

Name: **Cynthia Onyia**

Batch Code: **LISUM44**

Submission Date: **05<sup>th</sup> May 2025**











Submitted To: **Data Glacier Team**

**Details:** This task is based on and continued from the task of week four (4). The toy dataset used for this task was the Iris dataset which consist of 150 records (observations) of five attributes: sepal length, sepal width, petal length, petal width and species (which are of three categories/classification: **0** for Iris setosa, **1** for Iris versicolor, and **2** for Iris virginica). The species are also known as the target variables.

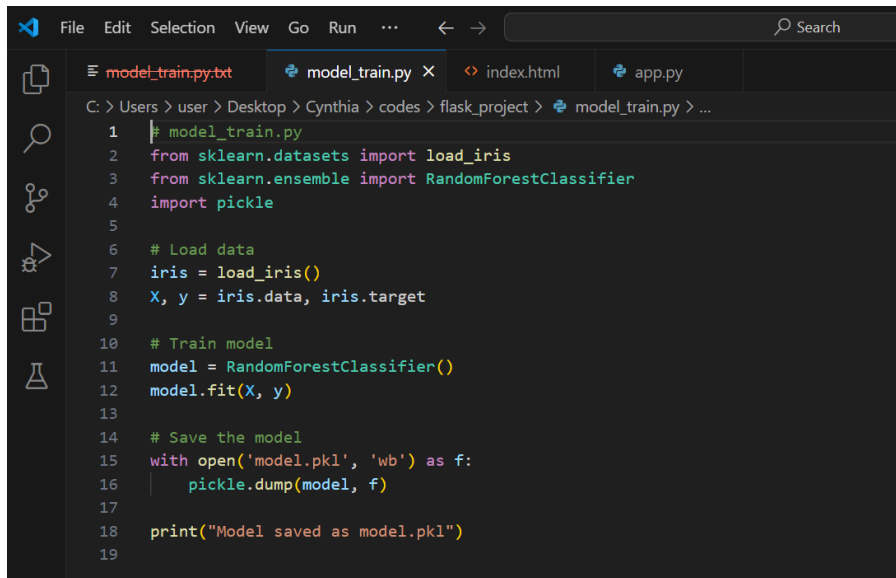
The following contains screenshots of the processes I used to deploy the model I trained using Random Forest on the cloud (I used Heroku). These screenshots will also include some screenshots from the week 4 task on my **first Flask app** (Flask deployment) using this same Iris dataset (toy data).

- 1. App structure:** This shows the folder layouts of the files and documents needed for app creation and deployment.

top > Cynthia > codes > flask\_project

<input type="checkbox"/> Name	Date modified	Type	Size
 .git	04/05/2025 14:51	File folder	
 templates	27/04/2025 16:49	File folder	
 .gitignore	03/05/2025 15:41	Text Document	1 KB
 app.py	28/04/2025 22:26	Python File	1 KB
 model.pkl	27/04/2025 16:43	PKL File	161 KB
 model_train.py	27/04/2025 16:43	Python File	1 KB
 Procfile	03/05/2025 15:33	File	1 KB
 requirements - full.txt	03/05/2025 15:35	Text Document	3 KB
 requirements.txt	04/05/2025 14:49	Text Document	1 KB
 runtime.txt	03/05/2025 15:41	Text Document	1 KB

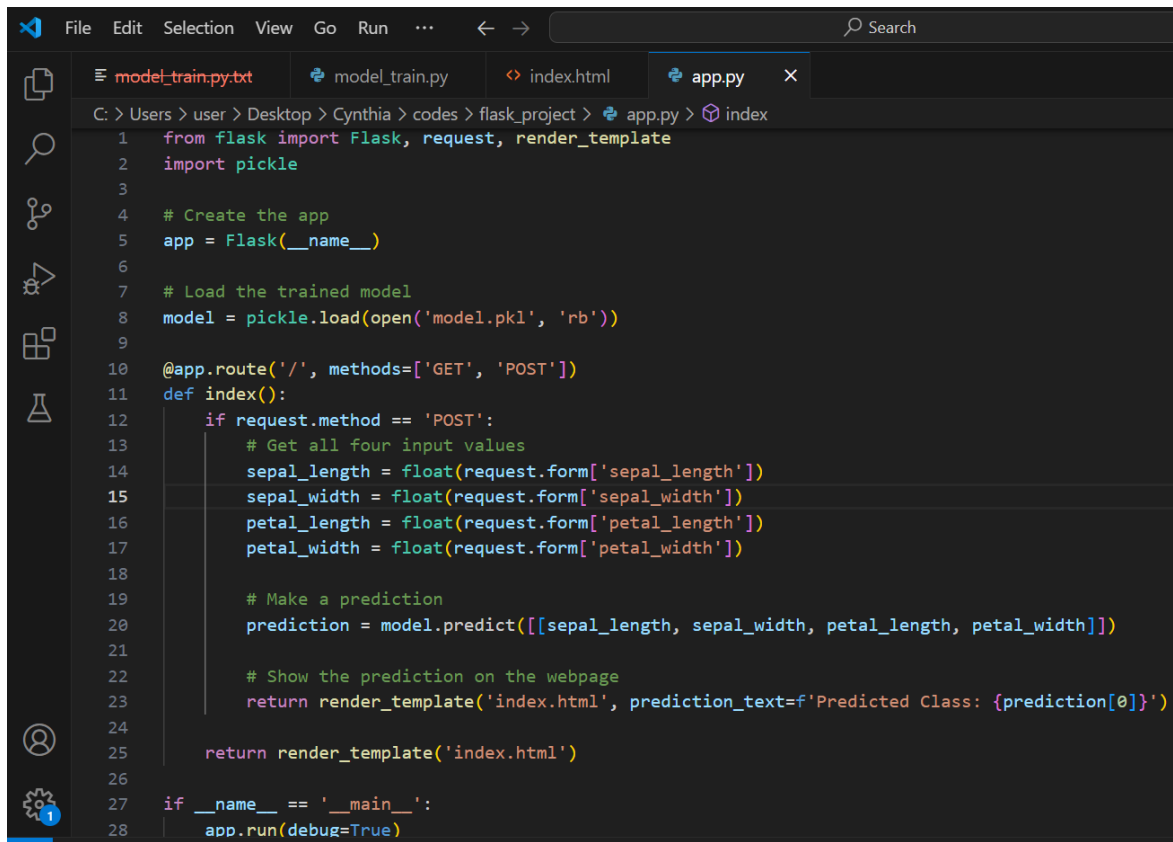
- 2. Code for model\_train.py:** The model was trained using Random Forest and saved using pkl. Below is a snippet of the training code.



```
1 # model_train.py
2 from sklearn.datasets import load_iris
3 from sklearn.ensemble import RandomForestClassifier
4 import pickle
5
6 # Load data
7 iris = load_iris()
8 X, y = iris.data, iris.target
9
10 # Train model
11 model = RandomForestClassifier()
12 model.fit(X, y)
13
14 # Save the model
15 with open('model.pkl', 'wb') as f:
16     pickle.dump(model, f)
17
18 print("Model saved as model.pkl")
19
```

**3. Flask App and API Code:** The Flask app was created with both a web interface and API endpoint (/predict).

### Code for app.py



```
1 from flask import Flask, request, render_template
2 import pickle
3
4 # Create the app
5 app = Flask(__name__)
6
7 # Load the trained model
8 model = pickle.load(open('model.pkl', 'rb'))
9
10 @app.route('/', methods=['GET', 'POST'])
11 def index():
12     if request.method == 'POST':
13         # Get all four input values
14         sepal_length = float(request.form['sepal_length'])
15         sepal_width = float(request.form['sepal_width'])
16         petal_length = float(request.form['petal_length'])
17         petal_width = float(request.form['petal_width'])
18
19         # Make a prediction
20         prediction = model.predict([[sepal_length, sepal_width, petal_length, petal_width]])
21
22         # Show the prediction on the webpage
23         return render_template('index.html', prediction_text=f'Predicted Class: {prediction[0]}')
24
25     return render_template('index.html')
26
27 if __name__ == '__main__':
28     app.run(debug=True)
```

## Code for index.html



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Flask Deployment Project</title>
6   <style>
7     body {
8       font-family: Arial, sans-serif;
9       text-align: center;
10      margin-top: 50px;
11      background-color: #f2f2f2;
12    }
13    form {
14      margin-top: 20px;
15    }
16    input[type="text"] {
17      padding: 10px;
18      width: 200px;
19      font-size: 16px;
20    }
21    button {
22      padding: 10px 20px;
23      font-size: 16px;
24      background-color: #4CAF50;
25      color: white;
26      border: none;
27      cursor: pointer;
28    }
29    button:hover {
30      background-color: #45a049;
31    }
32    h1, h2 {
33      color: #333;
34    }
35  </style>
36 </head>
37
38 <body>
39   <h1>Iris Prediction</h1>
40
41   <form action="/" method="post">
42     <label for="sepal_length">Sepal Length:</label><br>
43     <input type="text" id="sepal_length" name="sepal_length" required><br><br>
44
45     <label for="sepal_width">Sepal Width:</label><br>
46     <input type="text" id="sepal_width" name="sepal_width" required><br><br>
47
48     <label for="petal_length">Petal Length:</label><br>
49     <input type="text" id="petal_length" name="petal_length" required><br><br>
```

```

50
51     <label for="petal_width">Petal Width:</label><br>
52     <input type="text" id="petal_width" name="petal_width" required><br><br>
53
54     <button type="submit">Predict</button>
55 </form>
56
57 {% if prediction_text %}
58     <h2>{{ prediction_text }}</h2>
59 {% endif %}
60

```

## Terminal running Flask (python app.py)

```

C:\Users\user\Desktop\Cynthia\codes\flask_project>python app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 897-334-901
127.0.0.1 - - [28/Apr/2025 22:35:25] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [28/Apr/2025 22:36:14] "POST / HTTP/1.1" 200 -

```

## Browser showing your form (before prediction)

The screenshot shows a web browser window titled "Flask Deployment Project" with the address bar displaying "127.0.0.1:5000". The page content is titled "Iris Prediction" and contains a form with the following inputs:

- Sepal Length: 5.1
- Sepal Width: 3.5
- Petal Length: 1.4
- Petal Width: 0.2

A green "Predict" button is located at the bottom of the form.

## Browser showing the prediction result

The screenshot shows a web browser window with the following details:

- Tab:** Flask Deployment Project
- Address Bar:** 127.0.0.1:5000
- Page Title:** Iris Prediction
- Form Fields:**
  - Sepal Length:
  - Sepal Width:
  - Petal Length:
  - Petal Width:
- Action Button:** A green button labeled "Predict".
- Result:** Predicted Class: 0

**4. Heroku Deployment:** The next thing was to deploy to the cloud. Heroku was chosen because it's a great Flask app, which is beginner-friendly and free to use. The following steps were taken to ensure successful deployment. I,

- created Procfile
- installed Gunicorn
- initialised Git, committed files
- pushed to Heroku via Git

Snapshots are given below

## Screenshot of terminal showing git push Heroku main and deployment logs.

```
C:\Users\user\Desktop\Cynthia\codes\flask_project>git push heroku main
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 4 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (14/14), 23.58 KiB | 243.00 KiB/s, done.
Total 14 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Updated 9 paths from 2a75927
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Building on the Heroku-24 stack
remote: -----> Determining which buildpack to use for this app
remote: -----> Python app detected
remote: -----> Using Python 3.10.12 specified in runtime.txt
remote:
remote: !       Warning: The runtime.txt file is deprecated.
remote: !
remote: !       The runtime.txt file is deprecated since it has been replaced
remote: !       by the more widely supported .python-version file:
remote: !       https://devcenter.heroku.com/changelog-items/3141
remote: !
remote: !       Please delete your runtime.txt file and create a new file named:
remote: !       .python-version
remote: !
remote: !       Make sure to include the '.' at the start of the filename.
remote: !
remote: !       In the new file, specify your app's Python version without
remote: !       quotes or a 'python-' prefix. For example:
remote: !       3.10

remote: !       We strongly recommend that you use the major version form
remote: !       instead of pinning to an exact version, since it will allow
remote: !       your app to receive Python security updates.
remote: !
remote: !       In the future support for runtime.txt will be removed and
remote: !       this warning will be made an error.
remote:
remote:
remote: !       Warning: A Python patch update is available!
remote: !
remote: !       Your app is using Python 3.10.12, however, there is a newer
remote: !       patch release of Python 3.10 available: 3.10.17
remote: !
remote: !       It is important to always use the latest patch version of
remote: !       Python to keep your app secure.
remote: !
remote: !       Update your runtime.txt file to use the new version.
remote: !
remote: !       We strongly recommend that you do not pin your app to an
remote: !       exact Python version such as 3.10.12, and instead only specify
remote: !       the major Python version of 3.10 in your runtime.txt file.
remote: !       This will allow your app to receive the latest available Python
remote: !       patch version automatically and prevent this warning.
remote:
remote: -----> Installing Python 3.10.12
remote: -----> Installing pip 25.0.1, setuptools 70.3.0 and wheel 0.45.1
remote: -----> Installing SQLite3
remote: -----> Installing dependencies using 'pip install -r requirements.txt'
remote:       Collecting Flask==3.1.0 (from -r requirements.txt (line 1))
remote:       Downloading flask-3.1.0-py3-none-any.whl.metadata (2.7 kB)
```

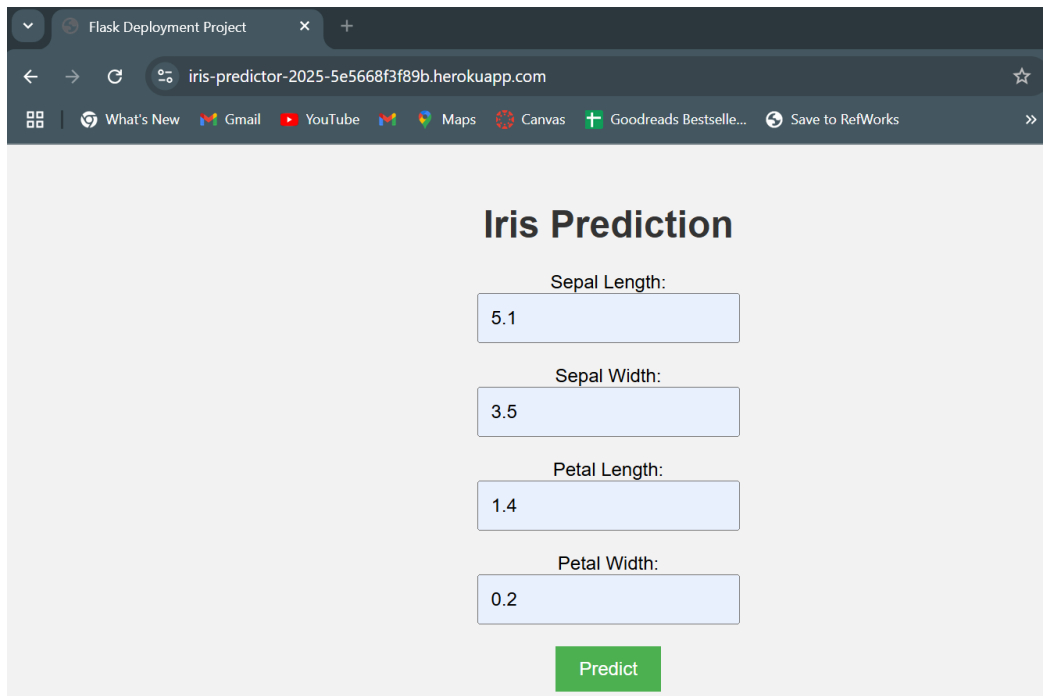
```

remote: Collecting gunicorn==21.2.0 (from -r requirements.txt (line 2))
remote:   Downloading gunicorn-21.2.0-py3-none-any.whl.metadata (4.1 kB)
remote: Collecting numpy==1.24.2 (from -r requirements.txt (line 3))
remote:   Downloading numpy-1.24.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (5.6 kB)
remote: Collecting pandas==1.5.3 (from -r requirements.txt (line 4))
remote:   Downloading pandas-1.5.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (11 kB)
remote: Collecting scikit-learn==1.2.1 (from -r requirements.txt (line 5))
remote:   Downloading scikit_learn-1.2.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (11
kB)
remote: Collecting joblib==1.2.0 (from -r requirements.txt (line 6))
remote:   Downloading joblib-1.2.0-py3-none-any.whl.metadata (5.3 kB)
remote: Collecting xgboost==2.0.3 (from -r requirements.txt (line 7))
remote:   Downloading xgboost-2.0.3-py3-none-manylinux2014_x86_64.whl.metadata (2.0 kB)
remote: Collecting Werkzeug==3.1 (from Flask==3.1.0->-r requirements.txt (line 1))
remote:   Downloading werkzeug-3.1.3-py3-none-any.whl.metadata (3.7 kB)
remote: Collecting Jinja2>=3.1.2 (from Flask==3.1.0->-r requirements.txt (line 1))
remote:   Downloading jinja2-3.1.6-py3-none-any.whl.metadata (2.9 kB)
remote: Collecting itsdangerous>=2.2 (from Flask==3.1.0->-r requirements.txt (line 1))
remote:   Downloading itsdangerous-2.2.0-py3-none-any.whl.metadata (1.9 kB)
remote: Collecting click>=8.1.3 (from Flask==3.1.0->-r requirements.txt (line 1))
remote:   Downloading click-8.1.8-py3-none-any.whl.metadata (2.3 kB)
remote: Collecting blinker>=1.9 (from Flask==3.1.0->-r requirements.txt (line 1))
remote:   Downloading blinker-1.9.0-py3-none-any.whl.metadata (1.6 kB)
remote: Collecting packaging (from gunicorn==21.2.0->-r requirements.txt (line 2))
remote:   Downloading packaging-25.0-py3-none-any.whl.metadata (3.3 kB)
remote: Collecting python-dateutil>=2.8.1 (from pandas==1.5.3->-r requirements.txt (line 4))
remote:   Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl.metadata (8.4 kB)
remote: Collecting pytz>=2020.1 (from pandas==1.5.3->-r requirements.txt (line 4))
remote:   Downloading pytz-2025.2-py2.py3-none-any.whl.metadata (22 kB)
remote: Collecting scipy>=1.3.2 (from scikit-learn==1.2.1->-r requirements.txt (line 5))
remote:   Downloading scipy-1.15.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (61 kB)
remote: Collecting threadpoolctl>=2.0.0 (from scikit-learn==1.2.1->-r requirements.txt (line 5))
remote:   Downloading threadpoolctl-3.6.0-py3-none-any.whl.metadata (13 kB)
remote: Collecting MarkupSafe>=2.0 (from Jinja2>=3.1.2->Flask==3.1.0->-r requirements.txt (line 1))
remote:   Downloading MarkupSafe-3.0.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (4.0 k
B)
remote: Collecting six>=1.5 (from python-dateutil>=2.8.1->pandas==1.5.3->-r requirements.txt (line 4))
remote:   Downloading six-1.17.0-py2.py3-none-any.whl.metadata (1.7 kB)
remote: Downloading flask-3.1.0-py3-none-any.whl (102 kB)
remote: Downloading gunicorn-21.2.0-py3-none-any.whl (80 kB)
remote: Downloading numpy-1.24.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (17.3 MB)
remote: Downloading pandas-1.5.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (12.1 MB)
remote: Downloading scikit_learn-1.2.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (9.6 MB)
remote: Downloading joblib-1.2.0-py3-none-any.whl (297 kB)
remote: Downloading xgboost-2.0.3-py3-none-manylinux2014_x86_64.whl (297.1 MB)
remote: Downloading blinker-1.9.0-py3-none-any.whl (8.5 kB)
remote: Downloading click-8.1.8-py3-none-any.whl (98 kB)
remote: Downloading itsdangerous-2.2.0-py3-none-any.whl (16 kB)
remote: Downloading jinja2-3.1.6-py3-none-any.whl (134 kB)
remote: Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
remote: Downloading pytz-2025.2-py2.py3-none-any.whl (509 kB)
remote: Downloading scipy-1.15.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (37.6 MB)
remote: Downloading threadpoolctl-3.6.0-py3-none-any.whl (18 kB)
remote: Downloading werkzeug-3.1.3-py3-none-any.whl (224 kB)
remote: Downloading packaging-25.0-py3-none-any.whl (66 kB)
remote: Downloading MarkupSafe-3.0.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (20 kB)
remote: Downloading six-1.17.0-py2.py3-none-any.whl (11 kB)
remote: Installing collected packages: pytz, threadpoolctl, six, packaging, numpy, MarkupSafe, joblib, itsdangero
us, click, blinker, Werkzeug, scipy, python-dateutil, Jinja2, gunicorn, xgboost, scikit-learn, pandas, Flask
remote: Successfully installed Flask-3.1.0 Jinja2-3.1.6 MarkupSafe-3.0.2 Werkzeug-3.1.3 blinker-1.9.0 click-8.1.8
gunicorn-21.2.0 itsdangerous-2.2.0 joblib-1.2.0 numpy-1.24.2 packaging-25.0 pandas-1.5.3 python-dateutil-2.9.0.post0 py
tz-2025.2 scikit-learn-1.2.1 scipy-1.15.2 six-1.17.0 threadpoolctl-3.6.0 xgboost-2.0.3
remote: -----> Discovering process types
remote:   Procfile declares types -> web
remote:
remote: -----> Compressing...
remote:   Done: 392M
remote: -----> Launching...
remote:   !   Warning: Your slug size (392 MB) exceeds our soft limit (300 MB) which may affect boot time.
remote:   Released v3
remote:   https://iris-predictor-2025-5e5668f3f89b.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/iris-predictor-2025.git
* [new branch]    main -> main

C:\Users\user\Desktop\Cynthia\codes\Flask_project>

```

## Screenshot of the live Heroku app running (browser view)



A screenshot of a web browser showing the 'Iris Prediction' app. The browser's address bar displays 'iris-predictor-2025-5e5668f3f89b.herokuapp.com'. The app's title is 'Iris Prediction'. It features four input fields for 'Sepal Length:', 'Sepal Width:', 'Petal Length:', and 'Petal Width:'. The values entered are 5.1, 3.5, 1.4, and 0.2 respectively. A green 'Predict' button is located below the input fields.

**Iris Prediction**

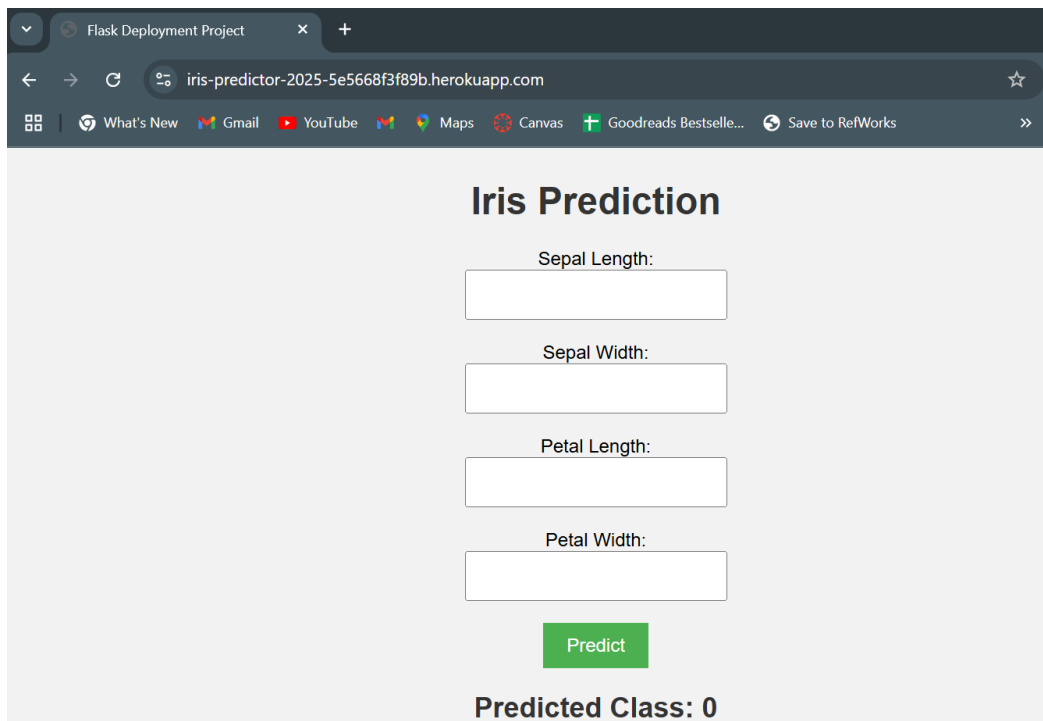
Sepal Length: 5.1

Sepal Width: 3.5

Petal Length: 1.4

Petal Width: 0.2

Predict



A second screenshot of the same 'Iris Prediction' app. The input fields are now empty. Below the 'Predict' button, the text 'Predicted Class: 0' is displayed.

**Iris Prediction**

Sepal Length:

Sepal Width:

Petal Length:

Petal Width:

Predict

**Predicted Class: 0**