Problem Statement

Title : "Predicting Perth House Prices: Enhancing Real Estate Decision-Making”

Problem Statement: The problem at hand is the significant variation in house prices across different neighborhoods in Perth (relevant to other places as well provided the data is available), driven by factors like proximity to amenities, property attributes, and land area.

This project aims to answer the following key questions:

- How can we develop a near accurate machine learning-based model to predict house prices i.e. are they highly, lowly, or moderately priced in various Perth neighborhoods based on relevant factors?
- What is the significance of accurate house price predictions for both buyers and sellers in making informed real estate transactions?
- How can market transparency be improved through reliable price predictions, and what benefits does this bring to the real estate market?

Problem Background: House prices in Perth vary significantly due to factors like location, property features, and land size. Buyers cannot always have the transparency on price that the seller is deciding.

Project Contribution: Our machine learning model tailored to Perth will use factors like proximity to school and station, property details, and land size to provide:

- Accurate Pricing: Empowering buyers and sellers to reduce financial risks.
- Market Efficiency: Enhancing transparency and reducing price disparities.

In conclusion, our project plays a pivotal role in addressing Perth's house price prediction challenges, benefiting the buyers and sellers to predict the price of house in Perth based on the analysis on the dataset provided by the government.

Data Sources

Link of the dataset - https://www.kaggle.com/datasets/syuzai/perth-house-prices
Dataset Shape - 33656 rows x 19 columns

Features of the dataset:
ADDRESS : Physical address of the property (we will set to index)
SUBURB : Specific locality in Perth; a list of all Perth suburb can be found here
PRICE : Price at which a property was sold (AUD)
BEDROOMS : Number of bedrooms
BATHROOMS : Number of bathrooms
GARAGE : Number of garage places
LAND_AREA : Total land area (m^2)
FLOOR_AREA : Internal floor area (m^2)
BUILD_YEAR : Year in which the property was built
CBD_DIST : Distance from the center of Perth (m)
NEAREST_STN : The nearest public transport station from the property
NEAREST_STN_DIST : The nearest station distance (m)
DATE_SOLD : Month & year in which the property was sold
POSTCODE : Local Area Identifier
LATITUDE : Geographic Location (lat) of ADDRESS
LONGITUDE : Geographic Location (long) of ADDRESS
NEAREST_SCH : Location of the nearest School
NEAREST_SCH_DIST : Distance to the nearest school
NEAREST_SCH_RANK : Ranking of the nearest school

Making the dataset noisy -

Steps performed for uncleaning of the data (added noise) :

1. Added 1000 duplicate entries as part noise addition to the dataset and shuffled the data.

2. Added random number (any number between 2000-3000) of null values for columns 'PRICE', 'BEDROOMS', 'BATHROOMS', 'GARAGE', 'LAND_AREA', 'FLOOR_AREA', 'BUILD_YEAR', 'CBD_DIST', 'NEAREST_STN_DIST', 'NEAREST_SCH_DIST'.

Final shape of the dataset after adding noise - 34656 rows x 19 columns

Note - The kaggle dataset and noisy dataset with the names "Kaggle_Dataset_Perth_Housing.csv" and "uncleaned_dataset.csv" have been added as source files, along with the python code file for uncleaning titled "uncleaning.ipynb.

Data Cleaning / Processing

Steps :

1. Dropping duplicate rows based on the same feature values for ‘ADDRESS’, ‘SUBURB’, ‘LATITUDE’ and ‘LONGITUDE’ after which we were left with 33656 rows x 19 columns.
2. Dropped features that are not required like ‘ADDRESS’, ‘SUBURB’, ‘NEAREST_STN’, ‘NEAREST_SCH’, ‘NEAREST_SCH_RANK’ as they were irrelevant for further analysis. After which we were left with 33656 rows x 14 columns.
3. Only considered the year value from the DATE_SOLD feature (later dropped) and made a new feature SOLD_YEAR as the BUILD_YEAR feature only contains years, for the ease of comparison.
4. Filling missing values :

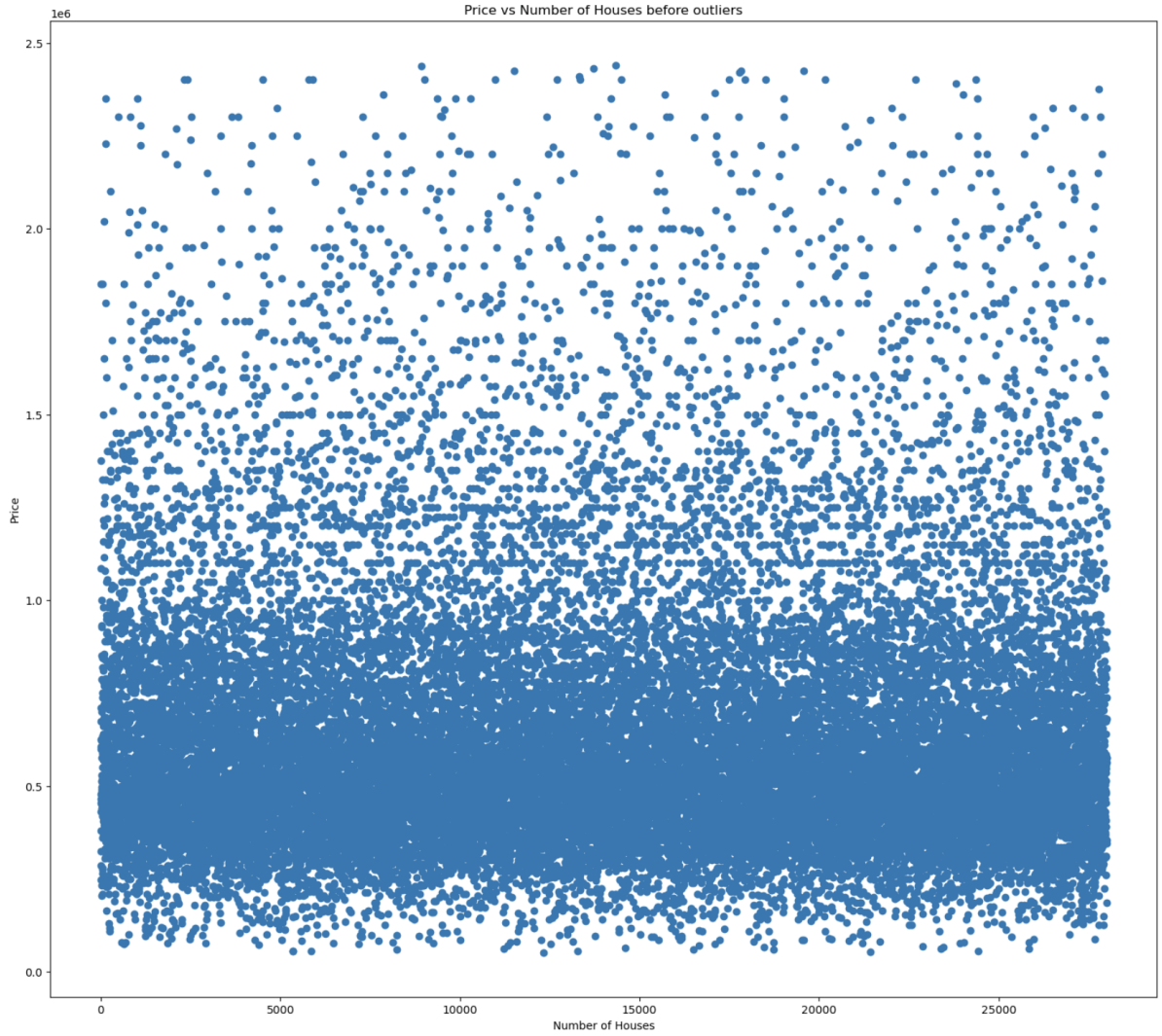
    a. Filled the missing values of features ‘PRICE’, ‘BEDROOMS’, ‘BATHROOMS’, ‘LAND_AREA’ and ‘FLOOR_AREA’ using KNN imputation (selected these columns assuming that they are related to each other).
    b. Filled missing values of feature ‘GARAGE’ with mode value.
    c. Filled missing values of each feature ‘CBD_DIST’, ‘NEAREST_STN_DIST’ and ‘NEAREST_SCH_DIST’ based on ‘LATITUDE’ and ‘LONGITUDE’ using KNN imputation.
5. As KNN imputed float values for features ‘BEDROOMS’ and ‘BATHROOMS’, these values were transformed to floor values in this step.
6. After filling the majority of the missing values, the remaining rows with NA values were dropped. After these operations the shape of the dataset was 28010 rows x 14 columns.
7. Added ‘PRICE_CATEGORY’ feature with HIGH, LOW and MEDIUM values based on the interquartile range of the ‘PRICE’ feature.
8. Performed one hot encoding for ‘PRICE_CATEGORY’ feature for the ease of classification.
9. Converted data type of ‘BUILD_YEAR’ feature to int as it cannot be a float value.
10. Removal of outliers (performed in combination with EDA). Shape after removing the outliers 27017 rows × 17 columns.
11. Normalize the data for features "PRICE", "BEDROOMS", "BATHROOMS", "GARAGE", "LAND_AREA", "FLOOR_AREA", "CBD_DIST", "NEAREST_STN_DIST" and "NEAREST_SCH_DIST" for the ease of computation. (Performed as the last step of cleaning after EDA)

Exploratory Data Analysis (EDA)

Steps :

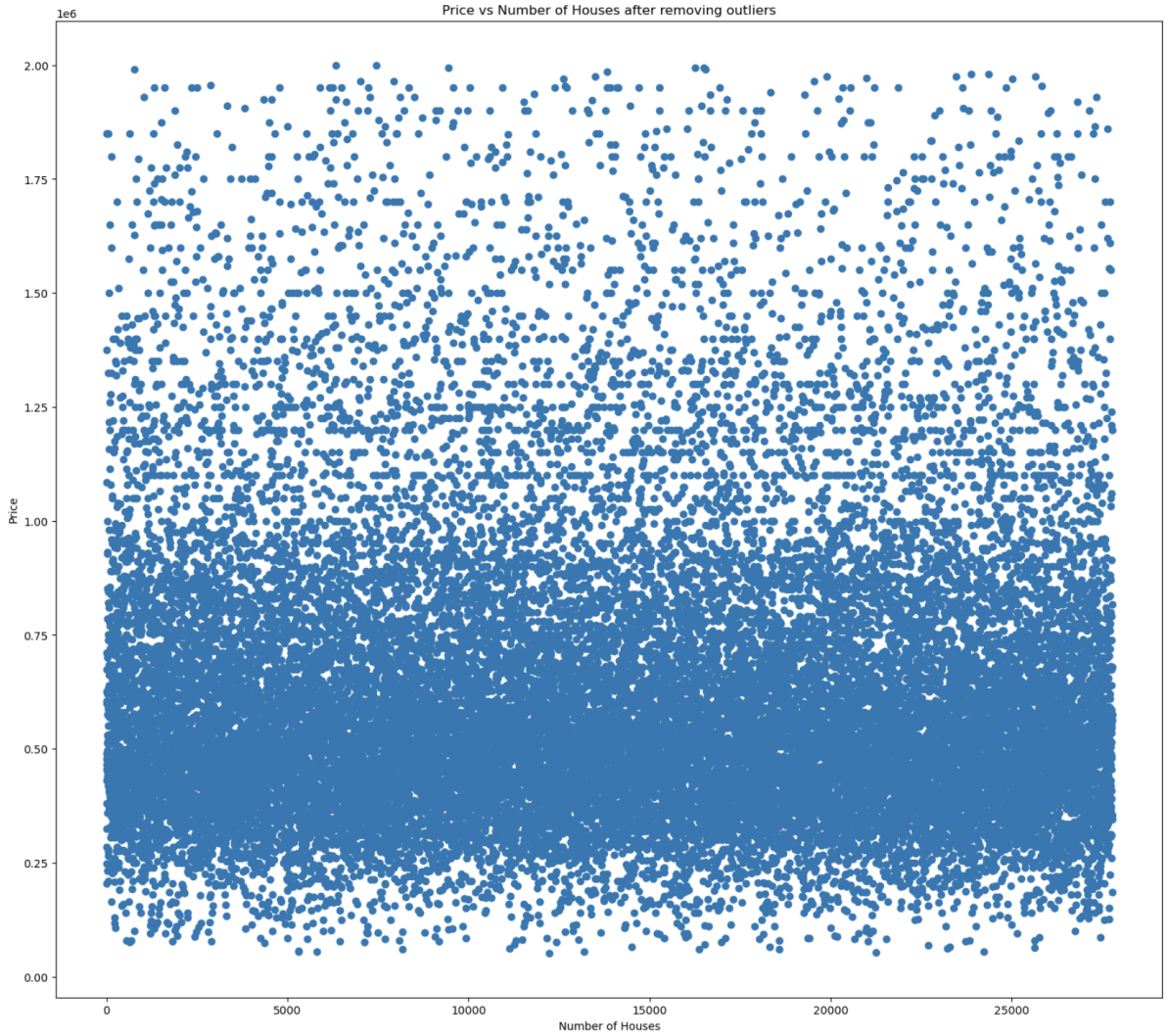
1. Visualizing outliers using scatter plots

Scatter plot for ‘PRICE’

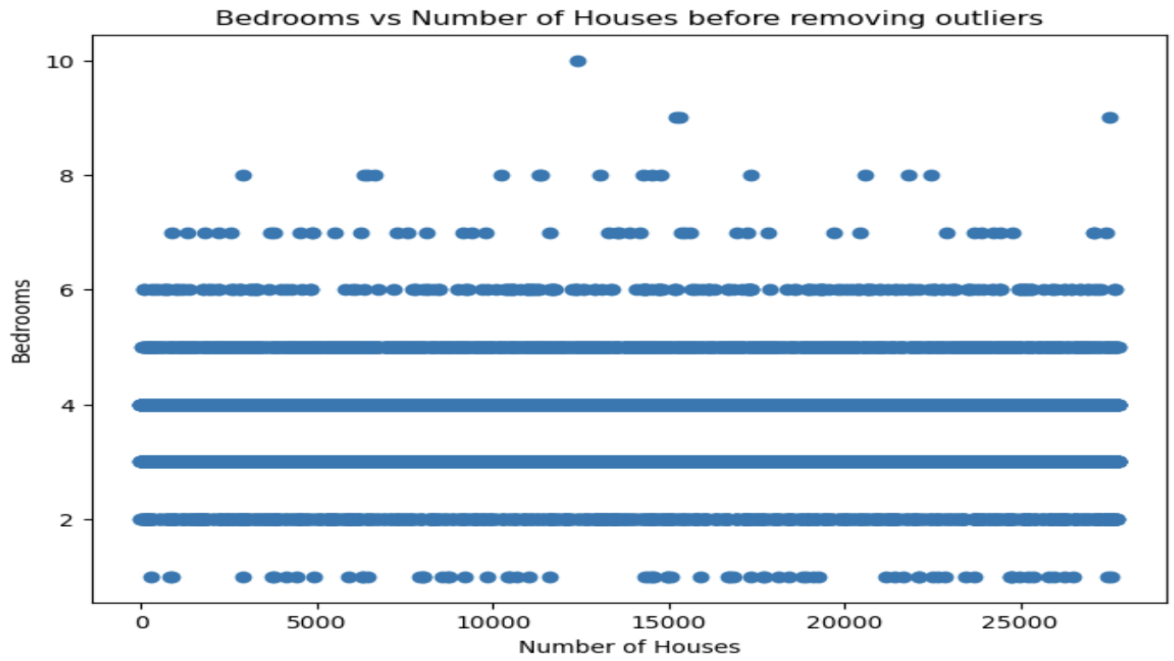


After observing the scatter plot we considered values greater than 2e6 as outliers and removed them.

Scatter plot after removal of outliers for ‘PRICE’

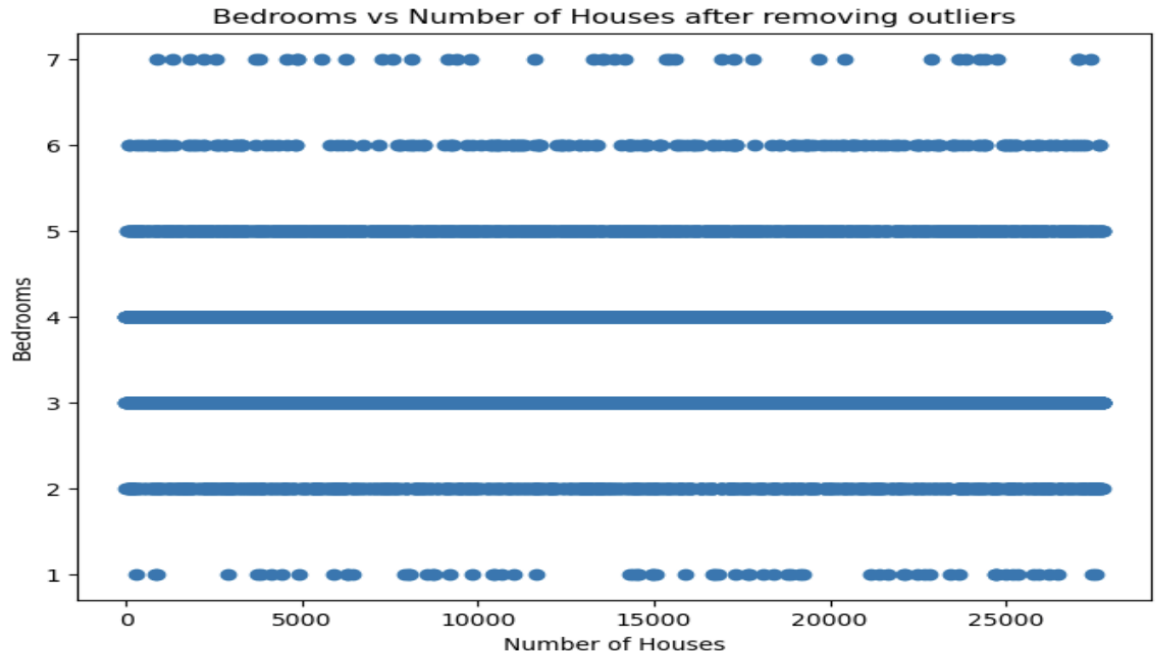


Scatter plot for ‘BEDROOMS’



After observing the scatter plot we considered values greater than 7 as outliers and removed them.

Scatter plot after removal of outliers for ‘BEDROOMS’



Similarly visualized, identified and removed the outliers for features ‘BATHROOMS’, ‘GARAGE’, ‘LAND_AREA’, ‘FLOOR_AREA’, ‘BUILD_YEAR’, ‘CBD_DIST’, ‘NEAREST_STN_DIST’, ‘LATITUDE’, ‘LONGITUDE’, ‘NEAREST_SCH_DIST’ and ‘SOLD_YEAR’.

2. Checked statistics of the dataset after cleaning it.

```
dataset.describe()
```

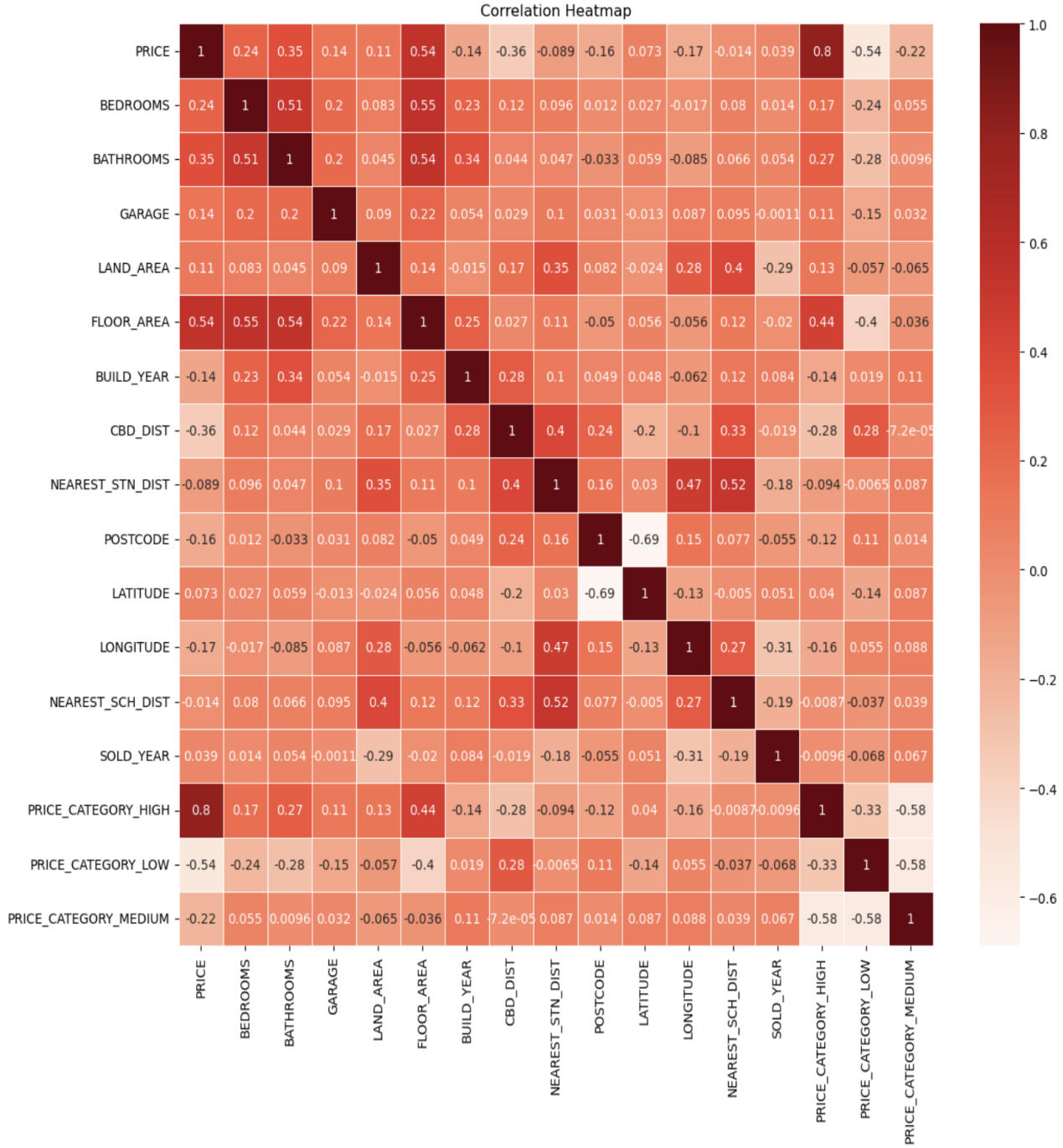
	PRICE	BEDROOMS	BATHROOMS	GARAGE	LAND_AREA	FLOOR_AREA	BUILD_YEAR	CBD_DIST	NEAREST_STN_DIST	POSTCODE
count	2.701700e+04	27017.000000	27017.000000	27017.000000	27017.000000	27017.000000	27017.000000	27017.000000	27017.000000	27017.000000
mean	6.237420e+05	3.614280	1.766147	2.128253	1766.162950	179.746266	1990.083429	19472.534959	4270.415786	6088.717067
std	3.129853e+05	0.721754	0.573400	0.948570	4629.689786	64.650002	20.274130	11163.125818	4004.758570	59.846854
min	5.300000e+04	1.000000	1.000000	1.000000	61.000000	1.000000	1901.000000	681.000000	46.000000	6003.000000
25%	4.150000e+05	3.000000	1.000000	2.000000	510.000000	130.000000	1979.000000	11100.000000	1700.000000	6038.000000
50%	5.430000e+05	4.000000	2.000000	2.000000	684.000000	171.000000	1995.000000	17200.000000	3100.000000	6069.000000
75%	7.500000e+05	4.000000	2.000000	2.000000	845.000000	218.000000	2005.000000	26100.000000	5157.142857	6150.000000
max	1.999990e+06	7.000000	4.000000	8.000000	50000.000000	450.000000	2017.000000	58100.000000	29700.000000	6558.000000

```
dataset.describe()
```

POSTCODE	LATITUDE	LONGITUDE	NEAREST_SCH_DIST	SOLD_YEAR	PRICE_CATEGORY_HIGH	PRICE_CATEGORY_LOW	PRICE_CATEGORY_MEDIUM
27017.000000	27017.000000	27017.000000	27017.000000	27017.000000	27017.000000	27017.000000	27017.000000
6088.717067	-31.960321	115.876569	1.678757	2016.820002	0.244661	0.246697	0.508643
59.846854	0.175431	0.115668	1.361687	3.003870	0.429894	0.431097	0.499935
6003.000000	-32.455550	115.583610	0.070912	2000.000000	0.000000	0.000000	0.000000
6038.000000	-32.067380	115.789050	0.866417	2016.000000	0.000000	0.000000	0.000000
6069.000000	-31.933030	115.851990	1.313455	2017.000000	0.000000	0.000000	1.000000
6150.000000	-31.844300	115.965115	2.012603	2019.000000	0.000000	0.000000	1.000000
6558.000000	-31.465818	116.298804	10.819908	2020.000000	1.000000	1.000000	1.000000

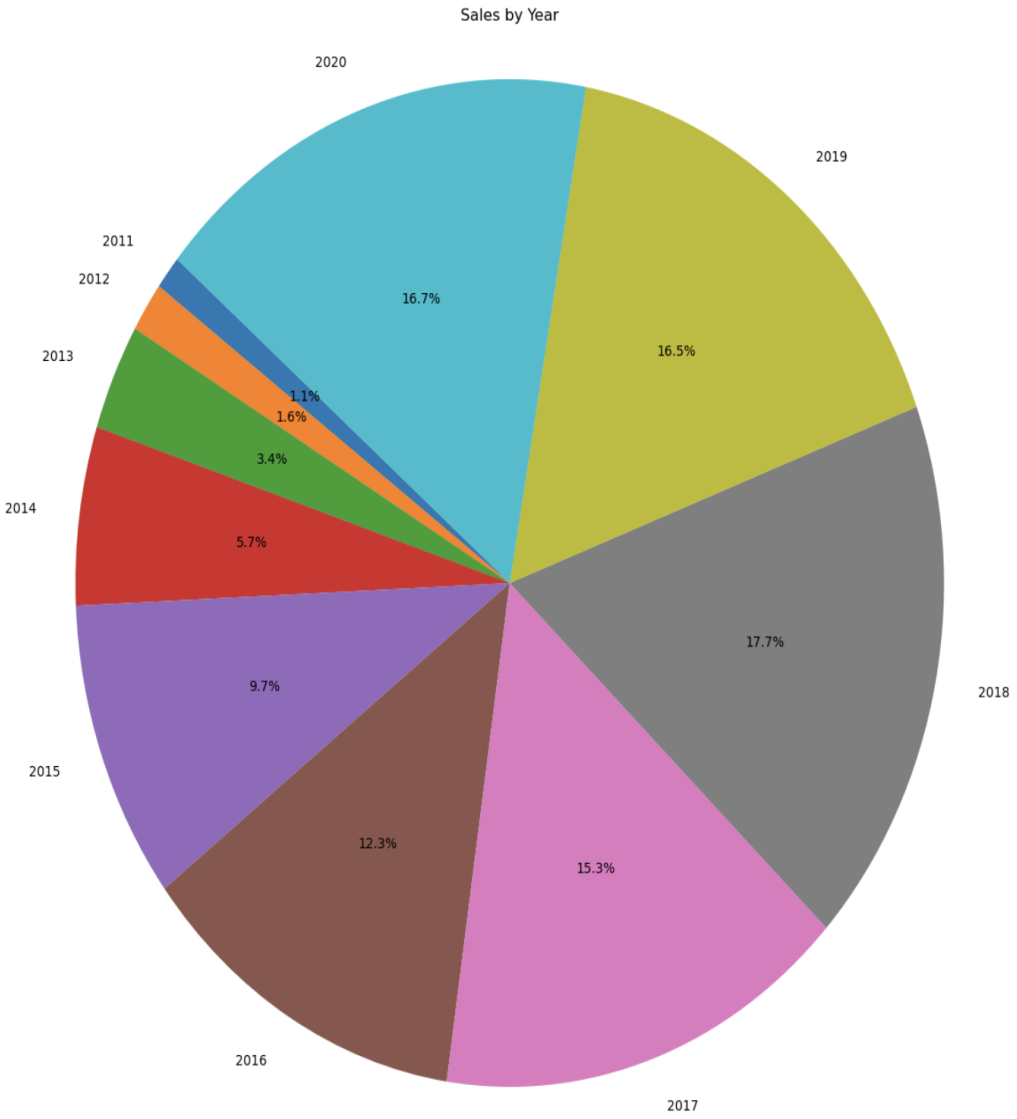
Used the min, 25%, 50%, 75% and max values to create PRICE_CATEGORY feature.

3. Displayed the correlation matrix



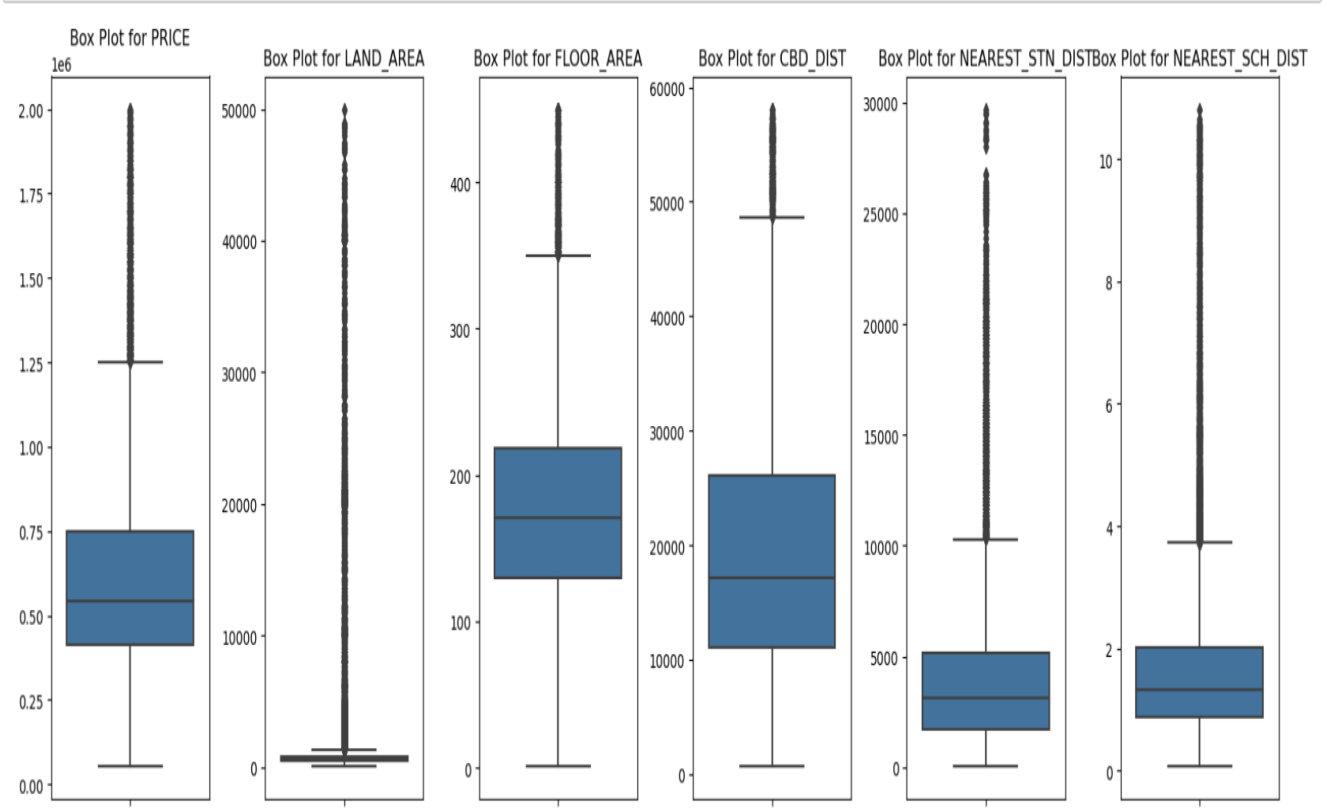
No features were highly correlated to each other so no features were dropped in this step.

4. Displayed the pie chart for sales of houses from the year 2011-2020



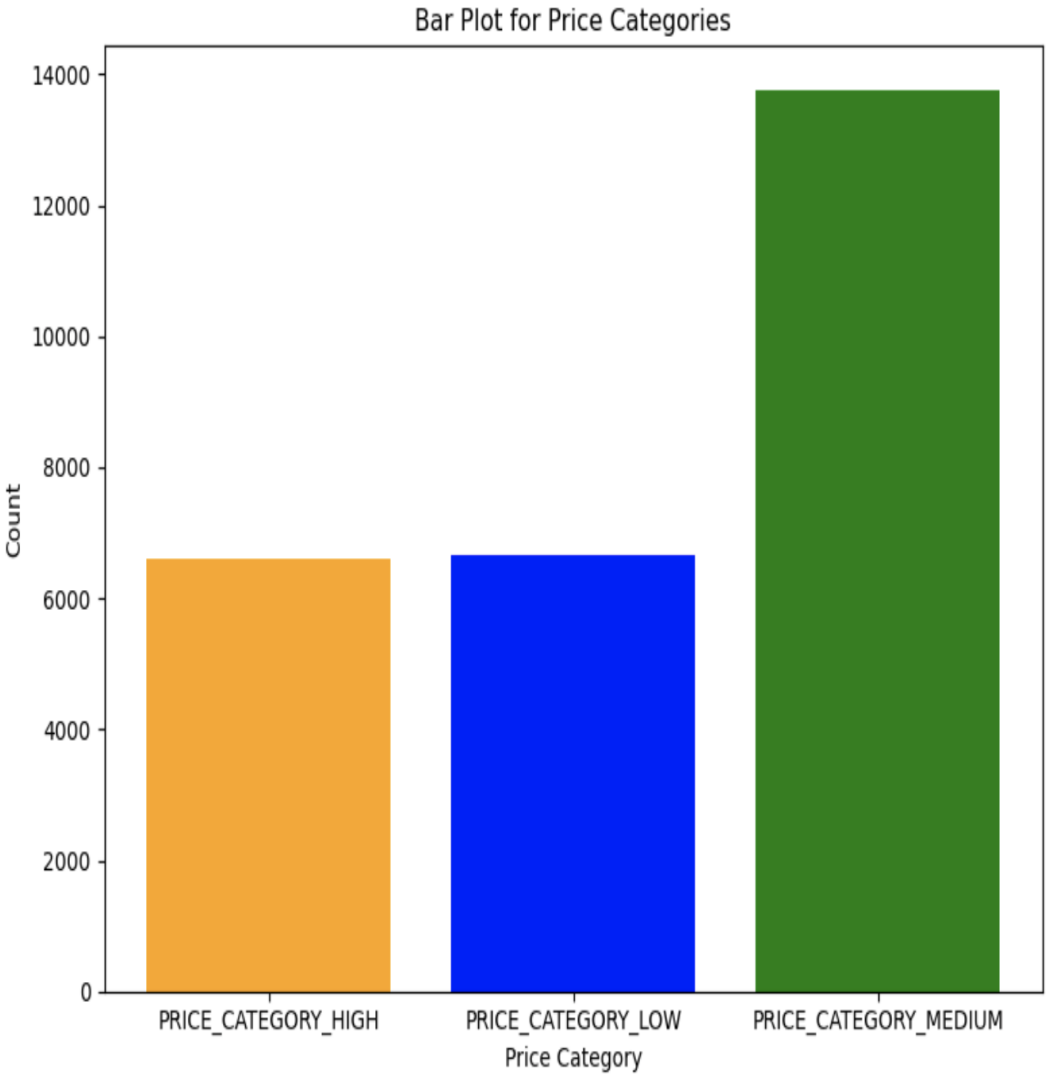
This visualization gave us an idea of the percentage of houses which were sold from 2011-2020 in a particular year. There was a steady increase in the sales of houses as the years progressed.

5. Displaying Box plots for 'PRICE', 'LAND_AREA', 'FLOOR_AREA', 'CBD_DIST', 'NEAREST_STN_DIST', 'NEAREST_SCH_DIST'



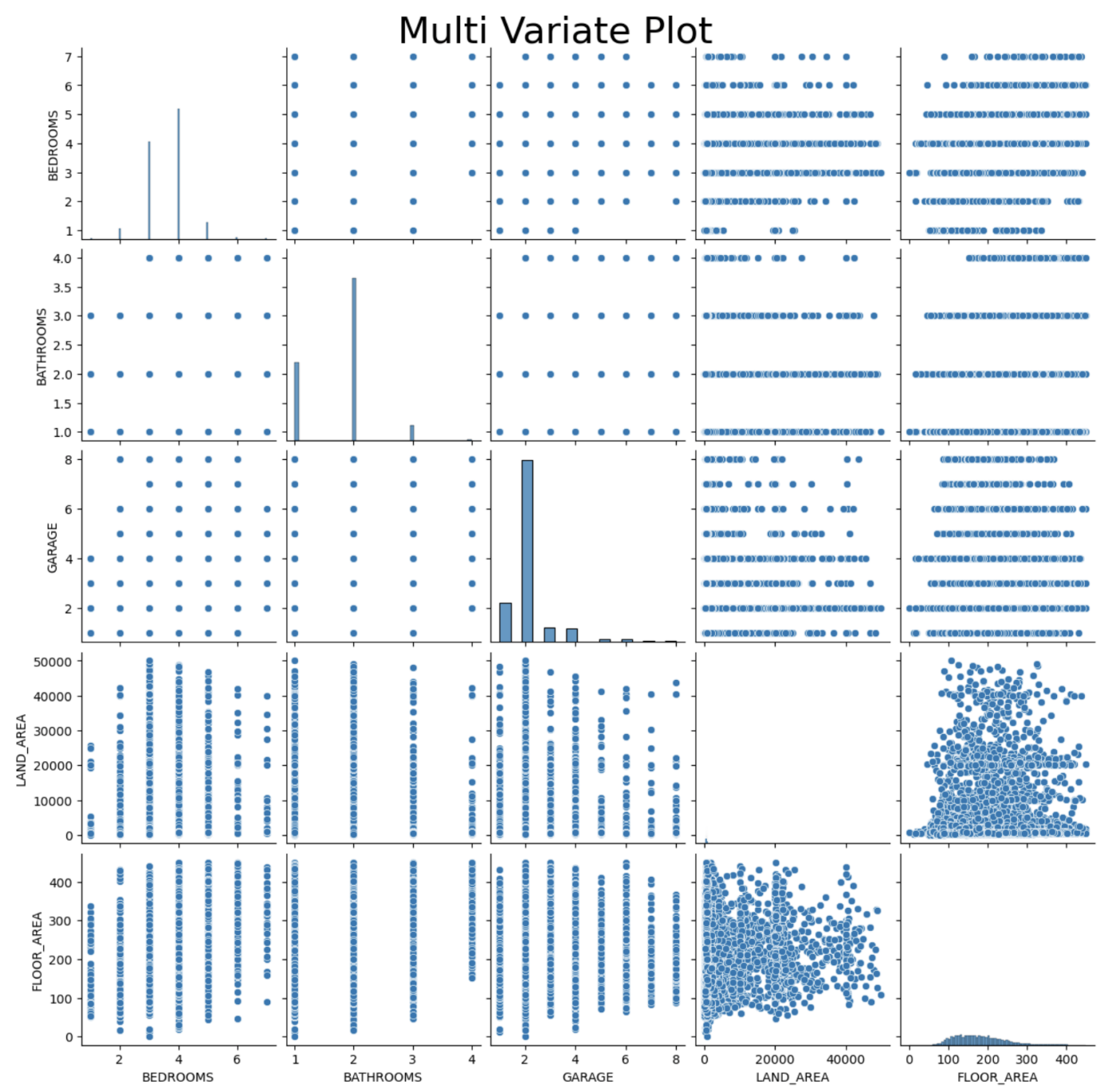
After observing the box plot showed us that there were still many outliers present in the features which were considered during plotting but, we did not remove these outliers as we think these are just one or two rather many which can be normalized later.

6. Displaying the Bar chart for price categories



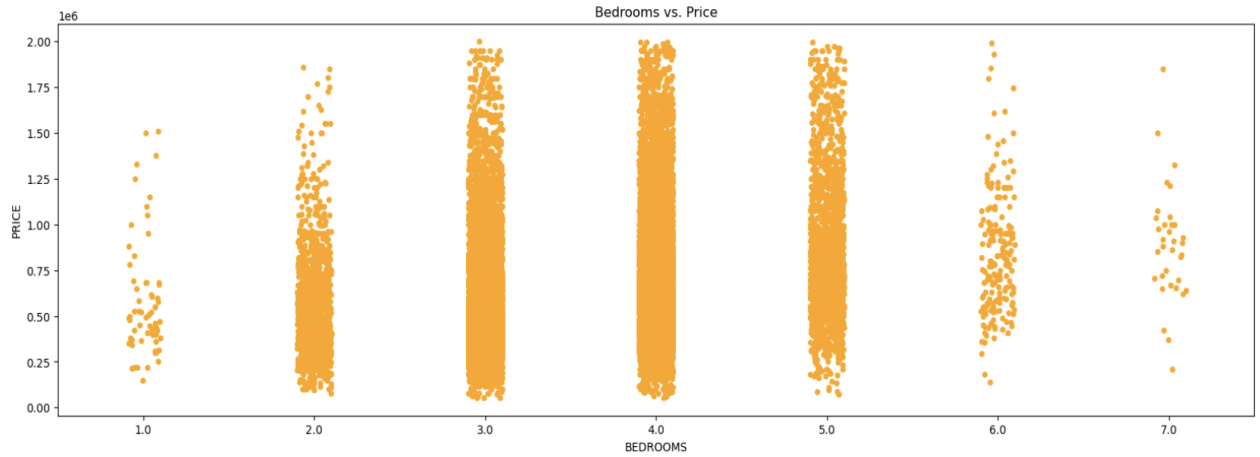
After examining the bar graph, we concluded that, in comparison to the other two groups, the houses that come within the medium price category had the most sales.

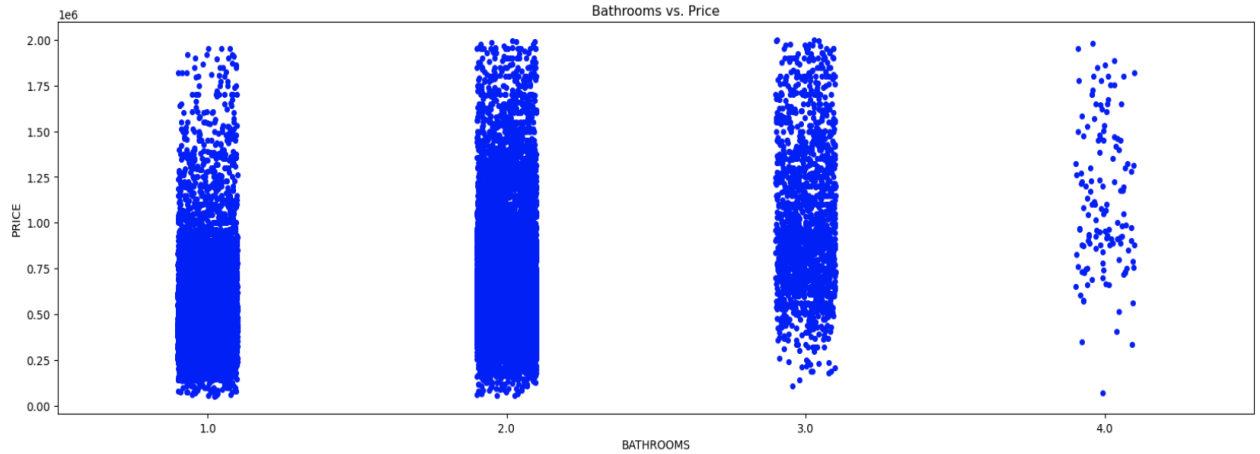
7. Displaying the pair plot for columns 'PRICE', 'BEDROOMS', 'BATHROOMS', 'GARAGE', 'LAND_AREA', 'FLOOR_AREA','CBD_DIST', 'NEAREST_STN_DIST', 'NEAREST_SCH_DIST'

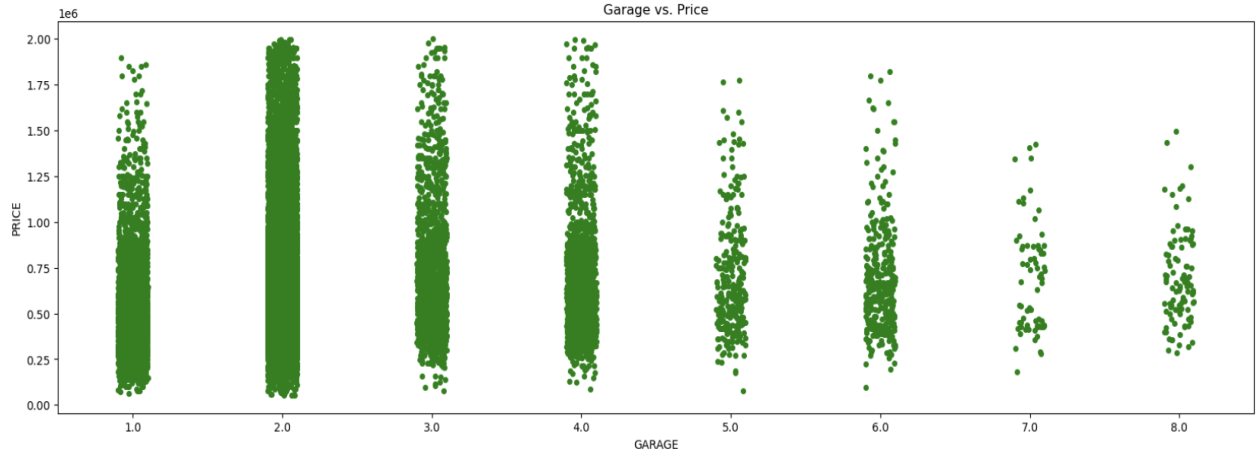


Upon visualizing, we observe that the plots in the lower triangle of the rectangular pair plot do not exhibit clustering; instead, they appear to be evenly distributed. This suggests that there may not be strong correlations or groupings among these features, and they might be relatively independent of each other.

8. Displaying Strip Plot for number of BEDROOMS, BATHROOMS and GARAGE against PRICE

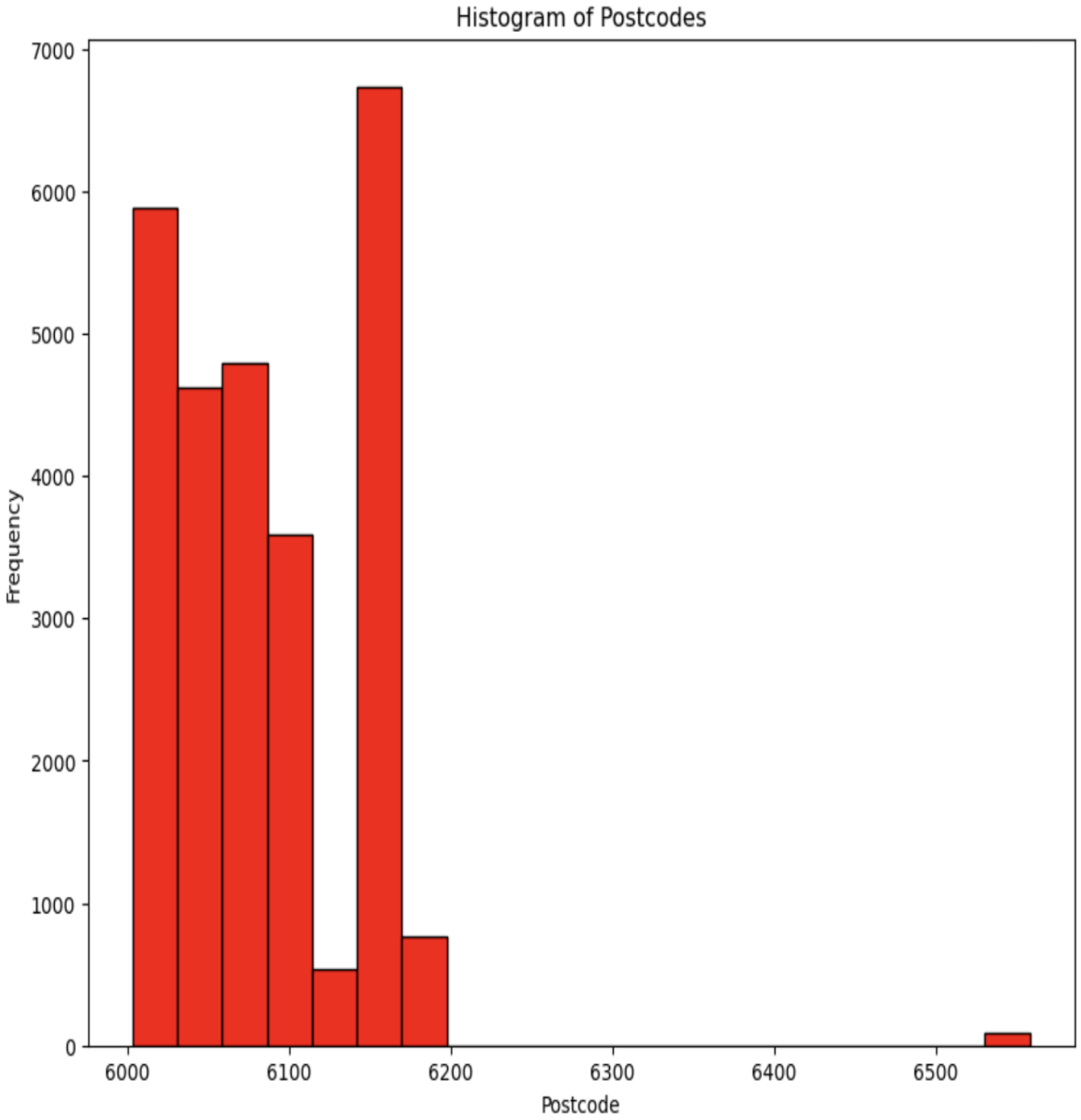






After observing the strip plot it gave the insights on how the number of ‘BEDROOMS’, ‘BATHROOMS’ and ‘GARAGE’ differ for different ‘PRICE’ of the house. We can conclude that for some number of ‘BEDROOMS’, ‘BATHROOMS’ and ‘GARAGE’ the price does not matter and we can find the same number of features in almost every price range.

9. Displaying Histogram plot of POSTCODES



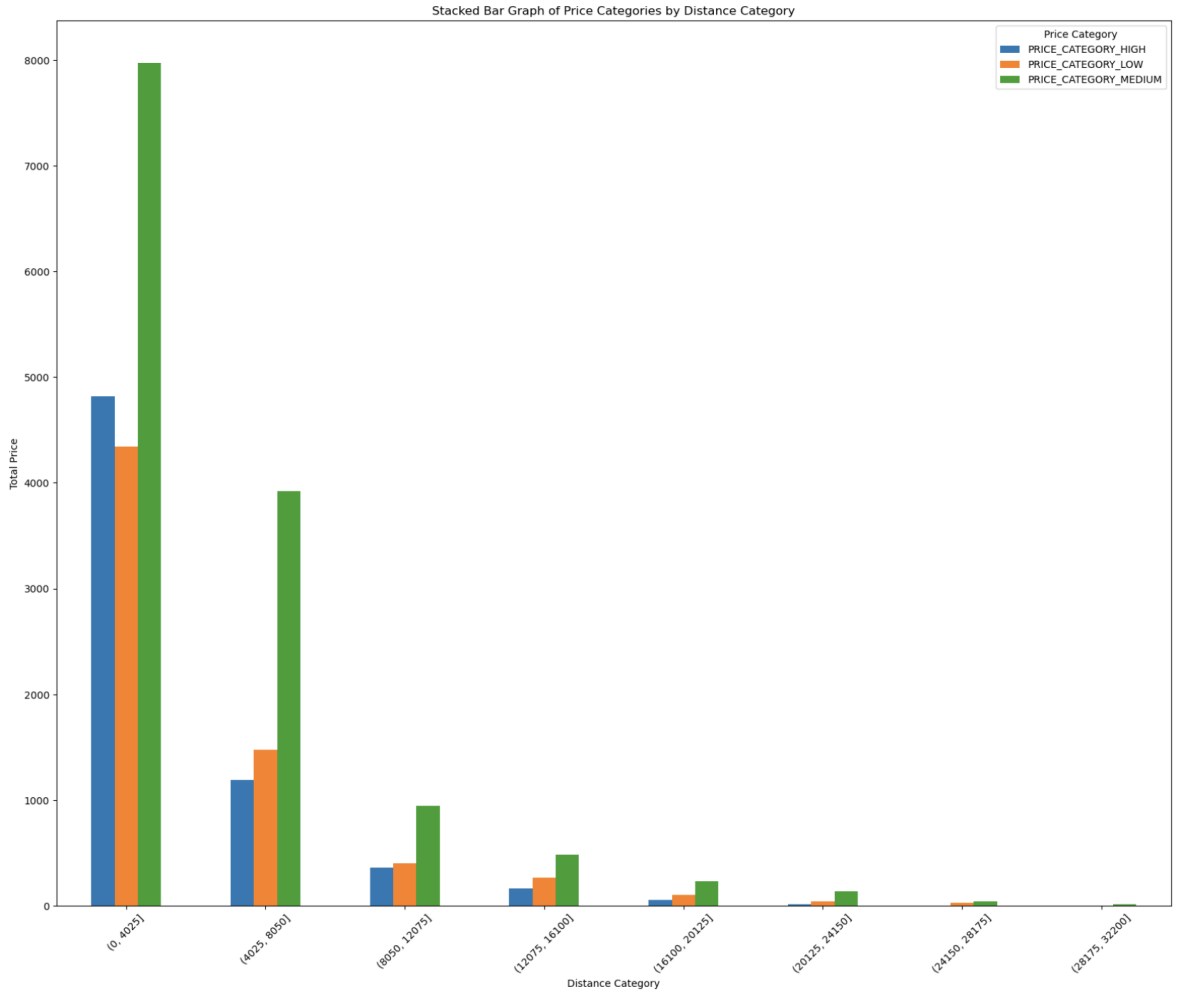
The histogram plot illustrates the distribution of houses in various areas of Perth, categorized by their respective postal codes.

10. Displaying bar graph based on NEAREST_STN_DIST

To visualize this we first divided the NEAREST_STN_DIST into 8 groups

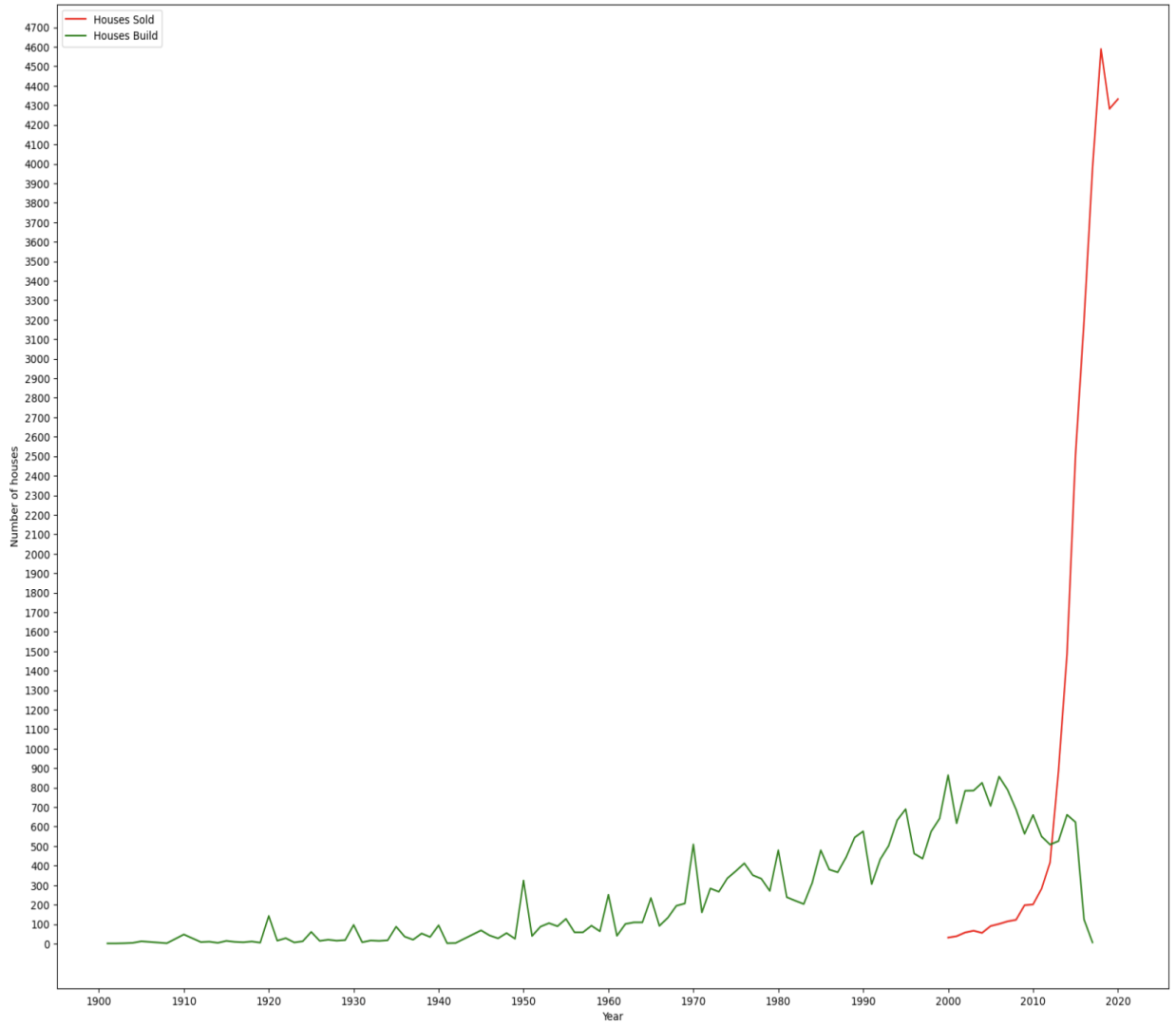
NEAREST_STN_DIST_CATEGORY	PRICE_CATEGORY_HIGH	PRICE_CATEGORY_LOW	PRICE_CATEGORY_MEDIUM
(0, 4025]	4820	4344	7973
(4025, 8050]	1193	1477	3924
(8050, 12075]	358	405	944
(12075, 16100]	167	265	483
(16100, 20125]	55	104	229
(20125, 24150]	14	42	136
(24150, 28175]	2	26	42
(28175, 32200]	1	2	11

Based on this group we display the number of houses sold for different price categories.



After observing we can conclude that people prefer houses that are close to the nearest station in every price category.

11. Line graph to display the trend of houses built and sold over years



The graph helped us understand how houses were constructed and sold over the years after we had looked at it. Similar to the majority of houses sold between 2010 and 2020, the majority of houses were built between 1960 and 2015.

After performing all the EDA techniques normalization was done on the dataset.
Final shape of the dataset - 27017 rows x 17 columns
Note :

- Cleaning and EDA operations are implemented and daved in the file named ‘phase_1.ipynb’ in the src directory.
- The cleaned dataset was saved into the file named ‘cleaned_dataset.csv’ which can be found in the src directory.

End of Report