House Price Prediction Using Linear Regression Model

ELİF DURAN

Department of Computer Engineering, Eskişehir Technical University, Eskişehir
{elif_duran}eskisehir.edu.tr

I. INTRODUCTION

In this homework, I have implemented Linear Regression Machine Learning algorithm to predict the price of houses according to multiple features. The homework aims to find the best price for houses which is searched in terms of lot area, living area, number of floors, number of bedrooms, number of bathrooms, water front, year of building and year of renovation. I work with prices.csv file which contains all features of houses and its prices. According to information in csv file, train and test models is split with 0.2 test size and 0.8 training size. After the model is created by linear regression, the score and root mean squared error is calculated.

II. ALGORITHM

1. Linear Regression: It is a Machine Learning Algorithm which is used to predict values within continuous range. It shows a linear relationship between a dependent and one or more independent variables.

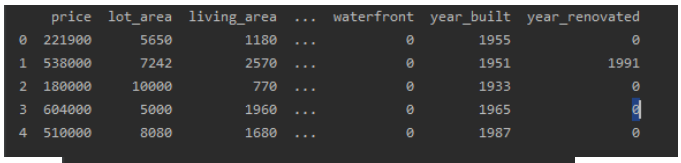
III. IMPLEMENTATION

The steps which are taken by me are shown in below:

1. I have read csv file thanks to read_csv() method of pandas library.
2. I have used head() method to show first our datas which come from csv file.
3. I have used info() method to get a concise summary of the data frame.
4. I have defined independent features of houses in X variable.
5. I have defined dependent feature of houses which is price as a Y variable.
6. After I have defined X and Y variables, I have split data into train and test sets thanks to train_test_split() method which is provided by scikit-learn. I have defined test size 0.2 to use 0.8 of dataset as a training dataset.
7. Then I have built Linear Regression model and fit X_train and Y_train datas thanks to fit() method of scikit-learn.
8. Then I have found prediction of X_test thanks to Linear Regression model's methods.
9. I have calculates score of model with X _test and Y_test.
10. In the end, I have calculated root mean squared error value thanks to metrics of scikit-learn and numpy libraries. I have calculated mean squared error with scikit-learn and got its square with numpy library.

IV. RESULTS

After we run prediction.py, we should see dataset, the information of csv file, score and root mean squared error value as an output.



	price	lot_area	living_area	...	waterfront	year_built	year_renovated
0	221900	5650	1180	...	0	1955	0
1	538000	7242	2570	...	0	1951	1991
2	180000	10000	770	...	0	1933	0
3	604000	5000	1960	...	0	1965	0
4	510000	8080	1680	...	0	1987	0

```
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   price           21613 non-null  int64
 1   lot_area        21613 non-null  int64
 2   living_area     21613 non-null  int64
 3   num_floors      21613 non-null  float64
 4   num_bedrooms    21613 non-null  int64
 5   num_bathrooms   21613 non-null  float64
 6   waterfront      21613 non-null  int64
 7   year_built      21613 non-null  int64
 8   year_renovated  21613 non-null  int64
dtypes: float64(2), int64(7)
memory usage: 1.5 MB
None
Score :  0.5614759334807018
Root Mean Squared Error :  237874.96170010715
```

V. REQUIREMENTS

It should have csv fie to initialize features. It has to install scikit-learn, numpy and pandas libraries.