

Flask Deployment of Dummy Model

Step 1. Download the following dataset from: <https://www.kaggle.com/datasets/syuzai/perth-house-prices> (saved as all_perth_310121.csv)

Step 2. Create model by the filename model.py, saved in same directory as csv file

```
1 import pandas as pd
2 import pickle
3 from datetime import datetime
4 from sklearn.linear_model import LinearRegression
5 from sklearn.model_selection import train_test_split
6
7 toy_data = pd.read_csv("all_perth_310121.csv")
8
9 toy_data["DATE_SOLD"] = toy_data["DATE_SOLD"].map(lambda x: datetime.strptime(x, "%m-%Y").toordinal())
10 toy_data["GARAGE"] = toy_data["GARAGE"].fillna(0)
11 toy_data["BUILD_YEAR"] = toy_data["BUILD_YEAR"].fillna(2022)
12 toy_data["NEAREST_SCH_RANK"] = toy_data["NEAREST_SCH_RANK"].fillna(0)
13
14 X = toy_data[["BEDROOMS", \
15               "BATHROOMS", \
16               "GARAGE", \
17               "LAND_AREA", \
18               "FLOOR_AREA", \
19               "BUILD_YEAR", \
20               "#CBD_DIST", \
21               "#NEAREST_STN_DIST", \
22               "#DATE_SOLD", \
23               "#LATITUDE", \
24               "#LONGITUDE", \
25               "#NEAREST_SCH_DIST", \
26               "#NEAREST_SCH_RANK" \
27               ]]
28 y = toy_data["PRICE"]
29
30 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3)
31
32 linreg = LinearRegression()
33 linreg.fit(X_train, y_train)
34
35 pickle.dump(linreg, open("model.pickle", "wb"))
```

Step 3. After running the Python code, which generates a file model.pickle, create another file app.py in the same directory

```
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('model.pickle', 'rb'))
7
8 @app.route('/')
9 def home():
10     return render_template('index.html')
11
12 @app.route('/predict', methods=['POST'])
13 def predict():
14     """
15     For rendering results on HTML GUI
16     """
17     int_features = [float(x) for x in request.form.values()]
18     final_features = [np.array(int_features)]
19     prediction = model.predict(final_features)
20
21     output = round(prediction[0], 2)
22
23     return render_template('index.html', prediction_text='House price should be $ {}'.format(output))
24
25 if __name__ == "__main__":
26     app.run(debug=True)
```

Step 4. Download and copy the static and templates folders in the Flask-Deployment available from Canvas

Step 5. Modify the highlighted section in index.html to use the prediction variables: Bedrooms, Bathrooms, Garage spaces, Land area (m²), Floor area (m²), and Year of construction

```
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>House Price Prediction in Perth</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="GARAGE" placeholder="Number of Garage Spaces" required="required" />
23       <input type="text" name="LAND_AREA" placeholder="Land Area (m2)" required="required" />
24       <input type="text" name="FLOOR_AREA" placeholder="Floor Area (m2)" required="required" />
25       <input type="text" name="BUILD_YEAR" placeholder="Year Built" required="required" />
26
27       <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
28     </form>
29
30     <br>
31     <br>
32     {{ prediction_text }}
33
34   </div>
35   
36
37 </body>
38 </html>
39
```

Step 6. To run the code, run app.py and paste the highlighted URL in a browser

```
Select C:\windows\py.exe
* Serving Flask app 'app' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000 (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 113-145-372
```

Step 7. Running in a browser opens the app interface. Enter desired values to predict house prices based on the model (serialized into model.pickle)

ML API x +

127.0.0.1:5000/predict

Inbox - ray.ng.1999... Inbox - ray.ng.2018... Canvas UBC Student Servic... Messenger Your Projects - Ove... TAO Connect v3 Data Science: A Firs...

House Price Prediction in Perth

3

2

0


500

70

2012

Predict

House price should be \$ 286452.41

 **Data Glacier**

Your Deep Learning Partner

17°C Clear

9:57 PM 2022-06-23