

Step 1: Defining the model.

```
In [59]: from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,random_state=10)
lr_clf = LinearRegression()
```


Step 2: Fitting the model.

```
In [60]: lr_clf.fit(X_train,y_train)
lr_clf.score(X_test,y_test)
```

Out[60]: 0.8863810189682368

Step 3: Exporting the model to be utilize in development

```
In [65]: import pickle
with open('ph_home_prices_model.pickle','wb') as f:
    pickle.dump(lr_clf,f)
```

 ph_home_prices_model.pickle

Step 4: Using K Fold cross validation to measure accuracy of our LinearRegression model.


```
In [61]: from sklearn.model_selection import ShuffleSplit
from sklearn.model_selection import cross_val_score

cv = ShuffleSplit(n_splits=5, test_size=0.2, random_state=0)

cross_val_score(LinearRegression(), X, y, cv=cv)
```

Out[61]: array([0.88716775, 0.80669104, 0.79492189, 0.82527956, 0.84929423])

Step 5: Importing the model That was created using python in Jupyter notebook.

 ph_home_prices_model.pickle

Step 6: Setting up the Python flask server to be able read the home prices model.

```
from flask import Flask, request, jsonify
import util

app = Flask(__name__)

@app.route('/get_location_names', methods=['GET'])
def get_location_names():
    response = jsonify({
        'locations': util.get_location_names()
    })
    response.headers.add('Access-Control-Allow-Origin', '*')

    return response

@app.route('/predict_home_price', methods=['GET', 'POST'])
def predict_home_price():
    total_sqft = float(request.form['total_sqft'])
    location = request.form['location']
    bhk = int(request.form['bhk'])
    bath = int(request.form['bath'])

    response = jsonify({
        'estimated_price': util.get_estimated_price(location, total_sqft, bhk, bath)
    })
    response.headers.add('Access-Control-Allow-Origin', '*')

    return response

if __name__ == "__main__":
    print("Starting Python Flask Server For Home Price Prediction...")
    util.load_saved_artifacts()
    app.run()
```

Step 7: Setting up the utilities in order to get the data endpoints from the model

- Loading the saved artifacts

```
def load_saved_artifacts():
    print("loading saved artifacts...start")
    global __data_columns
    global __locations

    with open("./artifacts/columns.json", "r") as f:
        __data_columns = json.load(f)['data_columns']
        __locations = __data_columns[3:] # first 3 columns are sqft, bath, bhk

    global __model
    if __model is None:
        with open('./artifacts/ph_home_prices_model.pickle', 'rb') as f:
            __model = pickle.load(f)
    print("loading saved artifacts...done")
```

- Getting estimated prices

```
def get_estimated_price(location, sqft, bhk, bath):
    try:
        loc_index = __data_columns.index(location.lower())
    except:
        loc_index = -1

    x = np.zeros(len(__data_columns))
    x[0] = sqft
    x[1] = bath
    x[2] = bhk
    if loc_index >= 0:
        x[loc_index] = 1

    return round(__model.predict([x])[0], 2)
```

- Getting location names and data columns

```
def get_location_names():
    return __locations

def get_data_columns():
    return __data_columns

if __name__ == '__main__':
    load_saved_artifacts()
    print(get_location_names())
    print(get_estimated_price('Davao city', 600, 3, 3))
    print(get_estimated_price('Alaminos', 800, 3, 3))
```

Step 8: Running the flask server

```
Kenneth@DESKTOP-S700130 MINGW64 ~/Documents/Jupyter-Repository/_CS1401P/server
$ python server.py
Starting Python Flask Server For Home Price Prediction...
loading saved artifacts...start
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
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Step 9: Connecting the endpoint to JavaScript to be able to use for web development.

```
function onPageLoad() {
    console.log( "document loaded" );
    var url = "http://127.0.0.1:5000/";
    $.get(url,function(data, status) {
        console.log("got response for get_location_names request");
        if(data) {
            var locations = data.locations;
            var uiLocations = document.getElementById("uiLocations");
            $('#uiLocations').empty();
            for(var i in locations) {
                var opt = new Option(locations[i]);
                $('#uiLocations').append(opt);
            }
        }
    });
}

window.onload = onPageLoad;
```