

Deployment On Flask

Name: SONIYA SUNNY

Batch Code: LISUM10: 30

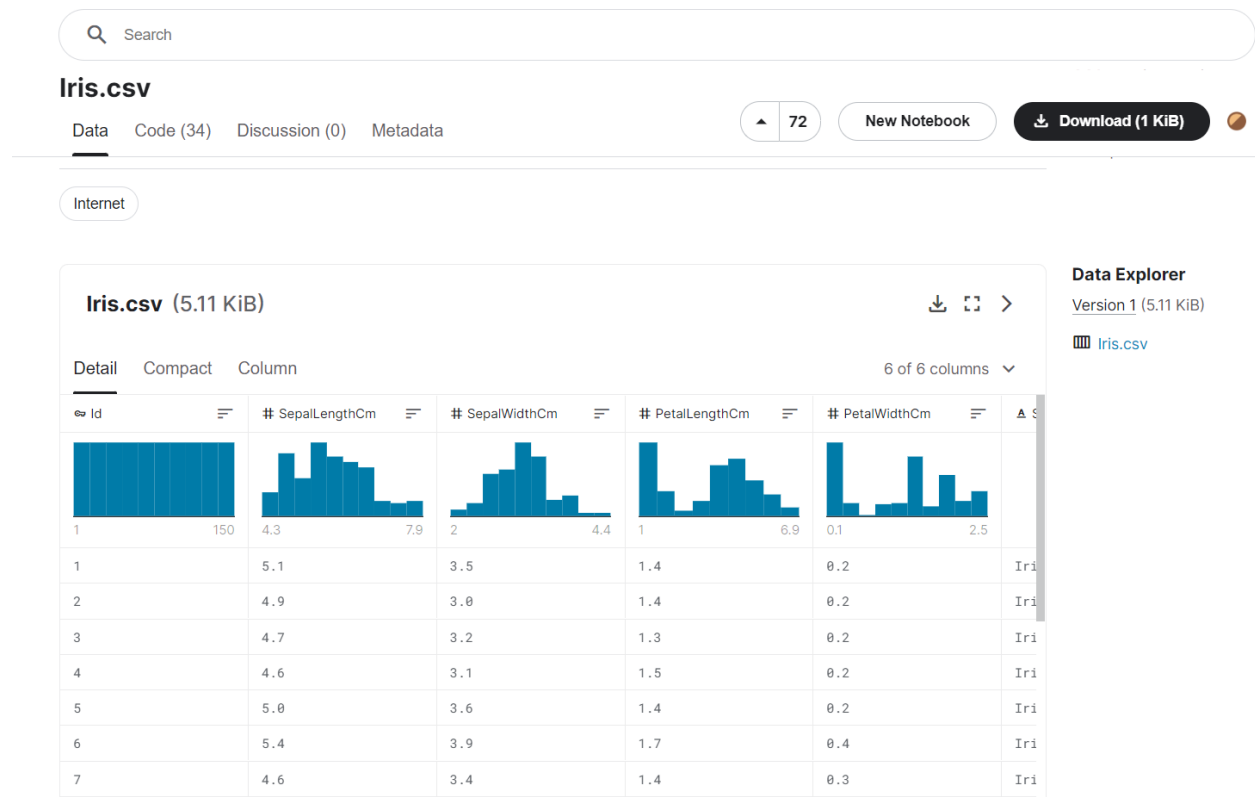
Submission Date: June 23, 2022

Submitted to: Data Glacier

Steps:

1. Data Collection

Selected the famous Iris Dataset, from Kaggle website.



2. Model Building

```

# Importing necessary libraries
import numpy as np
import pandas as pd
from pandas import Series, DataFrame
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
import pickle

# Reading the data
iris = pd.read_csv("Iris.csv")
print(iris.head())
iris.drop("Id", axis=1, inplace = True)
y = iris['Species']
iris.drop(columns='Species',inplace=True)
X = iris[['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']]

# Training the model
x_train,x_test,y_train,y_test = train_test_split(X,y, test_size=0.3)
model = LogisticRegression()
model.fit(x_train,y_train)

pickle.dump(model,open('model.pkl','wb'))









```

[1]

...	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

Built a Logistic Regression model and saved the model as a pickle file in the same directory.

3. Added more folders

Name	Date modified	Type	Size
 .git	2022-06-23 1:07 PM	File folder	
 static	2022-06-23 12:21 PM	File folder	
 templates	2022-06-23 12:21 PM	File folder	
 app.py	2022-06-23 1:06 PM	Python Source File	1 KB
 Iris.csv	2022-06-23 12:22 PM	Microsoft Excel Co...	5 KB
 iris_app.ipynb	2022-06-23 12:52 PM	Jupyter Source File	3 KB
 iris_model.ipynb	2022-06-23 12:24 PM	Jupyter Source File	3 KB
 model.pkl	2022-06-23 12:23 PM	PKL File	1 KB

Added static folder and templates folder.

4. Added index.html file in templates folder
5. Added style.css file in static folder
6. Created app.py file
7. Created iris_app.ipynb file for sample execution

```
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle

app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/predict',methods=['POST'])

def predict():
    init_features = [float(x) for x in request.form.values()]
    final_features = [np.array(init_features)]
    prediction = model.predict(final_features)
    return render_template('index.html', prediction_text='Predicted Class: {}'.format(prediction))

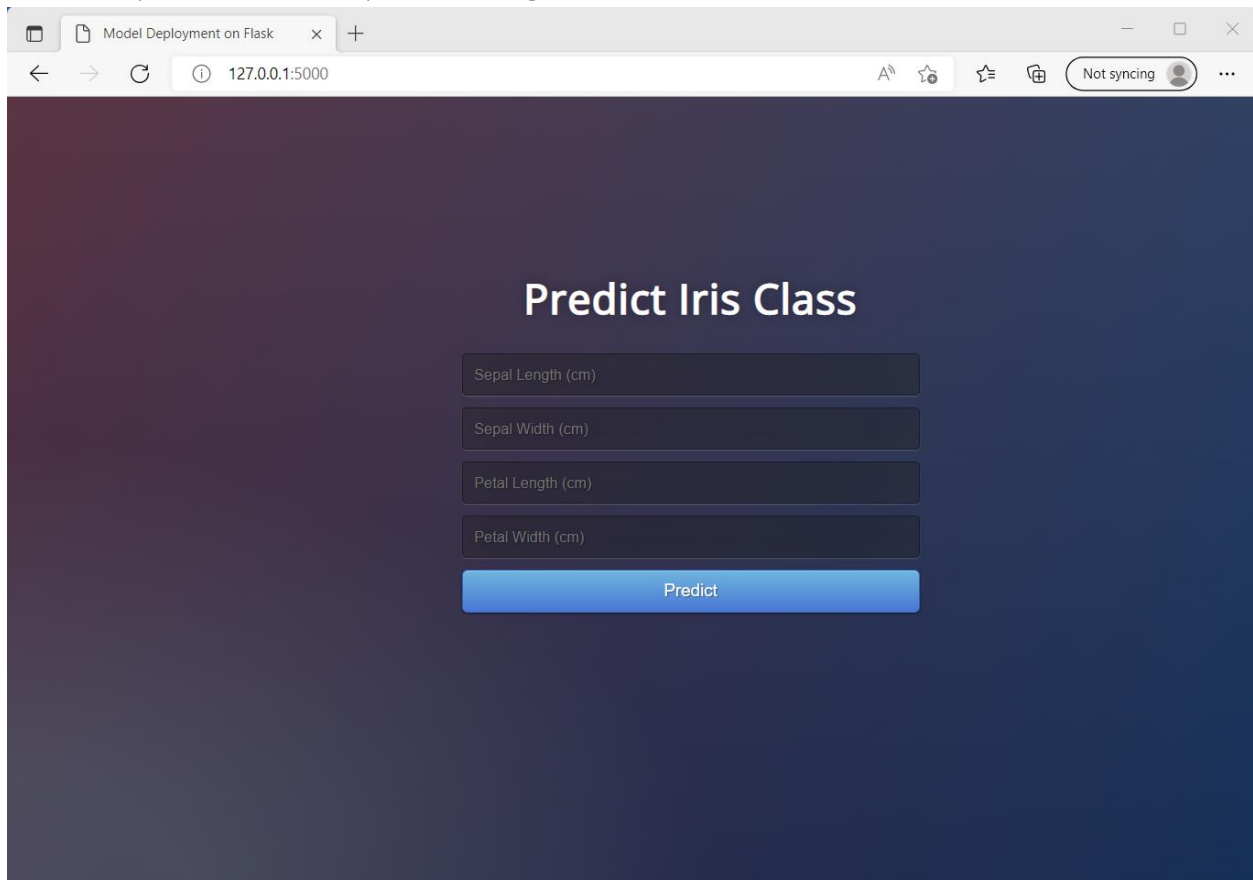
if __name__ == "__main__":
    app.run()
```

[1] ✓ 29.4s

```
... * Serving Flask app "__main__" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off

* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [23/Jun/2022 12:44:58] "GET / HTTP/1.1" 200 -
```

8. Copied the url and opened through a browser



The screenshot shows a web browser window with the title 'Model Deployment on Flask'. The address bar displays '127.0.0.1:5000'. The main content area has a dark blue gradient background and features the heading 'Predict Iris Class' in white. Below the heading, there are four dark blue input fields with white text labels: 'Sepal Length (cm)', 'Sepal Width (cm)', 'Petal Length (cm)', and 'Petal Width (cm)'. At the bottom of these fields is a blue button with the text 'Predict' in white.

9. Tested the model

Model Deployment on Flask

127.0.0.1:5000/predict

Predict Iris Class

5.1

3.5

1.4

2

Predict

Predicted Class: ['Iris-setosa']