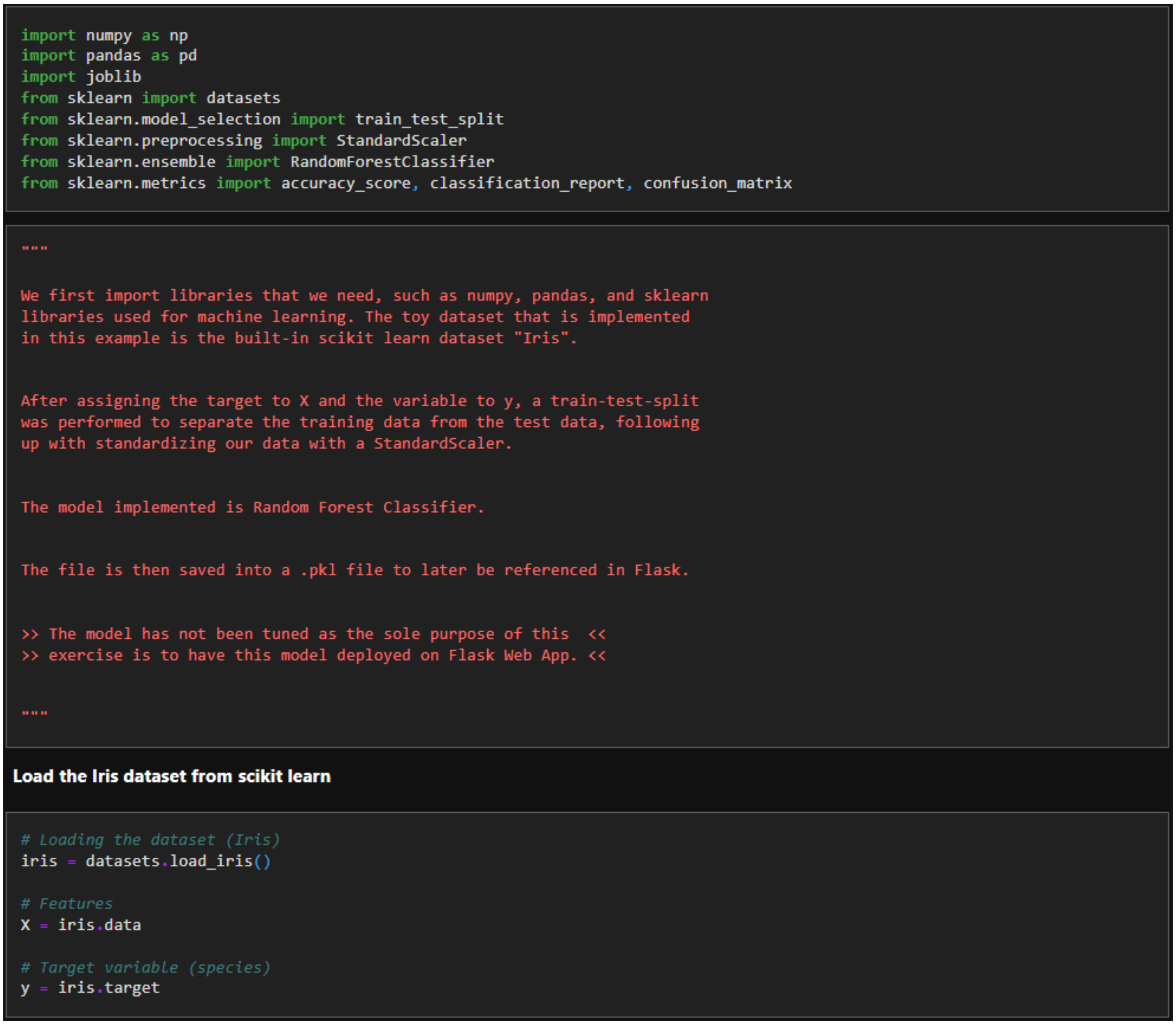
Name:	Ansel Vallejo
Batch Code:	LISUM25
Submission Date:	2023-09-27
Submitted To:	Data Glacier

Deployment on Flask

1. Created a .ipynb notebook called “Iris Model.ipynb”, where I created a model from toy data used from scikit lab called “Iris”



```
import numpy as np
import pandas as pd
import joblib
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

```
"""

We first import libraries that we need, such as numpy, pandas, and sklearn
libraries used for machine learning. The toy dataset that is implemented
in this example is the built-in scikit learn dataset "Iris".


After assigning the target to X and the variable to y, a train-test-split
was performed to separate the training data from the test data, following
up with standardizing our data with a StandardScaler.


The model implemented is Random Forest Classifier.


The file is then saved into a .pkl file to later be referenced in Flask.


>> The model has not been tuned as the sole purpose of this  <<
>> exercise is to have this model deployed on Flask Web App. <<


"""
```

Load the Iris dataset from scikit learn

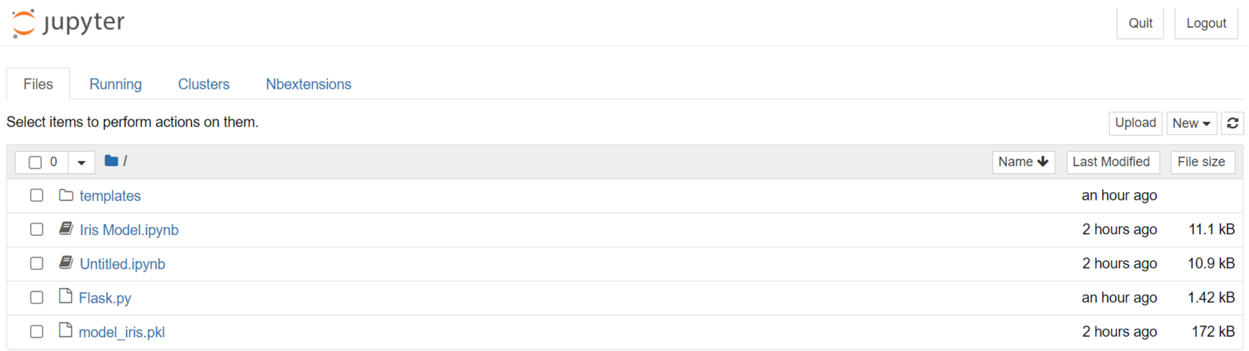
```
# Loading the dataset (Iris)
iris = datasets.load_iris()

# Features
X = iris.data

# Target variable (species)
y = iris.target
```

(The model was not tuned to perform accurate predictions – it was build on the surface to satisfy the Flask Model Deployment assignment)

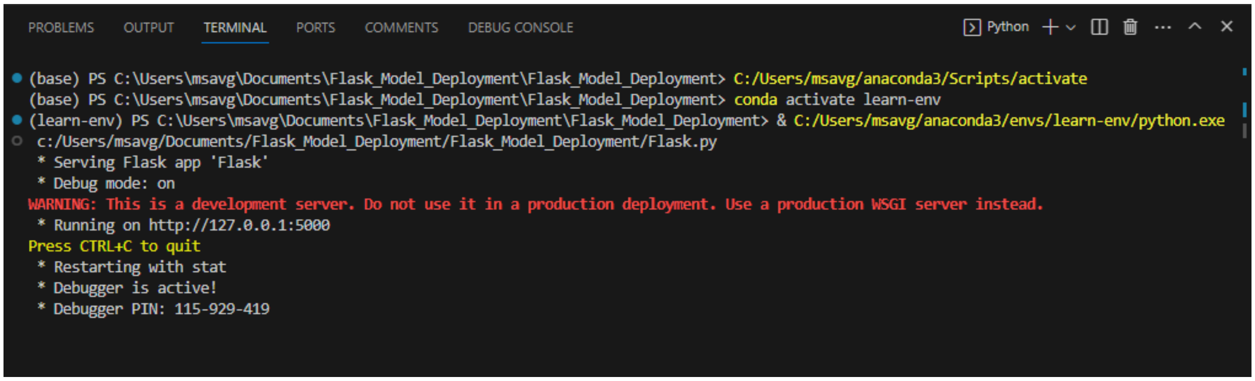
2. A Python script (Flask.py) was created to implement model deployment via Flask



Select items to perform actions on them.

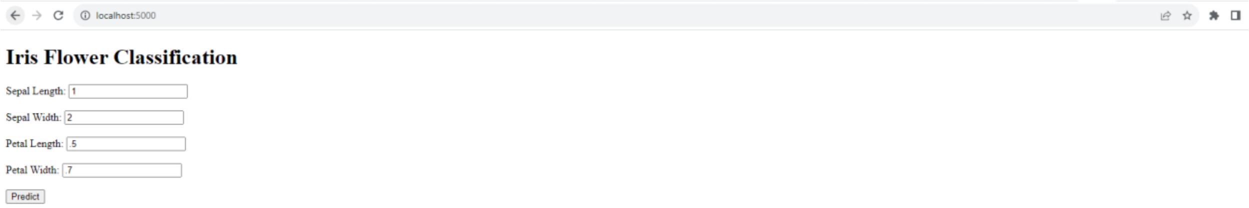
Name	Last Modified	File size
templates	an hour ago	
Iris Model.ipynb	2 hours ago	11.1 kB
Untitled.ipynb	2 hours ago	10.9 kB
Flask.py	an hour ago	1.42 kB
model_iris.pkl	2 hours ago	172 kB

3. Running Flask.py (Visual Studio)



```
(base) PS C:\Users\msavg\Documents\Flask_Model_Deployment\Flask_Model_Deployment> C:/Users/msavg/anaconda3/Scripts/activate
(base) PS C:\Users\msavg\Documents\Flask_Model_Deployment\Flask_Model_Deployment> conda activate learn-env
(learn-env) PS C:\Users\msavg\Documents\Flask_Model_Deployment\Flask_Model_Deployment> & C:/Users/msavg/anaconda3/envs/learn-env/python.exe
 c:/Users/msavg/Documents/Flask_Model_Deployment/Flask_Model_Deployment/Flask.py
 * Serving Flask app 'Flask'
 * 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: 115-929-419
```

4. Flask HTML User Interface (localhost:5000)



localhost:5000

Iris Flower Classification

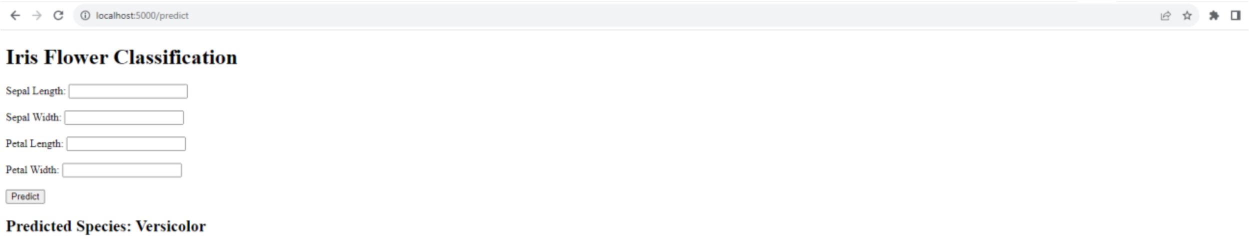
Sepal Length: 1

Sepal Width: 2

Petal Length: .5

Petal Width: .7

5. Submitting values to be predicted (with result)



localhost:5000/predict

Iris Flower Classification

Sepal Length:

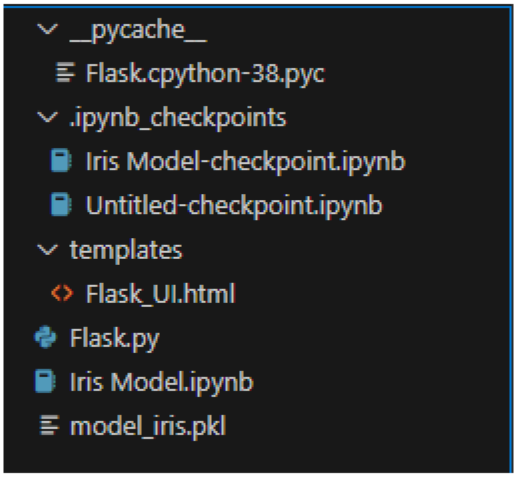
Sepal Width:

Petal Length:

Petal Width:

Predicted Species: Versicolor

File Structure in GitHub



```
⌄ __pycache__
    Flask.cpython-38.pyc
⌄ .ipynb_checkpoints
    Iris Model-checkpoint.ipynb
    Untitled-checkpoint.ipynb
⌄ templates
    Flask_UI.html
  Flask.py
  Iris Model.ipynb
  model_iris.pkl
```