

TITLE: MODEL DEPLOYMENT ON GOOGLE CLOUD PLATFORM

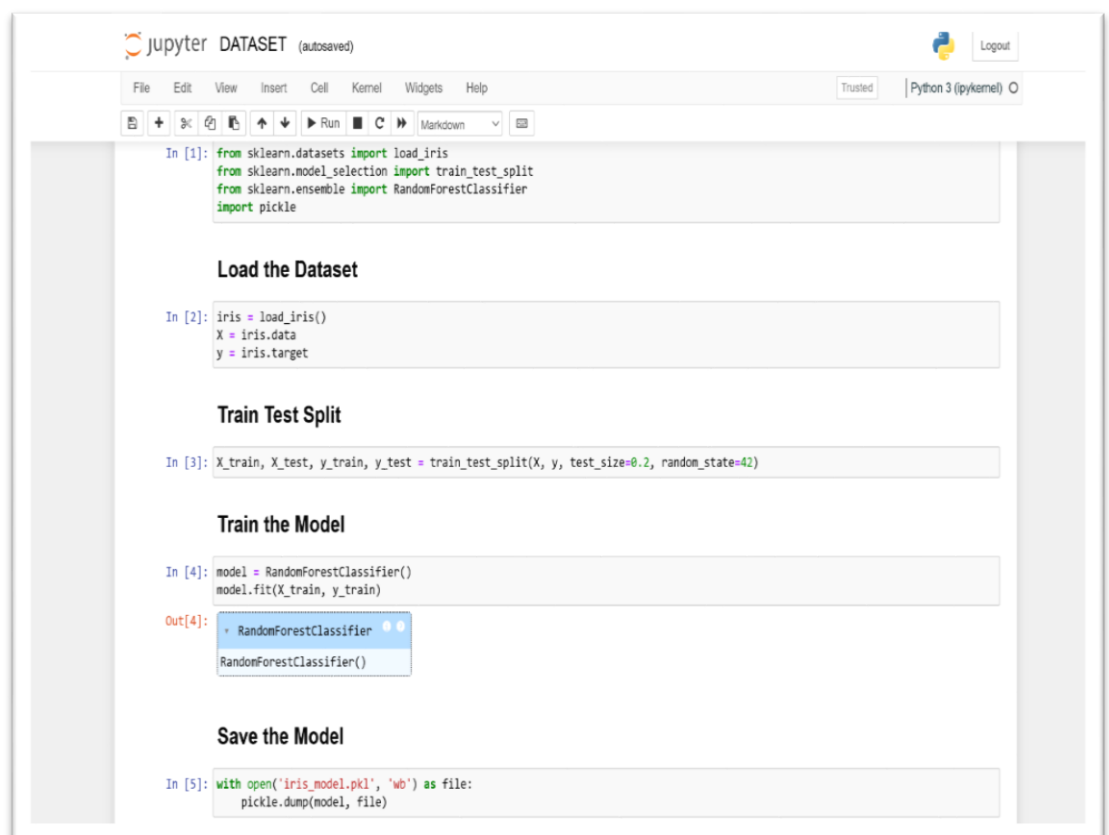
OVERVIEW:

- This project demonstrates the end-to-end deployment of a machine learning model using Google Cloud Platform (GCP)
- The app includes both a web interface and an API endpoint to make predictions on the Iris dataset using a Random Forest Classifier model
- Technologies used: Python, Flask, scikit-learn, GCP App Engine

STEP-BY-STEP DEPLOYMENT PROCESS:

✓ Step 1: Select & Train the Model

- **Description:** Used the same Iris dataset which was used in week 4 for flask deployment and trained a classifier using Random Forest Classifier model



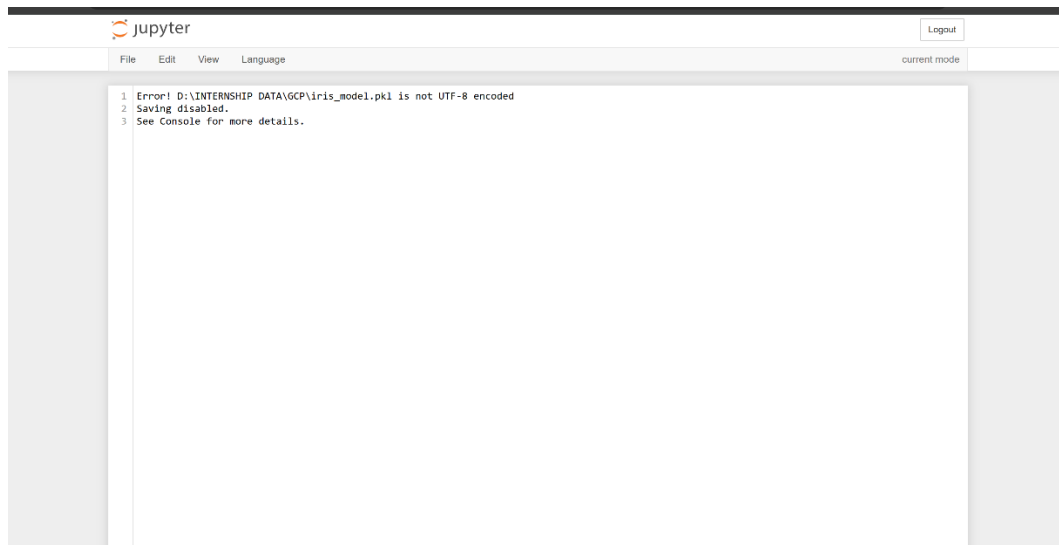
```
Jupyter DATASET (autosaved)
File Edit View Insert Cell Kernel Widgets Help
In [1]: from sklearn.datasets import load_iris
        from sklearn.model_selection import train_test_split
        from sklearn.ensemble import RandomForestClassifier
        import pickle

Load the Dataset
In [2]: iris = load_iris()
        X = iris.data
        y = iris.target

Train Test Split
In [3]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

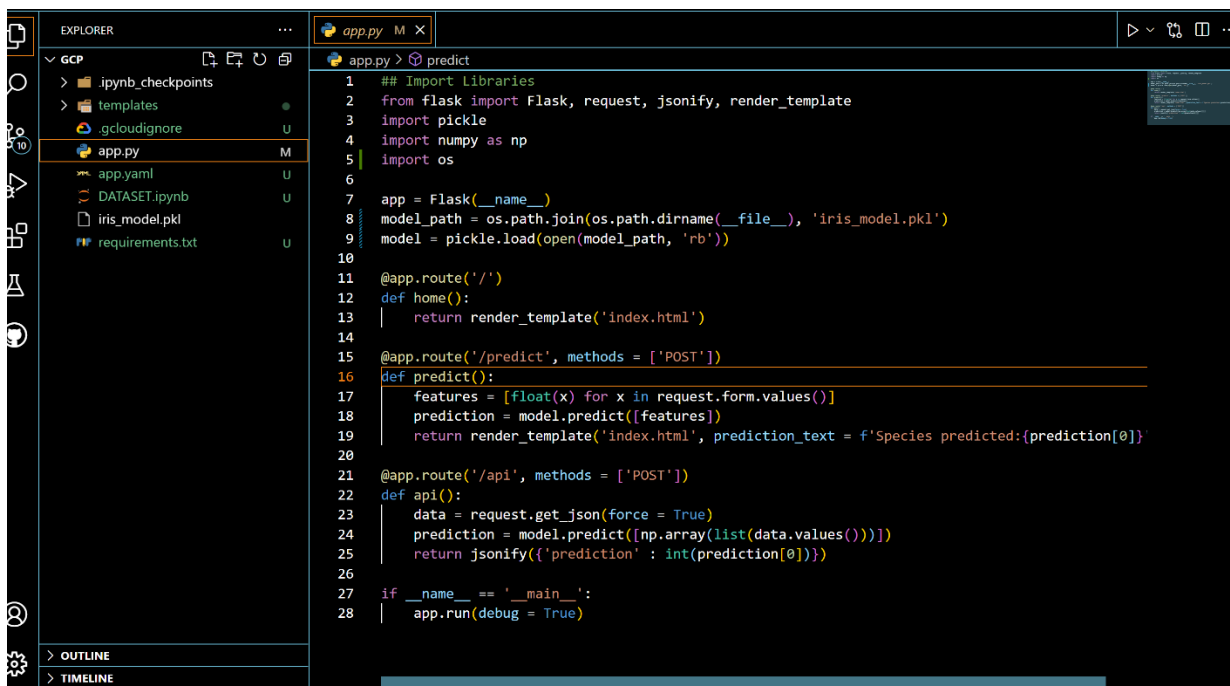
Train the Model
In [4]: model = RandomForestClassifier()
        model.fit(X_train, y_train)
Out[4]: RandomForestClassifier
        RandomForestClassifier()

Save the Model
In [5]: with open('iris_model.pkl', 'wb') as file:
        pickle.dump(model, file)
```



✓ Step 2: Flask App Setup

- **Description:** Created app.py with / for web UI and /api for JSON prediction
- **Files Included:** app.py, iris_model.pkl, requirements.txt, app.yaml, templates/index.html



✓ Step 3: Create GCP App Files

- **app.yaml:** Config for App Engine
- **requirements.txt:** Installed Flask, sklearn, numpy, gunicorn

EXPLORER

...

app.yaml U X

EXPLORE

...

app.yaml

1 runtime: python39

2 entrypoint: gunicorn -b :\$PORT app:app

3

4

5 handlers:

6 - url: /static

7 | static_dir: static

8

9

10 - url: /*

11 | script: auto

12

> OUTLINE

> TIMELINE

EXPLORER

...

requirements.txt U X

EXPLORE

...

requirements.txt

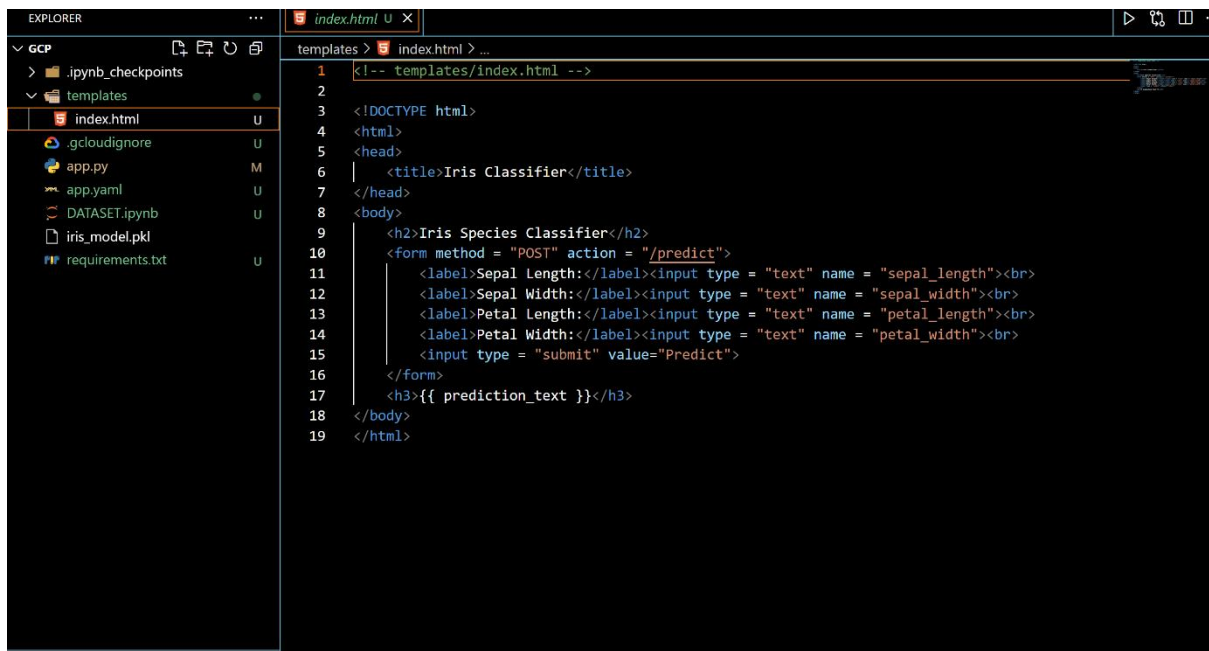
1 Flask

2 numpy

3 scikit-learn

4 gunicorn

5

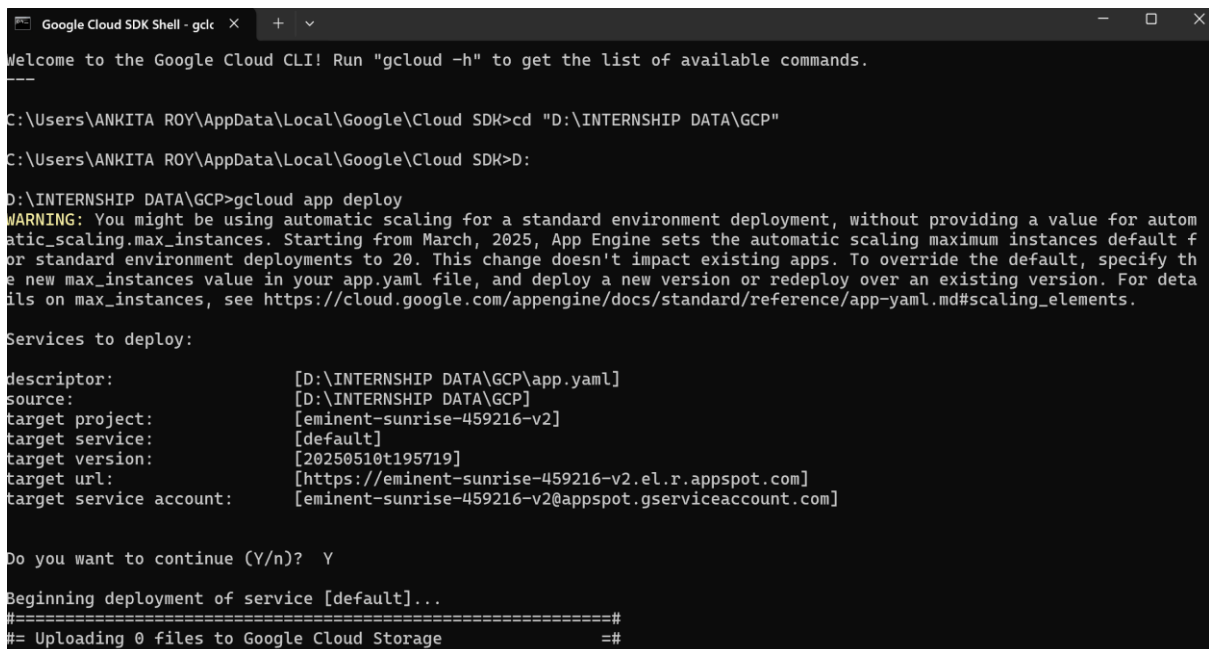


The screenshot shows a Visual Studio Code editor with two panes. The left pane, titled 'EXPLORER', shows a file tree for a project named 'GCP'. The tree includes files like '.ipynb_checkpoints', 'templates', 'index.html', '.gcloudignore', 'app.py', 'app.yaml', 'DATASET.ipynb', 'iris_model.pkl', and 'requirements.txt'. The right pane shows the content of 'index.html', which is an HTML form titled 'Iris Species Classifier'. The form has four text input fields for 'Sepal Length', 'Sepal Width', 'Petal Length', and 'Petal Width', followed by a 'Predict' button. The form is configured to use a POST method and a '/predict' action.

```
1 <!-- templates/index.html -->
2
3 <!DOCTYPE html>
4 <html>
5 <head>
6   <title>Iris Classifier</title>
7 </head>
8 <body>
9   <h2>Iris Species Classifier</h2>
10  <form method = "POST" action = "/predict">
11    <label>Sepal Length:</label><input type = "text" name = "sepal_length"><br>
12    <label>Sepal Width:</label><input type = "text" name = "sepal_width"><br>
13    <label>Petal Length:</label><input type = "text" name = "petal_length"><br>
14    <label>Petal Width:</label><input type = "text" name = "petal_width"><br>
15    <input type = "submit" value="Predict">
16  </form>
17  <h3>{{ prediction_text }}</h3>
18 </body>
19 </html>
```

✓ Step 4: GCP Setup

- gcloud init configuration
- Region selection
- Service creation
- Successful gcloud app deploy



The screenshot shows a terminal window titled 'Google Cloud SDK Shell - gcl'. The prompt is 'Welcome to the Google Cloud CLI! Run "gcloud -h" to get the list of available commands.' The user has navigated to the directory 'D:\INTERNSHIP DATA\GCP' and executed the command 'gcloud app deploy'. The output shows a warning about automatic scaling and then lists the services to be deployed. The deployment is successful, and the user is prompted to continue (Y/n) with 'Y'.

```
Google Cloud SDK Shell - gcl
Welcome to the Google Cloud CLI! Run "gcloud -h" to get the list of available commands.

C:\Users\ANKITA ROY\AppData\Local\Google\Cloud SDK>cd "D:\INTERNSHIP DATA\GCP"

C:\Users\ANKITA ROY\AppData\Local\Google\Cloud SDK>D:
D:\INTERNSHIP DATA\GCP>gcloud app deploy
WARNING: You might be using automatic scaling for a standard environment deployment, without providing a value for autom
atic_scaling.max_instances. Starting from March, 2025, App Engine sets the automatic scaling maximum instances default f
or standard environment deployments to 20. This change doesn't impact existing apps. To override the default, specify th
e new max_instances value in your app.yaml file, and deploy a new version or redeploy over an existing version. For deta
ils on max_instances, see https://cloud.google.com/appengine/docs/standard/reference/app-yaml.md#scaling_elements.

Services to deploy:

descriptor:      [D:\INTERNSHIP DATA\GCP\app.yaml]
source:         [D:\INTERNSHIP DATA\GCP]
target project:  [eminent-sunrise-459216-v2]
target service:  [default]
target version:  [20250510t195719]
target url:      [https://eminent-sunrise-459216-v2.el.r.appspot.com]
target service account: [eminent-sunrise-459216-v2@appspot.gserviceaccount.com]

Do you want to continue (Y/n)? Y

Beginning deployment of service [default]...
#=====
#= Uploading 0 files to Google Cloud Storage      =#
```

```

target version:      [20250510t195719]
target url:          [https://eminent-sunrise-459216-v2.el.r.appspot.com]
target service account: [eminent-sunrise-459216-v2@appspot.gserviceaccount.com]

Do you want to continue (Y/n)? Y

Beginning deployment of service [default]...
=====
#= Uploading 0 files to Google Cloud Storage      =#
=====
File upload done.
Updating service [default]...done.
Setting traffic split for service [default]...done.
Deployed service [default] to [https://eminent-sunrise-459216-v2.el.r.appspot.com]

You can stream logs from the command line by running:
$ gcloud app logs tail -s default

To view your application in the web browser run:
$ gcloud app browse

D:\INTERNSHIP DATA\GCP>

```

✓ Step 5: Test the Web App

<https://eminent-sunrise-459216-v2.el.r.appspot.com>



The screenshot shows a web browser window with the title "Iris Species Classifier". The page contains four input fields labeled "Sepal Length:", "Sepal Width:", "Petal Length:", and "Petal Width:". Below these fields is a "Predict" button. Under the button, the text "Species predicted:0" is displayed.

✓ Step 6: Test the API

Curl

```

C:\Users\ANKITA ROY>curl -X POST "https://eminent-sunrise-459216-v2.el.r.appspot.com/api" -H "Content-Type: application/
json" -d "{\"feature1\":5.1,\"feature2\":3.5,\"feature3\":1.4,\"feature4\":0.2}"
{"prediction":0}

C:\Users\ANKITA ROY>

```

Conclusion:

This project successfully demonstrates the complete deployment pipeline of a machine learning model using the **Google Cloud Platform (GCP)**. A **Random Forest Classifier** trained on the Iris dataset was integrated with a Flask application, which was then deployed on GCP App Engine.

Both a **web-based interface** and an **API endpoint** were implemented to interact with the model. The deployment utilized free-tier credits provided by GCP, making it accessible and cost-effective.

Through this project, key cloud concepts such as environment configuration, dependency management, and scalable deployment were applied in practice, reinforcing both machine learning and DevOps skills.

Submitted by:

Ankita Roy

Batch Code: [LISUM44](#)

Submission Date: 05/05/25

Submitted to: Data Glacier