Flask Deployment Web App Document

Name: New York Housing Market App
Report date: 28/02/2024
Internship Batch: LISUM30
Version: 1.0
Data intake by: Konstantinos Soufleros
Data intake reviewer: Data Glacier
Data storage location: https://github.com/kostas696/DG_Intern

Tabular data details: NY-House-Dataset.csv

Total number of observations	4801
Total number of files	1
Total number of features	17
Base format of the file	.csv
Size of the data	1.27 MB

1. my_flask_app.py

```
from flask import Flask, render_template, request
import numpy as np
import pandas as pd
import pickle
from sklearn.preprocessing import StandardScaler

app = Flask(__name__)

# Set the static folder path
app.config['STATIC_FOLDER'] = 'static'

# Load the scaler
with open('scaler.pkl', 'rb') as scaler_file:
    scaler = pickle.load(scaler_file)

# Load the model
with open('gradient_boosting_model.pkl', 'rb') as model_file:
    model = pickle.load(model_file)

# Define the prediction route
@app.route('/', methods=['GET', 'POST'])
def predict():
    if request.method == 'POST':
        # Get the form data
        property_type = request.form['property_type']
        neighborhood = request.form['neighborhood']
        property_sqft = float(request.form['property_sqft'])
```

```
        bedrooms = int(request.form['bedrooms'])
        baths = int(request.form['baths'])

        # Load the column names of the encoded features
        with open('encoded_columns.pkl', 'rb') as f:
          encoded_columns = pickle.load(f)

        # Perform one-hot encoding for categorical variables
        type_encoded =
pd.get_dummies(pd.Series(property_type)).reindex(columns=encoded_column
s, fill_value=0)
        sublocality_encoded =
pd.get_dummies(pd.Series(neighborhood)).reindex(columns=encoded_columns
, fill_value=0)

        # Concatenate encoded features
        encoded_features = pd.concat([type_encoded,
sublocality_encoded], axis=1)

        # Scale the entire feature vector (both numerical and encoded
categorical features)
        input_features = np.array([[property_sqft, bedrooms,
baths]])  # Numerical features
        input_features_scaled =
scaler.transform(np.concatenate([input_features,
encoded_features.values], axis=1))

        # Make the prediction
        prediction = model.predict(input_features_scaled)
        output = round(prediction[0], 2)

        # Display the prediction result
        return render_template('index.html', prediction_text='House
price should be ${} '.format(output))

    return render_template('index.html')

if __name__ == '__main__':
    app.run(port=5000, debug=False)
```

2. index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>House Price Estimation in New York</title>
    <style>
        body {
            background-image: url('{{ url_for("static", filename="55443.jpg") }}');
            background-size: cover;
            background-repeat: no-repeat;
            background-attachment: fixed;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            margin: 0;
        }
        .form-container {
            background-color: rgba(255, 255, 255, 0.8);
            padding: 20px;
            border-radius: 10px;
        }

    </style>
</head>
<body>
    <div class="form-container">
        <h1 style="text-align: center; color: #333;">House Price Estimation in New York</h1>
        <form id="prediction-form" style="text-align: center;">
            <label for="property-type">Type of Property:</label>
            <select id="property-type" name="property_type">
            <option value="Condo for sale">Condo for sale</option>
            <option value="House for sale">House for sale</option>
            <option value="Townhouse for sale">Townhouse for sale</option>
            <option value="Co-op for sale">Co-op for sale</option>
            <option value="Multi-family home for sale">Multi-family home for sale</option>
```

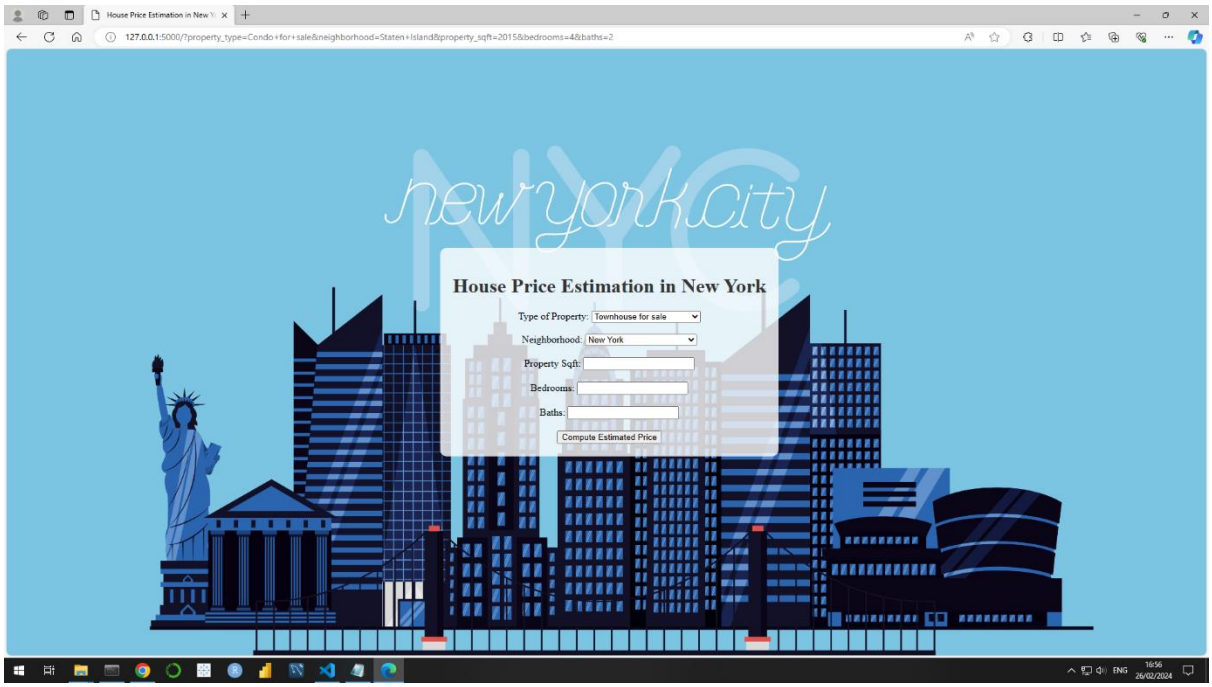
```
            <option value="For sale">For sale</option>
            <option value="Contingent">Contingent</option>
            <option value="Land for sale">Land for sale</option>
            <option value="Foreclosure">Foreclosure</option>
            <option value="Pending">Pending</option>
            <option value="Coming Soon">Coming Soon</option>
            <option value="Mobile house for sale">Mobile house for
   sale</option>
            <option value="Condop for sale">Condop for sale</option>
        </select>
        <br><br>
        <label for="neighborhood">Neighborhood:</label>
        <select id="neighborhood" name="neighborhood">
            <option value="New York">New York</option>
            <option value="Staten Island">Staten Island</option>
            <option value="Manhattan">Manhattan</option>
            <option value="Brooklyn">Brooklyn</option>
            <option value="Bronx">Bronx</option>
            <option value="Jackson Heights">Jackson Heights</option>
            <option value="Elmhurst">Elmhurst</option>
            <option value="Woodside">Woodside</option>
            <option value="Rego Park">Rego Park</option>
            <option value="Forest Hills">Forest Hills</option>
            <option value="Briarwood">Briarwood</option>
            <option value="Queens">Queens</option>
            <option value="Flushing">Flushing</option>
            <option value="Woodhaven">Woodhaven</option>
            <option value="Jamaica">Jamaica</option>
            <option value="Richmond Hill South">Richmond Hill
   South</option>
            <option value="Whitestone">Whitestone</option>
            <option value="Ridgewood">Ridgewood</option>
            <option value="Rosedale">Rosedale</option>
            <option value="Ozone Park">Ozone Park</option>
            <option value="Springfield Gardens">Springfield
   Gardens</option>
            <option value="Far Rockaway">Far Rockaway</option>
            <option value="Bellerose">Bellerose</option>
            <option value="Bayside">Bayside</option>
            <option value="Astoria">Astoria</option>
            <option value="Floral Park">Floral Park</option>
            <option value="Fresh Meadows">Fresh Meadows</option>
            <option value="Howard Beach">Howard Beach</option>
            <option value="Cambria Heights">Cambria Heights</option>
            <option value="Rockaway Park">Rockaway Park</option>
            <option value="East Elmhurst">East Elmhurst</option>
            <option value="Little Neck">Little Neck</option>
            <option value="Long Island City">Long Island City</option>
```

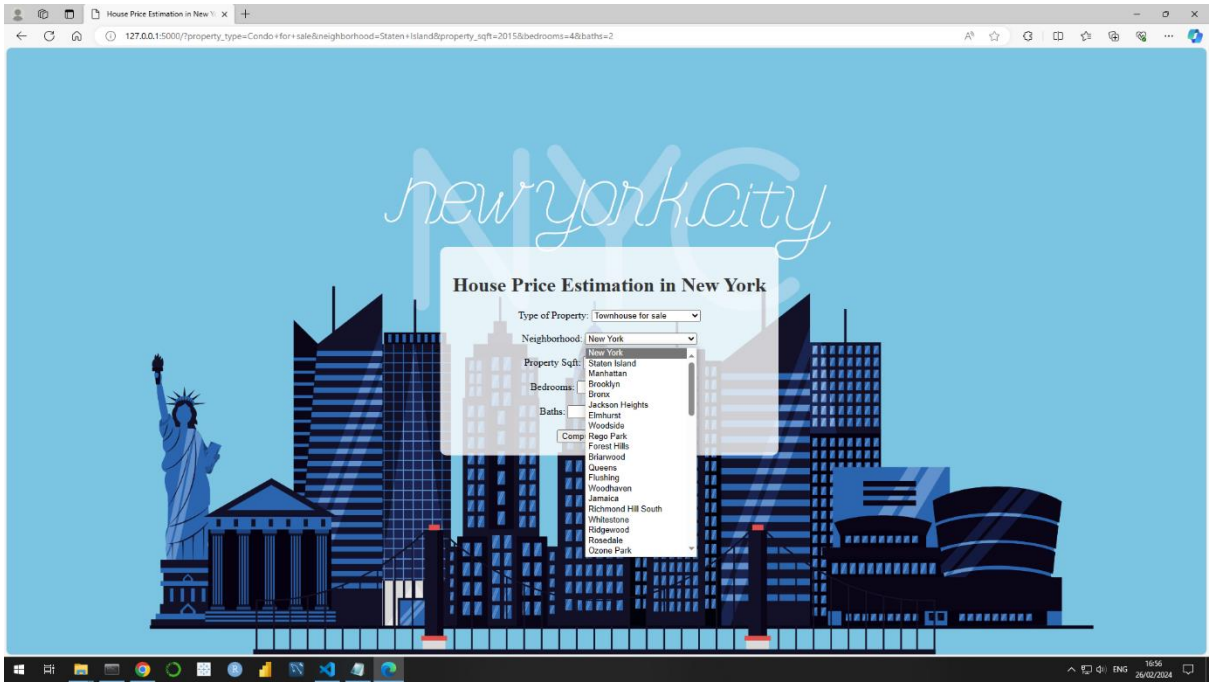
```
85.             <option value="Corona">Corona</option>
86.             <option value="Kew Gardens">Kew Gardens</option>
87.             <option value="Brownville">Brownville</option>
88.             <option value="Brooklyn Heights">Brooklyn Heights</option>
89.             <option value="Maspeth">Maspeth</option>
90.             <option value="Queens Village">Queens Village</option>
91.             <option value="Richmond Hill">Richmond Hill</option>
92.             <option value="Middle Village">Middle Village</option>
93.             <option value="Saint Albans">Saint Albans</option>
94.             <option value="Kensington">Kensington</option>
95.             <option value="South Ozone Park">South Ozone Park</option>
96.             <option value="Canarsie">Canarsie</option>
97.             <option value="College Point">College Point</option>
98.             <option value="Bedford Stuyvesant">Bedford
   Stuyvesant</option>
99.             <option value="Hollis">Hollis</option>
100.                <option value="Malba">Malba</option>
101.                <option value="Glen Oaks">Glen Oaks</option>
102.                <option value="Douglaston">Douglaston</option>
103.                <option value="Sunnyside">Sunnyside</option>
104.                <option value="Arverne">Arverne</option>
105.                <option value="Glendale">Glendale</option>
106.                <option value="New Hyde Park">New Hyde Park</option>
107.                <option value="Crown Heights">Crown Heights</option>
108.                <option value="Old Mill Basin">Old Mill
   Basin</option>
109.                <option value="Beechhurst">Beechhurst</option>
110.                <option value="Roosevelt Island">Roosevelt
   Island</option>
111.                <option value="Kew Gardens Hills">Kew Gardens
   Hills</option>
112.                <option value="Stuyvesant Heights">Stuyvesant
   Heights</option>
113.                <option value="Belle Harbor">Belle Harbor</option>
114.                <option value="East Flatbush">East Flatbush</option>
115.                <option value="Kew Garden Hills">Kew Garden
   Hills</option>
116.                <option value="Ditmas Park">Ditmas Park</option>
117.                <option value="Brighton Beach">Brighton
   Beach</option>
118.                <option value="Prospect Lefferts Gardens">Prospect
   Lefferts Gardens</option>
119.            </select>
120.            <br><br>
121.            <label for="property-sqft">Property Sqft:</label>
122.                <input type="number" id="property-sqft"
   name="property_sqft" min="0">
123.                <br><br>
```

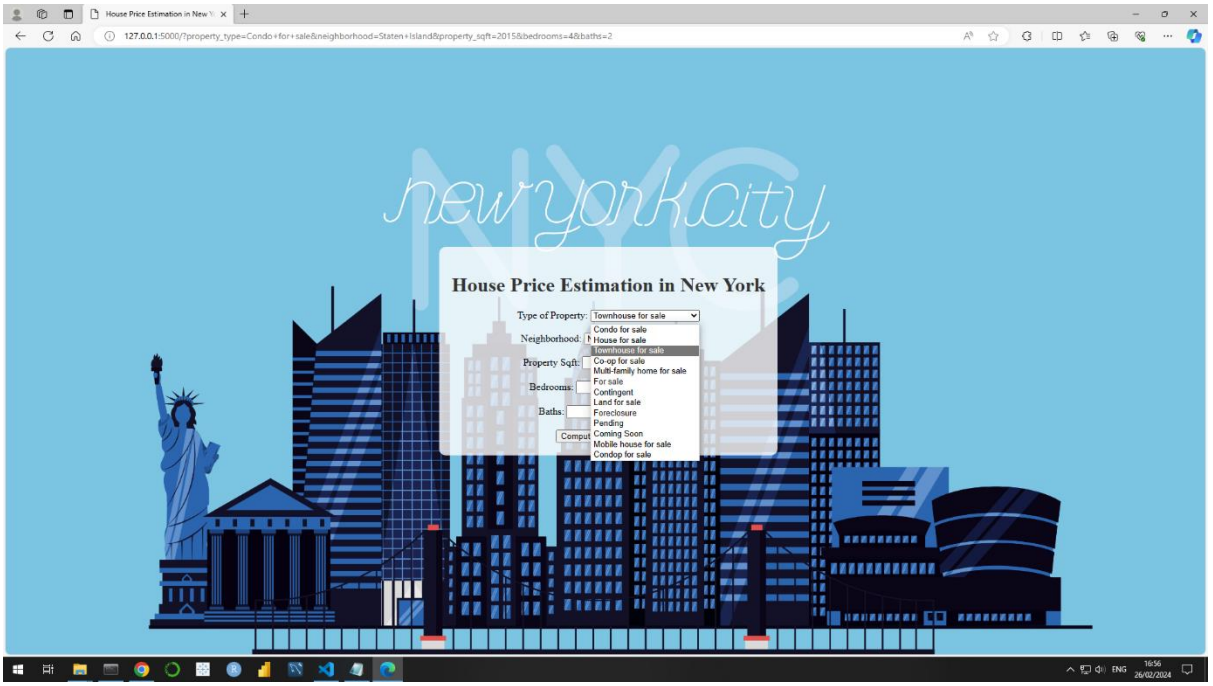
```
                <label for="bedrooms">Bedrooms:</label>
                <input type="number" id="bedrooms" name="bedrooms" min="0">
                <br><br>
                <label for="baths">Baths:</label>
                <input type="number" id="baths" name="baths" min="0">
                <br><br>
                <button type="submit">Compute Estimated Price</button>
            </form>
            <!-- Display prediction result here -->
            <div id="prediction-result"></div>
        </div>

        <script>
            const form = document.getElementById('prediction-form');

            form.addEventListener('submit', (event) => {
                event.preventDefault();

                // Get the form data
                const propertyType = document.getElementById('property-type').value;
                const neighborhood = document.getElementById('neighborhood').value;
                const propertySqft = document.getElementById('property-sqft').value;
                const bedrooms = document.getElementById('bedrooms').value;
                const baths = document.getElementById('baths').value;

                // Send the data to the server using a POST request
                fetch('/predict', {
                    method: 'POST',
                    headers: {
                        'Content-Type': 'application/json'
                    },
                    body: JSON.stringify({
                        property_type: propertyType,
                        neighborhood: neighborhood,
                        property_sqft: propertySqft,
                        bedrooms: bedrooms,
                        baths: baths
                    })
                })
                .then(response => response.json())
                .then(data => {
                    // Display the prediction result
```

```
                    document.getElementById('prediction-
    result').innerHTML = data.prediction_text;
                });
            });

        </script>
    </body>
    </html>
```

3. Result




House Price Estimation in New York
Type of Property:
Townhouse for sale
Neighborhood:
New York
Property Sqft:
Bedrooms:
Baths:
Comp
New York
Staten Island
Manhattan
Brooklyn
Bronx
Jackson Heights
Elmhurst
Woodside
Rego Park
Forest Hills
Briarwood
Queens
Flushing
Woodhaven
Jamaica
Richmond Hill South
Whitestone
Ridgewood
Rosedale
Ozone Park


House Price Estimation in New York
Type of Property:
Townhouse for sale
Neighborhood:
Property Sqft:
Bedrooms:
Baths:
Compu
Condo for sale
House for sale
Townhouse for sale
Co-op for sale
Multi-family home for sale
For sale
Contingent
Land for sale
Foreclosure
Pending
Coming Soon
Mobile house for sale
Condop for sale