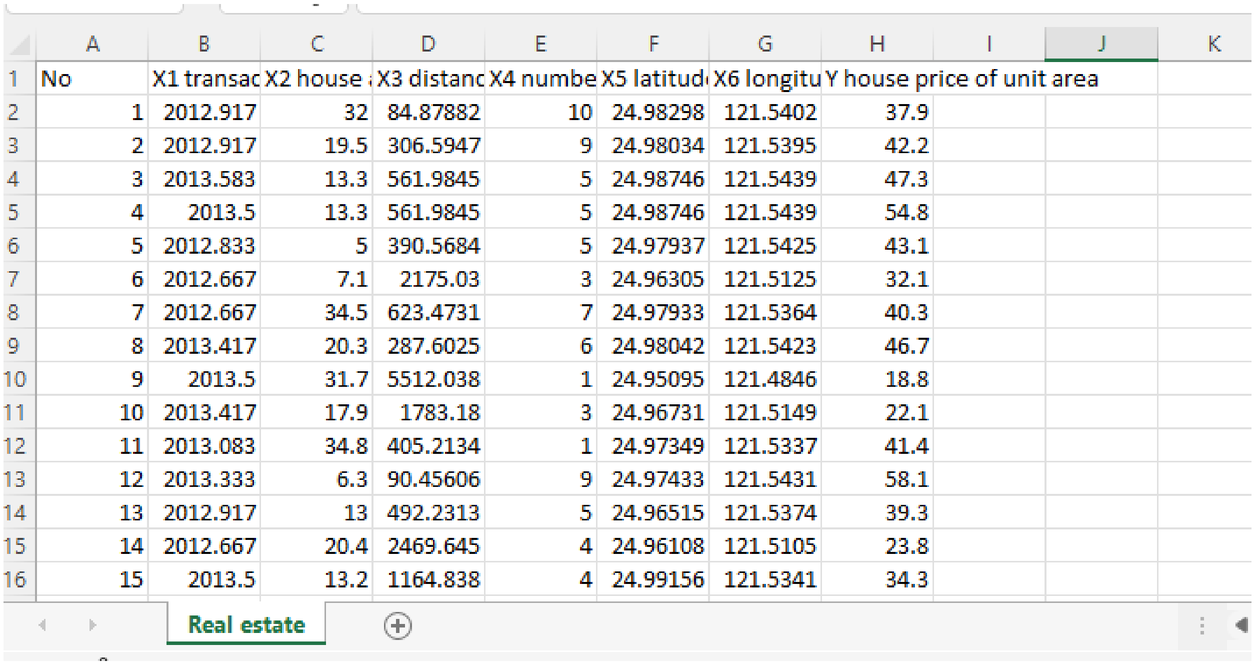
Week 4 Flask Development:

Name: Shanka Attanayake

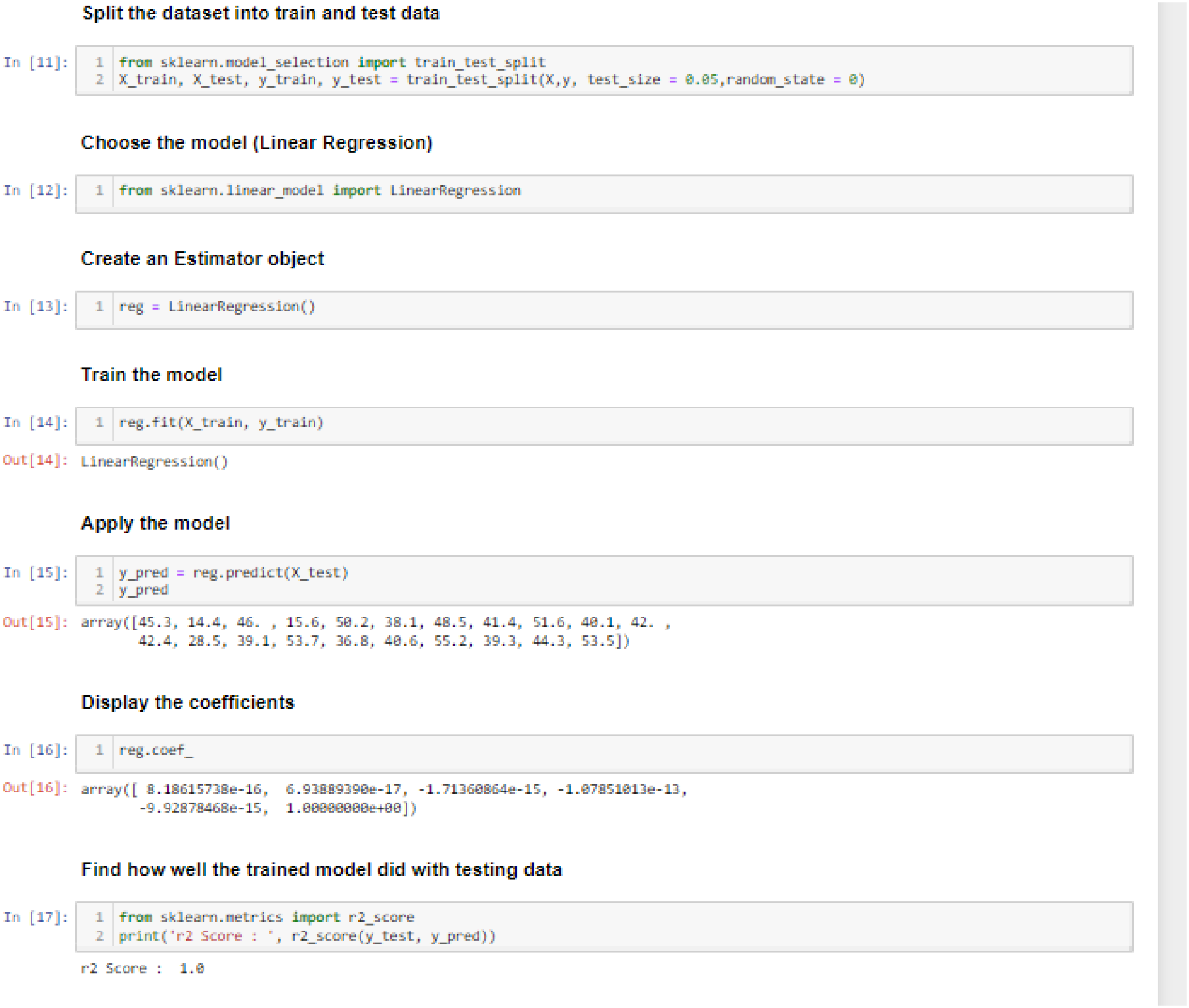
Dataset is of a House price dataset where it will be used be create a linear regression model to predict house prices.

Dataset:



	A	B	C	D	E	F	G	H	I	J	K
1	No	X1 transac	X2 house a	X3 distanc	X4 numbe	X5 latitud	X6 longitu	Y house price of unit area			
2	1	2012.917	32	84.87882	10	24.98298	121.5402	37.9			
3	2	2012.917	19.5	306.5947	9	24.98034	121.5395	42.2			
4	3	2013.583	13.3	561.9845	5	24.98746	121.5439	47.3			
5	4	2013.5	13.3	561.9845	5	24.98746	121.5439	54.8			
6	5	2012.833	5	390.5684	5	24.97937	121.5425	43.1			
7	6	2012.667	7.1	2175.03	3	24.96305	121.5125	32.1			
8	7	2012.667	34.5	623.4731	7	24.97933	121.5364	40.3			
9	8	2013.417	20.3	287.6025	6	24.98042	121.5423	46.7			
10	9	2013.5	31.7	5512.038	1	24.95095	121.4846	18.8			
11	10	2013.417	17.9	1783.18	3	24.96731	121.5149	22.1			
12	11	2013.083	34.8	405.2134	1	24.97349	121.5337	41.4			
13	12	2013.333	6.3	90.45606	9	24.97433	121.5431	58.1			
14	13	2012.917	13	492.2313	5	24.96515	121.5374	39.3			
15	14	2012.667	20.4	2469.645	4	24.96108	121.5105	23.8			
16	15	2013.5	13.2	1164.838	4	24.99156	121.5341	34.3			

Building Model:



Split the dataset into train and test data

```
In [11]:
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size = 0.05,random_state = 0)
```

Choose the model (Linear Regression)

```
In [12]:
from sklearn.linear_model import LinearRegression
```

Create an Estimator object

```
In [13]:
reg = LinearRegression()
```

Train the model

```
In [14]:
reg.fit(X_train, y_train)
```

```
Out[14]: LinearRegression()
```

Apply the model

```
In [15]:
y_pred = reg.predict(X_test)
y_pred
```

```
Out[15]: array([45.3, 14.4, 46. , 15.6, 50.2, 38.1, 48.5, 41.4, 51.6, 40.1, 42. ,
               42.4, 28.5, 39.1, 53.7, 36.8, 40.6, 55.2, 39.3, 44.3, 53.5])
```

Display the coefficients

```
In [16]:
reg.coef_
```

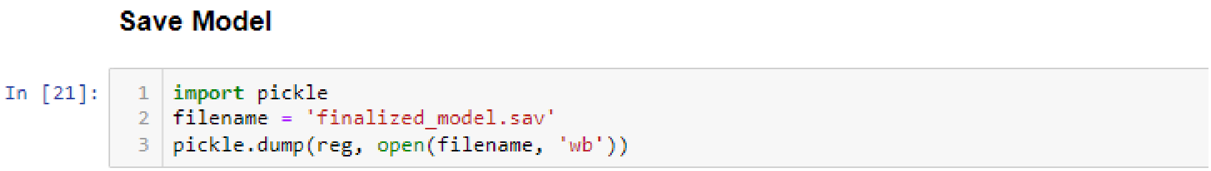
```
Out[16]: array([ 8.18615738e-16,  6.93889390e-17, -1.71360864e-15, -1.07851013e-13,
               -9.92878468e-15,  1.00000000e+00])
```

Find how well the trained model did with testing data

```
In [17]:
from sklearn.metrics import r2_score
print('r2 Score : ', r2_score(y_test, y_pred))
```

```
r2 Score :  1.0
```

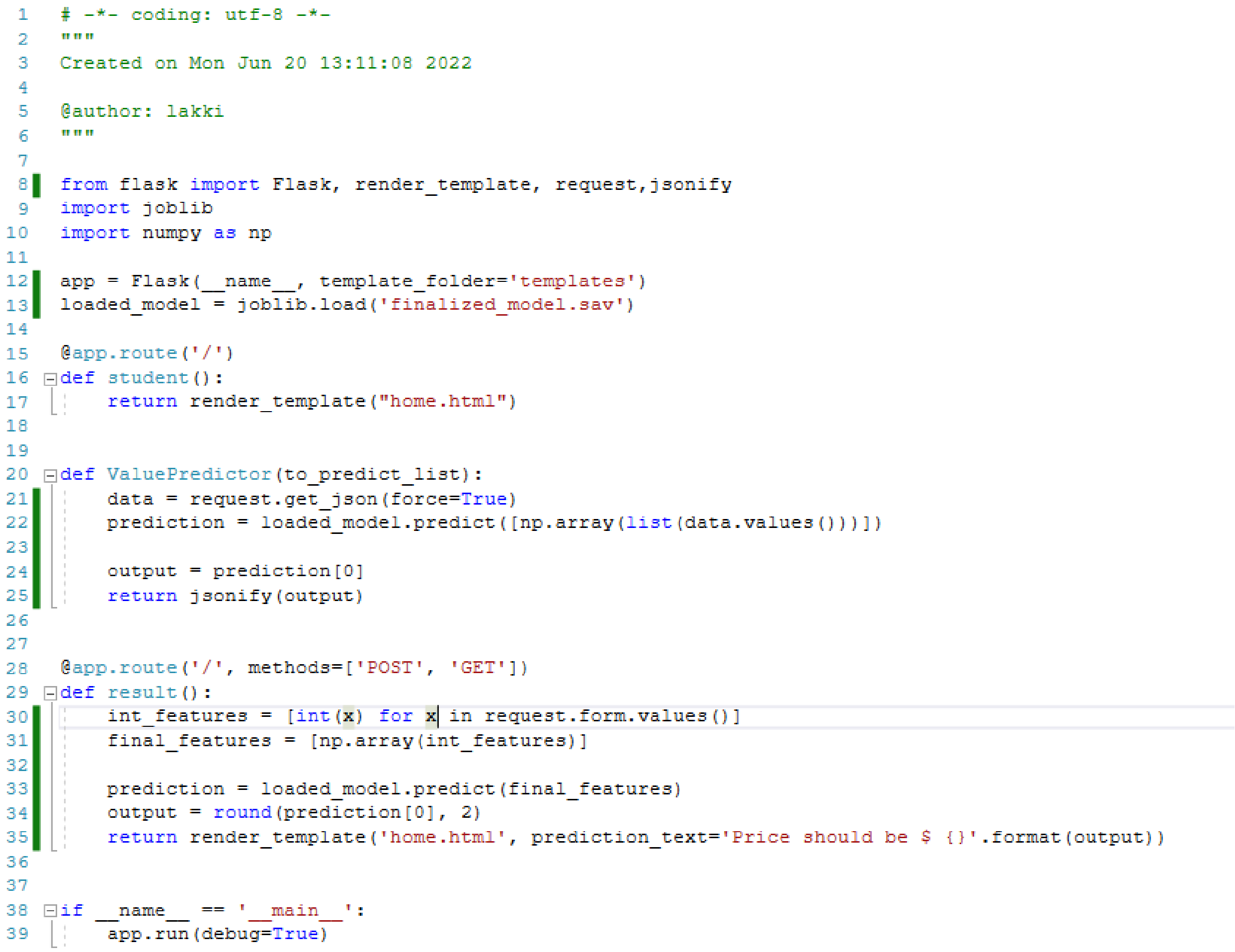
Saving model



Save Model

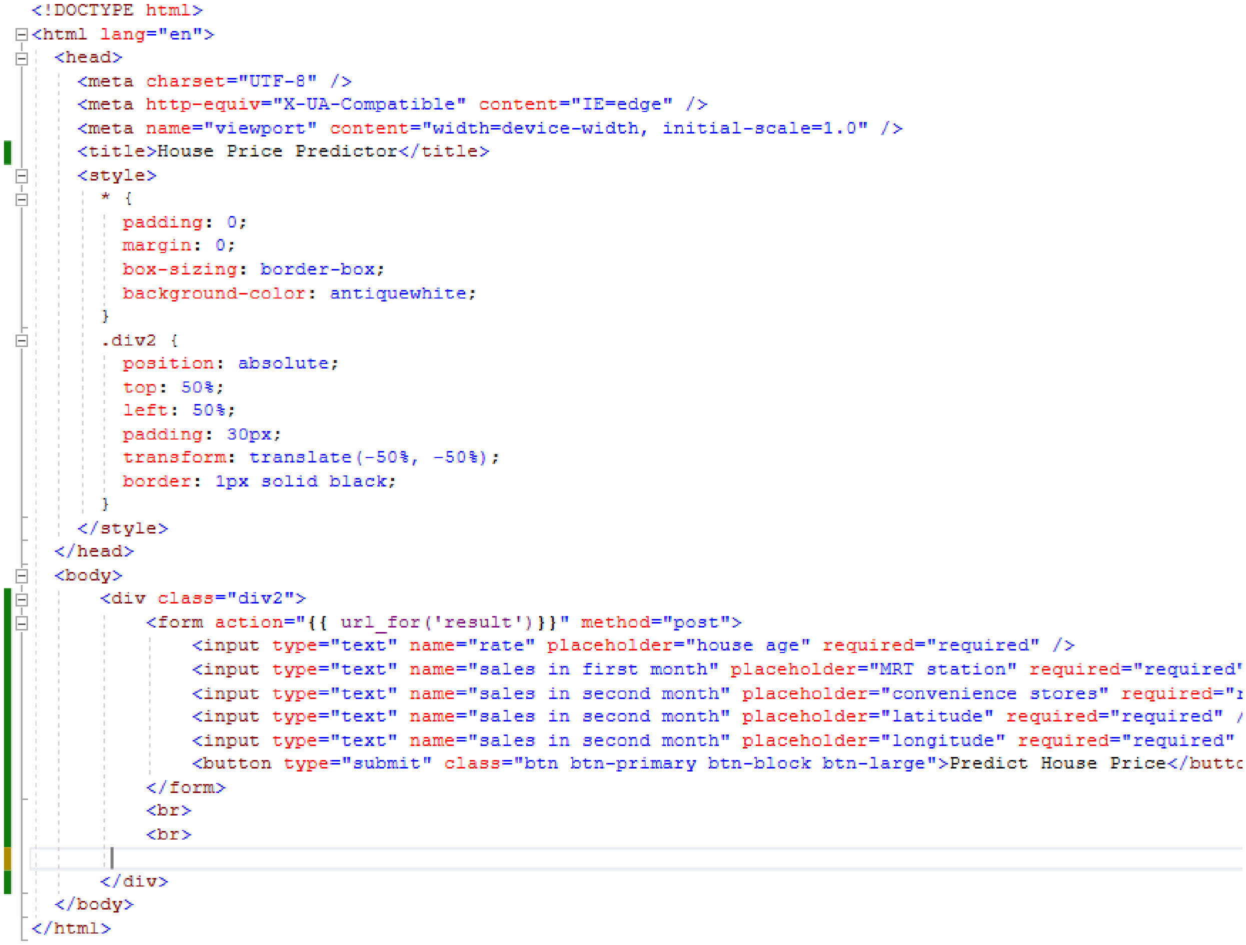
```
In [21]:
import pickle
filename = 'finalized_model.sav'
pickle.dump(reg, open(filename, 'wb'))
```

App File:



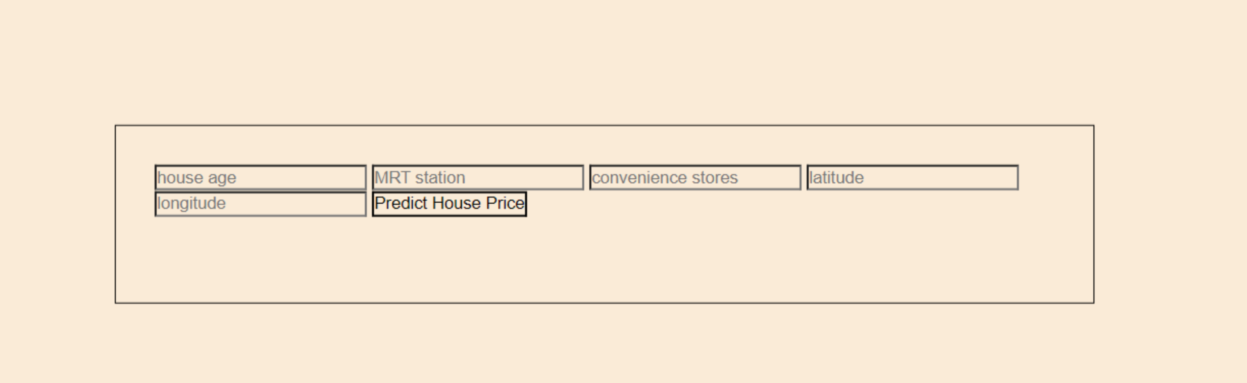
```python
# -*- coding: utf-8 -*-
"""
Created on Mon Jun 20 13:11:08 2022

@author: lakki
"""

from flask import Flask, render_template, request,jsonify
import joblib
import numpy as np

app = Flask(__name__, template_folder='templates')
loaded_model = joblib.load('finalized_model.sav')

@app.route('/')
def student():
    return render_template("home.html")


def ValuePredictor(to_predict_list):
    data = request.get_json(force=True)
    prediction = loaded_model.predict([np.array(list(data.values()))])

    output = prediction[0]
    return jsonify(output)


@app.route('/', methods=['POST', 'GET'])
def result():
    int_features = [int(x) for x in request.form.values()]
    final_features = [np.array(int_features)]

    prediction = loaded_model.predict(final_features)
    output = round(prediction[0], 2)
    return render_template('home.html', prediction_text='Price should be $ {}'.format(output))


if __name__ == '__main__':
    app.run(debug=True)
```

HTML template File

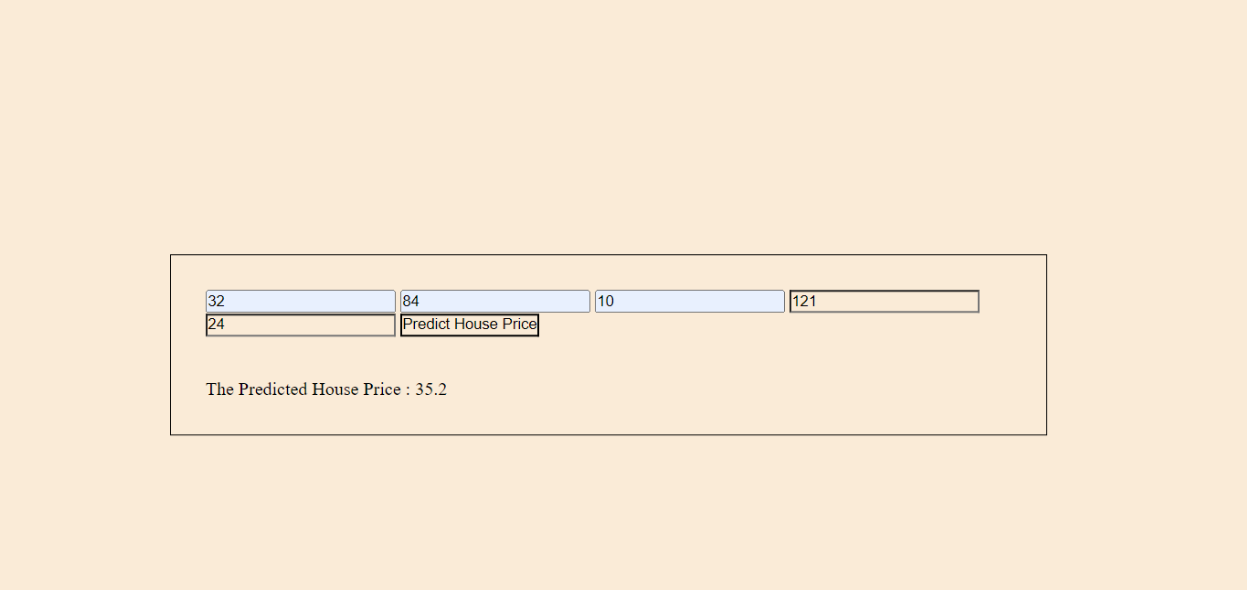


```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>House Price Predictor</title>
    <style>
      * {
        padding: 0;
        margin: 0;
        box-sizing: border-box;
        background-color: antiquewhite;
      }
      .div2 {
        position: absolute;
        top: 50%;
        left: 50%;
        padding: 30px;
        transform: translate(-50%, -50%);
        border: 1px solid black;
      }
    </style>
  </head>
  <body>
      <div class="div2">
          <form action="{{ url_for('result')}}" method="post">
              <input type="text" name="rate" placeholder="house age" required="required" />
              <input type="text" name="sales in first month" placeholder="MRT station" required="required"
              <input type="text" name="sales in second month" placeholder="convenience stores" required="r
              <input type="text" name="sales in second month" placeholder="latitude" required="required" /
              <input type="text" name="sales in second month" placeholder="longitude" required="required"
              <button type="submit" class="btn btn-primary btn-block btn-large">Predict House Price</butto
          </form>
          <br>
          <br>

      </div>
  </body>
</html>
```

Web Page:



Web Page Result after filling in features:


32
84
10
121
24

The Predicted House Price : 35.2