Transport and Telecommunication Institute &

UWE Bristol

COMPUTER PRACTICE 3

REGRESSION ANALYSIS

Name : Anjali Shibu

Student ID :St86860

Group :4403 MDA

CONTENTS

1. Introduction

In this lab exercise, we are given an excel containing 18 variables and 21597 entries.Dataset contains prices of houses from King County in US state of Washington and also covers Seattle. We load the dataset into pandas data frame to know the variables and datatypes(fig.1).We also make sure there are no missing values .Here ,price is the variable we are going to predict. A detailed description of the given data is given in fig 2.

```
Data loaded successfully. Shape: (21597, 18)

Data Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21597 entries, 0 to 21596
Data columns (total 18 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   id             21597 non-null  int64
 1   date           21597 non-null  object
 2   price          21597 non-null  float64
 3   bedrooms       21597 non-null  int64
 4   bathrooms      21597 non-null  float64
 5   sqft_living    21597 non-null  int64
 6   sqft_lot       21597 non-null  int64
 7   floors         21597 non-null  float64
 8   waterfront     21597 non-null  int64
 9   view           21597 non-null  int64
 10  condition      21597 non-null  int64
 11  grade          21597 non-null  int64
 12  sqft_above     21597 non-null  int64
 13  sqft_basement  21597 non-null  int64
 14  yr_built       21597 non-null  int64
 15  yr_renovated   21597 non-null  int64
 16  sqft_living15  21597 non-null  int64
 17  sqft_lot15     21597 non-null  int64
dtypes: float64(3), int64(14), object(1)
memory usage: 3.0+ MB
None
```

Figure 1. Data information

Variable	Description	Data Type
id	A notation for a house	Numeric
date	Date house was sold	String
price	Price is prediction target	Numeric
bedrooms	Number of Bedrooms/House	Numeric
bathrooms	Number of bathrooms/bedrooms	Numeric
sqft_living	Square footage of the home	Numeric
sqft_lot	Square footage of the lot	Numeric
floors	Total floors (levels) in house	Numeric
waterfront	House which has a view to a waterfront	Numeric
view	Has been viewed	Numeric
condition	How good the condition is (1 = worn out, 5 = excellent)	Numeric
grade	Overall grade given to the housing unit (1 = poor, 13 = excellent)	Numeric
sqft_above	Square footage of house apart from basement	Numeric
sqft_basement	Square footage of the basement	Numeric
yr_built	Built Year	Numeric
yr_renovated	Year when house was renovated	Numeric
sqft_living15	Living room area in 2015 (implies renovations)	Numeric
sqft_lot15	Lot size area in 2015 (implies renovations)	Numeric

Figure 2. Data description

2. Data cleaning

We decided to clean our data to change some data types to improve consistency, eliminate redundancy, and enhance clarity for analysis.

```
 0   id             21420 non-null  object
 1   date           21420 non-null  datetime64[ns]
 2   price          21420 non-null  float64
 3   bedrooms       21420 non-null  int64
 4   bathrooms      21420 non-null  float64
 5   sqft_living    21420 non-null  int64
 6   sqft_lot       21420 non-null  int64
 7   floors         21420 non-null  float64
 8   waterfront     21420 non-null  int64
 9   view           21420 non-null  int64
 10  condition      21420 non-null  int64
 11  grade          21420 non-null  int64
 12  sqft_above     21420 non-null  int64
 13  sqft_basement  21420 non-null  int64
 14  yr_built       21420 non-null  int64
 15  yr_renovated   910 non-null    float64
 16  sqft_living15  21420 non-null  int64
 17  sqft_lot15     21420 non-null  int64
dtypes: datetime64[ns](1), float64(4), int64(12), object(1)
```

Figure 3. Data cleaned

- ‘Id’ converted to string from numeric to object since it is a categorical identifier.
- ‘date’ datatype changed from object to datetime format to do time based analysis.
- Checked for duplicates in ‘id’ column and no duplicates exist.

- The year renovated (‘yr_renovated’) that had values ‘0’ changed to NaN to properly indicate properties that was never renovated.

3. Statistics of dependent variable.

From fig.4 we get the statistics of dependent variable. There are 21597 price values, the total mean is 540,296.57 $. Minimum is 78000$ and maximum is 7,700,000$ shows a wide range of prices, and 75th percentile shows value 645,000$ .This confirms the data do probably have outliers. Half of the house prices are below 450,000 $.

```
Dependent variable: price

Price Statistics Table:
Statistic        Value
    count    21,597.00
     mean   540,296.57
      std   367,368.14
      min    78,000.00
      25%   322,000.00
      50%   450,000.00
      75%   645,000.00
      max 7,700,000.00
```

Figure 4. Price statistics

4. Visualising the distribution of dependent variable.

Here we plot the histogram of the house prices to know if the distribution is normal, skewed or has outliers. Figure 5 shows histogram of house prices.

Figure 5. Histogram of house prices.

The distribution shows high right skewness, most houses clustered at lower prices and long tail of expensive homes on right side of graph.Few ultra-high priced homes (luxury outliers) exist.So ,there is significant market inequality.

5. Exploratory data analysis.

From figure 6. We get an idea that waterfront houses have higher median price than non-waterfront ones.There is greater variability in luxury property values in waterfront ones.

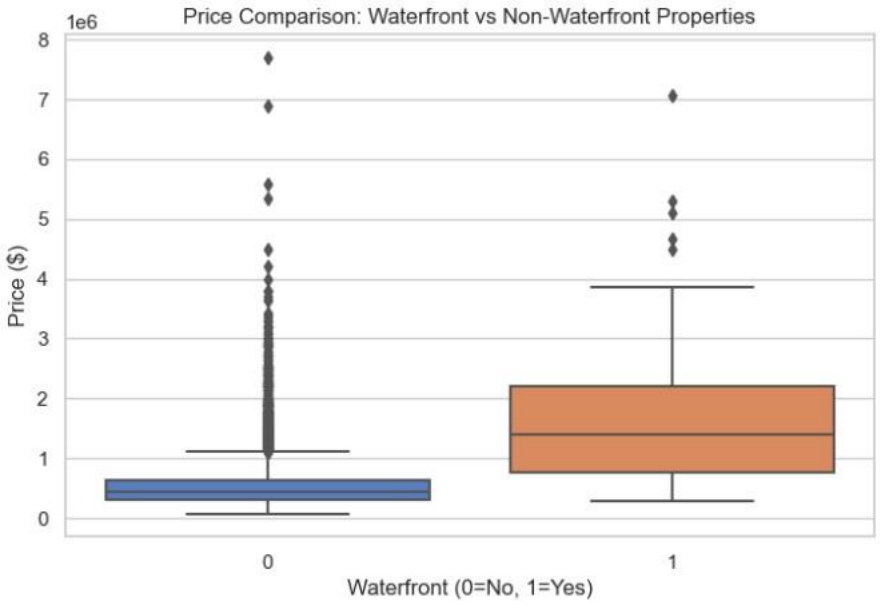


Fig6.Waterfront v.s price

From figure 7, we get a picture that new home after 1980 have higher prices, pre-1940 has high price variability,maybe because of renovation status or historic significance.Outliers can be luxury features.

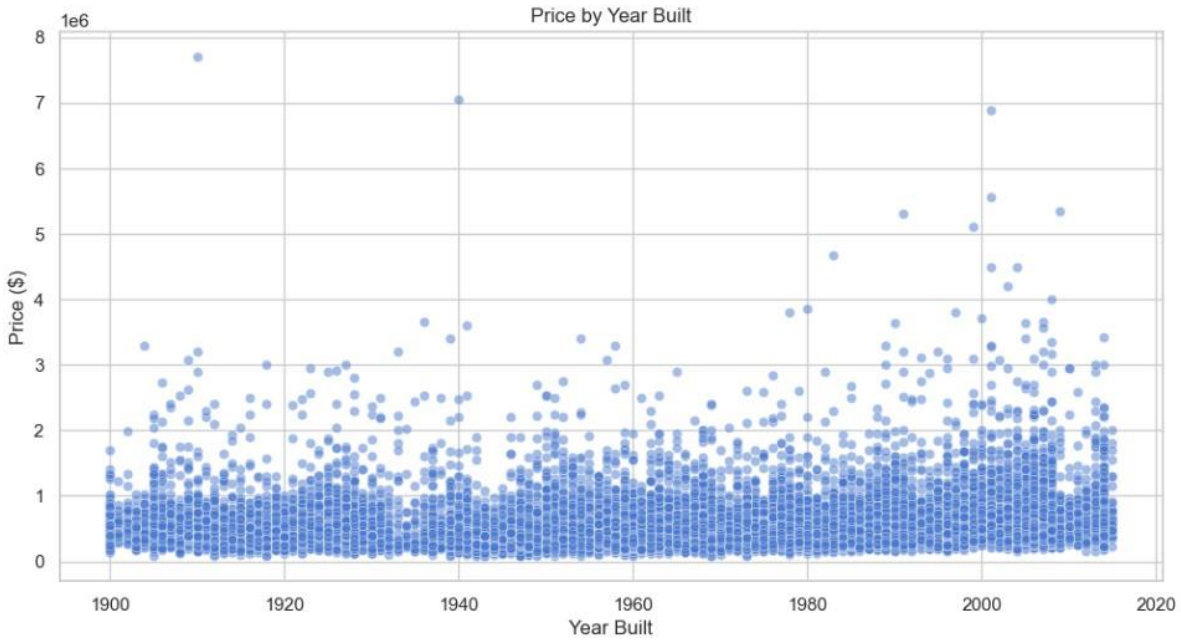


Figure 7. Price and year built scatter plot

From figure 8 we see positive relationship. High view rating and better condition have higher prices.Both show price variability.There must be other factors also responsible for this.

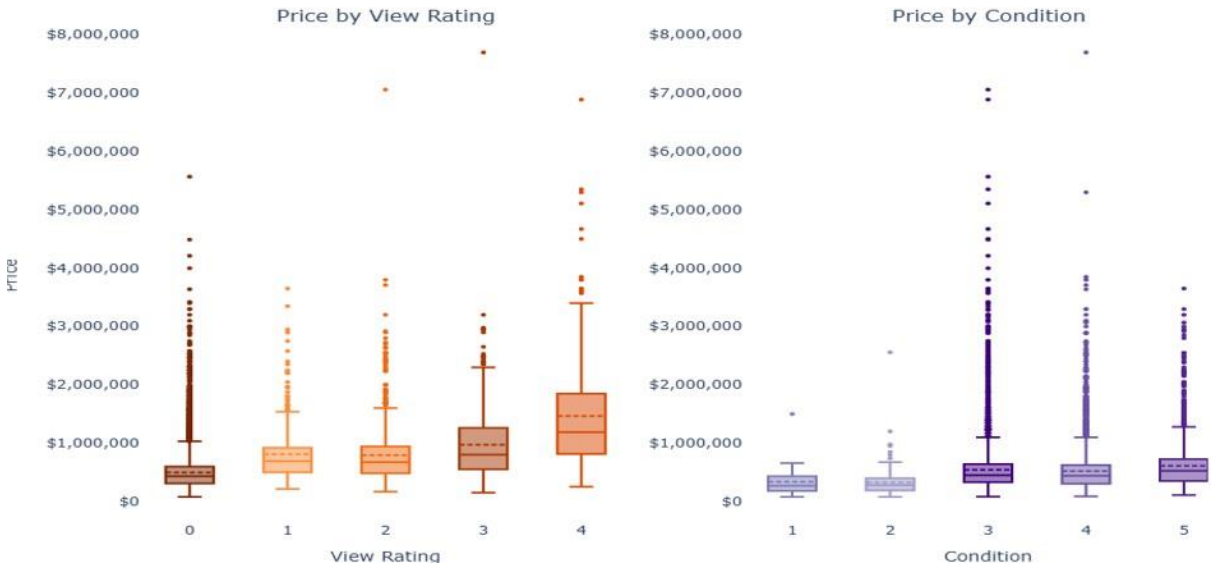


Figure 8. Price comparison with view rating and condition box plot

From figure 9 we see homes with 3-4 bedrooms show most consistent pricing, more than 5 bedrooms have wider price ranges,show luxury features.1-2 bedroom houses maintain stable mid-range prices.

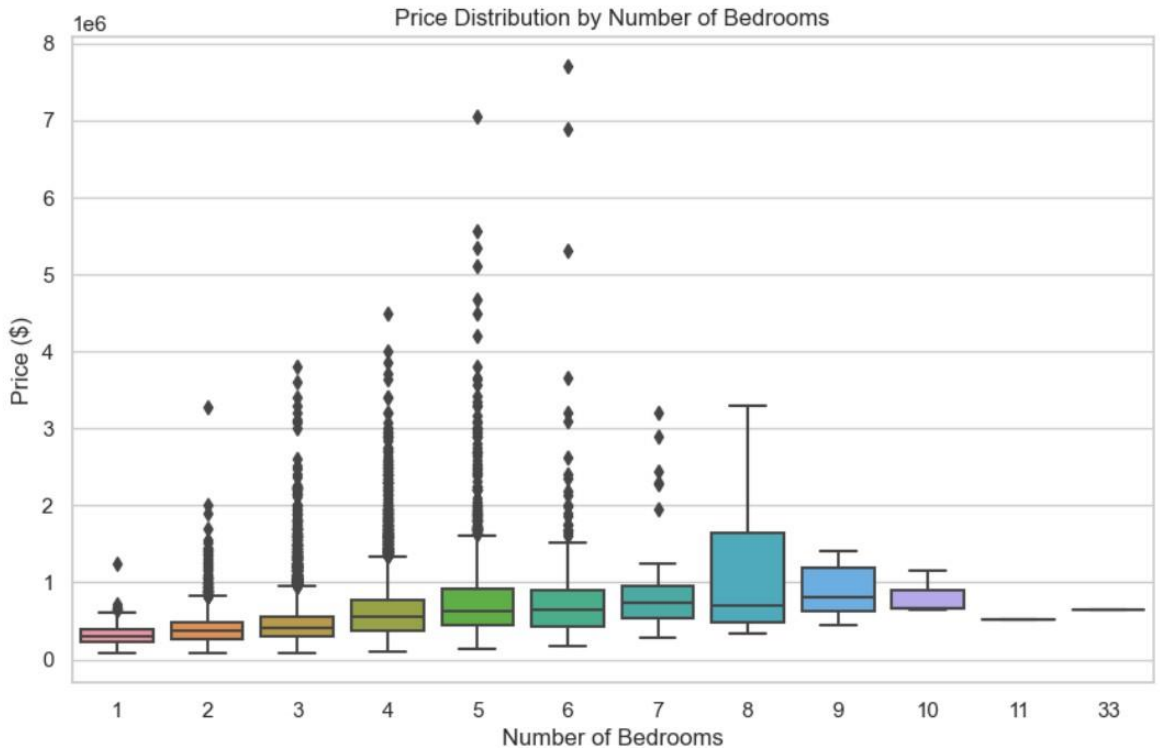


Figure 9. Price v.s no.of bedrooms

From figure 10. LOESS curve shows a U shape, mid-ages homes show lower prices,newer and historic properties achieve premium pricing.So renovation status and vintage appeal might be the reason for such a pricing.

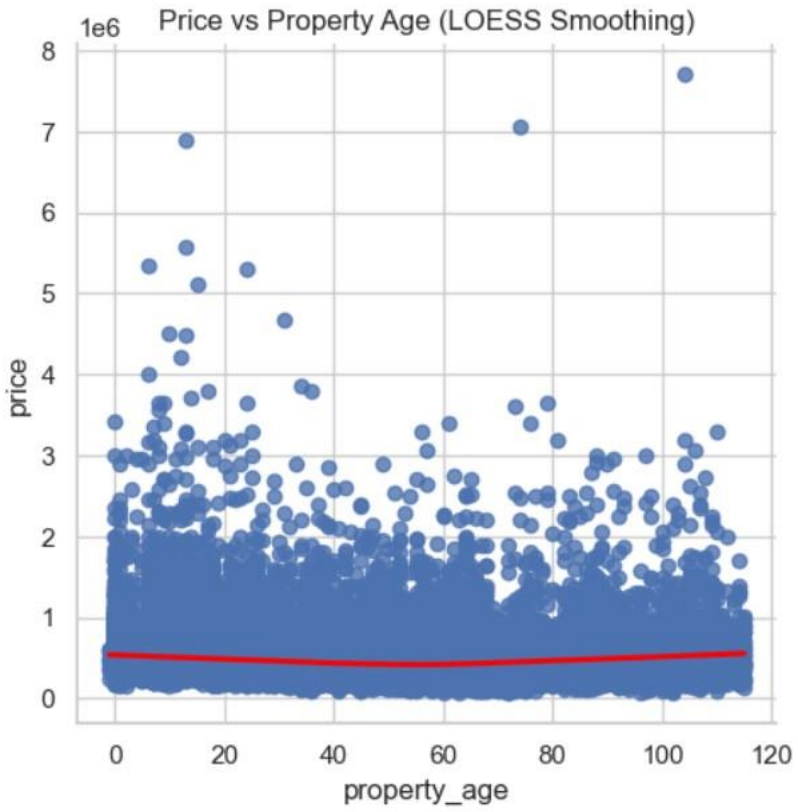


Figure 10. Price v.s property age

6. Data sample correlation

Pair plot analysis with price as target variable with regression line

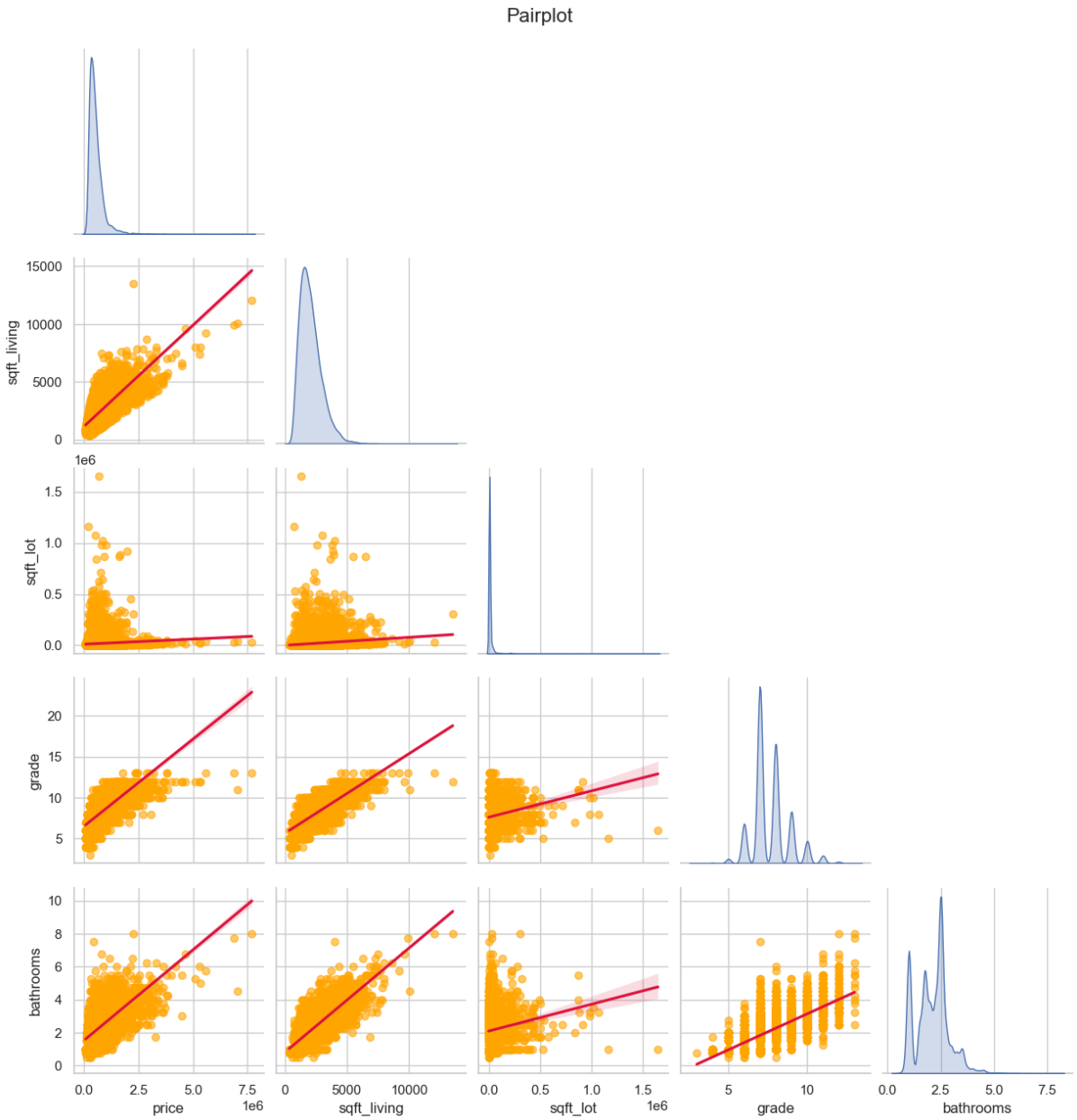


Figure 11. Pairplot

The pairplot examines relationship between price and other variables,namely sqft_living,grade,bathrooms,sqft_lot. We will understand how these features influence housing prices.

i) Price and sqft_living

- Strong positive correlation : Larger homes have higher prices.
- Spread of price values increase at larger sqft living values.Heteroscedasticity exists.
- Some smaller homes also have high prices ,possibly due to other premium features.

- The regression line show upward trend, fit the data closely, indicates a good predictor for price.

ii) Price and grade

- Positive correlation : Higher grades (10-13) correlate strongly with higher prices.
- Grade is a significant factor in predicting house prices.
- Scatter plot show clear upward trend,relationship plateau at highest grades.
- Higher grades show better quality and finish,hence higher prices.

iii) Price and sqft_lot

- The scatter plot shows a weak positive correlation between house price and lot square footage.
- Points are widely dispersed shows non linear relationship.
- Higher lot prices do not consistently correspond to higher prices, suggesting other factors may dominate.
- The relationship is weak and scattered with high variability and outliers.

iv) Price and bathrooms

- Positive correlation: Homes with more bathrooms tend to have higher prices.
- The increase in price is not uniform across different bathroom counts.
- There is variability especially in midrange bathroom counts.
- Scatter plot show a positive correlation with a red linear regression line indicating upward trend.

Correlation matrix heat map of housing variables (fig.12)

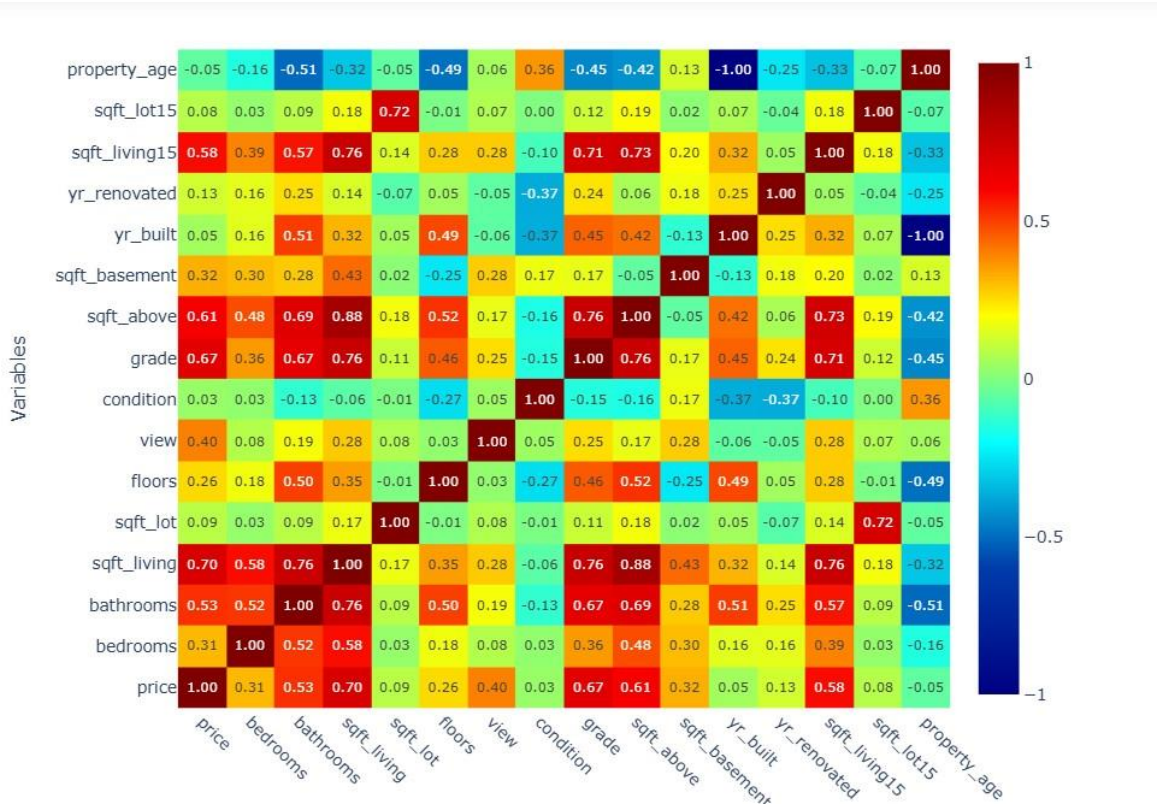


Figure 12. Heat map

Variables significantly correlated with price in positive direction

From this heatmap we see that , positive correlation(>0.5) exist with price :

Sqft_living : 0.70 - Large living area strongly increases price

Grade: 0.67 – Higher-grade homes command premium prices

Sqft_above: 0.61 Above-ground square footage boosts price.

Sqft_living15: 0.58 Recent living area impacts price

Bathrooms : 0.53 More bathrooms correlate with higher prices

sqft_living ,grade, and bathrooms are strongest drivers of price.(sqft_living and sqft_above are identical and can cause multicollinearity,so considering sqft_living =0.7 since it is greater than sqft_above=0.61). If two predictors are too correlated, then regression coefficients become unstable.Dropping sqft_above reduces noise without predictive power being sacrificed.We need non overlapping high impact variables.

Most significant correlation between other variables(>0.5 positive correlation)

Sqft_living and sqft_above : 0.88 - near perfect correlation, likely redundant.

Sqft_living and grade : 0.76 - Higher grade houses have more above ground space.

Bedrooms and bathrooms : 0.52 – More bedrooms typically accompany more baths.

Floors and sqft_above : 0.52- Multi-story homes have more above ground area.

7. Scatter plots between dependent and continuous variables.

a) Price v.s bathrooms

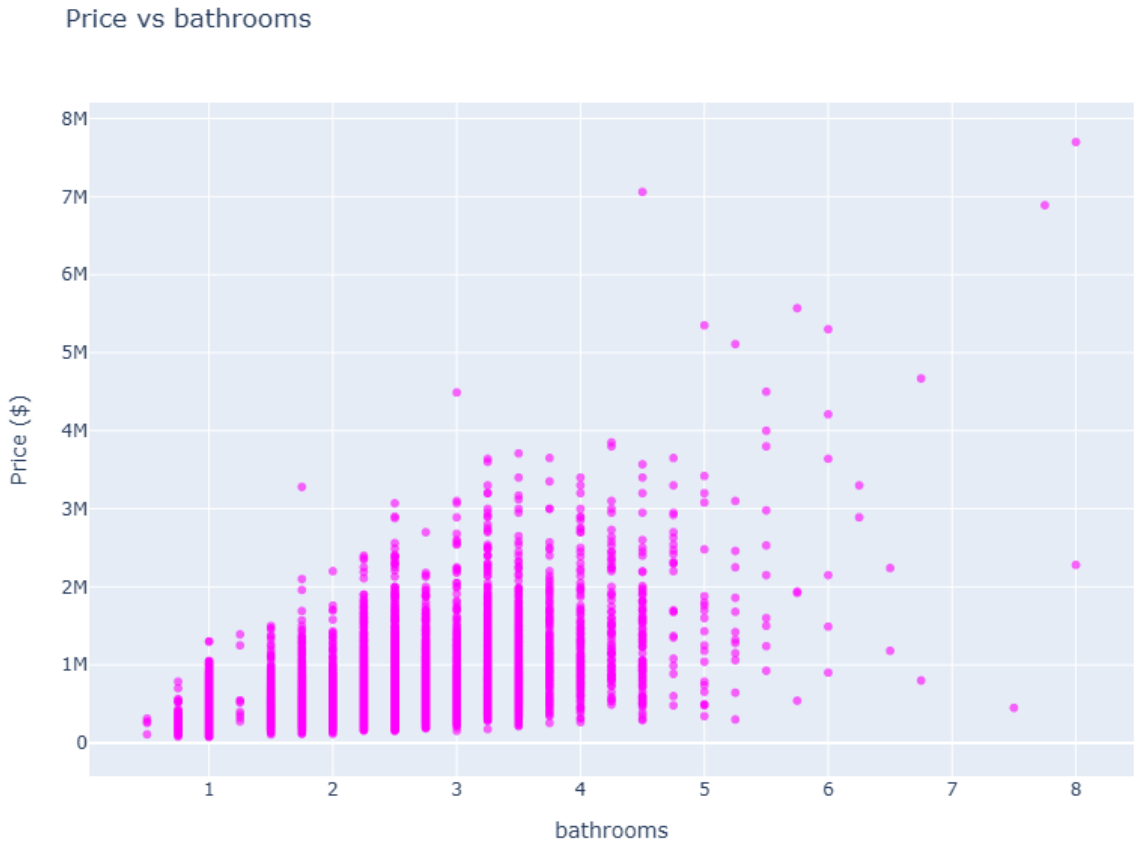


Figure 13 : Price .vs bathrooms scatter plot

Figure 13 shows diminishing returns. Rate increase steeply till 3 bathrooms and plateaus thereafter.Buyers do not prefer excess capacity but functional adequacy.Potential outlier more than 6 bathrooms with less prices. Here we'll treat bathrooms as ordinal variable binning into 1-2,2-3 etc. to capture non-linearity.Outliers are retained to see luxury behaviour. The box

plots below confirm the plateau effect and 5 or more bathrooms has 50% premium increase over 3-4 bathrooms(fig.14).

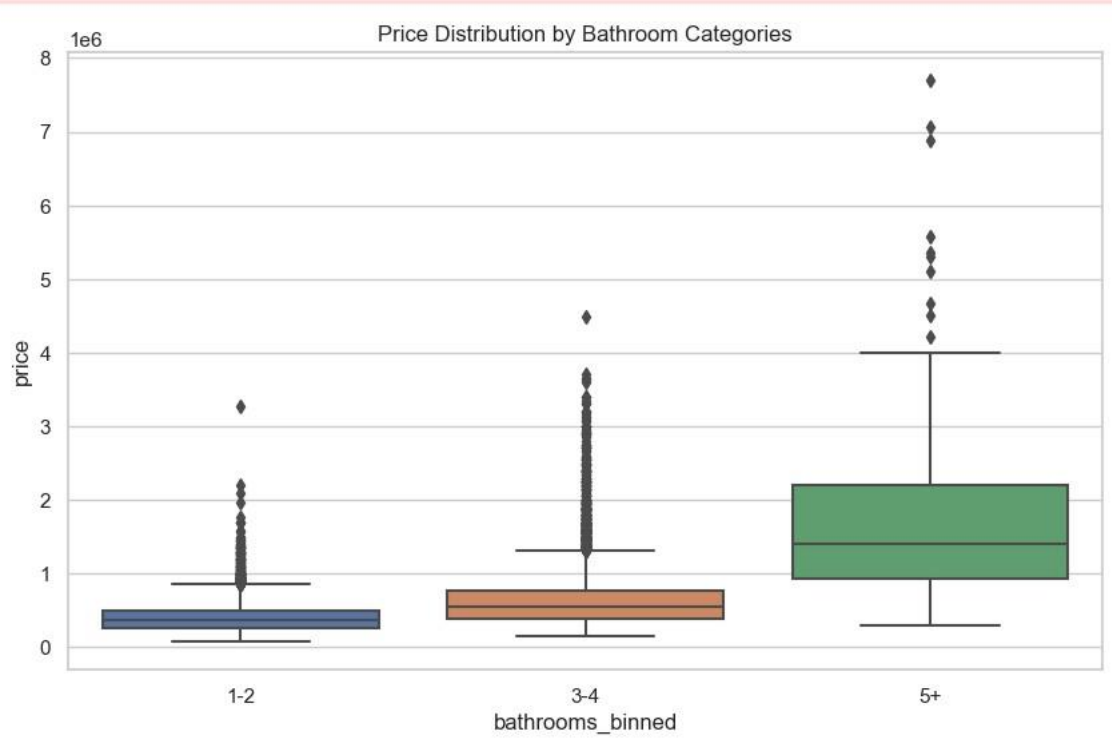


Figure 14. Binned bathrooms box plot

outlier validation

```
# Identify homes with 6+ bathrooms
luxury_homes = df[df['bathrooms'] >= 6][['price', 'sqft_living', 'grade', 'bathrooms']]

# Summary stats
print("Luxury Properties (6+ baths):")
print(luxury_homes.describe())

# Contextual check - compare to top 1% of all homes
print("\nTop 1% Price Threshold:")
print(df['price'].quantile(0.99))
```

```
Luxury Properties (6+ baths):
              price   sqft_living      grade  bathrooms
count  1.600000e+01     16.000000  16.000000  16.000000
mean   3.130625e+06   7845.625000  11.437500   6.640625
std    2.162827e+06   2520.150706   1.672075   0.752600
min    4.500000e+05   4050.000000   7.000000   6.000000
25%    1.412500e+06   6725.000000  11.000000   6.000000
50%    2.585000e+06   7415.000000  12.000000   6.375000
75%    4.325000e+06   8912.500000  12.000000   6.937500
max    7.700000e+06  13540.000000  13.000000   8.000000

Top 1% Price Threshold:
1970000.0
```

Figure 15.Outlier validation

All 16 properties exceed the top 1% threshold, thus validating their inclusion.

b) Price v.s sqft_living scatter

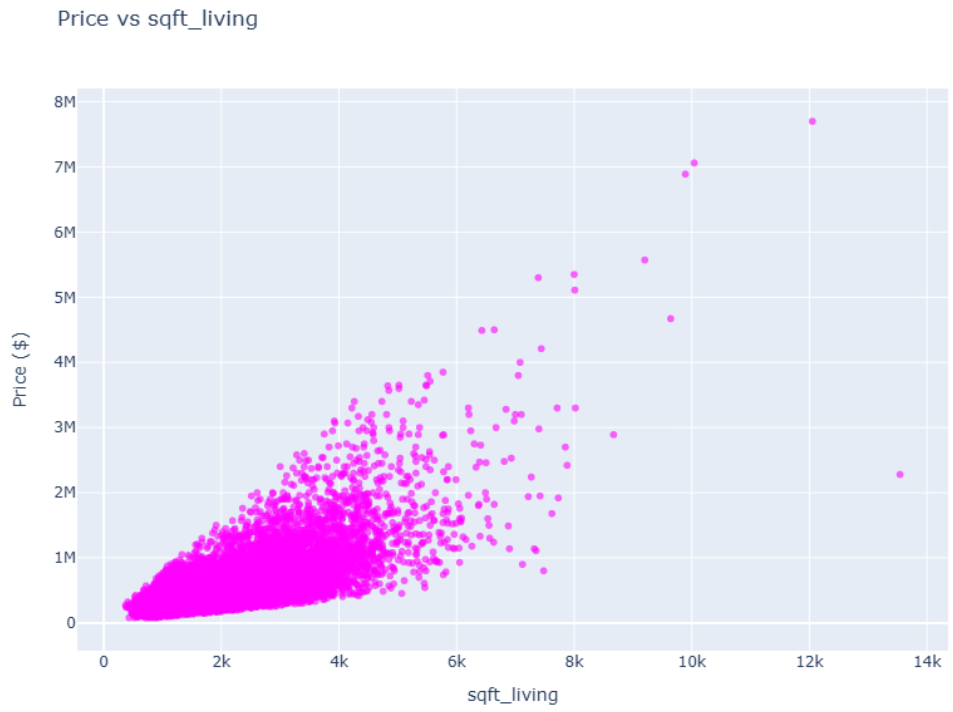


Figure 16. Price v.s sqft_living scatter plot

Strong positive coreelation with larger homes commanding high prices,non –linear relation and has notable outlier at high price point.

c) Price vs sqft_lot

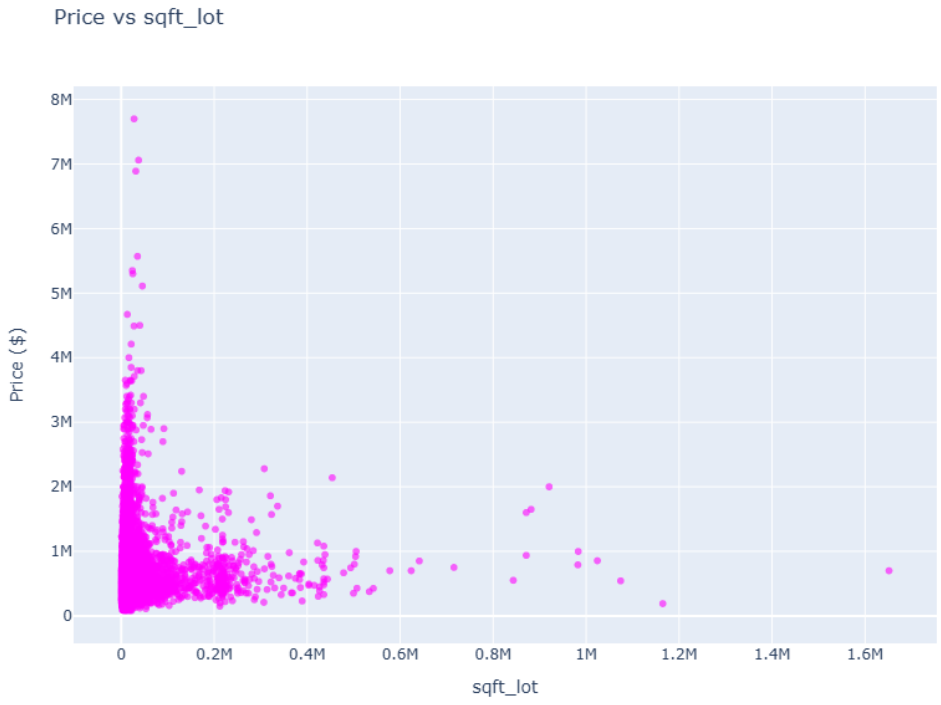


Figure 17. Price v.s sqft_lot scatter plot

Weak positive relation exist.No consistent price trend ,few outliers exist.Lot size show minimal linear association with price.

d)Price v.s floors

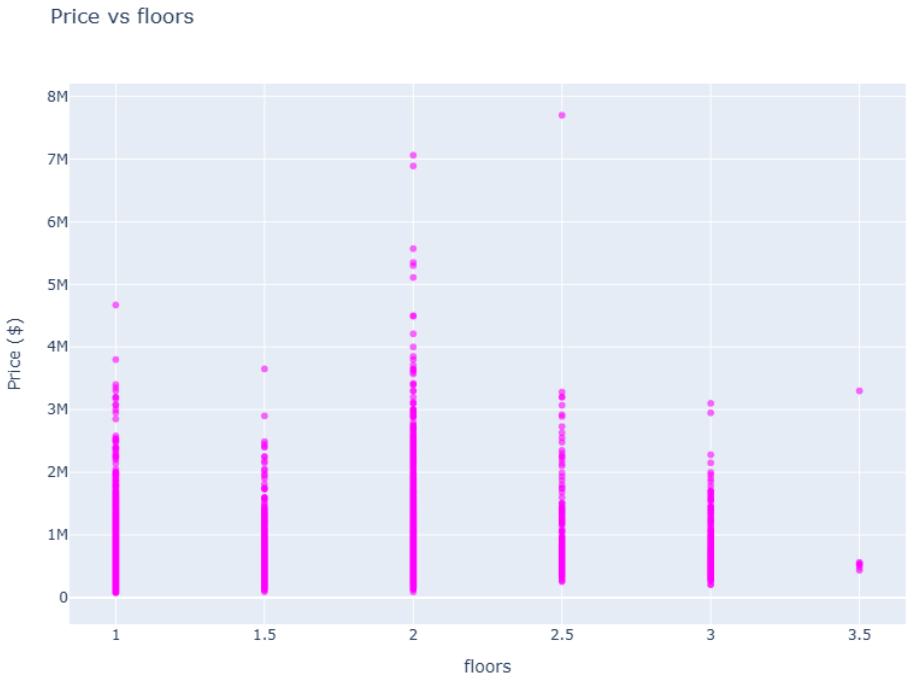


Figure 18. Price v.s floors scatterplot

Homes with 2 floors are common and have wider price range.Beyond 2 floors price does not consistently increase,suggesting luxury markets or architechtural factors may dominate.Extreme high price home are there in all floors, so outlier exist.Floor count and price has non linear relationship.

e) Price v.s sqft_above

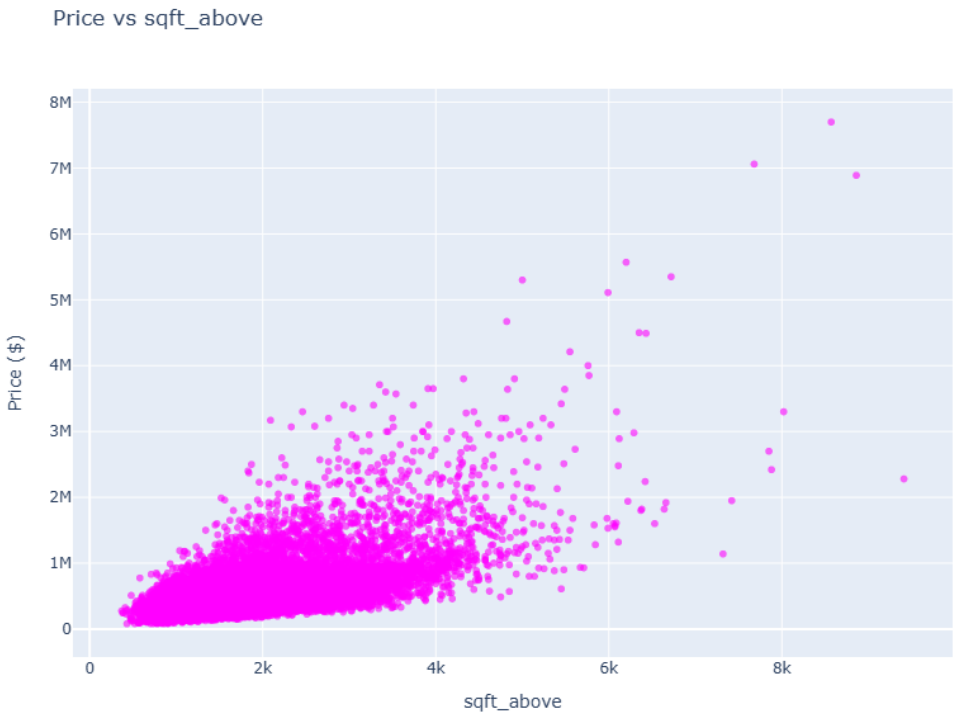


Figure 19. Price v.s sqft_above scatter plot

Large above ground space commands higher value.Trend appears linear for moderate sizes upto 4k sqft but may accelerate slightly for larger homes.Most homes clustered below 3M $ and 4k sqft, with tight price consistency. A few high priced outliers above 6M$ show trend deviation,might be due to luxury features or waterfront locations. So,sqft_above is a strong predictor of price.

f) Price vs sqft basement

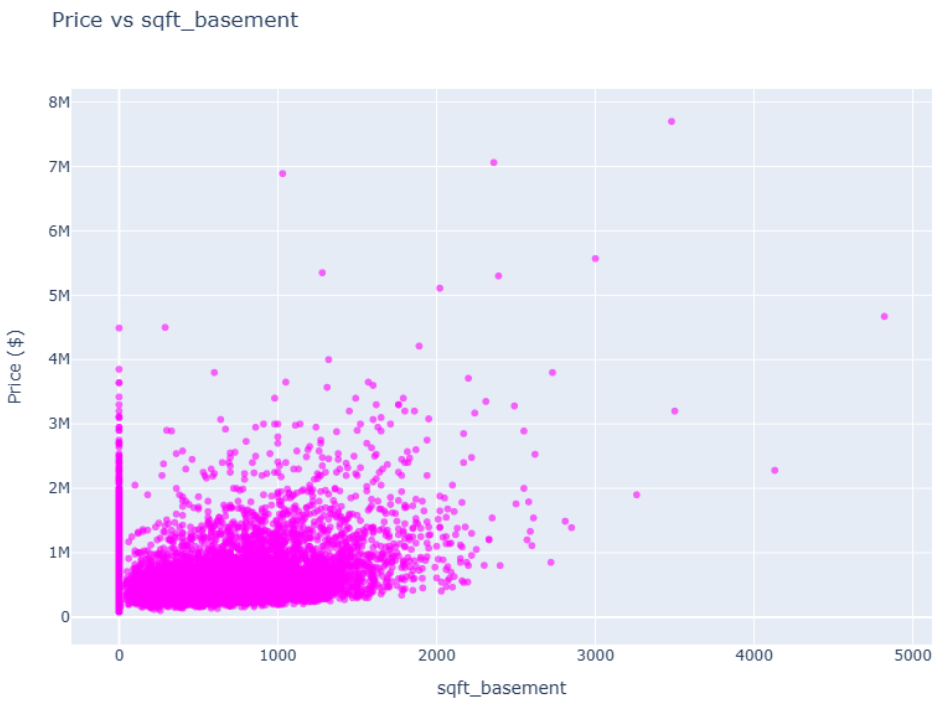


Figure 20. Price v.s sqft_basement scatter plot

Homes with zero basement cover full price range ,indicates basement is not a price determinant.Several high price homes have mid range basements.Limited direct relationship exists between price and basement area.After 1000 sqft effects are noticeable.Non uniform relationship.

g) Price v.s year_built

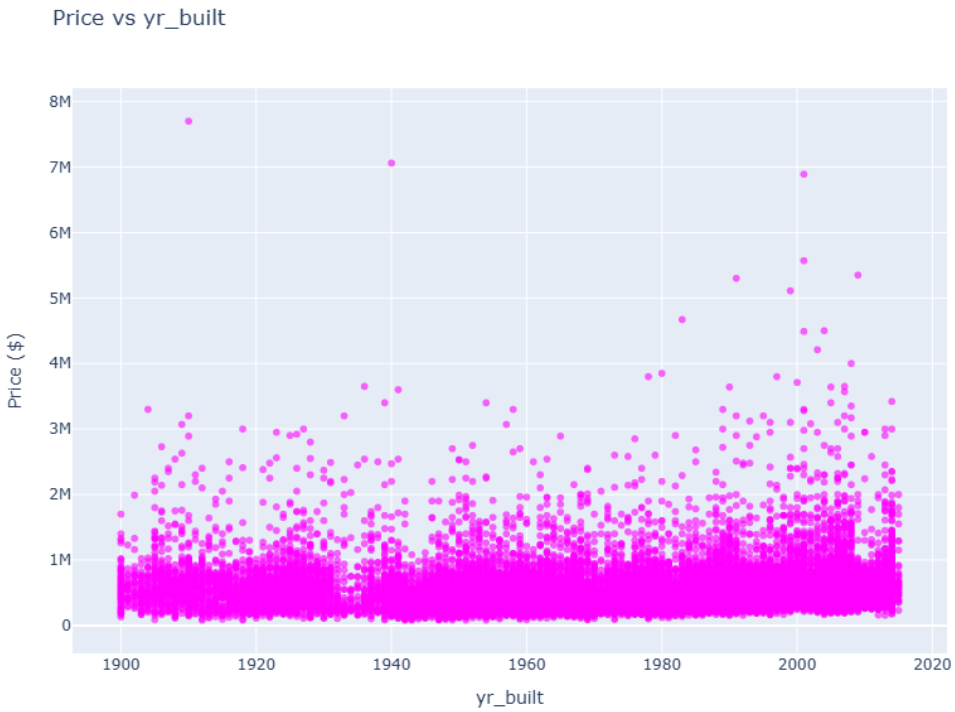


Figure 21. Price v.s yr_built scatter plot

Post 1980s show high prices,pre 1940 old homes a mix of low value properties and high outliers,may be historic significance. 1940-80 homes show moderate prices with high variability.Few post 2000 homes have lower prices,possibly due to smaller sizes or less desirable locations.Heteroscedasticity exist.Variability in prices increases for newer homes,indicating that others factors(e.g.luxury features) dominate modern properties.Further testing is needed to quantify its contribution relative to other features.

h) Price v.s sqft_living15

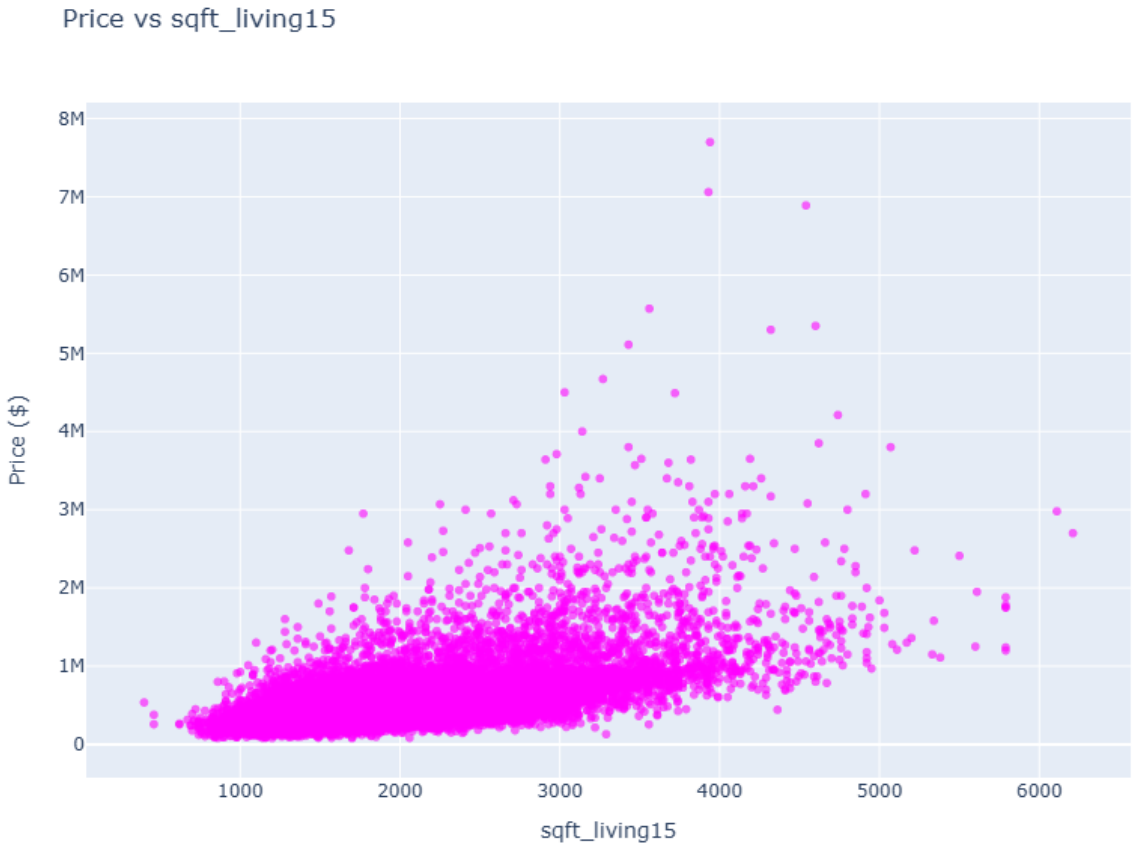


Figure 22 Price v.s sqft_living15 scatterplot

Strong positive correlation clear upward trend shows larger homes have higher prices.High value outlier exists in all size ranges.Sqft_living15 can be a significant variable in the model.

i) Price v.s sqft_lot15

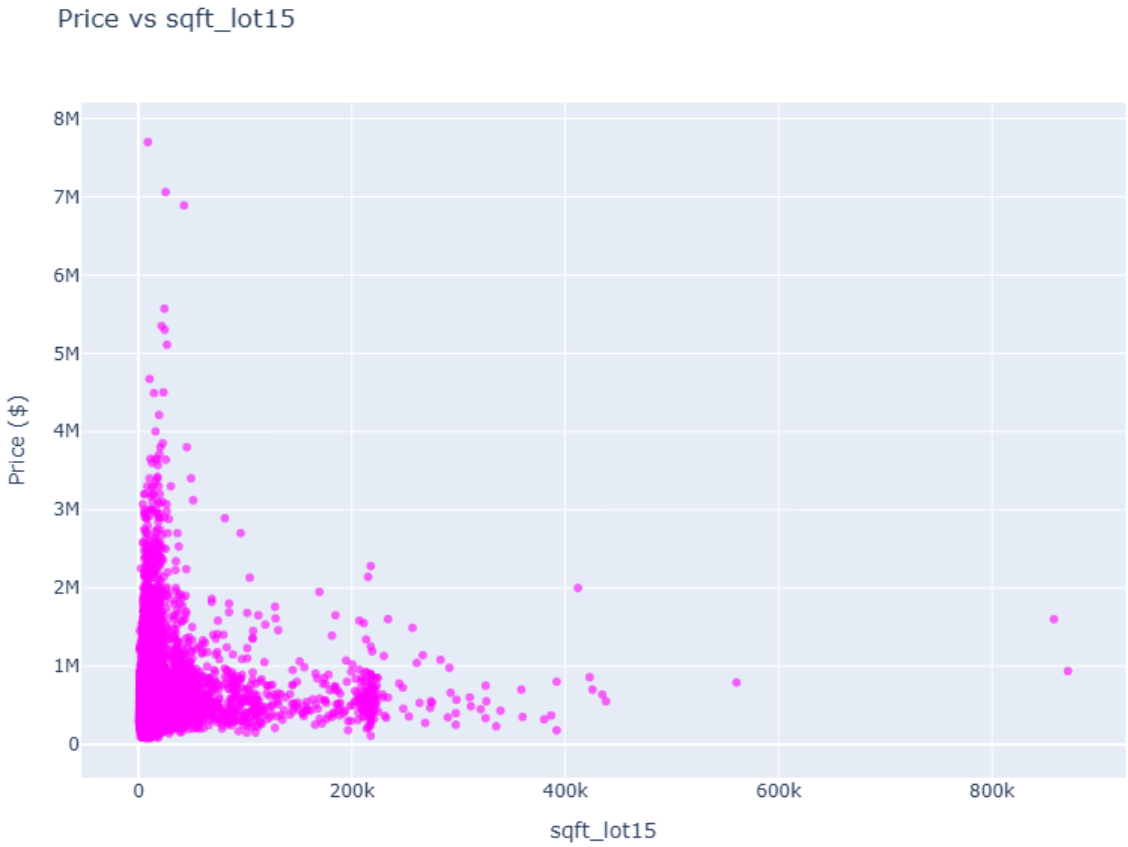


Figure 23 .Price v.s sqft_lot15 scatterplot

Most homes are at lower lot szies.Many similar sized lots vary widely in price,so other factors dominate.Sqft_lot shows weak overall correlation with price with exceptions for large lots that command premium values.

8. Dummy values creation, feature engineering and data transformation.

We created dummy variables for :**Floors** -as fractional values indicate categories like split level homes not continuous measurements.**View,condition,grade**-Dummy encoding avoid assuming linear relationships like a grade of 12 may not be twice as good as 6.

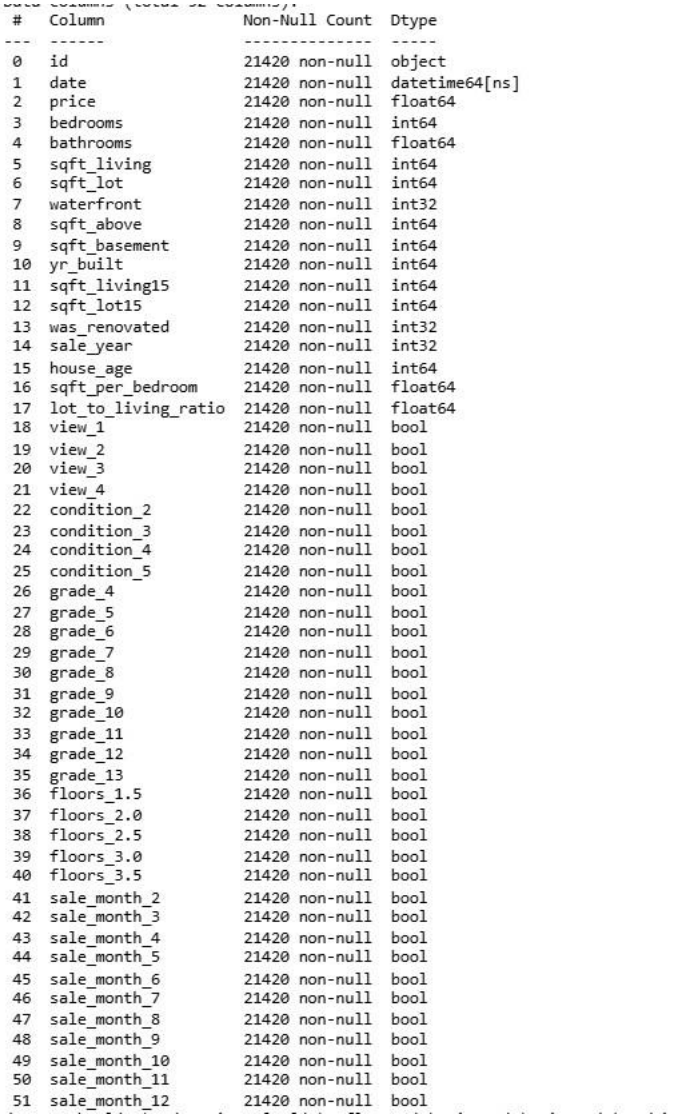
In one-hot encoding each category became a binary column(0 or 1). To avoid multi collinearity ,first category of each variable was dropped. Yr_renovated changed to binary was_renovated because it had lot of missing values.

Feature engineering is creating new variables from existing data to improve the predictive power of model by capturing more relevant patterns or relationships.Raw data may not directly capture relationships that drive house prices.We create new variables **sale_yr , sale_month, house_age, sqft_per_bedroom,lot_to_living_ratio**.Redundancy is reduced as house_age replace yr_built and sale_year in a way that is more relecant to price prediction.Lot_to_living_ratio reflects real estate preferences for balanced indoor-outdoor space.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 21420 entries, 0 to 21596
Data columns (total 23 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   id                   21420 non-null  object
 1   date                 21420 non-null  datetime64[ns]
 2   price                21420 non-null  float64
 3   bedrooms             21420 non-null  int64
 4   bathrooms            21420 non-null  float64
 5   sqft_living          21420 non-null  int64
 6   sqft_lot             21420 non-null  int64
 7   floors               21420 non-null  float64
 8   waterfront           21420 non-null  category
 9   view                 21420 non-null  int64
 10  condition            21420 non-null  int64
 11  grade                21420 non-null  int64
 12  sqft_above           21420 non-null  int64
 13  sqft_basement        21420 non-null  int64
 14  yr_built             21420 non-null  int64
 15  sqft_living15        21420 non-null  int64
 16  sqft_lot15           21420 non-null  int64
 17  was_renovated        21420 non-null  int64
 18  sale_year            21420 non-null  int32
 19  sale_month           21420 non-null  int32
 20  house_age            21420 non-null  int64
 21  sqft_per_bedroom     21420 non-null  float64
 22  lot_to_living_ratio  21420 non-null  float64
dtypes: category(1), datetime64[ns](1), float64(5), int32(2), int64(13), object(1)
memory usage: 3.6+ MB
```

Figure 24 : Variables added and before dummy creation



#	Column	Non-Null Count	Dtype
0	id	21420 non-null	object
1	date	21420 non-null	datetime64[ns]
2	price	21420 non-null	float64
3	bedrooms	21420 non-null	int64
4	bathrooms	21420 non-null	float64
5	sqft_living	21420 non-null	int64
6	sqft_lot	21420 non-null	int64
7	waterfront	21420 non-null	int32
8	sqft_above	21420 non-null	int64
9	sqft_basement	21420 non-null	int64
10	yr_built	21420 non-null	int64
11	sqft_living15	21420 non-null	int64
12	sqft_lot15	21420 non-null	int64
13	was_renovated	21420 non-null	int32
14	sale_year	21420 non-null	int32
15	house_age	21420 non-null	int64
16	sqft_per_bedroom	21420 non-null	float64
17	lot_to_living_ratio	21420 non-null	float64
18	view_1	21420 non-null	bool
19	view_2	21420 non-null	bool
20	view_3	21420 non-null	bool
21	view_4	21420 non-null	bool
22	condition_2	21420 non-null	bool
23	condition_3	21420 non-null	bool
24	condition_4	21420 non-null	bool
25	condition_5	21420 non-null	bool
26	grade_4	21420 non-null	bool
27	grade_5	21420 non-null	bool
28	grade_6	21420 non-null	bool
29	grade_7	21420 non-null	bool
30	grade_8	21420 non-null	bool
31	grade_9	21420 non-null	bool
32	grade_10	21420 non-null	bool
33	grade_11	21420 non-null	bool
34	grade_12	21420 non-null	bool
35	grade_13	21420 non-null	bool
36	floors_1.5	21420 non-null	bool
37	floors_2.0	21420 non-null	bool
38	floors_2.5	21420 non-null	bool
39	floors_3.0	21420 non-null	bool
40	floors_3.5	21420 non-null	bool
41	sale_month_2	21420 non-null	bool
42	sale_month_3	21420 non-null	bool
43	sale_month_4	21420 non-null	bool
44	sale_month_5	21420 non-null	bool
45	sale_month_6	21420 non-null	bool
46	sale_month_7	21420 non-null	bool
47	sale_month_8	21420 non-null	bool
48	sale_month_9	21420 non-null	bool
49	sale_month_10	21420 non-null	bool
50	sale_month_11	21420 non-null	bool
51	sale_month_12	21420 non-null	bool

Figure 25 : Variables after dummy creation

There are 52 variables after dummy creation.

We also created dummy variables for **sale_month** to check seasonal trends due to market dynamics. So 11 dummy variables (12-1) is created so model can capture non linear seasonal effects.

9. Estimate linear regression model

Steps to create model

- Removed irrelevant columns(price,id,date) from predictors. Target variable price is log-transformed for better modelling.

Figure 26. Log transformed housing prices.

After log transformation in (figure)more symmetric and bell-shaped approximating normal distribution is seen.Log transformation compress high values.This makes data evenly distributed,reduce model bias suitable for regression.

- Add intercept : Included constant term using sm.add_constant() for OLS(ordinary least squares) regression.
- Converted Boolean dummy variables to integers to ensure compatibility with regression modelling.(fig 27)

```
view_1                float64
view_2                float64
view_3                float64
view_4                float64
condition_2           float64
condition_3           float64
condition_4           float64
condition_5           float64
grade_4               float64
grade_5               float64
grade_6               float64
grade_7               float64
grade_8               float64
grade_9               float64
grade_10              float64
grade_11              float64
grade_12              float64
grade_13              float64
floors_1.5            float64
floors_2.0            float64
floors_2.5            float64
floors_3.0            float64
floors_3.5            float64
sale_month_2          float64
sale_month_3          float64
sale_month_4          float64
sale_month_5          float64
sale_month_6          float64
sale_month_7          float64
sale_month_8          float64
sale_month_9          float64
sale_month_10         float64
sale_month_11         float64
sale_month_12         float64
```

Figure 27. Changed bool to float

- Split the data into training and testing set (80/20) for evaluation of model.
- Standardized numerical features(excluding the intercept) using StandarScaler for consistent coefficient interpretation.
- Printed data types and checked for missing values in features and the target variable.
- Fitted an OLS linear regression model on the training data using statsmodels.

```
Any NaNs in X_train: 0
Any NaNs in y_train: 0
```

Linear Regression Model Summary

OLS Regression Results

Dep. Variable:	price	R-squared:	0.668
Model:	OLS	Adj. R-squared:	0.667
Method:	Least Squares	F-statistic:	730.9
Date:	Wed, 07 May 2025	Prob (F-statistic):	0.00
Time:	07:26:57	Log-Likelihood:	-3899.3
No. Observations:	17136	AIC:	7895.
Df Residuals:	17088	BIC:	8267.
Df Model:	47		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	13.0524	0.002	5616.274	0.000	13.048	13.057
bedrooms	-0.0013	0.006	-0.216	0.829	-0.013	0.010
bathrooms	0.0621	0.004	14.432	0.000	0.054	0.071
sqft_living	0.0425	0.005	8.301	0.000	0.032	0.052
sqft_lot	-0.0013	0.007	-0.194	0.846	-0.014	0.012
waterfront	0.0304	0.003	10.333	0.000	0.025	0.036
sqft_above	0.0263	0.005	5.573	0.000	0.017	0.036
sqft_basement	0.0391	0.003	11.546	0.000	0.032	0.046
yr_built	-0.0823	0.002	-44.591	0.000	-0.086	-0.079
sqft_living15	0.0822	0.004	20.515	0.000	0.074	0.090
sqft_lot15	-0.0186	0.004	-4.879	0.000	-0.026	-0.011
was_renovated	0.0044	0.003	1.746	0.081	-0.001	0.009
sale_year	0.0351	0.007	4.737	0.000	0.021	0.050
house_age	0.0829	0.002	44.916	0.000	0.079	0.087
sqft_per_bedroom	0.0373	0.007	5.207	0.000	0.023	0.051
lot_to_living_ratio	0.0160	0.005	2.916	0.004	0.005	0.027
view 1	0.0177	0.002	7.500	0.000	0.013	0.022
view_2	0.0141	0.002	5.873	0.000	0.009	0.019
view_3	0.0129	0.002	5.328	0.000	0.008	0.018
view_4	0.0233	0.003	7.739	0.000	0.017	0.029
condition_2	0.0082	0.006	1.457	0.145	-0.003	0.019
condition_3	0.1239	0.029	4.224	0.000	0.066	0.181
condition_4	0.1214	0.027	4.492	0.000	0.068	0.174
condition_5	0.0935	0.017	5.591	0.000	0.061	0.126
grade_4	-0.0085	0.010	-0.822	0.411	-0.029	0.012
grade_5	-0.0095	0.032	-0.296	0.767	-0.072	0.053
grade_6	0.0436	0.088	0.496	0.620	-0.129	0.216
grade_7	0.2050	0.150	1.367	0.172	-0.089	0.499
grade_8	0.2815	0.137	2.053	0.040	0.013	0.550
grade_9	0.2776	0.100	2.775	0.006	0.082	0.474
grade_10	0.2236	0.069	3.249	0.001	0.089	0.359
grade_11	0.1526	0.042	3.645	0.000	0.071	0.235
grade_12	0.0831	0.021	3.968	0.000	0.042	0.124
grade_13	0.0367	0.008	4.353	0.000	0.020	0.053
floors_1.5	0.0166	0.003	6.321	0.000	0.011	0.022
floors_2.0	0.0330	0.004	9.112	0.000	0.026	0.040
floors_2.5	0.0115	0.002	4.785	0.000	0.007	0.016
floors_3.0	0.0475	0.003	18.044	0.000	0.042	0.053
floors_3.5	0.0041	0.002	1.760	0.079	-0.000	0.009
sale_month_2	0.0015	0.003	0.441	0.660	-0.005	0.008
sale_month_3	0.0170	0.004	4.522	0.000	0.010	0.024
sale_month_4	0.0247	0.004	6.297	0.000	0.017	0.032
sale_month_5	0.0304	0.005	5.549	0.000	0.020	0.041
sale_month_6	0.0275	0.006	4.417	0.000	0.015	0.040
sale_month_7	0.0279	0.006	4.430	0.000	0.016	0.040
sale_month_8	0.0258	0.006	4.342	0.000	0.014	0.037
sale_month_9	0.0233	0.006	4.066	0.000	0.012	0.035
sale_month_10	0.0226	0.006	3.898	0.000	0.011	0.034
sale_month_11	0.0204	0.005	3.917	0.000	0.010	0.031
sale_month_12	0.0185	0.005	3.506	0.000	0.008	0.029

Omnibus:	54.223	Durbin-Watson:	1.987

```
==============================================================================
Omnibus:                       54.223   Durbin-Watson:                   1.987
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               60.573
Skew:                          -0.094   Prob(JB):                     7.03e-14
Kurtosis:                       3.222   Cond. No.                     8.26e+15
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 1.56e-27. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
R²: 0.6678
Adjusted R²: 0.6669

Significant Variables (p < 0.05):
                            Variable  Coefficient  P-Value Direction
const                          const      13.0524   0.0000  Positive
bathrooms                  bathrooms       0.0621   0.0000  Positive
sqft_living              sqft_living       0.0425   0.0000  Positive
waterfront                waterfront       0.0304   0.0000  Positive
sqft_above                sqft_above       0.0263   0.0000  Positive
sqft_basement          sqft_basement       0.0391   0.0000  Positive
yr_built                    yr_built      -0.0823   0.0000  Negative
sqft_living15          sqft_living15       0.0822   0.0000  Positive
sqft_lot15                sqft_lot15      -0.0186   0.0000  Negative
sale_year                  sale_year       0.0351   0.0000  Positive
house_age                  house_age       0.0829   0.0000  Positive
sqft_per_bedroom    sqft_per_bedroom       0.0373   0.0000  Positive
lot_to_living_ratio lot_to_living_ratio    0.0160   0.0035  Positive
view_1                        view_1       0.0177   0.0000  Positive
view_2                        view_2       0.0141   0.0000  Positive
view_3                        view_3       0.0129   0.0000  Positive
view_4                        view_4       0.0233   0.0000  Positive
condition_3              condition_3       0.1239   0.0000  Positive
condition_4              condition_4       0.1214   0.0000  Positive
condition_5              condition_5       0.0935   0.0000  Positive
grade_8                      grade_8       0.2815   0.0401  Positive
grade_9                      grade_9       0.2776   0.0055  Positive
grade_10                    grade_10       0.2236   0.0012  Positive
grade_11                    grade_11       0.1526   0.0003  Positive
grade_12                    grade_12       0.0831   0.0001  Positive
grade_13                    grade_13       0.0367   0.0000  Positive
floors_1.5                floors_1.5       0.0166   0.0000  Positive
floors_2.0                floors_2.0       0.0330   0.0000  Positive
floors_2.5                floors_2.5       0.0115   0.0000  Positive
floors_3.0                floors_3.0       0.0475   0.0000  Positive
sale_month_3            sale_month_3       0.0170   0.0000  Positive
sale_month_4            sale_month_4       0.0247   0.0000  Positive
sale_month_5            sale_month_5       0.0304   0.0000  Positive
sale_month_6            sale_month_6       0.0275   0.0000  Positive
sale_month_7            sale_month_7       0.0279   0.0000  Positive
sale_month_8            sale_month_8       0.0258   0.0000  Positive
sale_month_9            sale_month_9       0.0233   0.0000  Positive
sale_month_10          sale_month_10       0.0226   0.0001  Positive
sale_month_11          sale_month_11       0.0204   0.0001  Positive
sale_month_12          sale_month_12       0.0185   0.0005  Positive
```

Figure 28 . Regression model summary

- Model output displayed model summary , R^2(it measures the proportion of the variance in dependent variable that is explained by independent variables,value range 0-1), and adjusted R^2 to evaluate performance of model.

[If R^2 is 1 then it indicate model explain most of variability in target variable.Lower means do not explain much]

- Identified statistically significant variables ($p<0.05$) and analysed their coefficients and effect directions on log(price).

Variable effects

Variable	Effect Description
const	A one-unit increase in const (standardized) increases log(price) by 13.0524 (e.g., const = 1 vs. 0)
bathrooms	A one-unit increase in bathrooms (standardized) increases log(price) by 0.0621 (per standard deviation)
sqft_living	A one-unit increase in sqft_living (standardized) increases log(price) by 0.0425 (per standard deviation)
waterfront	A one-unit increase in waterfront (standardized) increases log(price) by 0.0304 (per standard deviation)
sqft_above	A one-unit increase in sqft_above (standardized) increases log(price) by 0.0263 (per standard deviation)
sqft_basement	A one-unit increase in sqft_basement (standardized) increases log(price) by 0.0391 (per standard deviation)
yr_built	A one-unit increase in yr_built (standardized) increases log(price) by -0.0823 (per standard deviation)
sqft_living15	A one-unit increase in sqft_living15 (standardized) increases log(price) by 0.0822 (per standard deviation)
sqft_lot15	A one-unit increase in sqft_lot15 (standardized) increases log(price) by -0.0186 (per standard deviation)
sale_year	A one-unit increase in sale_year (standardized) increases log(price) by 0.0351 (per standard deviation)
house_age	A one-unit increase in house_age (standardized) increases log(price) by 0.0829 (per standard deviation)
sqft_per_bedroom	A one-unit increase in sqft_per_bedroom (standardized) increases log(price) by 0.0373 (per standard deviation)
lot_to_living_ratio	A one-unit increase in lot_to_living_ratio (standardized) increases log(price) by 0.0160 (per standard deviation)
view_1	A one-unit increase in view_1 (standardized) increases log(price) by 0.0177 (per standard deviation)
view_2	A one-unit increase in view_2 (standardized) increases log(price) by 0.0141 (per standard deviation)
view_3	A one-unit increase in view_3 (standardized) increases log(price) by 0.0129 (per standard deviation)
view_4	A one-unit increase in view_4 (standardized) increases log(price) by 0.0233 (per standard deviation)
condition_3	A one-unit increase in condition_3 (standardized) increases log(price) by 0.1239 (per standard deviation)
condition_4	A one-unit increase in condition_4 (standardized) increases log(price) by 0.1214 (per standard deviation)
condition_5	A one-unit increase in condition_5 (standardized) increases log(price) by 0.0935 (per standard deviation)
grade_8	A one-unit increase in grade_8 (standardized) increases log(price) by 0.2815 (per standard deviation)
grade_9	A one-unit increase in grade_9 (standardized) increases log(price) by 0.2776 (per standard deviation)
grade_10	A one-unit increase in grade_10 (standardized) increases log(price) by 0.2236 (per standard deviation)
grade_11	A one-unit increase in grade_11 (standardized) increases log(price) by 0.1526 (per standard deviation)
grade_12	A one-unit increase in grade_12 (standardized) increases log(price) by 0.0831 (per standard deviation)
grade_13	A one-unit increase in grade_13 (standardized) increases log(price) by 0.0367 (per standard deviation)
floors_1.5	A one-unit increase in floors_1.5 (standardized) increases log(price) by 0.0166 (per standard deviation)
floors_2.0	A one-unit increase in floors_2.0 (standardized) increases log(price) by 0.0330 (per standard deviation)
floors_2.5	A one-unit increase in floors_2.5 (standardized) increases log(price) by 0.0115 (per standard deviation)
floors_3.0	A one-unit increase in floors_3.0 (standardized) increases log(price) by 0.0475 (per standard deviation)
sale_month_3	A one-unit increase in sale_month_3 (standardized) increases log(price) by 0.0170 (per standard deviation)
sale_month_4	A one-unit increase in sale_month_4 (standardized) increases log(price) by 0.0247 (per standard deviation)
sale_month_5	A one-unit increase in sale_month_5 (standardized) increases log(price) by 0.0304 (per standard deviation)
sale_month_6	A one-unit increase in sale_month_6 (standardized) increases log(price) by 0.0275 (per standard deviation)
sale_month_7	A one-unit increase in sale_month_7 (standardized) increases log(price) by 0.0279 (per standard deviation)
sale_month_8	A one-unit increase in sale_month_8 (standardized) increases log(price) by 0.0258 (per standard deviation)
sale_month_9	A one-unit increase in sale_month_9 (standardized) increases log(price) by 0.0233 (per standard deviation)
sale_month_10	A one-unit increase in sale_month_10 (standardized) increases log(price) by 0.0226 (per standard deviation)
sale_month_11	A one-unit increase in sale_month_11 (standardized) increases log(price) by 0.0204 (per standard deviation)
sale_month_12	A one-unit increase in sale_month_12 (standardized) increases log(price) by 0.0185 (per standard deviation)

Figure 29. Variable effects

Linear regression model report

a. Goodness of fit

- R^2 (Coefficient of Determination):0.668

The model explains 66.8% of the variance in house prices, indicating reasonably good fit.

- Adjusted R^2: 0.667

After adjusting model complexity,explanatiory power remains strong,confirming the predictors contribute meaningfully to the model.

- F-statistic:730.9 (p = 0.000)

The overall model is statistically significant, suggesting predictors are collectively have a significant effect on house prices.

Residual analysis

- Durbin-Watson statistic (1.987)

It is a statistic used to measure auto correlation in the residuals from regression analysis.It helps determine if residuals are independent from one another. <2 means positive correlation, >2 negative correlation. Close to 2, no correlation.

In our case, 1.987 supports independence assumption in linear regression.

- Omnibus test(omnibus = 54.223,prob.=0.000)

It is used to check if the residuals (errors) of a regression model are normally distributed. It combines skewness(symmetry) and kurtosis(tailedness) into one.A low p-value (<0.05) shows residuals are not normally distributed.

In our case residuals are not normally distributed.

- Jarque-Bera (p < 0.001)

It is test for normality of residuals, also based on skewness and kurtosis.A high p-value means residuals normally distributed.

In our case, residuals are not perfectly normal.May affect confidence intervlas,but less critical if sample is large.

b. Significance of model coefficients (p <0.05)

Those variables that have positive effects on price are :

Structural features

- House_age (0.0829, p=0.00) - Older houses (since last renovation) are more valuable.
- Sqft_living15 (0.0822 ,p=0.00) - Larger living space in the neighbourhood increases value.
- Bathrooms(0.0621, p=0.00) – More bathrooms lead to higher prices.
- Sqft-living (0.0425, p=0.00) – Larger interior living space positively impacts price.
- Waterfront (0.0304, p= 0.00)- Waterfront properties are significantly more valuable.
- Floors : Multistory homes (floors_1.5 to 3) are more valuable (eg. Floors_3.0 : 0.0475,p=0.00)

Other feature

- Views : Better views(view_1 to 4) increase price (eg. View_4 : +0.0233, p=0.00)

Quality and condition

- Condition_3 (0.1239, p=0),condition_4(0.1214, p=0),condition_05 (0.0935 ,p =0) .Better condition means higher price.
- Higher grade (8-13) : Premium for superior construction (eg : grade_8 : 0.2815,p = 0.04 and grade_13: 0.0367 , p=0)

Seasonal effects

- Sale_month_3 to sale_month_12 (for example, sale_month5 : 0.0304 , p= 0): Spring/summer sales yield higher prices.

Variable with strong negative effect is :

Yr_built (-0.0823, p=0) - older construction years reduce price , likely due to depreciation or outdated features.

Sqft_lot15 (-0.0186 , p=0): Larger lot size (without living space) may decrease value.

Insignificant predictors

- Bedrooms(-0.013, p=0.829) : No meaningful impact when living space(sqft_living) is controlled.
- Was_renovated (0.0044, p=0.081) : mariginal effect, not statistically significant.
- Lower grade categories (grade4 to grade7) : No significant price difeerences (all p>0.05).
- Sale_month_2 (0.015, p=0.660) : February sales show no seasonal premium.

  There are multicollinearity issues,we will check VIF.

  From this , we should focus on living space and quality,sqft_living/sqft_living 15 and grade,condition.Premium features like waterfront and view_4 justifies higher pricing.List properties in sale_month_3 – sale month 12 for better returns.Bedroom count and low-tier grades do not significantly influence prices.

c. Direction of significant variable (p<0.05)

Variables with positive direction are :

sqft_living15, house_age,bathrooms, sqft_living,waterfront, sqft_above,sqft_basement,sale_year,sqft_per_bedroom,lot_to_living_ratio, view 1 to 4,condition 3 to 5,grade 8 to 13,floors 1.5 to 3,sale month 3 to 12.

Variables with negative direction are :

Yr_built,sqft_lot15.

d. Effect of variables with significant relationships.

1. Strongest practical impact:

- House_age (+0.0829) – A 1 year increase since renovation raises price by 8.3% ,holding other factors constant.
- Yr_built (-0.0823) : A 1 year older construction year reduces price by 8.2% (depreciation).
- Sqft_living15 (+0.0822) : A 100 sqft increase in neighbourhood living space boosts price by 8.2%

2.Premium features:

- Waterfront(+0.0304) : Waterfront properties are 3% more expensive.

- View_4(+0.0233): The best view category adds 2.3% to price.

3.Quality and condition :

- Condition_5 (+0.0935) : Top-condition homes are 9.4% more valuable than baseline.
- Grade_8(+0.2815) : A grade-8 home is 28.15 % more expensive than lower grades.

4.Seasonal effects:

- Sale_month_5 (+0.0304) ; May sales are 3% pricier than January (reference month).

From this we understand, list in spring/summer increase sales by 1.7-3%Target waterfront/views for premium pricing.Sqft_living ,grade and condition highest ROI.

10. Construct histogram of residuals. Explanations for the histogram form.

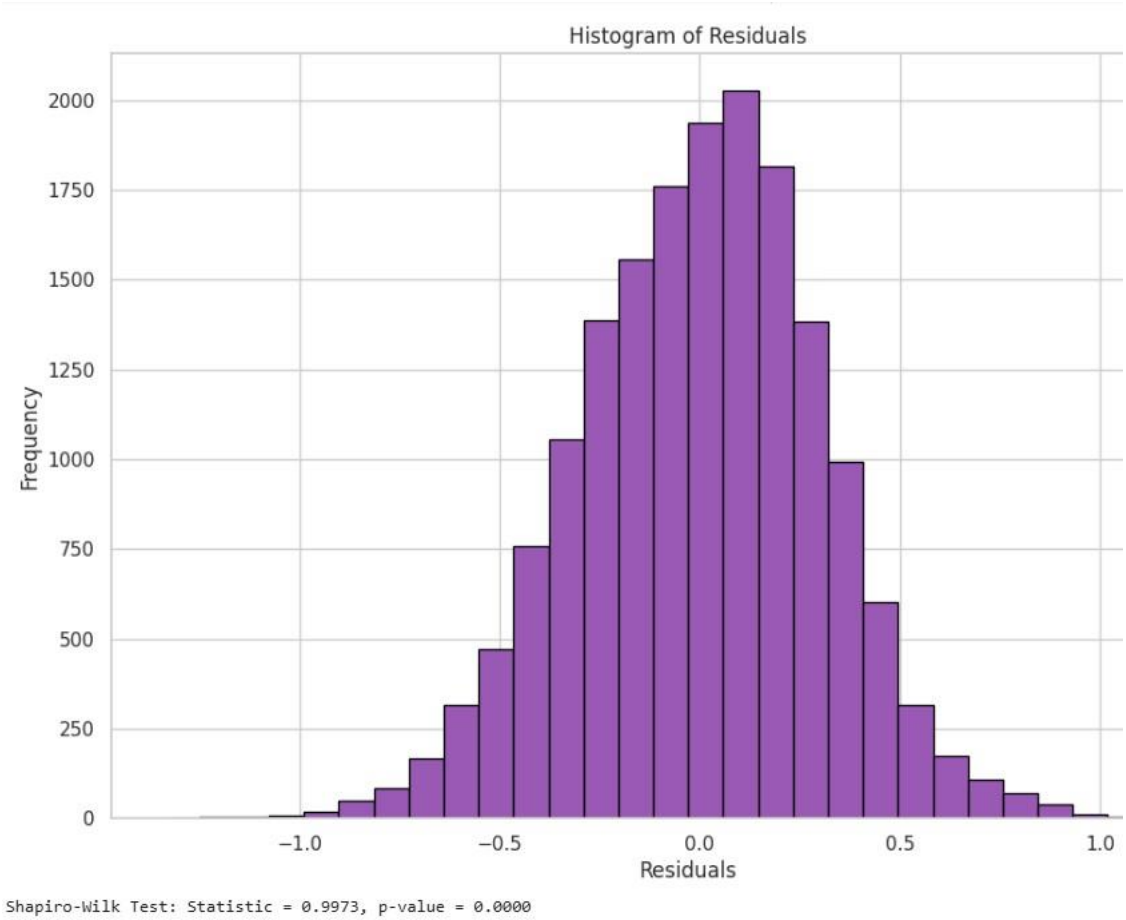


Figure 30. Histogram of residuals

Analysis of histogram

- The residuals appear to be symmetrically distributed around zero,so the model predictions are unbiased (no systematic over or under prediction).
- There is a concentration of residuals near zero.Most predictors are accurate with small errors.
- Residuals extend from -1 to +1 suggest some outliers where the model predictions were significantly off.

Shapiro-Wilk test interpretation

This test checks whether a residuals from a regression model follows a normal distribution.Null hypothesis is data is normally distributed,alternate hypothesis – data is not normally distributed.If p>0.05 –fail to reject null hypothesis,means residuals are normal. P<0.05 reject null hypothesis, residuals are not normal.

In our case,Shapiro-wilk statistic shows 0.9973 very close to 1 and p-value of 0.000 (<0.05). So we reject null hypothesis, but residuals are nearly normal.The non normality is likely because of heavy tail (outliers) than skewness.

The model is good fit for most data. A small number of houses were poorly predicted (residuals near +-1). These can be unique properties (luxury home,extreme conditions) or missing key predictors.(eg. Proximity to landmarks). Near normality and symmetry means regression assumptions (linearity and homoscedasticity) are likely met.

11.Scatter plot of residuals and decision regarding outliers.

Scatter plot of residual

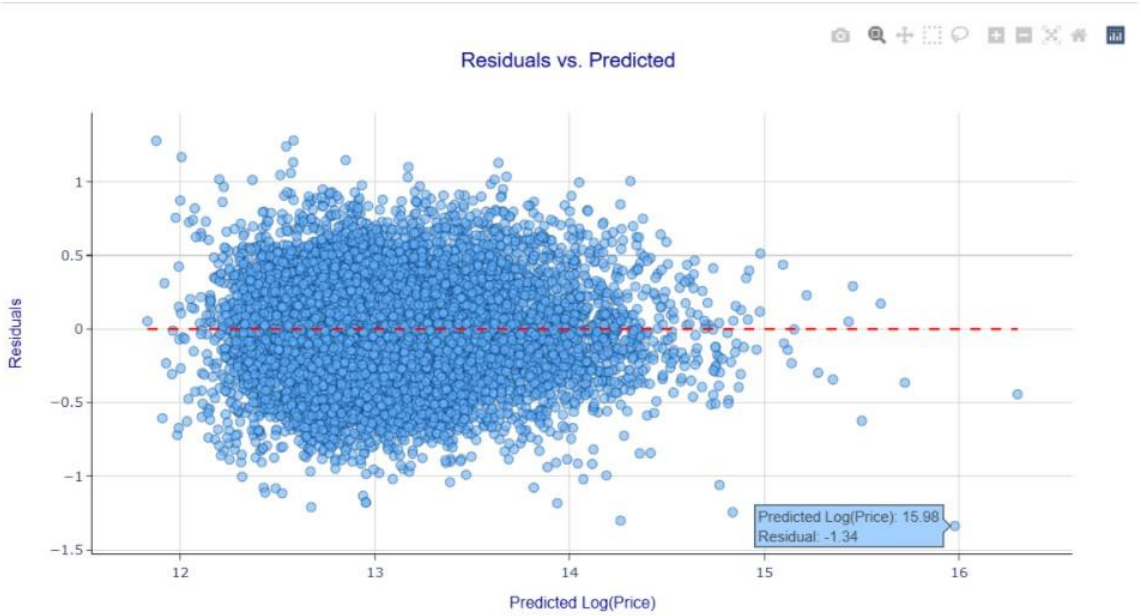


Figure 31 .Scatterplot of residual

- The residuals (actual – predicted) is plotted gainst predicted log(price) values as scatter plot .
- The dashed red line at residual =0 represents perfect prediction.
- The highlighted point for example show predicted =15.98,residual = -1.34. This means significant underprediction.Actual value was much higher than model prediction.
- Most residuals cluster near zero, suggesting good model fit for typical homes.
- The spread of residuals appears reltively constant across predicted values (homoscedasticity).But extreme residuals suggest potential outliers or missing features for luxury properties.

a.Exclude Outliers and Re-run the Model

We remove outliers and fit a new model.

```
Number of outliers: 65

Original Model: R² = 0.6678215900891277 Adjusted R² = 0.666907943947636
Model without Outliers: R² = 0.6778547317613015 Adjusted R² = 0.676965298194526
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.678
Model:                            OLS   Adj. R-squared:                  0.677
Method:                 Least Squares   F-statistic:                     762.1
Date:                Wed, 07 May 2025   Prob (F-statistic):               0.00
Time:                        13:39:20   Log-Likelihood:                -3521.0
No. Observations:               17071   AIC:                             7138.
Df Residuals:                   17023   BIC:                             7510.
Df Model:                          47
Covariance Type:            nonrobust
------------------------------------------------------------------------------
```

Figure 32.1: Re-running model after outlier removal

Identified 65 extreme outliers and removed where standardised residuals exceeded 3 standard deviations from the mean. Re-ran regression model on cleaned dataset (n = 17071)

12. Model performance after outlier removal- Comparison with old model

1. R^2 improved from 0.668 to 0.678.

   New model explains a marginally higher proportion of variance in price variable.

2. Adjusted R^2 improved from 0.667 to 0.677

   New model has better fit even after adjusting the number of predictors ,reinforcing positive effect of outlier removal.

3. Sample size: 17136 to 17071.

   Removal of 65 outliers had a minimal impact on the sample size, preserved model robustness with a large dataset.

4. F-Statistic and significance : 730.9,p(0.00) to 762.1 ,p(0.00)

   Both models are highly significant,but new model has higher F-statistic.So, there is slightly stronger overall fit after removing outliers.

5. Log-likelihood : -7899.3 to -3521.0

   It measures how well model explains observed data. Less negative value indicates a better fit.

6. AIC (Akaike Information Criterion):-7895 to 7138

Balances model fit and complexity, penalize extra parameters. Low AIC means better model.

7. BIC(Bayesian Information Criterion) : - 8267 to 7510
   Similar to AIC but imposes a stronger penalty for complexity favouring simpler models. Lower BIC means better model.
8. Omnibus test : 26.960 (p =0.00)

   It rejects null hypothesis of normal distributed residuals.

9. Jarque-Bera Test (JB) : 26.813 (p= 1.51 e-06)
   Confirms significant deviation from normality. Residuals have skewness or kurtosis issue.
10. Skewness : -0.094 to -0.089
    New model is marginally better .Negative value show left tail,but negligible difference.
11. Kurtosis: 3.222 to 2.924
    Leptokurtic to platykurtic(heavier tail than normal to lighter tail than normal).New model has few extreme values than a normal distribution.

	coef	std err	t	P>|t|	[0.025	0.975]
const	13.0525	0.002	5726.108	0.000	13.048	13.057
bedrooms	-0.0080	0.006	-1.386	0.166	-0.019	0.003
bathrooms	0.0630	0.004	14.884	0.000	0.055	0.071
sqft_living	0.0475	0.005	9.334	0.000	0.038	0.057
sqft_lot	0.0054	0.007	0.820	0.412	-0.008	0.018
waterfront	0.0317	0.003	10.874	0.000	0.026	0.037
sqft_above	0.0300	0.005	6.407	0.000	0.021	0.039
sqft_basement	0.0426	0.003	12.719	0.000	0.036	0.049
yr_built	-0.0849	0.002	-46.771	0.000	-0.088	-0.081
sqft_living15	0.0816	0.004	20.705	0.000	0.074	0.089
sqft_lot15	-0.0191	0.004	-5.109	0.000	-0.026	-0.012
was_renovated	0.0041	0.002	1.675	0.094	-0.001	0.009
sale_year	0.0342	0.007	4.708	0.000	0.020	0.048
house_age	0.0855	0.002	47.093	0.000	0.082	0.089
sqft_per_bedroom	0.0286	0.007	4.041	0.000	0.015	0.043
lot_to_living_ratio	0.0100	0.005	1.844	0.065	-0.001	0.021
view_1	0.0176	0.002	7.631	0.000	0.013	0.022
view_2	0.0142	0.002	6.041	0.000	0.010	0.019
view_3	0.0143	0.002	5.992	0.000	0.010	0.019
view_4	0.0245	0.003	8.269	0.000	0.019	0.030
condition_2	0.0106	0.006	1.820	0.069	-0.001	0.022
condition_3	0.1417	0.031	4.634	0.000	0.082	0.202
condition_4	0.1371	0.028	4.864	0.000	0.082	0.192
condition_5	0.1028	0.017	5.896	0.000	0.069	0.137
grade_4	-0.0085	0.010	-0.838	0.402	-0.028	0.011
grade_5	-0.0102	0.031	-0.323	0.747	-0.072	0.052
grade_6	0.0421	0.086	0.489	0.625	-0.127	0.211
grade_7	0.2064	0.147	1.406	0.160	-0.081	0.494
grade_8	0.2831	0.134	2.110	0.035	0.020	0.546
grade_9	0.2782	0.098	2.841	0.004	0.086	0.470
grade_10	0.2233	0.067	3.315	0.001	0.091	0.355
grade_11	0.1522	0.041	3.713	0.000	0.072	0.233
grade_12	0.0838	0.021	4.085	0.000	0.044	0.124
grade_13	0.0362	0.008	4.390	0.000	0.020	0.052
floors_1.5	0.0167	0.003	6.490	0.000	0.012	0.022
floors_2.0	0.0355	0.004	9.968	0.000	0.028	0.042
floors_2.5	0.0124	0.002	5.224	0.000	0.008	0.017
floors_3.0	0.0492	0.003	19.028	0.000	0.044	0.054
floors_3.5	0.0041	0.002	1.783	0.075	-0.000	0.009
sale_month_2	0.0015	0.003	0.460	0.645	-0.005	0.008
sale_month_3	0.0177	0.004	4.788	0.000	0.010	0.025
sale_month_4	0.0249	0.004	6.471	0.000	0.017	0.032
sale_month_5	0.0311	0.005	5.792	0.000	0.021	0.042
sale_month_6	0.0274	0.006	4.474	0.000	0.015	0.039
sale_month_7	0.0268	0.006	4.345	0.000	0.015	0.039
sale_month_8	0.0253	0.006	4.336	0.000	0.014	0.037
sale_month_9	0.0231	0.006	4.106	0.000	0.012	0.034
sale_month_10	0.0215	0.006	3.791	0.000	0.010	0.033
sale_month_11	0.0196	0.005	3.832	0.000	0.010	0.030
sale_month_12	0.0176	0.005	3.411	0.001	0.008	0.028

Omnibus:	26.960	Durbin-Watson:	1.990
Prob(Omnibus):	0.000	Jarque-Bera (JB):	26.813
Skew:	-0.089	Prob(JB):	1.51e-06
Kurtosis:	2.924	Cond. No.	9.22e+15

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 1.23e-27. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

Figure 32.2 Re-run regression model

13. Stepwise elimination of insignificant independent variables

Tried different methods like keeping some variables ,combining them into one like the ordinal ones as they shows multi-collinearity,giving weights to ordinal variables,manually removing one by one checking if it is significant or not ,and using a python code to remove insignificant variable ,check VIF ,then remove high VIF variable to reach to significant variables.

Removing variables

The eigen value 1.23e-27 indicate severe multi collinearity.So that means some predictors are highly correlated with each other.

From the variables ,we remove insignifiant ones namely:

- bedrooms (p=0.166)
- sqft_lot (p=0.4172)
- was_renovated(p=0.094)
- condition_2(p=0.0677)
- grade_5 (p=0.0734)
- floors_3.5 (p=0.075)
- sale_month_2(p=0.64)

All remaining variables have p<0.05 .But multicollinearity exists. We run the model.

```
Removing grade_5 (p=0.7467)
Removing sale_month_2 (p=0.6403)
Removing sqft_lot (p=0.4172)
Removing bedrooms (p=0.1663)
Removing was_renovated (p=0.0967)
Removing floors_3.5 (p=0.0734)
Removing condition_2 (p=0.0677)

=== Final Model Summary ===
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.678
Model:                            OLS   Adj. R-squared:                  0.677
Method:                 Least Squares   F-statistic:                     894.9
Date:                Thu, 08 May 2025   Prob (F-statistic):               0.00
Time:                        10:23:36   Log-Likelihood:                -3527.1
No. Observations:               17071   AIC:                             7136.
Df Residuals:                   17030   BIC:                             7454.
Df Model:                          40
Covariance Type:            nonrobust
================================================================================================
                           coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------
const                   13.0525      0.002   5725.292      0.000      13.048      13.057
bathrooms                0.0635      0.004     15.169      0.000       0.055       0.072
sqft_living              0.0422      0.003     14.417      0.000       0.036       0.048
waterfront               0.0318      0.003     10.939      0.000       0.026       0.038
sqft_above               0.0256      0.003      8.370      0.000       0.020       0.032
```

Figure 33.1 : stepwise variable removal and running regression model

sqft_above	0.0256	0.003	8.370	0.000	0.020	0.032
sqft_basement	0.0398	0.003	14.889	0.000	0.035	0.045
yr_built	-0.0858	0.002	-50.024	0.000	-0.089	-0.082
sqft_living15	0.0806	0.004	20.632	0.000	0.073	0.088
sqft_lot15	-0.0172	0.003	-5.634	0.000	-0.023	-0.011
sale_year	0.0338	0.007	4.649	0.000	0.020	0.048
house_age	0.0863	0.002	50.344	0.000	0.083	0.090
sqft_per_bedroom	0.0373	0.004	10.149	0.000	0.030	0.045
lot_to_living_ratio	0.0137	0.003	4.597	0.000	0.008	0.020
view_1	0.0177	0.002	7.663	0.000	0.013	0.022
view_2	0.0142	0.002	6.021	0.000	0.010	0.019
view_3	0.0145	0.002	6.079	0.000	0.010	0.019
view_4	0.0245	0.003	8.281	0.000	0.019	0.030
condition_3	0.0916	0.012	7.574	0.000	0.068	0.115
condition_4	0.0903	0.011	8.064	0.000	0.068	0.112
condition_5	0.0738	0.007	10.342	0.000	0.060	0.088
grade_4	-0.0051	0.002	-2.135	0.033	-0.010	-0.000
grade_6	0.0707	0.007	10.603	0.000	0.058	0.084
grade_7	0.2550	0.011	23.006	0.000	0.233	0.277
grade_8	0.3277	0.010	31.240	0.000	0.307	0.348
grade_9	0.3110	0.008	38.230	0.000	0.295	0.327
grade_10	0.2461	0.006	40.504	0.000	0.234	0.258
grade_11	0.1663	0.004	38.173	0.000	0.158	0.175
grade_12	0.0912	0.003	29.712	0.000	0.085	0.097
grade_13	0.0393	0.002	15.944	0.000	0.034	0.044
floors_1.5	0.0161	0.003	6.293	0.000	0.011	0.021
floors_2.0	0.0354	0.004	9.978	0.000	0.028	0.042
floors_2.5	0.0123	0.002	5.185	0.000	0.008	0.017
floors_3.0	0.0493	0.003	19.090	0.000	0.044	0.054
sale_month_3	0.0167	0.003	5.639	0.000	0.011	0.022
sale_month_4	0.0238	0.003	7.844	0.000	0.018	0.030
sale_month_5	0.0299	0.005	6.254	0.000	0.020	0.039
sale_month_6	0.0260	0.006	4.611	0.000	0.015	0.037
sale_month_7	0.0254	0.006	4.475	0.000	0.014	0.037
sale_month_8	0.0239	0.005	4.452	0.000	0.013	0.034
sale_month_9	0.0219	0.005	4.223	0.000	0.012	0.032
sale_month_10	0.0203	0.005	3.876	0.000	0.010	0.031
sale_month_11	0.0184	0.005	3.900	0.000	0.009	0.028
sale_month_12	0.0164	0.005	3.429	0.001	0.007	0.026

Omnibus:	26.968	Durbin-Watson:	1.991
Prob(Omnibus):	0.000	Jarque-Bera (JB):	26.806
Skew:	-0.089	Prob(JB):	1.51e-06
Kurtosis:	2.923	Cond. No.	1.24e+16

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 6.58e-28. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

Figure 33.2 Re-ran model after initial variable removal

Still multicollinearity exists. So we fix multi collinearity.

	Variable	VIF
2	sqft_living	inf
4	sqft_above	inf
5	sqft_basement	inf
6	yr_built	inf
9	sale_year	inf
10	house_age	inf
17	condition_3	28.171825
0	const	24.470261
18	condition_4	24.139627
22	grade_7	23.646939
23	grade_8	21.183823
24	grade_9	12.757948
19	condition_5	9.798134
21	grade_6	8.514927

Figure 34. VIF high variables

Some variables have infinite VIF(variance inflation factor). It is extent of multicollinearity for each predictor.Sqft_above and sqft_basement are parts of sqft_living.Yr_built and house_age are similar,sale_year could be collinear with time-related variables.So we drop sqft_above,sqft_basement,house_age,sale_yr.

OLS Regression Results

Dep. Variable:	price	R-squared:	0.676
Model:	OLS	Adj. R-squared:	0.675
Method:	Least Squares	F-statistic:	789.9
Date:	Thu, 08 May 2025	Prob (F-statistic):	0.00
Time:	22:54:53	Log-Likelihood:	-3566.1
No. Observations:	17071	AIC:	7224.
Df Residuals:	17025	BIC:	7580.
Df Model:	45		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	13.0525	0.002	5711.372	0.000	13.048	13.057
bedrooms	-0.0059	0.006	-1.030	0.303	-0.017	0.005
bathrooms	0.0683	0.004	16.317	0.000	0.060	0.077
sqft_living	0.0980	0.010	9.833	0.000	0.078	0.118

Figure 35 .Re-run model

Then we again eliminate variables with p-value greater than 0.05.

```
Removing sale_month_10 (p=0.8402)
Removing sale_month_11 (p=0.7885)
Removing grade_5 (p=0.7370)
Removing sale_month_2 (p=0.6119)
Removing sale_month_9 (p=0.5973)
Removing sqft_lot (p=0.5427)
Removing sale_month_8 (p=0.3944)
Removing sale_month_7 (p=0.4505)
Removing bedrooms (p=0.3069)
Removing sale_month_6 (p=0.2891)
Removing sale_month_12 (p=0.1708)
Removing was_renovated (p=0.1597)
Removing floors_3.5 (p=0.1033)
Removing condition_2 (p=0.0616)
```

Figure 36. Remove insignificant variables

Then we again run the model

```
--- Final Model (After p-value Elimination) ---
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.676
Model:                            OLS   Adj. R-squared:                  0.675
Method:                 Least Squares   F-statistic:                     1146.
Date:                Thu, 08 May 2025   Prob (F-statistic):               0.00
Time:                        22:54:55   Log-Likelihood:                -3573.4
No. Observations:               17071   AIC:                             7211.
Df Residuals:                   17039   BIC:                             7459.
Df Model:                          31
Covariance Type:            nonrobust
==============================================================================
```

Figure 37 . Re-run model

.Again we calculate VIF, combine grade_4 and grade_5 into grade_4_5, similarly we do for grades 6 and 7,8 and 9,10 and 11, grade_12+ for grade_12 and grade_13.

```
Combined grade columns: ['grade_4_5', 'grade_6_7', 'grade_8_9', 'grade_10_11', 'grade_12+']
```

Figure 38. Combined grade

And again check VIF. Then we map condition variable to ordinal values since it has high VIF.

```
                Variable         VIF
17          condition_3  179.025221
18          condition_4  152.204065
19          condition_5   58.185966
3           sqft_living   18.604813
10     sqft_per_bedroom    9.548020
4              sqft_lot    8.309400
16          condition_2    6.524521
1              bedrooms    6.323373
11  lot_to_living_ratio    5.635643
0                 const    5.072743
```

```
]:  1 # Map to ordinal values (1=Poor, 5=Excellent)
    2 X_train_final['condition_ordinal'] = (
    3     X_train_final['condition_3'] * 3 +
    4     X_train_final['condition_4'] * 4 +
    5     X_train_final['condition_5'] * 5
```

Figure 39. High VIF values (top).Combining ordinal_values to condition(bottom).

Sqft_living and sqft_per_bedroom are correlated. So dropping sqft_per_bedroom.

Then we repeat the process ,eliminate statistically insignificant variables.

```
==============================================================================
                        FINAL REFINED MODEL SUMMARY
==============================================================================
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.662
Model:                            OLS   Adj. R-squared:                  0.662
Method:                 Least Squares   F-statistic:                     1336.
Date:                Thu, 08 May 2025   Prob (F-statistic):               0.00
Time:                        22:55:27   Log-Likelihood:                -3928.2
No. Observations:               17071   AIC:                             7908.
Df Residuals:                   17045   BIC:                             8110.
Df Model:                          25
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         13.0725      0.005   2504.458      0.000      13.062      13.083
```

bedrooms	-0.0303	0.003	-9.899	0.000	-0.036	-0.024
bathrooms	0.0645	0.004	15.146	0.000	0.056	0.073
sqft_living	0.1534	0.005	28.500	0.000	0.143	0.164
waterfront	0.0309	0.003	10.374	0.000	0.025	0.037
yr_built	-0.1651	0.004	-45.614	0.000	-0.172	-0.158
sqft_living15	0.0887	0.004	22.746	0.000	0.081	0.096
sqft_lot15	-0.0109	0.002	-4.504	0.000	-0.016	-0.006
was_renovated	0.0060	0.003	2.403	0.016	0.001	0.011
view_1	0.0190	0.002	8.063	0.000	0.014	0.024
view_2	0.0170	0.002	7.093	0.000	0.012	0.022
view_3	0.0180	0.002	7.448	0.000	0.013	0.023
view_4	0.0278	0.003	9.206	0.000	0.022	0.034
floors_1.5	0.0141	0.003	5.506	0.000	0.009	0.019
floors_2.0	0.0265	0.003	8.251	0.000	0.020	0.033
floors_2.5	0.0106	0.002	4.443	0.000	0.006	0.015
floors_3.0	0.0457	0.003	17.573	0.000	0.041	0.051
sale_month_3	0.0165	0.002	6.986	0.000	0.012	0.021
sale_month_4	0.0228	0.002	9.638	0.000	0.018	0.027
sale_month_5	0.0120	0.002	5.080	0.000	0.007	0.017
grade_4_5	-0.0471	0.002	-26.364	0.000	-0.051	-0.044
grade_6_7	-0.1023	0.004	-25.432	0.000	-0.110	-0.094
grade_8_9	0.0553	0.004	14.628	0.000	0.048	0.063

Figure 40. Re-run model after combining variables

The refined OLS model achieves R^2= 0.662 ,with all retained variables significant (p,0.05). The intercept 13.07 represents baseline log(price)

grade_10_11	0.0709	0.003	25.587	0.000	0.065	0.076
grade_12+	0.0249	0.002	13.058	0.000	0.021	0.029
condition_ordinal	0.0080	0.001	10.703	0.000	0.007	0.009

Omnibus:	23.260	Durbin-Watson:	1.995
Prob(Omnibus):	0.000	Jarque-Bera (JB):	23.302
Skew:	-0.089	Prob(JB):	8.71e-06
Kurtosis:	3.029	Cond. No.	10.9

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

VIF CHECK FOR REFINED MODEL

	Variable	VIF
3	sqft_living	5.237350
0	const	5.005990
22	grade_8_9	3.598432
21	grade_6_7	3.427893
2	bathrooms	3.296880
6	sqft_living15	2.792743
23	grade_10_11	2.653449

5	yr_built	2.400189
14	floors_2.0	1.896800
1	bedrooms	1.707251
12	view_4	1.656193
4	waterfront	1.597754
24	grade_12+	1.315210
16	floors_3.0	1.241387
13	floors_1.5	1.210938
25	condition_ordinal	1.173161
20	grade_4_5	1.150689
8	was_renovated	1.150303
7	sqft_lot15	1.070594
11	view_3	1.067598
10	view_2	1.057097
15	floors_2.5	1.048424
19	sale_month_5	1.031377
18	sale_month_4	1.030478
17	sale_month_3	1.028689
9	view_1	1.023045

Figure 41. VIF for refined model

Thus significant variables are

SIGNIFICANT VARIABLES (p < 0.05)

	Variable	Coefficient	P-value	Impact
const	const	13.0725	0.0000	Positive
grade_10_11	grade_10_11	0.0709	0.0000	Positive
grade_8_9	grade_8_9	0.0553	0.0000	Positive
grade_6_7	grade_6_7	-0.1023	0.0000	Negative
grade_4_5	grade_4_5	-0.0471	0.0000	Negative
sale_month_5	sale_month_5	0.0120	0.0000	Positive
sale_month_4	sale_month_4	0.0228	0.0000	Positive
sale_month_3	sale_month_3	0.0165	0.0000	Positive
floors_3.0	floors_3.0	0.0457	0.0000	Positive
floors_2.5	floors_2.5	0.0106	0.0000	Positive
floors_2.0	floors_2.0	0.0265	0.0000	Positive
floors_1.5	floors_1.5	0.0141	0.0000	Positive
view_4	view_4	0.0278	0.0000	Positive
view_3	view_3	0.0180	0.0000	Positive
view_2	view_2	0.0170	0.0000	Positive
view_1	view_1	0.0190	0.0000	Positive
sqft_lot15	sqft_lot15	-0.0109	0.0000	Negative
sqft_living15	sqft_living15	0.0887	0.0000	Positive
yr_built	yr_built	-0.1651	0.0000	Negative
waterfront	waterfront	0.0309	0.0000	Positive
sqft_living	sqft_living	0.1534	0.0000	Positive
bathrooms	bathrooms	0.0645	0.0000	Positive
bedrooms	bedrooms	-0.0303	0.0000	Negative
grade_12+	grade_12+	0.0249	0.0000	Positive
condition_ordinal	condition_ordinal	0.0080	0.0000	Positive
was_renovated	was_renovated	0.0060	0.0163	Positive

Figure 42. Significant variables according to this model

14 . Analysis of prediction results- confidence intervals

```
Predictions and confidence intervals saved to 'prediction_results_with_ci.csv'

Sample of Prediction Results:
       Actual_Price  Predicted_Price  CI_Lower_Price  CI_Upper_Price  Predicted_Log_Price  CI_Lower_Log_Price  CI_Upper_Log_Pri
ce
6132       337500.0    432439.818073   237821.592095   786320.684110            12.977201           12.379280           13.5751
21
8993       680000.0    392361.437859   215858.170795   713187.520440            12.879941           12.282381           13.4775
01
559        331500.0    347371.884158   191101.204987   631430.336213            12.758154           12.160564           13.3557
44
11931      571000.0    405553.621885   223093.761357   737239.759631            12.913011           12.315352           13.5106
70
15176      431000.0    481585.557188   264921.421010   875446.276904            13.084841           12.487192           13.6824
90
```

Figure 43 : Confidence intervals

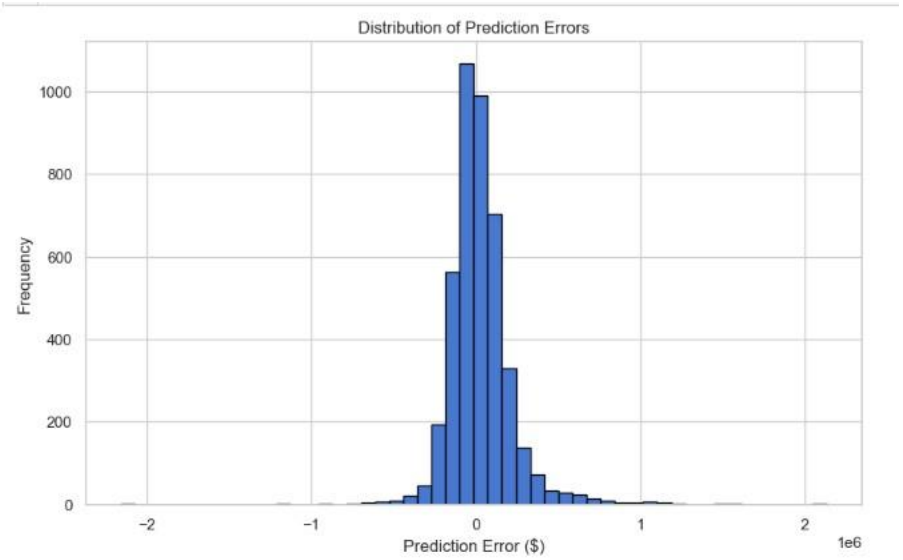
Sample predictions showing actual vs. predicted prices with 95% confidence intervals (CI) in both dollar and log scales.The asymmetric CIs (e.g:$ 432K +_ $274K/-$195K) due to log-normal transformation.

```python
# Calculate coverage
within_ci = (results_df['Actual_Price'] >= results_df['CI_Lower_Price']) & \
            (results_df['Actual_Price'] <= results_df['CI_Upper_Price'])
coverage = within_ci.mean() * 100
print(f"Percentage of actual prices within 95% confidence intervals: {coverage:.2f}%")
```

```
Percentage of actual prices within 95% confidence intervals: 94.05%
```

Figure 44 : Percentage of actual prices within 95% interval

94.05% of actual prices fall within their 95% CIs, indicating well- calibrated uncertainity estimates-slightly conservative but statistically valid.




```
Prediction Error Statistics:
count    4.284000e+03
mean     2.476025e+04
std      1.962357e+05
min     -2.157855e+06
25%     -7.915543e+04
50%      4.577606e+03
75%      1.037733e+05
max      2.129683e+06
Name: Error, dtype: float64
```

Figure 45 : Distribution of prediction error

Histogram of prediction errors(actual-predicted),most errors cluster near to zero (mean $24.8k) but right skew reveals overestimates for high-value homes(max. error : $ 2.1M).

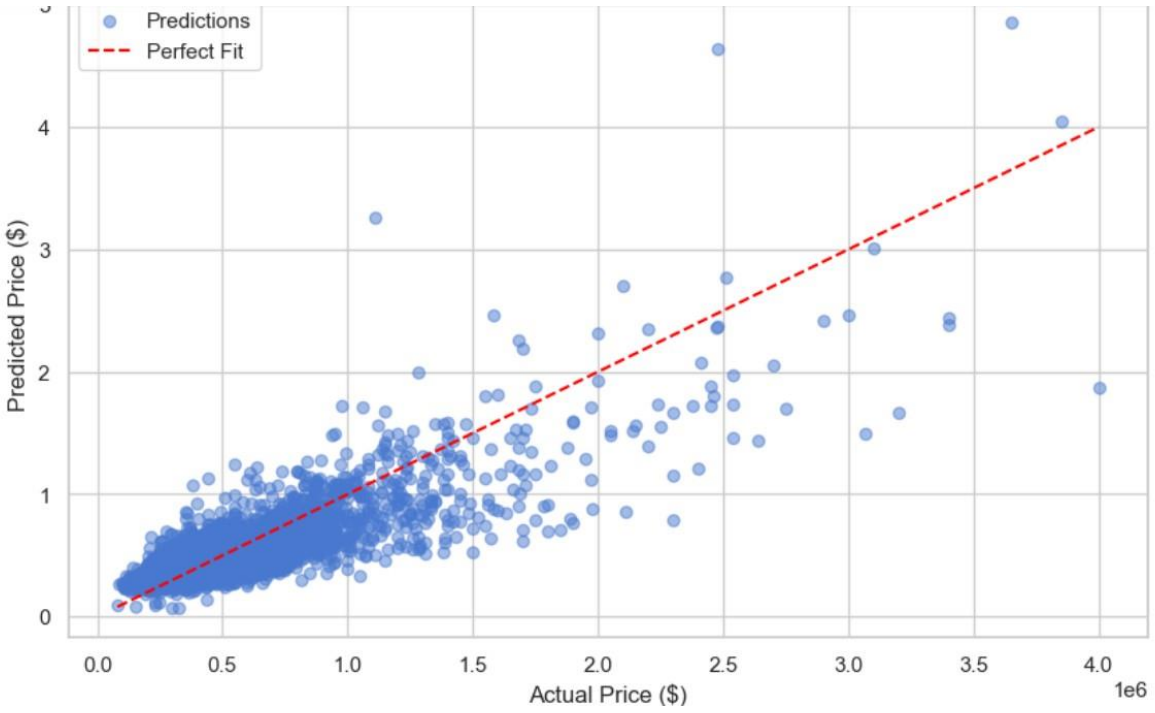


Figure 46: Scatterplot actual v.s predicted

Points deviating from the line reflect model biases especially in higher price ranges.

15 . Applying Random forest regression

```
Random Forest Test RMSE (Price Scale): 174659.4602
Linear Model Test RMSE (Price Scale): 197768.8933 (for comparison)

Random Forest predictions and prediction intervals saved to 'rf_prediction_results_with_pi.csv'

Sample of Random Forest Prediction Results:
       Actual_Price  Predicted_Price  PI_Lower_Price  PI_Upper_Price  Predicted_Log_Price  PI_Lower_Log_Price  PI_Upper_Log_Pri
ce
6132       337500.0    283666.025355   266593.169641   389092.282503            12.555556           12.493483           12.8715
74
8993       680000.0    457217.773692   400119.909839   564464.356664            13.032917           12.899522           13.2436
34
559        331500.0    316263.989182   295906.901778   355068.332585            12.664336           12.597804           12.7800
68
11931      571000.0    493756.549028   435481.806001   520122.998196            13.109800           12.984211           13.1618
23
15176      431000.0    406621.410070   357679.216157   459140.224193            12.915640           12.787395           13.0371
13

Percentage of actual prices within 95% prediction intervals: 38.19%
```

Figure 47 : Random forest regression intervals

We applied random forest regression.It outperforms linear regression (RMSE : $174.7k vs 197.8k) but prediction intervals are overly narrow. Only 38.2% of actual prices fall within 95% prediction intervals, shows underestimated uncertainty.

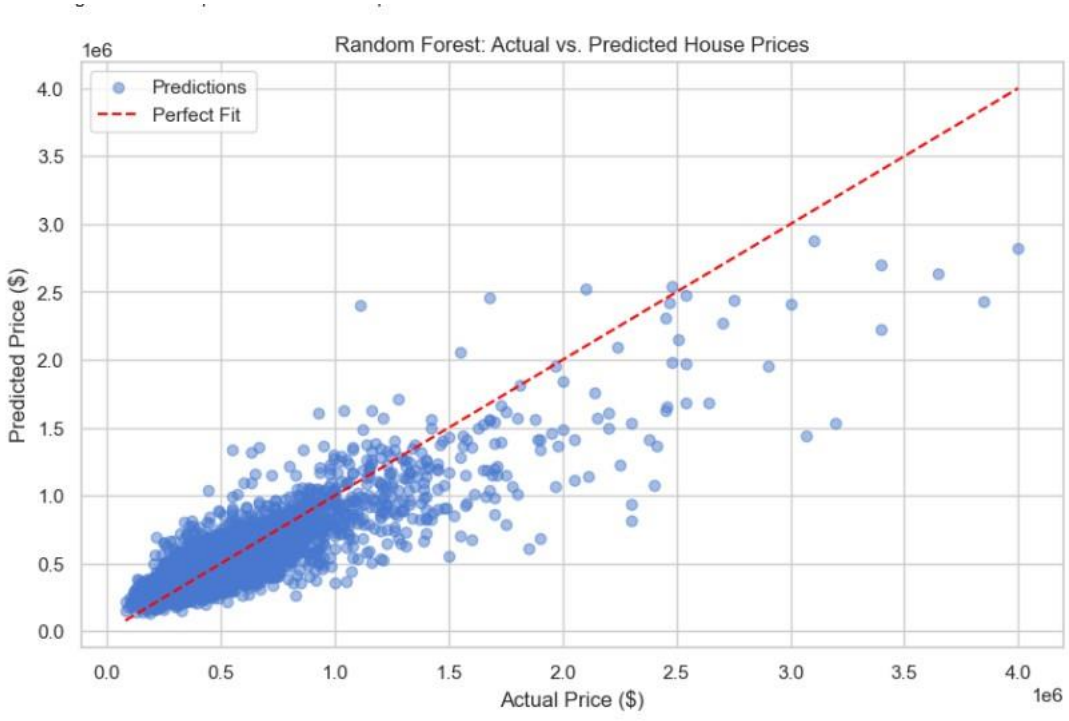


Figure 48: Random forest : actual v.s predicted scatterplot

Predictions cluster near the diagonal but there is systematic deviations at higher prices ($1M+) reveal model limitation in capturing premium home valuations.

Random forest reduces error by around 11.7% compared to linear model, but suffers from poor interval calibration.Scatter plot shows improved point estimates but highlights bias in luxury home predictions.

Another approach

Using python code , iteratively removed variables with insignificant variables (P>0.05) and high multi collinearity VIF>5.

```
=== Significant Variables ===
const, bathrooms, waterfront, sqft_basement, sqft_living15, sqft_lot15, house_age, sqft_per_bedroom, lot_to_living_ratio, vie
w_1, view_2, view_3, view_4, condition_4, condition_5, grade_4, grade_6, grade_8, grade_9, grade_10, grade_11, grade_12, grad
e_13, floors_1.5, floors_2.0, floors_2.5, floors_3.0, sale_month_3, sale_month_4, sale_month_5
```

These were the significant variables.

const	13.0526	0.002	5596.525	0.000	13.048	13.057
bathrooms	0.0884	0.004	22.735	0.000	0.081	0.096
waterfront	0.0301	0.003	10.107	0.000	0.024	0.036
sqft_basement	0.0445	0.003	14.529	0.000	0.038	0.050
sqft_living15	0.1005	0.004	27.246	0.000	0.093	0.108
sqft_lot15	-0.0140	0.003	-4.494	0.000	-0.020	-0.008
house_age	0.1650	0.003	47.439	0.000	0.158	0.172
sqft_per_bedroom	0.0574	0.003	16.808	0.000	0.051	0.064
lot_to_living_ratio	0.0081	0.003	2.654	0.008	0.002	0.014
view_1	0.0169	0.002	7.172	0.000	0.012	0.022
view_2	0.0133	0.002	5.500	0.000	0.009	0.018
view_3	0.0133	0.002	5.461	0.000	0.009	0.018
view_4	0.0233	0.003	7.673	0.000	0.017	0.029
condition_4	0.0103	0.003	4.033	0.000	0.005	0.015
condition_5	0.0230	0.003	9.116	0.000	0.018	0.028
grade_4	-0.0227	0.002	-9.718	0.000	-0.027	-0.018
grade_6	-0.0734	0.003	-28.253	0.000	-0.079	-0.068
grade_8	0.0979	0.003	33.160	0.000	0.092	0.104
grade_9	0.1463	0.003	45.304	0.000	0.140	0.153
grade_10	0.1349	0.003	43.249	0.000	0.129	0.141
grade_11	0.1009	0.003	34.736	0.000	0.095	0.107
grade_12	0.0603	0.003	23.102	0.000	0.055	0.065
grade_13	0.0290	0.002	12.090	0.000	0.024	0.034
floors_1.5	0.0227	0.003	8.758	0.000	0.018	0.028
floors_2.0	0.0428	0.004	11.957	0.000	0.036	0.050
floors_2.5	0.0151	0.002	6.254	0.000	0.010	0.020
floors_3.0	0.0486	0.003	18.393	0.000	0.043	0.054
sale_month_3	0.0149	0.002	6.278	0.000	0.010	0.020
sale_month_4	0.0209	0.002	8.848	0.000	0.016	0.026
sale_month_5	0.0110	0.002	4.655	0.000	0.006	0.016

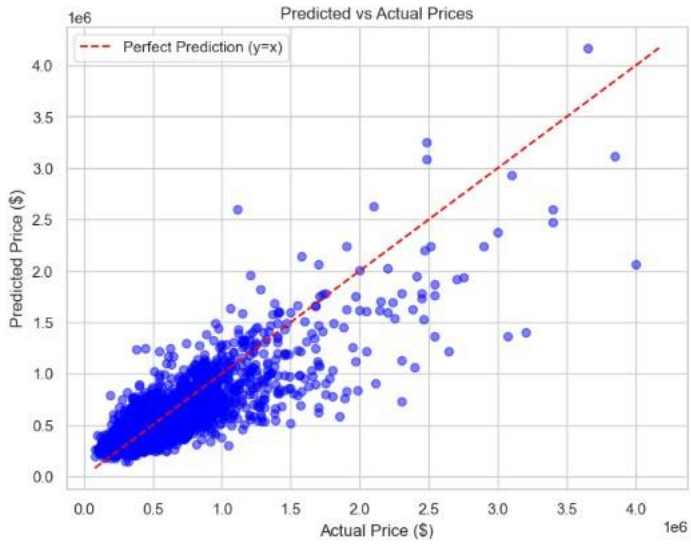
Omnibus:	99.024	Durbin-Watson:	1.988
Prob(Omnibus):	0.000	Jarque-Bera (JB):	103.342
Skew:	-0.168	Prob(JB):	3.63e-23
Kurtosis:	3.179	Cond. No.	4.45

This model also shows R^2= 0.662.Residuals here show near-normality with no auto correlation.

In the table below, sample predictions show large errors for high value homes with average width 25K in confidence intervals.

```
First 5 predictions with 95% confidence intervals (original price scale):
       predicted_price  ci_lower_price  ci_upper_price  true_price  absolute_error_price
6132     442208.885281   431682.080161   452992.391743    337500.0         104708.885281
8993     404820.204182   400121.561779   409574.022778    680000.0         275179.795818
559      344174.640667   339843.024085   348561.467583    331500.0          12674.640667
11931    395601.888548   389217.752319   402090.740171    571000.0         175398.111452
15176    496098.205556   489500.188622   502785.157569    431000.0          65098.205556

Prediction Summary (original price scale):
Average predicted price: $506048.12
Average confidence interval width: $25683.45
Mean Absolute Error: $127460.98
```



In graph :there is under estimation of luxury property.Mid range cluster close to ideal price.

16. Conclusion and pricing model

The final model shows us 29 significant variables that affect prices.The strongest determinants are :

- **Structural features** : Bathrooms(+9.2%per unit) , waterfront (+3.1%) and living area(+10.6% per standard deviation).
- **Quality grades :** Grade 9 homes command a +15.7% premium over baseline.Grade 6 shows -7.1 % penalty.
- **Temporal effects :** Spring sales(March-May) add +1.5% to +2.1% to the prices.

The model predicts price by the given formula :

$$\text{Price} = e^{(\beta_0 + \beta_1 * X_1 + \beta_2 * X_2 + \ldots + \beta_n * X_n)} - 1$$

Here n=29 , for example waterfront home (x1 =1),2 bathrooms(x2 =2)and grade 9 (x3=1)

Waterfront coefficient β_1=0.1463,bathrooms coeff. is β_2 =0.0884,grade_9 coeff. β_3=0.0301

So, price = e^(13.05 + 0.0884×2 + 0.0301×1 + 0.1463×1) − 1 ≈ $585,000

- **Model selection**

  Linear regression is preferred for interpretability and reliable confidence intervals despite higher RMSE.

For applications that require both accuracy and interpretability ,the linear model is preferable.Future work should address the limitations with luxury properties through advanced feature engineering or hybrid modelling approaches.