

## Week 3: Deployment on Flask

Name: Suhas Yogeshwara

Batch Code: LISUM11: 30

Submission Date: 05/08/2022

Submitted to: Data Glaciers

Step 1: Collect a Dataset (simple Data) i.e., Kaggle. In my Case, I have Collected Salary Dataset for predicting the Salary based on Number of Years of Experience (Linear Regression Model).

```
In [1]: import pandas as pd
```

```
In [2]: ds=pd.read_csv('Salary_Data.csv')
ds
```

Out[2]:

	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0
5	2.9	56642.0
6	3.0	60150.0
7	3.2	54445.0
8	3.2	64445.0
9	3.7	57189.0
10	3.9	63218.0
11	4.0	55794.0
12	4.0	56957.0
13	4.1	57081.0
14	4.5	61111.0
15	4.9	67938.0
16	5.1	66029.0

Step 2: Import the Data Using Suitable Libraries for the analysis (preferably pandas, in this case is Flask, Requests and Numpy).

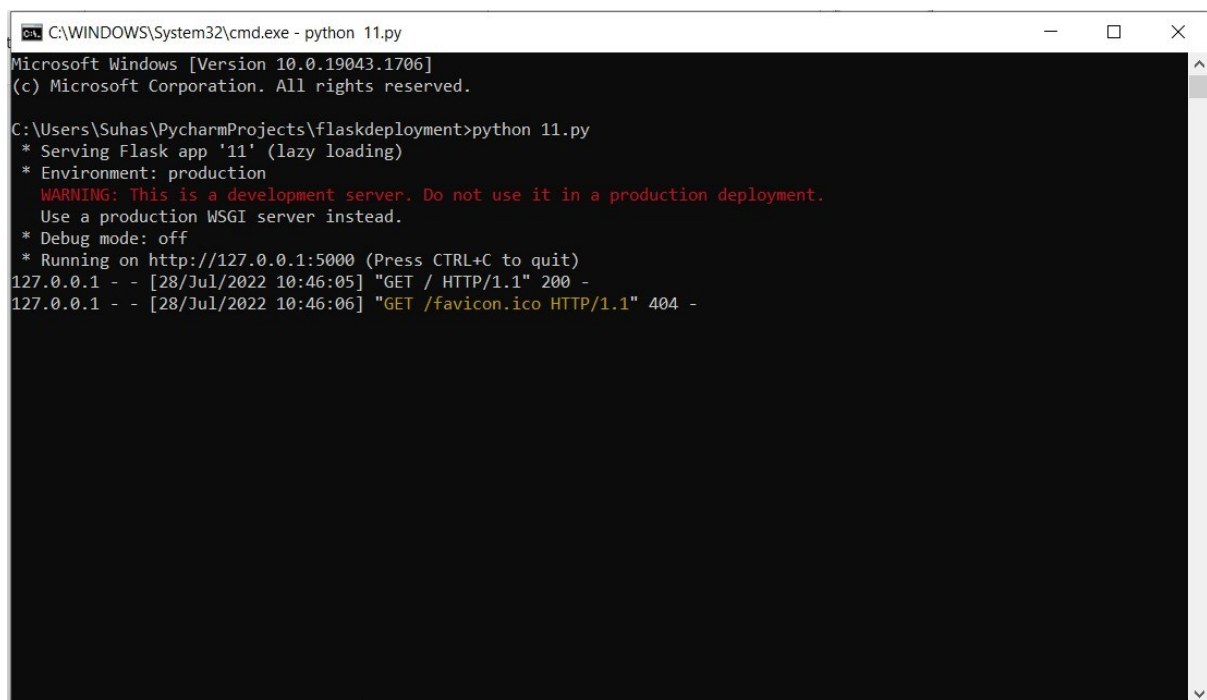
```
1 import numpy as np
2 from flask import Flask, request, render_template
3 import pandas as pd
```

Step 3: Write a basic Code to see if the Flask and the required Libraires are Installed Properly.

1. Write the Code

```
1  from flask import Flask
2
3  app=Flask(__name__)
4
5  @app.route('/')
6  def home():
7      return 'Home World!'
8
9  app.run(port=5000)
```

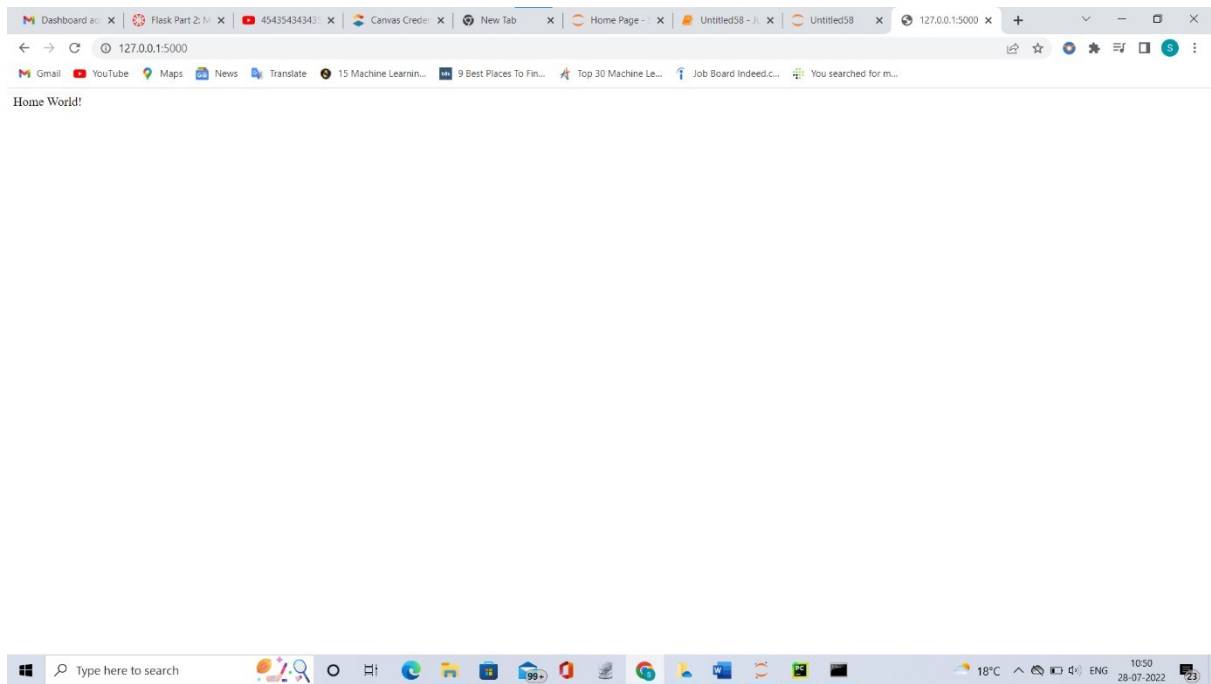
2.Run the Code Using Command Prompt or Terminal.



```
C:\WINDOWS\System32\cmd.exe - python 11.py
Microsoft Windows [Version 10.0.19043.1706]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Suhas\PycharmProjects\flaskdeployment>python 11.py
* Serving Flask app '11' (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 - - [28/Jul/2022 10:46:05] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [28/Jul/2022 10:46:06] "GET /favicon.ico HTTP/1.1" 404 -
```

3.Output



Step 4: Well, the Code is Correct and seems to be working Fine. Now write Code to Load the Data and the required Libraries should be installed.

```
1 import numpy as np
2 from flask import Flask, request, render_template
3 import pandas as pd
4
5 app = Flask(__name__)
6 model = pd.read_csv('C:/Users/Suhas/Downloads/Salary_Data.csv')
7
8
9 @app.route('/')
10 def home():
11     return render_template('insert.html')
12
13
14 @app.route('/predict', methods=['post'])
15 def predict():
16     '''
17     Predict Salary
18     '''
19
20     int_features = [int(x) for x in request.form.values()]
21     final_features = [np.array(int_features)]
22     prediction = model.predict(final_features)
23     output = round(prediction[0], 2)
24     return render_template('index.html', prediction_text='The average salary is {}'.format(output))
25
26
27 if __name__ == '__main__':
28     app.run(port=5000, debug=True)
29
```

Step 5: Write a HTML Code or the pre-written Code given in Instruction Videos (which has been Modified based on the Requirements)

```

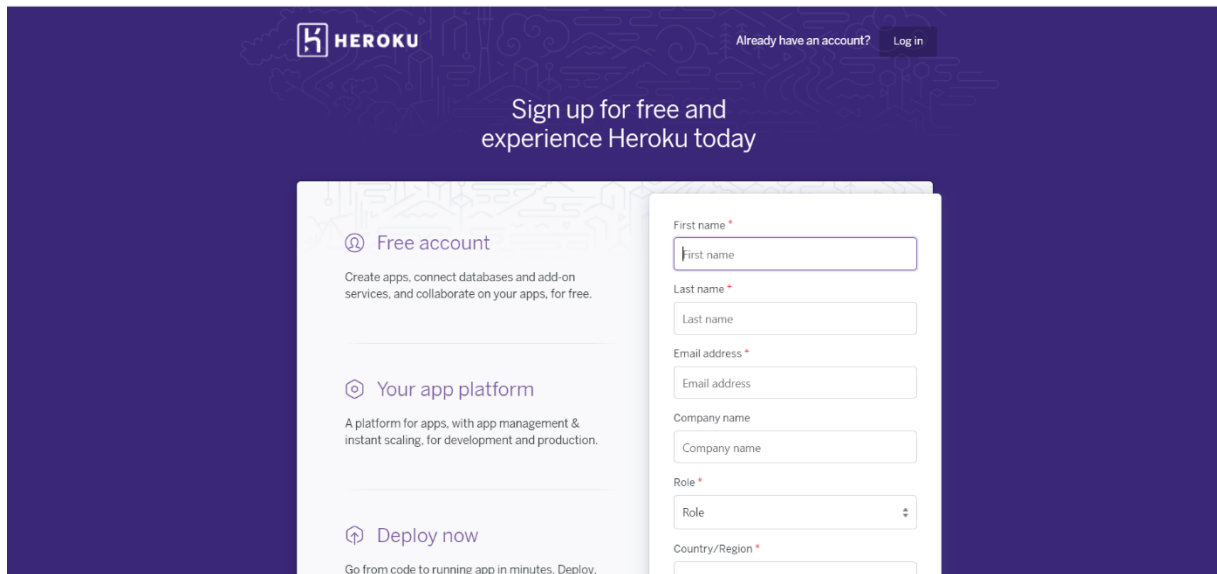
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>ML API</title>
6  </head>
7  <body>
8
9      <h1>Predict Salary</h1>
10
11     <form action="{{url_for('predict')}}" method="post">
12         <input type="text" name="Number of years of experience" required="required" />
13         <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
14     </form>
15
16     <br>
17     <div>
18         {{Average Salary based on years of experience $}}
19     </div>
20 </body>
21 </html>

```

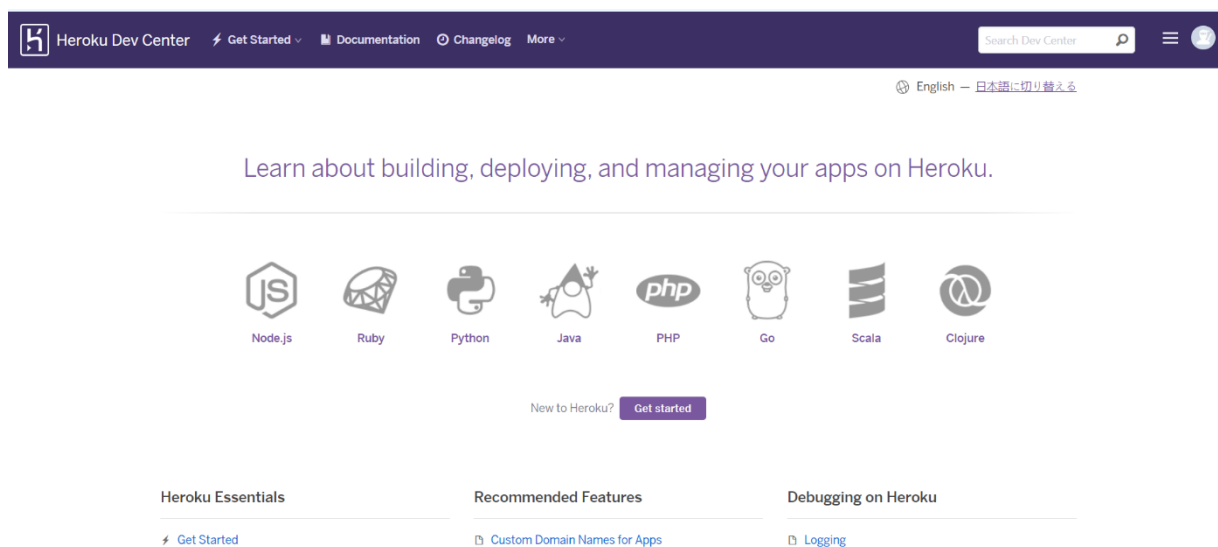
Step 6: Note (No Stylers were used here as the requirements were simple), Now run the Python Code to see the output.

The screenshot shows a code editor with two panes. The left pane contains the HTML code from the previous block. The right pane shows the rendered output of the HTML. The rendered output is a web page with the title "Predict Salary". It features a text input field with the placeholder text "Number of years of experience" and a "Predict" button. Below the form, it displays the Jinja2 template output: "{{Average Salary based on years of experience \$}}".

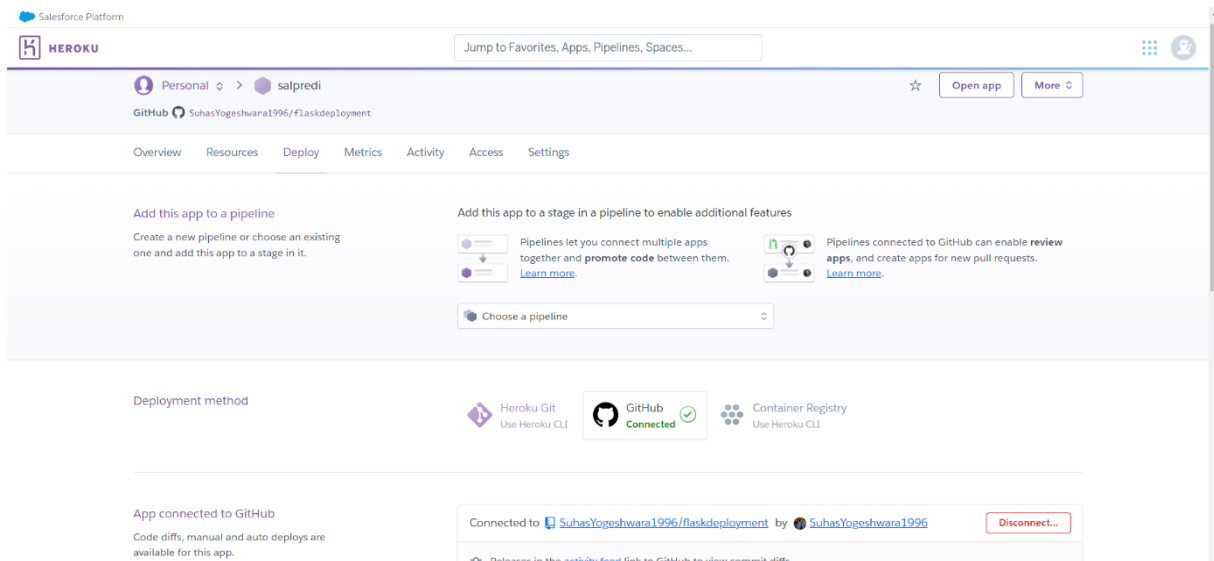
Step 7: Deployment of the model in Heroku Cloud Platform. By creating an account in Heroku.

The image shows the Heroku sign-up page. At the top, there's a Heroku logo and a link to log in for existing users. The main heading says "Sign up for free and experience Heroku today". Below this, there are three sections: "Free account" (explaining that users can create apps, connect databases, and collaborate for free), "Your app platform" (describing it as a platform for app management and scaling), and "Deploy now" (encouraging users to go from code to running an app in minutes). On the right side, there is a sign-up form with fields for First name, Last name, Email address, Company name, Role (a dropdown menu), and Country/Region (a dropdown menu).

Step 8: Supports Various Programming Languages.



Step 9 : either Connect GitHub to Heroku (the required repository for the deployment) or by manual means of installation i.e., Command Prompt.



Step 10: once the required dependencies are installed in the automatic deploy or manual deploy the app is deployed in cloud.

```
Anaconda Prompt (anakonda) - heroku login
(base) C:\Users\Suhas>
(base) C:\Users\Suhas>heroku login
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/d01559c1-25ce-4e83-a897-9b3ff5227212?requestor=SFMyNTY.g
2gDbQAAAA00Ni4xMTQuMzkuMTU1bgYAWOFIaIIBYgABUYA.s2wklcgKyjdEu980GblGM-46gjusJYeN5v912NhHYqM
Logging in... done
Logged in as suhas.gys1996@gmail.com
(base) C:\Users\Suhas>
(base) C:\Users\Suhas>
```

## 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

master

☐ 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

## 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

master

Deploy Branch