
airbnb

❶ 프로젝트 목적 개요

Airbnb Market Strategy

에어비앤비 스마트 가격

에어비앤비 호스팅 경험 시, 에어비앤비의 가격 추천 서비스를 확인하였고 주변 가격에 따라 적정 가격 예측을 해주는 것을 경험

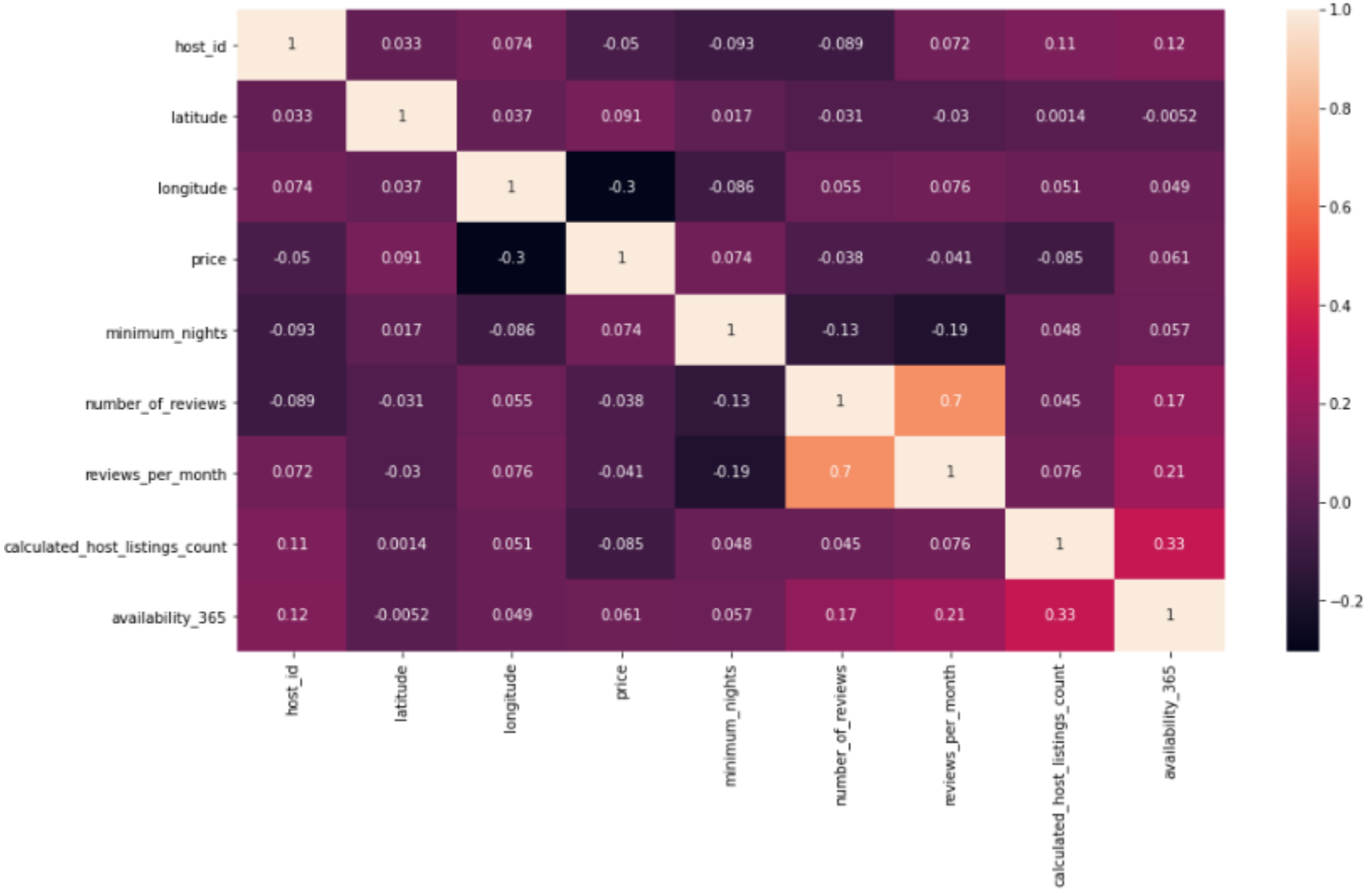
주변 숙소에 따라 가격을 예측하는 방식으로 머신러닝 회귀를 통해 구현이 가능

뉴욕 에어비앤비 데이터를 통한 숙소 가격 예측 모델

데이터 출처 : 캐글

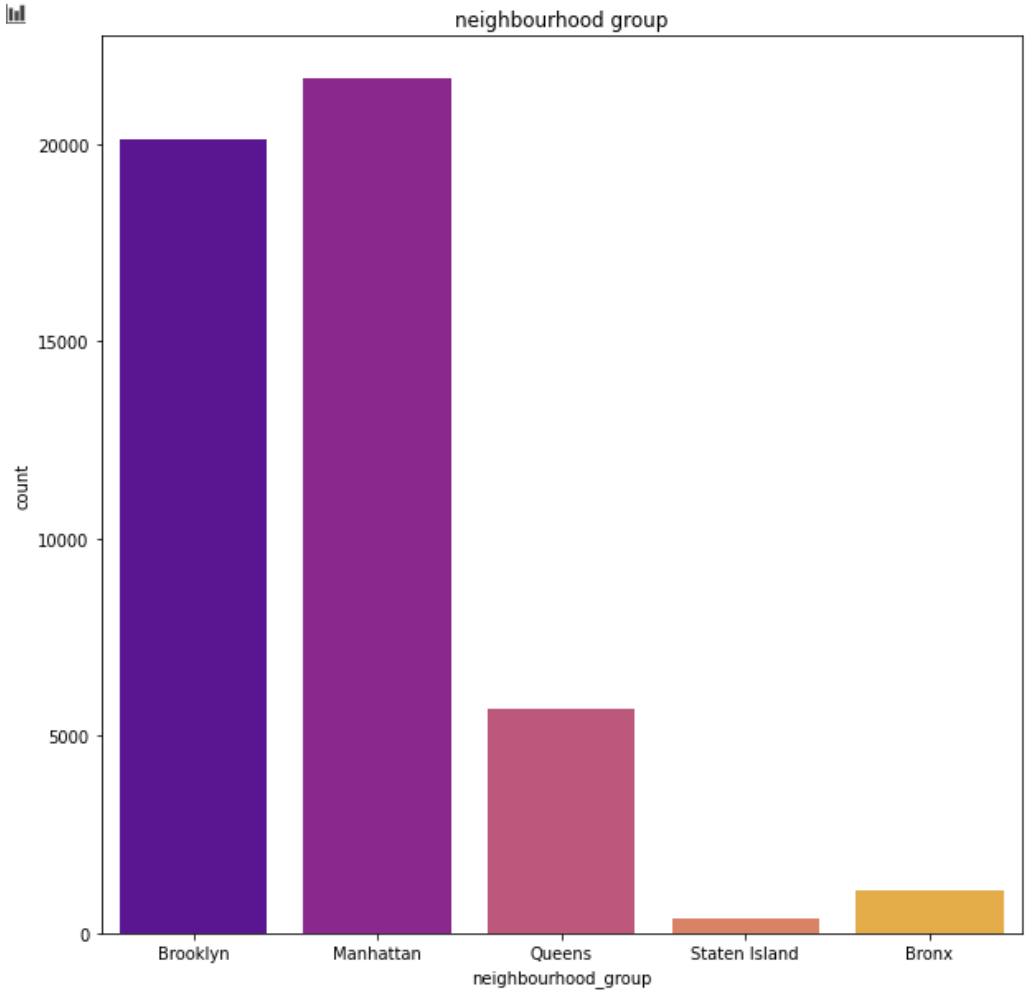
❷ 프로젝트 EDA

EDA



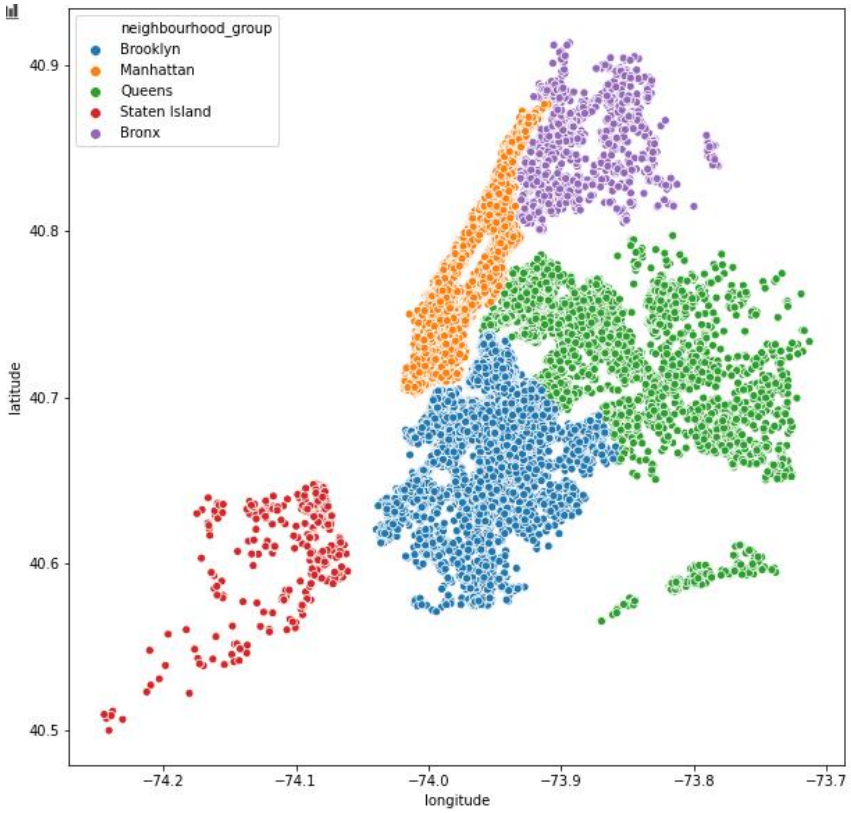
- feature 간의 상관관계 확인

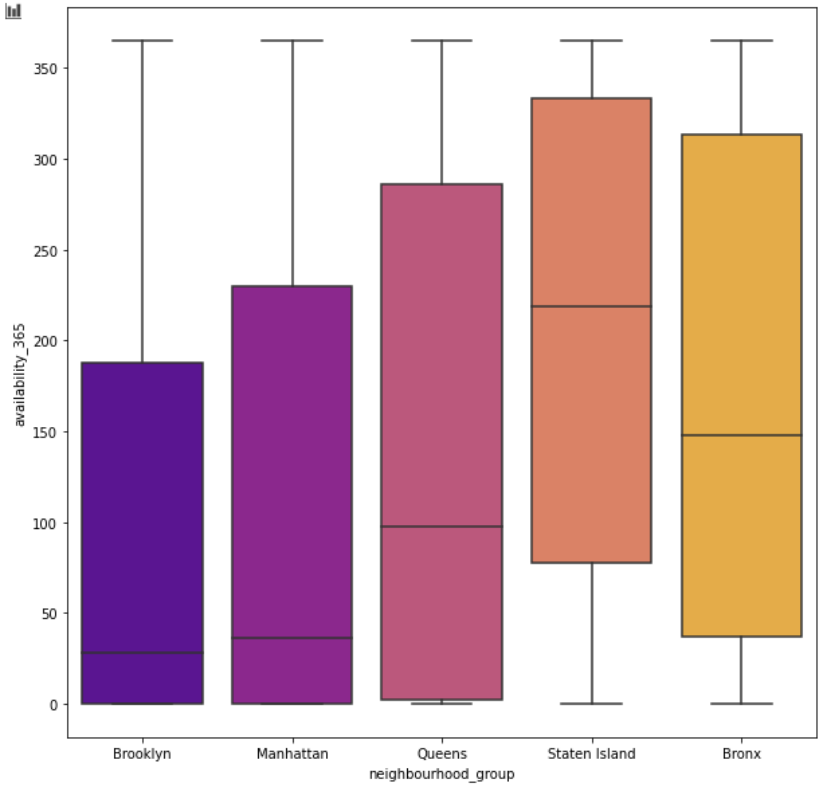
EDA



- 지역별 숙소의 개수 bar plot
- Brooklyn, Manhattan 지역에 집중

EDA



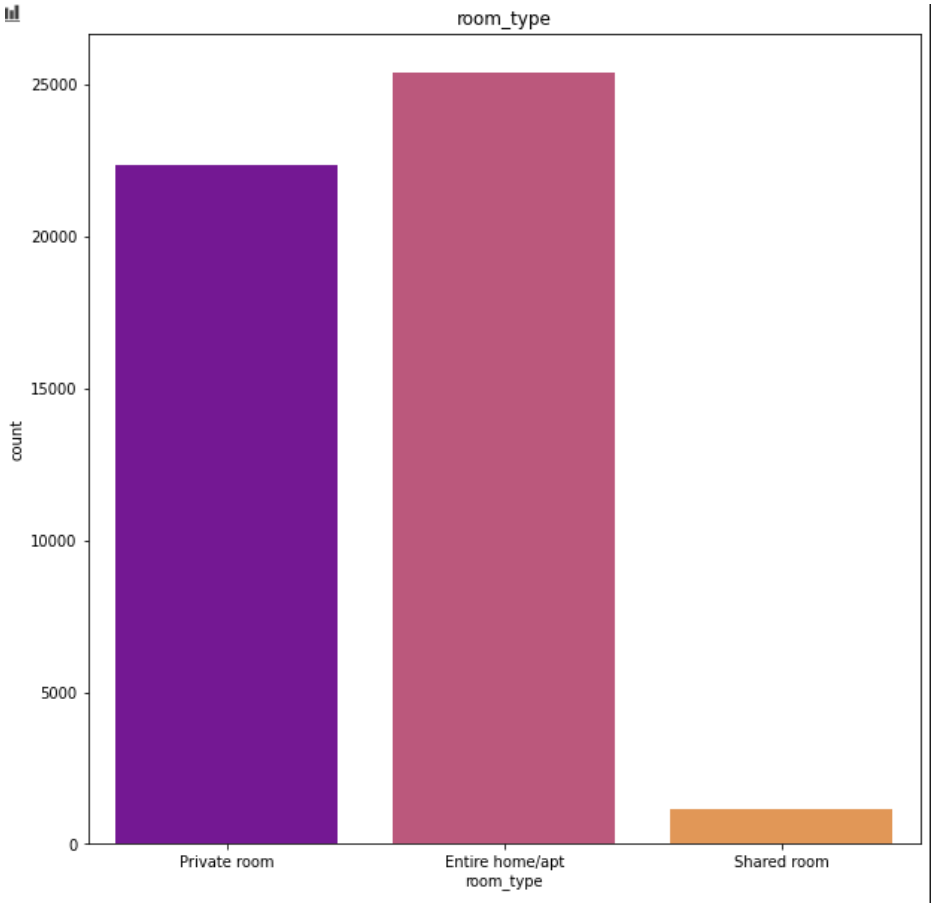


- 맨하탄의 경우 많은 숙소대비 지역에 집중되어 있음
- Staten Island 의 경우 숙소 분포가 매우 넓고 개수도 적음
  ▶관광지나 비즈니스 목적이 아닐 것으로 유추

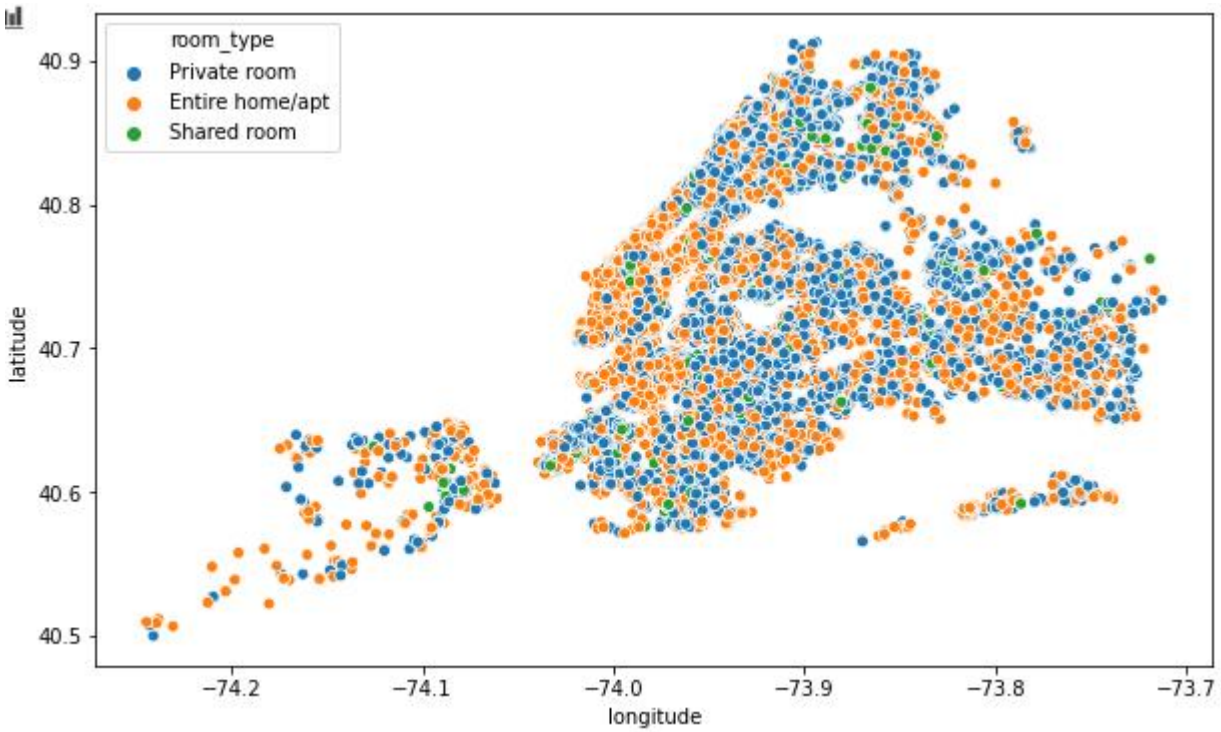
- Brooklyn, Manhattan의 경우,
  많은 객실 대비 가능일수가 1년 전체가 아님
  높은 집값으로 인하여 공유가 더욱 활성화를 유추

EDA

- 숙소 유형은 쉐어 형태보다 개별룸 또는 집 전체에 대한 유형이 많음
- ‘Private Room’의 경우에도 방을 제외한 거실이나 욕실 공간에 대한 공유가 많을 것으로 예상

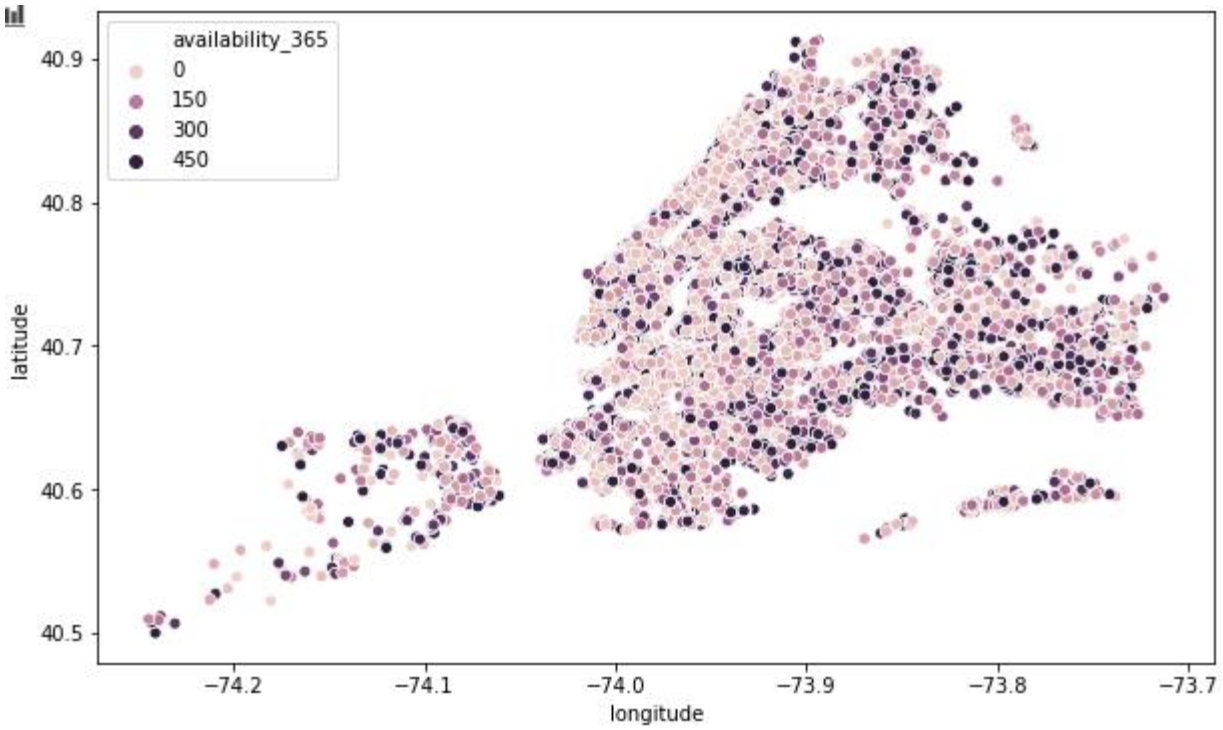


EDA



- 지역별 private / entire 형태가 비슷하게 분포

EDA



- 대부분의 숙소 중 활성화가 안된 곳이 많음

EDA

- 주요 도시에 대한 시각화, 관광지역이 눈에 띔

❸ 회귀 분석

회귀 머신러닝

데이터 전처리

데이터 전처리

```
# drop column 타겟에 영향을 미치지 않는 독립변수 제거

airbnb.drop(['host_id', 'latitude', 'longitude', 'neighbourhood', 'number_of_reviews', 'revi
airbnb.head()
```

	neighbourhood_group	room_type	price	minimum_nights	calculated_host_listings_count	availability_365
0	Brooklyn	Private room	149	1	6	365
1	Manhattan	Entire home/apt	225	1	2	355
2	Manhattan	Private room	150	3	1	365
3	Brooklyn	Entire home/apt	89	1	1	194
4	Manhattan	Entire home/apt	80	10	1	0

- 숙소 가격에 영향을 주지 않는 feature 삭제
- null 값이 너무 많은 경우 삭제를 진행

데이터 전처리

```
# 범주형 데이터는 인코딩

def Encode(airbnb):
    for column in airbnb.columns[airbnb.columns.isin(['neighbourhood_group','room_type'])]:
        airbnb[column] = airbnb[column].factorize()[0] #factorize 범주형
    return airbnb
airbnb_en = Encode(airbnb.copy())
```

```
airbnb_en.tail()
```

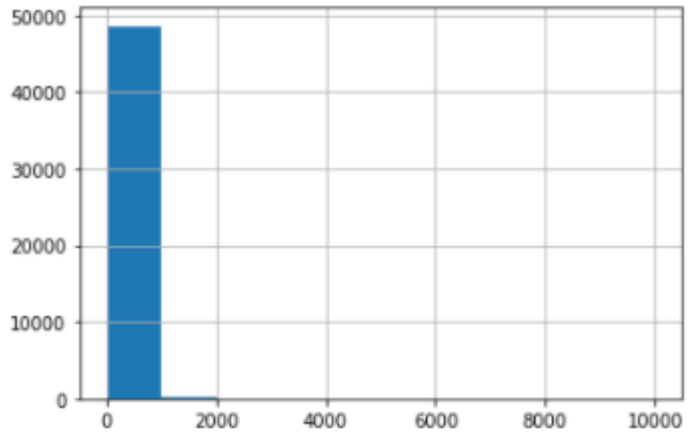
	neighbourhood_group	room_type	price	minimum_nights	calculated_host_listings_count	availability_365
48890	0	0	70	2	2	9
48891	0	0	40	4	2	36
48892	1	1	115	10	1	27
48893	1	2	55	1	6	2
48894	1	0	90	7	1	23

- 범주형 데이터의 경우, 인코딩을 진행

데이터 전처리

```
1 y_target.hist()
2 # 타겟의 분포가 심하게 편중되어 있음
3 # 로그화 처리하여 분포를 변경하기
```

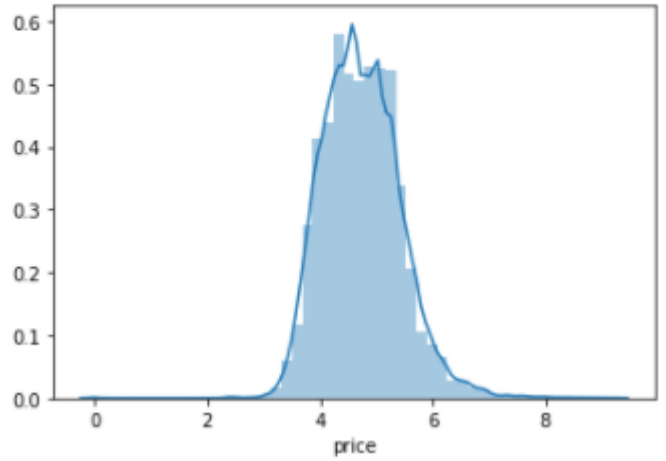
```
<AxesSubplot:>
```



- 타겟 [price]에 대한 분포를 확인
- 정규분포의 형태가 아님

```
1 log_price = np.log1p(airbnb_en['price'])
2 sns.distplot(log_price)
```

```
<AxesSubplot:xlabel='price'>
```

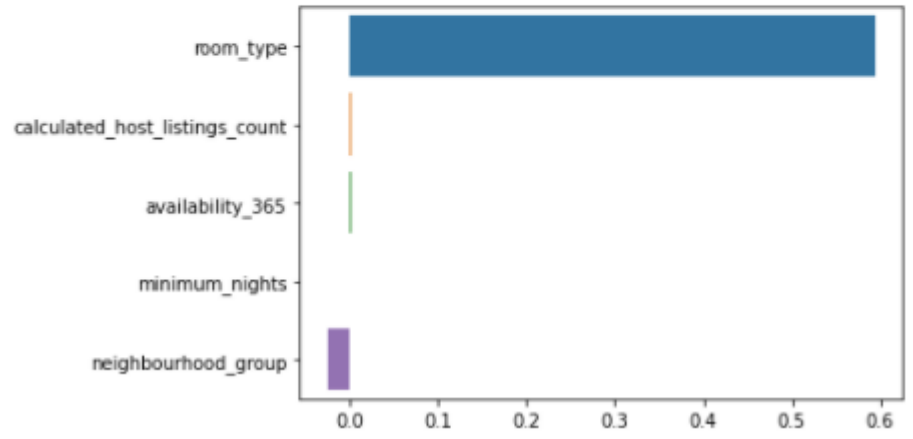


- 타겟에 대한 log 변환을 적용하여 정규 분포의 형태로 변환

데이터 전처리

```
1  # 회귀 계수 확인
2
3  coef = pd.Series(lr_reg.coef_, index=X_features.columns)
4  coef_sort = coef.sort_values(ascending=False)
5  sns.barplot(x= coef_sort.values, y= coef_sort.index)
6
7  # room_type 에 대해 영향을 많이 받음을 알 수 있다
```

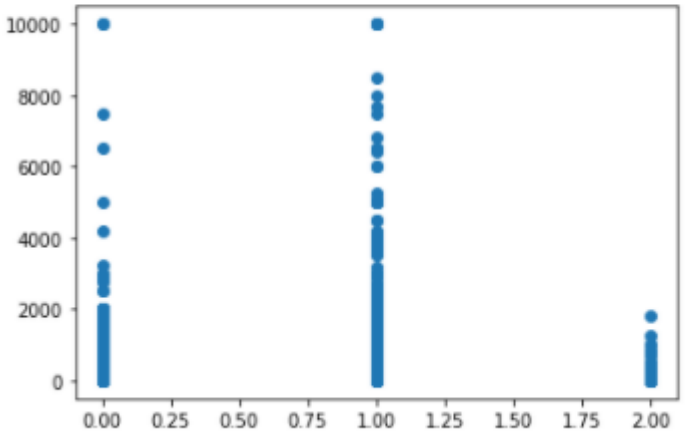
```
<AxesSubplot:>
```



- 타겟에 대한 회귀 계수를 확인
- Room type이 회귀 계수가 높아 해당 컬럼에 대한 전처리를 시도

데이터 전처리

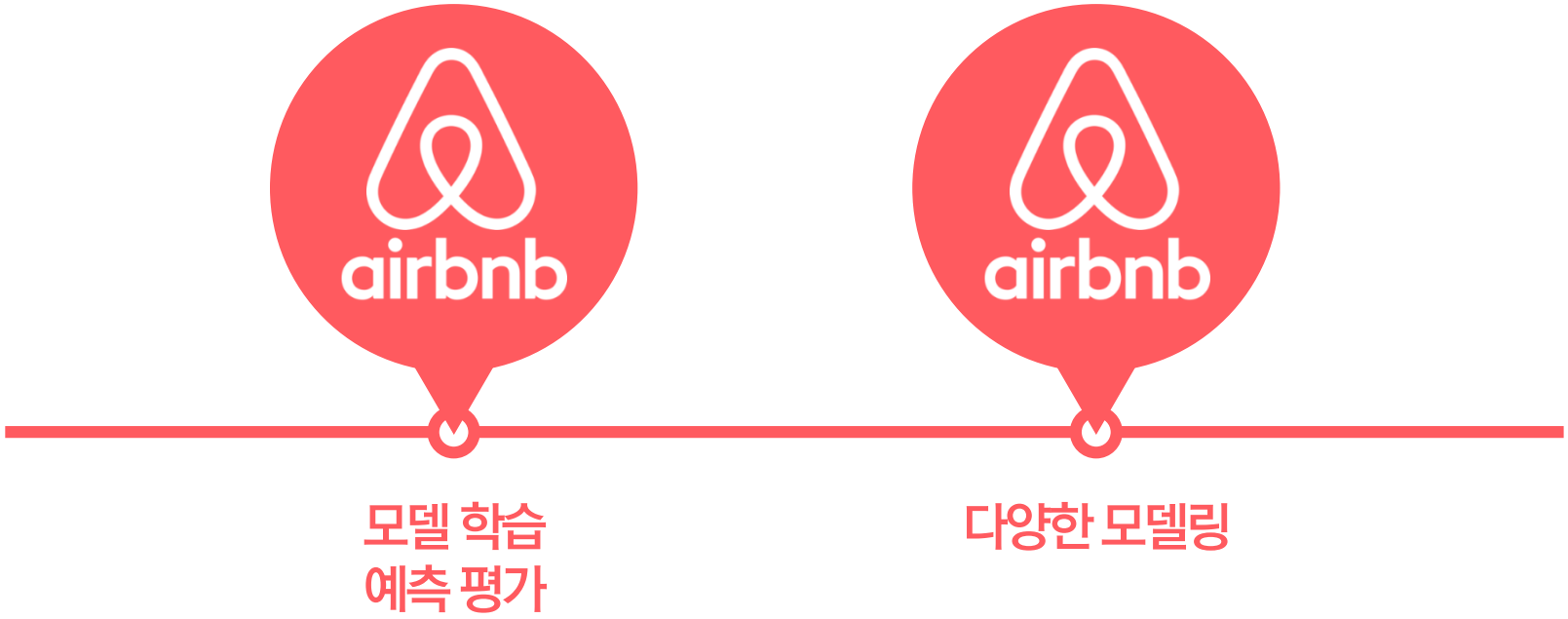
```
# 이상치 제거
# - room_type 과 가격간의 관계에 대해 plot 시각화

plt.scatter(x= airbnb_en['room_type'], y=airbnb_en['price'])
plt.show()
```




```
# 방이 0개임에도 가격이 매우 높은 경우

cond1 = airbnb_en['room_type'] == 0
cond2 = airbnb_en['price'] > 4000
outlier_index = airbnb_en[cond1&cond2].index

airbnb_en.drop(outlier_index, axis=0, inplace= True)

# 방이 1개임에도 가격이 매우 높은 경우

cond1 = airbnb_en['room_type'] == 1
cond2 = airbnb_en['price'] > 6000
outlier_index = airbnb_en[cond1&cond2].index

airbnb_en.drop(outlier_index, axis=0, inplace= True)
```

- 방의 개수에 따라 가격에 대한 분포 중 일정 이상치가 확인
- 특정 숙소의 경우 매우 고급 숙소가 있거나 특수 숙소가 있을 것을 판단
- 방이 0개 인 경우, 4000달러 이상인 숙소와 방이 1개인 경우, 6000달러 이상의 숙소 이상치 제거

회귀 머신러닝



회귀 머신러닝

```python
# 모델 생성 Linear Regression

lr = LinearRegression()
lr.fit(X_train, y_train)
pred = lr.predict(X_test)

from sklearn.metrics import r2_score
r2_score(y_test, pred)
```

```
0.038494045147954 96
```

- R2 스코어 0.03으로 매우 낮음
- 다양한 feature가 없어서, 현재 데이터 내에서의 성능 향상을 고민

회귀 머신러닝

```python
# 모델 학습/예측/평가

def get_eval(model):
    pred= model.predict(X_test)
    mse = mean_squared_error(y_test, pred)
    r2 = r2_score(y_test, pred)
    rmse = np.sqrt(mse)
    return round(rmse,2), r2

def get_evals(models):
    evals=[]
    for model in models:
        eval=get_eval(model)
        evals.append(eval)
    return evals

```

- RMSE 와 R2스코어를 성능 평가에 사용

회귀 머신러닝

```
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error

X_features = airbnb_en.drop(columns='price', axis=1)
y_target = airbnb_en['price']

X_train, X_test, y_train, y_test = train_test_split(X_features, y_target, test_size=0.2, ra

# 모델 학습 객체 생성
lr_reg = LinearRegression()
lr_reg.fit(X_train, y_train)
ridge_reg = Ridge()
ridge_reg.fit(X_train, y_train)
lasso_reg = Lasso()
lasso_reg.fit(X_train, y_train)

models = [lr_reg, ridge_reg, lasso_reg]
get_evals(models)

# 약간의 성능 향상은 있지만 많이 낮음

```

```
[(170.62, 0.08770936468770252),
 (170.62, 0.08770906248719279),
 (170.65, 0.08748350108946668)]
```

- 이상치 제거 후, 모델 성능 향상
- 각 모델별 성능차이가 없어 다른 모델 사용을 고려

회귀 머신러닝

```
# polynomial 사용하기
from sklearn.preprocessing import PolynomialFeatures

poly_ftr = PolynomialFeatures(degree=3).fit_transform(X_features)
X_train, X_test, y_train, y_test = train_test_split(poly_ftr, y_target, test_size=0.2, rand

# 모델 학습 객체 생성
lr_reg = LinearRegression()
lr_reg.fit(X_train, y_train)
ridge_reg = Ridge()
ridge_reg.fit(X_train, y_train)
lasso_reg = Lasso()
lasso_reg.fit(X_train, y_train)

models = [lr_reg, ridge_reg, lasso_reg]
get_evals(models)
```

```
[(164.44, 0.15261923143560607),
 (164.44, 0.1526298887331603),
 (164.41, 0.15292048783794998)]
```

- Feature 가 적고 선형으로 표현이 어려울 것 판단
- Polinomial을 사용하여 다항회귀를 구현
- 성능 향상을 확인

회귀 머신러닝

```
# 다른 모델 사용
from sklearn.ensemble import RandomForestRegressor
from lightgbm import LGBMRegressor


# polynomial 사용하기
from sklearn.preprocessing import PolynomialFeatures

poly_ftr = PolynomialFeatures(degree=3).fit_transform(X_features)
X_train, X_test, y_train, y_test = train_test_split(poly_ftr, y_target, test_size=0.2, rand

# 모델 학습 객체 생성
lr_reg = LinearRegression()
lr_reg.fit(X_train, y_train)
ridge_reg = Ridge()
ridge_reg.fit(X_train, y_train)
lasso_reg = Lasso()
lasso_reg.fit(X_train, y_train)
rf_reg = RandomForestRegressor()
rf_reg.fit(X_train, y_train)
lgbm_reg = LGBMRegressor()
lgbm_reg.fit(X_train, y_train)

models = [lr_reg, ridge_reg, lasso_reg, rf_reg, lgbm_reg]
get_evals(models)

```

```
[(164.44, 0.15261923143560607),
 (164.44, 0.1526298887331603),
 (164.41, 0.15292048783794998),
 (170.8, 0.08588630279342069),
 (160.69, 0.19083383791464337)]
```

- 랜덤포레스트, LGBM 모델 추가 사용
- 성능 향상 확인

최종 리뷰

- 제한된 데이터에서 모델 성능 향상이 어려움
- 숙소 평점, 편의시설 등 숙소 가격에 영향을 주는 정보에 대해 좀 더 고민이 필요
- 다양한 모델링과 전처리 그리고 다항회귀를 통한 성능 향상


airbnb