



Week4: Deployment on Flask

Name: Munirah Alfehaid

Batch code: LISUM35

Submission date: 24/7/2024

Submitted to: Data Glacier

Introduction

This assignment involves building and deploying a machine learning model for predicting real estate prices based on various features. The primary objectives are to:

1. Train a regression model using a provided real estate dataset.
2. Save the trained model using `pickle` for later use.
3. Create a web application using Flask to accept input data, predict the house price per unit area, and display the results.

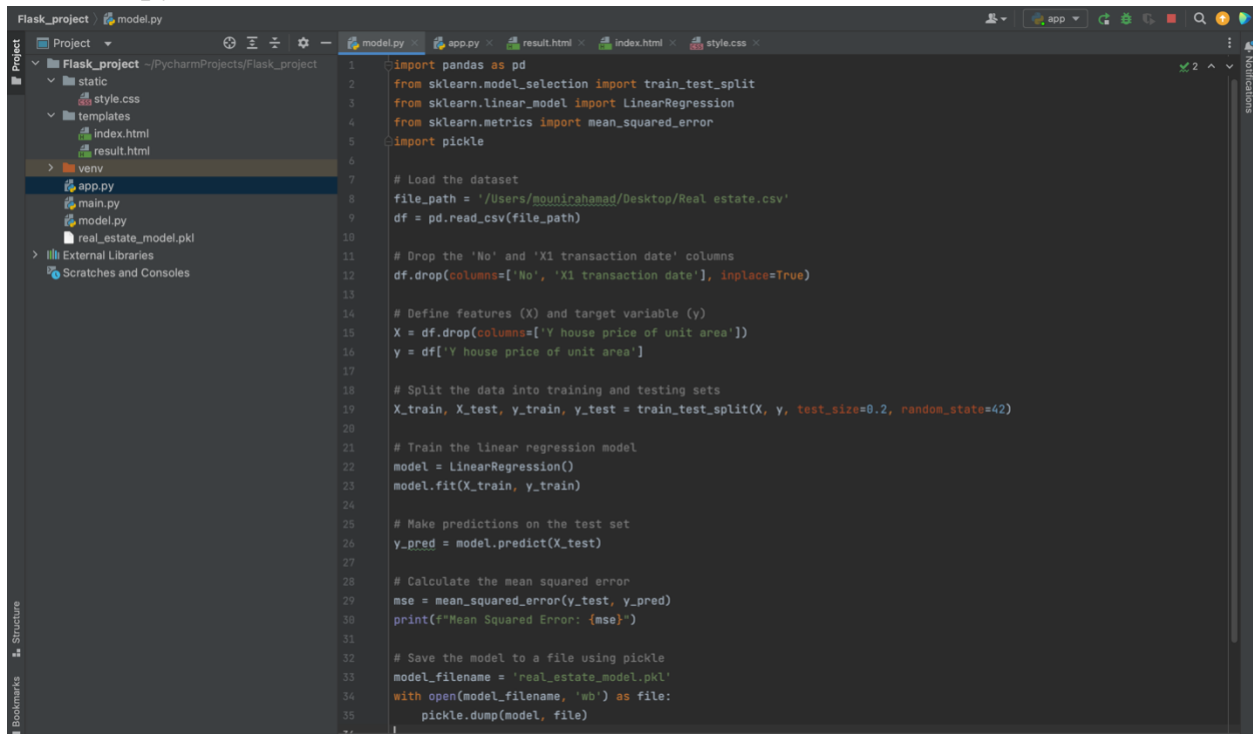
The dataset consists of information about house transactions, including the age of the house, distance to the nearest MRT station, number of convenience stores nearby, and the geographical coordinates (latitude and longitude). The target variable is the house price per unit area.

To complete this assignment, the project is structured with separate files for training the model, setting up the Flask web application, and defining HTML templates and CSS for the user interface. This clear structure ensures that the project is organized and maintainable.

The web application allows users to input the necessary features, predicts the house price using the trained model, and displays the prediction on a results page. This assignment demonstrates the practical application of machine learning in a real-world scenario, integrating model training, persistence, and deployment in a user-friendly web interface.

Snapshot of deployment steps:

1-model.py



```
1 import pandas as pd
2 from sklearn.model_selection import train_test_split
3 from sklearn.linear_model import LinearRegression
4 from sklearn.metrics import mean_squared_error
5 import pickle
6
7 # Load the dataset
8 file_path = '/Users/mounirahamad/Desktop/Real estate.csv'
9 df = pd.read_csv(file_path)
10
11 # Drop the 'No' and 'X1 transaction date' columns
12 df.drop(columns=['No', 'X1 transaction date'], inplace=True)
13
14 # Define features (X) and target variable (y)
15 X = df.drop(columns=['Y house price of unit area'])
16 y = df['Y house price of unit area']
17
18 # Split the data into training and testing sets
19 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
20
21 # Train the linear regression model
22 model = LinearRegression()
23 model.fit(X_train, y_train)
24
25 # Make predictions on the test set
26 y_pred = model.predict(X_test)
27
28 # Calculate the mean squared error
29 mse = mean_squared_error(y_test, y_pred)
30 print(f"Mean Squared Error: {mse}")
31
32 # Save the model to a file using pickle
33 model_filename = 'real_estate_model.pkl'
34 with open(model_filename, 'wb') as file:
35     pickle.dump(model, file)
```

2-app.py

The image displays two screenshots of a PyCharm IDE window, showing the development of a Flask application named 'app.py'.

Top Screenshot: The code editor shows the initial setup of the application. It imports Flask, request, render_template, pickle, and pandas. It defines the Flask app and loads a pre-trained model from 'real_estate_model.pkl'. The routes defined are a home page and a prediction endpoint.

```
1 from flask import Flask, request, render_template
2 import pickle
3 import pandas as pd
4
5 app = Flask(__name__)
6
7 # Load the model using pickle
8 model_filename = 'real_estate_model.pkl'
9 with open(model_filename, 'rb') as file:
10     model = pickle.load(file)
11
12
13 @app.route('/')
14 def home():
15     return render_template('index.html')
16
17
18 @app.route('/predict', methods=['POST'])
19 def predict():
20     # Get data from the form
21     data = request.form.to_dict()
22     df = pd.DataFrame([data])
23
24     # Convert data types
25     df = df.astype(float)
26
27     # Ensure the input features match the training features
28     df = df[['X2 house age', 'X3 distance to the nearest MRT station', 'X4 number of convenience stores', 'X5 latitude',
29             'X6 longitude']]
30
31     # Make prediction
32     prediction = model.predict(df)
33
34     # Render the result template with the prediction
35     return render_template('result.html', prediction=prediction[0])
```

Bottom Screenshot: The code editor shows the completed application. It includes the same routes as the top screenshot, but adds a main block to run the application in debug mode. The terminal window at the bottom shows the command 'python app.py' being executed.

```
11
12
13 @app.route('/')
14 def home():
15     return render_template('index.html')
16
17
18 @app.route('/predict', methods=['POST'])
19 def predict():
20     # Get data from the form
21     data = request.form.to_dict()
22     df = pd.DataFrame([data])
23
24     # Convert data types
25     df = df.astype(float)
26
27     # Ensure the input features match the training features
28     df = df[['X2 house age', 'X3 distance to the nearest MRT station', 'X4 number of convenience stores', 'X5 latitude',
29             'X6 longitude']]
30
31     # Make prediction
32     prediction = model.predict(df)
33
34     # Render the result template with the prediction
35     return render_template('result.html', prediction=prediction[0])
36
37
38 if __name__ == '__main__':
39     app.run(debug=True)
40
```

The terminal window shows the command 'python app.py' being executed, and the output 'Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)'.

3-index.html

```
Flask_project / templates / index.html
Project
  Flask_project - /PycharmProjects/Flask_project
    static
      style.css
    templates
      index.html
      result.html
    venv
      app.py
      main.py
      model.py
      real_estate_model.pkl
  External Libraries
  Scratches and Consoles

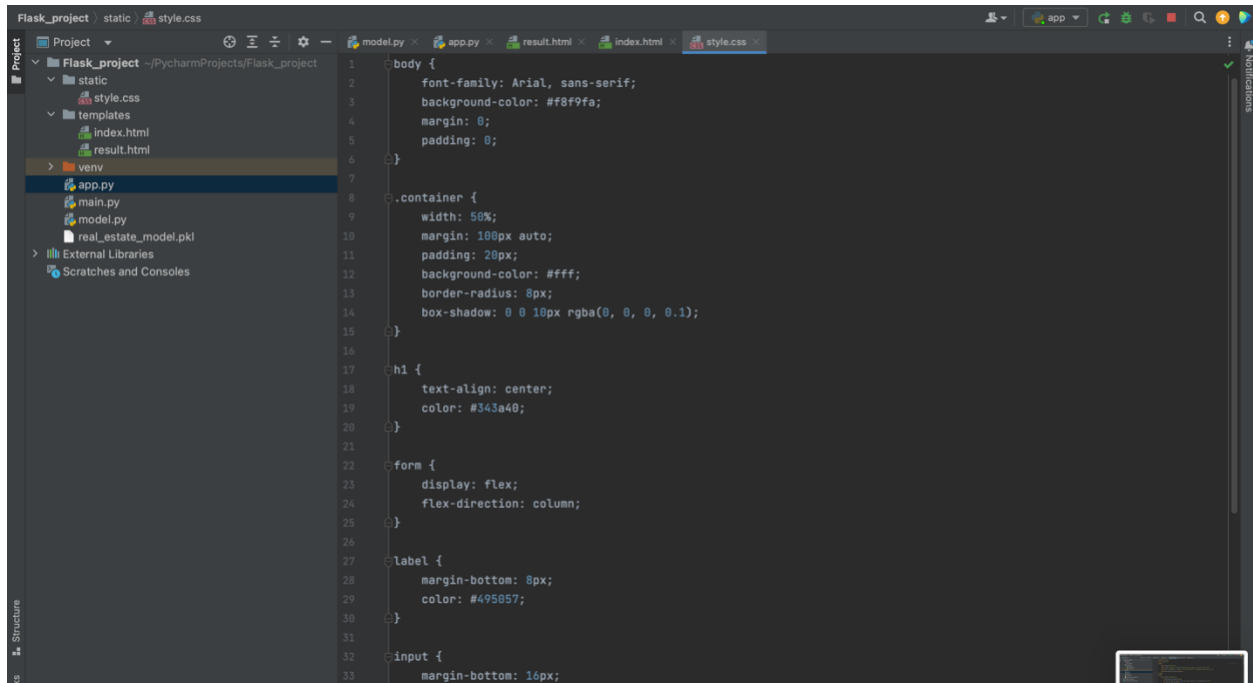
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
7   <title>Real Estate Price Prediction</title>
8 </head>
9 <body>
10  <div class="container">
11    <h1>Real Estate Price Prediction</h1>
12    <form action="/predict" method="post">
13      <label for="X2 house age">House Age:</label>
14      <input type="text" id="X2 house age" name="X2 house age" required<br>
15
16      <label for="X3 distance to the nearest MRT station">Distance to MRT:</label>
17      <input type="text" id="X3 distance to the nearest MRT station" name="X3 distance to the nearest MRT station"
18
19      <label for="X4 number of convenience stores">Number of Convenience Stores:</label>
20      <input type="text" id="X4 number of convenience stores" name="X4 number of convenience stores" required<br>
21
22      <label for="X5 latitude">Latitude:</label>
23      <input type="text" id="X5 latitude" name="X5 latitude" required<br>
24
25      <label for="X6 longitude">Longitude:</label>
26      <input type="text" id="X6 longitude" name="X6 longitude" required<br>
27
28      <button type="submit">Predict</button>
29    </form>
30  </div>
31 </body>
32 </html>
33
```

4-result.html

```
Flask_project / templates / result.html
Project
  Flask_project - /PycharmProjects/Flask_project
    static
      style.css
    templates
      index.html
      result.html
    venv
      app.py
      main.py
      model.py
      real_estate_model.pkl
  External Libraries
  Scratches and Consoles

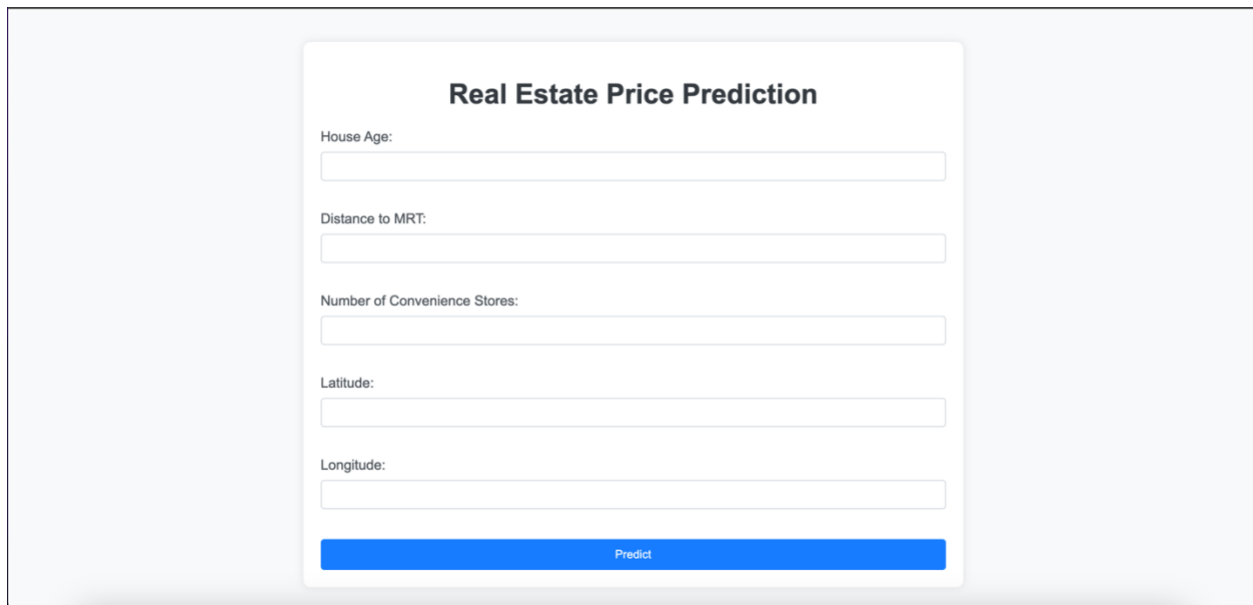
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
7   <title>Prediction Result</title>
8 </head>
9 <body>
10  <div class="container">
11    <h1>Prediction Result</h1>
12    <p>The predicted house price per unit area is: {{ prediction }}</p>
13    <a href="/">Go back</a>
14  </div>
15 </body>
16 </html>
17
```

5-style.css



```
1 body {  
2     font-family: Arial, sans-serif;  
3     background-color: #f8f9fa;  
4     margin: 0;  
5     padding: 0;  
6 }  
7  
8 .container {  
9     width: 50%;  
10    margin: 100px auto;  
11    padding: 20px;  
12    background-color: #fff;  
13    border-radius: 8px;  
14    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
15 }  
16  
17 h1 {  
18     text-align: center;  
19     color: #343a40;  
20 }  
21  
22 form {  
23     display: flex;  
24     flex-direction: column;  
25 }  
26  
27 label {  
28     margin-bottom: 8px;  
29     color: #495b57;  
30 }  
31  
32 input {  
33     margin-bottom: 16px;
```

6-Final result:



Real Estate Price Prediction

House Age:

Distance to MRT:

Number of Convenience Stores:

Latitude:

Longitude:

Real Estate Price Prediction

House Age:

15

Distance to MRT:

390

Number of Convenience Stores:

4

Latitude:

24.9

Longitude:

121.5

Predict

Prediction Result

The predicted house price per unit area is: 26.339898609290685

[Go back](#)