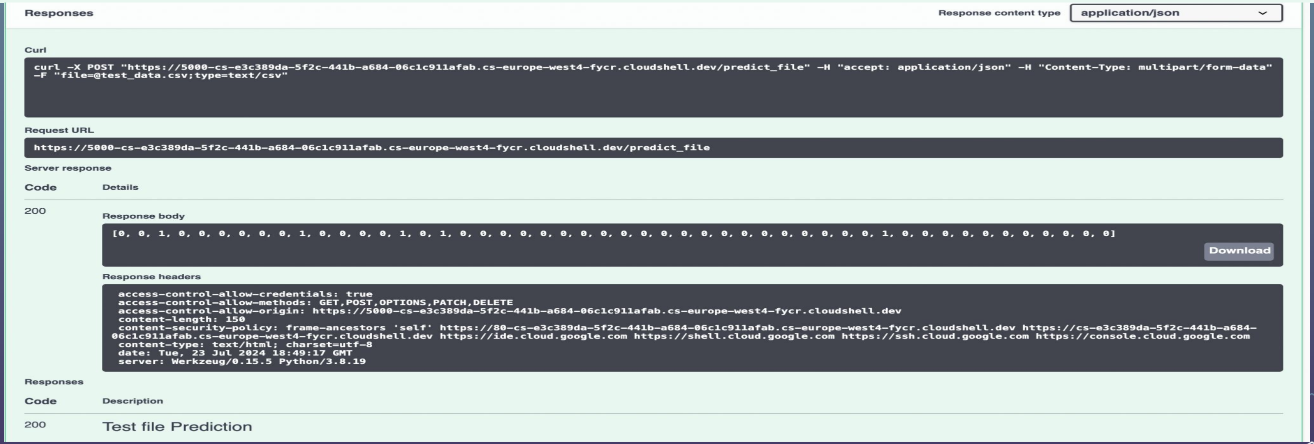
Step 8: Try out the POST button



/predict_file Prediction on multiple input test file.

Parameters

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



Responses

Response content type: application/json

Curl

```
curl -X POST "https://5000-cs-e3c389da-5f2c-441b-a684-06c1c911afab.cs-europe-west4-fycr.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-e3c389da-5f2c-441b-a684-06c1c911afab.cs-europe-west4-fycr.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]
```

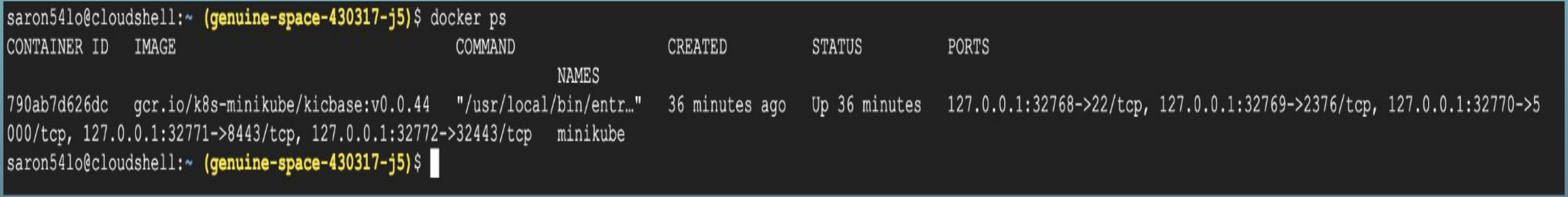
Response headers

```
 access-control-allow-credentials: true
 access-control-allow-methods: GET,POST,OPTIONS,PATCH,DELETE
 access-control-allow-origin: https://5000-cs-e3c389da-5f2c-441b-a684-06c1c911afab.cs-europe-west4-fycr.cloudshell.dev
 content-length: 150
 content-security-policy: frame-ancestors 'self' https://80-cs-e3c389da-5f2c-441b-a684-06c1c911afab.cs-europe-west4-fycr.cloudshell.dev https://cs-e3c389da-5f2c-441b-a684-
06c1c911afab.cs-europe-west4-fycr.cloudshell.dev https://ide.cloud.google.com https://shell.cloud.google.com https://ssh.cloud.google.com https://console.cloud.google.com
 content-type: text/html; charset=utf-8
 date: Tue, 23 Jul 2024 18:49:17 GMT
 server: Werkzeug/0.15.5 Python/3.8.19
```

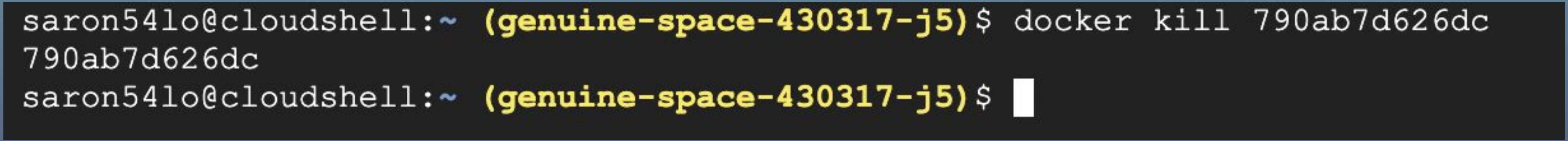
Responses

Code	Description
200	Test file Prediction

Step 9: Stop the running container



```
saron54lo@cloudshell:~ (genuine-space-430317-j5)$ docker ps
CONTAINER ID   IMAGE                                 COMMAND                  CREATED          STATUS          PORTS
                                                           NAMES
790ab7d626dc   gcr.io/k8s-minikube/kicbase:v0.0.44   "/usr/local/bin/entr…"   36 minutes ago   Up 36 minutes   127.0.0.1:32768->22/tcp, 127.0.0.1:32769->2376/tcp, 127.0.0.1:32770->5
000/tcp, 127.0.0.1:32771->8443/tcp, 127.0.0.1:32772->32443/tcp   minikube
saron54lo@cloudshell:~ (genuine-space-430317-j5)$
```



```
saron54lo@cloudshell:~ (genuine-space-430317-j5)$ docker kill 790ab7d626dc
790ab7d626dc
saron54lo@cloudshell:~ (genuine-space-430317-j5)$
```

Enhancement 05

- **Model Improvement:** Train and deploy a more complex model for better accuracy.
- **Scalability:** Use Kubernetes for scaling the application based on demand.
- **Logging and Monitoring:** Implement logging and monitoring for better maintenance and troubleshooting.

Conclusion

06

This project demonstrates how to containerize a machine learning model and deploy it using Docker and Flask. By integrating Kubernetes, the application can be scaled and managed efficiently, showcasing the power of cloud-based machine learning deployments.

GITHUB LINK

https://github.com/Sharon20222/Cloud-Computing/tree/main/Kubernetes/Machine%20Learning

References

07

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