Week 5: Cloud and API deployment

LISUM02
11th August 2021

By,
Joseph Antony

Submitted to: https://github.com/joeanton719/Week5_Heroku

Dataset

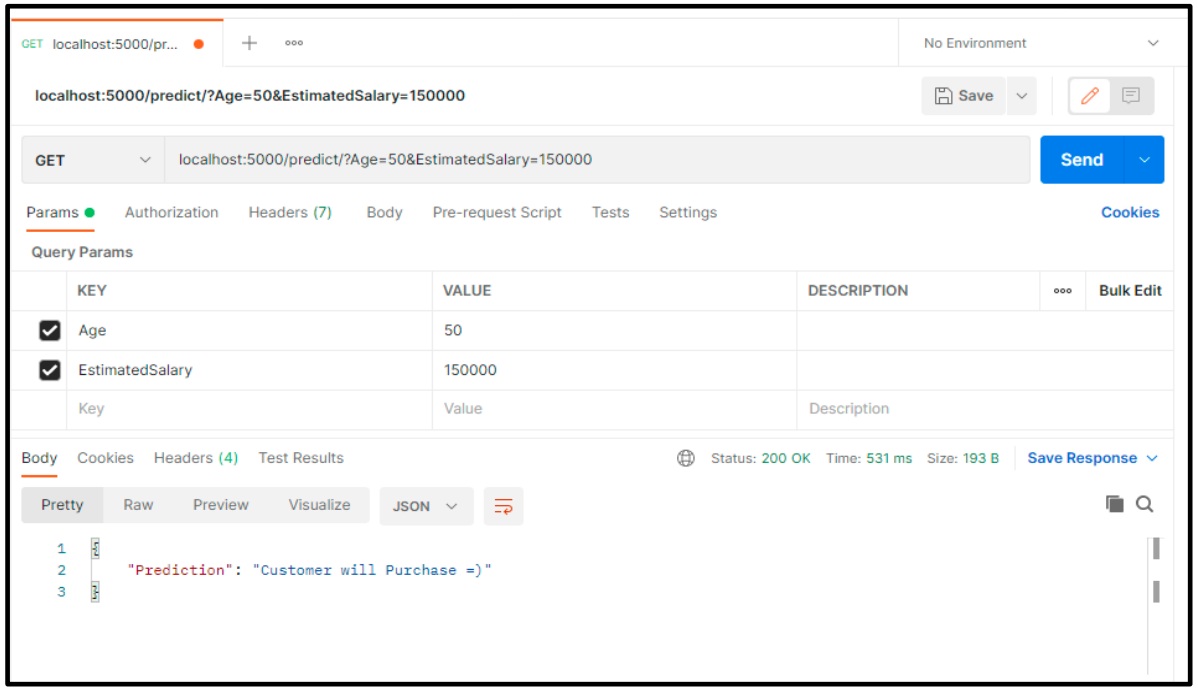
- Same dataset was used as the one used for deploying using Web-App last week.
- Serialization was done using python pickling.
- Like last time, created an application using Flask API.
- Ran the app using windows command prompt.

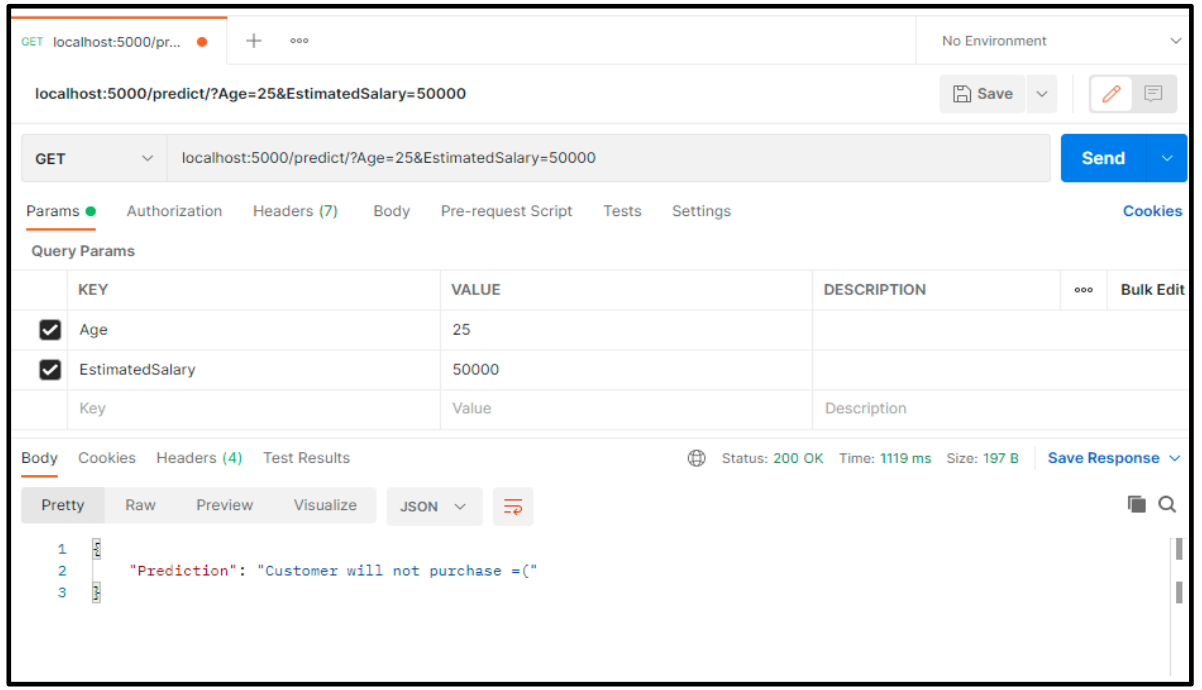
app.py

```python
import pandas as pd
import pickle
from flask import Flask, request, jsonify

app = Flask(__name__)



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

    model = pickle.load(open('lr_model.pkl', 'rb'))
    age = request.args.get('Age')
    salary = request.args.get('EstimatedSalary')

    df = pd.DataFrame({'Age' : [age], 'EstimatedSalary' : [salary]})

    predicition = model.predict(df)

    if predicition == 0:
        output = "Customer will not purchase =("
    else:
        output = "Customer will Purchase =)"

    return jsonify({"Prediction" : output})


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

```
 warnings.warn(
27.0.0.1 - - [11/Aug/2021 23:04:24] "GET /predict/?Age=25&Estimate
?/1.1" 200 -
* Detected change in 'C:\\Users\\joean\\Documents\\GitHub\\Data-Gl
d_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)

:\Users\joean\Documents\GitHub\Data-Glacier\Week5_Cloud_API>py app
* Serving Flask app 'app' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a produc

  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)
```

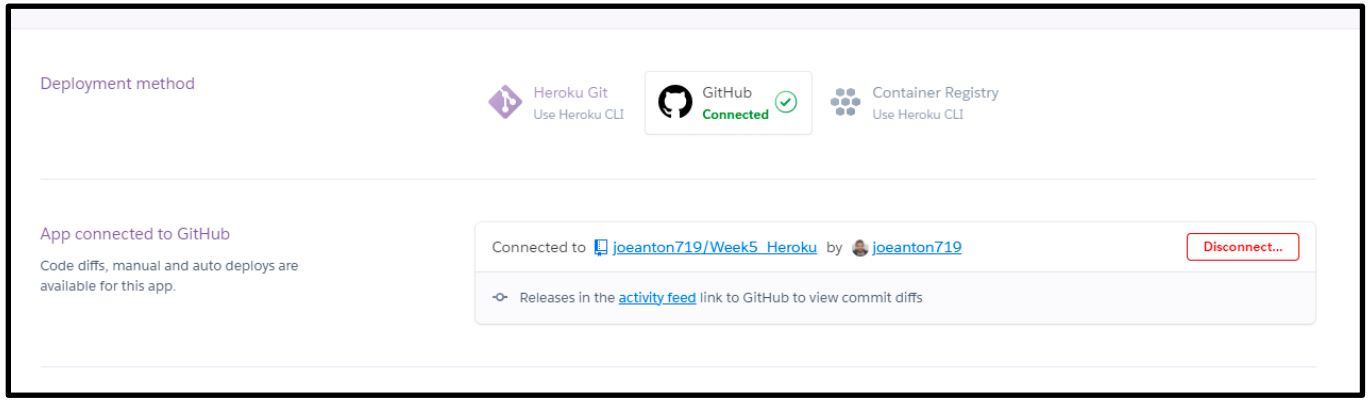
Model Deployment on Postman API



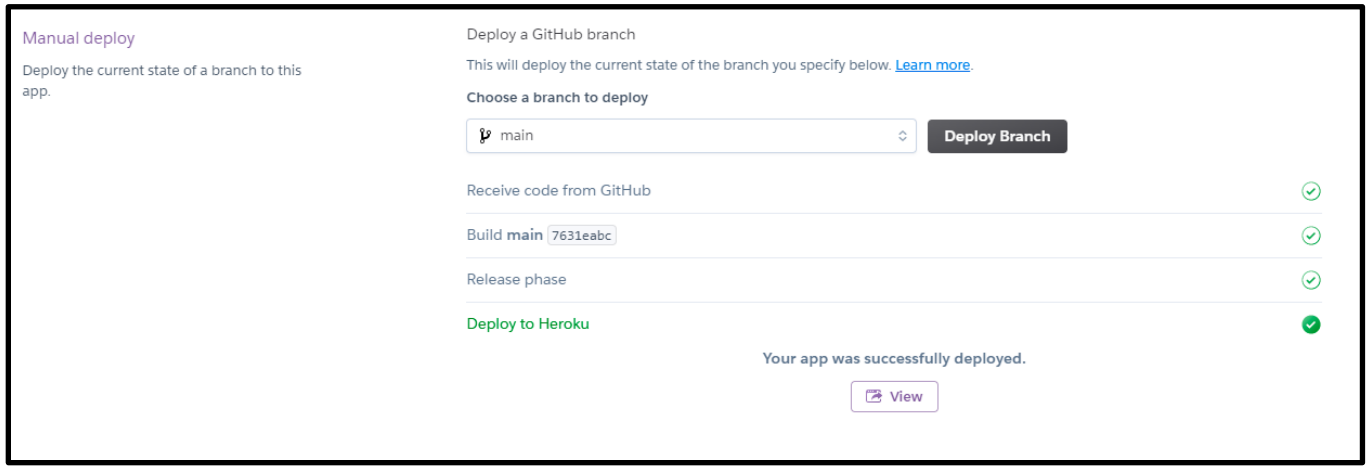


- Deployed the model on to Postman API. The model worked perfectly.

Model Deployment on Heroku App



HEROKU



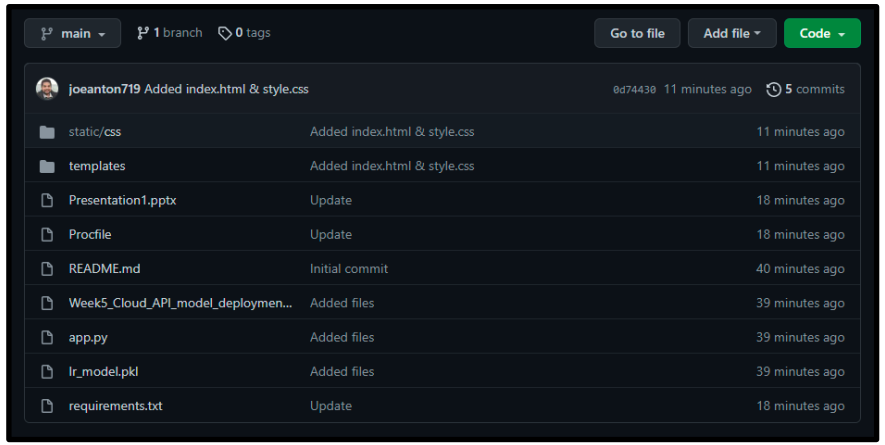
- Next, Also deployed the model on to Heroku Platform.
- Connected GitHub repo with necessary files to the Heroku App.

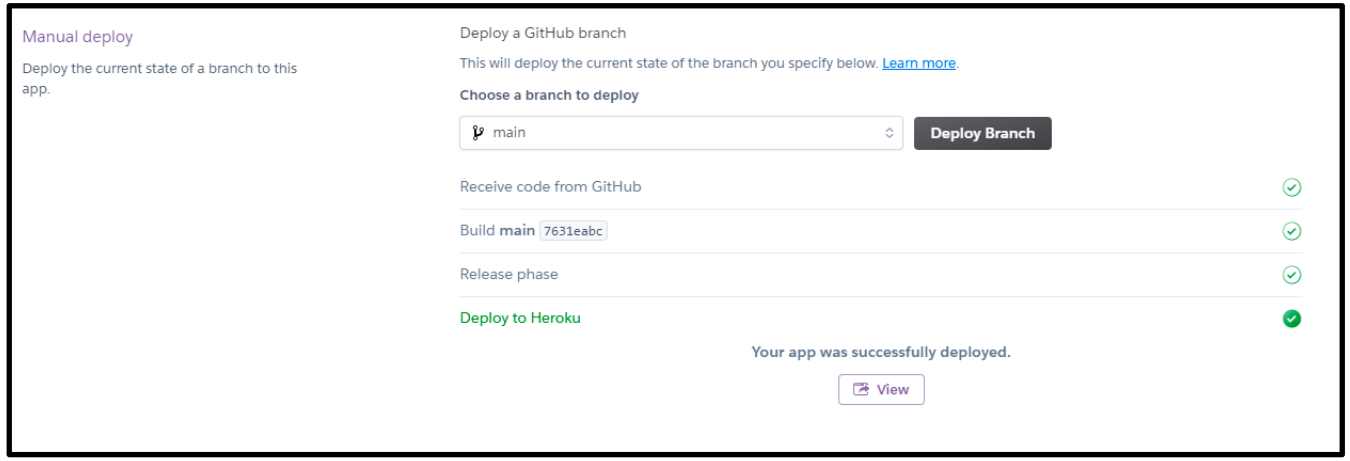
Model Deployment

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

- The model was successfully deployed on Heroku Application.

HEROKU

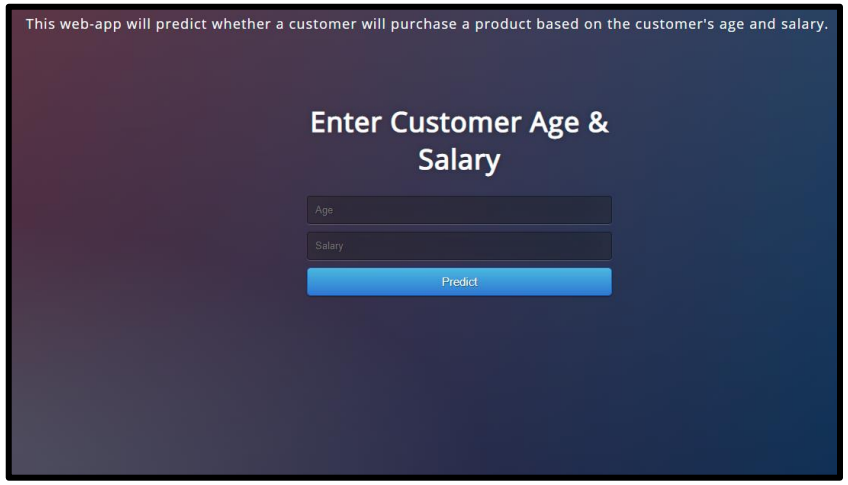


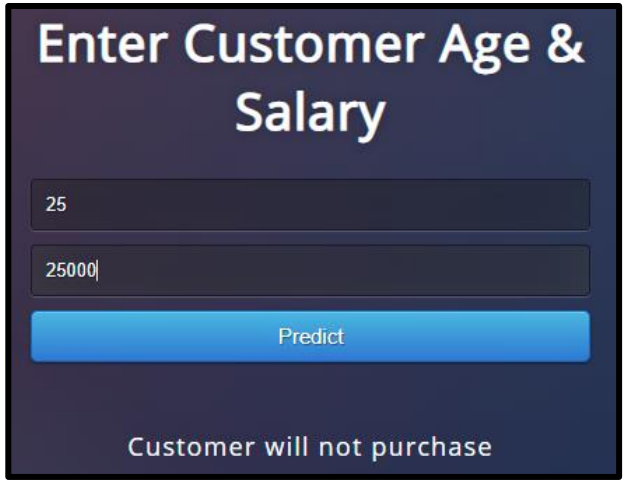


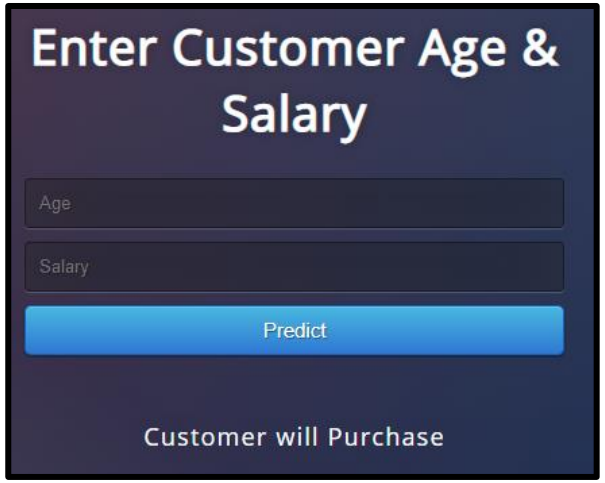
Data Glacier

Model Deployment

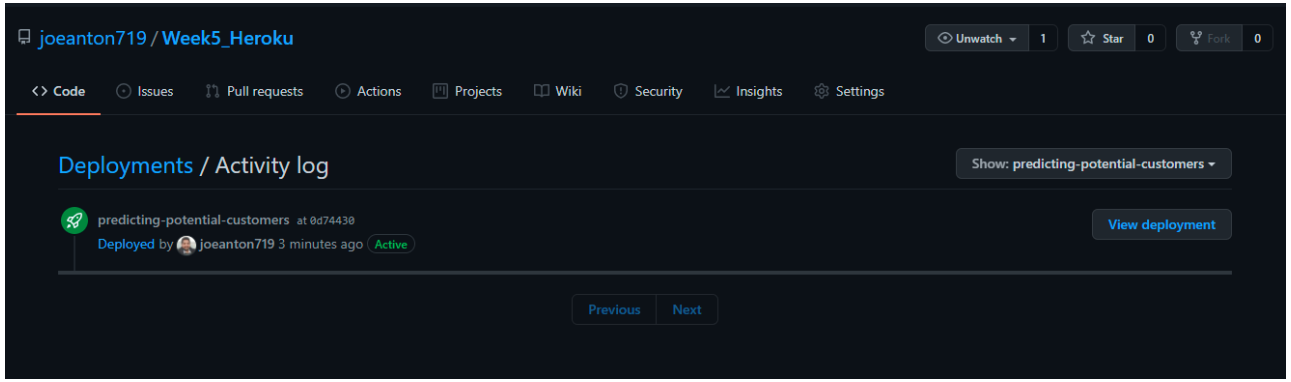
HEROKU







Model deployed and ran successfully on the Heroku App.



The End