
NYC Housing

PRESENTED BY:

JHARANA ADHIKARI

UMA MAHESHWARI

YASASWIN PALUKURI

Overview

- Why NYC Housing Datasets?
- Characteristics of NYC Datasets
- Lifecycle of Big Data Analytics
- Data Analysis
- Utilization of Analysis Result
- Conclusion
- Reference

Why NYC housing Datasets?

Dynamic real e-state market so, offers us numerous opportunities to utilize of data analytical knowledge.

One of the world's most diverse real e-state market we can get numerous features to perform our task.

Inform strategic decision making for developers investors, & support urban planning initiatives.

Regularly updated data on property sales, making it transparent and easy to conduct research and analysis.

Characteristics of NYC housing Data

Properties sold in New York City over a 12-month period from September 2016 to September 2017.

Detailed information on each property sold, such as location (borough, address), sale price, sale date, and type of building.

Contains 22 columns and 84584 rows with data types: int64(10), object(12) Memory : 14.2+ MB

Business Case Evaluation

The business case comprise of using NYC housing sales data for exploratory data analysis (EDA) linear regression analysis and random-forest regression.

Inform strategic decisions regarding real estate investment, urban planning and policymaking via ongoing property sales, factors influencing prices, and forecasting future growth of New York.

Goal: Assess the a viability of making a real estate investment in New York City. Consider market demand, pricing trends, rental yields, regulatory environment, and potential ROI..

Data Identification

Focuses on identifying and understanding the data types, sources, and characteristics of available data related to a New York Housing.

Identifying the most needed data, identifying features possible outcomes, evaluate the quality of the data, and understand any limitations or limitations of New York Housing Data.

The New York City housing Data is available in different source like Kaggle, UCI repository but decided the Kaggle.

Data Acquisition & Filtering

Access data from https://www.kaggle.com/datasets/new-york-city/nyc-property-sales.

NYC Property Sales

A year's worth of properties sold on the NYC real estate market

Data Card | Code (55) | Discussion (5)

About Dataset

Usability ⓘ
8.24

License
CC0: Public Domain

Context

This dataset is a record of every building or building unit (apartment, etc.) sold in the New York City

```
NewYork_data=pd.read_csv("nyc-rolling-sales.csv")
```

```
#how many rows and columns?
NewYork_data.shape
```

```
(84548, 22)
```

```
NewYork_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 84548 entries, 0 to 84547
Data columns (total 22 columns):
 #   Column                          Non-Null Count  Dtype 
---  ------                          --------------  ----- 
 0   Unnamed: 0                      84548 non-null  int64 
 1   BOROUGH                         84548 non-null  int64 
 2   NEIGHBORHOOD                    84548 non-null  object
 3   BUILDING CLASS CATEGORY         84548 non-null  object
 4   TAX CLASS AT PRESENT            84548 non-null  object
 5   BLOCK                           84548 non-null  int64 
 6   LOT                             84548 non-null  int64 
 7   EASE-MENT                       84548 non-null  object
 8   BUILDING CLASS AT PRESENT       84548 non-null  object
 9   ADDRESS                         84548 non-null  object
 10  APARTMENT NUMBER                84548 non-null  object
 11  ZIP CODE                        84548 non-null  int64 
 12  RESIDENTIAL UNITS               84548 non-null  int64 
 13  COMMERCIAL UNITS                84548 non-null  int64 
 14  TOTAL UNITS                     84548 non-null  int64 
 15  LAND SQUARE FEET                84548 non-null  object
 16  GROSS SQUARE FEET               84548 non-null  object
 17  YEAR BUILT                      84548 non-null  int64 
 18  TAX CLASS AT TIME OF SALE       84548 non-null  int64 
 19  BUILDING CLASS AT TIME OF SALE  84548 non-null  object
 20  SALE PRICE                      84548 non-null  object
 21  SALE DATE                       84548 non-null  object
dtypes: int64(10), object(12)
memory usage: 14.2+ MB
```

Data Extraction

Extracting the relevant data from the following datasets: building class at present, tax class at the time of sale, residential units, commercial units, demographic profiles, and property sales records.

Data Validation & Cleansing

Examining the data for errors, missing values, and discrepancies.

Verifying data integrity and validate property information against government records.

Eliminating duplication, fix errors, and standardize formats to clean up data.

Data Validation & Cleansing

```
NewYork_data.drop(columns = ['Unnamed: 0'], inplace = True)
```

```
#converting 'SALE DATE' column to datetime format
NewYork_data['SALE DATE'] = pd.to_datetime(NewYork_data['SALE DATE'])
```

```
NewYork_data['SALE DATE']
```

```
0       2017-07-19
1       2016-12-14
2       2016-12-09
3       2016-09-23
4       2016-11-17
           ...
84543   2016-11-28
84544   2017-04-21
84545   2017-07-05
84546   2016-12-21
84547   2016-10-27
Name: SALE DATE, Length: 84548, dtype: datetime64[ns]
```

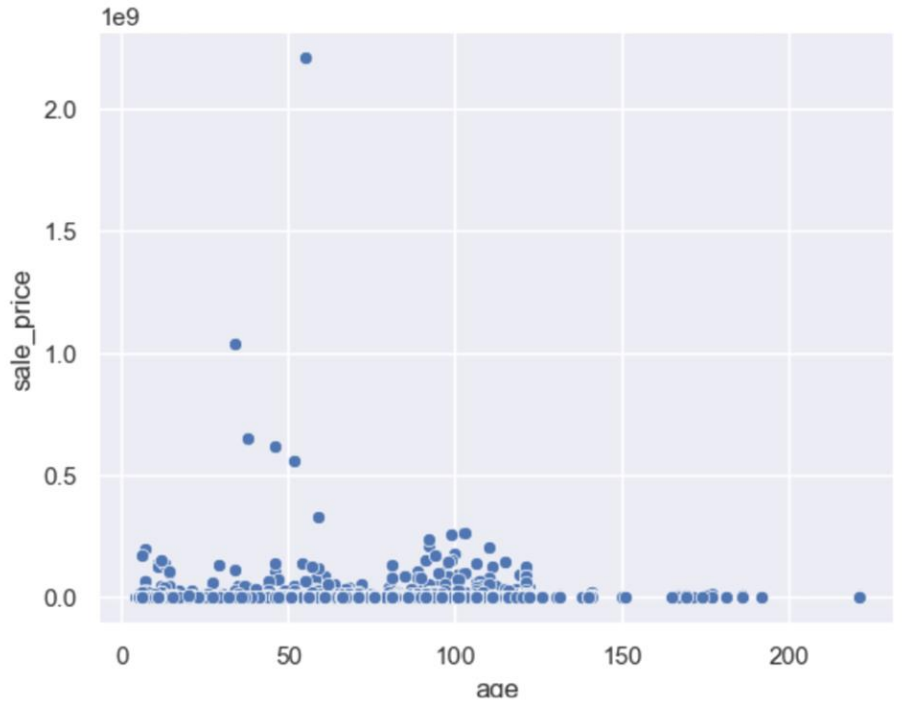
MISSING VALUES

```
NewYork_data.isnull().sum()
```

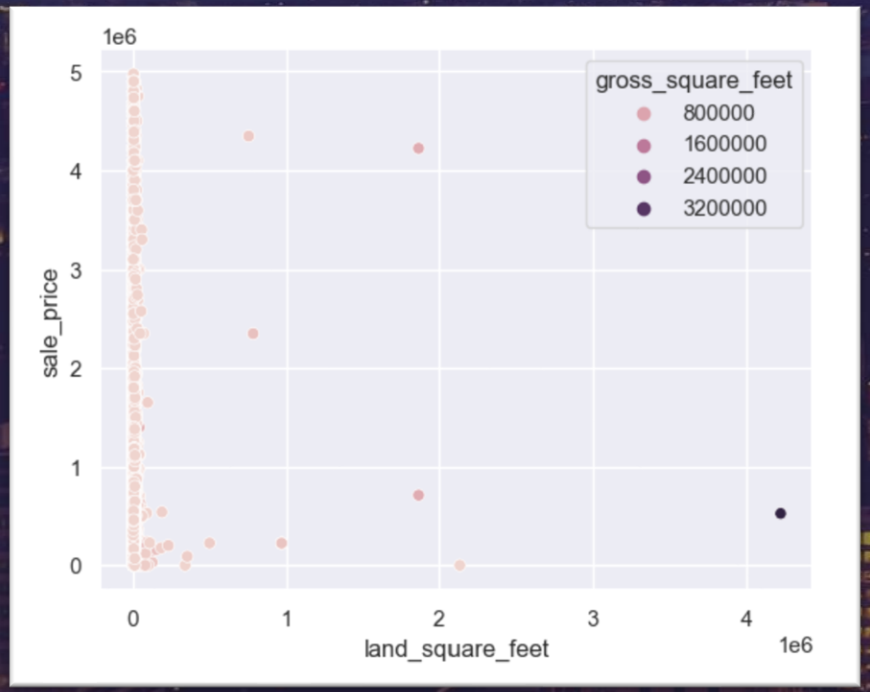
```
borough                          0
neighborhood                     0
building_class_category          0
tax_class_at_present             0
block                            0
lot                              0
ease-ment                        0
building_class_at_present        0
address                          0
apartment_number                 0
zip_code                         0
residential_units                0
commercial_units                 0
total_units                      0
land_square_feet             26252
gross_square_feet            27612
year_built                       0
tax_class_at_time_of_sale        0
building_class_at_time_of_sale   0
sale_price                   14561
sale_date                        0
dtype: int64
```

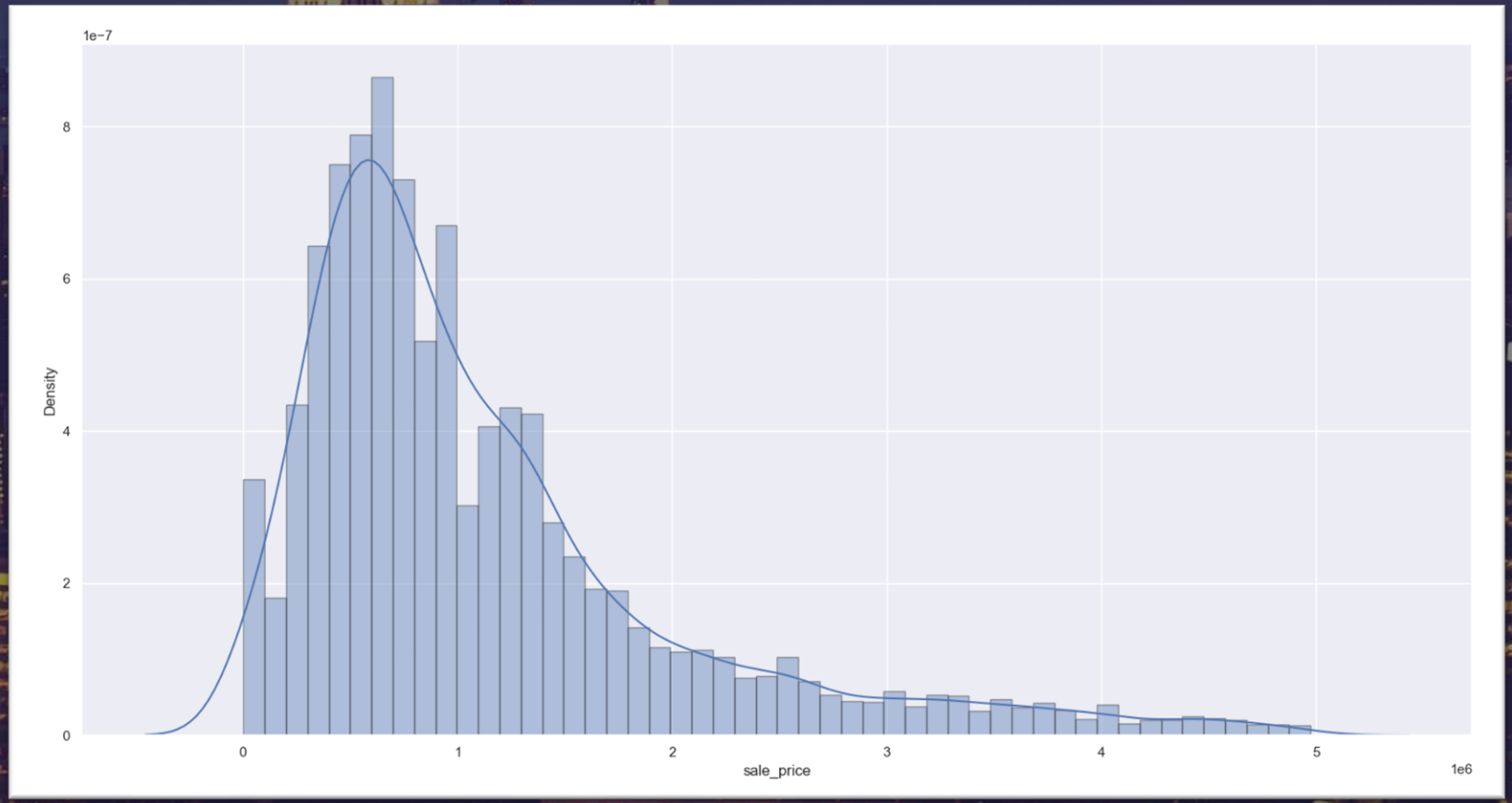
```
sns.scatterplot(data = NewYork_data, x="age", y="sale_price")
```

```
<Axes: xlabel='age', ylabel='sale_price'>
```



Data Aggregation & Representation

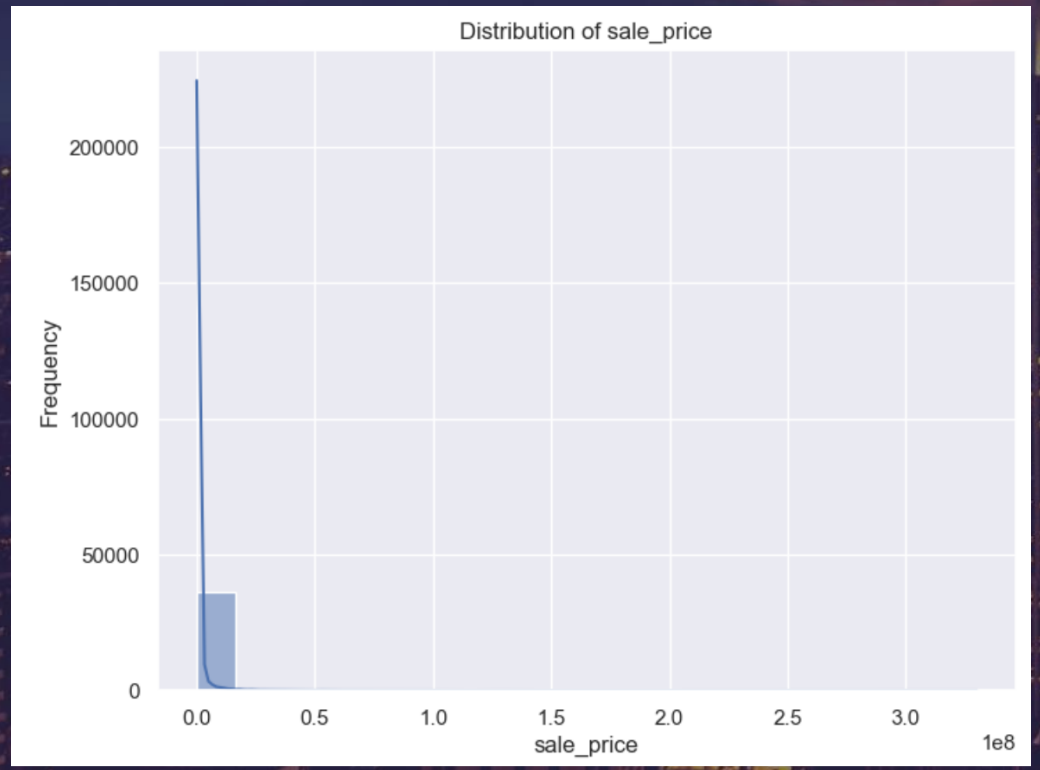


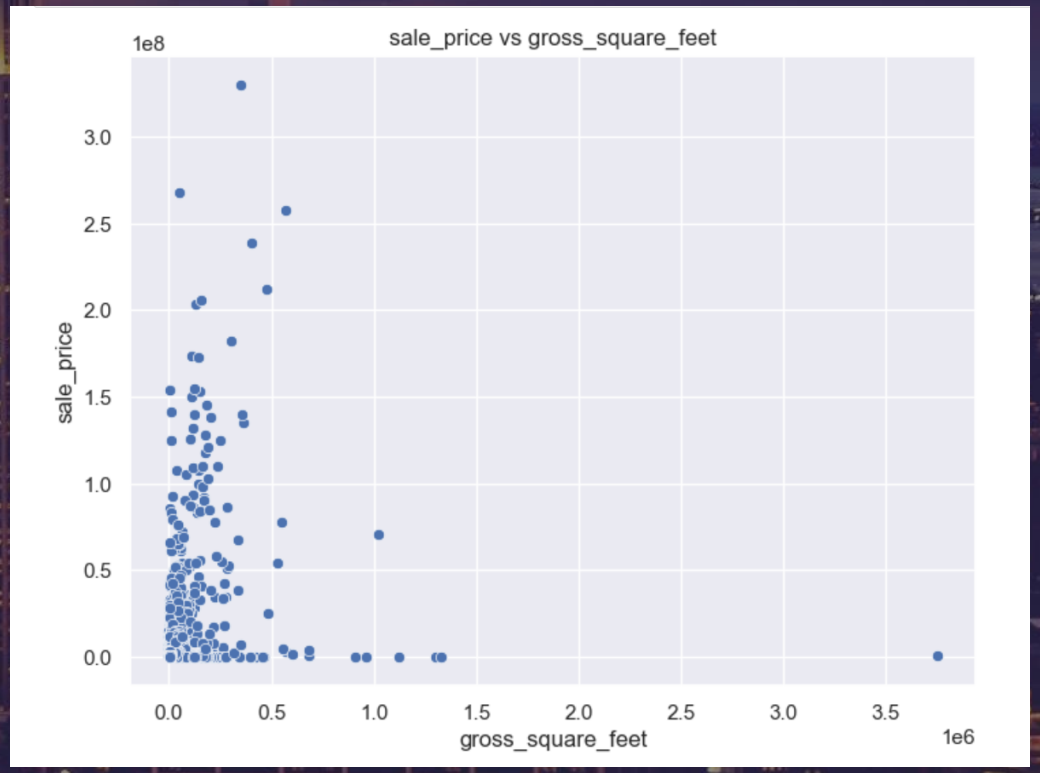


Data Analysis

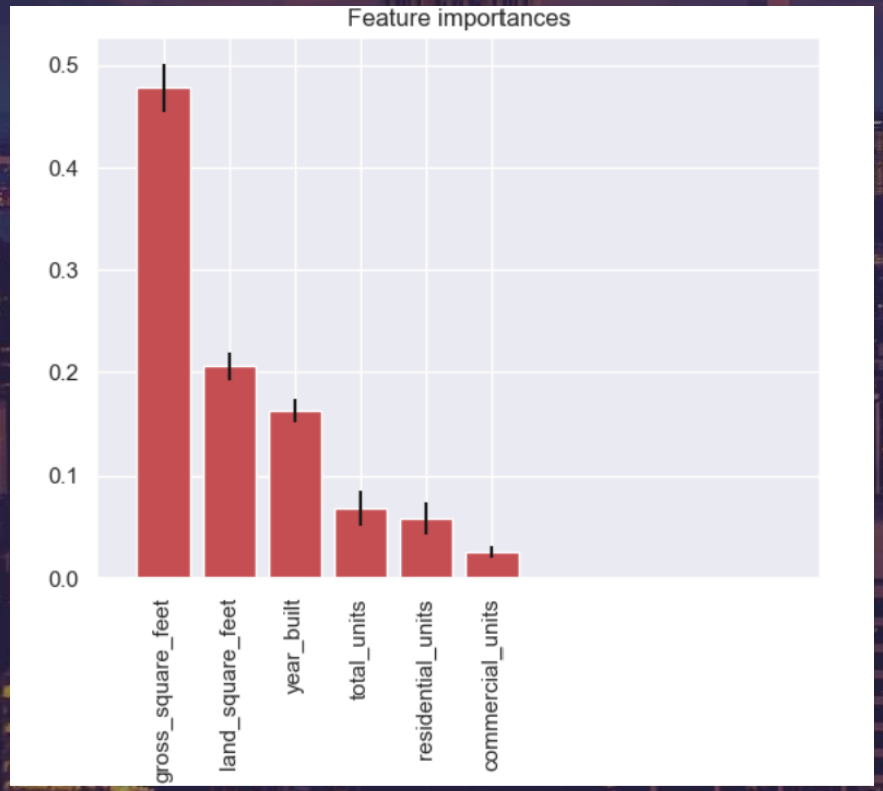
- Target variable exploration
- Numerical variables exploration
- Finding the correlation between variables
- Normalizing values using standard scaling, label encoding, or one-hot encoding
- Splitting the data into test and train subsets
- Build a machine-learning model
- Perform a cross-validation technique

Target variable exploration

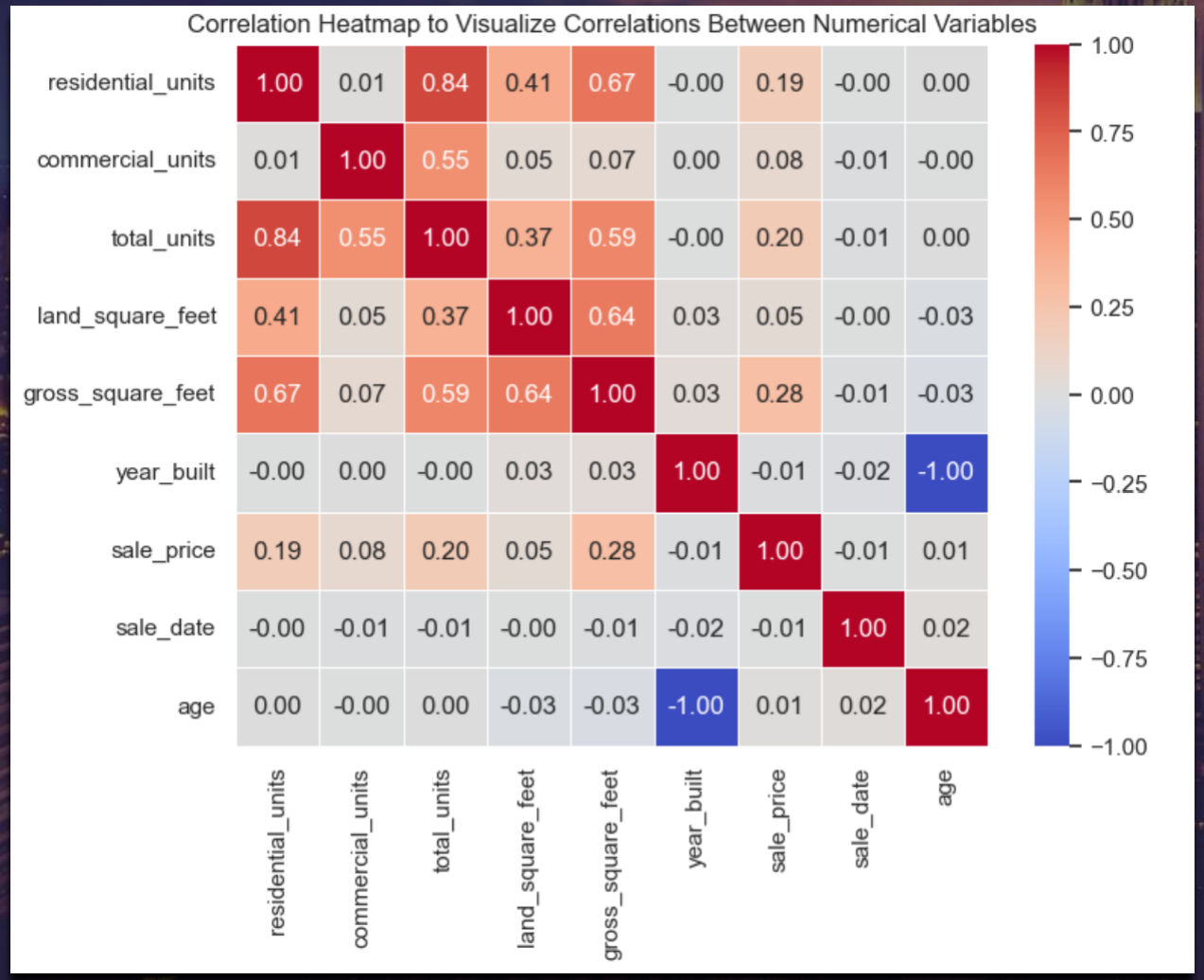




Numerical variables exploration

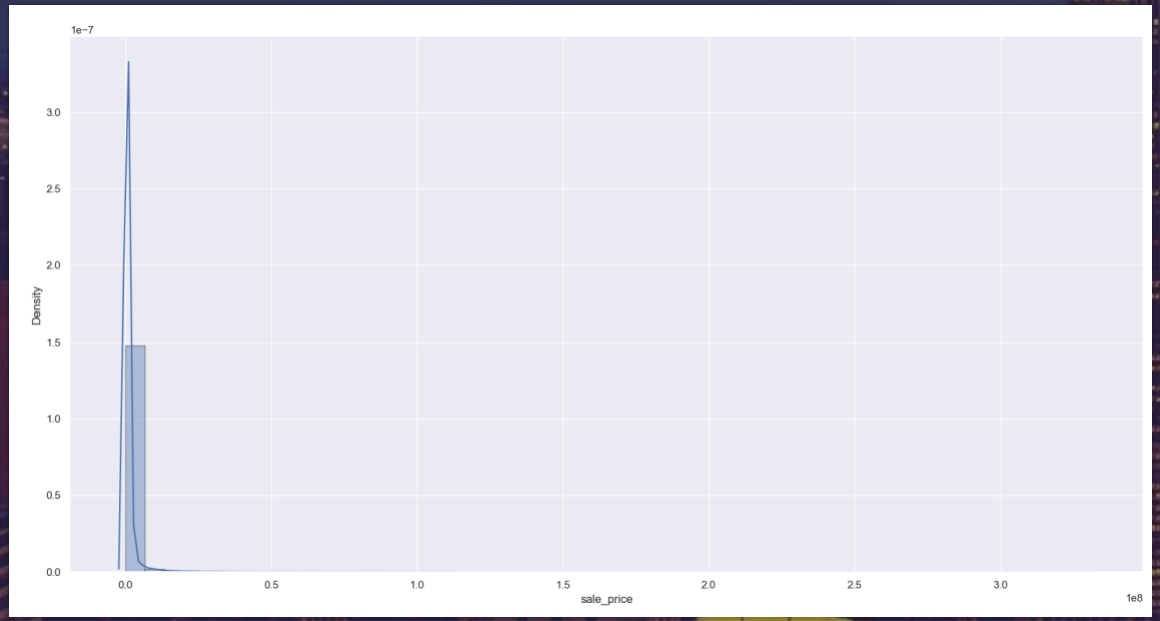


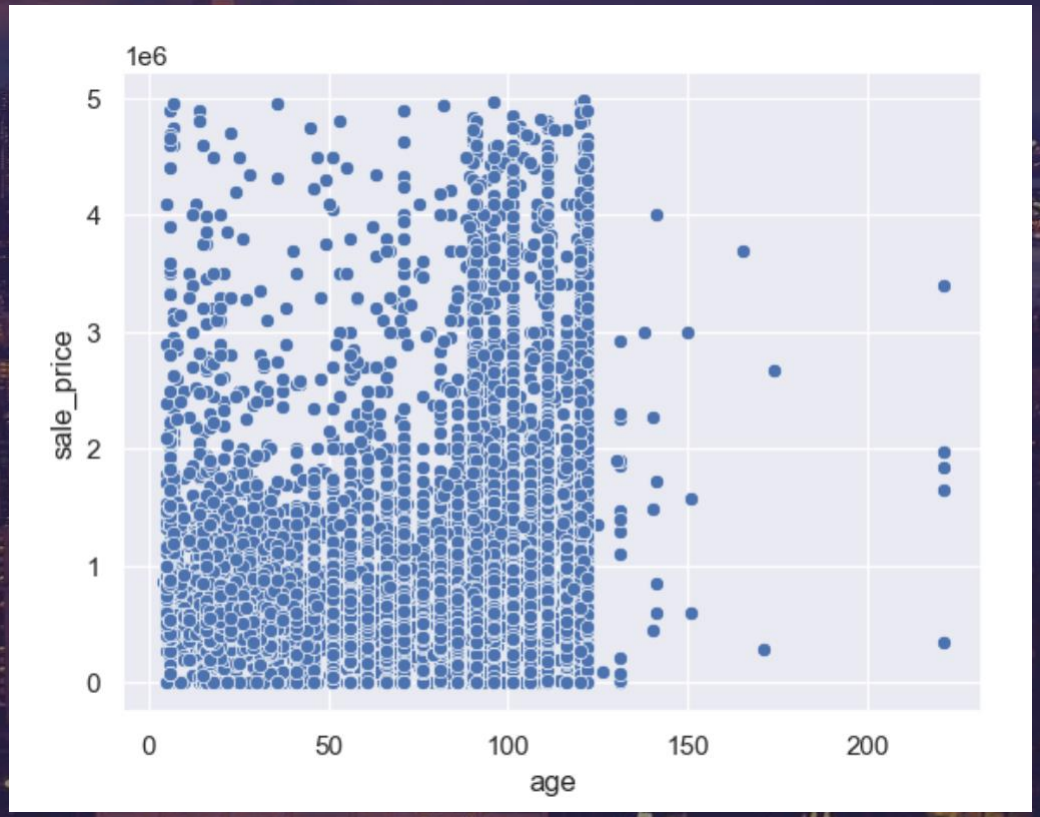
Finding the correlation between variables



- Cells with colors closer to 1 or -1 suggest strong correlations (positive or negative), whereas colors closer to 0 indicate weaker correlations.
- Understanding whether variables are positively or adversely related with one another is made easier with the help of this representation.

ED Analysis


1e−7
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Density
0.0
0.5
1.0
1.5
2.0
2.5
3.0
1e8
sale_price


1e6
5
4
3
2
1
0
sale_price
0
50
100
150
200
age

Splitting the data into test and train subsets

- Assesses the model's capacity to generalize.
- Keeps the model from overfitting by making sure it learns patterns that apply to new data instead of memorizing the training set.

```
#importing libraries

from sklearn.model_selection import train_test_split
training_data , testing_data = train_test_split(NewYork_data, test_size=0.3, random_state=39)
```

```
# Create features variable
X = new_data[['residential_units', 'commercial_units', 'total_units', 'land_square_feet',
              'gross_square_feet', 'year_built']]

# Create target variable
y = new_data['sale_price']

# Train, test, split
from sklearn.model_selection import train_test_split
X_tr, X_te, y_tr, y_te = train_test_split(X,y, test_size = 0.25, random_state= 10)
```

Linear Regression

- Coefficient (Slope): 331250.9124024468
- Intercept: 10859687.442460658
- MAE: 643203.9126975213
- MSE: 796704438398.4537
- RMSE on Train Set: 877834.9426623622
- RMSE on Test Set: 892583.0148498535
- R-squared on Train Set: 0.06612722349218891
- R-squared on Test Set: 0.07191242066418901

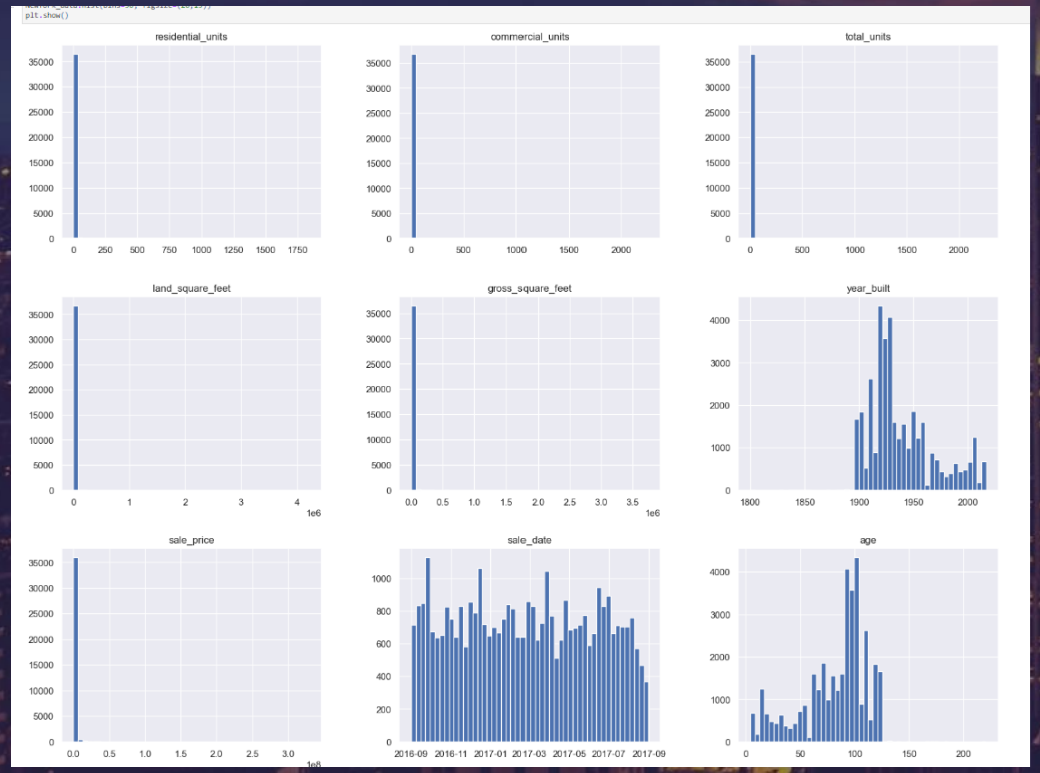
Random Forest Regressor

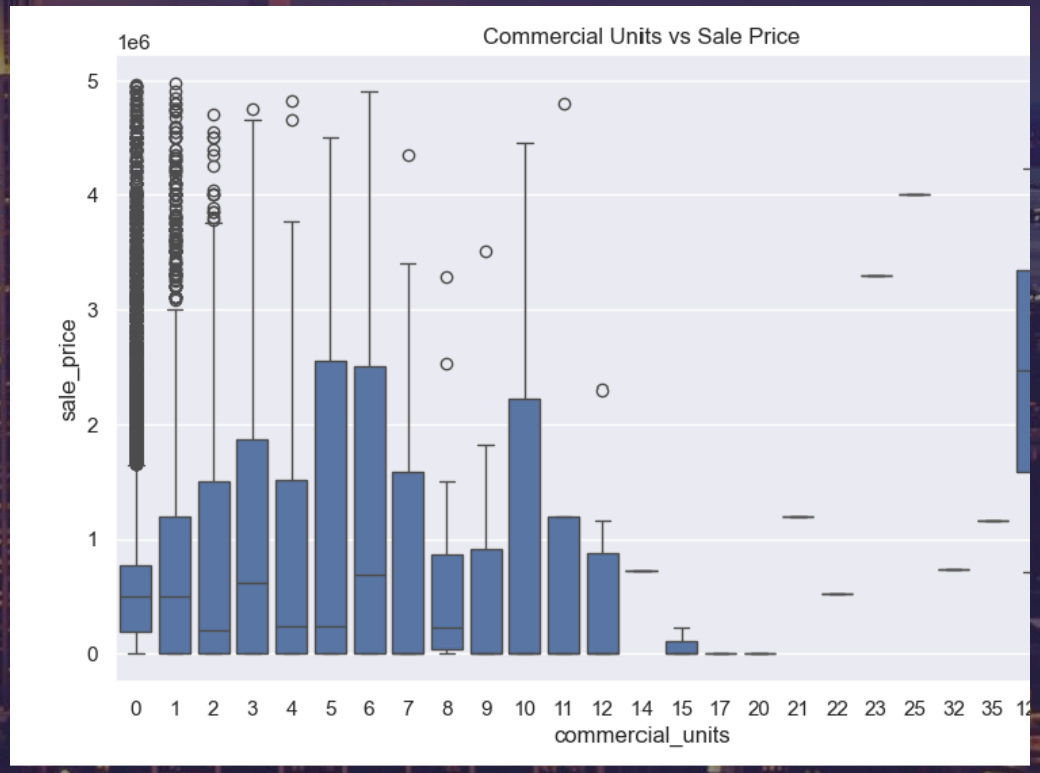
- R^2: 0.3473942966717847
- Root Mean Squared Error: 748478.914868223
- MAE: 643203.9126975213
- MSE: 796704438398.4537
- Average 5-Fold CV Score: 0.3461667312679969 [0.32506704 0.32413393 0.3571147 0.3765361 0.34798188]

Perform a cross-validation technique

- Use of Standard Scaler
- In Random Forest Regressor
- Model Accuracy with Scaling: 0.35299433681480363
- R-squared with Scaling: 0.35299433681480363

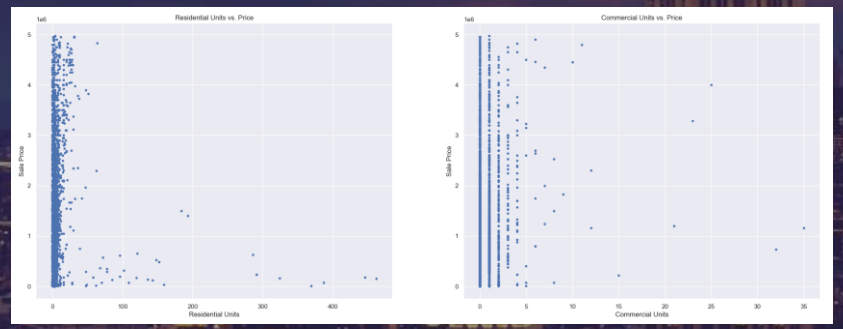
Data Visualization

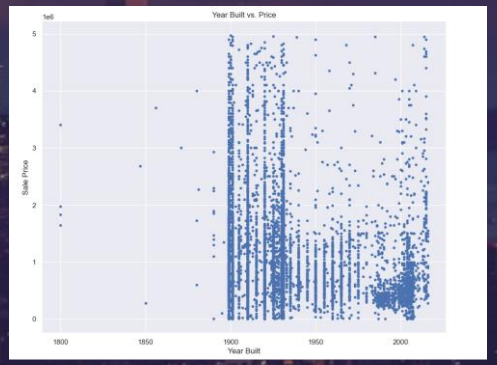


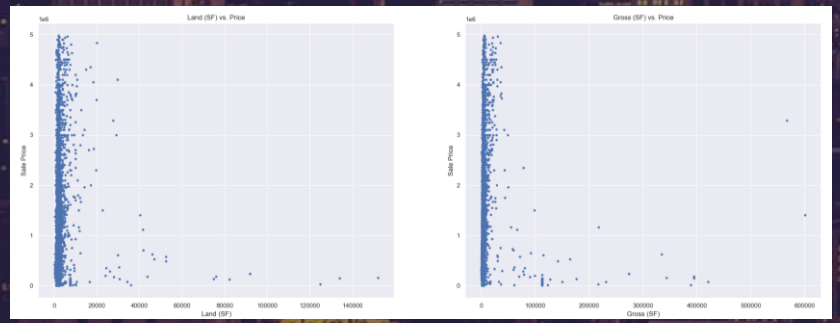


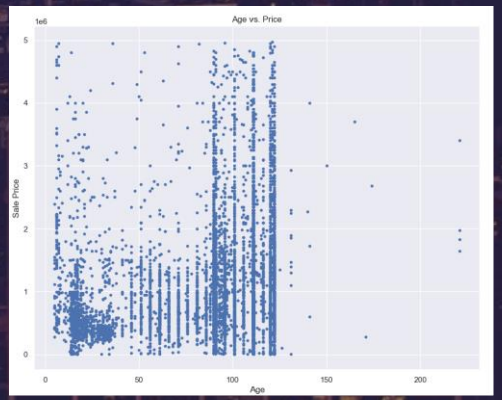
Data Visualization

Data Visualization

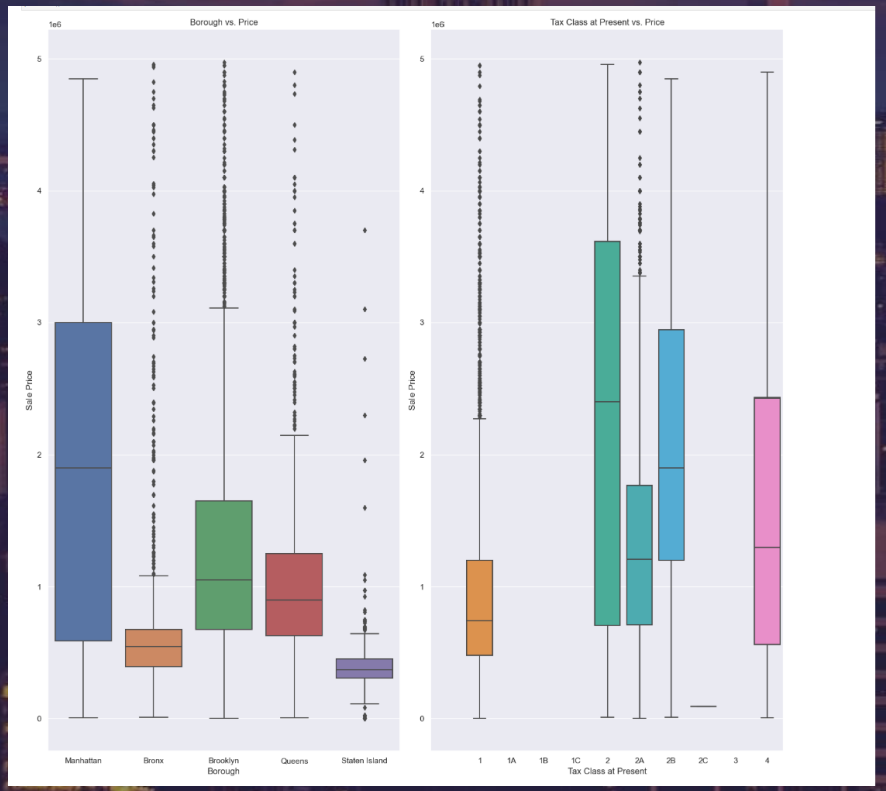








Data Visualization



Data Visualization

Utilization of Analysis Result

Every one unit:

- Residential Units is associated with a increase in Sale Price by $331250.91
- Commercial Units is associated with a increase in Sale Price by $454624.13
- Total Units is associated with a decrease in Sale Price by $-325424.49
- Land Square Foot is associated with a increase in Sale Price by $6.43
- Gross Square Foot is associated with a decrease in Sale Price by $-5.37
- Year Built is associated with a decrease in Sale Price by $-5049.75

REFERENCES

- https://www.ibm.com/topics/linear-regression
- https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html
- https://www.kaggle.com/datasets/new-york-city/nyc-property-sales
- https://www.geeksforgeeks.org/big-data-analytics-life-cycle/


ANY
QUESTIONS ?


THANK
YOU!!