Name – Apurwa Bhausaheb Sontakke

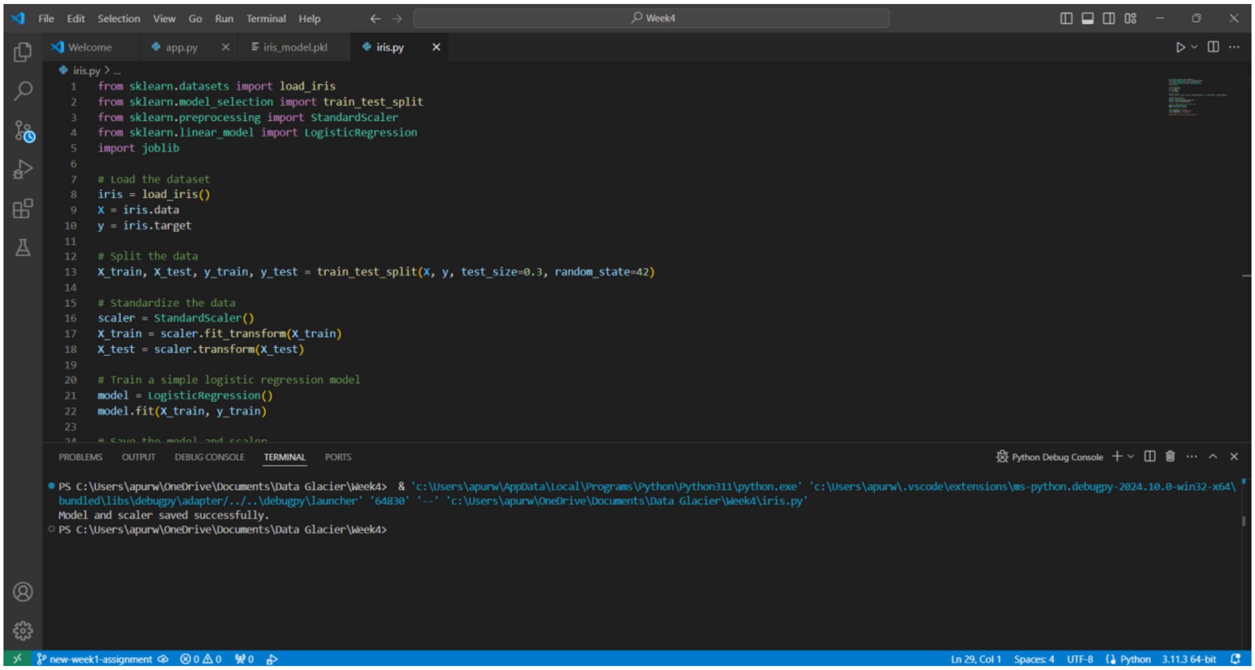
Batch code -LISUM32

Submission date – 03/28/2024

Submitted to – Data Glacier

<u>Snapshots of the Deployment Process</u>

Selected Iris Dataset and used the regression model:



```python
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
import joblib

# Load the dataset
iris = load_iris()
X = iris.data
y = iris.target

# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Standardize the data
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Train a simple logistic regression model
model = LogisticRegression()
model.fit(X_train, y_train)

# Save the model and scaler
```

```
PS C:\Users\apurw\OneDrive\Documents\Data Glacier\Week4>  & 'c:\Users\apurw\AppData\Local\Programs\Python\Python311\python.exe' 'c:\Users\apurw\.vscode\extensions\ms-python.debugpy-2024.10.0-win32-x64\bundled\libs\debugpy\adapter/../..\debugpy\launcher' '64830' '--' 'c:\Users\apurw\OneDrive\Documents\Data Glacier\Week4\iris.py'
Model and scaler saved successfully.
PS C:\Users\apurw\OneDrive\Documents\Data Glacier\Week4>
```

This step involves selecting the Iris dataset, which is used to train a logistic regression model. The dataset includes features such as sepal length, sepal width, petal length, and petal width to classify different species of Iris flowers.

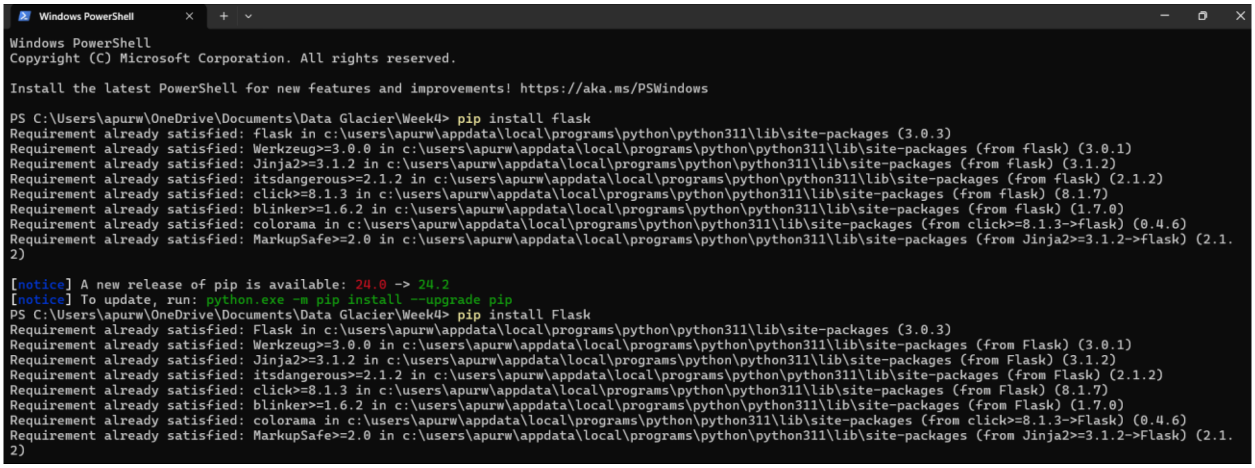
Both the model was saved:

iris_model.pkl

scaler.pkl

This step demonstrates the successful saving of the trained logistic regression model and the scaler used for standardizing the dataset. The files were saved using the joblib library.

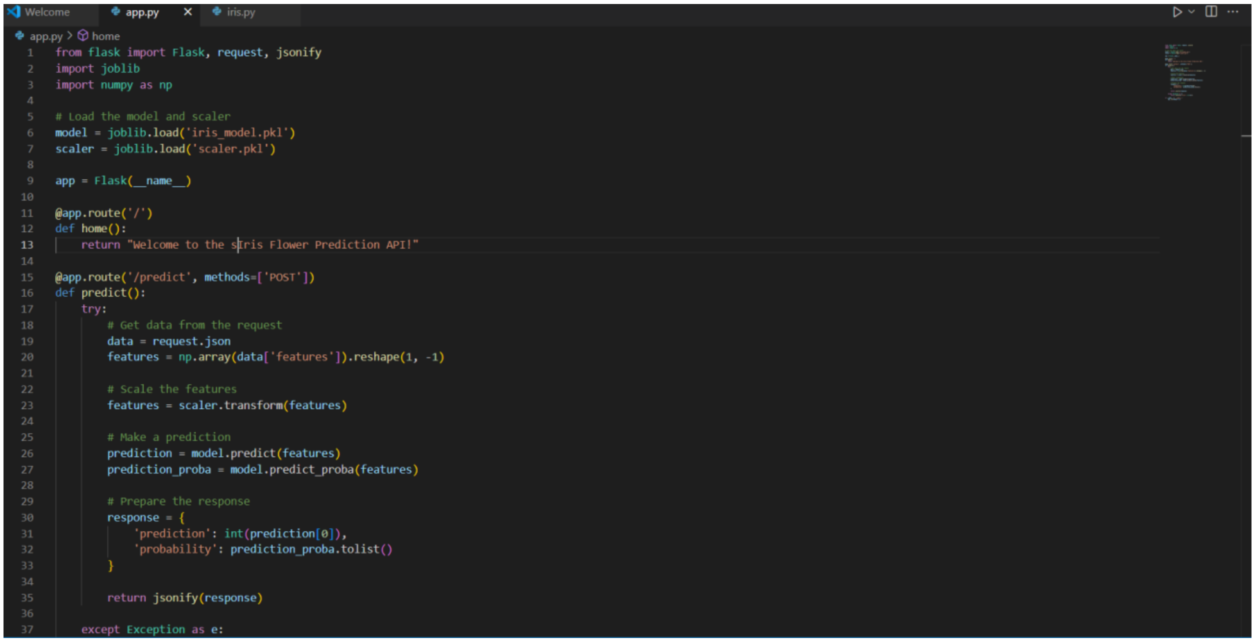
Installed and made sure Flask is running properly



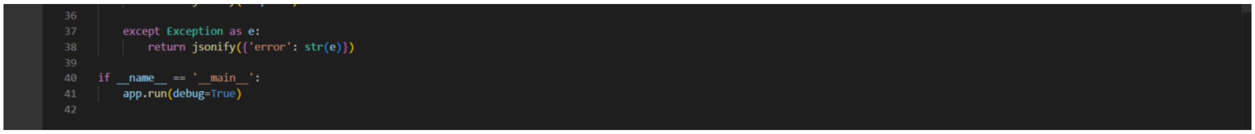
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\apurw\OneDrive\Documents\Data Glacier\Week4> pip install flask
Requirement already satisfied: flask in c:\users\apurw\appdata\local\programs\python\python311\lib\site-packages (3.0.3)
Requirement already satisfied: Werkzeug>=3.0.0 in c:\users\apurw\appdata\local\programs\python\python311\lib\site-packages (from flask) (3.0.1)
Requirement already satisfied: Jinja2>=3.1.2 in c:\users\apurw\appdata\local\programs\python\python311\lib\site-packages (from flask) (3.1.2)
Requirement already satisfied: itsdangerous>=2.1.2 in c:\users\apurw\appdata\local\programs\python\python311\lib\site-packages (from flask) (2.1.2)
Requirement already satisfied: click>=8.1.3 in c:\users\apurw\appdata\local\programs\python\python311\lib\site-packages (from flask) (8.1.7)
Requirement already satisfied: blinker>=1.6.2 in c:\users\apurw\appdata\local\programs\python\python311\lib\site-packages (from flask) (1.7.0)
Requirement already satisfied: colorama in c:\users\apurw\appdata\local\programs\python\python311\lib\site-packages (from click>=8.1.3->flask) (0.4.6)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\apurw\appdata\local\programs\python\python311\lib\site-packages (from Jinja2>=3.1.2->flask) (2.1.
2)

[notice] A new release of pip is available: 24.0 -> 24.2
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\apurw\OneDrive\Documents\Data Glacier\Week4> pip install Flask
Requirement already satisfied: Flask in c:\users\apurw\appdata\local\programs\python\python311\lib\site-packages (3.0.3)
Requirement already satisfied: Werkzeug>=3.0.0 in c:\users\apurw\appdata\local\programs\python\python311\lib\site-packages (from Flask) (3.0.1)
Requirement already satisfied: Jinja2>=3.1.2 in c:\users\apurw\appdata\local\programs\python\python311\lib\site-packages (from Flask) (3.1.2)
Requirement already satisfied: itsdangerous>=2.1.2 in c:\users\apurw\appdata\local\programs\python\python311\lib\site-packages (from Flask) (2.1.2)
Requirement already satisfied: click>=8.1.3 in c:\users\apurw\appdata\local\programs\python\python311\lib\site-packages (from Flask) (8.1.7)
Requirement already satisfied: blinker>=1.6.2 in c:\users\apurw\appdata\local\programs\python\python311\lib\site-packages (from Flask) (1.7.0)
Requirement already satisfied: colorama in c:\users\apurw\appdata\local\programs\python\python311\lib\site-packages (from click>=8.1.3->Flask) (0.4.6)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\apurw\appdata\local\programs\python\python311\lib\site-packages (from Jinja2>=3.1.2->Flask) (2.1.
2)
```

In this step, Flask was installed and verified to be running correctly on the local machine. This is crucial for deploying the machine learning model as a web application.

Flask app: Created a file named app.py -



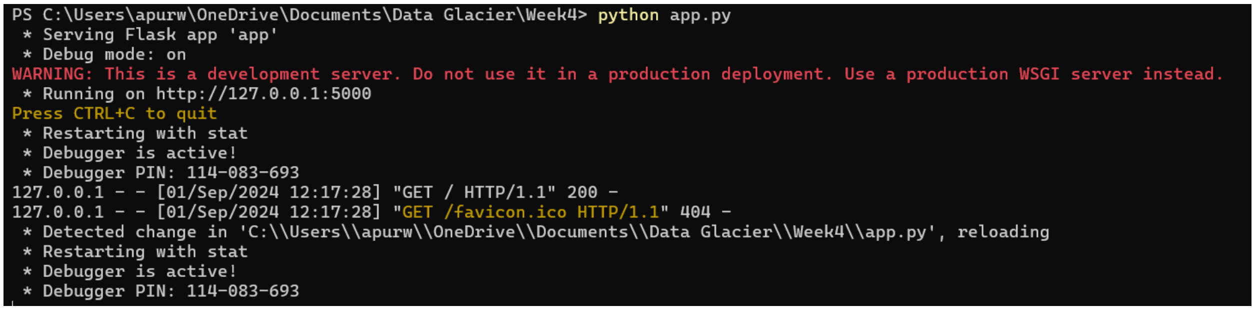
```python
from flask import Flask, request, jsonify
import joblib
import numpy as np

# Load the model and scaler
model = joblib.load('iris_model.pkl')
scaler = joblib.load('scaler.pkl')

app = Flask(__name__)

@app.route('/')
def home():
    return "Welcome to the sIris Flower Prediction API!"

@app.route('/predict', methods=['POST'])
def predict():
    try:
        # Get data from the request
        data = request.json
        features = np.array(data['features']).reshape(1, -1)

        # Scale the features
        features = scaler.transform(features)

        # Make a prediction
        prediction = model.predict(features)
        prediction_proba = model.predict_proba(features)

        # Prepare the response
        response = {
            'prediction': int(prediction[0]),
            'probability': prediction_proba.tolist()
        }

        return jsonify(response)

    except Exception as e:
```



```python

    except Exception as e:
        return jsonify({'error': str(e)})

if __name__ == '__main__':
    app.run(debug=True)

```

This step highlights the creation of the app.py file, which includes the code for the Flask web application. The application serves the trained model for predictions via a REST API.

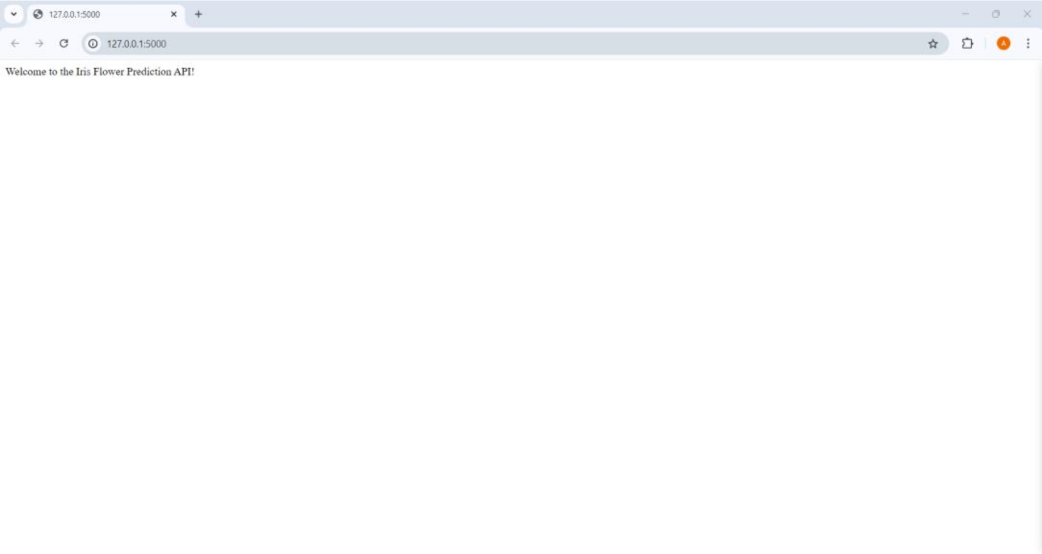
Running the Flask App



```
PS C:\Users\apurw\OneDrive\Documents\Data Glacier\Week4> 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: 114-083-693
127.0.0.1 - - [01/Sep/2024 12:17:28] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [01/Sep/2024 12:17:28] "GET /favicon.ico HTTP/1.1" 404 -
 * Detected change in 'C:\\Users\\apurw\\OneDrive\\Documents\\Data Glacier\\Week4\\app.py', reloading
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 114-083-693
```

This step shows the Flask application being executed in the terminal. The app is now running and ready to accept HTTP requests for predictions.

Deployment final output on server:



Welcome to the Iris Flower Prediction API!

This final step displays the successful deployment of the Flask app, with the application running on a local server. The message 'Welcome to the Iris Flower Prediction API!' confirms that the server is live and the application is accessible through the browser.