[1]:
```
import pandas as pd
```

[2]:
```
data_set = pd.read_csv('Real_Estate.csv')
```

[3]:
```
data_set
```

[3]:

	Transaction date	House age	Distance to the nearest MRT station	Number of convenience stores	Latitude	Longitude	House price of unit area
0	2012-09-02 16:42:30.519336	13.3	4082.01500	8	25.007059	121.561694	6.488673
1	2012-09-04 22:52:29.919544	35.5	274.01440	2	25.012148	121.546990	24.970725
2	2012-09-05 01:10:52.349449	1.1	1978.67100	10	25.003850	121.528336	26.694267
3	2012-09-05 13:26:01.189083	22.2	1055.06700	5	24.962887	121.482178	38.091638
4	2012-09-06 08:29:47.910523	8.5	967.40000	6	25.011037	121.479946	21.654710
...	...	...	...	...	...	...	...
409	2013-07-25 15:30:36.565239	18.3	170.12890	6	24.981186	121.486798	29.096310
410	2013-07-26 17:16:34.019780	11.9	323.69120	2	24.950070	121.483918	33.871347
411	2013-07-28 21:47:23.339050	0.0	451.64190	8	24.963901	121.543387	25.255105
412	2013-07-29 13:33:29.405317	35.9	292.99780	5	24.997863	121.558286	25.285620
413	2013-08-01 09:49:41.506402	12.0	90.45606	6	24.952904	121.526395	37.580554

414 rows × 7 columns

[4]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error,r2_score
```

[5]:
```
features = ['Distance to the nearest MRT station','Number of convenience stores','Latitude','Longitude']
target = ['House price of unit area']

x = data_set[features]
y = data_set[target]
```

[6]:
```
x
```

[6]:

	Distance to the nearest MRT station	Number of convenience stores	Latitude	Longitude
0	4082.01500	8	25.007059	121.561694
1	274.01440	2	25.012148	121.546990
2	1978.67100	10	25.003850	121.528336
3	1055.06700	5	24.962887	121.482178
4	967.40000	6	25.011037	121.479946
...	...	...	...	...
409	170.12890	6	24.981186	121.486798
410	323.69120	2	24.950070	121.483918
411	451.64190	8	24.963901	121.543387
412	292.99780	5	24.997863	121.558286
413	90.45606	6	24.952904	121.526395

414 rows × 4 columns

[7]:
```
y
```

[7]:

	House price of unit area
0	6.488673
1	24.970725
2	26.694267
3	38.091638
4	21.654710
...	...
409	29.096310
410	33.871347
411	25.255105
412	25.285620
413	37.580554

414 rows × 1 columns

[8]:
```
# Splitting dataset into training and test data
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=42)
```

[9]:
```
# model initialization
model = LinearRegression()

# Train the model
model.fit(x_train,y_train)
```

[9]:
```
▾ LinearRegression
LinearRegression()
```

[10]:
```
import dash
from dash import html, dcc, Input, Output, State
import pandas as pd

# Initialize the Dash app
app = dash.Dash(__name__)

# Define the layout of the app
app.layout = html.Div([
    html.Div([
        html.H1("Real Estate Price Prediction", style={'text-align': 'center'}),

        html.Div([
            dcc.Input(id='distance_to_mrt', type='number', placeholder='Distance to MRT Station (meters)',
                      style={'margin': '10px', 'padding': '10px'}),
            dcc.Input(id='num_convenience_stores', type='number', placeholder='Number of Convenience Stores',
                      style={'margin': '10px', 'padding': '10px'}),
            dcc.Input(id='latitude', type='number', placeholder='Latitude',
                      style={'margin': '10px', 'padding': '10px'}),
            dcc.Input(id='longitude', type='number', placeholder='Longitude',
                      style={'margin': '10px', 'padding': '10px'}),
            html.Button('Predict Price', id='predict_button', n_clicks=0,
                        style={'margin': '10px', 'padding': '10px', 'background-color': '#007BFF', 'color': 'white'}),
        ], style={'text-align': 'center'}),

        html.Div(id='prediction_output', style={'text-align': 'center', 'font-size': '20px', 'margin-top': '20px'})
    ], style={'width': '50%', 'margin': '0 auto', 'border': '2px solid #007BFF', 'padding': '20px', 'border-radius': '10px'})
])
```

```python
# Define callback to update output
@app.callback(
    Output('prediction_output', 'children'),
    [Input('predict_button', 'n_clicks')],
    [State('distance_to_mrt', 'value'),
     State('num_convenience_stores', 'value'),
     State('latitude', 'value'),
     State('longitude', 'value')]
)
def update_output(n_clicks, distance_to_mrt, num_convenience_stores, latitude, longitude):
    if n_clicks > 0 and all(v is not None for v in [distance_to_mrt, num_convenience_stores, latitude, longitude]):
        # Prepare the feature vector
        features = pd.DataFrame([[distance_to_mrt, num_convenience_stores, latitude, longitude]],
                                columns=['distance_to_mrt', 'num_convenience_stores', 'latitude', 'longitude'])
        # Predict
        prediction = model.predict(features)[0]
        return f'Predicted House Price of Unit Area: {prediction:.2f}'
    elif n_clicks > 0:
        return 'Please enter all values to get a prediction'
    return ''

# Run the app
if __name__ == '__main__':
    app.run_server(debug=True)
```