Predictive Model for Manhattan House Prices

Aakash Sharma

October 12, 2023

1 Introduction

This document delves into an exploratory analysis of a property sales dataset, capturing various attributes such as neighborhood, building class, sale date, and more. The main objective is to derive meaningful insights, identify key predictors influencing property prices, and employ machine learning algorithms to build predictive models. Through this analysis, we aim to better understand the underlying patterns in property sales, the factors that significantly affect pricing, and the potential strategies for future property valuations.

2 Part 1 - Building up a basic predictive model

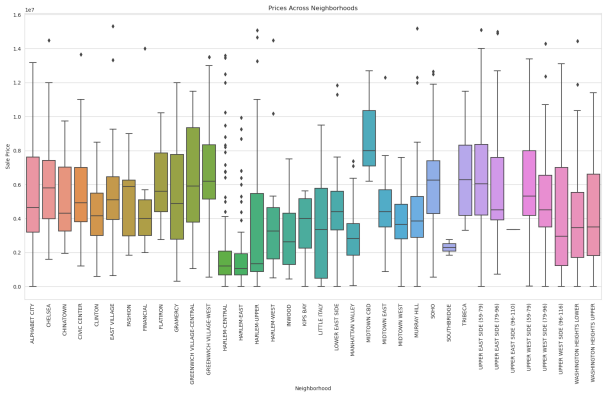
2.1 Data Cleaning and Transformation

- The dataset was loaded using `pd.read_csv(file_path, skiprows=4)` due to the presence of 4 irrelevant initial rows. The resulting shape was (27399, 21).
- Column names were adjusted to remove unnecessary newline characters using the method `df.columns.str.replace ' ').str.strip()`.
- Categorical and numerical variables were identified using the `select_dtypes` function. This separation facilitated the subsequent data transformation steps.
- For numerical data transformation, columns were converted to appropriate numerical formats. The 'SALE DATE' column was converted to a datetime format, enhancing its utility for future operations.
- Categorical variables underwent cleaning, where empty and NaN values were addressed.
- Columns 'BOROUGH', 'EASE-MENT', and 'APARTMENT NUMBER' were deemed unnecessary and were dropped. Additionally, rows with missing values, outliers, and duplicates were removed. Post these operations, the dataset's shape was streamlined to (1125, 18).
- To aid in subsequent analysis, the 'SALE PRICE' column was logarithmically transformed, resulting in the 'LOG_SALE_PRICE' column. This was further normalized to produce the 'NORM_LOG_SALE_PRICE' column using the formula:

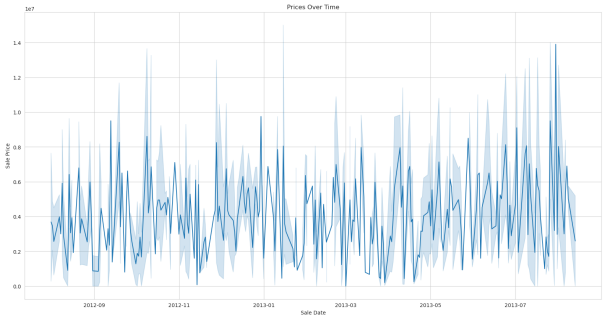
$$\text{NORM\_LOG\_SALE\_PRICE} = \frac{\text{LOG\_SALE\_PRICE} - \text{mean(LOG\_SALE\_PRICE)}}{\text{std(LOG\_SALE\_PRICE)}}$$

2.2 Data Exploration

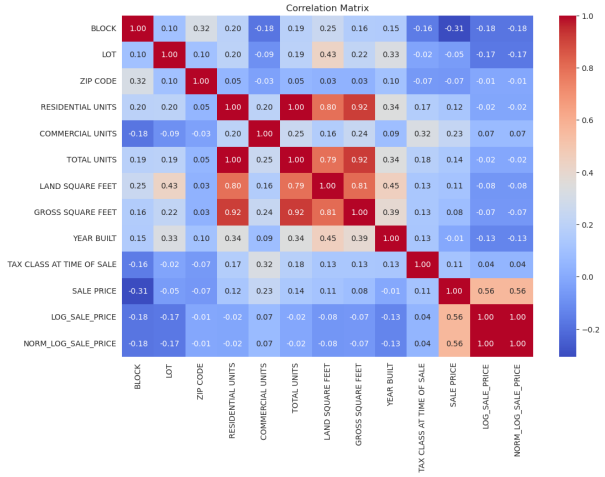
- **Prices Across Neighborhoods:** The box plot illustrates a variation in house prices across different neighborhoods. Some neighborhoods like "Midtown CBD" and "SoHo" appear to have higher median prices, while neighborhoods like "Harlem East" and "Greenwich Village-West" display a broader price range.



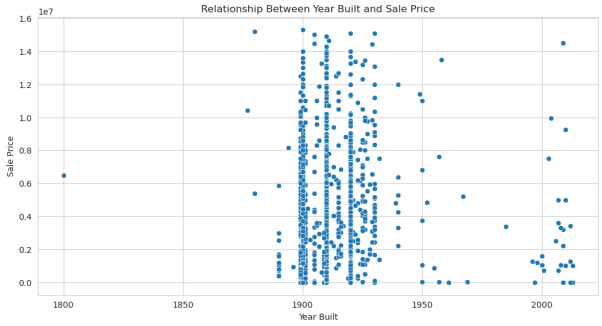
- **Prices Over Time:** The time series plot shows fluctuations in house prices over time, but does not reveal a clear trend.



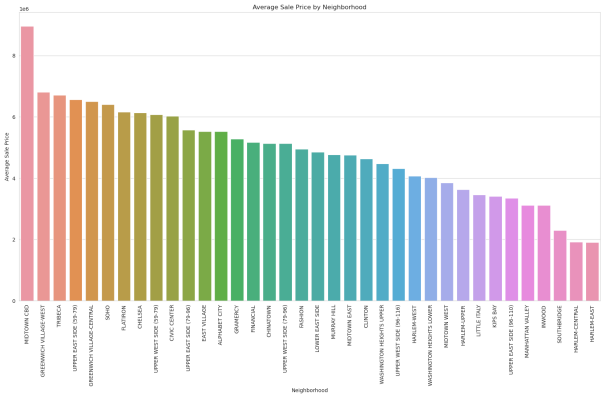
- **Correlation Matrix:** The heatmap of the correlation matrix assists in identifying potential relationships between variables. Notably, there's a high correlation between "Gross Square Feet" and "Total Units," hinting at larger buildings having more units.



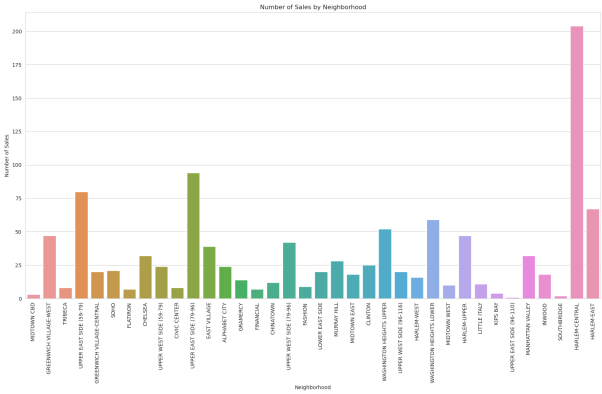
- **Relationship Between Year Built and Sale Price:** The scatter plot of the year a property was built against its sale price suggests no strong trend between the age of a property and its price.



- **Average Sale Price by Neighborhood:** The bar chart represents the average sale prices of properties across various neighborhoods. Notably, "Midtown CBD" leads with the highest average price, followed closely by "Tribeca" and "SoHo". On the contrary, areas such as "Harlem-West" and "Southbridge" display lower average prices.



- **Number of Sales by Neighborhood:** The chart depicts the number of property sales in different neighborhoods. It's evident that "Harlem-East" had the highest number of sales, significantly surpassing other areas, while neighborhoods like "Greenwich Village-Central" and "Midtown CBD" had a comparatively lesser number of transactions.

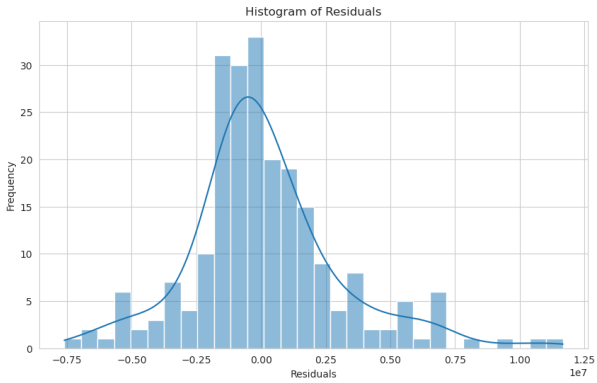


2.3 Model Building

In the model building phase:

- **Algorithm Selection:** We transformed categorical columns `NEIGHBORHOOD` and `BUILDING CLASS CATEGORY` using the `get_dummies` method. This helped in converting categorical data into a format suitable for linear modeling.
- **Data Splitting:** We utilized the `train_test_split` method from `sklearn.model_selection` to partition the dataset into training and testing sets.

- **Linear Regression Model:** For our baseline model, a linear regression was applied. The resulting mean squared error (MSE) was an overwhelming 8.55 trillion, indicating room for improvement. The histogram of residuals for this model is shown below:

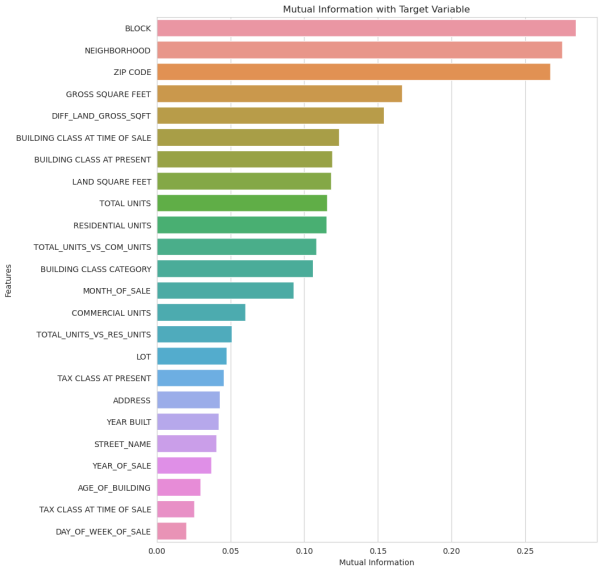


- **Advanced Models:** Next, we tested more sophisticated models. The RandomForestRegressor yielded an MSE of 6.22 trillion, and the GradientBoostingRegressor produced an MSE of 6.45 trillion. The performance was consistent when tested with XGBoost.

3 Part 2 - Improved Model

3.1 Building an Improved Model

- **Feature Engineering:** Introduced new features such as `TOTAL_UNITS_VS_RES_UNITS`, `TOTAL_UNITS_VS_COM_UNITS`, and `DIFF_LAND_GROSS_SQFT` to capture differences between units and square footage. Extracted the month, year, and day of the week from the `SALE DATE`. Furthermore, isolated the `STREET_NAME` from the address. Their importance was guaged using mutual info graph.

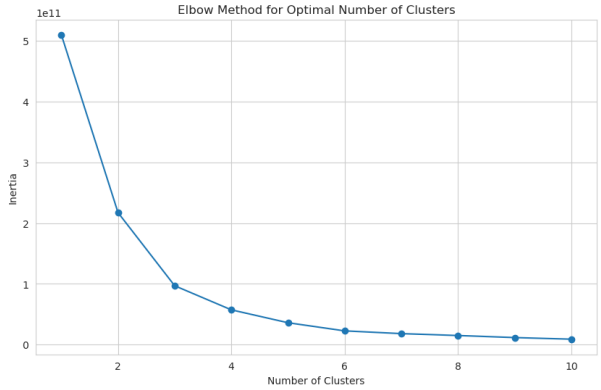


- **Improved Model Structure:** Ordinal encoding was used on categorical variables. The mutual importance of the new features with the target variable was evaluated. The most significant improvement was observed with the Random Forest algorithm, after hyperparameter tuning, which achieved an MSE of 6.33×10^{12}.

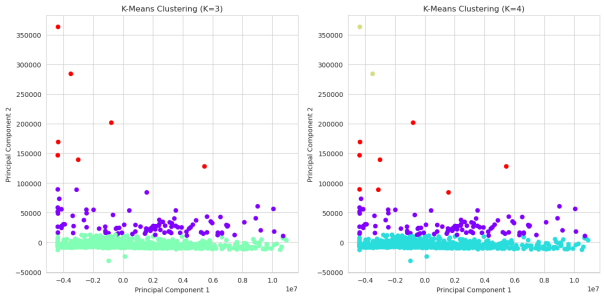
- **Model Evaluation:** The Random Forest model, after tuning, outperformed the basic model with an R^2 of 0.405. A stacking model approach was tried with various base models and Ridge as the final estimator; however, the performance remained consistent.

3.2 K-Means Clustering

- **Strategy:** The optimal number of clusters was determined to be 3 using the elbow method.



- **Visualization of Clusters:** After applying PCA, distinct clusters were observed in the dataset as illustrated in the provided graph.



- **Insights:** The clustering revealed specific patterns in the dataset, allowing for a more tailored modeling approach for different segments of the data.

3.3 Clusters-based Regression

- **Local Regressors:** Separate regressors were constructed based on the clustering results, leading to models tailored to each cluster's unique properties.
- **Comparison:** The clusters-based regression model for Cluster 3 demonstrated better performance with an MSE of 5.995×10^{12} and an R^2 of 0.425, suggesting the effectiveness of this segmented approach.

4 Conclusion

In conclusion, the introduction of new features and a more tailored approach using clustering significantly improved the model's performance. By understanding the patterns within clusters, we could achieve a better fit for specific segments of the dataset. This tailored approach not only provides more accurate predictions but also reveals underlying patterns in the data. The findings suggest that a combination of feature engineering and segment-specific modeling can significantly enhance predictive accuracy. Future work can delve deeper into understanding the nuances within each cluster to further refine the models.