

Flask Deployment – Iris Flower Prediction (Example PDF Layout)

Title: Deployment on Flask

Name: Radhika Diyora

Batch Code: LISUM51

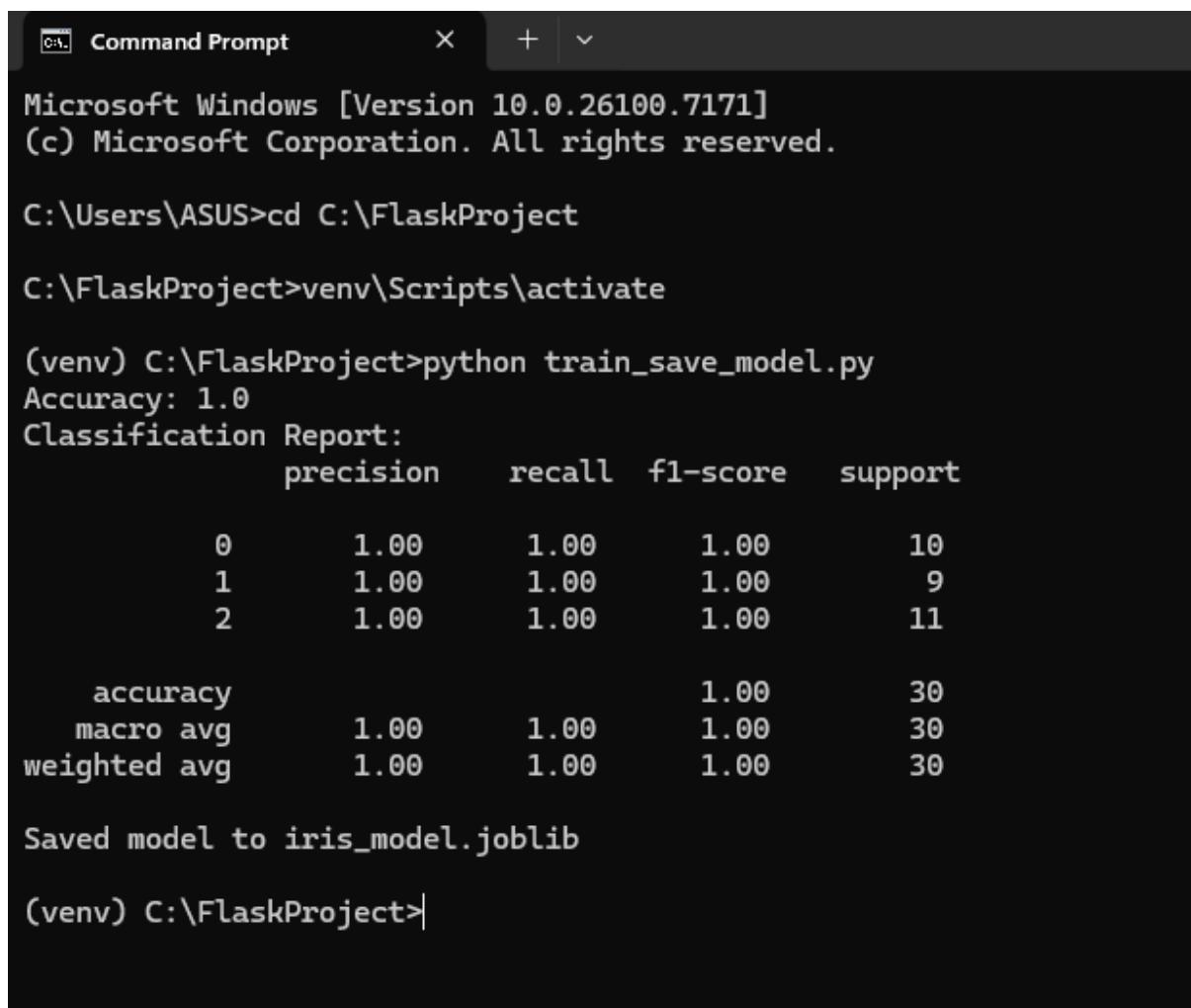
Submission Date: 25-Nov-2025

Submitted to: Data Glacier

1 Training Script Output

Description:

This screenshot shows the model being trained on the Iris dataset. The accuracy, classification report, and confirmation of saved model (iris_model.joblib) are visible.



```
Command Prompt      X + ^

Microsoft Windows [Version 10.0.26100.7171]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS>cd C:\FlaskProject

C:\FlaskProject>venv\Scripts\activate

(venv) C:\FlaskProject>python train_save_model.py
Accuracy: 1.0
Classification Report:
precision    recall   f1-score   support
          0       1.00     1.00     1.00      10
          1       1.00     1.00     1.00       9
          2       1.00     1.00     1.00      11

accuracy                           1.00      30
macro avg                           1.00      1.00      30
weighted avg                        1.00     1.00      30

Saved model to iris_model.joblib

(venv) C:\FlaskProject>
```

2 Flask App Running

Description:

This screenshot shows the Flask web app running locally on `http://127.0.0.1:5000/`. The CMD window shows the app is ready to receive input.

```
(venv) C:\FlaskProject>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: 174-851-603
```

```
Microsoft Windows [Version 10.0.26100.7171]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS>cd C:\FlaskProject

C:\FlaskProject>venv\Scripts\activate

(venv) C:\FlaskProject>python train_save_model.py
Accuracy: 1.0
Classification Report:
precision    recall   f1-score   support
      0       1.00     1.00      1.00      10
      1       1.00     1.00      1.00       9
      2       1.00     1.00      1.00      11

accuracy                           1.00      30
macro avg       1.00     1.00      1.00      30
weighted avg    1.00     1.00      1.00      30

Saved model to iris_model.joblib

(venv) C:\FlaskProject>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: 174-851-603
```

3 Web App Prediction

Description:

This screenshot shows the web page where the user inputs Iris flower features. After submitting the form, the predicted species is displayed.

The screenshot shows a web browser window with the URL 127.0.0.1:5000. At the top, there are navigation icons (back, forward, refresh) and a status bar message: "Set Google Chrome as your default browser and pin it to your taskbar" with a "Set as default" button. Below the header, the title "Iris Flower Prediction" is displayed. The form contains four input fields for Sepal Length (5.1), Sepal Width (3.5), Petal Length (1.4), and Petal Width (0.2). A "Predict" button is located below the input fields.

Iris Flower Prediction

Sepal Length:
Sepal Width:
Petal Length:
Petal Width:

Prediction:

The screenshot shows a web browser window with the URL 127.0.0.1:5000. At the top, there are navigation icons (back, forward, refresh) and a status bar message: "Set Google Chrome as your default browser and pin it to your taskbar" with a "Set as default" button. Below the header, the title "Iris Flower Prediction" is displayed. The prediction result "setosa" is shown in bold text.

Iris Flower Prediction

Sepal Length:
Sepal Width:
Petal Length:
Petal Width:

Prediction: setosa

Example values for testing:

Sepal Length	Sepal Width	Petal Length	Petal Width	Prediction
5.1	3.5	1.4	0.2	setosa
6.0	2.9	4.5	1.5	versicolor
6.5	3.0	5.8	2.2	virginica

4 Workflow / Explanation

Workflow Steps:

1. Train Model: Use `train_save_model.py` to train a Random Forest classifier on the Iris dataset.
2. Save Model: Save the trained model using `joblib` as `iris_model.joblib`.
3. Create Flask App: Build `app.py` to take user input from a web form.
4. Load Model in Flask: Load `iris_model.joblib` and make predictions.
5. Display Prediction: Show the predicted species on the web page (`index.html`).