

# MODEL DEPLOYMENT ON HEROKU

Name: Mohamed Derbeli  
Batch Code: LISUM12  
Date of submission: 06/09/2022  
Submitted to: Data Glacier

## Model Deployment steps

### Step1: Selecting the model

I have used the data and model that used in the deployment on Flask (Assignment of week 4)

### Step2: Build the HTML for the graphic interface

The [index.html](#) is a file that contains the structure of the web app design and [AppStyle.css](#) is used to beautify the web design.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Insurance Web Application</title>
6   <meta charset="utf-8">
7   <meta name="author" content="AnikeSadia">
8   <meta name="viewport" content="width=device-width, initial-scale=1.0,maximum-scale=1,user-scalable=no">
9   <link rel="stylesheet" href="static/AppStyle.css">
10 </head>
11 <body>
12   <div class="model_desc">
13     <h2> Model Description </h2><br>
14     <p>This web application is designed to help users of Pink and Yellow Cab companies by providing information regarding the prices.
15       The aim of this project is to predict the price charged by each company based on certain information. The application requires
16       several input data that should be entered by the user. Finally, the predict bottom will generate the price charged based on the user data.<br><br>
17     Please enter your data:<br>
18     <span class="bold yellow">Age:   </span> 1 --> 70 <br>
19     <span class="bold yellow">Gender: </span> female=0, male=1 <br>
20     <span class="bold yellow">Month:  </span> 1 --> 12 <br>
21     <span class="bold yellow">KM-Travelled: </span> 1 --> 50<br>
22     <span class="bold yellow">Company: </span> pink=0, yellow=1 <br>
23     <span class="bold yellow">City:    </span> atlanta =0, austin=1, boston=2, chicago=3, dallas=4, denver=5, los angeles=6, miami=7, nashville=8, new york=9,
24       phoenix=11, pittsburgh=12, sacramento=13, san diego=14, seattle=15, silicon valley=16, tucson=17, washington=18<br>
25
```

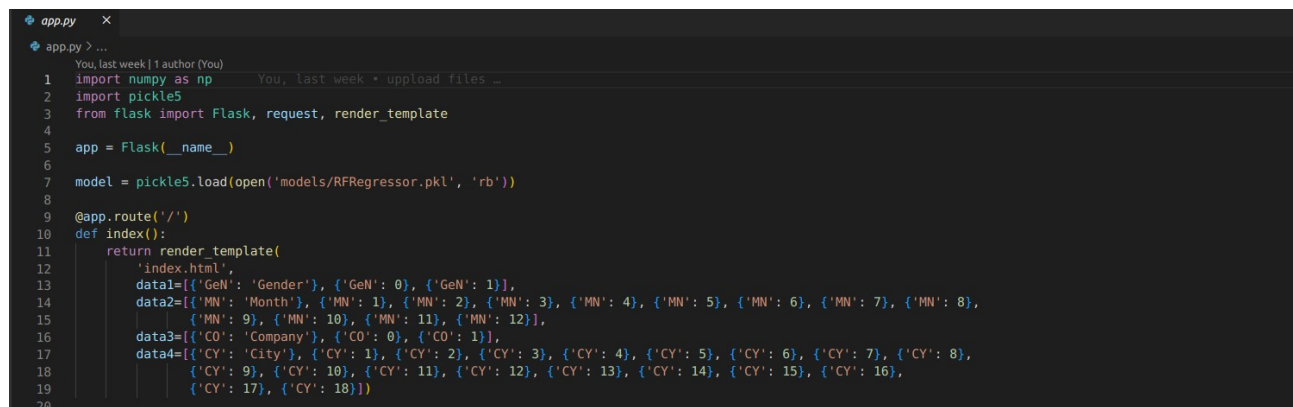
```

27     </p></div>
28 <div class="data_part">
29     <h1>PRICE CHARGED PREDICTION</h1>
30     <form action="{{url_for('predict')}}" method="post">
31         <input class="Input" type="text" name="A1" placeholder="Age" required/>
32         <input class="Input" type="text" name="A2" placeholder="KM-Travelled" required/>
33         <select name="comp_select2" class="Input">
34             {% for o in data3 %}
35                 <option value="{{ o.CO }}">{{ o.CO }}</option>
36             {% endfor %}
37         </select><br><br>
38         <select name="comp_select1" class="Input">
39             {% for o in data2 %}
40                 <option value="{{ o.MN }}">{{ o.MN }}</option>
41             {% endfor %}
42         </select><br><br>
43
44         <select name="comp_select" class="Input">
45             {% for o in data1 %}
46                 <option value="{{ o.GeN }}">{{ o.GeN }}</option>
47             {% endfor %}
48         </select><br><br>
49
50         <select name="comp_select3" class="Input">
51             {% for o in data4 %}
52                 <option value="{{ o.CY }}">{{ o.CY }}</option>
53             {% endfor %}
54         </select><br><br>
55
56
57         <button type="submit" class="pred-btn"> Predict </button>
58
59     </form>
60     <h3> {{prediction_text}} </h3>
61
62 </div>
63
64

```

## Step3: Build the Flask app

[App.py](#) is built, a flask app that used the deserialized model to accept new data and predict a the price charged.



```

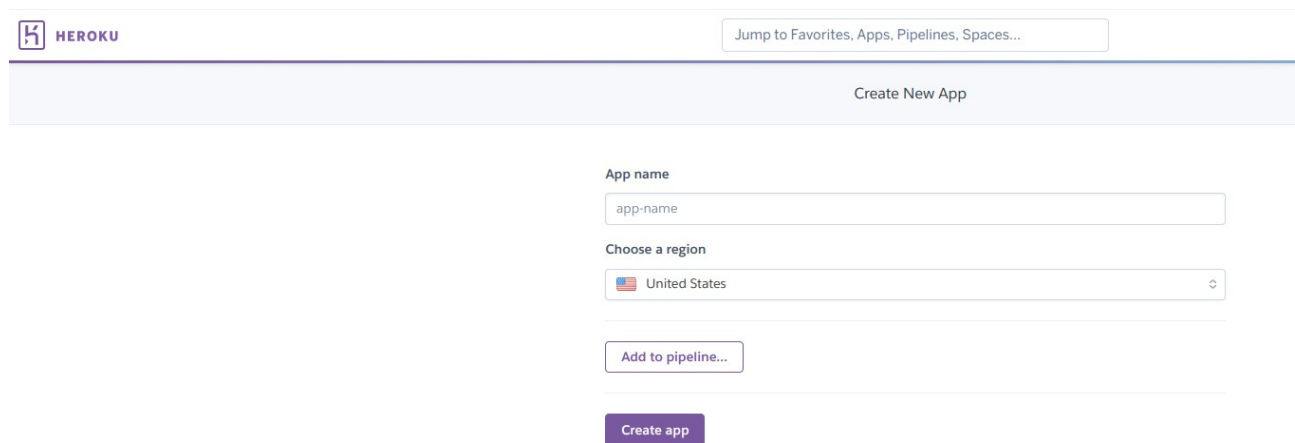
app.py
app.py > ...
You, last week | 1 author (You)
1 import numpy as np
2 import pickle5
3 from flask import Flask, request, render_template
4
5 app = Flask(__name__)
6
7 model = pickle5.load(open('models/RFRegressor.pkl', 'rb'))
8
9 @app.route('/')
10 def index():
11     return render_template(
12         'index.html',
13         data1=[{'GeN': 'Gender'}, {'GeN': 0}, {'GeN': 1}],
14         data2=[{'MN': 'Month'}, {'MN': 1}, {'MN': 2}, {'MN': 3}, {'MN': 4}, {'MN': 5}, {'MN': 6}, {'MN': 7}, {'MN': 8},
15                 {'MN': 9}, {'MN': 10}, {'MN': 11}, {'MN': 12}],
16         data3=[{'CO': 'Company'}, {'CO': 0}, {'CO': 1}],
17         data4=[{'CY': 'City'}, {'CY': 1}, {'CY': 2}, {'CY': 3}, {'CY': 4}, {'CY': 5}, {'CY': 6}, {'CY': 7}, {'CY': 8},
18                 {'CY': 9}, {'CY': 10}, {'CY': 11}, {'CY': 12}, {'CY': 13}, {'CY': 14}, {'CY': 15}, {'CY': 16},
19                 {'CY': 17}, {'CY': 18}])
20

```

Now the function below accepts the data and return the predicted percentage

```
21
22 @app.route("/predict", methods=['GET', 'POST'])
23 def predict():
24     input_data = list(request.form.values())
25     if int(input_data[0]) & int(input_data[1]) & input_data[2].isdigit() & input_data[3].isdigit() & input_data[4].isdigit() & input_data[4].isdigit() == True:
26         pass
27     else:
28         print(ValueError)
29
30     input_values = [x for x in input_data]
31     arr_val = [np.array(input_values)]
32     prediction = model.predict(arr_val)
33     output = round(prediction[0], 2)
34     return render_template('index.html', prediction_text=" The price of the transaction will be around: {}".format(output),
35                             data1=[{'GeN': 'Gender'}, {'GeN': 0}, {'GeN': 1}],
36                             data2=[{'MN': 'Month'}, {'MN': 1}, {'MN': 2}, {'MN': 3}, {'MN': 4}, {'MN': 5}, {'MN': 6}, {'MN': 7}, {'MN': 8}, {'MN': 9}, {'MN': 10}],
37                             data3=[{'CO': 'Company'}, {'CO': 0}, {'CO': 1}],
38                             data4=[{'CY': 'City'}, {'CY': 1}, {'CY': 2}, {'CY': 3}, {'CY': 4}, {'CY': 5}, {'CY': 6}, {'CY': 7}, {'CY': 8}, {'CY': 9}, {'CY': 10}, {
39
40
41 if __name__ == '__main__':
42     app.run(debug=True)
43
```

Then, create a new app in Heroku and push the repository to Heroku using the terminal. If everything goes well, a link will generated.



The screenshot shows the Heroku dashboard with the 'Create New App' button highlighted. Below the button, the 'App name' field is filled with 'app-name'. The 'Choose a region' dropdown menu is set to 'United States'. There is an 'Add to pipeline...' button and a 'Create app' button at the bottom.

Inside the folder of your project, open terminal and implement the following code:

```
git init
```

```
git add .
```

```
git commit -m "first commit"
```

```
heroku git:remote -a name_of_app
```

```
git push heroku master
```

Finally, clicking on the link will direct you to the flask web application interface shown below.

### Model Description

This web application is designed to help users of Pink and Yellow Cab companies by providing information regarding the prices. The aim of this project is to predict the price charged by each company based on certain information. The application requires several input data that should be entered by the user. Finally, the predict button will generate the price charged based on the user data.

Please enter your data:

**Age:** 1 --> 70  
**Gender:** female=0, male=1  
**Month:** 1 --> 12  
**KM-Travelled:** 1 --> 50  
**Company:** pink=0, yellow=1  
**City:** atlanta =0, austin=1, boston=2, chicago=3, dallas=4, denver=5, los angeles=6, miami=7, nashville=8, new york=9, orange county=10, phoenix=11, pittsburgh=12, sacramento=13, san diego=14, seattle=15, silicon valley=16, tucson=17, washington=18

## PRICE CHARGED PREDICTION

Age

KM-Travelled

Company

▼

Month

▼

Gender

▼

City

▼

Predict