# Regression ML Model Deployment On Heroku

Name: Mohammad Aminian

- Batch Code: LISUM04
- Submission Date: 20/10/21
- Submitted to: Data Glacier

# 1-Data

In [3]:
```
import pandas as pd
df=pd.read_csv("/media/elliot/HDD/my_tipdata.csv")
df
```

Out[3]:

| | total_bill | sex | smoker | day | time | size | tip |
|---|---|---|---|---|---|---|---|
| **0** | 12.16 | 1.0 | Yes | Friday | Lunch | 2 | 2.20 |
| **1** | 21.50 | 1.0 | No | Sunday | Dinner | 4 | 3.50 |
| **2** | 10.33 | 0.0 | No | Thursday | Lunch | 2 | 2.00 |
| **3** | 14.78 | 1.0 | No | Sunday | Dinner | 2 | 3.23 |
| **4** | 18.04 | 1.0 | No | Sunday | Dinner | 2 | 3.00 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **194** | 18.28 | 1.0 | No | Thursday | Lunch | 2 | 4.00 |
| **195** | 17.29 | 1.0 | No | Thursday | Lunch | 2 | 2.71 |
| **196** | 18.43 | 1.0 | No | Sunday | Dinner | 4 | 3.00 |
| **197** | 18.78 | 0.0 | No | Thursday | Dinner | 2 | 3.00 |
| **198** | 15.98 | 0.0 | No | Friday | Lunch | 3 | 3.00 |

199 rows × 7 columns

# 2-Model

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import pandas as pd
4 import pickle
5
6 df = pd.read_csv('my_tipdata.csv')
7 pd.DataFrame(df,columns=['total_bill','sex','smoker','day','time','size','tip'])
8 df.sex = df.sex.astype('category')
9
10 X = df[['total_bill','sex','size']]
11 Y = df['tip']
12
13 from sklearn.linear_model import LinearRegression
14 regressor = LinearRegression()
15 regressor.fit(X, Y)
16 pickle.dump(regressor, open('model.pkl','wb'))
17 model = pickle.load(open('model.pkl','rb'))
18
19
20
```

# 3-HTML

```
<!DOCTYPE html>
<html >
<head>
  <meta charset="UTF-8">
  <title>Deployment Tutorial with Flask</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
<link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">

</head>

<body style="background: #000;">
 <div class="login">
        <h1>Tip Forecasting</h1>

    <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('predict')}}"method="post">
       <input type="text" name="Total Bill" placeholder="Total bill" required="required" />
       <input type="text" name="Gender" placeholder="0 - Female 1 - Male" required="required" />
              <input type="text" name="Size" placeholder="Size of people" required="required" />
       <button type="submit" class="btn btn-primary btn-block btn-large">Predict tip </button>
    </form>

   <br>
   <br>
   {{ prediction_text }}

 </div>
</body>
```

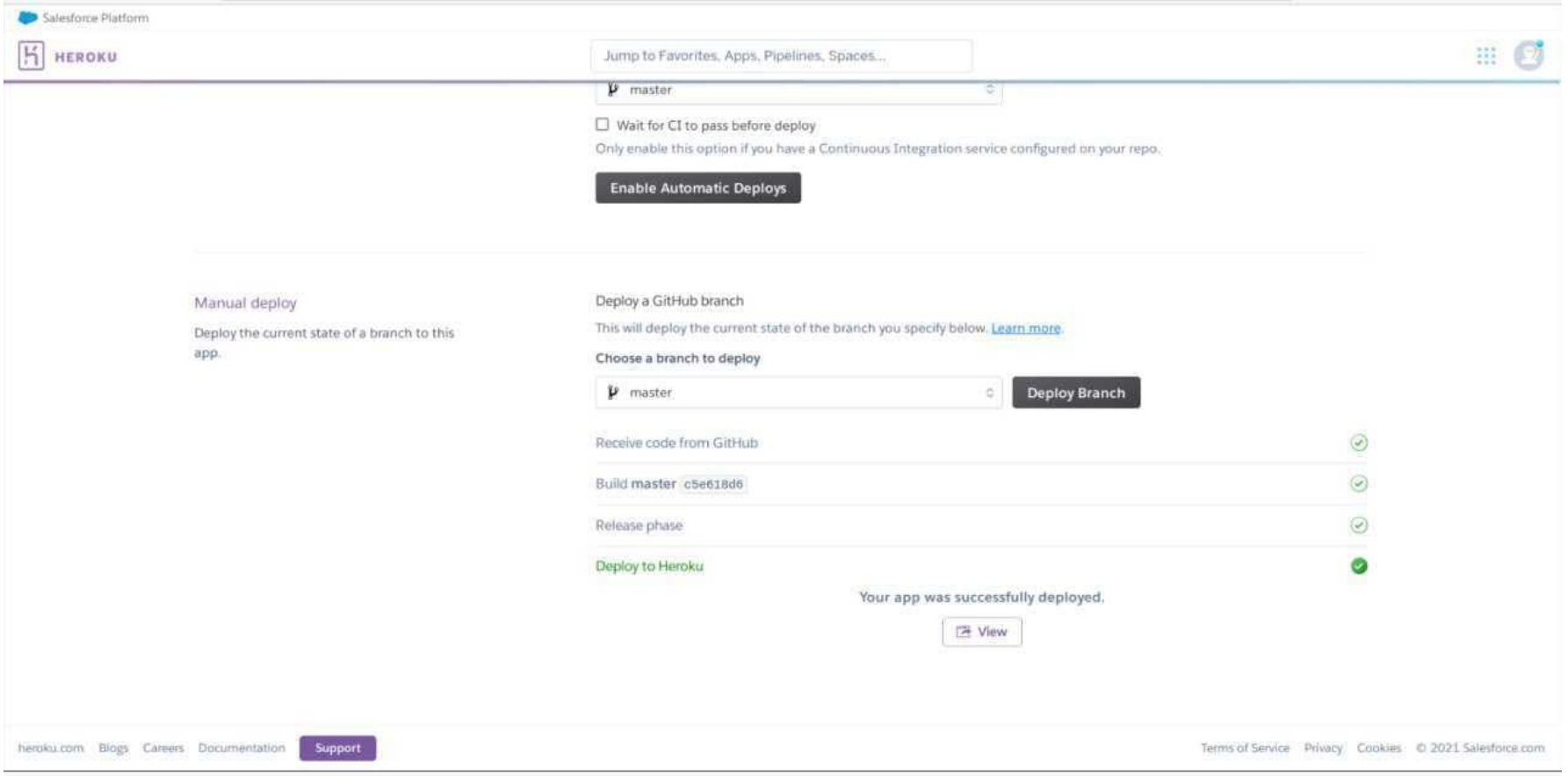
# 4-CSS

```css
@import url(https://fonts.googleapis.com/css?family=Open+Sans);

html { width: 100%; height:100%; overflow:hidden; }

body {
	width: 100%;
	height:100%;
	font-family: 'Helvetica';
	background: #000;
	color: #fff;
	font-size: 24px;
	text-align:center;
	letter-spacing:1.4px;

}
.login {
	position: absolute;
	top: 40%;
	left: 50%;
	margin: -150px 0 0 -150px;
	width:400px;
	height:400px;
}

.login h1 { color: #fff; text-shadow: 0 0 10px rgba(0,0,0,0.3); letter-spacing:1px; text-align:center; }

input {
	width: 100%;
	margin-bottom: 10px;
	background: rgba(0,0,0,0.3);
	border: none;
	outline: none;
	padding: 10px;
	font-size: 13px;
	color: #fff;
	text-shadow: 1px 1px 1px rgba(0,0,0,0.3);
	border: 1px solid rgba(0,0,0,0.3);
	border-radius: 4px;
	box-shadow: inset 0 -5px 45px rgba(100,100,100,0.2), 0 1px 1px rgba(255,255,255,0.2);
	-webkit-transition: box-shadow .5s ease;
	-moz-transition: box-shadow .5s ease;
	-o-transition: box-shadow .5s ease;
	-ms-transition: box-shadow .5s ease;
	transition: box-shadow .5s ease;
}
```

# 5-App.py

```python
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle

app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))

@app.route('/')
def home():
    return render_template('index.html')

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

    int_features = [float(x) for x in request.form.values()]
    final_features = [np.array(int_features)]
    prediction = model.predict(final_features)

    output = round(prediction[0], 2)

    return render_template('index.html', prediction_text='Tip should be $ {}'.format(output))

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

    data = request.get_json(force=True)
    prediction = model.predict([np.array(list(data.values()))])

    output = prediction[0]
    return jsonify(output)

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

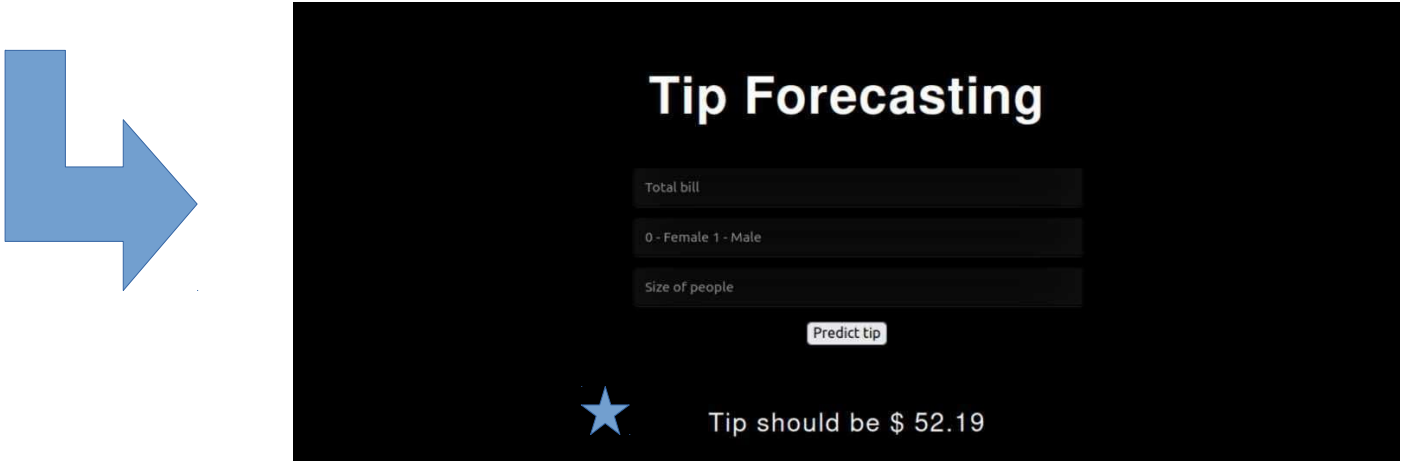
# 6-Deploy on Heroku



# 7-Results


Tip Forecasting
500
1
3
Predict tip


Tip Forecasting
Total bill
0 - Female 1 - Male
Size of people
Predict tip


Tip Forecasting
Total bill
0 - Female 1 - Male
Size of people
Predict tip
Tip should be $ 52.19