

Name	Anju Paul
Batch Code	LISUM10: 30
Submission Date	03-07-2022

### **Deployment in Heroku using webapp:**

#### 1) Select the Dataset:

The IRIS dataset from Kaggle

url: <https://gist.github.com/netj/8836201>

#### 2) Save the Model:

Read the data, identify the features, divide it to test and train, fit and transform the model, and save the model.

```
model.py x app.py index.html
model.py > ...
1 import pandas as pd
2 import numpy as np
3 from sklearn.linear_model import LogisticRegression
4
5 df = pd.read_csv('./iris.csv')
6
7 print(df.head())
8
9 from sklearn.preprocessing import LabelEncoder
10 le = LabelEncoder()
11
12 df['variety'] = le.fit_transform(df['variety'])
13 print(df.head())
14
15 # Import the feature values:
16 x = df.iloc[:, 0:-1].values
17 y = df.iloc[:, -1].values
18
19
20 # Split the dataset into train and test
21 from sklearn.model_selection import train_test_split
22 X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=50)
23
24 # Feature scaling
25 from sklearn.preprocessing import StandardScaler
26 sc = StandardScaler()
27 X_train = sc.fit_transform(X_train)
28 X_test= sc.transform(X_test)
29
30
31 print("X_train : ")
32 print(X_train)
33
34 print("y_train : ")
35 print(X_test)
36
37 log_reg = LogisticRegression(max_iter = 1000)
38 log_reg.fit(X_train, y_train)
39
40 prediction = log_reg.predict(X_test)
41 print(prediction)
42
43 import pickle
44 pickle.dump(log_reg, open('model.pickle', 'wb'))
45
46 |
```

### 3) Deploy the model in Flask

Create an html web page to accept the inputs to identify the type of flower.

```
model.py app.py index.html X
templates > index.html > ...
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5     <meta charset="UTF-8">
6     <title>ML API</title>
7 </head>
8
9 <body>
10     <div class="login">
11         <h1>Predict Flower Type</h1>
12
13         <form action = "{{ url_for('predict')}}" method="post">
14             <input type="text" name="sepal_length" placeholder="Sepal Length" required="required" />
15             <input type="text" name="sepal_width" placeholder="Sepal Width" required="required" />
16             <input type="text" name="petal_length" placeholder="Petal Length" required="required" />
17             <input type="text" name="petal_width" placeholder="Petal Width" required="required" />
18
19             <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
20         </form>
21
22         <br>
23         <br>
24         {{ prediction_text }}
25     </div>
26 </body>
27 </html>
```

Create app.py to render the above webpage html to accept the inputs and predict the output using the saved model.

```
model.py  app.py  x  index.html
app.py > ...
1  import numpy as np
2  from flask import Flask, request, render_template
3  import pickle
4
5  app = Flask(__name__)
6  model = pickle.load(open('./model.pickle', 'rb'))
7
8
9  @app.route('/')
10 def home():
11     return render_template('index.html')
12
13 @app.route('/predict', methods=['POST'])
14 def predict():
15
16     features = [float(x) for x in request.form.values()]
17     final_features = [np.array(features)]
18     prediction = model.predict(final_features)
19
20     output = prediction[0]
21     if output == 1 :
22         o1 = 'Setosa'
23     elif output == 2:
24         o1 = 'Versicolor'
25     else:
26         o1 = 'Virginica'
27
28
29     return render_template('index.html', prediction_text='The category of the flower is {}'.format(o1))
30
31 if __name__ == "__main__":
32     app.run(port=5000, debug=True)
```

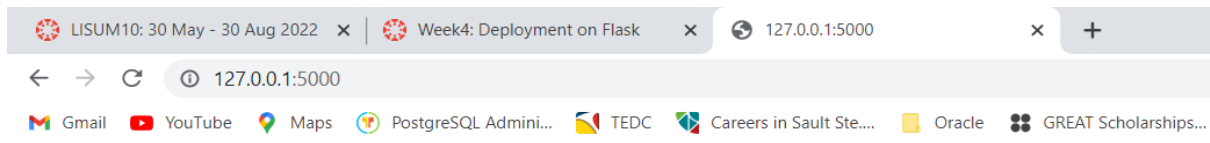
Run the app.py script

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  JUPYTER
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\Data Glacier Virtual Internship\Flask Deployment> & 'C:\Users\Anju Paul\AppData\Local\Programs\Python\Python310\python.exe' 'c:\Users\Anju Paul\.vscode\extensions\ms-python.python-2022.8.1\pythonFiles\lib\python\debugpy\launcher' '53770' '--' 'd:\Data Glacier Virtual Internship\Flask Deployment\app.py'
* Serving Flask app 'app' (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: on
* Restarting with watchdog (windowsapi)
* Debugger is active!
* Debugger PIN: 349-272-658
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Open the url in the web browser.



ML API

## Predict Flower Type

<input type="text" value="Sepal Length"/>	<input type="text" value="Sepal Width"/>	<input type="text" value="Petal Length"/>	<input type="text" value="Petal Width"/>	<input type="button" value="Predict"/>
---	--	---	--	--

Fill in the values and click predict.

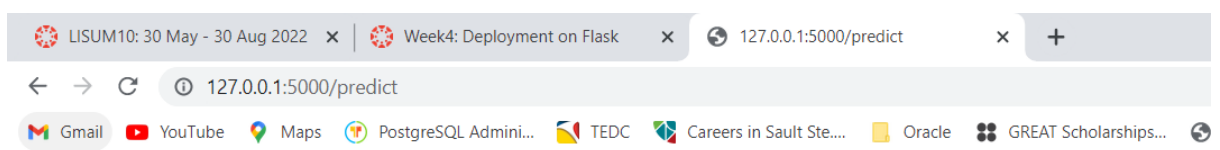


ML API

## Predict Flower Type

<input type="text" value="6.8"/>	<input type="text" value="3.1"/>	<input type="text" value="2.0"/>	<input type="text" value="4"/>	<input type="button" value="Predict"/>
----------------------------------	----------------------------------	----------------------------------	--------------------------------	--

The output will be as shown in the below figure.



ML API

## Predict Flower Type

<input type="text" value="Sepal Length"/>	<input type="text" value="Sepal Width"/>	<input type="text" value="Petal Length"/>	<input type="text" value="Petal Width"/>	<input type="button" value="Predict"/>
---	--	---	--	--

The category of the flower is Versicolor

## Deploy the model to cloud

### 1) Upload the scripts to the Github library

The screenshot shows the GitHub repository page for 'AnjuPaul14 / HerokuDeployment'. The repository is public and has 1 branch (main) and 0 tags. The commit history shows a series of updates to the repository, including adding templates, LICENSE, Procfile, README.md, app.py, iris.csv, model.pickle, model.py, and requirements.txt. The most recent commit is 'ed877ff' from 4 minutes ago, with 9 commits in total.

File	Commit Message	Time Ago
templates	Add files via upload	2 hours ago
LICENSE	Initial commit	2 hours ago
Procfile	Update Procfile	4 minutes ago
README.md	Initial commit	2 hours ago
app.py	Update app.py	10 minutes ago
iris.csv	Add files via upload	2 hours ago
model.pickle	Add files via upload	2 hours ago
model.py	Add files via upload	2 hours ago
requirements.txt	Update requirements.txt	16 minutes ago

### 2) Create Heroku Account

The screenshot shows the Heroku dashboard. The user is logged in as 'Personal'. The dashboard displays a welcome message: 'Welcome to Heroku. Now that your account has been set up, here's how to get started.' There is a 'Dismiss' button and a 'Show next steps' link.

### 3) Link the Github account to Heroku and connect the repository

The screenshot shows the Heroku deployment method configuration page. Under 'Deployment method', there are three options: 'Heroku Git' (Use Heroku CLI), 'GitHub' (Connected), and 'Container Registry' (Use Heroku CLI). The 'GitHub' option is selected and marked as 'Connected'. Below this, it says 'App connected to GitHub. Code diffs, manual and auto deploys are available for this app.' There is a 'Disconnect...' button. A message indicates the app is connected to 'AnjuPaul14/HerokuDeployment' by 'AnjuPaul14'. A link to 'Releases in the activity feed' is provided to view commit diffs.

### 4) Deploy the Github branch

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` `ed877ffe`

Release phase

Deploy to Heroku

✓

✓

✓

✓

Your app was successfully deployed.

[View](#)

## 5) Test the app

url : <https://heroku-flowertype-prediction.herokuapp.com/predict>

←

→

↻

heroku-flowertype-prediction.herokuapp.com/predict

Gmail

YouTube

Maps

PostgreSQL Admini...

TEDC

Careers in Sault Ste...

Oracle

GREAT Scholar

ML API

## Predict Flower Type

3

2

1

1.2

Predict

←

→

↻

heroku-flowertype-prediction.herokuapp.com/predict

Gmail

YouTube

Maps

PostgreSQL Admini...

TEDC

Careers in Sault Ste...

Oracle

GREAT Scholarships...

ML API

## Predict Flower Type

Sepal Length

Sepal Width

Petal Length

Petal Width

Predict

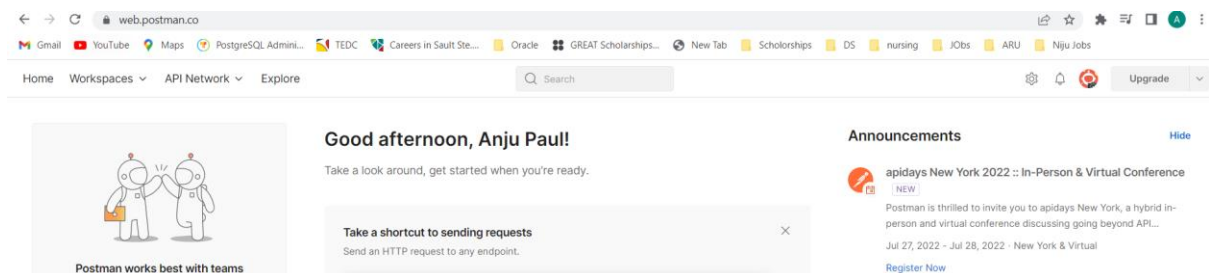
The category of the flower is Versicolor

## Deployment using API:

- 1) Create the model and save it
- 2) Create the API for the prediction

```
app.py index.html app1.py x
templates > app1.py > ...
1 from flask import Flask, jsonify, request
2 import pickle
3 import pandas as pd
4
5 app1 = Flask(__name__)
6
7 @app1.route('/', methods = ['GET','POST'])
8 def home():
9     if(request.method == 'GET'):
10         data = "Welcome to Flower Type Prediction App"
11         return jsonify({'data' : data})
12
13 @app1.route('/predict/')
14 def type_predict():
15     model = pickle.load(open('model.pickle','rb'))
16     sepal_length = request.args.get('sepal_length')
17     sepal_width = request.args.get('sepal_width')
18     petal_length = request.args.get('petal_length')
19     petal_width = request.args.get('petal_width')
20
21     df = pd.DataFrame({'sepal.length':[sepal_length], 'sepal.width':[sepal_width], 'petal.length':[petal_length], 'petal.width':[petal_width]})
22     pred_type = model.predict(df)
23     output = pred_type[0]
24     if output == 1 :
25         o1 = 'Setosa'
26     elif output == 2:
27         o1 = 'Versicolor'
28     else:
29         o1 = 'Virginica'
30
31     return jsonify({'Variety': str(o1)})
32
33 if __name__ == '__main__':
34     app1.run(debug = True)
35
```

### 3) Download and install postman and create an account in it



### 4) Run the app1.py script to get the local server up

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\Data Glacier Virtual Internship\Heroku Deployment> & "C:/Users/Anju Paul/AppData/Local/Programs/Python/Python310/python.exe" "D:/Data Glacier Virtual Internship/Heroku Deployment/templates/app1.py"
* Serving Flask app 'app1' (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: on
* Restarting with watchdog (windowsapi)
* Debugger is active!
* Debugger PIN: 349-272-658
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

### 5) In the postman terminal send a new request by using the above local url from step 4



http://127.0.0.1:5000/ Save Send

GET http://127.0.0.1:5000/ Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (4) Test Results Status: 200 OK Time: 252 ms Size: 200 B Save Response

Pretty Raw Preview Visualize JSON Send

```
1 {  
2   "data": "Welcome to Flower Type Prediction App"  
3 }
```

6) Use the /predict/ and provide the input variables and send the request.

Overview GET http://127.0.0.1:5000/predict/ GET http://127.0.0.1:5000/ No Environment

http://127.0.0.1:5000/predict/?sepal\_length=4&sepal\_width=2.3&petal\_length=4.1&petal\_width=2 Save Send

GET http://127.0.0.1:5000/predict/?sepal\_length=4&sepal\_width=2.3&petal\_length=4.1&petal\_width=2 Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> sepal_length	4			
<input checked="" type="checkbox"/> sepal_width	2.3			
<input checked="" type="checkbox"/> petal_length	4.1			
<input checked="" type="checkbox"/> petal_width	2			
Key	Value	Description		

Body Cookies Headers (4) Test Results Status: 200 OK Time: 2.04 s Size: 176 B Save Response

Pretty Raw Preview Visualize JSON Send

```
1 {  
2   "Variety": "Versicolor"  
3 }
```