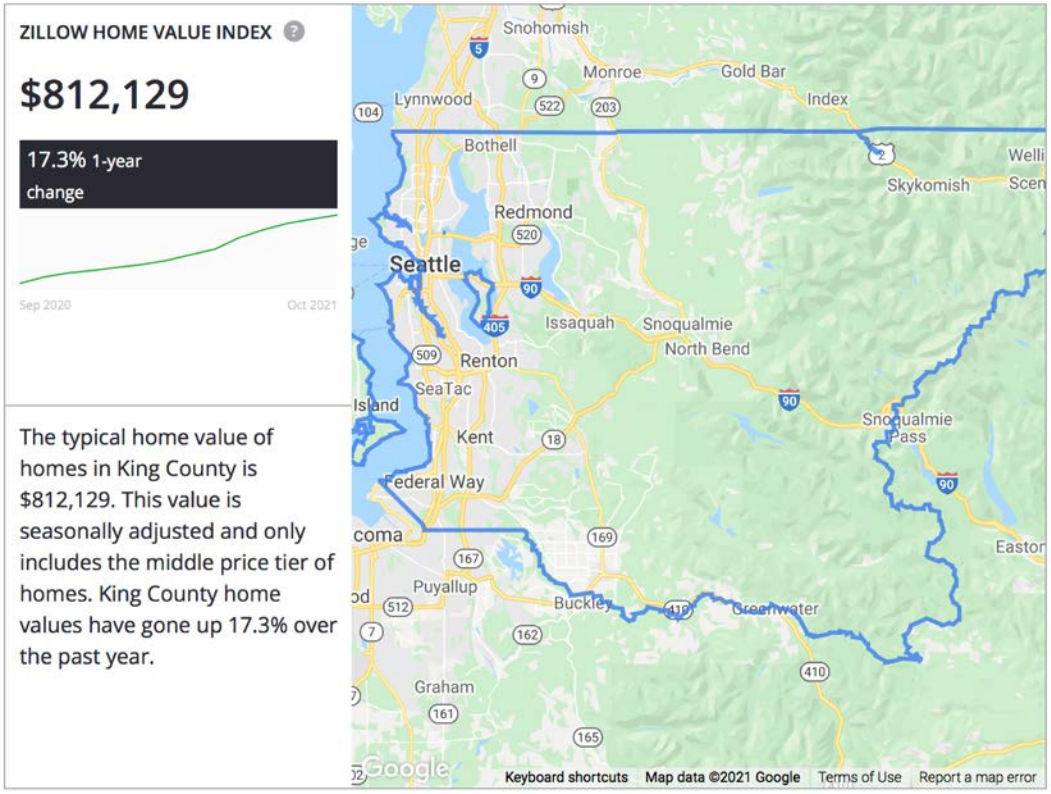
Predicting housing prices in King County

Housing prices in King County has been increasing over the years

Prices of mid-range homes has been increasing by around 8.8%[1] annually.

The average home is valued around USD 812,129 as of Oct 2021.

King County Home Values



[1] Compound annual growth rate of the average home price for from 2014 to 2021 from https://www.zillow.com/king-county-wa/home-values/

Prediction task

Using the King County house sale price dataset[1],

1. Identify the features which are related to price, and
2. Build a model to predict house prices in King County.

[1] Dataset includes homes sold between May 2014 and May 2015.

1. Exploratory Data Analysis

1. Exploratory Data Analysis

Overview of KC housing dataset

There are 21,313 observations (rows) and 21 variables (columns).

176 houses look like duplicates (same id), but had different transacted dates. These could be resale houses. We'll keep them in the analysis since they still represent a transaction.

Identifiers

1. `Id` Unique ID for each home sold

Dates

2. `Date` Date of the home sale
3. `Yr_built` The year the house was initially built
4. `Yr_renovated` The year of the house's last renovation

Numerical

3. `Price` Price of each home sold
4. `Bedrooms` Number of bedrooms
5. `Bathrooms` Number of bathrooms, (0.5 = toilet only, 0.75 = toilet + shower)
6. `Sqft_living` Square footage of the apartments interior living space
7. `Sqft_lot` Square footage of the land space
8. `Floors` Number of floors
9. `Sqft_above` The square footage of the interior housing space that is above ground level
10. `Sqft_basement` The square footage of the interior housing space that is below ground level
11. `Sqft_living15` The square footage of interior housing living space for the nearest 15 neighbors
12. `Sqft_lot15` The square footage of the land lots of the nearest 15 neighbors

Categorical

9. `Waterfront` A dummy variable for whether the apartment was overlooking the waterfront or not
10. `View` An index from 0 to 4 of how good the view of the property was
11. `Condition` An index from 1 (poor) to 5 (very good) on the condition of the apartment
12. `Grade` An index from 1 to 13, where 1-3 falls short of building construction and design, 7 has an average level of construction and design, and 11- 13 have a high quality level of construction and design

Location

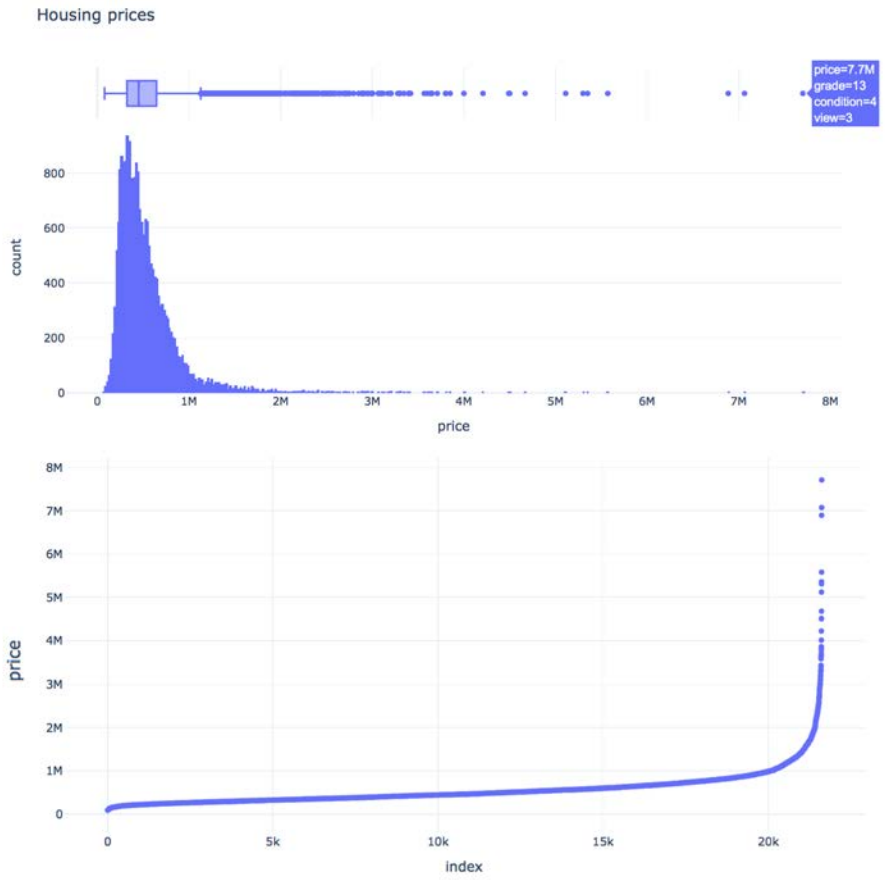
17. `Zipcode` What zipcode area the house is in
18. `Lat` Lattitude
19. `Long` Longitude

1. Exploratory Data Analysis

Price has many outliers and do not seem follow a linear trend.

Though there are many outliers, we decide to keep these entries as they appear to be houses with higher grade, condition and view scores.

Also, price does not seem to follow a normal distribution and will likely require some transformation.



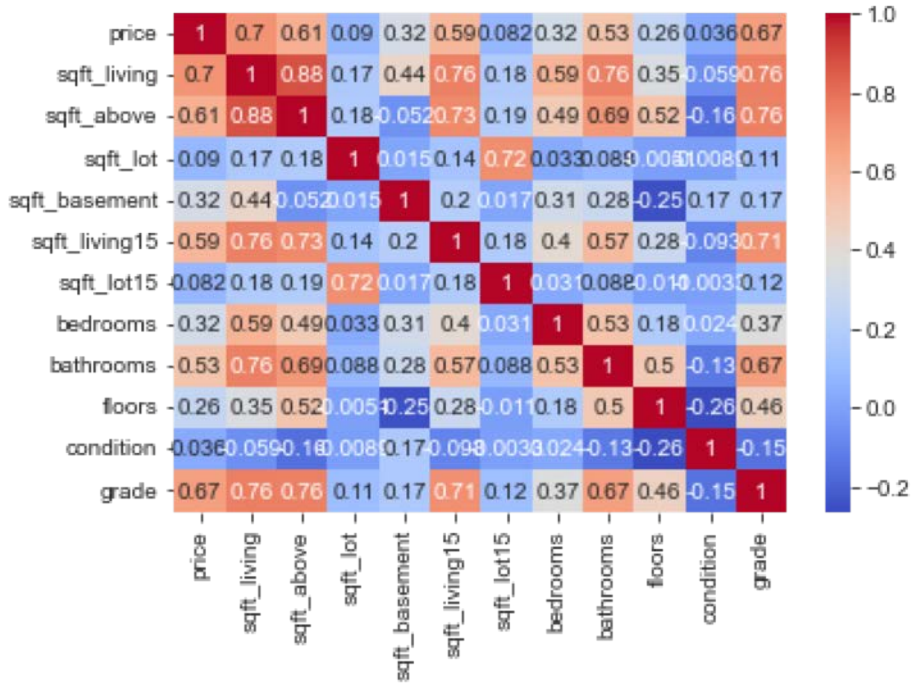
1. Exploratory Data Analysis

Likely to be good predictors of price:

1. sqft_living
2. sqft_above
3. sqft_living15
4. bathrooms
5. grade

These 5 variables are moderately correlated with price.

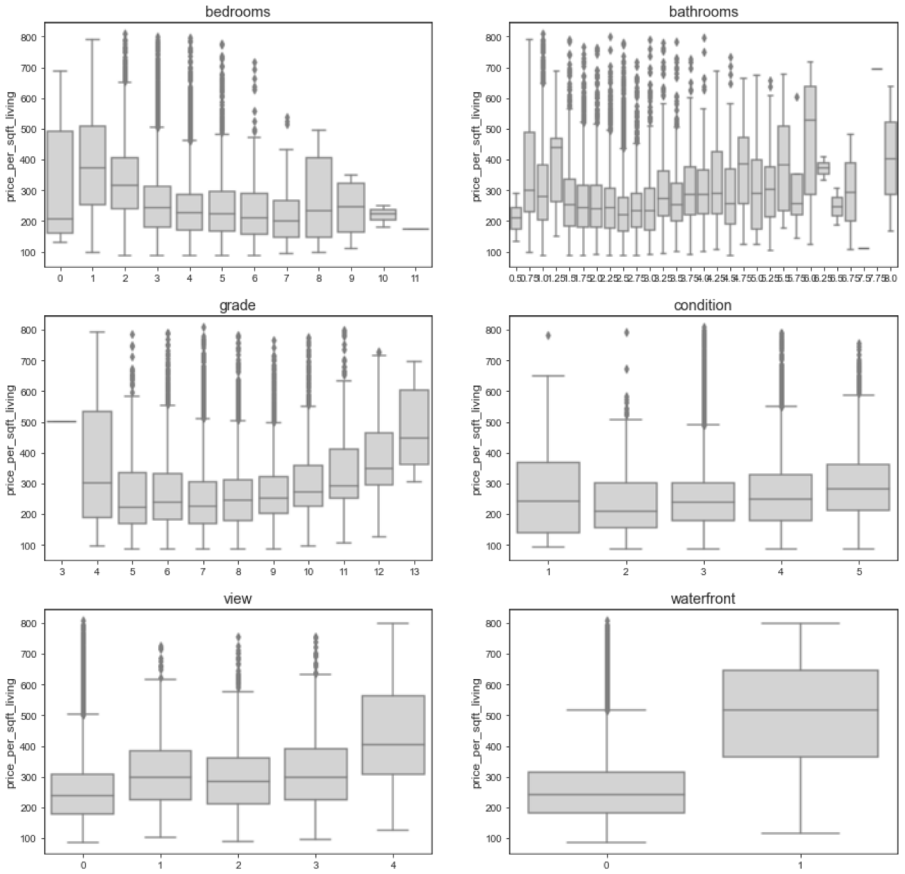
However, there are also notable correlations among variables which may pose multicollinearity problems. We will need to drop similar variables later.



1. Exploratory Data Analysis

We analyse if price per sqft_living differs across certain categories of homes.

1. Houses with better views have higher price_per_sqft_living.
2. Waterfront homes have higher price_per_sqft_living.
3. Price_per_sqft_living appears to decrease with more bedrooms.
4. Price_per_sqft_living appears to increase across higher grade.
5. Price_per_sqft_living has no discernable pattern with bathrooms and condition.

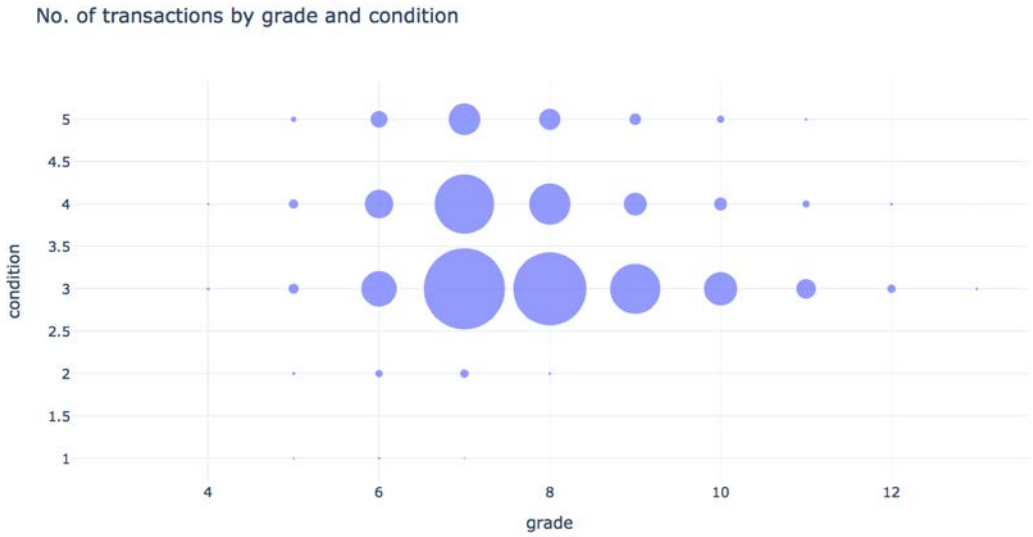


1. Exploratory Data Analysis

Grade and condition — are they the same?

From the bubble chart, we notice most of the transacted houses were grade 3 and above, and condition 7 and above.

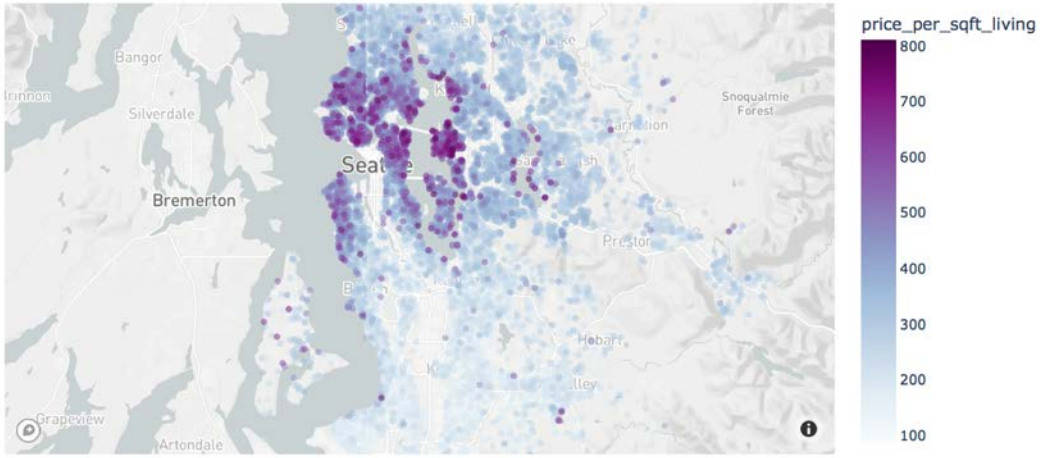
We would expect houses of a better grade to have a better corresponding condition. However, the weak negative correlation (r = 0.146) does not support this assumption.



1. Exploratory Data Analysis

Homes with higher price per sqft_living are clustered around downtown and waterfront areas.

As lat and long provide more granular geolocation info than postal code, we will retain these variables for analysis.



2. Data Preparation

Key principle: Keep as much info as possible first.

2. Data Preparation

Managing outliers

Bedrooms
House id 15870 has 33 bedrooms. We change it to 3 as it’s likely a It’s likely a data entry error.

Bathrooms
There are 10 entries where the houses have no bathrooms. While it could be a data entry error, it's also possible that the house does not have an attached bathroom (e.g. shared living spaces, which should be priced differently from normal houses). Given the uncertainty, we drop the 10 rows which should be inconsequential.

Price
Price is quite skewed, so we try a ln transformation.

3. Feature Engineering

3. Feature Engineering

We create some features for exploration…

Age and Reno_age
To replace the dates for year built and year renovated

Lat_1km and Long_1km
We bin lat and long into 1km x1kn grids – still more granular than postal code – to see if it helps to reduce noise.

Basement
We create a categorical basement variable where 1 = has basement and 0 = no basement.

Ln transform sqft variables
We also try ln transform on sqft_living, sqft_lot, sqft_above, sqft_basement, sqft_living15 and sqft_lot15 since the values are large.

4. Modelling

4. Modelling

For each model, we…

1. **Generated feature importances on the full model**
   Different models produce a different ranked list of variables, though mostly similar.
2. **Performed backward selection of features**
   Eliminate features that are least important or have high multicollinearity.
3. **Split training and testing data 80:20**
   Set seed = 21
4. **Performed standard scaling**
   Since variables are not homogenous
5. **Did cross-validation and hyperparameter tuning in every run**
   Using GridSearchCV, cv = 5. Because FOMO.
6. **Analysed residuals**
   To check if the models have sufficiently captured all the trends, leaving only white noise.
7. **Check for outliers and see if removing them improves models**

4. Modelling

We run Ordinary Least Squares Regression on the full model.

1. All features are significantly related to price ($p < \alpha < 0.05$)
2. Model did not sufficiently capture trends (residuals are not normal and do not have constant variance)
3. Multicollinearity issues – high variance inflation factors for age and reno age, the rest are moderate
4. Lots of outliers (using scipy.stats.outlier_test)

1

OLS Regression Results

Dep. Variable:	price	R-squared:	0.696
Model:	OLS	Adj. R-squared:	0.696
Method:	Least Squares	F-statistic:	2315.
Date:	Mon, 29 Nov 2021	Prob (F-statistic):	0.00
Time:	09:36:31	Log-Likelihood:	-2.3438e+05
No. Observations:	17190	AIC:	4.688e+05
Df Residuals:	17172	BIC:	4.689e+05
Df Model:	17		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	5.383e+05	1540.947	349.309	0.000	5.35e+05	5.41e+05
bedrooms	-3.624e+04	2015.345	-17.980	0.000	-4.02e+04	-3.23e+04
bathrooms	3.363e+04	2842.454	11.833	0.000	2.81e+04	3.92e+04
sqft_living	8.346e+04	1807.026	46.185	0.000	7.99e+04	8.7e+04
sqft_lot	4842.7114	2268.345	2.135	0.033	396.524	9288.898
floors	3721.3537	2190.561	1.699	0.089	-572.369	8015.076
waterfront	4.771e+04	1696.254	28.128	0.000	4.44e+04	5.1e+04
view	4.177e+04	1850.276	22.577	0.000	3.81e+04	4.54e+04
condition	1.848e+04	1728.564	10.693	0.000	1.51e+04	2.19e+04
grade	1.121e+05	2860.031	39.210	0.000	1.07e+05	1.18e+05
sqft_above	7.761e+04	1919.020	40.444	0.000	7.39e+04	8.14e+04
sqft_basement	2.758e+04	1758.484	15.683	0.000	2.41e+04	3.1e+04
zipcode	-3.069e+04	1989.678	-15.424	0.000	-3.46e+04	-2.68e+04
sqft_living15	1.19e+04	2669.841	4.459	0.000	6670.833	1.71e+04
sqft_lot15	-1.055e+04	2284.170	-4.618	0.000	-1.5e+04	-6072.042
age	9.275e+04	3838.221	24.166	0.000	8.52e+04	1e+05
reno_age	-1.485e+04	3905.951	-3.802	0.000	-2.25e+04	-7193.134
lat	8.24e+04	1677.402	49.124	0.000	7.91e+04	8.57e+04
long	-3.036e+04	2093.053	-14.505	0.000	-3.45e+04	-2.63e+04

Omnibus:	15227.229	Durbin-Watson:	1.985
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1771717.239
Skew:	3.770	Prob(JB):	0.00
Kurtosis:	52.161	Cond. No.	7.21e+15

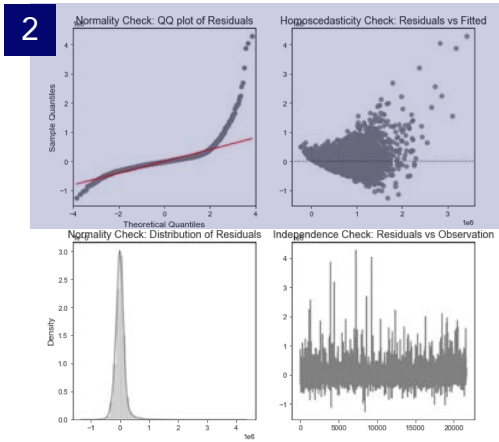
Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 1.87e-27. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

MSE = 40817966366.47772

MULTICOLLINEARITY TESTS
Variance Inflation Factor (VIF=1: not correlated, 1<VIF<5: moderately correlated, VIF>5: highly correlated)
VIF of bedrooms 1.7105016809749667
VIF of bathrooms 3.402603273486099
VIF of sqft_living inf
VIF of sqft_lot 2.1669184481446417
VIF of floors 2.0208545966996376
VIF of waterfront 1.2117318016927634
VIF of view 1.4417750643197815
VIF of condition 1.2583328681133006
VIF of grade 3.4448159545590482
VIF of sqft_above inf
VIF of sqft_basement inf
VIF of zipcode 1.6672098346301087
VIF of sqft_living15 3.001894278258008
VIF of sqft_lot15 2.197259948028945
VIF of age 6.2041800608983575
VIF of reno_age 6.42507217014261
VIF of lat 1.1849468357975164
VIF of long 1.8449516518604547

3

2



4

OUTLIER TEST

	id	date	price
300	3225069065	2014-06-24	3075000.0
656	3760500116	2014-11-20	3070000.0
1031	5700004029	2015-04-17	2450000.0
1164	1247600105	2014-10-20	5110800.0
1272	4389200765	2014-06-25	2300000.0
...	...	...	...
20441	1925059254	2015-05-07	2998000.0
20460	9808100150	2015-04-02	3345000.0
20535	1370800515	2014-10-30	2950000.0
21310	2154970020	2014-07-03	2351960.0
21530	8964800330	2015-04-07	3000000.0

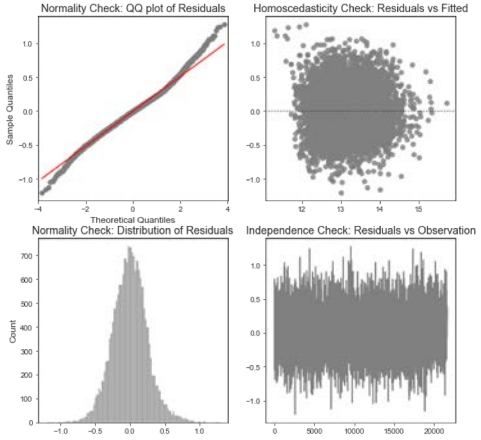
95 rows × 34 columns

4. Modelling

We try to take the ln of price and eliminate features.

1. Residual plots look better with price_ln and sqft_living_ln.
2. Eliminated features with higher VIF and low coefficients through backward selection.

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 price_ln   R-squared:                       0.760
Model:                            OLS   Adj. R-squared:                  0.760
Method:                 Least Squares   F-statistic:                     6052.
Date:                Mon, 29 Nov 2021   Prob (F-statistic):               0.00
Time:                        12:10:27   Log-Likelihood:                -1009.0
No. Observations:               17189   AIC:                             2038.
Df Residuals:                   17179   BIC:                             2116.
Df Model:                           9
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
Intercept       13.0476      0.002   6664.595      0.000      13.044      13.051
bathrooms        0.0520      0.003     15.207      0.000       0.045       0.059
sqft_living_ln   0.1535      0.004     43.306      0.000       0.147       0.160
floors           0.0293      0.002     11.972      0.000       0.025       0.034
waterfront       0.0329      0.002     15.396      0.000       0.029       0.037
view             0.0556      0.002     24.725      0.000       0.051       0.060
condition        0.0346      0.002     16.200      0.000       0.030       0.039
grade            0.2225      0.003     69.083      0.000       0.216       0.229
age              0.1089      0.003     40.859      0.000       0.104       0.114
lat              0.1872      0.002     91.637      0.000       0.183       0.191
==============================================================================
Omnibus:                      341.997   Durbin-Watson:                   1.970
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              602.502
Skew:                           0.161   Prob(JB):                    1.47e-131
Kurtosis:                       3.859   Cond. No.                         4.26
==============================================================================

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

MSE = 0.0658818186023037
```




```
***MULTICOLLINEARITY TESTS***
Variance Inflation Factor (VIF=1: not correlated, 1<VIF<5: moderately correlated, VIF>5: highly correlated)
VIF of bathrooms 3.0677171399823098
VIF of sqft_living_ln 3.3041856865308077
VIF of floors 1.5662291054492807
VIF of waterfront 1.2038823025702237
VIF of view 1.333755170248481
VIF of condition 1.188837921213283
VIF of grade 2.7306349559068366
VIF of age 1.840029963379432
VIF of lat 1.0907107595389294
```

OUTLIER TEST

	id	date	price
4390	2724089019	2014-05-23	527560.0
8606	2422059015	2014-06-06	533050.0
9563	3023039231	2014-07-14	650000.0
12551	1049010620	2014-05-13	90000.0
17363	4067600255	2014-05-22	398000.0

Regression equation

$$\ln(price) = b_1 bathrooms + b_2 \ln(sqft\_living) + b_3 floors + b_4 waterfront + b_5 view + b_6 condition + b_7 grade + b_8 lat + b_9 long$$

R^2_{adj}= 0.761

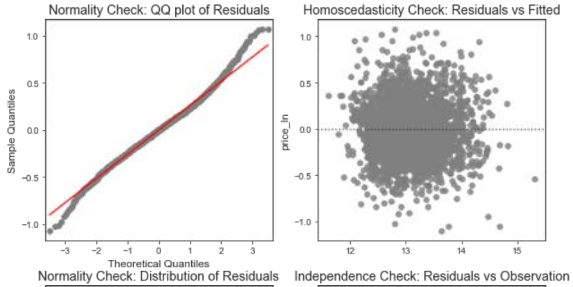
4. Modelling

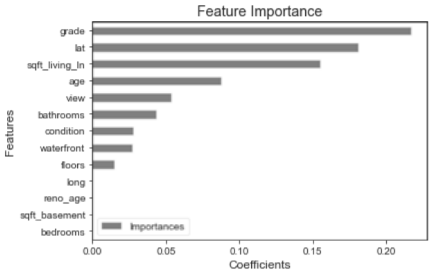
We try Lasso Regression to see if we can further reduce features.

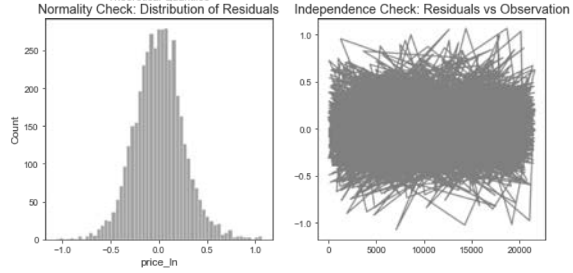
1. First, we removed sqft_above, sqft_lot, sqft_living15, sqft_lot15 and zipcode to reduce multicollinearity.
2. From the feature importance plot, some can be removed.
3. Error was also really large.

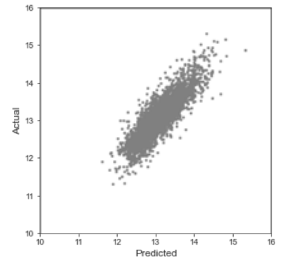
```
**********************************************************
METHOD: Lasso()
**********************************************************

Features: ['bedrooms', 'bathrooms', 'sqft_living_ln', 'floors', 'waterfront', 'view', 'condition', 'grade', 'sqft_ba
sement', 'age', 'reno_age', 'lat', 'long']
Standardized features: ['bedrooms', 'bathrooms', 'sqft_living_ln', 'floors', 'waterfront', 'view', 'condition', 'gra
de', 'sqft_basement', 'age', 'reno_age', 'lat', 'long']
Tuned Model Parameters: {'model__alpha': 0.01}

========== Metrics ==========

R²: 0.756
Adjusted R²: 0.759
RMSE: 0.2582751633014633
MAE: 0.1993955338146399
MAPE: 0.015259783336382695
```

Since we took ln(price), MAE ~MAPE= 20%
After transforming price back, RMSE ~ $190k!









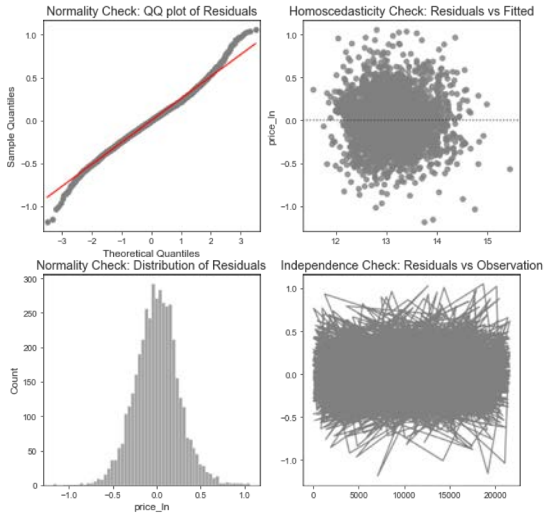
4. Modelling

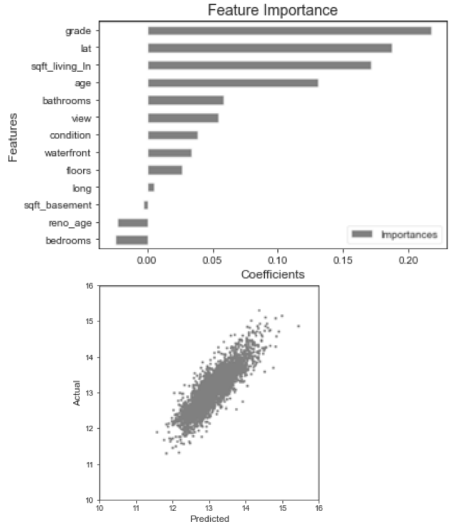
Maybe all the features are necessary in the model. So we try Ridge Regression.

Errors are still really large.

```
************************************************************
METHOD: Ridge()
************************************************************

Features: ['bedrooms', 'bathrooms', 'sqft_living_ln', 'floors', 'waterfront', 'view', 'condition', 'grade', 'sqft_ba
sement', 'age', 'reno_age', 'lat', 'long']
Standardized features: ['bedrooms', 'bathrooms', 'sqft_living_ln', 'floors', 'waterfront', 'view', 'condition', 'gra
de', 'sqft_basement', 'age', 'reno_age', 'lat', 'long']
Tuned Model Parameters: {'model__alpha': 0.9}

========== Metrics ==========

R²: 0.760
Adjusted R²: 0.763
RMSE: 0.25624780733880986
MAE: 0.19726834068182325
MAPE: 0.015095447349836146
```

Since we took ln(price), MAE ~MAPE= 20%
After transforming price back, RMSE ~ $190k!





4. Modelling

Accuracy is likely compromised by the skewness in price.

To improve accuracy, we try option 1 first to preserve as much data as possible.

Option 1: Try more models

Find the best way to split the dataset and build different models, then stack them back (e.g. model A for houses below grade 9, model B for houses grade 9 and above). Also try other models.

Option 2: Drop outliers

Drop outliers in price (either by business needs or outside of 1.5 times the interquartile range). This method may sacrifice some important data points.

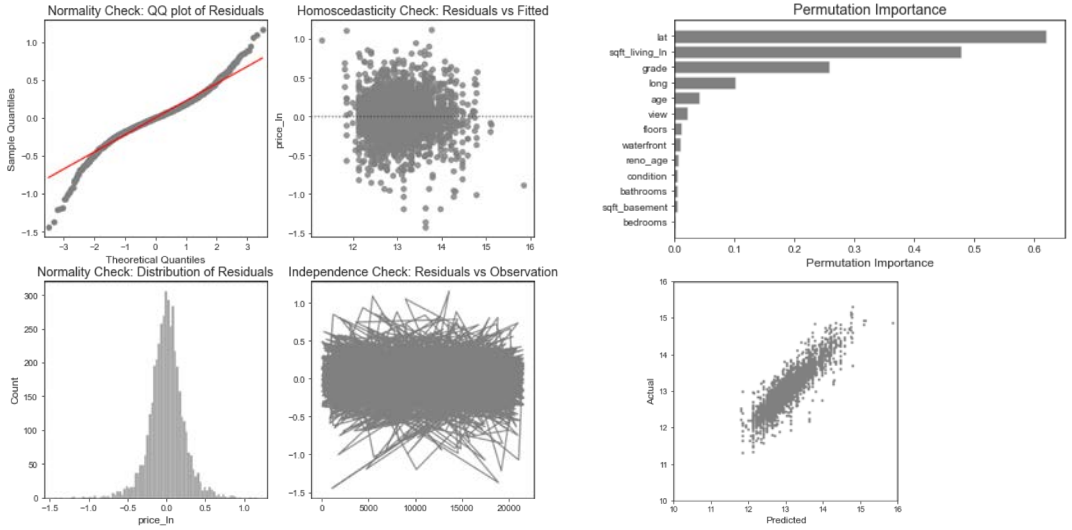
4. Modelling

For option 1, let's try a decision tree regressor to see if we can "split" the models.

Based on the decision tree, we can consider splitting by:

1. Grade 8 and below | 9 and above
2. Lat <47.53 | >47.53
3. Sqft_living > 4,000 | < 4,000 (upper outlier limit (1.5 times IQR) is 4,330 sqft)

```
************************************************************
METHOD: DecisionTreeRegressor()
************************************************************

Features: ['bedrooms', 'bathrooms', 'sqft_living_ln', 'floors', 'waterfront', 'view', 'condition', 'grade', 'sqft_ba
sement', 'age', 'reno_age', 'lat', 'long']
Standardized features: ['bedrooms', 'bathrooms', 'sqft_living_ln', 'floors', 'waterfront', 'view', 'condition', 'gra
de', 'sqft_basement', 'age', 'reno_age', 'lat', 'long']
Tuned Model Parameters: {'model__max_depth': 9, 'model__max_features': 'auto', 'model__splitter': 'best'}

========== Metrics ==========

R²: 0.814
Adjusted R²: 0.860
RMSE: 0.22574185631693866
MAE: 0.16327943560548674
MAPE: 0.012508637386423759
```



4. Modelling

Splitting by zipcode gave the lowest RMSE but errors were still large.

Different models also had different predictors. Workings in the Annex.

	Split model	Split model RMSE	Overall RMSE
Grade	Grade ≤ 8	129,295.93	192,360.99
	Grade > 8	346,16.87	
Lat	Lat ≤ 47.53	82,609.42	178,168.93
	Lat > 47.53	215,097.77	
Zipcode	Not 98039 | 98004	156,621.52	161,261.32
	Is 98039 | 98004	166,328.12	
Sqft_living	Sqft_living ≤ 4,000	144,002.17	176,476.90
	Sqft_living > 4,000	555,886.88	

4. Modelling

We tried other algorithms, and for each model, chose features through a systematic process

1. KNN Regression
2. SVR
3. XGBoost

1. **Keep only the best of correlated features**
   Since we have many derived features, we only keep the best. Dropping features, especially those high in feature importance, would have an impact on the feature ranks.
   e.g. Choosing either sqft_living or sqft_lot.

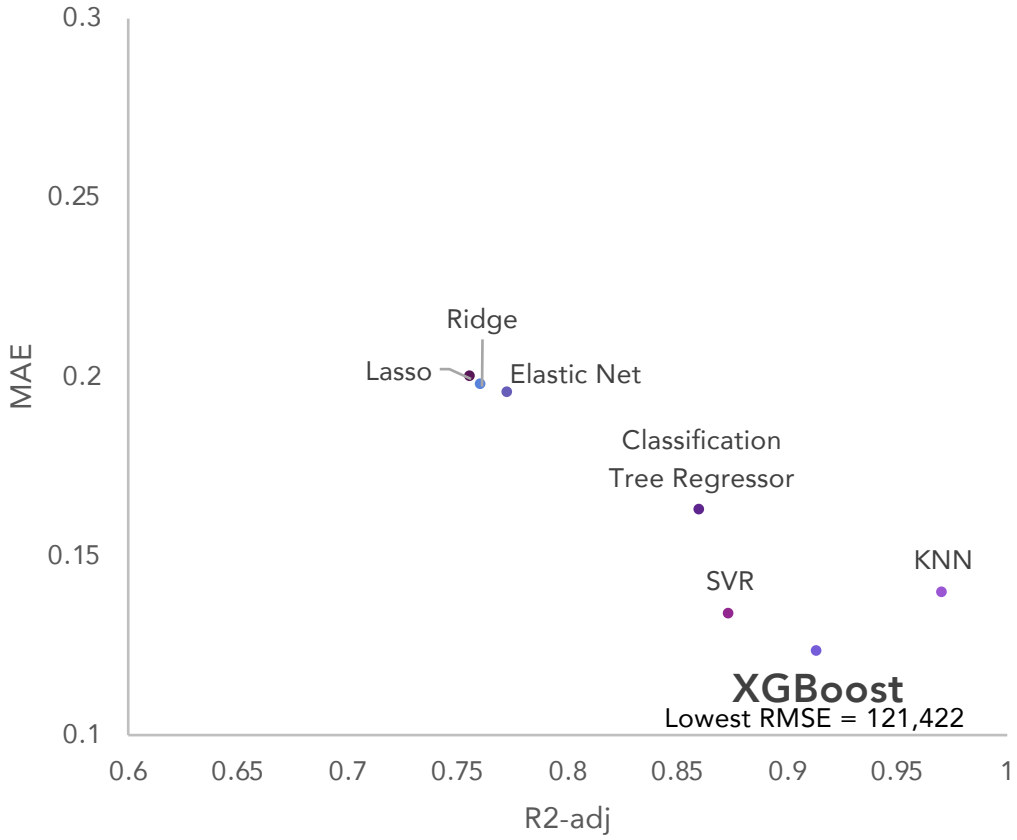
2. **Try replacing numerical features with categorical or binned features**
   Sometimes the features have too much/little noise, and we trial-and-error to see which type of features work best for the model.
   e.g. Exploring if lat or lat_1km is better

3. **Drop features that have very low importance scores / coefficients**
   As the final step, we can safely drop insignificant features.

4. Modelling

XGBoost had the lowest error

RMSE = USD 121,422, even after tuning hyperparameters. Can this be improved?

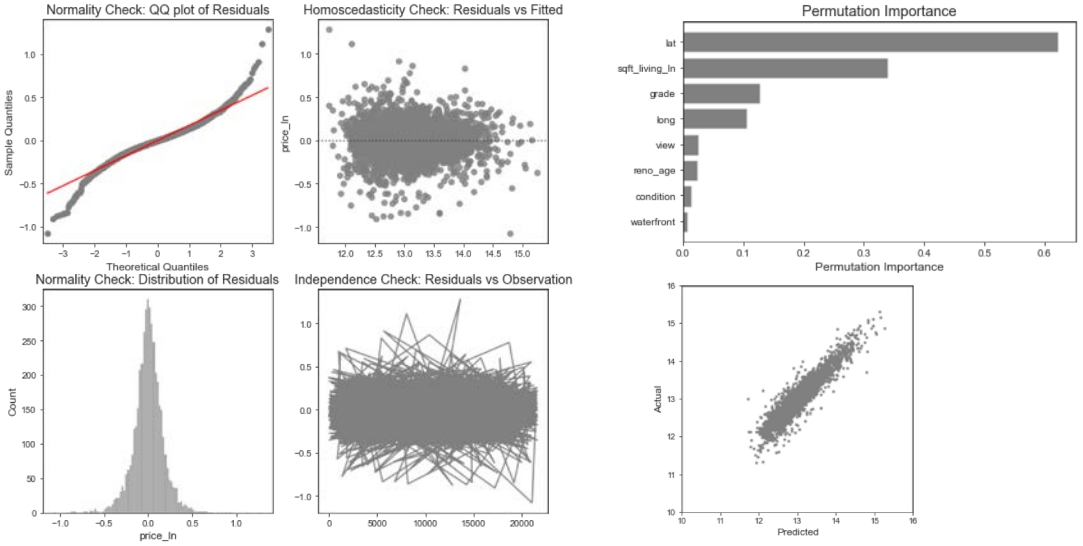


4. Modelling

Let's take a look at XGBoost's residuals

Residuals are not normally distributed, likely because of the outliers in price.

```
************************************************************
METHOD: XGBRegressor(n_estimators=1000, objective='reg:squarederror', seed=21)
************************************************************

Features: ['sqft_living_ln', 'condition', 'waterfront', 'view', 'grade', 'reno_age', 'lat', 'long']
Standardized features: ['sqft_living_ln', 'condition', 'waterfront', 'view', 'grade', 'reno_age', 'lat', 'long']
Tuned Model Parameters: {'model__alpha': 0.9, 'model__lambda': 0.9, 'model__lambda_bias': 0.9}


========== Metrics ==========

R²: 0.888
Adjusted R²: 0.912
RMSE: 0.1749099570933342
MAE: 0.1247368662241634
MAPE: 0.009588423183071739
```



4. Modelling

Can we improve the XGB model?

RMSE to beat = 121,422

We explored

1. **Square root transformation on price**
   RMSE = 123,730

2. **Splitting by postal code**
   RMSE = 119,760

3. **Drop outliers (price > 3M)**
   RMSE = 110,920, dropped 50 records

4. **Drop outliers (price > 2M)**
   RMSE = 100,879, dropped 205 records

5. **Split by postal code and remove outliers**
   RMSE = 89,842, dropped 237 records

4. Modelling

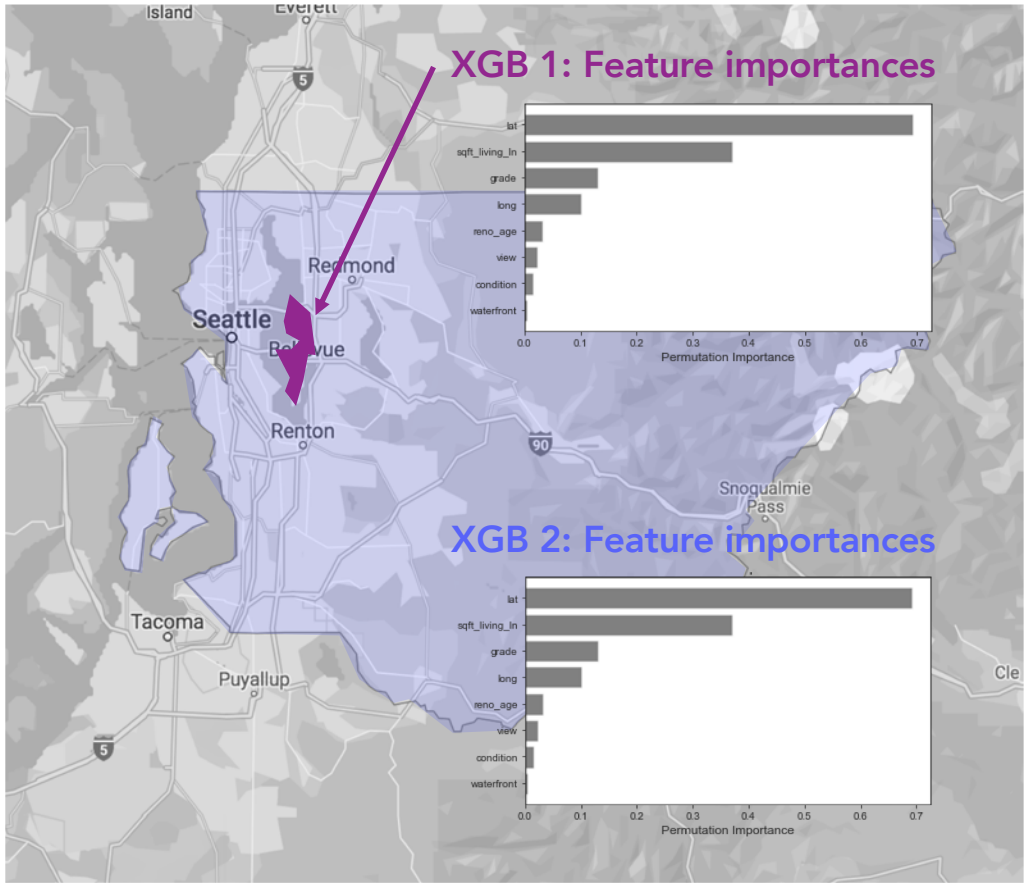
So... what model(s) should we choose?

RMSE is the most appropriate metric in this exercise. So we try our best to minimize it.

Hence we choose two XGB models for different regions, using the same features but different weights.

The residuals for XGB 1 (in Annex) show some deviance from normality, but not too serious.

We also tried other algorithms (e.g. KNN) for model 1 but XGB was by far the best.



5. Summary

In conclusion…

1. Objective: Predict housing prices in King County. For this task, we try to minimize RMSE in the predictions. We tried transformations, different algorithms, splitting models etc.

2. Data was from 2014 to 2015. It's too short to do a time series to predict prices now in 2021, but we can do a simple extrapolation based on annual growth rates.

3. We try to minimize and "de-trend" the errors as much as possible, but sometimes they still exist. There's a time to move on 😅…

An excerpt from George Box's 1976 paper (the famous "all models are wrong")

Worrying Selectively

Since all models are wrong the scientist must be alert to what is importantly wrong. It is inappropriate to be concerned about mice when there are tigers abroad.

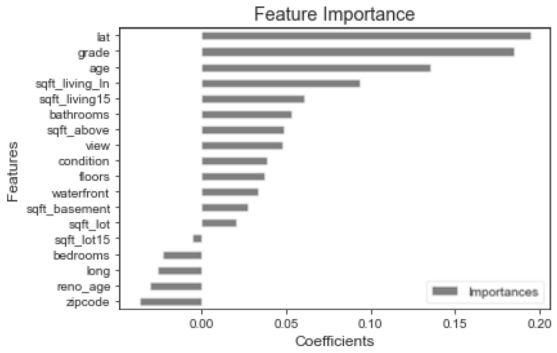
Annex

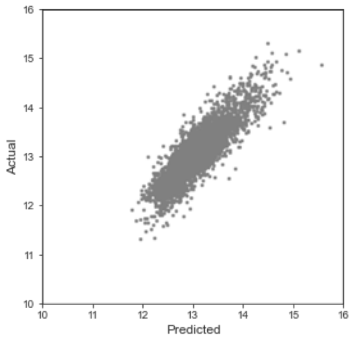
"All models are wrong, but some are useful." – George Box

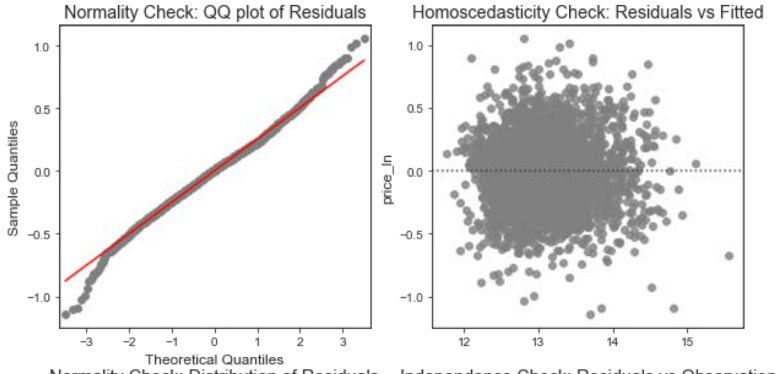
Elastic Net Regression

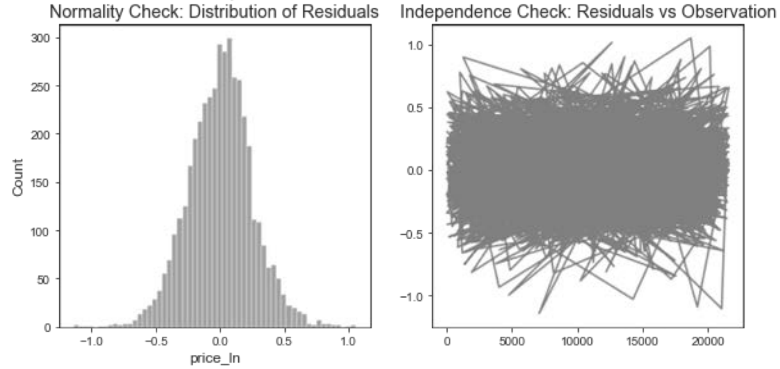
Full Model

```
Rows before dropping: 21603
Rows after dropping: 21603
************************************************************
METHOD: ElasticNet()
************************************************************

Features: ['bedrooms', 'bathrooms', 'sqft_living_ln', 'sqft_lot', 'floors', 'waterfront', 'view', 'condition', 'grad
e', 'sqft_above', 'sqft_basement', 'zipcode', 'sqft_living15', 'sqft_lot15', 'age', 'reno_age', 'lat', 'long']
Standardized features: ['bedrooms', 'bathrooms', 'sqft_living_ln', 'sqft_lot', 'floors', 'waterfront', 'view', 'cond
ition', 'grade', 'sqft_above', 'sqft_basement', 'zipcode', 'sqft_living15', 'sqft_lot15', 'age', 'reno_age', 'lat',
'long']
Tuned Model Parameters: {'model__alpha': 0.001, 'model__l1_ratio': 0.001}

========== Metrics ==========

R²: 0.769
Adjusted R²: 0.773
RMSE: 0.25152147427038607
MAE: 0.19515736144788937
MAPE: 0.014936049532329704
```







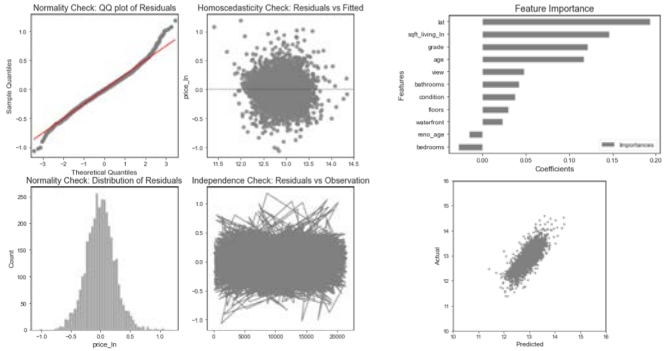


Elastic Net Regression

	RMSE
Total	192,360.99
Grade 8 and below	129,295.93
Grade 9 and above	346,16.87

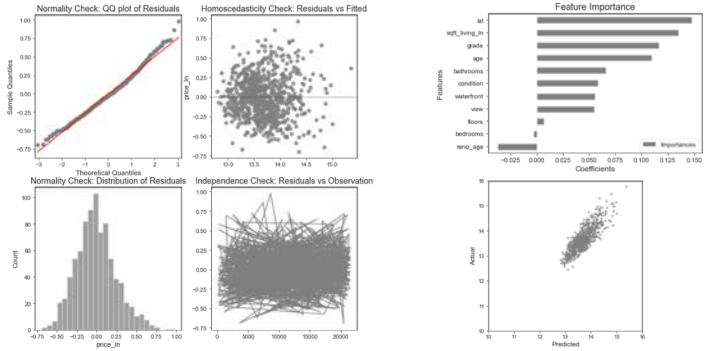
Grade 8 and below

```
Rows before dropping: 21603
Rows after dropping: 17353
************************************************************
METHOD: ElasticNet()
************************************************************

Features: ['bedrooms', 'bathrooms', 'sqft_living_ln', 'floors', 'waterfront', 'view', 'condition', 'grade', 'age', '
reno_age', 'lat']
Standardized features: ['bedrooms', 'bathrooms', 'sqft_living_ln', 'floors', 'waterfront', 'view', 'condition', 'gra
de', 'age', 'reno_age', 'lat']
Tuned Model Parameters: {'model__alpha': 0.001, 'model__l1_ratio': 0.001}

========== Metrics ==========

R²: 0.650
Adjusted R²: 0.645
RMSE: 0.25076349331557285
MAE: 0.19279136769400992
MAPE: 0.014932370842858107
```



Grade 9 and above

```
Rows before dropping: 21603
Rows after dropping: 4250
************************************************************
METHOD: ElasticNet()
************************************************************

Features: ['bedrooms', 'bathrooms', 'sqft_living_ln', 'floors', 'waterfront', 'view', 'condition', 'grade', 'age', '
reno_age', 'lat']
Standardized features: ['bedrooms', 'bathrooms', 'sqft_living_ln', 'floors', 'waterfront', 'view', 'condition', 'gra
de', 'age', 'reno_age', 'lat']
Tuned Model Parameters: {'model__alpha': 0.01, 'model__l1_ratio': 0.001}

========== Metrics ==========

R²: 0.708
Adjusted R²: 0.669
RMSE: 0.2551830335728748
MAE: 0.1992672036111299
MAPE: 0.014523419049953214
```

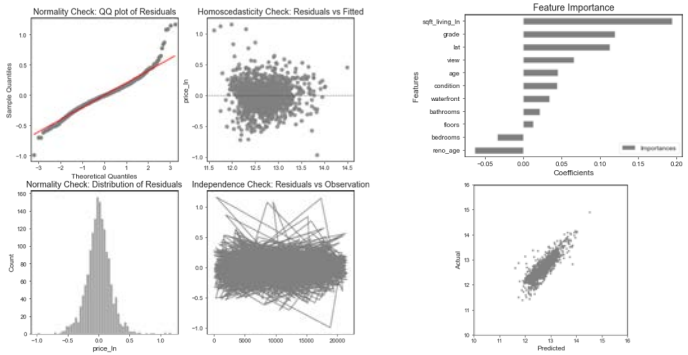


Elastic Net Regression

	RMSE
Total	178,168.93
Lat ≤ 47.53	82,609.42
Lat > 47.53	215,097.77

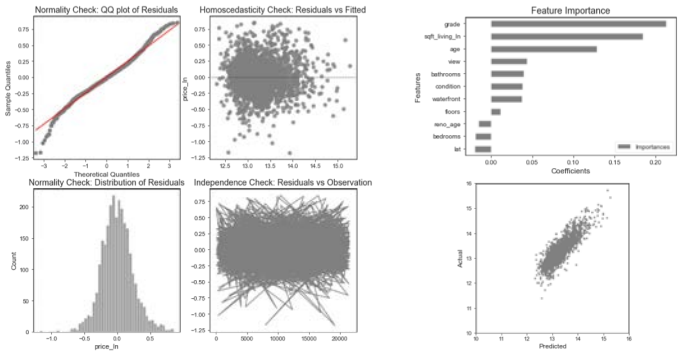
Lat ≤47.53

```
************************************************************
METHOD: ElasticNet()
************************************************************

Features: ['bedrooms', 'bathrooms', 'sqft_living_ln', 'floors', 'waterfront', 'view', 'condition', 'grade', 'age', '
reno_age', 'lat']
Standardized features: ['bedrooms', 'bathrooms', 'sqft_living_ln', 'floors', 'waterfront', 'view', 'condition', 'gra
de', 'age', 'reno_age', 'lat']
Tuned Model Parameters: {'model__alpha': 0.001, 'model__l1_ratio': 0.01}

========== Metrics ==========

R²: 0.741
Adjusted R²: 0.722
RMSE: 0.20197064205195794
MAE: 0.14778047873438055
MAPE: 0.01167668823955843
```



Lat >47.53

```
Rows before dropping: 21603
Rows after dropping: 13648
************************************************************
METHOD: ElasticNet()
************************************************************

Features: ['bedrooms', 'bathrooms', 'sqft_living_ln', 'floors', 'waterfront', 'view', 'condition', 'grade', 'age', '
reno_age', 'lat']
Standardized features: ['bedrooms', 'bathrooms', 'sqft_living_ln', 'floors', 'waterfront', 'view', 'condition', 'gra
de', 'age', 'reno_age', 'lat']
Tuned Model Parameters: {'model__alpha': 0.001, 'model__l1_ratio': 0.001}

========== Metrics ==========

R²: 0.729
Adjusted R²: 0.730
RMSE: 0.2438961156080185
MAE: 0.18611634954974907
MAPE: 0.014009398871943033
```

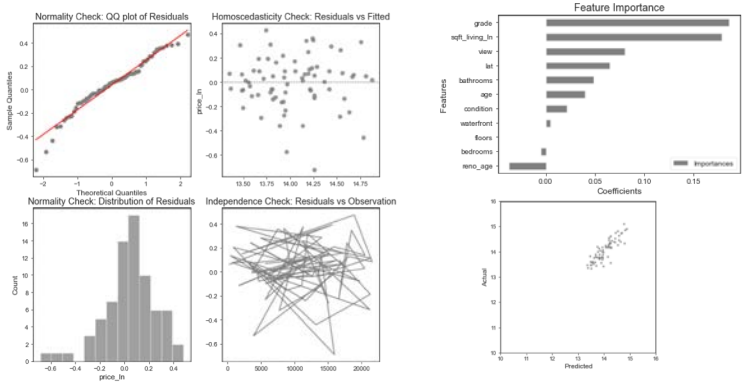


Elastic Net Regression

	RMSE
Total	161,261.32
Not 98039 | 98004	156,621.52
Is 98039 | 98004	166,328.12

Zipcode = 98039 | 98004

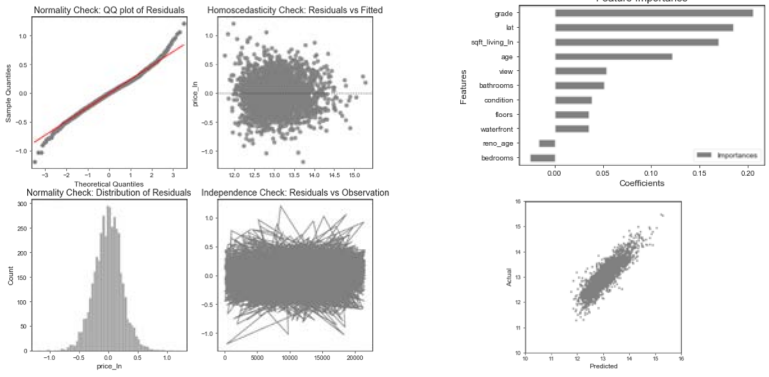
```
Rows before dropping: 21603
Rows after dropping: 365
************************************************************
METHOD: ElasticNet()
************************************************************

Features: ['bedrooms', 'bathrooms', 'sqft_living_ln', 'floors', 'waterfront', 'view', 'condition', 'grade', 'age', '
reno_age', 'lat']
Standardized features: ['bedrooms', 'bathrooms', 'sqft_living_ln', 'floors', 'waterfront', 'view', 'condition', 'gra
de', 'age', 'reno_age', 'lat']
Tuned Model Parameters: {'model__alpha': 0.01, 'model__l1_ratio': 0.1}

========== Metrics ==========

R²: 0.742
Adjusted R²: 0.796
RMSE: 0.2166561505227118
MAE: 0.16451738663604237
MAPE: 0.011661698081277753
```



Dropped 1 outlier where price > 6,000,000

Zipcode != 98039 | 98004

```
Rows before dropping: 21603
Rows after dropping: 21235
************************************************************
METHOD: ElasticNet()
************************************************************

Features: ['bedrooms', 'bathrooms', 'sqft_living_ln', 'floors', 'waterfront', 'view', 'condition', 'grade', 'age', '
reno_age', 'lat']
Standardized features: ['bedrooms', 'bathrooms', 'sqft_living_ln', 'floors', 'waterfront', 'view', 'condition', 'gra
de', 'age', 'reno_age', 'lat']
Tuned Model Parameters: {'model__alpha': 0.001, 'model__l1_ratio': 0.001}

========== Metrics ==========

R²: 0.770
Adjusted R²: 0.767
RMSE: 0.24219618320977854
MAE: 0.1875514073651856
MAPE: 0.014415390113407677
```

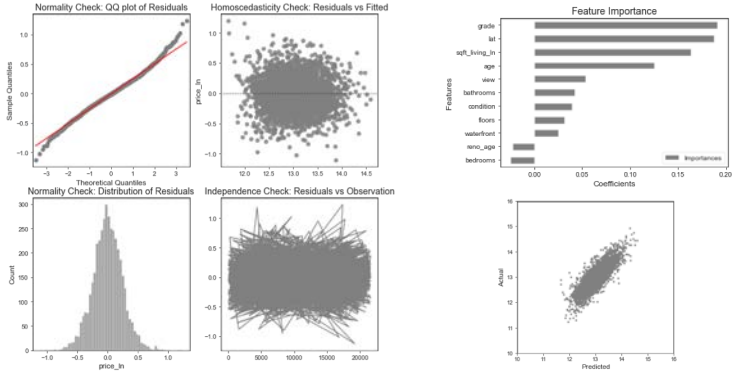


Elastic Net Regression

	RMSE
Total	176,476.90
Sqft_living ≤ 4000	144,002.17
Sqft_living > 4000	555,886.88

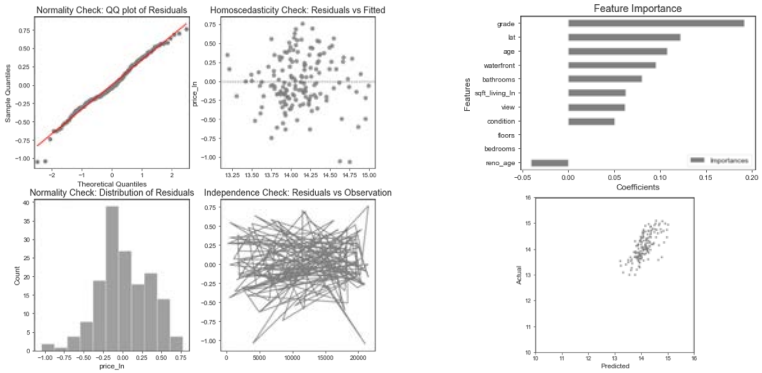
Sqft_living ≤ 4000

```
Rows before dropping: 21603
Rows after dropping: 20821
************************************************************
METHOD: ElasticNet()
************************************************************

Features: ['bedrooms', 'bathrooms', 'sqft_living_ln', 'floors', 'waterfront', 'view', 'condition', 'grade', 'age', '
reno_age', 'lat']
Standardized features: ['bedrooms', 'bathrooms', 'sqft_living_ln', 'floors', 'waterfront', 'view', 'condition', 'gra
de', 'age', 'reno_age', 'lat']
Tuned Model Parameters: {'model__alpha': 0.001, 'model__l1_ratio': 0.001}

========== Metrics ==========

R²: 0.717
Adjusted R²: 0.726
RMSE: 0.252202170690736l4
MAE: 0.194619945960484l5
MAPE: 0.0149847988178494
```



Sqft_living > 4000

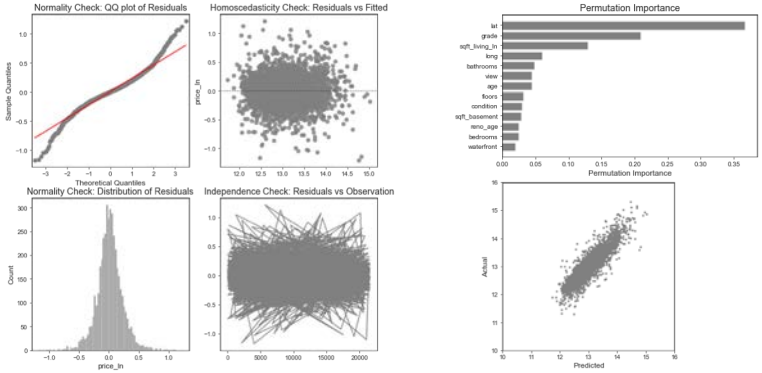
```
Rows before dropping: 21603
Rows after dropping: 782
************************************************************
METHOD: ElasticNet()
************************************************************

Features: ['bedrooms', 'bathrooms', 'sqft_living_ln', 'floors', 'waterfront', 'view', 'condition', 'grade', 'age', '
reno_age', 'lat']
Standardized features: ['bedrooms', 'bathrooms', 'sqft_living_ln', 'floors', 'waterfront', 'view', 'condition', 'gra
de', 'age', 'reno_age', 'lat']
Tuned Model Parameters: {'model__alpha': 0.01, 'model__l1_ratio': 0.5}

========== Metrics ==========

R²: 0.510
Adjusted R²: 0.569
RMSE: 0.33405954565621426
MAE: 0.2664491675854447
MAPE: 0.01885332807452369
```



K-nearest neighbours

Full model

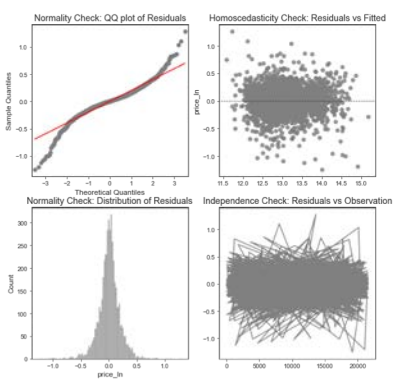
```
Rows before dropping: 21603
Rows after dropping: 21603
************************************************************
METHOD: KNeighborsRegressor()
************************************************************

Features: ['bedrooms', 'bathrooms', 'sqft_living_ln', 'floors', 'waterfront', 'view', 'condition', 'grade', 'sqft_ba
sement', 'age', 'reno_age', 'lat', 'long']
Standardized features: ['bedrooms', 'bathrooms', 'sqft_living_ln', 'floors', 'waterfront', 'view', 'condition', 'gra
de', 'sqft_basement', 'age', 'reno_age', 'lat', 'long']
Tuned Model Parameters: {'model__n_neighbors': 9}

========== Metrics ==========

R²: 0.810
Adjusted R²: 0.850
RMSE: 0.228212730195106
MAE: 0.16463478259175648
MAPE: 0.012585596103747886
```

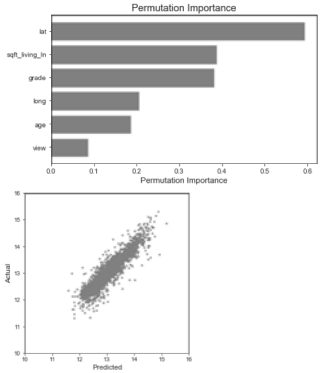


Best model

```
************************************************************
METHOD: KNeighborsRegressor()
************************************************************

Features: ['sqft_living_ln', 'view', 'grade', 'age', 'lat', 'long']
Standardized features: ['sqft_living_ln', 'view', 'grade', 'age', 'lat', 'long']
Tuned Model Parameters: {'model__n_neighbors': 11, 'model__weights': 'distance'}

========== Metrics ==========

R²: 0.855
Adjusted R²: 0.971
RMSE: 0.19927811207704046
MAE: 0.13915909033071636
MAPE: 0.010679065143844807
```

RMSE
151,726.77



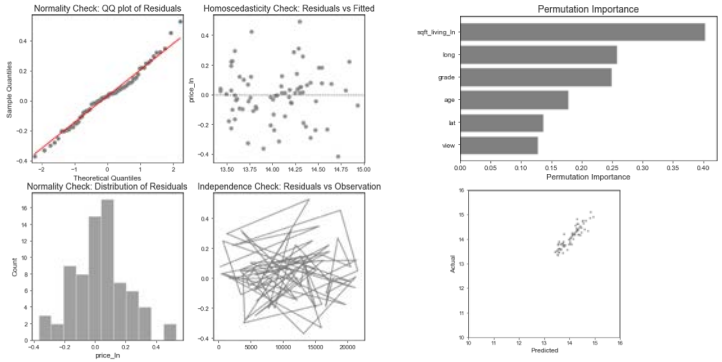


K-nearest neighbours

	RMSE
Total	152,391.40
Not 98039 | 98004	148,704.97
Is 98039 | 98004	157,922.57

Zipcode = 98039 | 98004

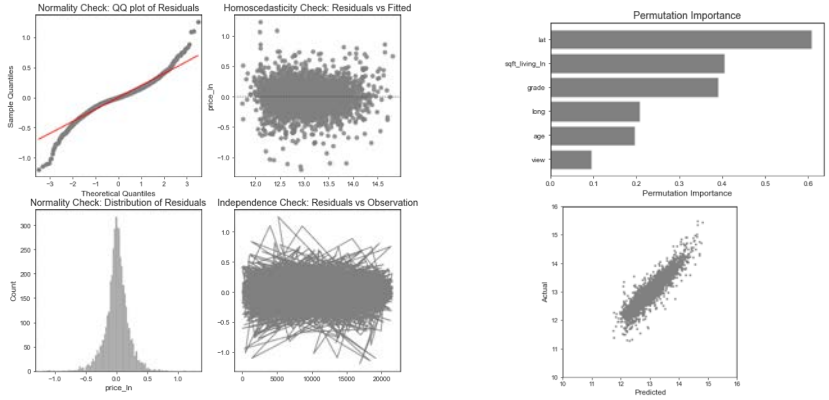
```
Rows before dropping: 21603
Rows after dropping: 365
************************************************************
METHOD: KNeighborsRegressor()
************************************************************

Features: ['sqft_living_ln', 'view', 'grade', 'age', 'lat', 'long']
Standardized features: ['sqft_living_ln', 'view', 'grade', 'age', 'lat', 'long']
Tuned Model Parameters: {'model__n_neighbors': 9, 'model__weights': 'distance'}

========== Metrics ==========

R²: 0.826
Adjusted R²: 0.970
RMSE: 0.17803833783627915
MAE: 0.1359044424050284
MAPE: 0.009608335125701402
```



Dropped 1 outlier where price > 6,000,000

Zipcode != 98039 | 98004

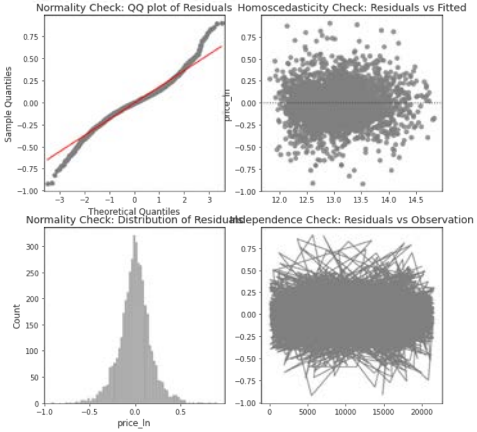
```
Rows before dropping: 21603
Rows after dropping: 21236
************************************************************
METHOD: KNeighborsRegressor()
************************************************************

Features: ['sqft_living_ln', 'view', 'grade', 'age', 'lat', 'long']
Standardized features: ['sqft_living_ln', 'view', 'grade', 'age', 'lat', 'long']
Tuned Model Parameters: {'model__n_neighbors': 9, 'model__weights': 'distance'}

========== Metrics ==========

R²: 0.848
Adjusted R²: 0.969
RMSE: 0.19882848586018886
MAE: 0.1391018239944003
MAPE: 0.010710084220512235
```

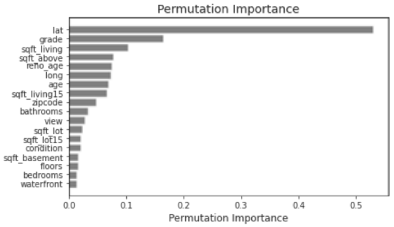


SVR

Full model

```
************************************************************
METHOD: SVR()
************************************************************

Features: ['bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot', 'floors', 'waterfront', 'view', 'condition', 'grade', 'sqft_above', 'sqft_basement', 'zipcode', 'sqft_living15', 'sqft_lot15', 'age', 'reno_age', 'lat', 'long']
Standardized features: ['bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot', 'floors', 'waterfront', 'view', 'condition', 'grade', 'sqft_above', 'sqft_basement', 'zipcode', 'sqft_living15', 'sqft_lot15', 'age', 'reno_age', 'lat', 'long']
Tuned Model Parameters: {'model__kernel': 'rbf'}


========== Metrics ==========

R²: 0.874
Adjusted R²: 0.892
RMSE: 0.18366525417660445
MAE: 0.13364031688026282
MAPE: 0.01026034709123131
```

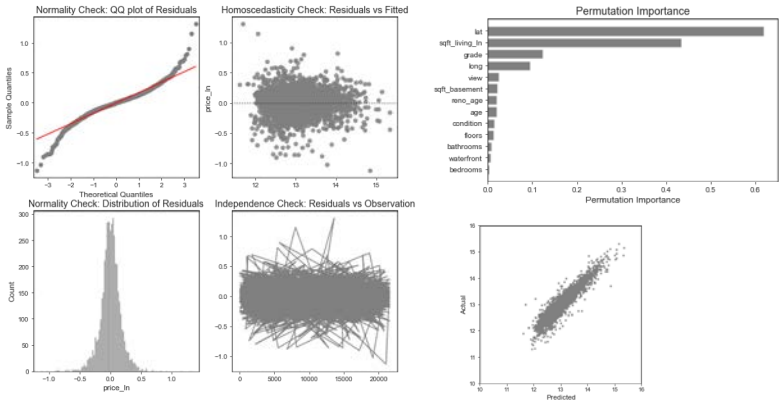




XGBoost

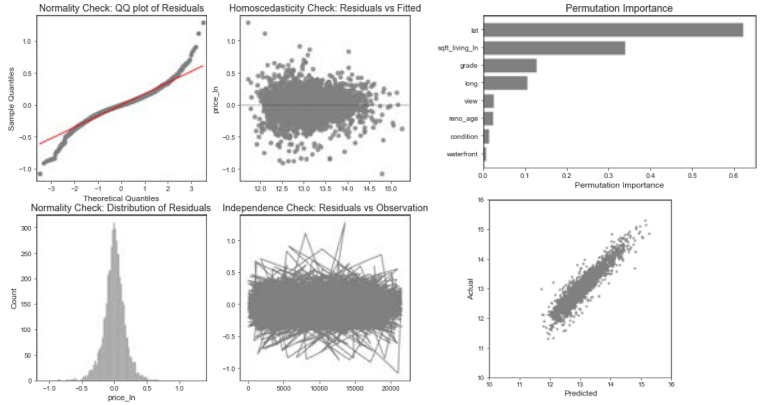
Full model on ln(price)

```
************************************************************
METHOD: XGBRegressor(n_estimators=1000, objective='reg:squarederror', seed=21)
************************************************************

Features: ['bedrooms', 'bathrooms', 'sqft_living_ln', 'floors', 'waterfront', 'view', 'condition', 'grade', 'sqft_ba
sement', 'age', 'reno_age', 'lat', 'long']
Standardized features: ['bedrooms', 'bathrooms', 'sqft_living_ln', 'floors', 'waterfront', 'view', 'condition', 'gra
de', 'sqft_basement', 'age', 'reno_age', 'lat', 'long']
Tuned Model Parameters: {'model__alpha': 0.9, 'model__lambda': 0.9, 'model__lambda_bias': 0.5}

========== Metrics ==========

R²: 0.890
Adjusted R²: 0.917
RMSE: 0.1733649317320754
MAE: 0.123780332660862 12
MAPE: 0.009518507206657548
```



Best model on ln(price)

```
************************************************************
METHOD: XGBRegressor(n_estimators=1000, objective='reg:squarederror', seed=21)
************************************************************

Features: ['sqft_living_ln', 'condition', 'waterfront', 'view', 'grade', 'reno_age', 'lat', 'long']
Standardized features: ['sqft_living_ln', 'condition', 'waterfront', 'view', 'grade', 'reno_age', 'lat', 'long']
Tuned Model Parameters: {'model__alpha': 0.9, 'model__lambda': 0.9, 'model__lambda_bias': 0.9}

========== Metrics ==========

R²: 0.888
Adjusted R²: 0.912
RMSE: 0.1749099570933342
MAE: 0.1247368662241634
MAPE: 0.009588423183071739
```

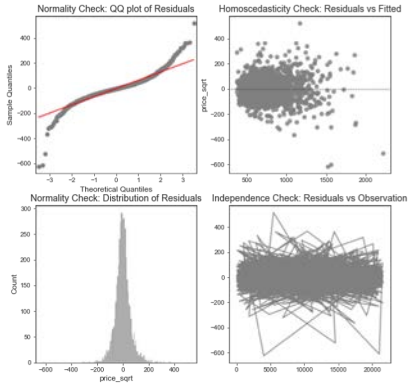


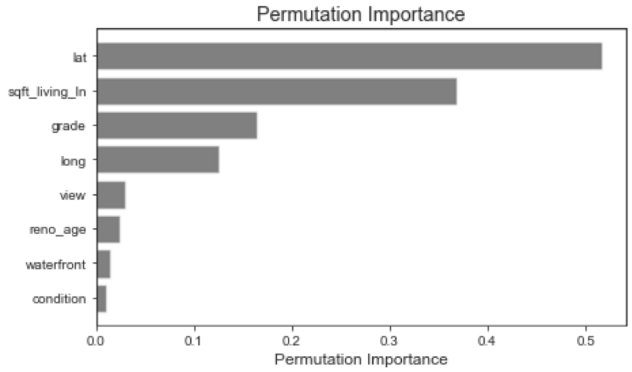
XGBoost

Best model on $\sqrt{price}$

```
************************************************************
METHOD: XGBRegressor(n_estimators=1000, objective='reg:squarederror', seed=21)
************************************************************

Features: ['sqft_living_ln', 'condition', 'waterfront', 'view', 'grade', 'reno_age', 'lat', 'long']
Standardized features: ['sqft_living_ln', 'condition', 'waterfront', 'view', 'grade', 'reno_age', 'lat', 'long']
Tuned Model Parameters: {'model__alpha': 0.9, 'model__lambda': 0.9, 'model__lambda_bias': 0.9}


========== Metrics ==========

R²: 0.894
Adjusted R²: 0.923
RMSE: 65.20722607758667
MAE: 44.51380952361337
MAPE: 0.06345690070060224
```

RMSE
123,730.53



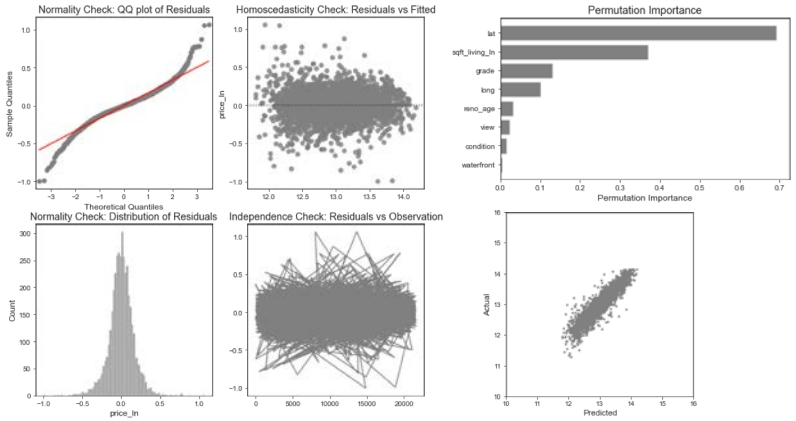


XGBoost

	XGB RMSE	KNN RMSE
Total	89,842.82	
Model 1	83,608.09	
Model 2	208,937.19	228,007

Model 1
(zipcode != 98039 | 98004 | 98040) & (price <1.4M)

```
************************************************************
METHOD: XGBRegressor(n_estimators=1000, objective='reg:squarederror', seed=21)
************************************************************

Features: ['sqft_living_ln', 'condition', 'waterfront', 'view', 'grade', 'reno_age', 'lat', 'long']
Standardized features: ['sqft_living_ln', 'condition', 'waterfront', 'view', 'grade', 'reno_age', 'lat', 'long']
Tuned Model Parameters: {'model__alpha': 0.9, 'model__lambda': 0.9, 'model__lambda_bias': 0.9}

========== Metrics ==========

R²: 0.867
Adjusted R²: 0.893
RMSE: 0.1681637983030543
MAE: 0.1206123808186605
MAPE: 0.009338227787616674
```



Model 2
(zipcode == 98039 | 98004 | 98040) & (price > 0.5M & price <3M)

```
************************************************************
METHOD: XGBRegressor(n_estimators=1000, objective='reg:squarederror', seed=21)
************************************************************

Features: ['sqft_living_ln', 'condition', 'waterfront', 'view', 'grade', 'reno_age', 'lat', 'long']
Standardized features: ['sqft_living_ln', 'condition', 'waterfront', 'view', 'grade', 'reno_age', 'lat', 'long']
Tuned Model Parameters: {'model__alpha': 0.9, 'model__lambda': 0.9, 'model__lambda_bias': 0.9}

========== Metrics ==========

R²: 0.787
Adjusted R²: 0.962
RMSE: 0.16510373302554965
MAE: 0.1286206934695774
MAPE: 0.009218106648316623
```