NAME: TEHESUMA MENSAH IMORO

BATCH CODE: LISUM30 (30TH JANUARY 2024 – 30TH

APRIL 2024)

SUBMISSION DATE: 13TH MARCH 2024

SUBMITTED TO: DATA GLACIER VIRTUAL INTERNSHIP

```
370650.01, 241167. , 115564. , 126014.01, 87474. , 235
                     352642.04, 234265. , 122811. , 126857. , 202030. 104650. , 156761. , 229656. , 279063. , 134161. 246696. , 185717.02, 263272. , 157872. , 339398.
                                                                                            , 110
                                                                                           , 94
                                                                                           , 214
                      85094. , 171112. , 121195. , 300593.03, 122171.
                                                                                           , 135
                     334572. , 97416. , 89004. , 140221. , 173967.
                     92733. , 86532. , 377824.06, 335437.01, 94702.
210329. , 361462. , 96162. , 264001.25, 189980.
                                                                                          , 101
                     117812. , 225855. , 195421. , 118046. , 254943. , 126
                     125248. , 161570. , 258126.01, 267996. , 249811. , 130 107985. , 360950.18, 149309. , 246804. , 380107.12, 162
                     216501. , 214788. , 130318. , 170642. , 135843. , 119 240119. , 141751. , 420149.27, 253704. , 347684. ])
In [22]: len(y pred)
Out[22]: 125
In [41]: # Savingvthe model
            import pickle
            pickle.dump(best_model, open('rfreg.pkl', 'wb'))
In [24]: # Loading the model
            loaded model = pickle.load(open("randomforest.pkl", "rb"))
```

Evaluation

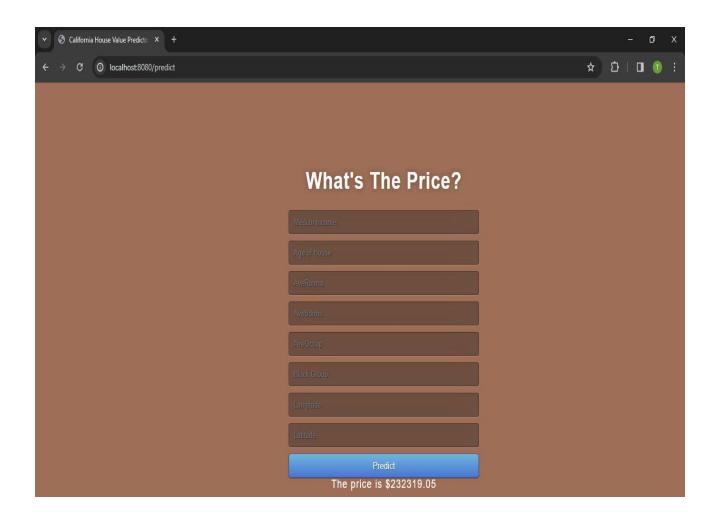
We are going to use the following to evaluate our model:

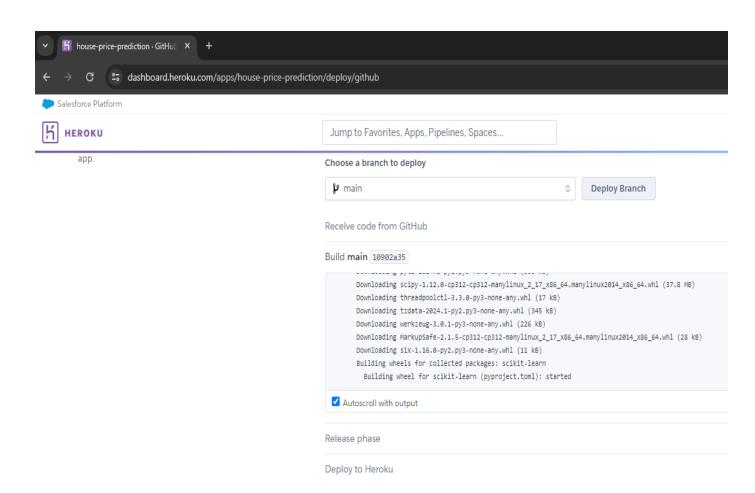
```
<!DOCTYPE html>
 <meta charset="UTF-8">
 <title>California House Value Predictor (/title>
 <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
<link href="https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300" rel='stylesheet' type='text/css'</pre>
<link rel='stylesheet' href="{{ url_for('static', filename='css/style.css') }}">
<div class="login">
 <h1>What's The Price? </h1>
     <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('predict')}}"method="post">
        <input type="text" name="MedInc" placeholder="Median Income" required="required" />
        <input type="text" name="HouseAge" placeholder="Age of House" required="required" />
        <input type="text" name="AveRooms" placeholder="AveRooms" required="required" />
        <input type="text" name="AveBdrms" placeholder="AveBdrms" required="required" />
        <input type="text" name="AveOccup" placeholder="AveOccup " required="required" />
        <input type="text" name="Population" placeholder="Block Group" required="required" />
```

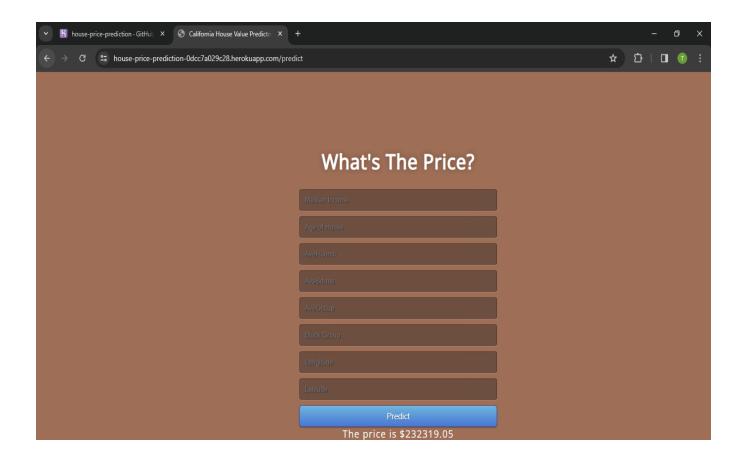
```
from flask import Flask, request, jsonify, render_template
     import pickle
     import numpy as np
     app = Flask(__name__)
     model = pickle.load(open('rfreg.pkl', 'rb'))
     @app.route('/')
     def home():
         return render_template('index.html')
     @app.route('/predict', methods=['POST'])
     def predict():
16
         For rendering results on HTML GUI
18
19
         int_features = [int(x) for x in request.form.values()]
20
         final_features = [np.array(int_features)]
         prediction = model.predict(final_features)
         output = round(prediction[0], 2)
25
         return render_template('index.html', prediction_text='The price is ${}'.format(output))
```

```
@import url(https://fonts.googleapis.com/css?family=Open+Sans);
    .btn { display: inline-block; *display: inline; *zoom: 1; padding: 4px 10px 4px; margin-bottom: 0; font-size:
    .btn:hover, .btn:active, .btn.active, .btn.disabled, .btn[disabled] { background-color: ■#e6e6e6; }
    .btn-large { padding: 9px 14px; font-size: 15px; line-height: normal; -webkit-border-radius: 5px; -moz-border
    .btn:hover { color: □#333333; text-decoration: none; background-color: □#e6e6e6; background-position: 0 -1
    .btn-primary, .btn-primary:hover { text-shadow: 0 -1px 0 □rgba(0, 0, 0, 0.25); color: ■#ffffff; }
    .btn-primary.active { color: ■rgba(255, 255, 255, 0.75); }
    .btn-primary { background-color: ■#4a77d4; background-image: -moz-linear-gradient(top, ■#6eb6de, ■#4a77d4
    btn-primary:hover, .btn-primary:active, .btn-primary.active, .btn-primary.disabled, .btn-primary[disabled]
    .btn-block { width: 100%; display:block; }
    * { -webkit-box-sizing:border-box; -moz-box-sizing:border-box; -ms-box-sizing:border-box; -o-box-sizing:borde
    html { width: 100%; height:100%; overflow:hidden; }
    body {
        width: 100%;
       height:100%;
        font-family: 'Open Sans', sans-serif;
        color: hsl(0, 0%, 98%);
                                                                               OBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL
warnings.warn(
```

```
↑ ≥ctenv) (
                                                                         rojer*\Real_Estate>python app.py
  Serving Flask app 'app'
 * Debug mode: on
  Running on http://localhost:8080
 ess CTRL+C to quit
 * Restarting with watchdog (windowsapi)
 * Debugger is active!
 * Debugger PIN: 615-196-968
127.0.0.1 - - [11/Mar/2024 20:31:31] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [11/Mar/2024 20:31:31] "GET /static/css/style.css HTTP/1.1" 200 -
127.0.0.1 - - [11/Mar/2024 20:31:32] "GET /favicon.ico HTTP/1.1" 404 -
                                      tenv\Lib\site-packages\sklearn\base.py:439: UserWarning: X does not have val
C:\U⊆
eature na , but wandon
                            ckegressor was fitted with feature names
 warnings.warn(
127.0.0.1 - - [11/Mar/2024 20:32:08] "POST /predict HTTP/1.1" 200 -
127.0.0.1 - - [11/Mar/2024 20:32:08] "GET /static/css/style.css HTTP/1.1" 304 -
```







```
main_app.py > ...
      from flask import Flask, request, jsonify
      import pickle
      import numpy as np
 4
      app = Flask(__name__)
      model = pickle.load(open('rfreg.pkl', 'rb'))
      @app.route('/')
      def home():
          return {'text': 'House Price Prediction'}
      @app.route('/predict', methods = ['GET'])
      def predict():
          MedInc = request.args.get('MedInc')
          HouseAge = request.args.get('HouseAge')
          AveRooms = request.args.get('AveRooms')
          AveBdrms = request.args.get('AveBdrms')
          AveOccup = request.args.get('AveOccup')
          Population = request.args.get('Population')
          Longitude = request.args.get('Longitude')
                  DEBUG CONSOLE

    py

                                 TERMINAL
 * Running on http://localhost:7000
Press CTRL+C to quit
 * Restarting with watchdog (windowsapi)
```

