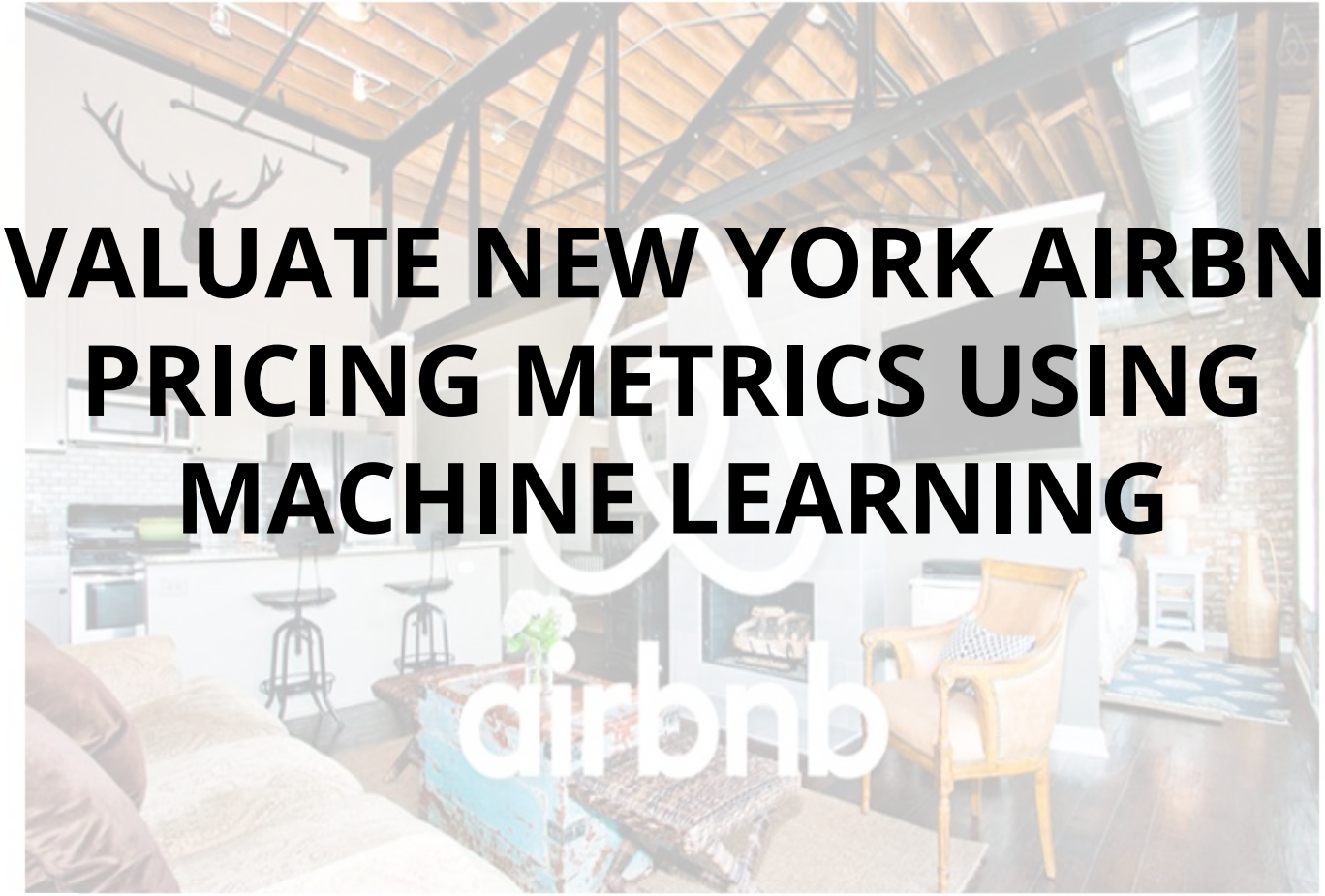
EVALUATE NEW YORK AIRBNB PRICING METRICS USING MACHINE LEARNING



Springboard

M Tariq Aziz

Project Layout

Goal

Overview

Data Specification

Data Wrangling

Exploratory Data Analysis - EDA

Pre-processing and Training Data

Data Modeling

Summary and Findings

Goal

Investment and venture capital firm based in New York owns several successful businesses, and now wants to invest in new york based Airbnbs but wanted to understand in-depth knowledge and information about the market before strategic decision-making

Overview

- Data source: Kaggle
- Tools utilized:
- Python (Google colab)
- Numpy
- Pandas
- Matplotlib
- Seaborn
- Plotly
- Scikit-learn

Data Specification

```
nyc_data_analysis.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48895 entries, 0 to 48894
Data columns (total 16 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   id                              48895 non-null  int64  
 1   name                            48879 non-null  object 
 2   host_id                         48895 non-null  int64  
 3   host_name                       48874 non-null  object 
 4   neighbourhood_group             48895 non-null  object 
 5   neighbourhood                   48895 non-null  object 
 6   latitude                        48895 non-null  float64
 7   longitude                       48895 non-null  float64
 8   room_type                       48895 non-null  object 
 9   price                           48895 non-null  int64  
 10  minimum_nights                  48895 non-null  int64  
 11  number_of_reviews               48895 non-null  int64  
 12  last_review                     38843 non-null  object 
 13  reviews_per_month               48895 non-null  float64
 14  calculated_host_listings_count  48895 non-null  int64  
 15  availability_365                48895 non-null  int64  
dtypes: float64(3), int64(7), object(6)
memory usage: 6.0+ MB
```

```
nyc_data.shape
```

```
(48895, 16)
```

Data Wrangling

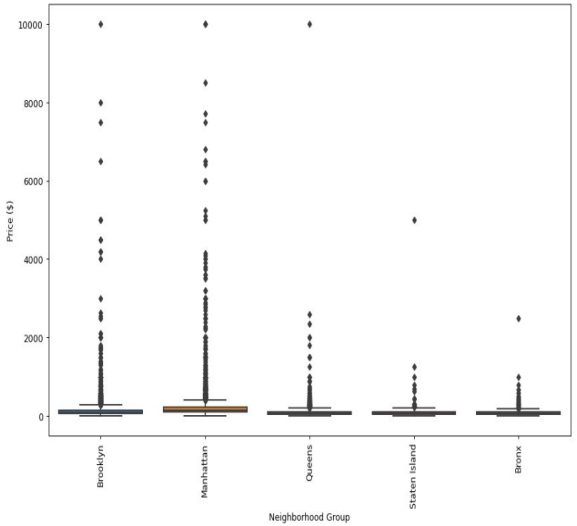
	0	1
id	0	0.000000
host_id	0	0.000000
neighbourhood_group	0	0.000000
neighbourhood	0	0.000000
latitude	0	0.000000
longitude	0	0.000000
room_type	0	0.000000
price	0	0.000000
minimum_nights	0	0.000000
number_of_reviews	0	0.000000
calculated_host_listings_count	0	0.000000
availability_365	0	0.000000
name	16	0.032723
host_name	21	0.042949
last_review	10052	20.558339
reviews_per_month	10052	20.558339

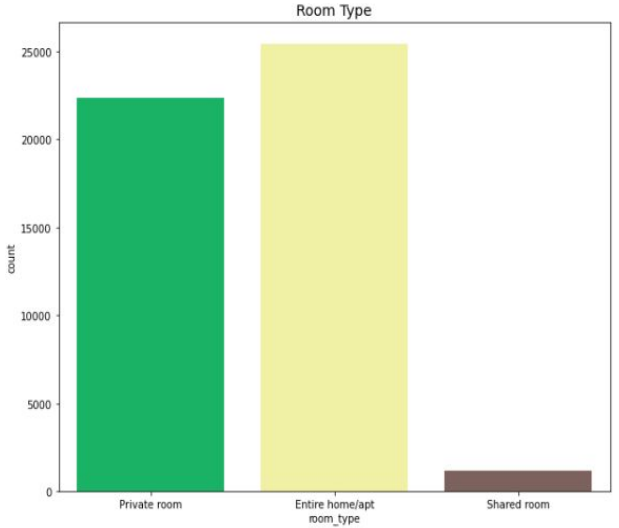
```
nyc_data['neighbourhood_group'].value_counts()
```

```
Manhattan        21661
Brooklyn         20104
Queens            5666
Bronx             1091
Staten Island      373
Name: neighbourhood_group, dtype: int64
```

```
nyc_data['room_type'].value_counts()
```

```
Entire home/apt    25409
Private room       22326
Shared room         1160
Name: room_type, dtype: int64
```



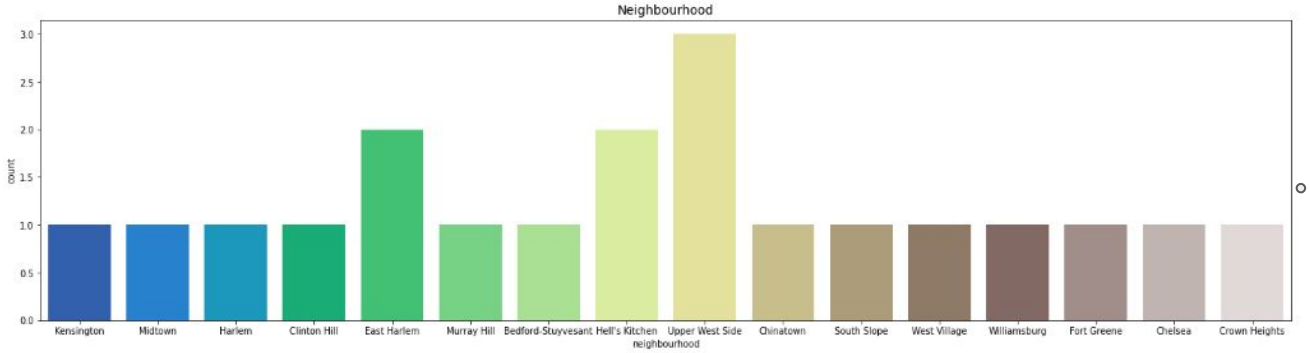


20% Null Values, Fill with 0

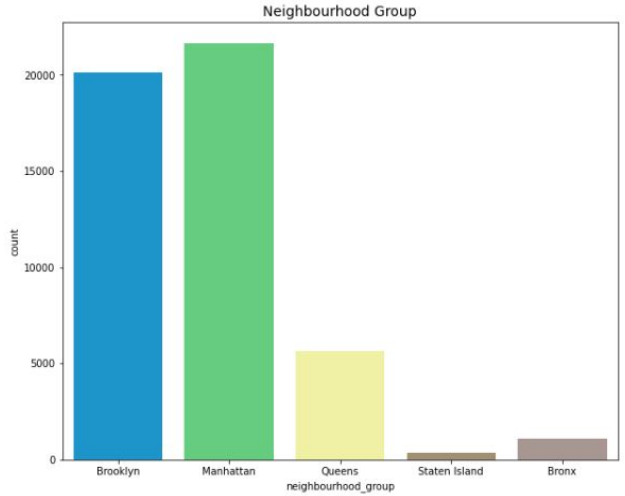
```
nyc_data_analysis.fillna('0', inplace=True)
```

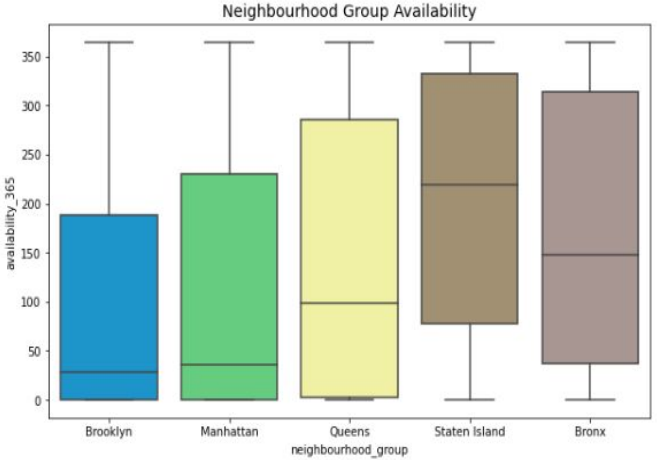

Springboard

Exploratory Data Analysis - EDA (1)



Top 20 Neighbourhood





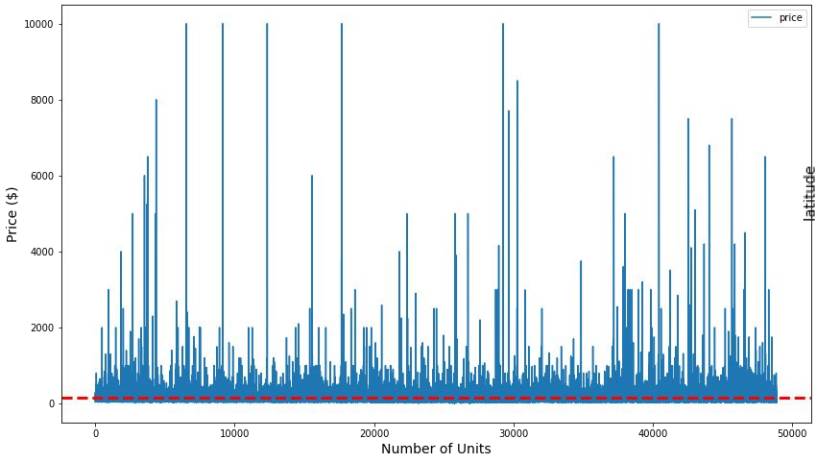
Springboard

Exploratory Data Analysis - EDA (2)

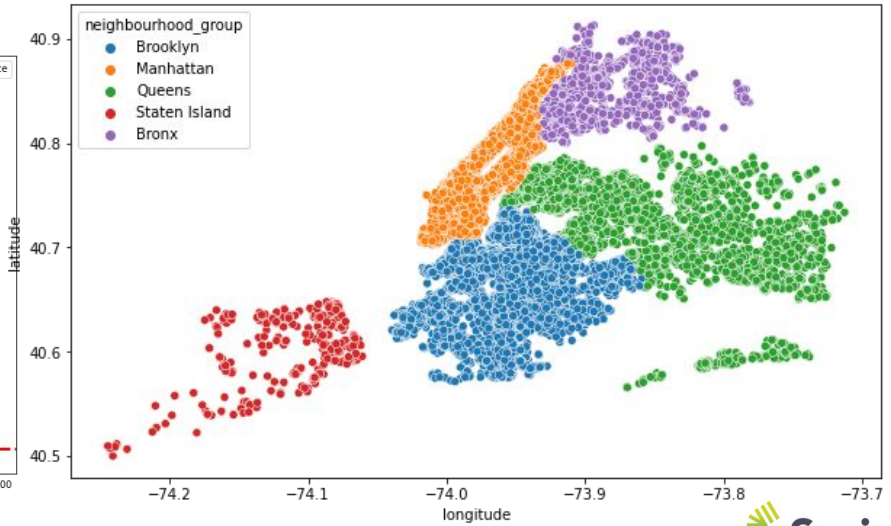
Average Price per Unit

```
import numpy as np
mean_prices = np.mean(nyc_data_analysis.price)
mean_prices
```

```
152.7206871868289
```



Geographical location



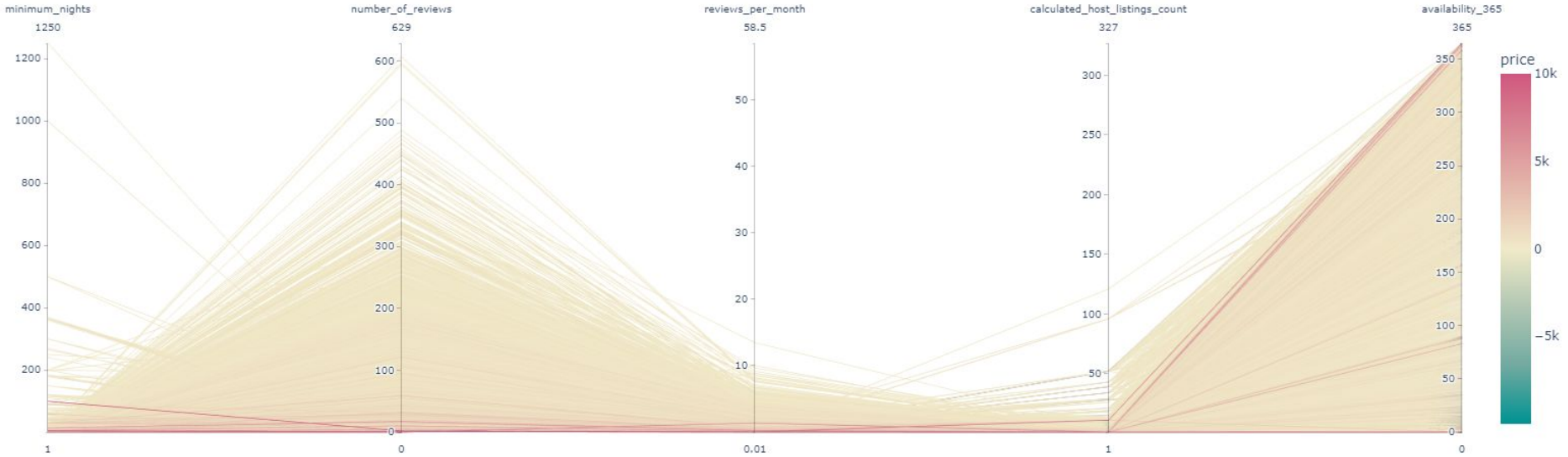
Springboard

Exploratory Data Analysis - EDA (3)

Room Distribution

Neighbourhood Group Word Cloud

Exploratory Data Analysis - EDA (4)



Continuous Variable Relationship with Price

Exploratory Data Analysis - EDA (5)

2D Correlation Matrix

Number of reviews and availability are highly correlated with the price

Pre-processing and Training Data

Dummy variables

```
dataset_new1 = pd.get_dummies(nyc_data_pre_process, columns=['neighbourhood_group', 'room_type'], prefix = ['ng', 'rt'], drop_first = True)
```

Splitting into Training and Testing data 80/20

```
# 20% Test set
# 80% Training set
x_train, x_test, y_train, y_test = train_test_split(X1, Y1, test_size=0.20, random_state=42)
```

```
print(x_test.shape, x_train.shape, y_test.shape, y_train.shape)
```

```
(9779, 19) (39116, 19) (9779,) (39116,)
```

Data Modeling (1)

Feature Selection and Normalization

```
nyc_data_modeling_new = nyc_data_modeling[nyc_data_modeling.price > 0]
nyc_data_modeling_new = nyc_data_modeling[nyc_data_modeling.availability_365 > 0]
```

Log base 10 to reduce outliers

Scaling features between 0 and 1

```
le = LabelEncoder()                                          # Fit label encoder
le.fit(nyc_data_modeling['neighbourhood_group'])
nyc_data_modeling['neighbourhood_group']=le.transform(nyc_data_modeling['neighbourhood_group'])    # Transform labels to normalized encoding.

le = LabelEncoder()
le.fit(nyc_data_modeling['neighbourhood'])
nyc_data_modeling['neighbourhood']=le.transform(nyc_data_modeling['neighbourhood'])

le =LabelEncoder()
le.fit(nyc_data_modeling['room_type'])
nyc_data_modeling['room_type']=le.transform(nyc_data_modeling['room_type'])

nyc_data_modeling.sort_values(by='price',ascending=True,inplace=True)

nyc_data_modeling.head()
```

Data Modeling (2)

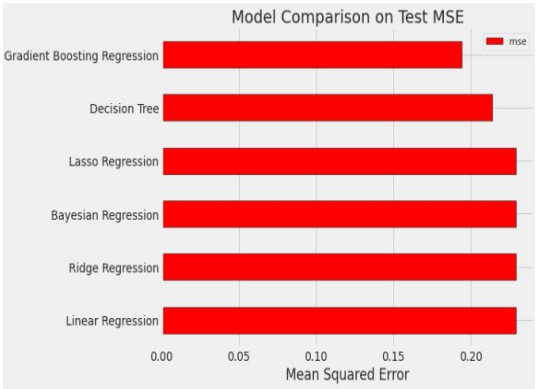
1. Linear Regression
2. Decision Tree
3. Bayesian Regression
4. Ridge Regression (linear model)
5. Lasso Regression (linear model)
6. Gradient Boosting Regression

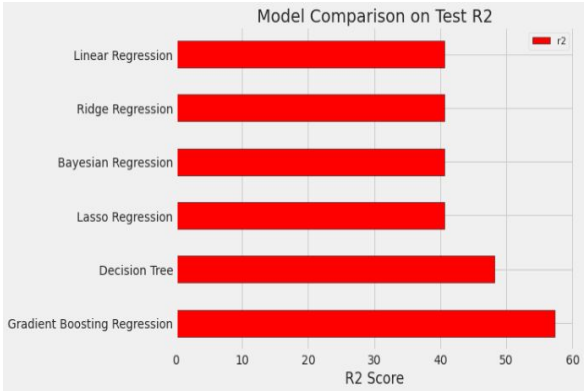
```
# Gradient Boost Regressor
xgb = xgboost.XGBRegressor(n_estimators=200, learning_rate=0.1, objective='reg:squarederror')

# Fit the model
xgb.fit(X_train, y_train)

# Prediction
xgb_pred = xgb.predict(X_test)

# Metric Calculation
xgb_mse = np.sqrt(metrics.mean_squared_error(y_test,xgb_pred))
xgb_r2 = r2_score(y_test,xgb_pred) * 100
xgb_mae = mean_absolute_error(y_test,xgb_pred)


print('''
        Mean Squared Error: {}
        R2 Score: {}
        Mean Absolute Error: {}
    '''.format(
      xgb_mse, xgb_r2, xgb_mae))
```

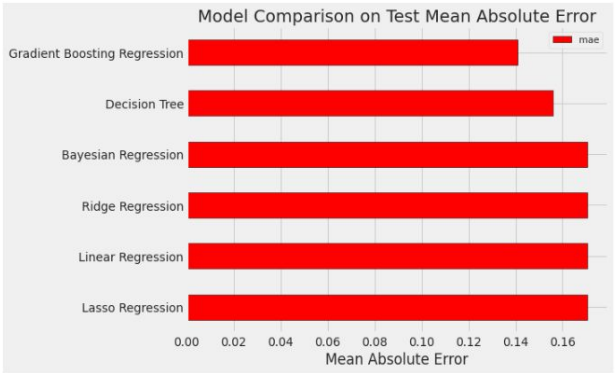
Data Modeling (3)

Model Metric Comparison

Model	Mean Squared Error	R2 Score	Mean Absolute Error
Linear Regression	0.22	40.58	0.17
Decision Tree	0.21	48.2	0.15
Bayesian Regression	0.22	40.58	0.17
Ridge Regression	0.22	40.58	0.17
Lasso Regression	0.22	40.58	0.17
Gradient Boosting Regression	0.19	57.31	0.14







Summary and Findings

- Exploratory data analysis shows data is clean and enough for this project
- New York Airbnb markets are hot any time of the year which means losses would be minimal
- Both linear and complex machine learning models accurately predict the data and performed well, with the complex model showing slightly better performance
- Analysis and comparison proved that data is good enough to make investment-based strategic business decisions