



Data Glacier Virtual Internship
Week 5: Cloud API Development

Name: Yusuf Yuhan

Batch Code: LISUM19

Date: 05 April 2023

Submitted to: Data Glacier

Introduction

in this project we are creating flower class prediction machine learning app with python and flask and publish that to a Heroku cloud application platform.

Data information

Iris.csv

Csv file with the size of 4 kb which contains 5 attributes
Sepal_Length, Sepal_Width, Petal_Length, Petal_Width, Class
contains 151 rows of excel data

Building model model.py

Importing libraries

```
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
import pickle
```

Load csv file

```
# Load the csv file
df = pd.read_csv("iris.csv")

print(df.head())
```

Building model

```
# Select independent and dependent variable
X = df[["Sepal_Length", "Sepal_Width", "Petal_Length", "Petal_Width"]]
y = df["Class"]

# Split the dataset into train and test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=50)

# Feature scaling
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

# Instantiate the model
classifier = RandomForestClassifier()

# Fit the model
classifier.fit(X_train, y_train)
```

Save model

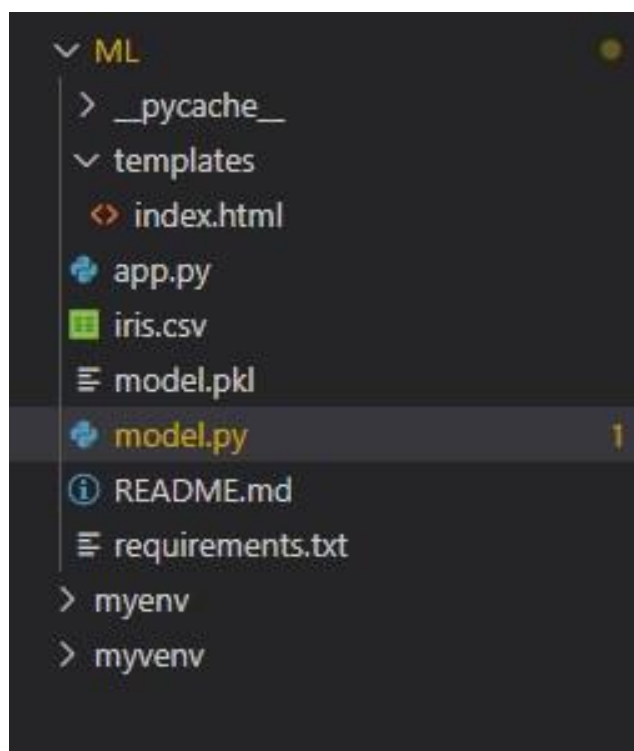
```
# Instantiate the model
classifier = RandomForestClassifier()

# Fit the model
classifier.fit(X_train, y_train)

# Make pickle file of our model
pickle.dump(classifier, open("model.pkl", "wb"))
```

Loading...

Turning model into a web application



App.py

```
app.py x
ML > app.py > ...
1 import numpy as np
2 from flask import Flask, request, jsonify, render_template
3 import pickle
4
5 # Create flask app
6 flask_app = Flask(__name__)
7 model = pickle.load(open("model.pkl", "rb"))
8
9 @flask_app.route("/")
10 def Home():
11     return render_template("index.html")
12
13 @flask_app.route("/predict", methods = ["POST"])
14 def predict():
15     float_features = [float(x) for x in request.form.values()]
16     features = [np.array(float_features)]
17     prediction = model.predict(features)
18     return render_template("index.html", prediction_text = "The flower species is {}".format(prediction))
19
20 if __name__ == "__main__":
21     flask_app.run(debug=True)
```

Import necessary libraries: The code imports numpy, Flask, request, jsonify, render_template, and pickle libraries.

Load the trained model: The code loads the trained Random Forest Classifier model from the file "model.pkl" using the pickle library.

Create Flask app and routes: The code creates a Flask app and defines two routes:

The "/" route displays the homepage of the web application using an HTML template.

The "/predict" route takes the input data from a form submitted by the user, converts it to a numpy array, and uses the trained model to make a prediction for the species of iris flower. The prediction is then displayed on the homepage using another HTML template.

Define the HTML templates: The code defines two HTML templates:

"index.html" is the homepage template that displays the input form and the prediction text.

"result.html" is the result template that displays the prediction text.

Run the Flask app: The code runs the Flask app in debug mode using the "run" method.

Index.html

```
ML > templates > <> index.html > ...
1  <!DOCTYPE html>
2  <html >
3  <!--From https://codepen.io/frytyler/pen/EGdtg-->
4  <head>
5      <meta charset="UTF-8">
6      <title>ML App</title>
7      <link href="https://fonts.googleapis.com/css?family=Pacifico" rel="stylesheet" type="text/css">
8      <link href="https://fonts.googleapis.com/css?family=Arimo" rel="stylesheet" type="text/css">
9      <link href="https://fonts.googleapis.com/css?family=Hind:300" rel="stylesheet" type="text/css">
10     <link href="https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300" rel="stylesheet" type="text/css">
11
12 </head>
13
14 <body>
15     <div class="login">
16         <h1>Flower Class Prediction</h1>
17
18         <!-- Main Input For Receiving Query to our ML -->
19         <form action="{{ url_for('predict')}}" method="post">
20             <input type="text" name="Sepal_Length" placeholder="Sepal_Length" required="required" />
21             <input type="text" name="Sepal_Width" placeholder="Sepal_Width" required="required" />
22             <input type="text" name="Petal_Length" placeholder="Petal_Length" required="required" />
23             <input type="text" name="Petal_Width" placeholder="Petal_Width" required="required" />
24
25             <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
26         </form>
27
28         <br>
29         <br>
30         {{ prediction_text }}
31
32     </div>
```

Running web application in web browser using flask

```
Nabeel@mlyyuhann MINGW64 ~/downloads/ML
$ source myenv/Scripts/activate
(myenv)
Nabeel@mlyyuhann MINGW64 ~/downloads/ML
$ cd ML
(myenv)
Nabeel@mlyyuhann MINGW64 ~/downloads/ML/ML
$ pip freeze > requirements.txt
(myenv)
Nabeel@mlyyuhann MINGW64 ~/downloads/ML/ML
$ export FLASK_APP=app.py
(myenv)
Nabeel@mlyyuhann MINGW64 ~/downloads/ML/ML
$ export FLASK_ENV=development
(myenv)
Nabeel@mlyyuhann MINGW64 ~/downloads/ML/ML
$ flask run
```

Opening web browser in http://127.0.0.1:5000

```
warnings.warn(
C:\Users\Nabeel\downloads\ML\myenv\lib\site-packages\sklearn\base.py:318: UserWarning: Trying to unpickle estimator RandomForestClassifier from version 0.23.1 when using version 1.2.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
warnings.warn(
'FLASK_ENV' is deprecated and will not be used in Flask 2.3. Use 'FLASK_DEBUG' instead.
'FLASK_ENV' is deprecated and will not be used in Flask 2.3. Use 'FLASK_DEBUG' instead.
* Serving Flask app 'app.py'
* 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
'FLASK_ENV' is deprecated and will not be used in Flask 2.3. Use 'FLASK_DEBUG' instead.
C:\Users\Nabeel\downloads\ML\myenv\lib\site-packages\sklearn\base.py:318: UserWarning: Trying to unpickle estimator DecisionTreeClassifier from version 0.23.1 when using version 1.2.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
warnings.warn(
C:\Users\Nabeel\downloads\ML\myenv\lib\site-packages\sklearn\base.py:318: UserWarning: Trying to unpickle estimator RandomForestClassifier from version 0.23.1 when using version 1.2.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
warnings.warn(
'FLASK_ENV' is deprecated and will not be used in Flask 2.3. Use 'FLASK_DEBUG' instead.
'FLASK_ENV' is deprecated and will not be used in Flask 2.3. Use 'FLASK_DEBUG' instead.
* Debugger is active!
* Debugger PIN: 724-722-346
```

Flower Class Prediction

Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Predict
--------------	-------------	--------------	-------------	---------

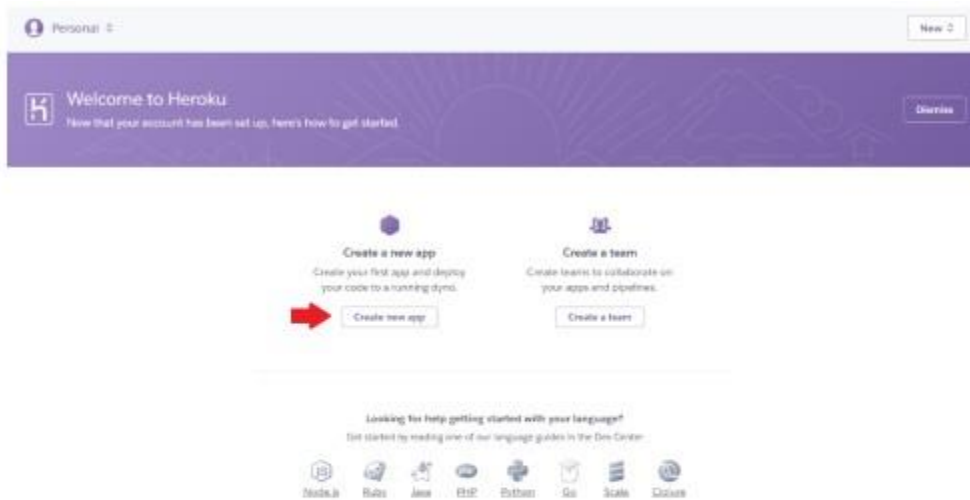
The flower species is ['Virginica']

Flower Class Prediction

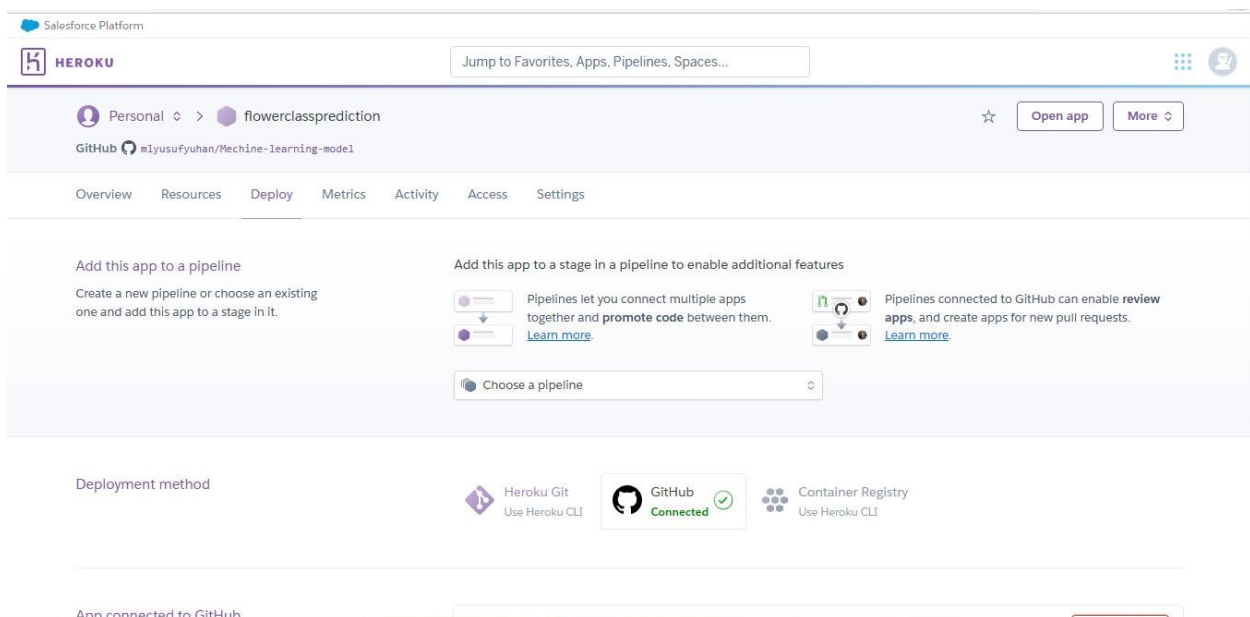
5	2	3.5	1	Predict
---	---	-----	---	---------

The flower species is ['Virginica']

Deploy the machine learning model to Heroku



Connect app to git repository



Salesforce Platform

HEROKU

Jump to Favorites, Apps, Pipelines, Spaces...

App connected to GitHub
Code diffs, manual and auto deploys are available for this app.

Connected to [mlyusufyuhan/Mechine-learning-model](#) by [mlyusufyuhan](#) [Disconnect...](#)
Releases in the [activity feed](#) link to GitHub to view commit diffs

Automatic deploys
Enables a chosen branch to be automatically deployed to this app.

You can now change your main deploy branch from "master" to "main" for both manual and automatic deploys, please follow the instructions [here](#).

Enable automatic deploys from GitHub
Every push to the branch you specify here will deploy a new version of this app. **Deploys happen automatically:** be sure that this branch is always in a deployable state and any tests have passed before you push. [Learn more](#).
Choose a branch to deploy

main

☐ Wait for CI to pass before deploy
Only enable this option if you have a Continuous Integration service configured on your repo.

Enable Automatic Deploys

Successfully deployed app

Salesforce Platform

HEROKU

Jump to Favorites, Apps, Pipelines, Spaces...

Manual deploy
Deploy the current state of a branch to this app.

Deploy a GitHub branch
This will deploy the current state of the branch you specify below. [Learn more](#).
Choose a branch to deploy

main

[Deploy Branch](#)
Receive code from GitHub

✓

Build main 7d24107c

✓

Release phase

✓

Deploy to Heroku

✓

Your app was successfully deployed.
[View](#)

heroku.com Blogs Careers Documentation [Support](#)

Terms of Service Privacy Cookies © 2023 Salesforce.com