

Model Deployment on Flask

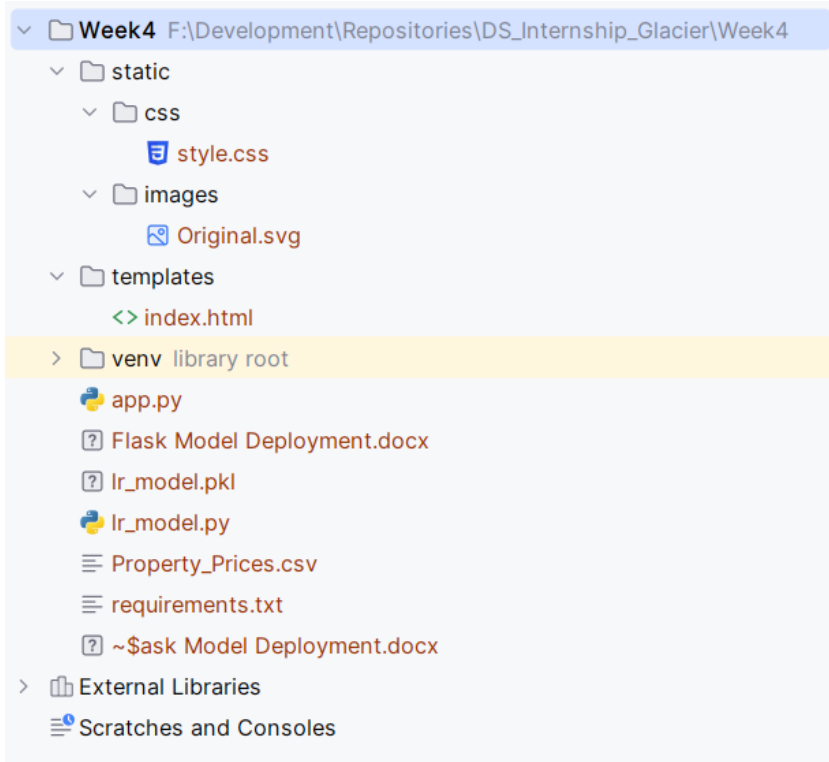
Name: Olena Panchenko

Batch code: LISUM30

Submission date: 28.02.2024

Submitted to: Data Glacier

1. Project Directory Structure



2. A list of dependencies and libraries used in this project

requirements.txt

```
1 blinker==1.7.0
2 click==8.1.7
3 colorama==0.4.6
4 Flask==3.0.2
5 itsdangerous==2.1.2
6 Jinja2==3.1.3
7 joblib==1.3.2
8 MarkupSafe==2.1.5
9 numpy==1.26.4
10 pandas==2.2.1
11 python-dateutil==2.8.2
12 pytz==2024.1
13 scikit-learn==1.4.1.post1
14 scipy==1.12.0
15 six==1.16.0
16 threadpoolctl==3.3.0
17 tzdata==2024.1
18 Werkzeug==3.0.1
```

3. Data

Bedrooms	Bathrooms	Floor Area(ft2)	Council Tax Band	Price(£)
3	1	1022	C	350000
5	3	2700	F	1500000
6	5	4300		1350000
7	5		F	1295000
3	1		A	260000
4	4		G	1200000
3	1		C	300000
7	2	4000	E	950000
5	3	3821	F	950000
6	3		F	925000
4	2	1283	F	925000
3	2		E	900000
2	2			325000
1	1			120000
2	1		A	120000
2	1		A	180000
1	1		A	90000

4. Data Preprocessing

lr_model.py ×

```
1 import pandas as pd
2 import pickle
3 from sklearn.linear_model import LinearRegression
4
5 dataset = pd.read_csv('Property Prices.csv')
6
7 dataset.loc[[13, 16], 'Floor Area(ft2)'] = 400
8 no_missing_area_data = dataset.loc[:, ['Bedrooms', 'Bathrooms', 'Floor Area(ft2)']].dropna(subset=['Floor Area(ft2)'])
9 missing_area_data = dataset.loc[:, ['Bedrooms', 'Bathrooms', 'Floor Area(ft2)']][dataset['Floor Area(ft2)'].isnull()]
10 no_missing_area_predictors = no_missing_area_data.drop(columns=['Floor Area(ft2)'])
11 no_missing_area_targets = no_missing_area_data['Floor Area(ft2)']
12 missing_area_predictors = missing_area_data.drop(columns=['Floor Area(ft2)'])
13
14 mv_model = LinearRegression()
15 mv_model.fit(no_missing_area_predictors, no_missing_area_targets)
16 predicted_areas = mv_model.predict(missing_area_predictors)
17 dataset.loc[dataset['Floor Area(ft2)'].isnull(), ['Floor Area(ft2)']] = predicted_areas
18
19 encoding_dict = {'A': 1, 'B': 2, 'C': 3, 'D': 4, 'E': 5, 'F': 6, 'G': 7, 'H': 8}
20 dataset['Council Tax Band'] = dataset['Council Tax Band'].map(encoding_dict)
21
22 mode_by_bedrooms = dataset.groupby('Bedrooms')['Council Tax Band'].transform(lambda x: x.mode().iloc[0])
23 dataset['Council Tax Band'] = dataset['Council Tax Band'].fillna(mode_by_bedrooms)
```

5. Model Training and Saving

```
25 model_predictors = dataset.iloc[:, :4].values
26 model_targets = dataset.iloc[:, -1]
27 regressor = LinearRegression()
28 regressor.fit(model_predictors, model_targets)
29 pickle.dump(regressor, open('lr_model.pkl', 'wb'))
```

6. HTML template

<> index.html x

```
1 <!DOCTYPE html>
2 <html >
3 <head>
4   <meta charset="UTF-8">
5   <title>ML API</title>
6   <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
7   <link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
8   <link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
9   <link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
10  <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
11
12 </head>
13
14 <body>
15   <div class="login">
16     <h1>Predict House Price</h1>
17
18     <!-- Main Input For Receiving Query to our ML -->
19     <form action="{{ url_for('predict')}}" method="post">
20       <input type="text" name="Bedrooms" placeholder="Number of Bedrooms" required="required" />
21       <input type="text" name="Bathrooms" placeholder="Number of Bathrooms" required="required" />
22       <input type="text" name="Floor Area(ft2)" placeholder="Floor Area(ft2)" required="required" />
23       <input type="text" name="Council Tax Band" placeholder="Council Tax Band" required="required" />
24
25       <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
26     </form>
27
28     <br>
29     <br>
30     {{ prediction_text }}
31
32   </div>
33   
34
35 </body>
36 </html>
```

6. Model Deployment

app.py x

```
1 import numpy as np
2 from flask import Flask, request, render_template
3 import pickle
4
5 app = Flask(__name__)
6 model = pickle.load(open('lr_model.pkl', 'rb'))
7
8
9 @app.route('/')
10 def home():
11     return render_template('index.html')
12
13
14 2 usages (2 dynamic)
15 @app.route(rule: '/predict', methods=['POST'])
16 def predict():
17     """
18     For rendering results on HTML GUI
19     """
20     int_features = [int(x) for x in request.form.values()]
21     final_features = [np.array(int_features)]
22     prediction = model.predict(final_features)
23
24     output = round(prediction[0], 2)
25
26     return render_template(template_name_or_list: 'index.html', prediction_text='House price should be £ {}'.format(output))
27
28 if __name__ == "__main__":
29     app.run(debug=True)
```

7. Model Testing

Predict House Price


Number of Bedrooms

Number of Bathrooms

Floor Area(ft2)

Council Tax Band

Predict

 **Data Glacier**

Your Deep Learning Partner

5:37 PM
2/28/2024

Predict House Price


3

1

3000

3

Predict

 **Data Glacier**

Your Deep Learning Partner

5:38 PM
2/28/2024

Predict House Price

Number of Bedrooms

Number of Bathrooms

Floor Area(ft2)

Council Tax Band

Predict

House price should be £ 172543.14



Data Glacier

Your Deep Learning Partner