Sosanya Adebayo   CS571   19944

Machine Learning on Kubernetes

Creating and uploading necessary files in GCP- Cloud Shell Terminal

Start minikube in Google Cloud Platform

```
asosanya166@cloudshell:~ (c5517-ml-deployment)$ minikube start
* minikube v1.34.0 on Ubuntu 24.04 (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 v0.0.45 ...
* Downloading Kubernetes v1.31.0 preload ...
    > preloaded-images-k8s-v18-v1...:  326.69 MiB / 326.69 MiB  100.00% 216.79
    > gcr.io/k8s-minikube/kicbase...:  487.90 MiB / 487.90 MiB  100.00% 96.15 M
* Creating docker container (CPUs=2, Memory=4000MB) ...
* Preparing Kubernetes v1.31.0 on Docker 27.2.0 ...
  - 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
```

Create requirements.txt file using the following command

- nano requirements.txt

```
  GNU nano 7.2
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
Flask==2.1.2
flask-pymongo==2.3.0
pymongo==4.1.1  # Or a version compatible with Flask-PyMongo
```

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

Terminal and then choose upload
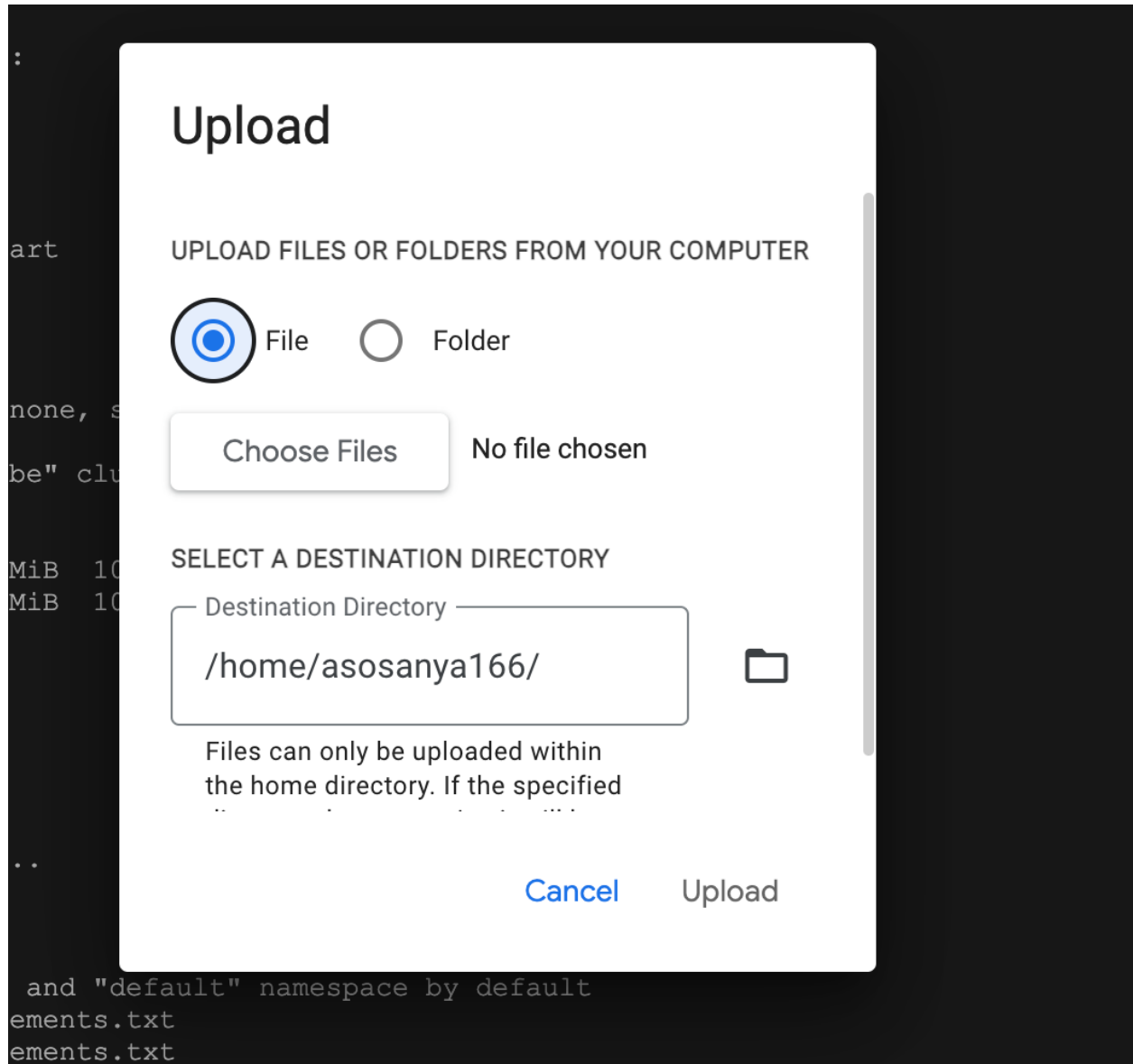
Create flask_api.py file using the command

- nano flask_api.py

```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
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
```

```python
          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)
```

Create Dockerfile using command

- nano Dockerfile

```
  GNU nano 7.2
FROM python:3.8-slim
WORKDIR /app
COPY . /app
EXPOSE 5000
RUN pip install -r requirements.txt
CMD ["python", "flask_api.py"]
```

To build the docker image use the command

- sudo docker build -t ml_app_docker .

```
asosanya166@cloudshell:~ (c5517-ml-deployment)$ sudo docker build -t ml_app_docker .
[+] Building 50.9s (9/9) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 162B
 => [internal] load metadata for docker.io/library/python:3.8-slim
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [1/4] FROM docker.io/library/python:3.8-slim@sha256:1d52838af602b4b5a831beb13a0e4d073280665ea7be7f69ce2382f29c5a613f
 => [internal] load build context
 => => transferring context: 889.99kB
 => CACHED [2/4] WORKDIR /app
 => [3/4] COPY . /app
 => [4/4] RUN pip install -r requirements.txt
 => exporting to image
 => => exporting layers
 => => writing image sha256:be7c01d200939de44e0948e13d105edd98c409b9ac2201a42fef0d63df59ff43
 => => naming to docker.io/library/ml_app_docker
asosanya166@cloudshell:~ (c5517-ml-deployment)$
```

This command runs a Docker container from the ml_app_docker image:

- docker container run -p 5000:5000 ml_app_docker

```
asosanya166@cloudshell:~ (c5517-ml-deployment)$ docker 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 mi
ght 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 0.23.2 when using version 0.24.2. This mi
ght lead to breaking code or invalid results. Use at your own risk.
  warnings.warn(
 * Debugger is active!
 * Debugger PIN: 320-660-712

                                                              Transferred 1 item         ^   ✕
```
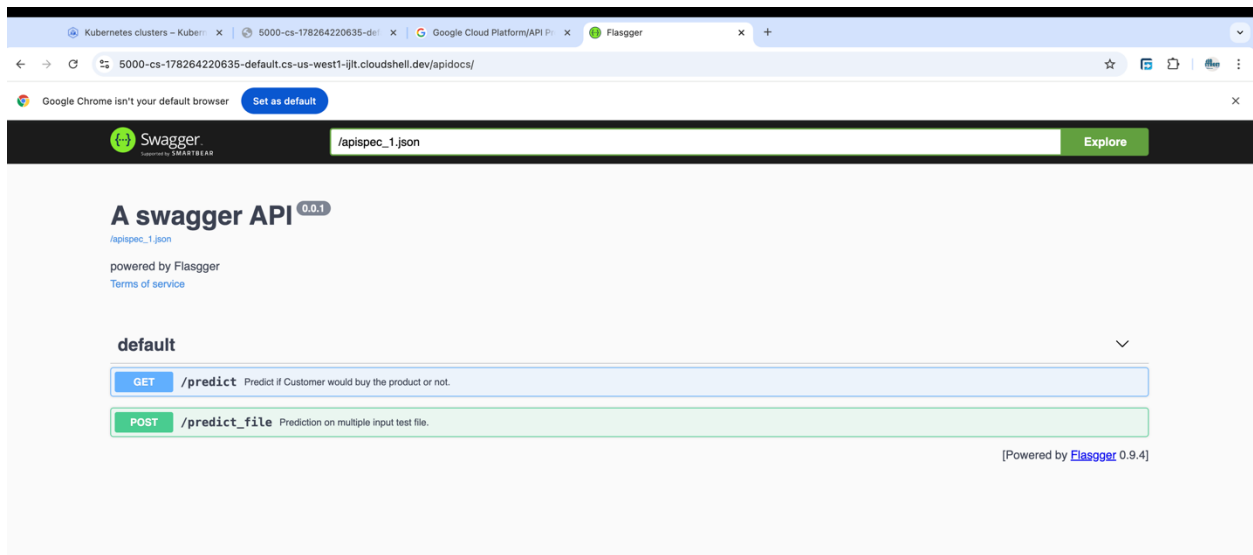
In the right-upper side of the terminal click the eye shaped button and then click Preview on port 5000. Change port if it is not 5000 by default.
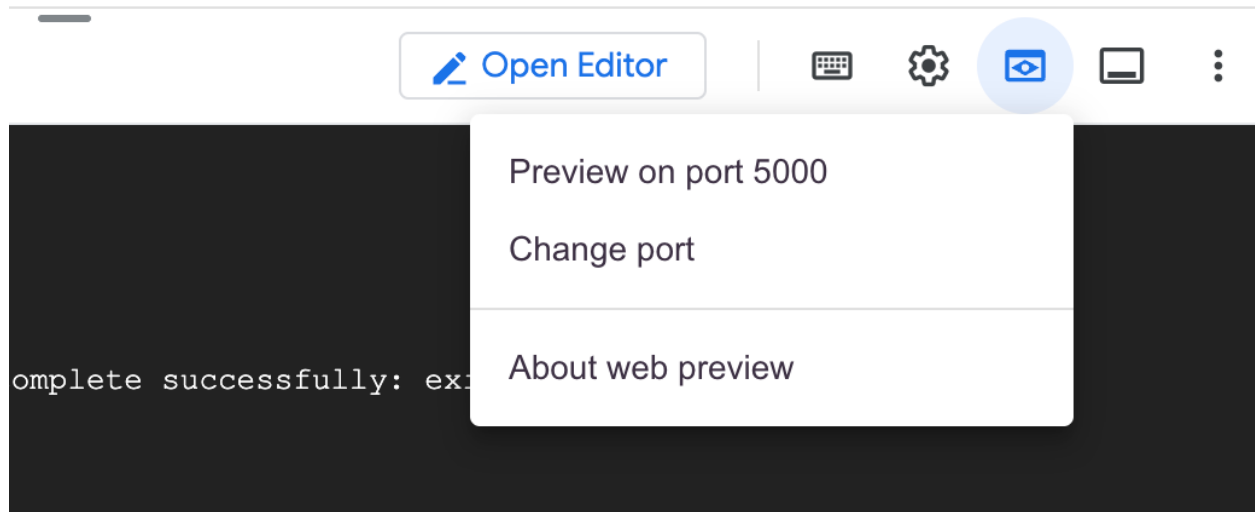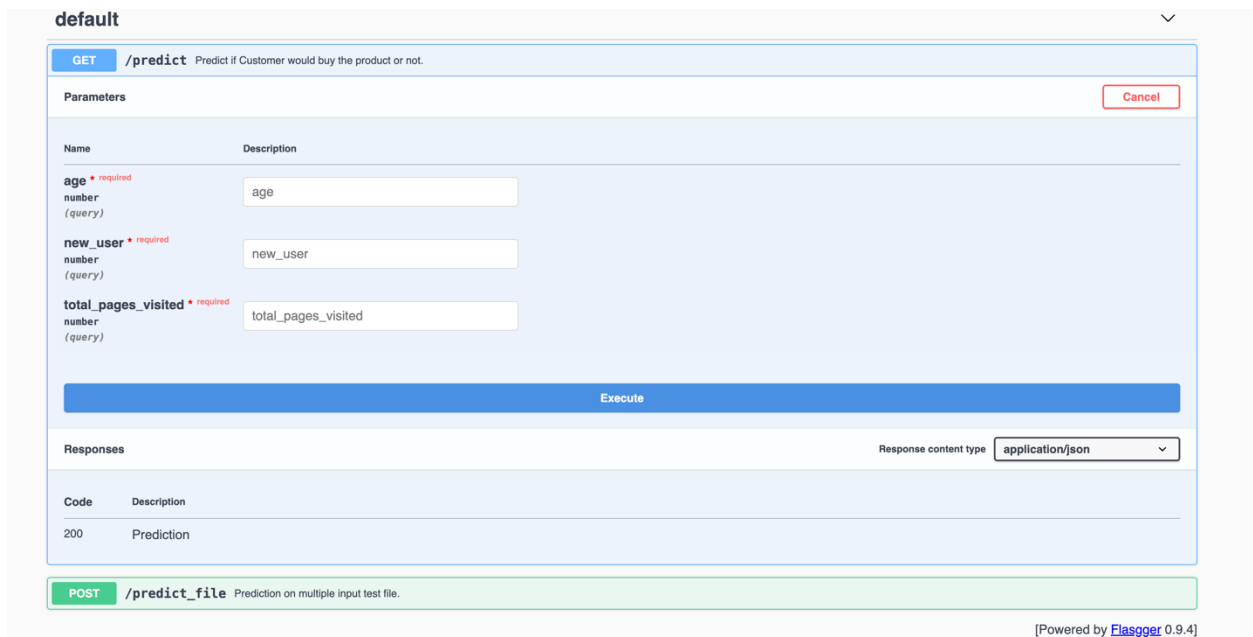

. You will see this using the web preview.
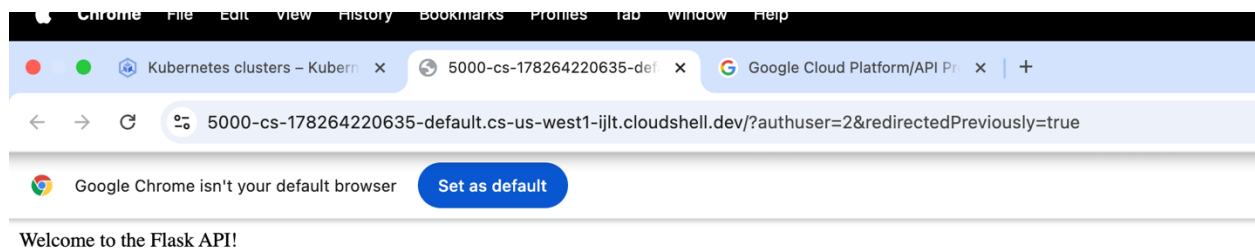
. Add /apidocs/ at the end of the link to access the running ml- app as following - There are two tabs GET and POST

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

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

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

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.

Action performed on Swagger showing on Cloud Shell

```
 * Debugger PIN: 320-660-712
172.17.0.1 - - [21/Nov/2024 13:10:18] "GET /?authuser=2&redirectedPreviously=true HTTP/1.1" 200 -
172.17.0.1 - - [21/Nov/2024 13:10:19] "GET /favicon.ico HTTP/1.1" 404 -
172.17.0.1 - - [21/Nov/2024 13:11:55] "GET /?authuser=2&redirectedPreviously=true/apidocs/ HTTP/1.1" 200 -
172.17.0.1 - - [21/Nov/2024 13:12:54] "GET /?authuser=2&redirectedPreviously=true:5000/apidocs/ HTTP/1.1" 200 -
172.17.0.1 - - [21/Nov/2024 13:14:00] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [21/Nov/2024 13:14:14] "GET /apidocs/ HTTP/1.1" 200 -
172.17.0.1 - - [21/Nov/2024 13:14:15] "GET /flasgger_static/swagger-ui.css HTTP/1.1" 200 -
172.17.0.1 - - [21/Nov/2024 13:14:15] "GET /flasgger_static/swagger-ui-bundle.js HTTP/1.1" 200 -
172.17.0.1 - - [21/Nov/2024 13:14:15] "GET /flasgger_static/swagger-ui-standalone-preset.js HTTP/1.1" 200 -
172.17.0.1 - - [21/Nov/2024 13:14:15] "GET /flasgger_static/lib/jquery.min.js HTTP/1.1" 200 -
172.17.0.1 - - [21/Nov/2024 13:14:15] "GET /flasgger_static/favicon-32x32.png HTTP/1.1" 200 -
172.17.0.1 - - [21/Nov/2024 13:14:15] "GET /apispec_1.json HTTP/1.1" 200 -
172.17.0.1 - - [21/Nov/2024 13:20:22] "GET /predict?age=23&new_user=2&total_pages_visited=5 HTTP/1.1" 200 -
172.17.0.1 - - [21/Nov/2024 13:27:04] "POST /predict_file HTTP/1.1" 200 -
```

Stopping/killing the running container

Use docker ps to list running Docker containers

```
^Casosanya166@cloudshell:~ (c5517-ml-deployment)$ docker ps
CONTAINER ID   IMAGE                                 COMMAND                  CREATED        STATUS        PORTS
                                                                                                          NAMES
604de60e2064   gcr.io/k8s-minikube/kicbase:v0.0.45   "/usr/local/bin/entr…"   45 minutes ago  Up 45 minutes  127.0.0.1:32768->22/tcp, 127.0.0.1:32769->2376/tcp, 127.0.0.1:32
770->5000/tcp, 127.0.0.1:32771->8443/tcp, 127.0.0.1:32772->32443/tcp   minikube
asosanya166@cloudshell:~ (c5517-ml-deployment)$ 
```

Use the command

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

docker kill 604de60e2064

```
770-93000/ccp, 127.0.0.1.32771-98443/tcp, 127.0.0.1.32772-932443/tcp   minikube
asosanya166@cloudshell:~ (c5517-ml-deployment)$ docker kill 604de60e2064
604de60e2064
asosanya166@cloudshell:~ (c5517-ml-deployment)$
```