



**Data Glacier**

Your Deep Learning Partner

# Week 5: Cloud and API deployment

**LISUM02**

**11<sup>th</sup> August 2021**

**By,  
Joseph Antony**

# Dataset

- Same dataset was used as the one used for deploying using Web-App last week.
- Serialization was done using python pickling.
- Like last time, created an application using Flask API.
- Ran the app using windows command prompt.

```
app.py X
1 import pandas as pd
2 import pickle
3 from flask import Flask, request, jsonify
4
5 app = Flask(__name__)
6
7
8
9 @app.route('/predict/', methods=['GET', 'POST'])
10 def predict():
11
12     model = pickle.load(open('lr_model.pkl', 'rb'))
13     age = request.args.get('Age')
14     salary = request.args.get('EstimatedSalary')
15
16     df = pd.DataFrame({'Age': [age], 'EstimatedSalary': [salary]})
17
18     prediction = model.predict(df)
19
20     if prediction == 0:
21         output = "Customer will not purchase ="
22     else:
23         output = "Customer will Purchase ="
24
25     return jsonify({"Prediction": output})
26
27
28 if __name__ == "__main__":
29     app.run(debug=True)
```

```
warnings.warn(
27.0.0.1 - - [11/Aug/2021 23:04:24] "GET /predict/?Age=25&EstimatedSalary=100000" 200 -
* Detected change in 'C:\Users\joean\Documents\GitHub\Data-Glacier\Week5_Cloud_API\app.py', reloading
* Restarting with stat
* Debugger is active!
* Debugger PIN: 132-629-808
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

C:\Users\joean\Documents\GitHub\Data-Glacier\Week5_Cloud_API>py app.py
* Serving Flask app 'app' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production
  environment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 132-629-808
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

# Model Deployment on Postman API

GET localhost:5000/predict/?Age=50&EstimatedSalary=150000

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

Query Params

| KEY             | VALUE  | DESCRIPTION |
|-----------------|--------|-------------|
| Age             | 50     |             |
| EstimatedSalary | 150000 |             |
| Key             | Value  | Description |

Body Cookies Headers (4) Test Results Status: 200 OK Time: 531 ms Size: 193 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "Prediction": "Customer will Purchase =)"
3 }
```

GET localhost:5000/predict/?Age=25&EstimatedSalary=50000

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

Query Params

| KEY             | VALUE | DESCRIPTION |
|-----------------|-------|-------------|
| Age             | 25    |             |
| EstimatedSalary | 50000 |             |
| Key             | Value | Description |

Body Cookies Headers (4) Test Results Status: 200 OK Time: 1119 ms Size: 197 B Save Response

Pretty Raw Preview Visualize JSON

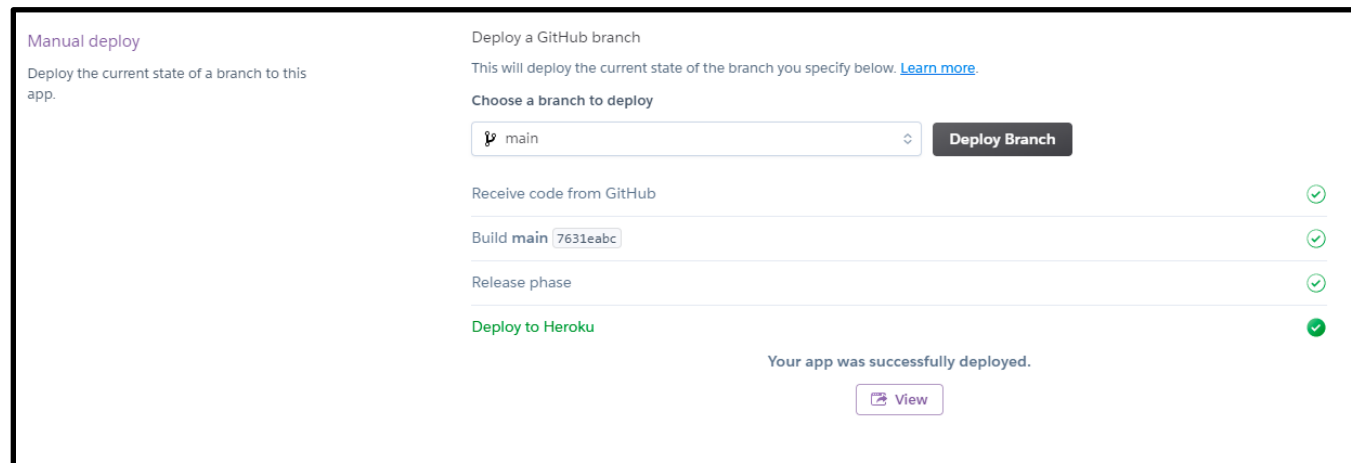
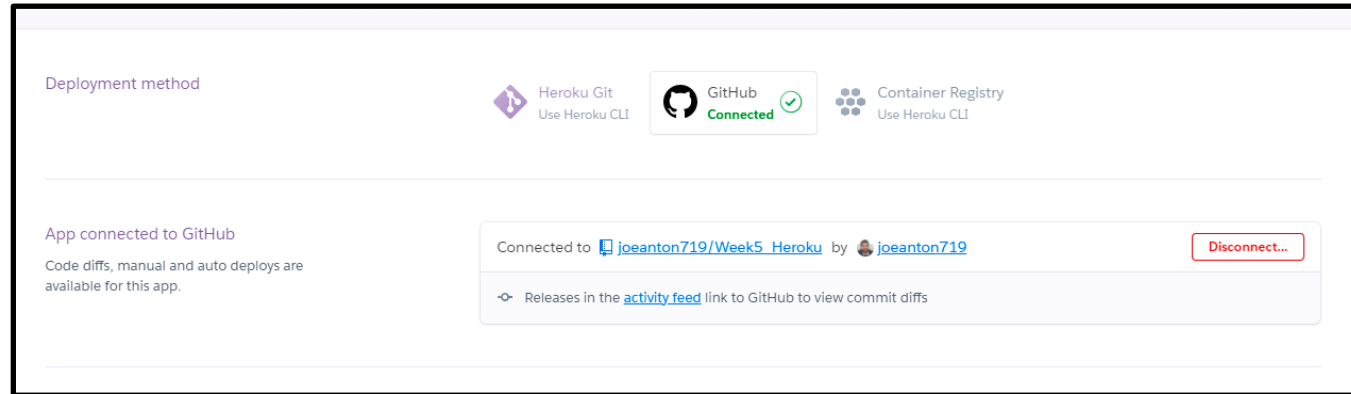
```
1 {
2   "Prediction": "Customer will not purchase =(
3 }
```

➤ Deployed the model on to Postman API. The model worked perfectly.

# Model Deployment on Heroku App

➤ Next, Also deployed the model on to Heroku Platform.

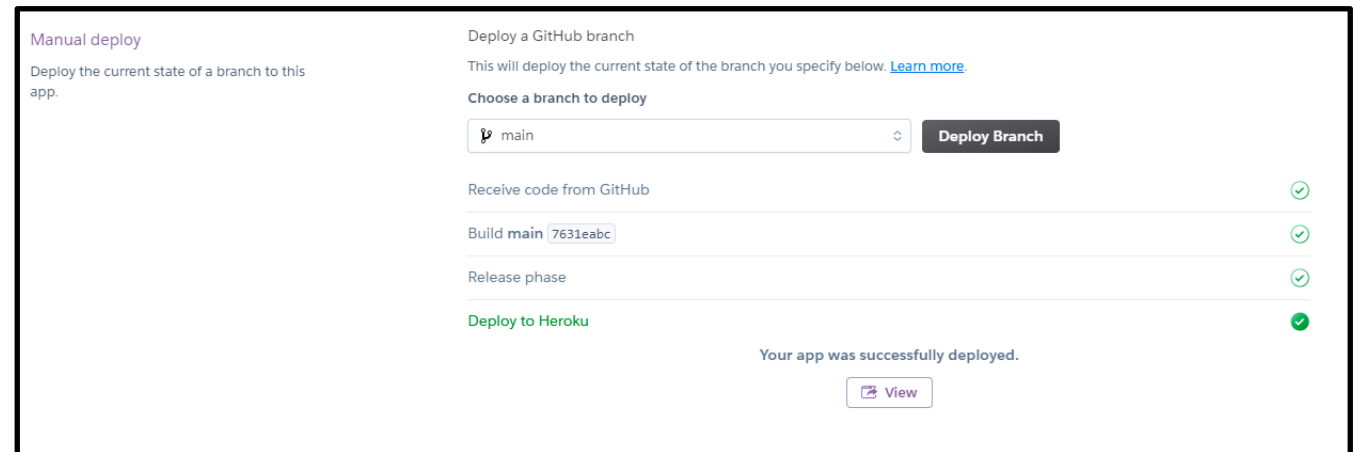
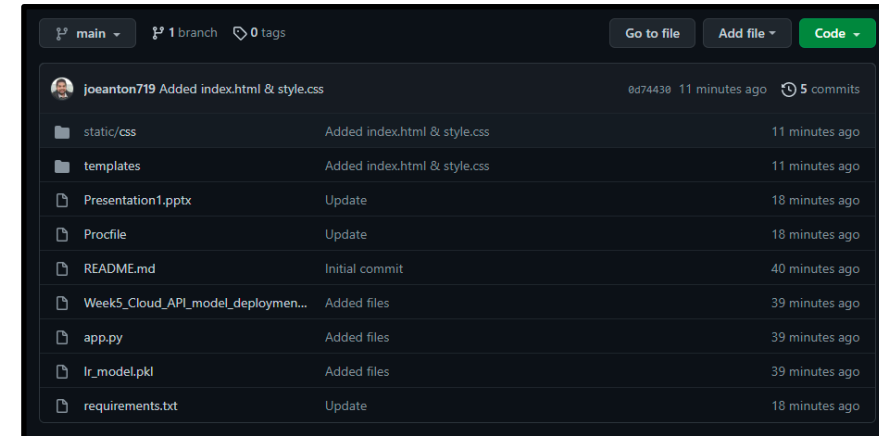
➤ Connected GitHub repo with necessary files to the Heroku App.



# Model Deployment

➤ The repo contains all necessary files including requirements.txt (which includes all necessary packages for deployment and Procfile)

➤ The model was successfully deployed on Heroku Application.



# Model Deployment

This web-app will predict whether a customer will purchase a product based on the customer's age and salary.

### Enter Customer Age & Salary

Predict

### Enter Customer Age & Salary

Predict

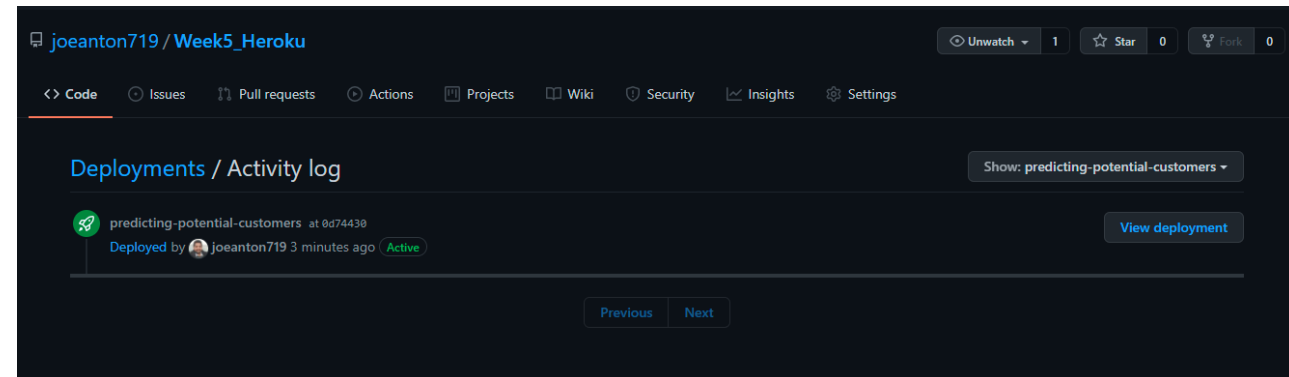
Customer will not purchase

### Enter Customer Age & Salary

Predict

Customer will Purchase

Model deployed and ran successfully on the Heroku App.



# The End