

Melbourne House Price Prediction



Name: Suvansh Vaid

Batch Code: LISP01

Submission Date: 20 March 2021

Submitted to: <https://github.com/SuvanshVaid27/Melbourne-House-Price-Estimation>

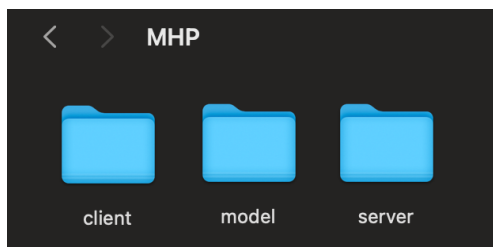
About the project:

This project was selected with the aim of understanding the model deployment process in Flask. The data was taken from a Kaggle repository:

<https://www.kaggle.com/anthonyypino/melbourne-housing-market>.

Project walkthrough:

1. Data Cleaning and model training:
The data was first downloaded from the Kaggle link and a basic exploratory analysis was performed in Python ([EDA file](#)).
2. Model training and serialization:
A random forest was trained on the data using sklearn libraries and a model with ~72% accuracy was created for now. The model was serialized using pickle.
3. Directory Structure for Flask App:
The directory structure for the flask app was deliberately chosen to be distributed among three folders namely Client, Model and Server, as shown.



4. Configuring server for Flask:
A simple Flask server was configured using python and was running in the background to be used for serving the client's http requests.

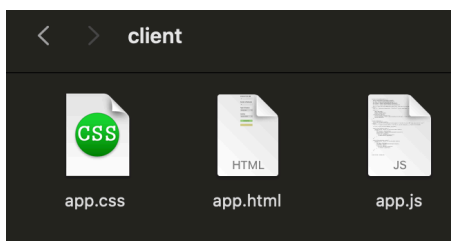
```

/usr/local/bin/python3.9 "/Users/badvendetta/Desktop/Projects/House Price Prediction/MHP/server/server.py"
Starting Python Flask Server For Melbourne House Price Prediction...
loading saved artifacts...start
/usr/local/lib/python3.9/site-packages/sklearn/base.py:310: UserWarning: Trying to unpickle estimator DecisionTreeRegressor
warnings.warn(
/usr/local/lib/python3.9/site-packages/sklearn/base.py:310: UserWarning: Trying to unpickle estimator RandomForestRegressor
warnings.warn(
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
loading saved artifacts...done
* Serving Flask app "server" (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: off

```

5. Configuring the client html, css and js stack:

A simple yet effective stack of html, css and javascript was used for the front-end. The application simply asks the user for some parameters such as distance from CBD, name of the suburb, etc. and passes them to the deserialized model which then predicts the price which is finally displayed to the user.



6. End Result:

The Flask website looks like the following and needs further optimization of the ML model used as well as more functionality which could be added later. Also, the next step would be to try to deploy this on docker or an AWS instance.

 A screenshot of a web application interface for estimating house prices. It features a blue header and a light blue background with a blurred image of a house. The form includes:

- Distance from CBD:** A text input field containing the value '10'.
- Number of bedrooms:** A text input field containing the value '3'.
- Type of Property:** A dropdown menu with 'type_u' selected.
- Location:** A dropdown menu with 'suburb_caulfield' selected.
- Estimate Price:** A green button to submit the form.
- Result:** A yellow box displaying the estimated price: '899831.52 AUD'.