Institute of Technology of Cambodia

Department of Applied Mathematics and Statistics

Project: Condo for Sale in Cambodia

Subject: Programming for Data Science

Lecturer: CHAN Sophal

Group 5:

1. SAO Samarth e20200084
2. SET Sophy e20201576
3. SANG Rithpork e20200232
4. SOK Sreyseey e20201226
5. SONG Phalla e20200439

Contents

1. Project Description and Objective
2. Dataset and Variable Description
3. Data Preprocessing
4. Exploratory Data Analysis
5. Feature Engineering
6. Model Selection
7. Decision Making

1. Project Description and Objective

Condo for Sale in Cambodia is to provide a high-quality, affordable, and convenient living space for individuals and families. And we choose this topic because we want to use some techniques to interpret and make prediction price of condo. We will study on the variable that make effect to predicting price.

2. Dataset and Variable Description

After we scraped dataset by using web scraping technique and regular expression technique. There exist 1547 rows with 15 columns. There exist 15 variables as following:

Ad ID	Category	Locations	Posted	Size(m2)	Bedroom	Bathroom	Link	Title	Price	Post Description	Sub Location	Bedrooms	Bathrooms	Floor
8538993	Condo for Sale	Phnom Penh	06-Jul-23	34	NaN	NaN	https://www.khmer24.com/en/property/new-condo-...	New condo for sell	$58,000	Very special promotion!! Full price $58,000 bu...	Toul Kork	1.0	1.0	6.0
8517524	Condo for Sale	Phnom Penh	06-Jul-23	58	NaN	NaN	https://www.khmer24.com/en/property/vista-cond...	Vista Condo Aeon2 Urgent	$75,000	ខុនដូសម្រាប់លក់ VISTA Condo\n\r\n(English Belo...	Sen Sok	NaN	NaN	19.0
9617339	Condo for Sale	Phnom Penh	03-Jul-23	34	NaN	NaN	https://www.khmer24.com/en/property/condo-uk-5...	Condo UK 548 for Sales	$42,000	បន្ទប់ខុនដូប្រភេទ Studio (UK Condo 548)\n\rកណ្...	NaN	NaN	NaN	15.0
9617339	Condo for Sale	Phnom Penh	03-Jul-23	34	NaN	NaN	https://www.khmer24.com/en/property/condo-%E1%...	Condo សម្រាប់ជួល/Condo for Rent (The Peak Priv...	$800	Property Code: VBRE00586\n\r\nCondo សម្រាប់ជួល...	Chamkar Mon	1.0	NaN	NaN
8134344	Condo for Sale	Phnom Penh	09-Apr-23	57	NaN	NaN	https://www.khmer24.com/en/property/condo-for-...	Condo for sale 57m2	$72,000	Fully furnished condo for sale / rent \n\nOwne...	NaN	1.0	1.0	9.0

3. Data Preprocessing

3.1 Data Cleaning

Dropped some features that not affected to predicting and building model.

```
df_new.head()
```

	Locations	Posted	Size(m2)	Price	Sub Location	Bedrooms	Bathrooms	Floor
0	Phnom Penh	06-Jul-23	34	$58,000	Toul Kork	1.0	1.0	6.0
1	Phnom Penh	06-Jul-23	58	$75,000	Sen Sok	NaN	NaN	19.0
2	Phnom Penh	03-Jul-23	34	$42,000	NaN	NaN	NaN	15.0
3	Phnom Penh	03-Jul-23	34	$800	Chamkar Mon	1.0	NaN	NaN
4	Phnom Penh	09-Apr-23	57	$72,000	NaN	1.0	1.0	9.0

3.1.1 Format Data Type

```
# Format Data type
df_new['Size(m2)']=df_new['Size(m2)'].astype(float)
df_new['Floor']=pd.to_numeric(df_new['Floor'], errors='coerce')
df_new['Price']=df_new['Price'].str.replace('$','').str.replace(',','').astype(float)
```

3.1.2 Checking missing values

```
Locations          0
Posted             0
Size(m2)           0
Price              0
Sub Location     931
Bedrooms         766
Bathrooms       1025
Floor            864
```

```
Sub Location 60.181 % Missing Values
Bedrooms 49.515 % Missing Values
Bathrooms 66.257 % Missing Values
Floor 55.85 % Missing Values
```

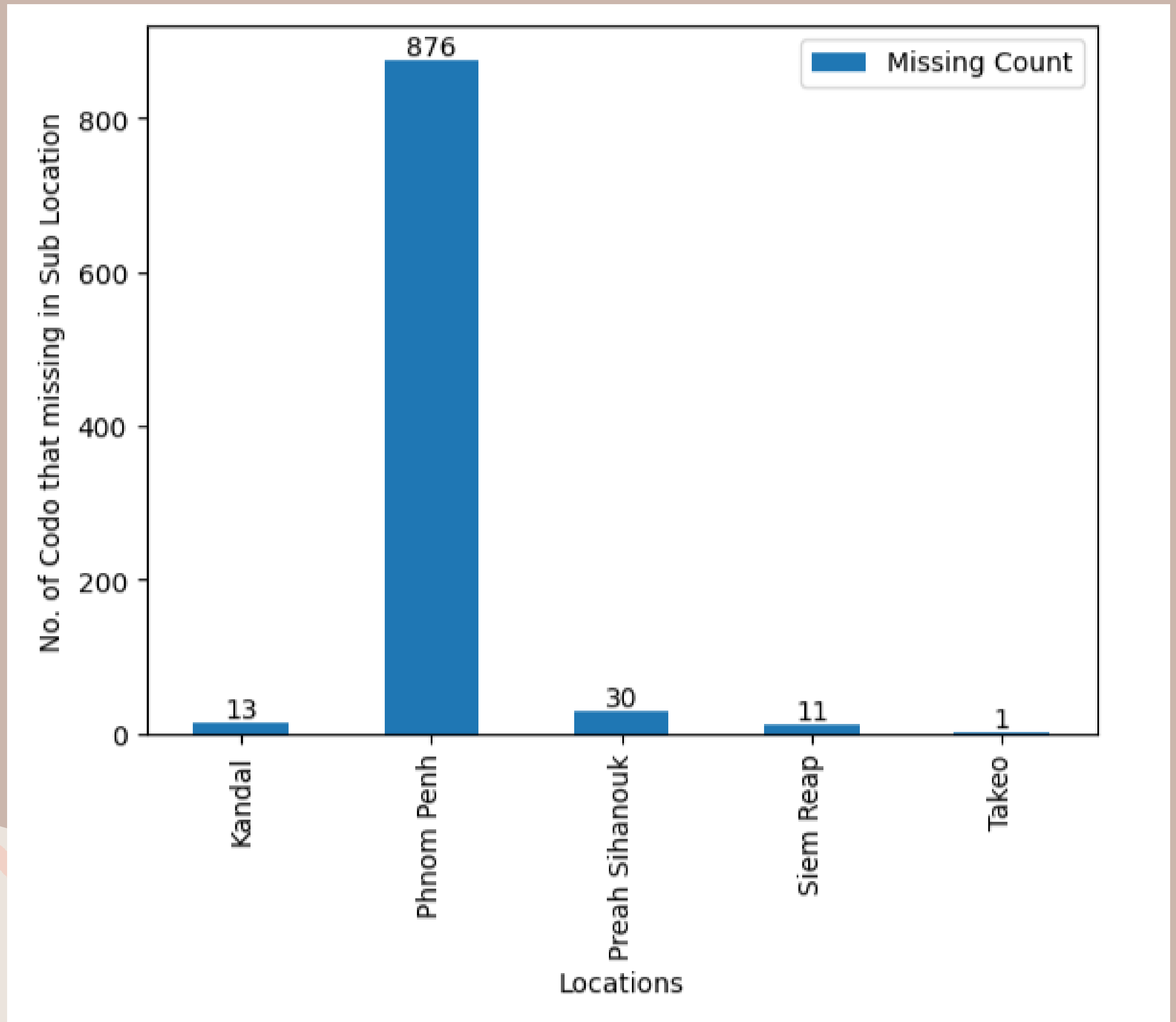
3.1.3 Checking Duplicated

```
df_new.duplicated().sum()
```

```
28
```

3.1.4 Filling missing value in feature Sub Location

Sub Location group by Location



- Based on my research, you can see that some of the sub places where most condos are in each province are:
- Phnom Penh: Chamkar Mon, Chraoy Chongvar, Tuol Kouk, Doun Penh, Saensokh
- Preah Sihanouk: Sihanoukville, Prey Nob
- Takeo: Daun Keo
- Siem Reap: Svay Dangkum, Sala Kamreuk, Slor Kram
- Kandal: Takhmao

3.1.5 Fill missing value in Feature Sub Location

```
default_sub_locations = {
    "Phnom Penh": "Chamkar Mon",
    "Preah Sihanouk": "Sihanoukville",
    "Takeo": "Daun Keo",
    "Siem Reap": "Svay Dangkum",
    "Kandal": "Takhmao"
} # create a dictionary of default sub locations
df_new["Sub Location"] = df_new["Sub Location"].fillna(df_new["Locations "].map(default_sub_locations))
```

Label Encoder: Categorical feature Location and Sub Location

	Locations	Posted	Size(m2)	Price	Sub Location	Bedrooms	Bathrooms	Floor
0	1	2023-07-06	34.0	58000.0	15	1.0	1.0	6.0
1	1	2023-07-06	58.0	75000.0	11	NaN	NaN	19.0
2	1	2023-07-03	34.0	42000.0	1	NaN	NaN	15.0
3	1	2023-07-03	34.0	800.0	1	1.0	NaN	NaN
4	1	2023-04-09	57.0	72000.0	1	1.0	1.0	9.0

3.1.7 Fixing missing values on Features Bedrooms, Bathrooms and Floor

```
import numpy as np
imputer = KNNImputer(n_neighbors=5,weights="distance")
```

```
ew3=pd.DataFrame(np.round(imputer.fit_transform(df_new2[['Locations ','Size(m2)', 'Price', 'Sub Location',
    'Bedrooms', 'Bathrooms', 'Floor']])),columns=['Locations ','Size(m2)', 'Price', 'Sub Location',
    'Bedrooms', 'Bathrooms', 'Floor'], dtype=int)
```

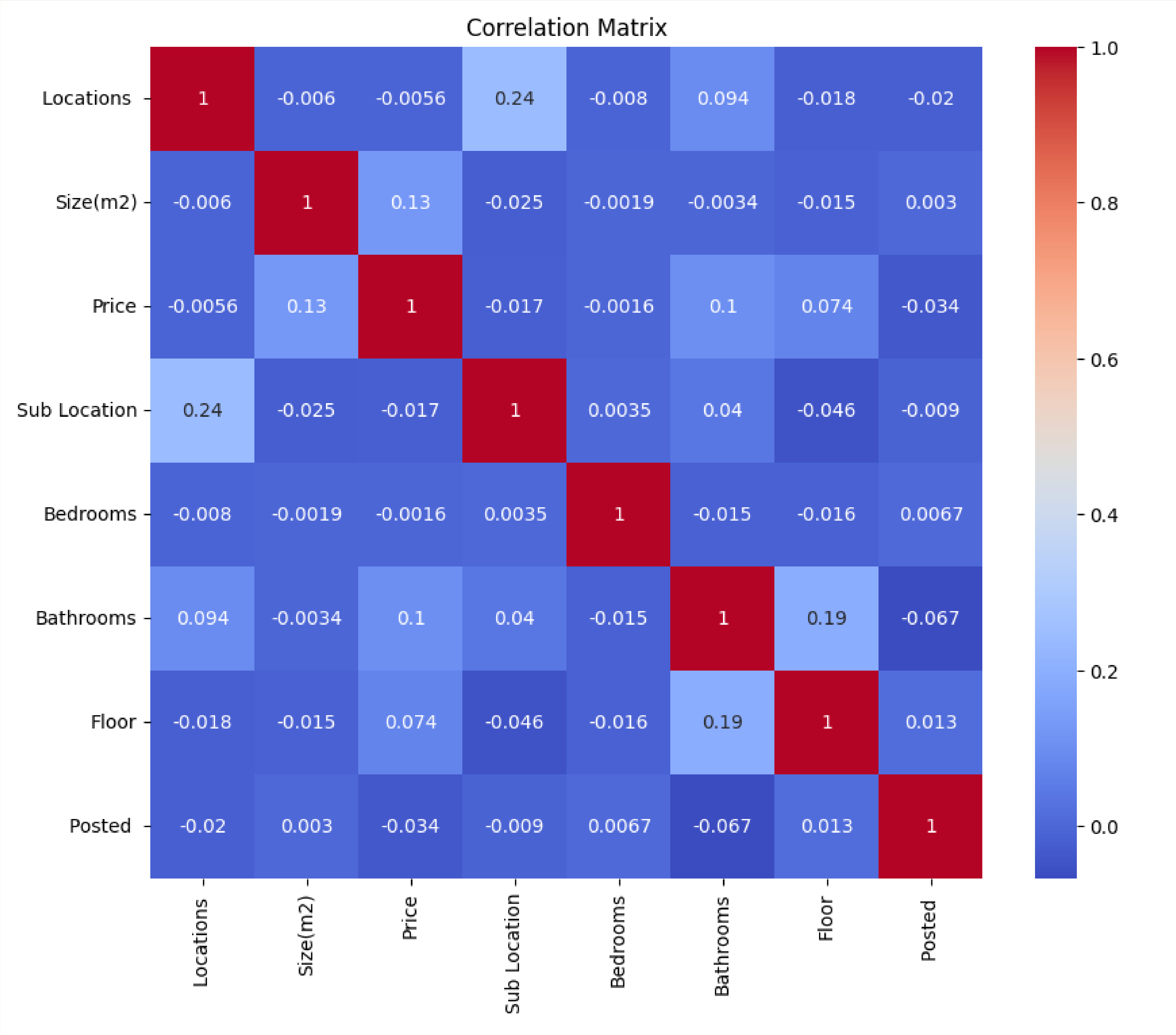
```
df_new3['Size(m2)'] = df_new3['Size(m2)'].astype(float)
df_new3['Price'] = df_new3['Price'].astype(float)
```

```
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Locations     1506 non-null   int32
 1   Size(m2)      1506 non-null   float64
 2   Price         1506 non-null   float64
 3   Sub Location  1506 non-null   int32
 4   Bedrooms      1506 non-null   int32
 5   Bathrooms     1506 non-null   int32
 6   Floor         1506 non-null   int32
dtypes: float64(2), int32(5)
```

Drop duplicate values

```
df_new1.drop_duplicates(subset=df_new1, keep='first', inplace =True)
```

Heatmap


Correlation Matrix
Locations 1 -0.006 -0.0056 0.24 -0.008 0.094 -0.018 -0.02
Size(m2) -0.006 1 0.13 -0.025 -0.0019 -0.0034 -0.015 0.003
Price -0.0056 0.13 1 -0.017 -0.0016 0.1 0.074 -0.034
Sub Location 0.24 -0.025 -0.017 1 0.0035 0.04 -0.046 -0.009
Bedrooms -0.008 -0.0019 -0.0016 0.0035 1 -0.015 -0.016 0.0067
Bathrooms 0.094 -0.0034 0.1 0.04 -0.015 1 0.19 -0.067
Floor -0.018 -0.015 0.074 -0.046 -0.016 0.19 1 0.013
Posted -0.02 0.003 -0.034 -0.009 0.0067 -0.067 0.013 1
Locations
Size(m2)
Price
Sub Location
Bedrooms
Bathrooms
Floor
Posted
1.0
0.8
0.6
0.4
0.2
0.0

3.1.9 Checking Outliers

	Size(m2)	Price	Sub Location	Bedrooms	Bathrooms	Floor
20	45.0	0.0	1	1	1	38
22	175.0	914985.0	1	4	4	26
23	50.0	84375.0	1	1	1	45
32	154.0	999000.0	0	2	3	26
39	166.0	330000.0	1	3	2	77
...	...	...	...	...	...	...
1489	127.0	323000.0	1	3	2	26
1497	47.0	68000.0	1	44	1	10
1498	354.0	488000.0	1	60	4	20
1502	298616.0	17900.0	1	2	1	8
1503	119.0	350000.0	11	3	2	24

255 rows × 6 columns

We found outliers 255 rows and 6 columns by using IQR.

Remove outliers.

```
# Remove outliers from the DataFrame
df_new3 = df_new3[~outliers]
df_new3.shape
```

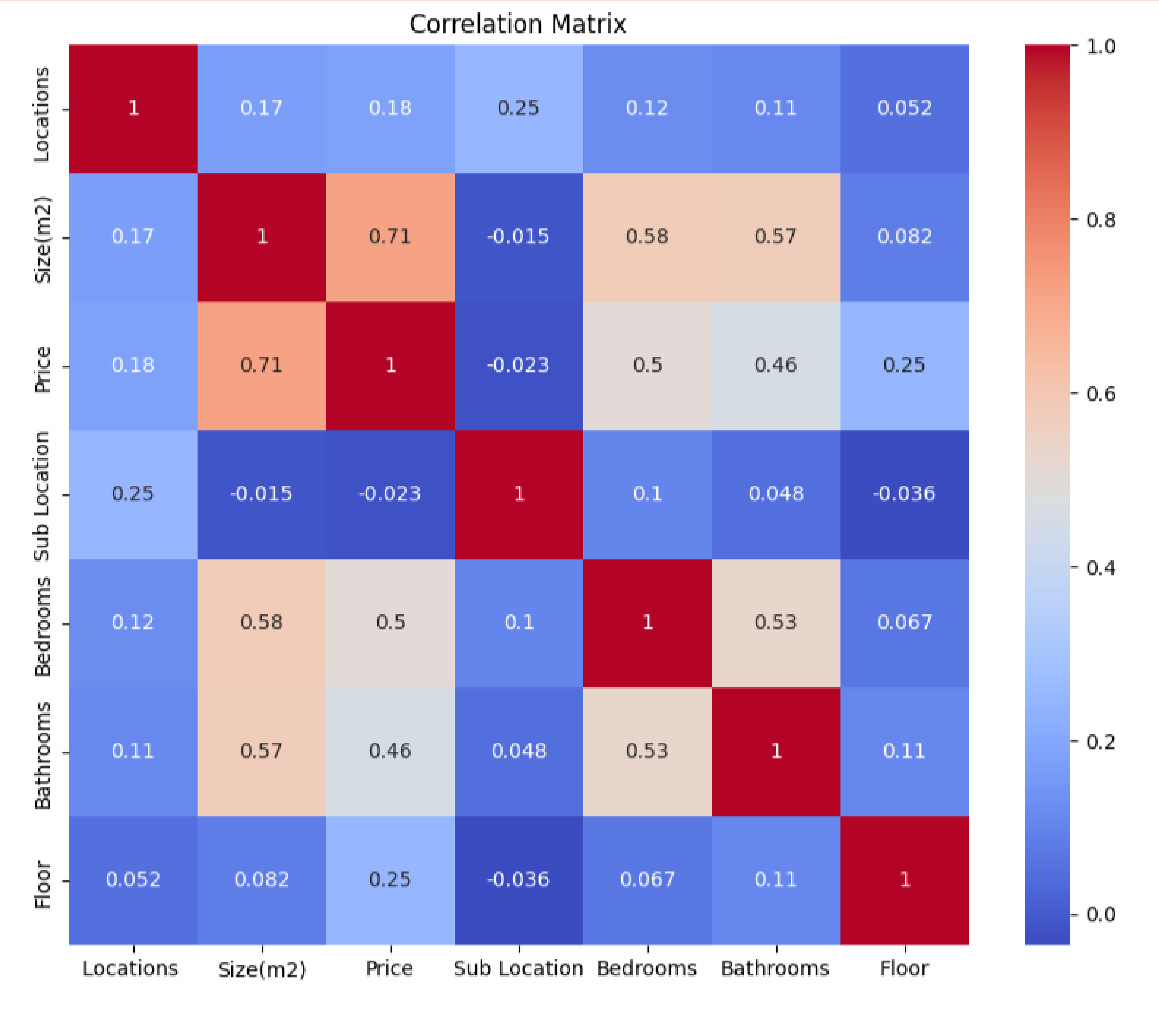
```
(1251, 8)
```

And then we decided to drop the condos that have price less than 5000.

```
df_new3 = df_new3.drop(df_new3[df_new3['Price'] < 5000].index)
```

4.Exploratory Data Analysis

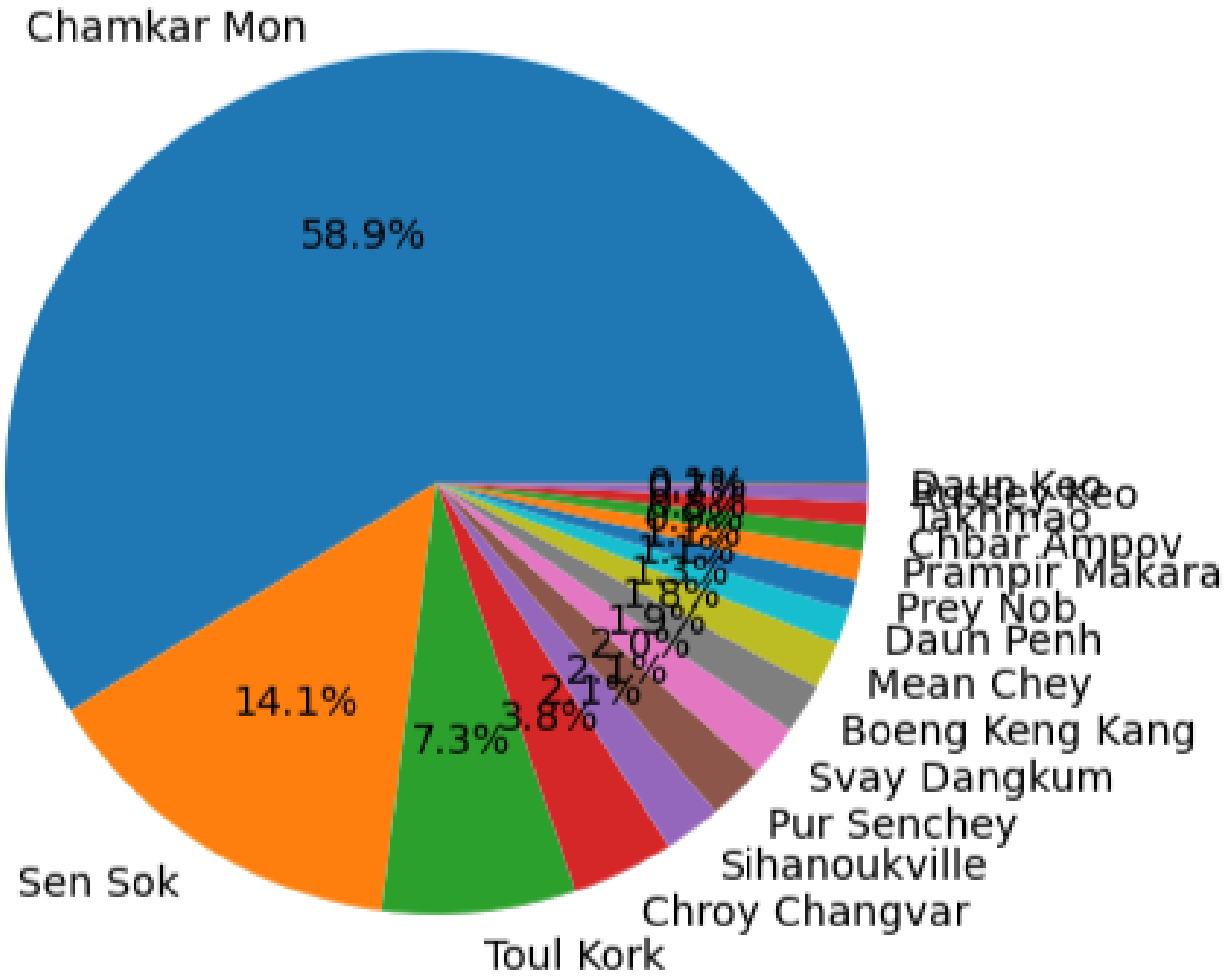
Heatmap



Sub location that has the most condo for sale is Chamkar Mon in location Phnom Penh.

Locations	Sub Location	
Kandal	Takhmao	9
Phnom Penh	Chamkar Mon	631
	Sen Sok	151
	Toul Kork	78
	Chroy Changvar	41
	Pur Senchey	23
	Boeng Keng Kang	20
	Mean Chey	19
	Daun Penh	14
	Prampir Makara	12
	Chbar Ampov	10
	Russey Keo	7
Preah Sihanouk	Sihanoukville	23
	Prey Nob	12
Siem Reap	Svay Dangkum	21
Takeo	Daun Keo	1

Pie-chat


Chamkar Mon
58.9%
Sen Sok
14.1%
Toul Kork
7.3%
3.8%
Chroy Changvar
Sihanoukville
Pur Senchey
Svay Dangkum
Boeng Keng Kang
Mean Chey
Daun Penh
Prey Nob
Prampir Makara
Chbar Ampov
Takhmao

Analyze average price base on location.

```
Price by Location:
 Locations
Kandal               29962.777778
Phnom Penh           86807.785288
Preah Sihanouk       82840.342857
Siem Reap           146414.190476
Takeo               225000.000000
```

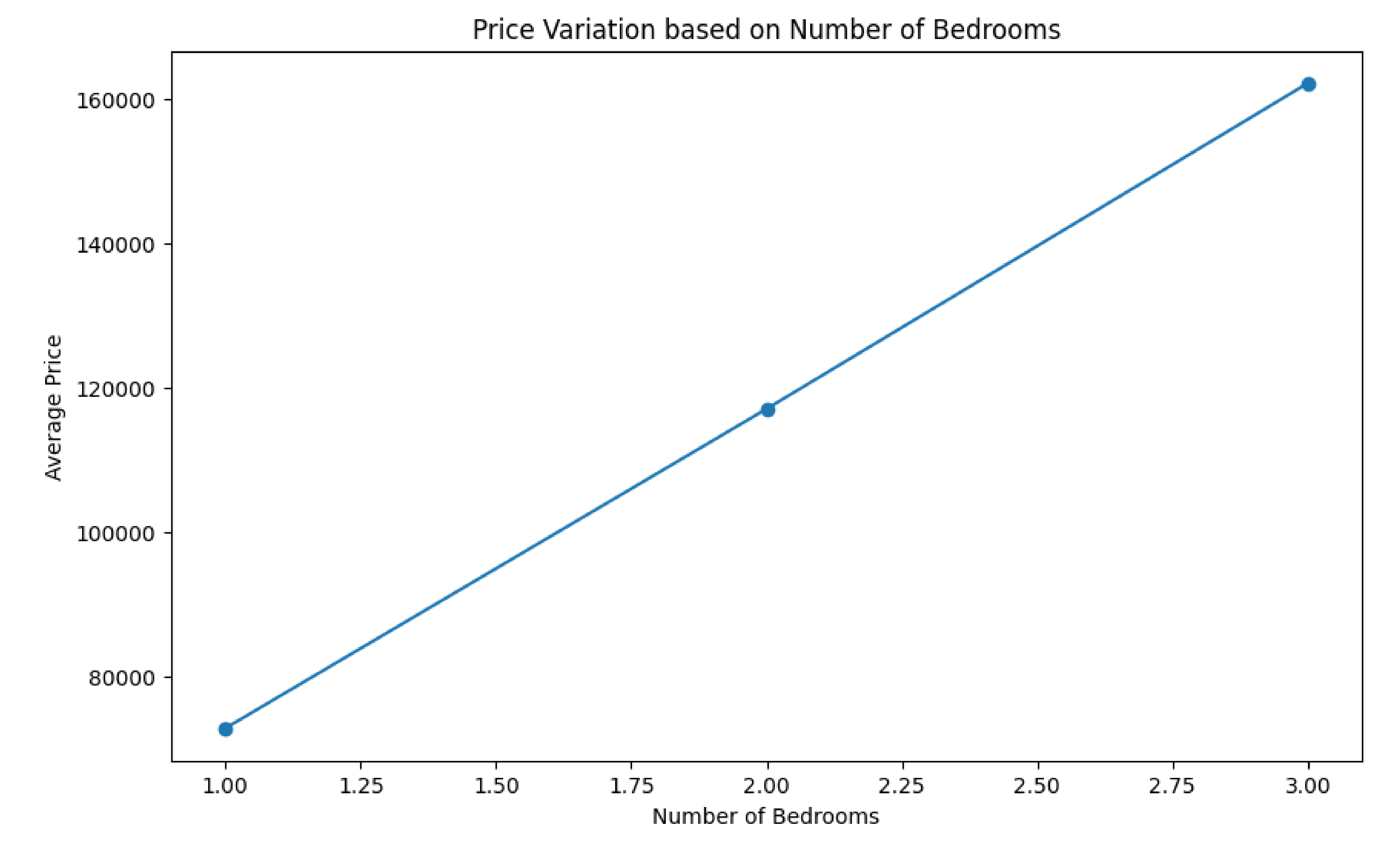
Analyze average price base on sub location.

```
Price by Sub Location:
 Sub Location
Boeng Keng Kang    100850.000000
Chamkar Mon         86680.153724
Chbar Ampov         75608.800000
Chroy Changvar      92284.390244
Daun Keo           225000.000000
Daun Penh          105792.714286
Mean Chey           78157.789474
Prampir Makara     130383.333333
Prey Nob            95624.333333
Pur Senchey         66147.782609
Russey Keo          45047.428571
Sen Sok             94555.172185
Sihanoukville       76170.434783
Svay Dangkum       146414.190476
Takhmao             29962.777778
Toul Kork           69633.961538
```

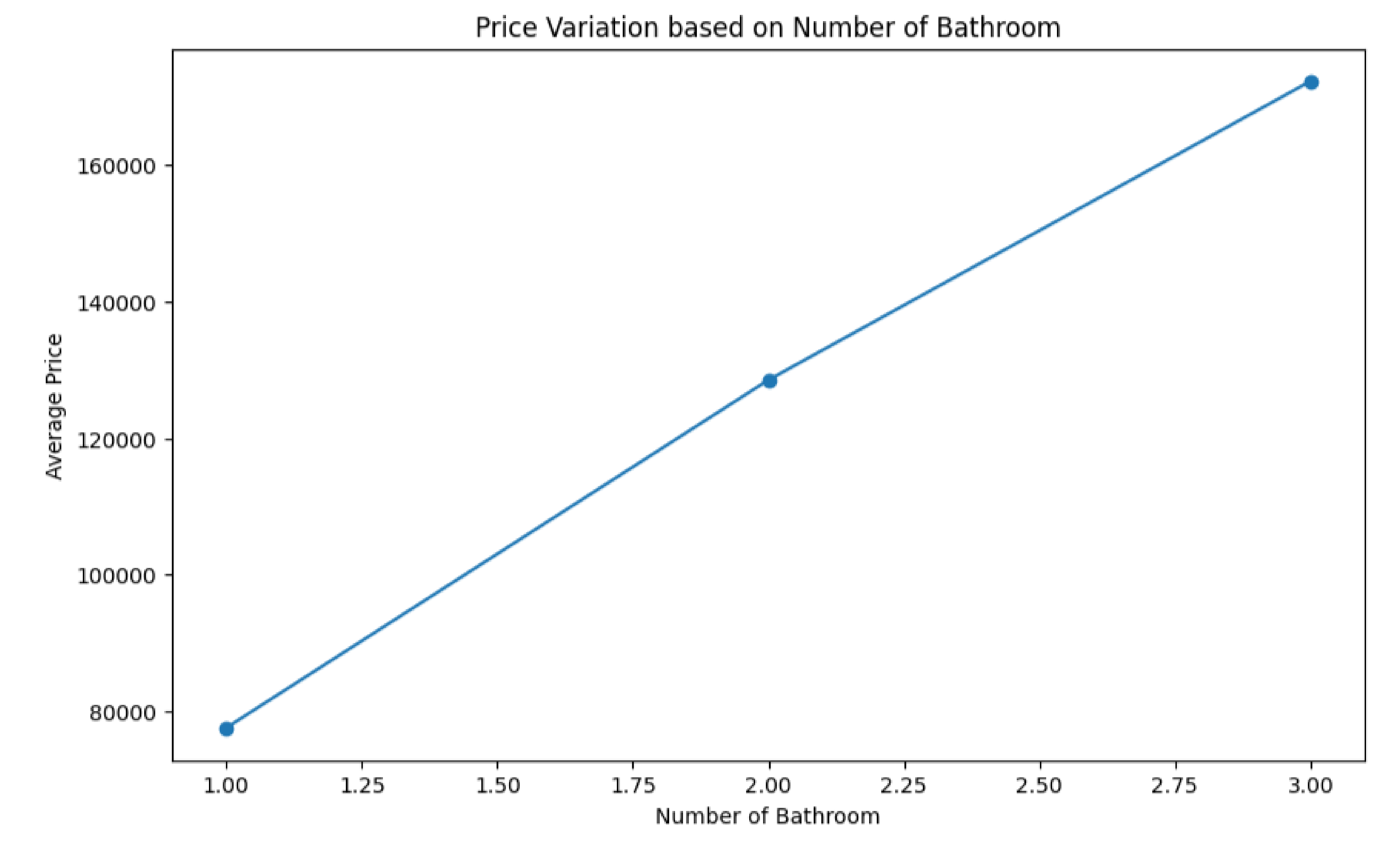
Analyze average price base on floor.

```
Average price by floor level:
 Floor
1       82910.083333
2      116238.800000
3       39423.562500
4       53521.444444
5       60932.966667
6       78249.102041
7       77925.235294
8       69979.493506
9      103633.296296
10      76441.785714
11      79324.053763
12      76538.789474
13      80385.771429
14      91191.490909
15      84509.900000
16     100460.909091
17      93367.949153
18     110694.600000
19     107668.594595
20     125886.352941
21     119452.450000
22     114025.666667
23      76560.357143
24     116414.333333
25     124199.900000
26      86181.142857
27     127840.000000
28     125149.454545
29     102988.888889
30      90372.500000
31     113600.000000
```

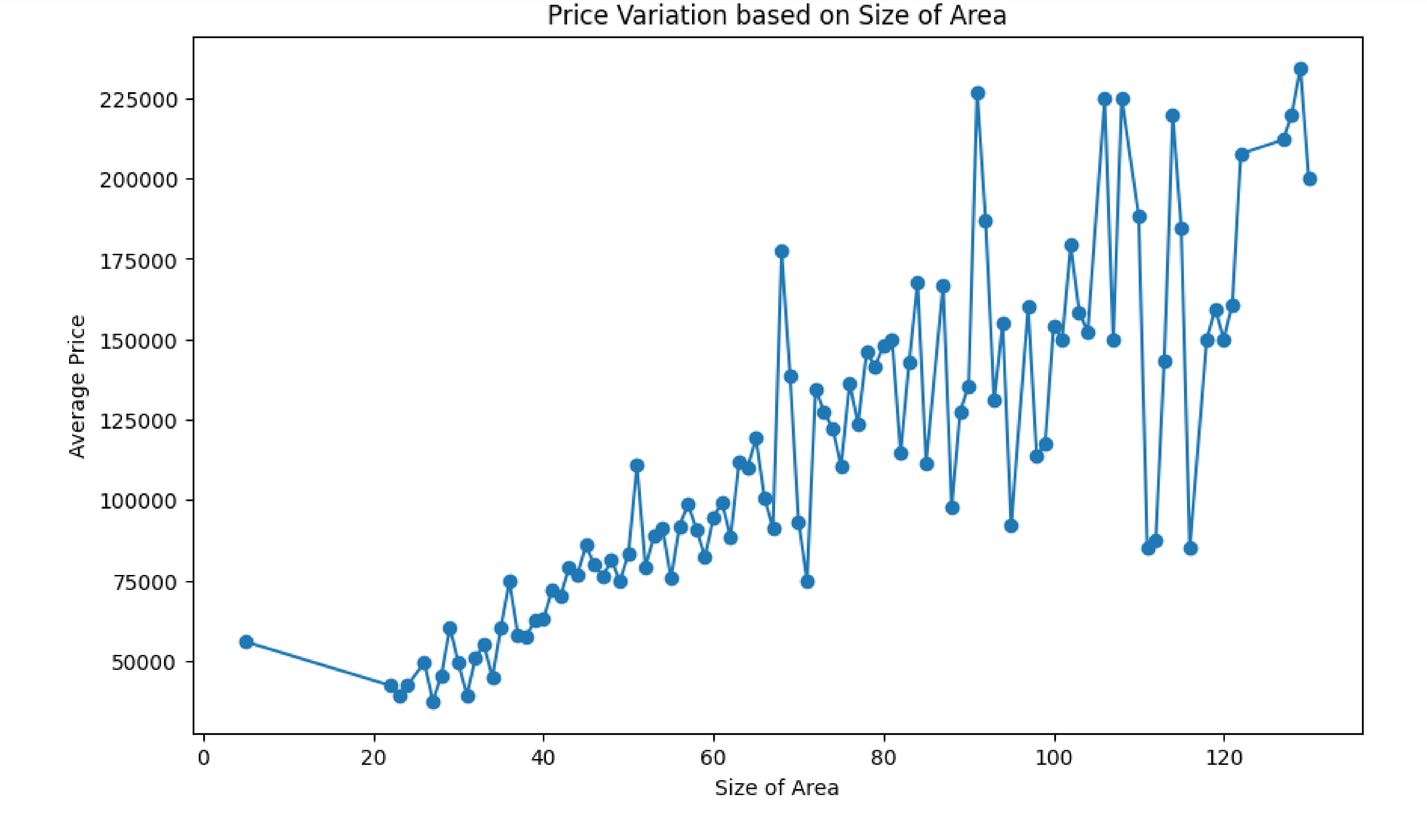
Price variation base on bedrooms.



Price variation base on bathrooms.



Price variation base on size.



5. Feature Engineering

```
from sklearn.feature_selection import SelectFromModel
from sklearn.ensemble import RandomForestRegressor
from catboost import CatBoostRegressor

X = df_new3[['Locations ', 'Size(m2)','Sub Location', 'Bedrooms',
       'Bathrooms', 'Floor']]
y = df_new3['Price']

model = RandomForestRegressor()
selector = SelectFromModel(model)
X_new = selector.fit_transform(X, y)

selected_features = X.columns[selector.get_support()]
correlation_selection = df_new3[selected_features.tolist() + ['Price']].corr()
print(correlation_selection)
```

```
          Size(m2)     Floor     Price
Size(m2)  1.000000  0.081549  0.713162
Floor     0.081549  1.000000  0.248376
Price     0.713162  0.248376  1.000000
```

We do dummy on categorical feature.

	Size(m2)	Price	Bedrooms	Bathrooms	Floor	Locations_Kandal	Locations_Phnom Penh	Locations_Preah Sihanouk	Locations_Siem Reap	Locations_Takeo	...	Sub Location_Mean Chey	Sub Location_Prampir Makara
0	34.0	58000.0	1	1	6	False	True	False	False	False	...	False	False
1	58.0	75000.0	1	1	19	False	True	False	False	False	...	False	False
2	34.0	42000.0	1	1	15	False	True	False	False	False	...	False	False
4	57.0	72000.0	1	1	9	False	True	False	False	False	...	False	False
5	33.0	59000.0	1	1	25	False	True	False	False	False	...	False	False

5 rows × 26 columns

Sub Location_Prey Nob	Sub Location_Pur Senchey	Sub Location_Russey Keo	Sub Location_Sen Sok	Sub Location_Sihanoukville	Sub Location_Svay Dangkum	Sub Location_Takhmao	Sub Location_Toul Kork
False	False	False	False	False	False	False	True
False	False	False	True	False	False	False	False
False	False	False	False	False	False	False	False
False	False	False	False	False	False	False	False
False	False	False	False	False	False	False	False

Data scaling

	Size(m2)	Price	Bedrooms	Bathrooms	Floor
0	-0.878176	-0.623373	-0.627318	-0.454026	-1.135238
1	0.189176	-0.264111	-0.627318	-0.454026	1.024581
2	-0.878176	-0.961501	-0.627318	-0.454026	0.360021
4	0.144703	-0.327510	-0.627318	-0.454026	-0.636818
5	-0.922649	-0.602240	-0.627318	-0.454026	2.021420
...	...	...	...	...	...
1496	-1.056068	-1.005880	-0.627318	-0.454026	-0.802958
1499	-0.433446	-0.581107	-0.627318	-0.454026	-0.304538
1500	-0.967122	-1.574358	-0.627318	-0.454026	-0.802958
1504	-0.967122	-1.637757	-0.627318	-0.454026	0.692301
1505	-0.967122	-0.591673	-0.627318	-0.454026	0.193882

Select and Split data to training and testing.

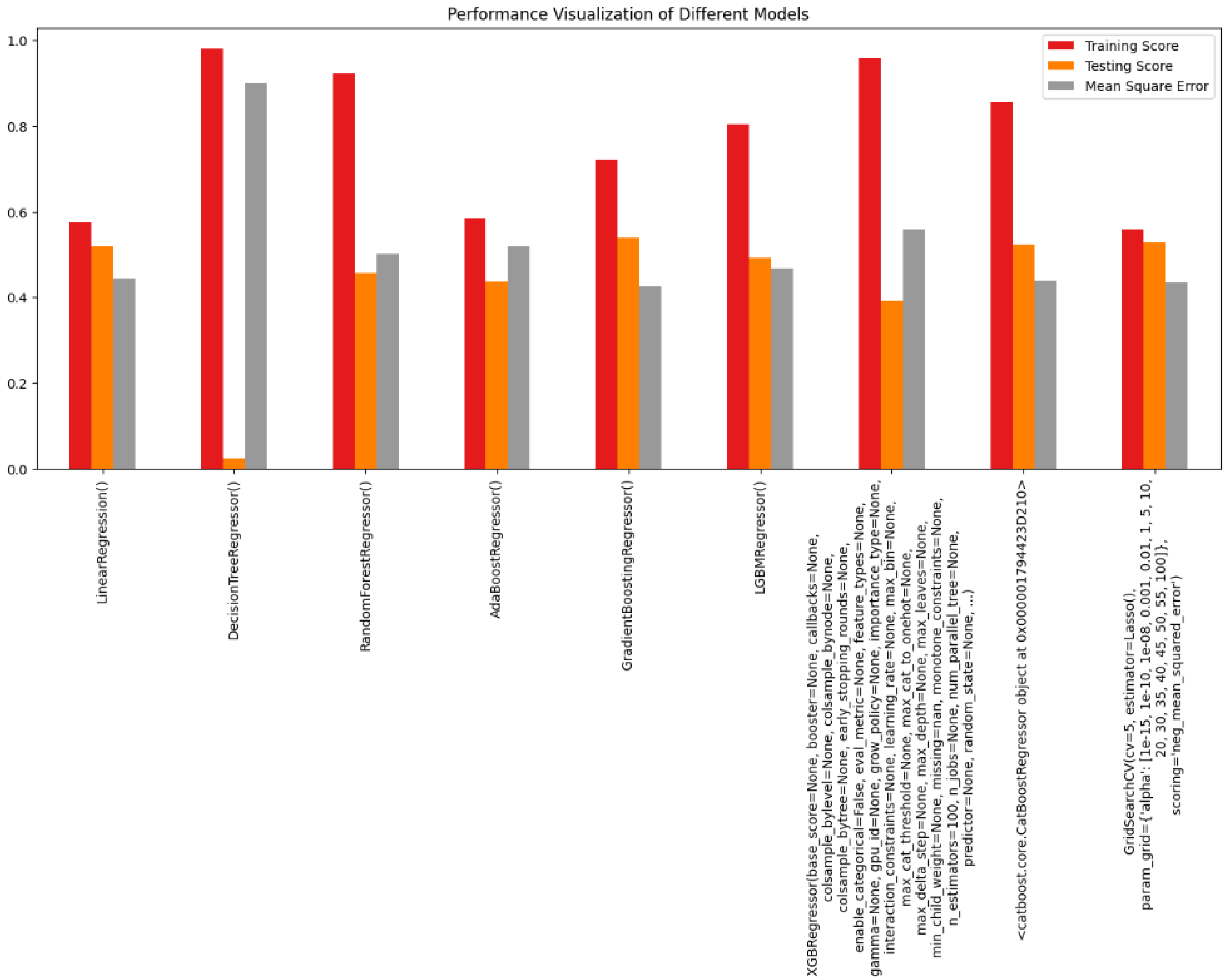
```
X = new_df_dummies_and_scaler.drop(columns=["Price"])
y = new_df_dummies_and_scaler["Price"]
X.shape, y.shape
```

```
((1072, 25), (1072,))
```

```
x_train - >  (857, 25)
x_test - >  (215, 25)
y_train - >  (857,)
y_test - >  (215,)
```

6. Model Selection

	Algorithms	Training Score	Testing Score	Mean Square Error
0	LinearRegression()	0.574740	0.518607	0.444076
1	DecisionTreeRegressor()	0.980405	0.025413	0.899039
2	RandomForestRegressor()	0.922320	0.457235	0.500692
3	AdaBoostRegressor()	0.583103	0.437667	0.518743
4	GradientBoostingRegressor()	0.721175	0.539468	0.424833
5	LGBMRegressor()	0.803285	0.492251	0.468390
6	XGBRegressor(base_score=None, booster=None, ca...	0.957994	0.393075	0.559878
7	<catboost.core.CatBoostRegressor object at 0x0...	0.856320	0.524492	0.438648
8	GridSearchCV(cv=5, estimator=Lasso(),\n ...	0.559526	0.528684	0.434781


Performance Visualization of Different Models
Training Score
Testing Score
Mean Square Error
0.0
0.2
0.4
0.6
0.8
1.0
LinearRegression()
DecisionTreeRegressor()
RandomForestRegressor()
AdaBoostRegressor()
GradientBoostingRegressor()
LGBMRegressor()
XGBRegressor(base_score=None, booster=None, callbacks=None,
colsample_bylevel=None, colsample_bynode=None,
colsample_bytree=None, early_stopping_rounds=None,
enable_categorical=False, eval_metric=None, feature_types=None,
gamma=None, gpu_id=None, grow_policy=None, importance_type=None,
interaction_constraints=None, learning_rate=None, max_bin=None,
max_cat_threshold=None, max_cat_to_onehot=None,
max_delta_step=None, max_depth=None, max_leaves=None,
min_child_weight=None, missing=nan, monotone_constraints=None,
n_estimators=100, n_jobs=None, num_parallel_tree=None,
predictor=None, random_state=None, ...)
<catboost.core.CatBoostRegressor object at 0x000001794423D210>
GridSearchCV(cv=5, estimator=Lasso(),
param_grid={'alpha': [1e-15, 1e-10, 1e-08, 0.001, 0.01, 1, 5, 10,
20, 30, 35, 40, 45, 50, 55, 100]},
scoring='neg_mean_squared_error')

7. Decision making

In terms of all models that we tested, it is the best that we choose the GradientBoostingRegressor Model.

Thank you!