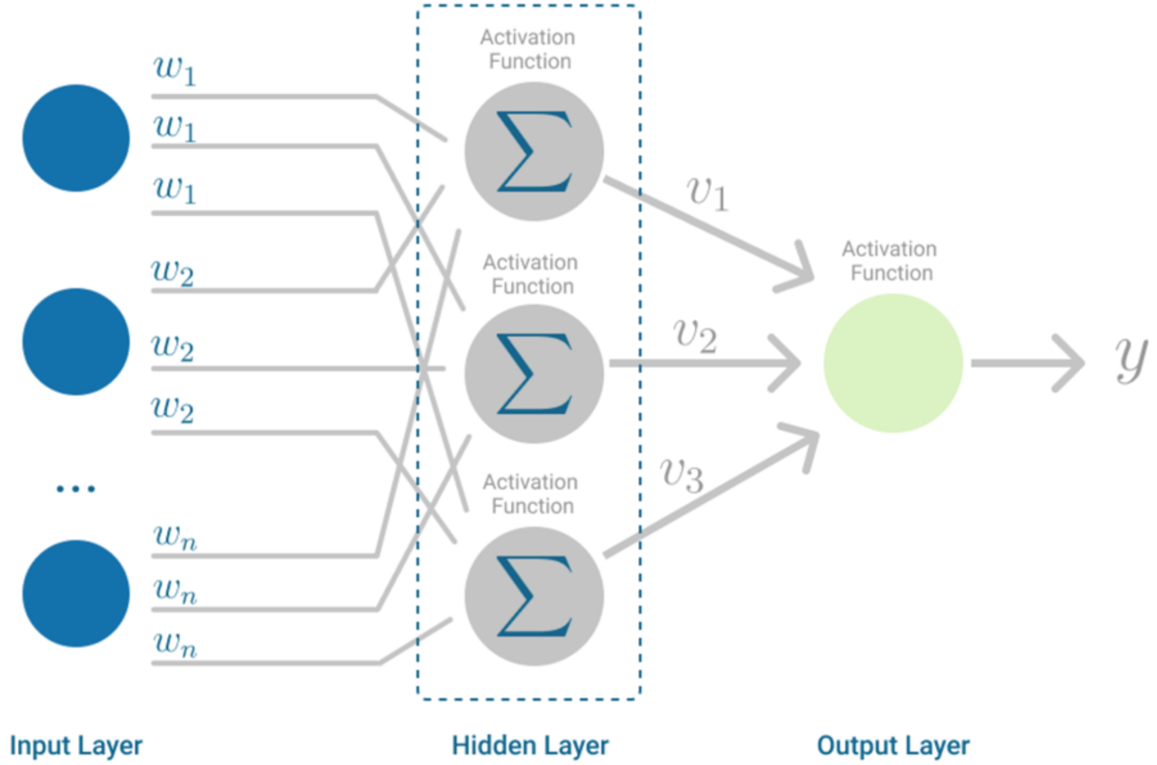
A Model for Prediction of Housing Price Using MLP, XGBOOST

12190180 Nurjol

Contents

Model Introduction

MLP(Multi-Layer Perceptron)

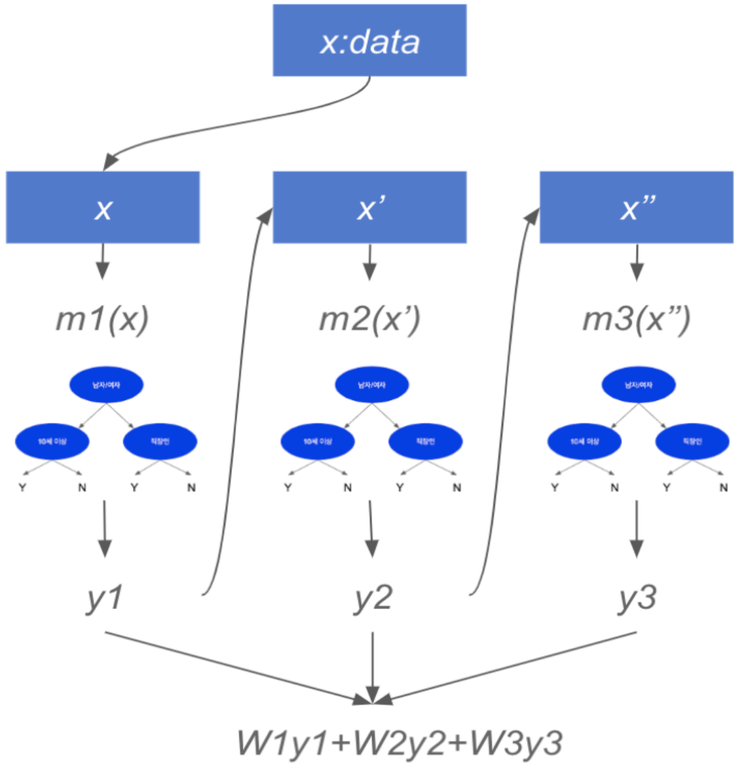


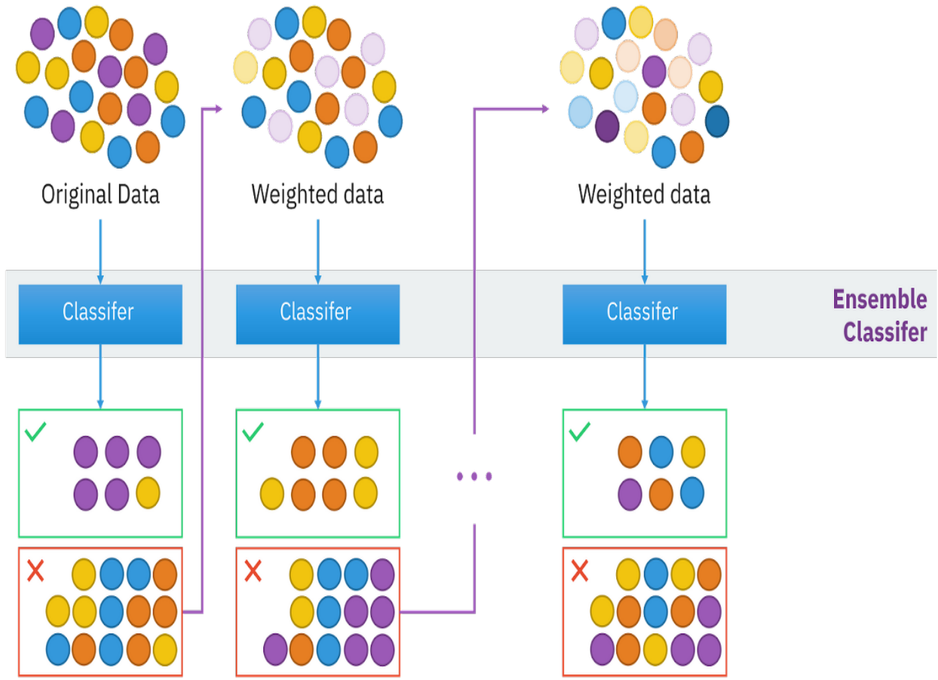
A MLP has input and output layers, and one or more **hidden layers** with many neurons st acked together.

MLP is a neural network where the mapping between inputs and output is **non-linear.**

MLP uses **backpropogation for training** the network. Cross-validation techniques must b e used to find ideal values for these.

Model Introduction





<Boosting Structure>

Label <-> Prediction => Loss

$$X_{n+1} = x_{n+1} + Weight_n \times Loss_n$$

$$y'_i = \sum_{k=1}^{K} f_k(x_i)\ , f_k\ \in F$$

$$\text{Obj} = \sum_{i=1}^{n} l(y_i, y'_i) + \sum_{k=1}^{K} \Omega(f_k)$$

$* where\ y'_i = \text{predict score corresponding to } x_i,$

$f_k = k\ th\ decision\ tree\ in\ function\ space\ F,$

$l = loss\ function,$

$\Omega = regularization\ term (= complexity\ of\ Trees)$

Feature Engineering

Data Preprocess

Setting variable

City, county, district
Dedicated area
Contract year and month
Transaction amount
Floor level
Year of construction
Starbucks
Within 600ms of
-a subway station
-schools and daycare centers
-a large store
-a hospital
-a park

Data preprocess

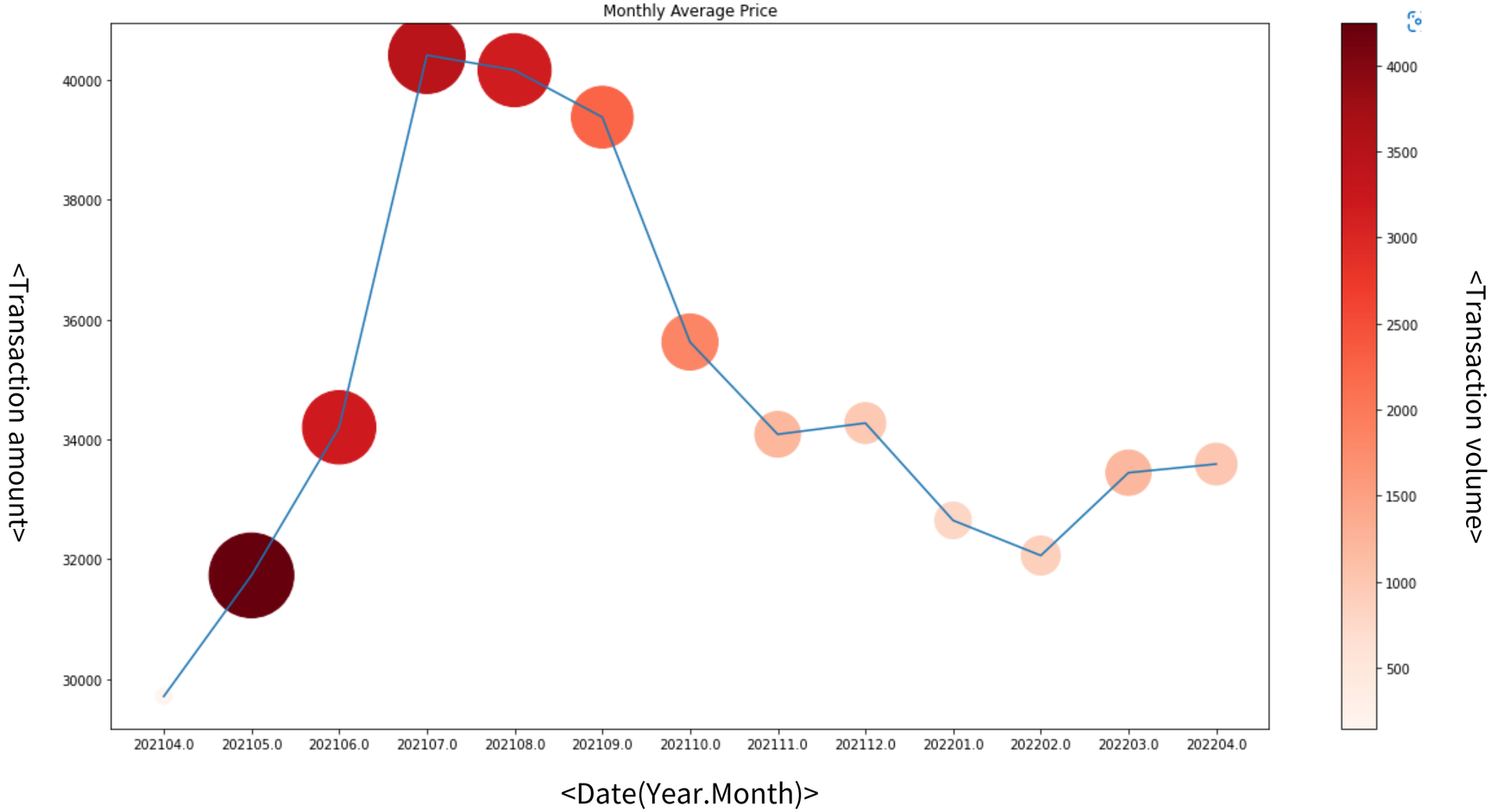
Address of variable

Eup/myeon/dong

Merge excel files

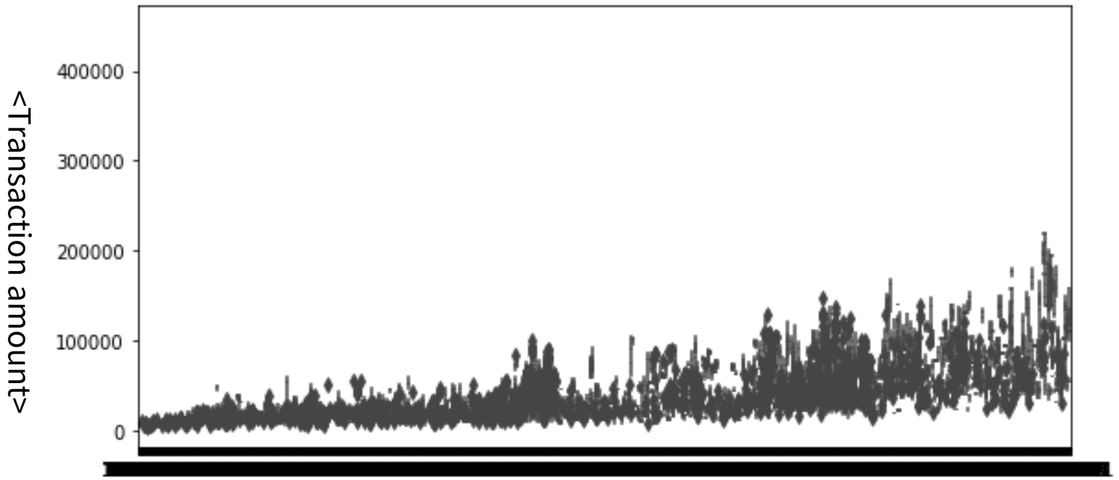
Feature Engineering

Data Correlation Analysis

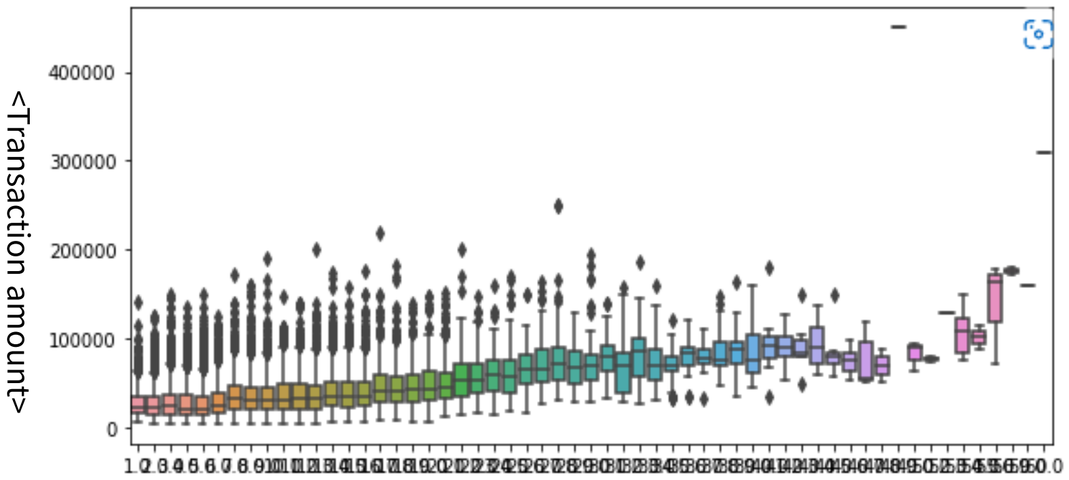


Feature Engineering

Data Correlation Analysis



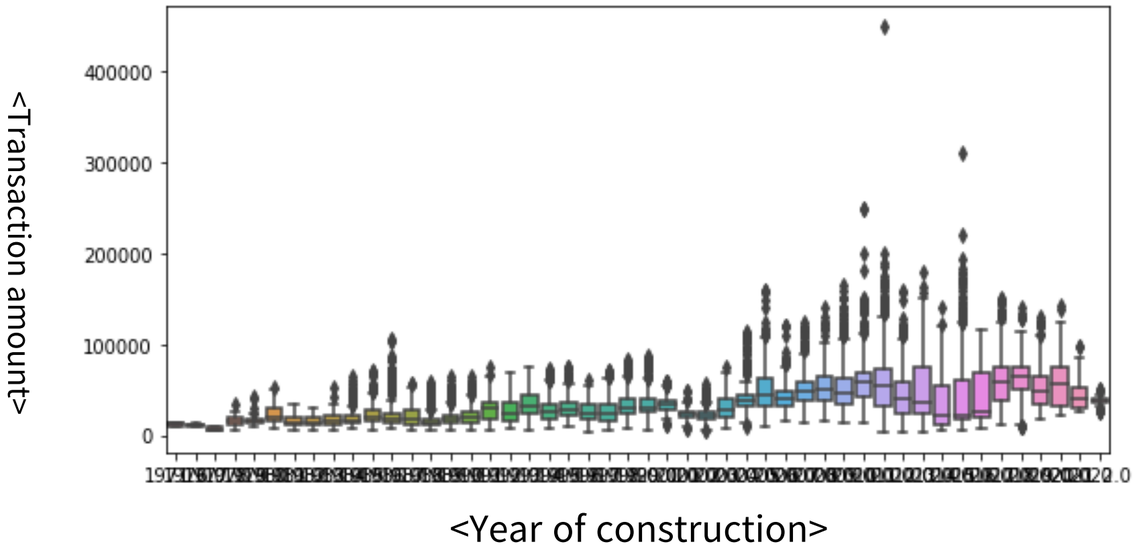
<Dedicated area >

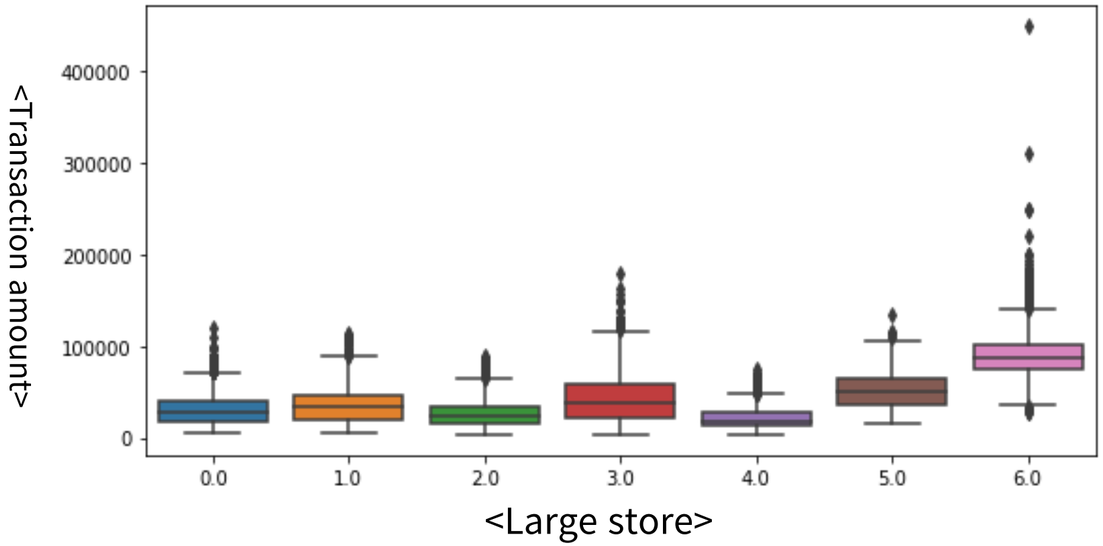


<Floor level>

Feature Engineering

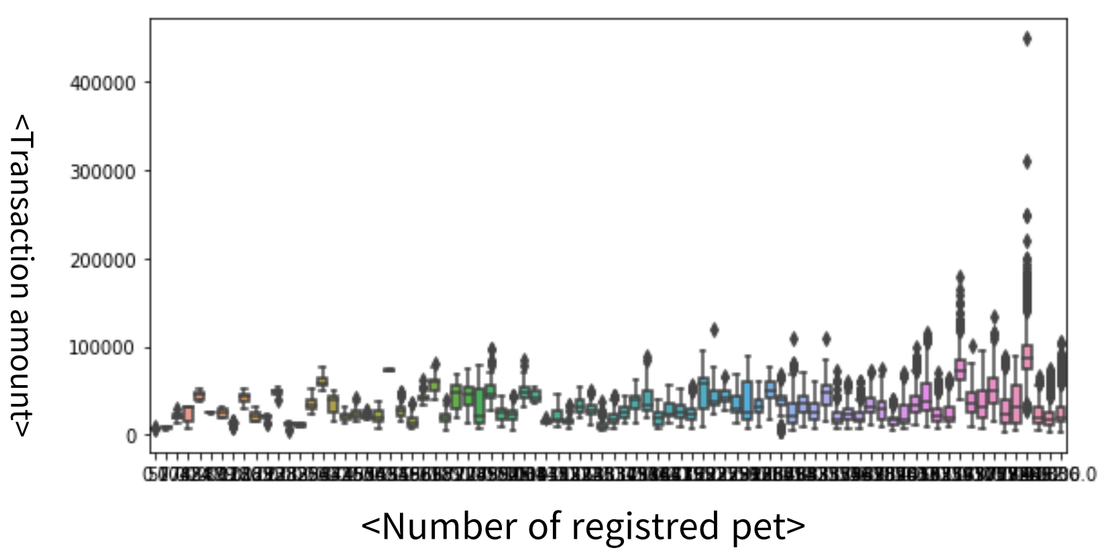
Data Correlation Analysis

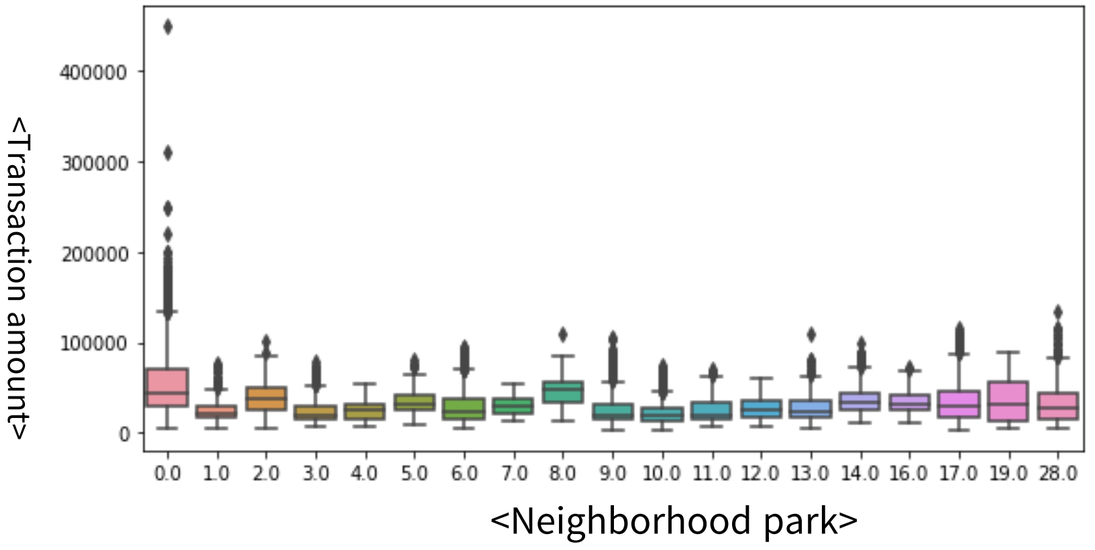




Feature Engineering

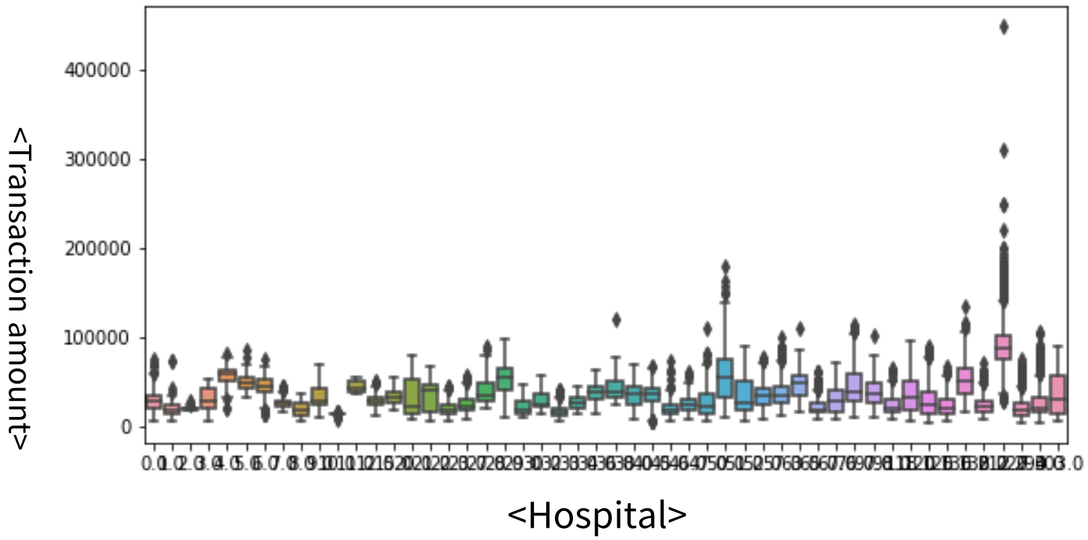
Data Correlation Analysis

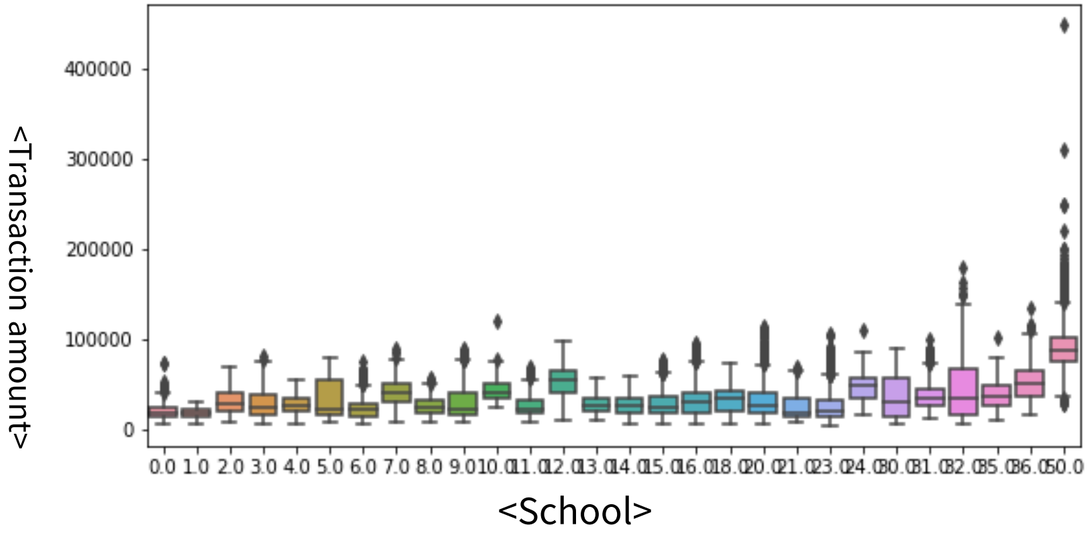




Feature Engineering

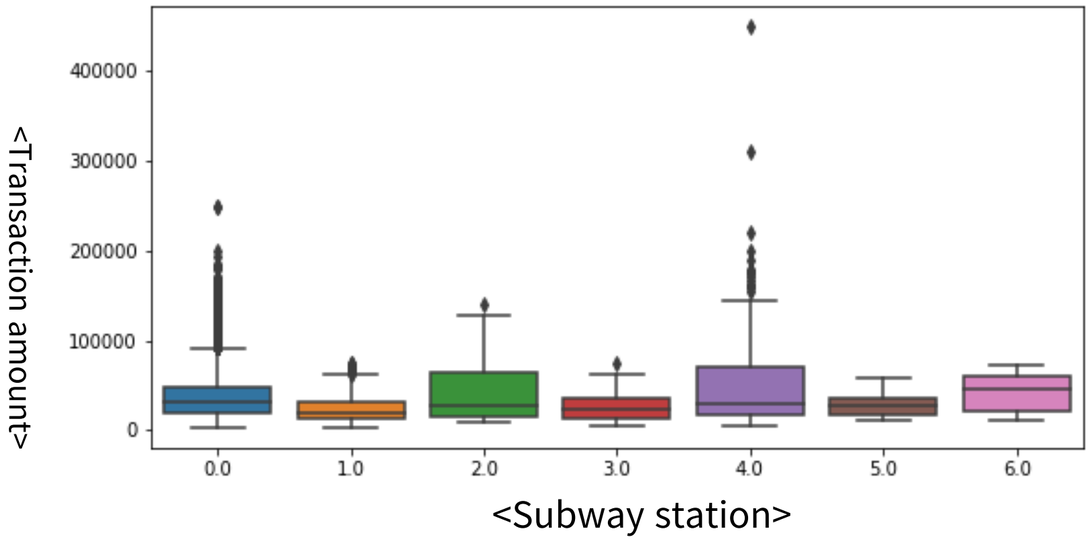
Data Correlation Analysis

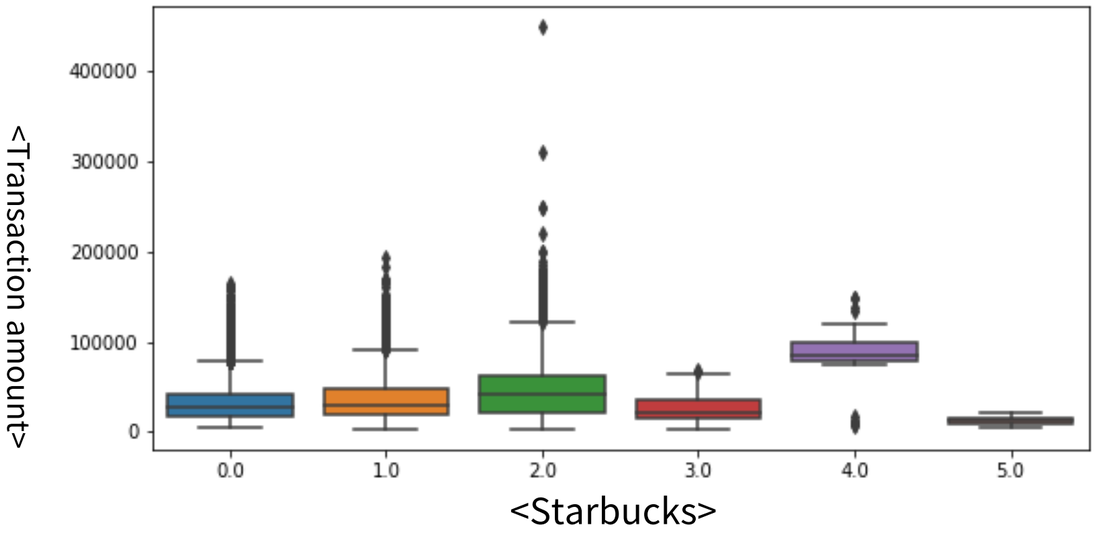




Feature Engineering

Data Correlation Analysis

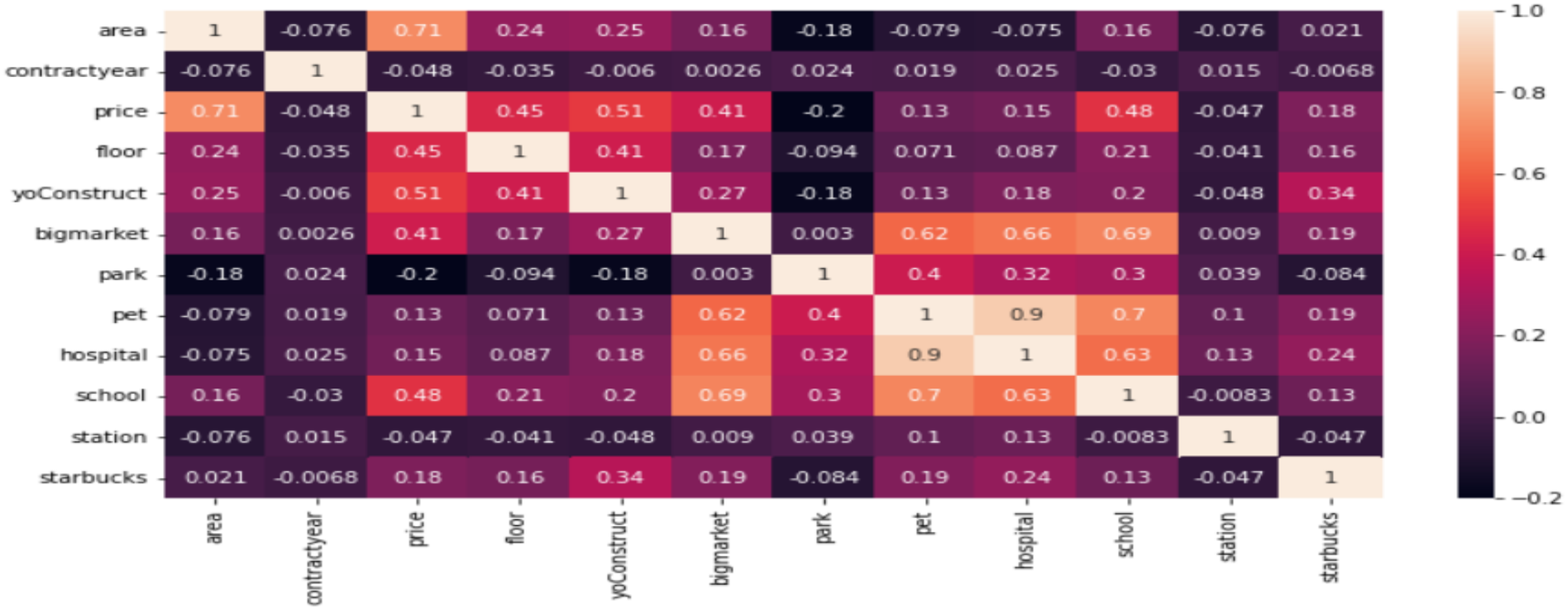




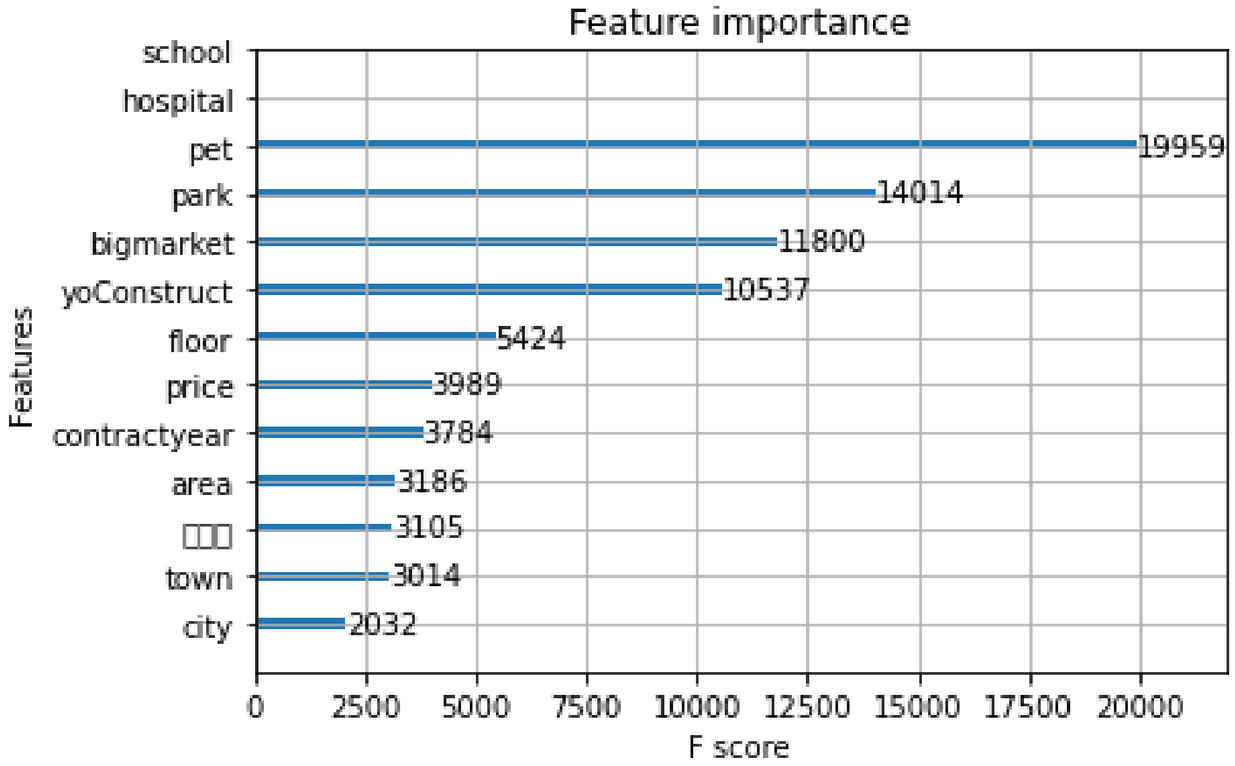
Feature Engineering

Data Correlation Analysis

```
#상관계수 분석
plt.figure(figsize=(12,6))
sns.heatmap(df_copy_english.select_dtypes(np.number).corr(),annot=True)
plt.show()
```



Feature Engineering



<Feature importance>

```
from xgboost import plot_importance # 중요변수 시각화
plot_importance(myxgb)
col_names = df_copy_english.columns
plt.yticks(range(13), col_names)
plt.show()
```

Feature Engineering

```
df_copy['계약년월'] = df_copy['계약년월'].astype('float32')
df_copy_make = df_copy.filter(['전용면적', '계약년월', '거래금액', '층','건축년도','대규모 점포','근린 공원','반려동물 등록수', '병원','학교','station','starbucks'])
df_copy_make.info()
X = df_copy_make.drop('거래금액', axis=1).to_numpy() #데이터 프레임에서 타겟값(거래금액(만원))을 제외하고 넘파이 배열로 만들어준다.
Y = df_copy_make['거래금액'].to_numpy().reshape((-1,1)) # 데이터 프레임 형태의 타겟 값을 넘파이 배열로 만들어준다.
#데이터 스케일링
#sklearn에서 제공하는 MinMaxScaler
#(X- min(X)) / (max(X)-min(X))을 계산
scaler = MinMaxScaler()
scaler.fit(X)
X = scaler.transform(X)

scaler.fit(Y)
Y = scaler.transform(Y)
```

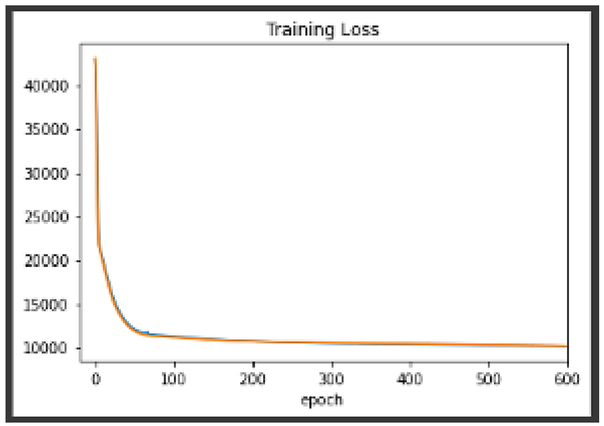
<Data Scaling>

Hyperparameter Optimization

MLP(Multi-Layer Perceptron)

Epoch = 600

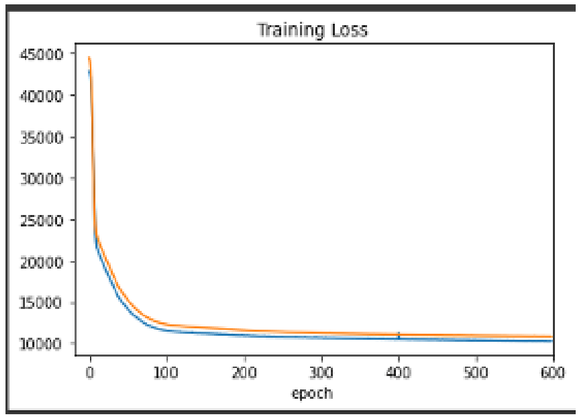
Learning rate = 0.001 Mini-batch = 32



RMSE value:

```
Train RMSE:  10108.052
Vali RMSE:  10169.695
Test RMSE:  10055.026
```

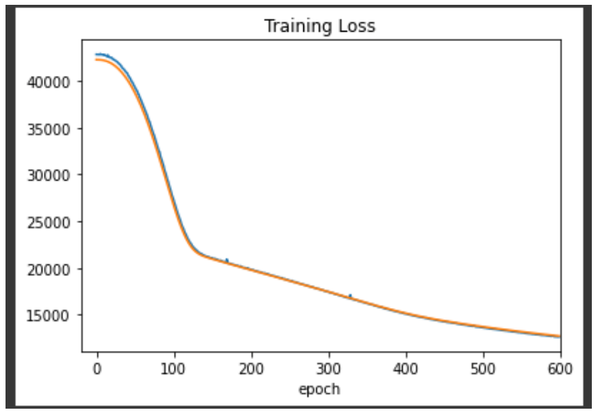
Learning rate = 0.001 Mini-batch = 64



RMSE value:

```
Train RMSE:  10273.312
Vali RMSE:  10841.611
Test RMSE:  9976.493
```

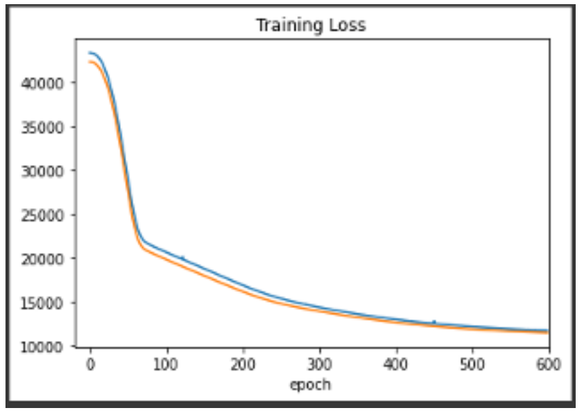
Learning rate = 0.0001 Mini-batch = 64



RMSE value:

```
Train RMSE:  12630.631
Vali RMSE:  12730.155
Test RMSE:  13457.403
```

Learning rate = 0.0001 Mini-batch = 32



RMSE value:

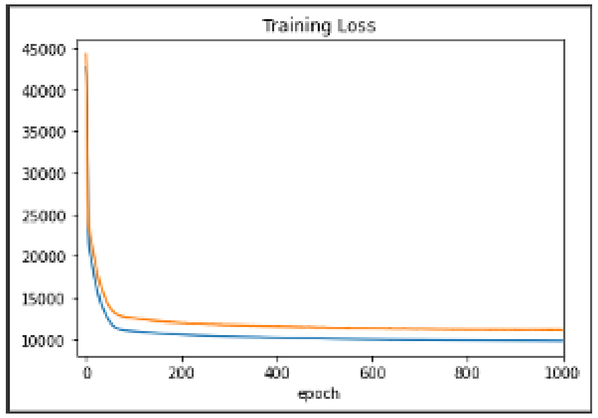
```
Train RMSE:  11736.45
Vali RMSE:  11473.281
Test RMSE:  11647.665
```

Hyperparameter Optimization

MLP(Multi-Layer Perceptron)

Epoch = 1000

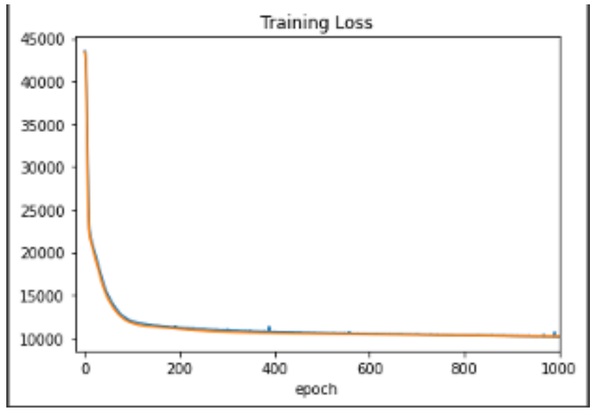
Learning rate = 0.001 Mini-batch = 32



RMSE value:

```
Train MSE:  9799.492
Vali MSE:  11157.86
Test MSE:  9886.362
```

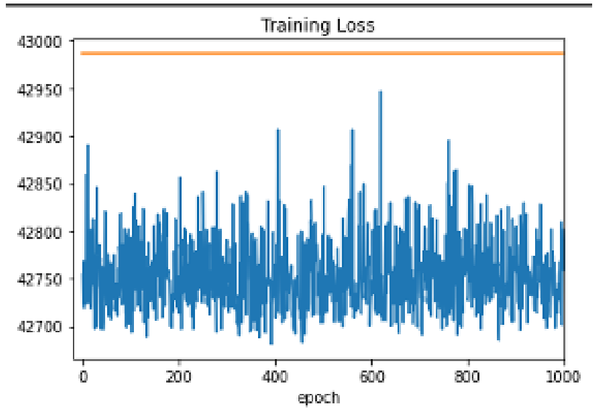
Learning rate = 0.001 Mini-batch = 64



RMSE value:

```
Train RMSE:  10168.68
Vali RMSE:  10187.696
Test RMSE:  9798.836
```

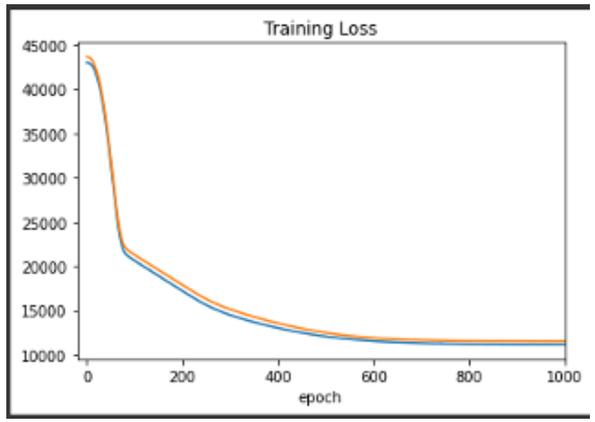
Learning rate = 0.0001 Mini-batch = 64



RMSE value:

```
Train RMSE:  42753.88
Vali RMSE:  42986.383
Test RMSE:  43357.9
```

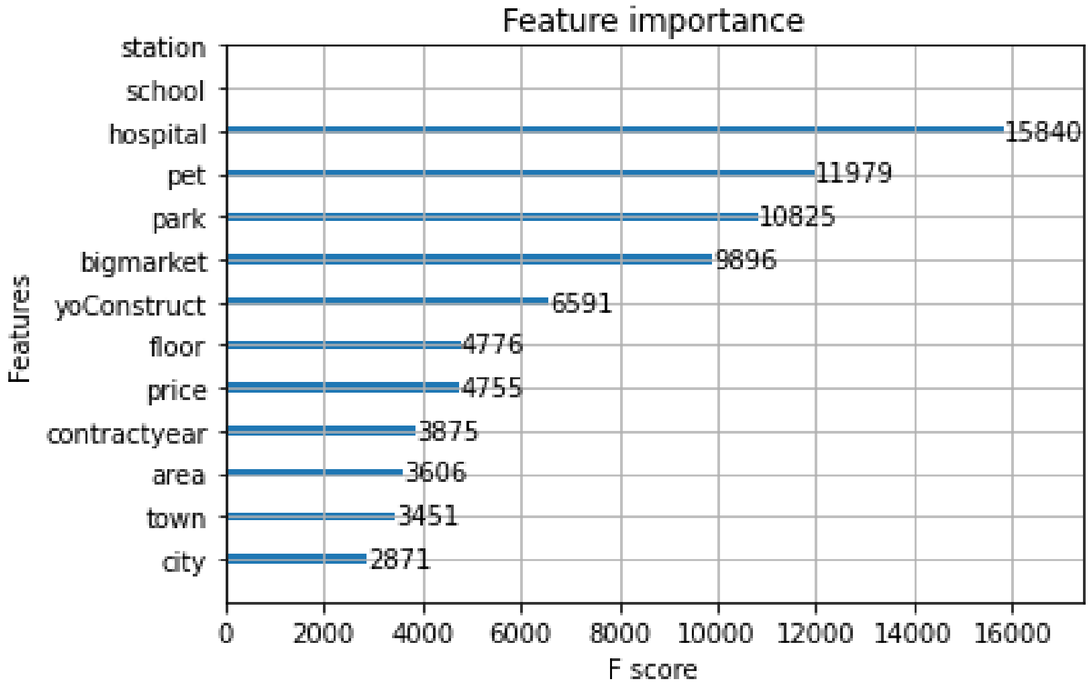
Learning rate = 0.0001 Mini-batch = 32



RMSE value:

```
Train RMSE:  11164.69
Vali RMSE:  11527.402
Test RMSE:  11408.517
```

Hyperparameter Optimization



<Feature importance>

Reference

- https://bcho.tistory.com/1354

- https://webnautes.tistory.com/1643

- '고양시 집값 예측 보고서', 신예진 외 3명

- https://brunch.co.kr/@snobberys/137

THANK YOU

Q&A