Capstone Project 2: House Price Prediction Milestone Report

Problem Statement

The question in this data science projection I'm trying to answer is:

How much is the residential homes in Ames, Iowa given that we know all different aspects of the homes such as the proximity to the public transportation, number of bedrooms, the height of the basement or types of neighborhood the home is located.

Accurate prediction on the house price based on attributes and quality of the house would be useful for the house buyers to set a more data-driven budget for themselves when they are searching for homes. On the other hand, this would be a great tool for sellers to set a more reasonable price when they are selling their properties.

To solve the problem, I have chosen to use regression models to leverage house attributes for predicting the sale price of the houses in the dataset. The features with great contribution/importance to the house price will shed a light on what really matters when it comes to the pricing.

Description of the dataset

The Ames housing datasets is provided on Kaggle.

Here the link where you can download the dataset:

https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data

The data has 80 fields, in which there are sale price (which I'm trying to predict) and 79 different kinds of attributes related to the residential homes.

To learn more about all the features, I have created a data dictionary, which you will find in the Data_MetaData_Documentation.xlsx within the documentation folder. Within in the data metadata dictionary, I have categorized all the features into four different buckets: Exterior, Interior, Location and Other.

- Exterior features: these are features that are related to things outside of the living area.
  For example: Masonry veneer type, Exterior material quality, Type of foundation, Interior finish of the garage
- Interior features: these are features that are for things within the living area.
  For example: Height of the basement, Heating quality and condition, Central air conditioning, Kitchen quality
- Location: these are features that describe the location of the property.
  For example: general zoning classification, slope of property, physical locations within Ames city limits.
- Other features: for example, month sold, year sold, type of sale

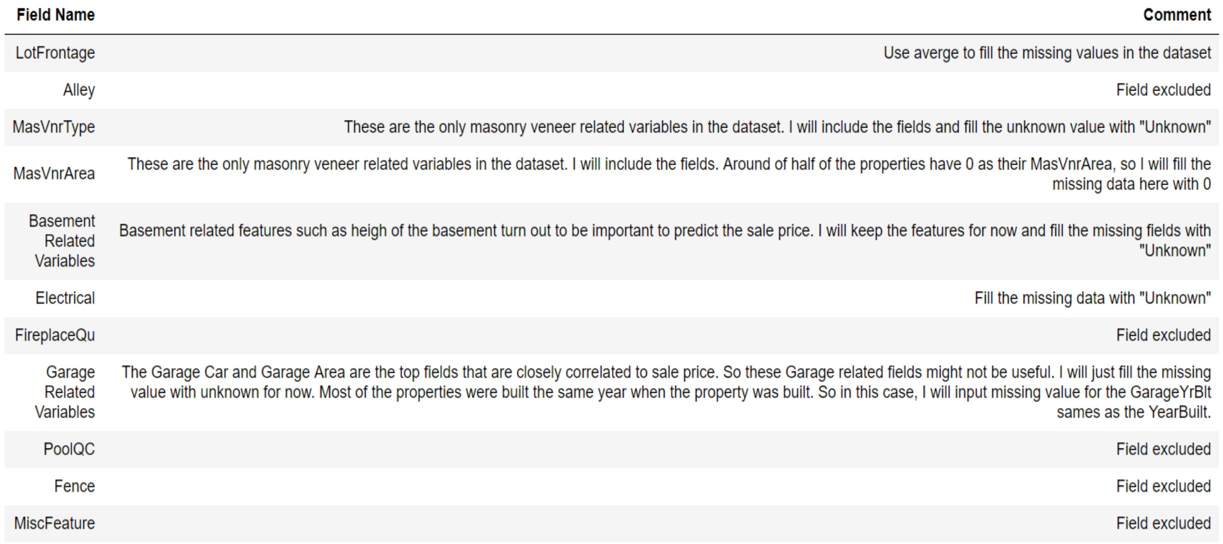
Data Cleaning & Feature engineering Process

Step 1: Imputing missing values:

The whole dataset has 1460 rows (1460 properties). There are several fields with missing values.

For the fields with high proportion of missing values, I just deleted the features given that I don't have much training data. These fields are not going to very useful for forecasting the price. For the fields that I would like to keep, I'm just using the mean to impute the missing data points.

Here's how the fields with missing values have been handled:



Field Name	Comment
LotFrontage	Use averge to fill the missing values in the dataset
Alley	Field excluded
MasVnrType	These are the only masonry veneer related variables in the dataset. I will include the fields and fill the unknown value with "Unknown"
MasVnrArea	These are the only masonry veneer related variables in the dataset. I will include the fields. Around of half of the properties have 0 as their MasVnrArea, so I will fill the missing data here with 0
Basement Related Variables	Basement related features such as heigh of the basement turn out to be important to predict the sale price. I will keep the features for now and fill the missing fields with "Unknown"
Electrical	Fill the missing data with "Unknown"
FireplaceQu	Field excluded
Garage Related Variables	The Garage Car and Garage Area are the top fields that are closely correlated to sale price. So these Garage related fields might not be useful. I will just fill the missing value with unknown for now. Most of the properties were built the same year when the property was built. So in this case, I will input missing value for the GarageYrBlt sames as the YearBuilt.
PoolQC	Field excluded
Fence	Field excluded
MiscFeature	Field excluded

Step 2: Outlier Detection

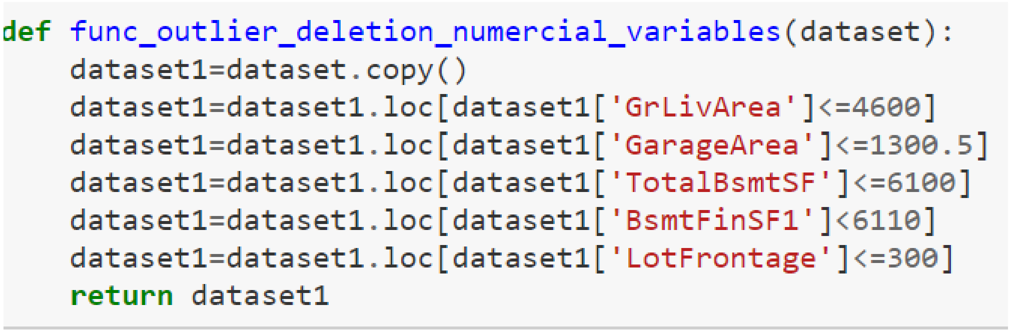
For the numerical values, I used interquantile range to detect the outliers within the data.

Outlier upper bond: 75% quantile + 3 * (75% quantile – 25% quantile)

Outlier lower bond: 25% quantile – 3 * (75% quantile – 25% quantile)

Outliers that fall outside of the outlier upper bound and lower bound will be delted.

The fields below have detected outliers. The strategy for dealing with these outliers is simply deleting the whole records with outliers since there's not much data to be removed in this case.



```
def func_outlier_deletion_numercial_variables(dataset):
    dataset1=dataset.copy()
    dataset1=dataset1.loc[dataset1['GrLivArea']<=4600]
    dataset1=dataset1.loc[dataset1['GarageArea']<=1300.5]
    dataset1=dataset1.loc[dataset1['TotalBsmtSF']<=6100]
    dataset1=dataset1.loc[dataset1['BsmtFinSF1']<6110]
    dataset1=dataset1.loc[dataset1['LotFrontage']<=300]
    return dataset1
```

Step 3: Encoding the categorical variables

For the categorical variables, I used one-hot encoding to convert these variables into dummy variables since Sklearn cannot take in the categorical variables directly for modeling.

Step 4: Adding one feature

I have also added one more feature total square feet.

Total area in square feet = first floor area in square feet + Second floor area in square feet + Basement area in square feet

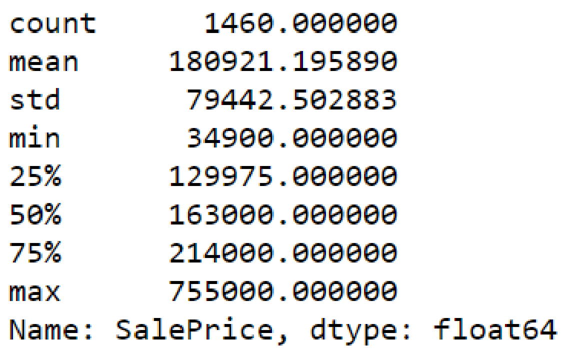
Step 5: Data Transformation

For the current model I'm running, I have used log transformation for variables with a non-normal distribution and also used the standard scaler (mean and standard deviation) to standardize all the variables.

Exploratory Data Analysis

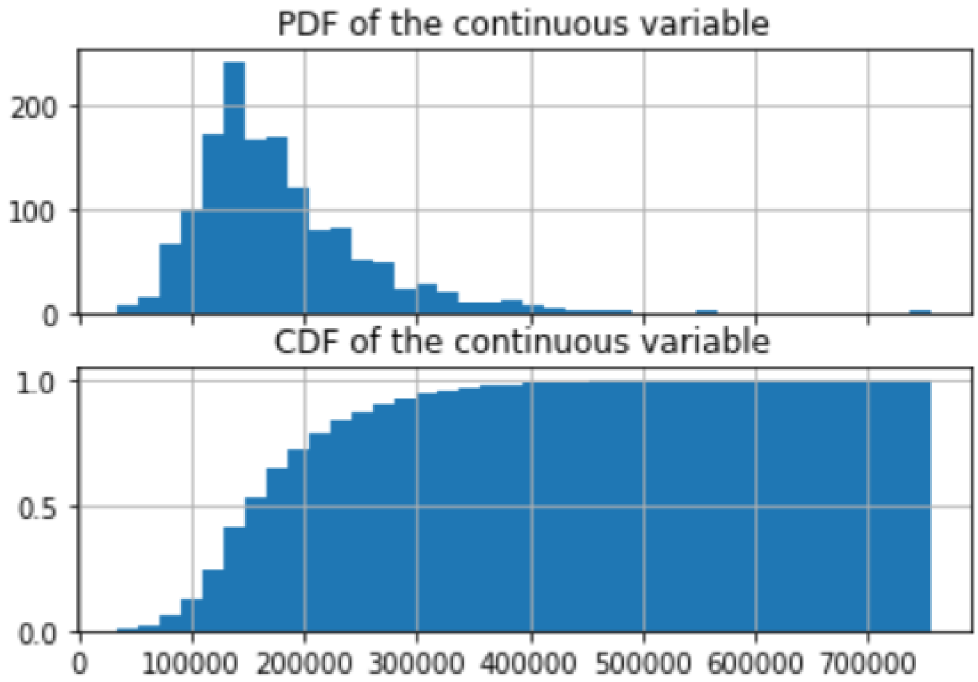
- ***Target Variable: Sale Price***

Below is the descriptive statistics for Sale Price:



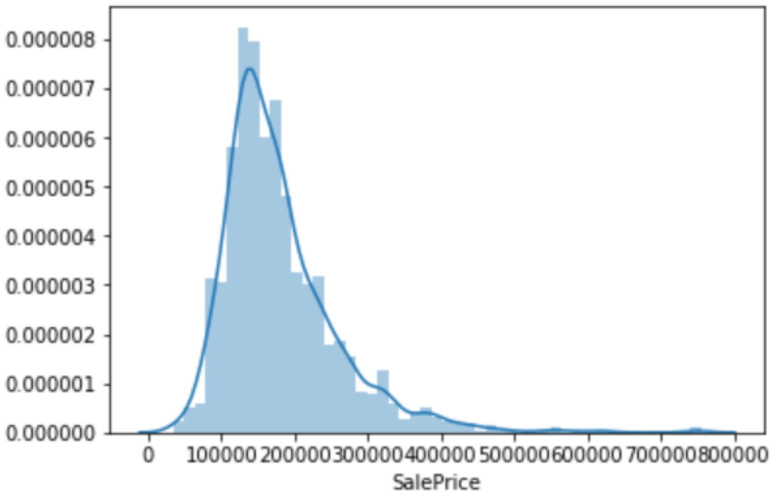
```
count      1460.000000
mean     180921.195890
std       79442.502883
min       34900.000000
25%      129975.000000
50%      163000.000000
75%      214000.000000
max      755000.000000
Name: SalePrice, dtype: float64
```

PDF and CDF of the house sale price:



The average house sale price is around $ 180,000 in Ames. Majority of the house is priced under $ 214,400.

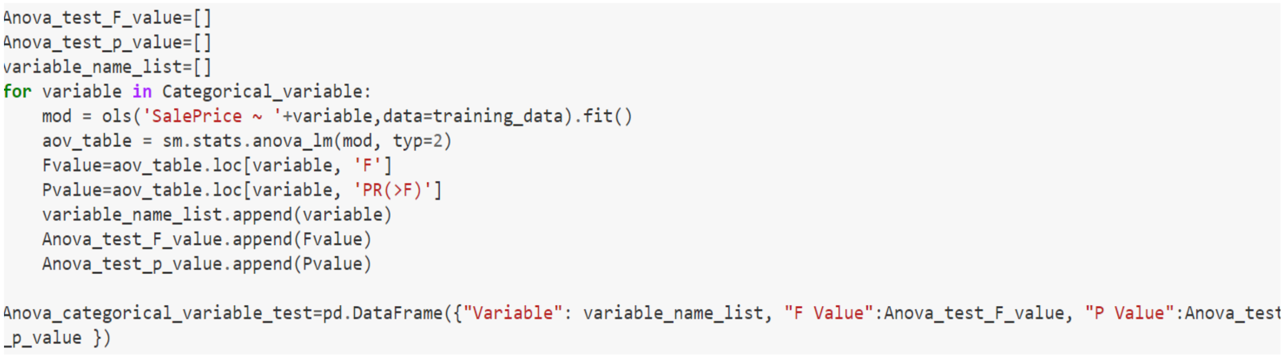
Here's the probability density plot of sale price:



By looking at the histogram of the sale price, the data is slightly skewed to the right. The distribution is also peaky compared with a normal distribution. So log transformation will be needed for the sale price before fitting the data into the model.

- What are the categorical variables that are highly correlated with Sale Price?

Methodology: For categorical variables, I'm using Anova test to check the significance of the categorical variables as below. The higher the F value (the lower the p value), the more significant the categorical variables would be to the sale price.

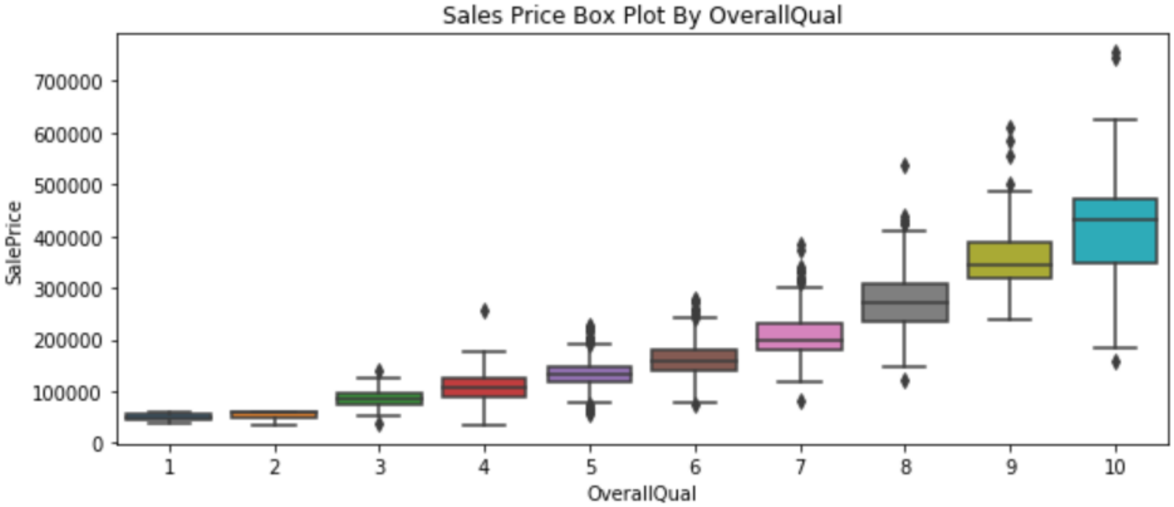


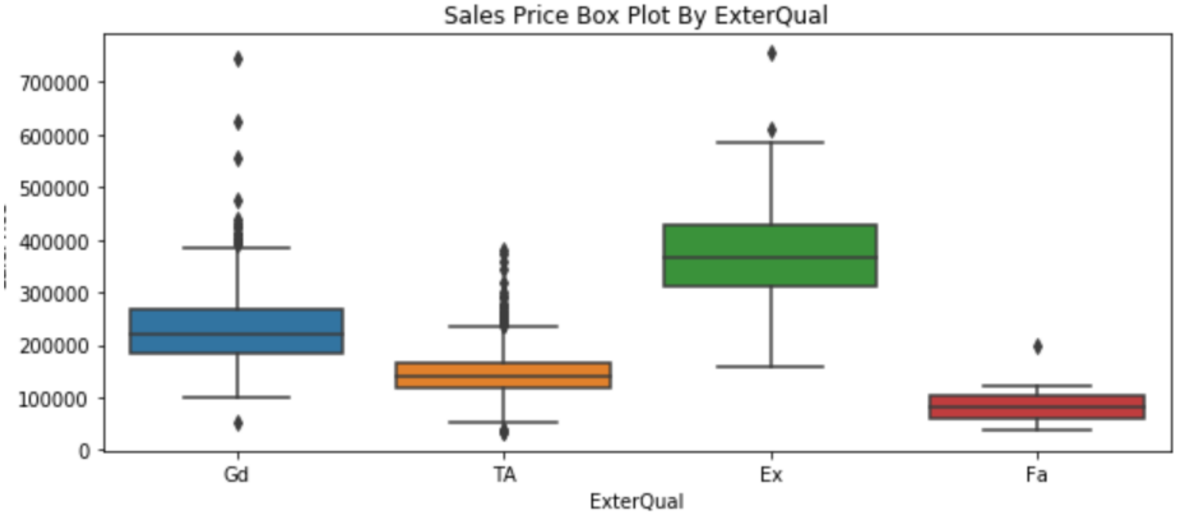
```
Anova_test_F_value=[]
Anova_test_p_value=[]
variable_name_list=[]
for variable in Categorical_variable:
    mod = ols('SalePrice ~ '+variable,data=training_data).fit()
    aov_table = sm.stats.anova_lm(mod, typ=2)
    Fvalue=aov_table.loc[variable, 'F']
    Pvalue=aov_table.loc[variable, 'PR(>F)']
    variable_name_list.append(variable)
    Anova_test_F_value.append(Fvalue)
    Anova_test_p_value.append(Pvalue)

Anova_categorical_variable_test=pd.DataFrame({"Variable": variable_name_list, "F Value":Anova_test_F_value, "P Value":Anova_test
_p_value })
```

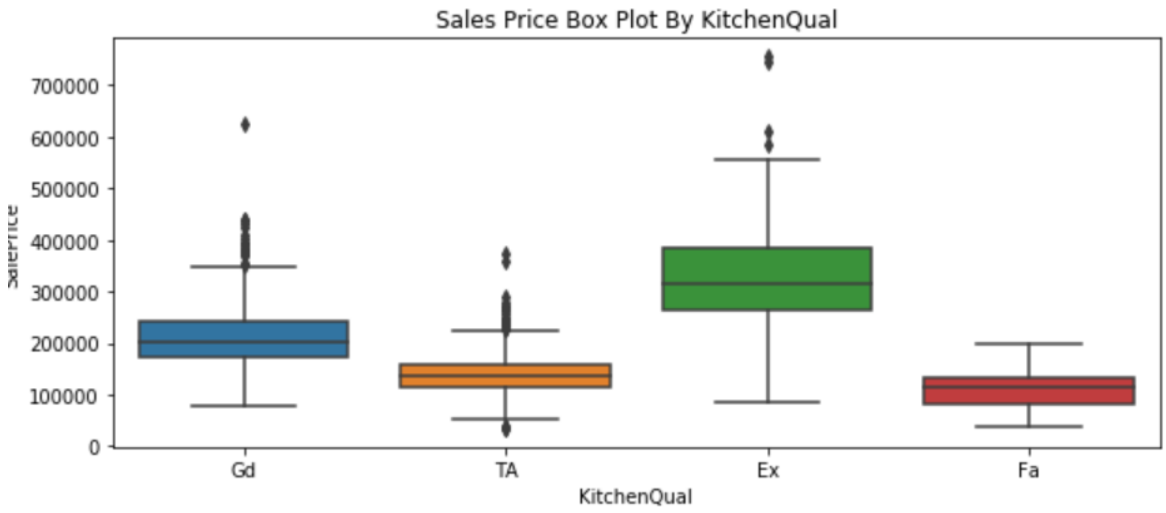
Findings:

- Overall quality turns out to be the most significant factor in predicting the sale price of the property. The higher quality score, the higher the sale price is.
- Exterior material quality is also a very important factor. The property excellent or good exterior quality tends to have higher sale price.
- Other than variables above, kitchen quality, Height of the basement, Interior finish of the garage, Masonry veneer type, Type of foundation, Central air conditioning are also closely related to the sale price. For example, the properties with brick or stone made masonry veneers have higher price point. The properties with central air conditioning tend to be more expensive than properties with no air conditioning.

Visualization for top correlated categorical variables:


Sales Price Box Plot By OverallQual

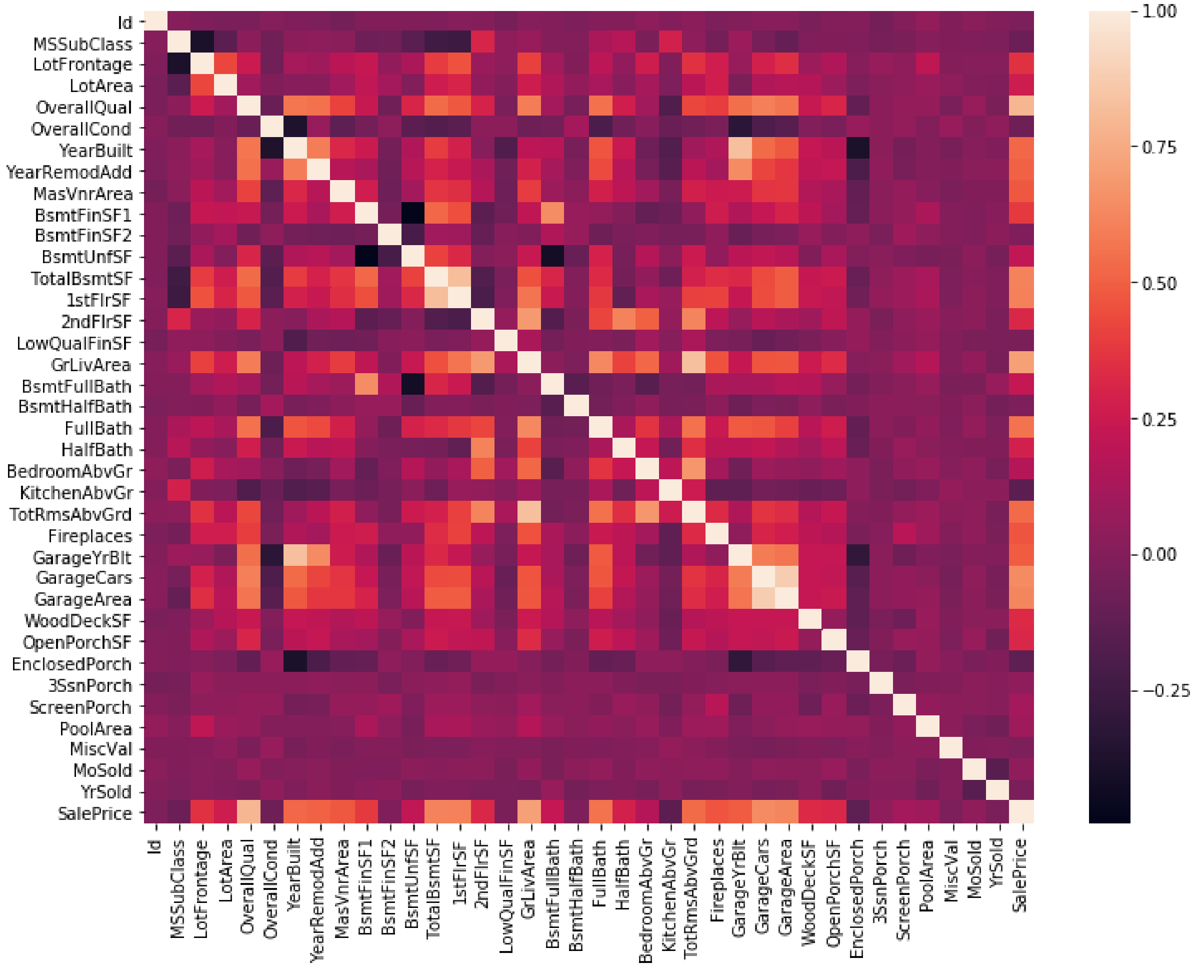

Sales Price Box Plot By ExterQual



- What are the numerical variables that are highly correlated with Sale Price?

  Methodology:
  To identify the highly correlated numerical variables, I'm using the correlation analysis by computing the R squared of each numerical features with the Sale Price.

  Below is the heat map from the R squared:

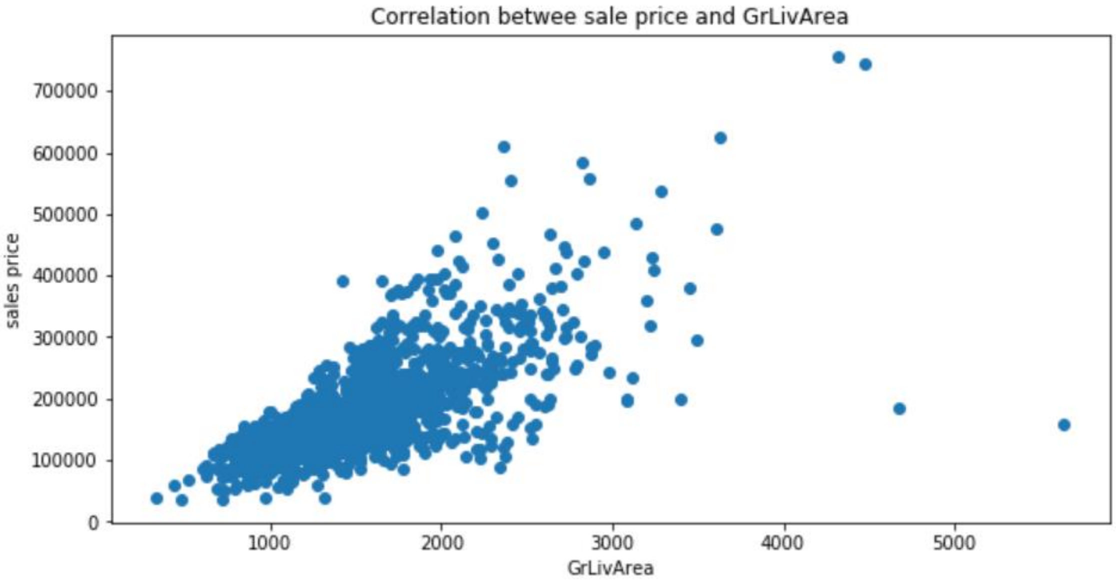


By ranking on the R Squared, here are the top numerical features that are most correlated with the house price:

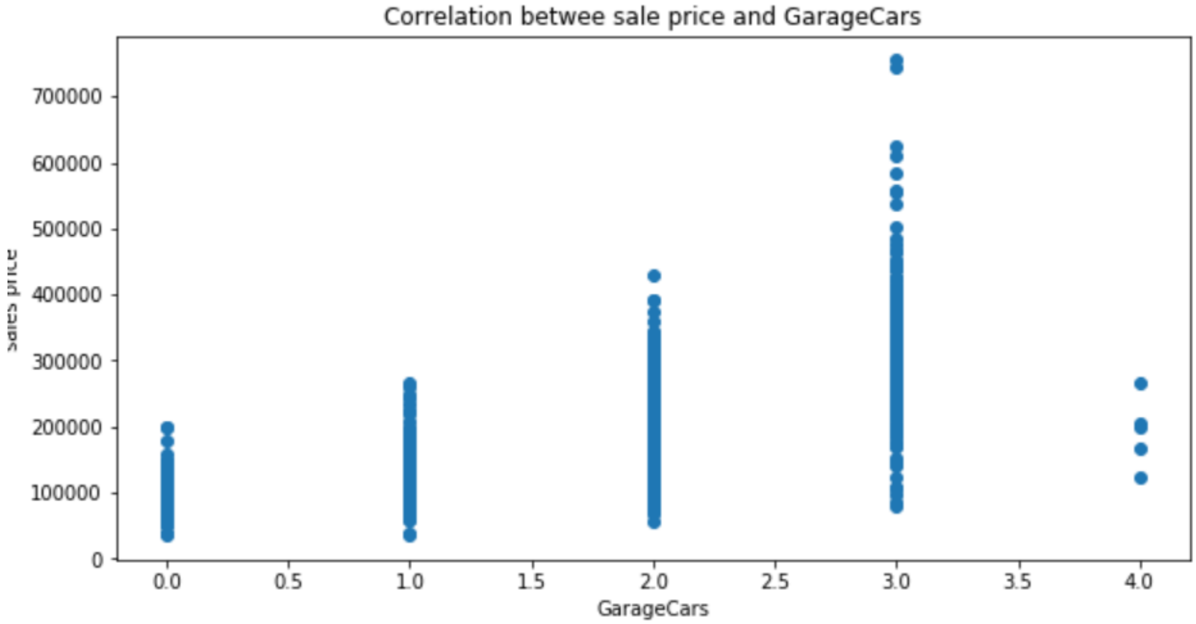
- GrLivArea: Above grade (ground) living area square feet
- GarageCars: Size of garage in car capacity
- GarageArea: Size of garage in square feet
- TotalBsmSf: Total square feet of basement area
- 1stFlrSF: First Floor square feet
- FullBath: Full bathrooms above grade
- TotRmsAbvGrd: Total rooms above grade (does not include bathrooms)
- MasVnrArea: Masonry veneer area in square feet
- Fireplaces: Number of fireplaces
- BsmtFinSF1: Type 1 finished square feet

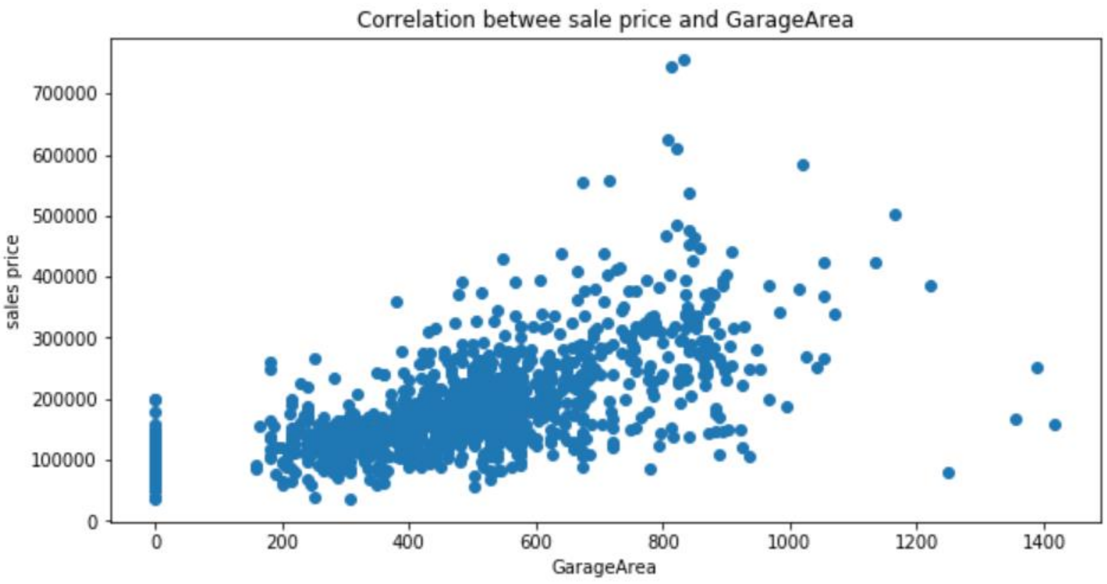
Findings on the top features with visualization:

- The more ground living area, the higher the price.

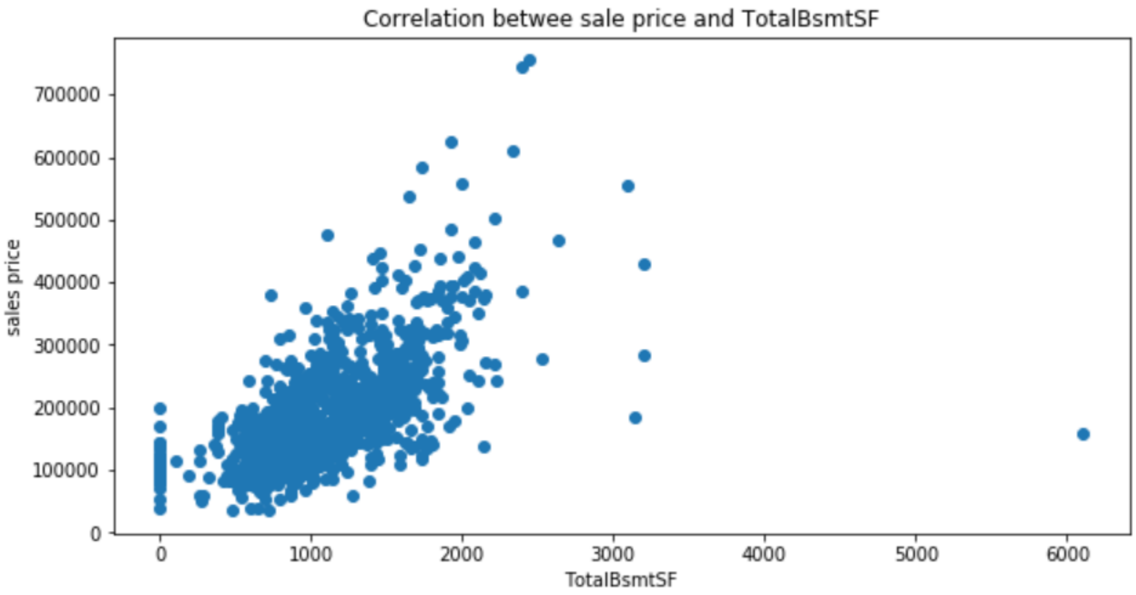


- There's some variation on the sale price based on garage size in car capacity. But in general, the price is going up. same finding on the garage size in square feet.

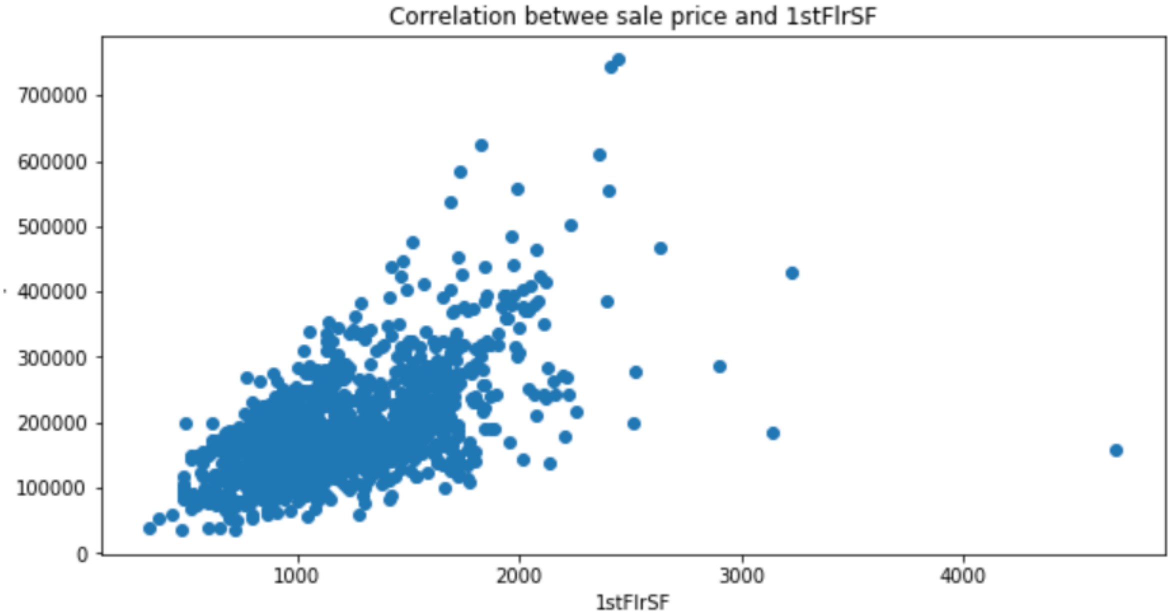




- There's a strong correlation between sale price and total basement size in square feet and first floor in square feet.



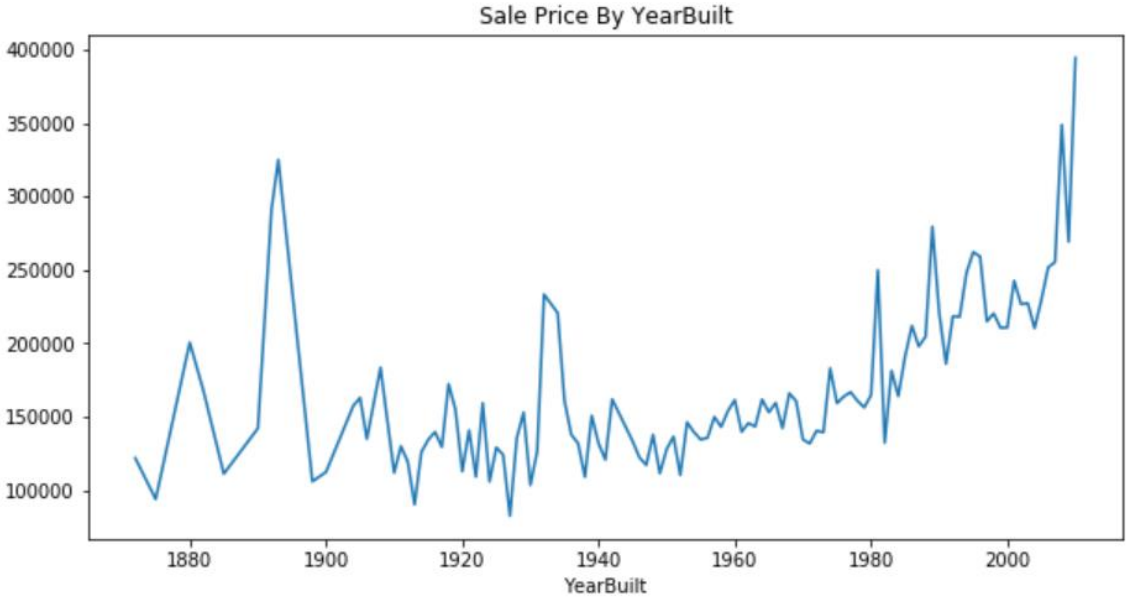
- For the full bathrooms above grade, the total rooms above grade, Masonry veneer area in square feet, number of fireplaces and type 1 finished square feet, there's some correlation between these variables with the price. But the correlation is not that strong.

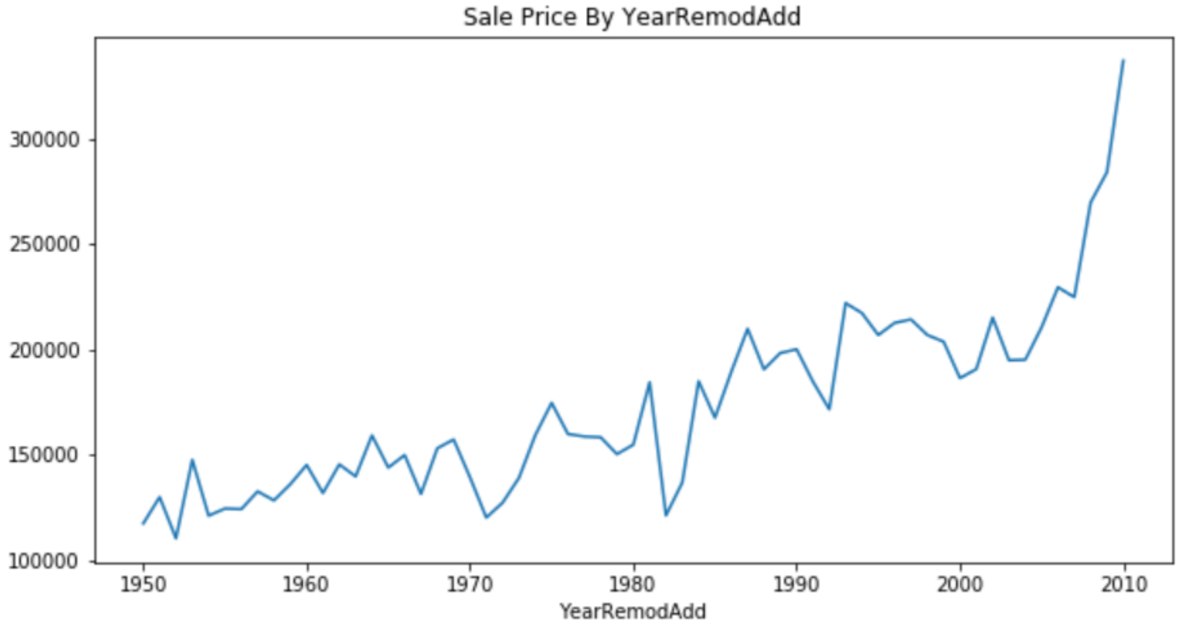


- What are the datetime variables that are highly correlated with Sale Price?

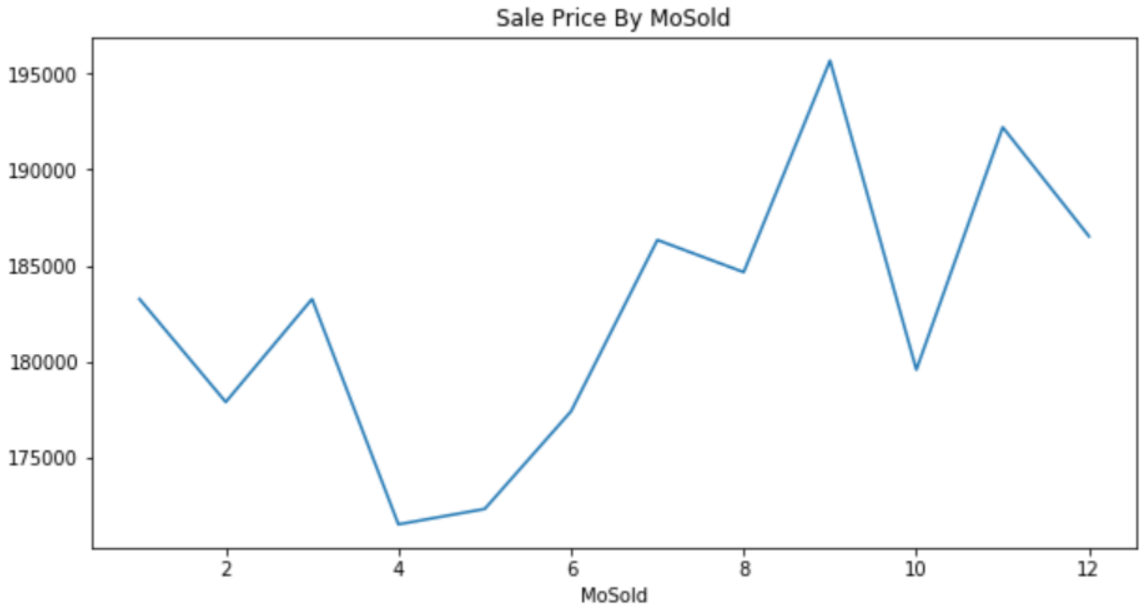
  Findings with Visualizations:

  - The more recent the properties were built or remodeled, the more expensive they get by looking at the saleprice by YearBuilt. But in the trend chart below, the average Sale price at some point right before 1900 is as high as the average price of the properties built after 2000.





- For the month when the properties were sold, the properties that were sold in the second half of the year were more expensive on average than the properties that were sold before June.



In-depth data analysis and model

- Cross Validation:
  For cross validation, I have split the data into 75% for training and 25% for testing.

- Base Models:

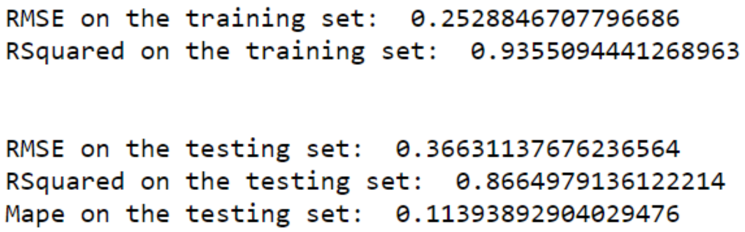
Since this is a regression problem, I have used the following machine learning techniques:

- Ridge
- Lasso
- Elastic Net
- Gradient Boosting
- Xg Boosting
- Light GBM

For each of the algorithm above, I have done hyperparameter tuning by using randomized search and gird search on the training set.

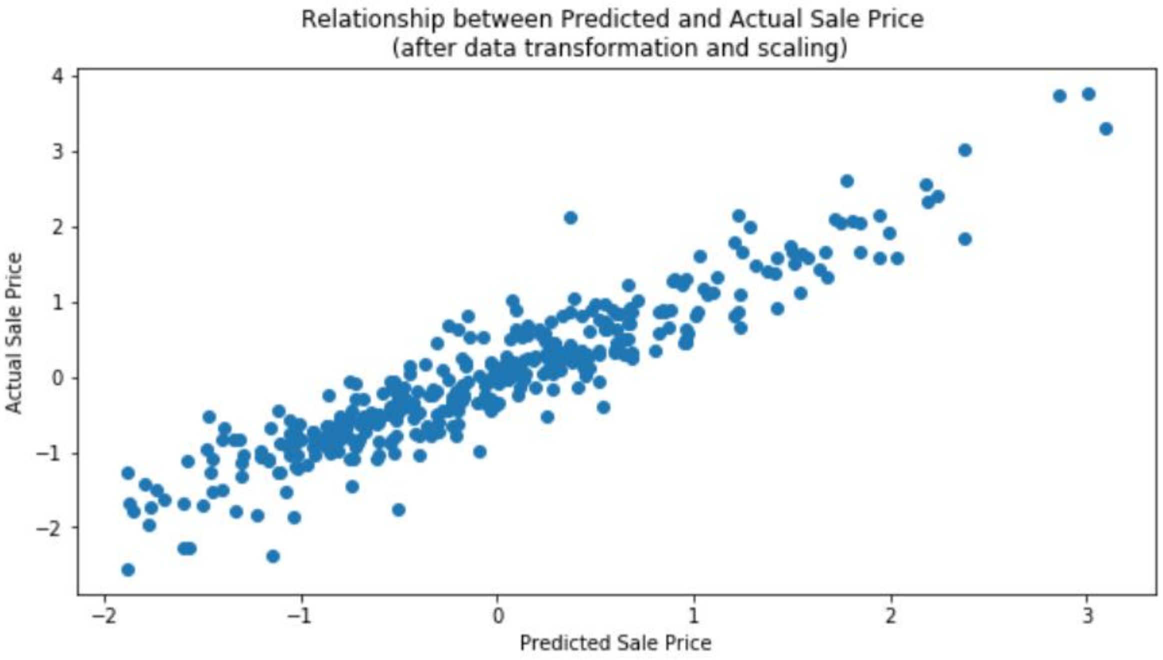
Below is an example of one of the models output (Light GBM):

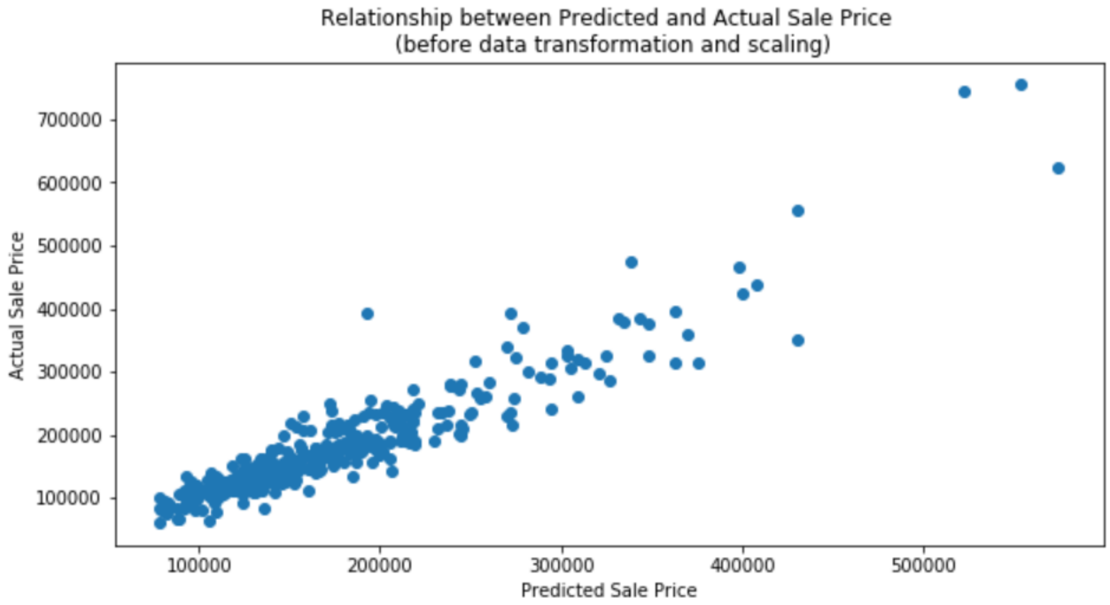
RMSE, R squared and Mape (after data scaling and transformation) on both training and testing set:



```
RMSE on the training set:  0.2528846707796686
RSquared on the training set:  0.9355094441268963


RMSE on the testing set:  0.36631137676236564
RSquared on the testing set:  0.8664979136122214
Mape on the testing set:  0.11393892904029476
```

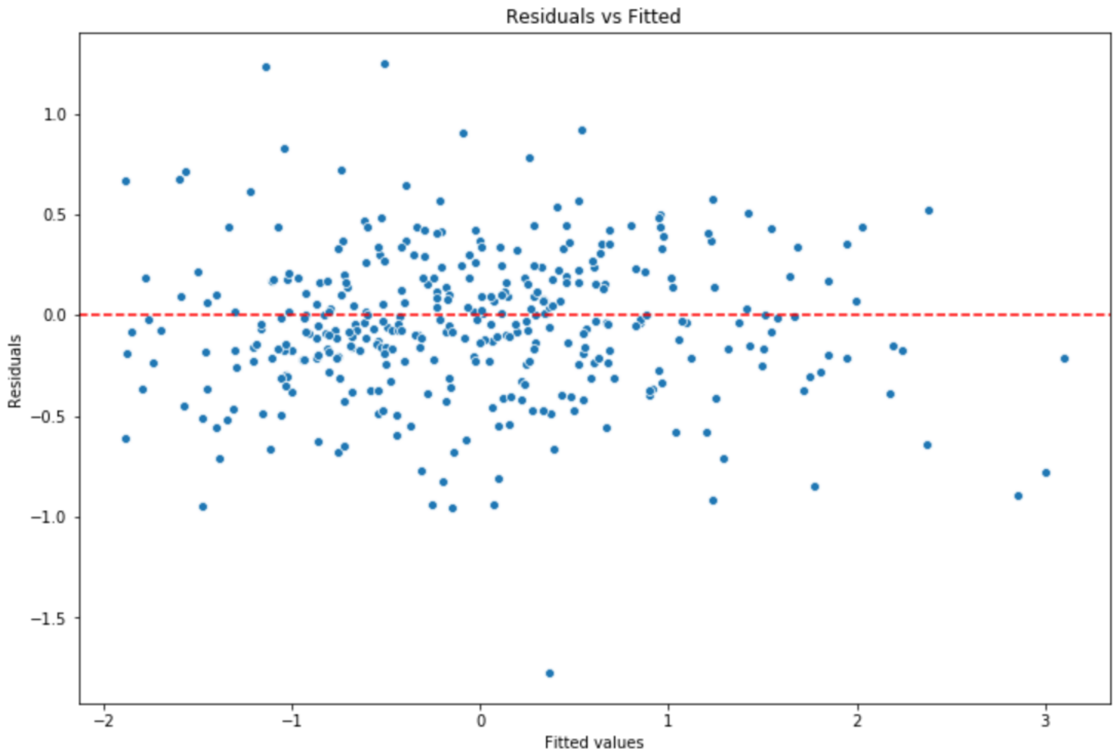
Relationship between actual data and prediction:

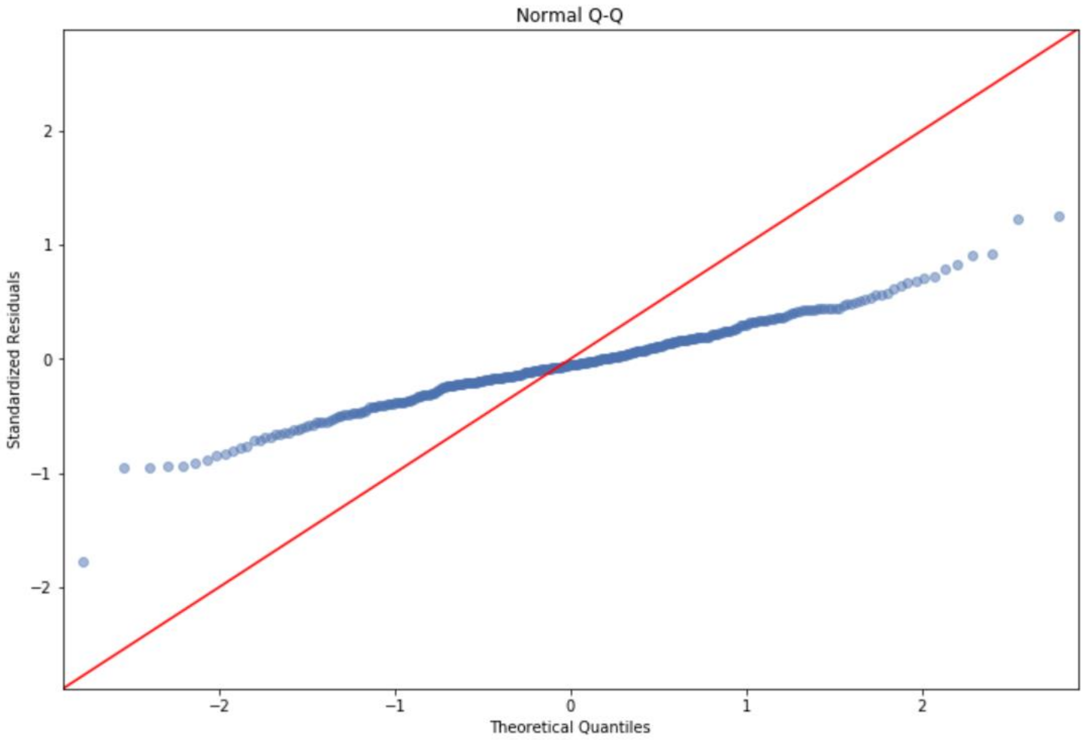




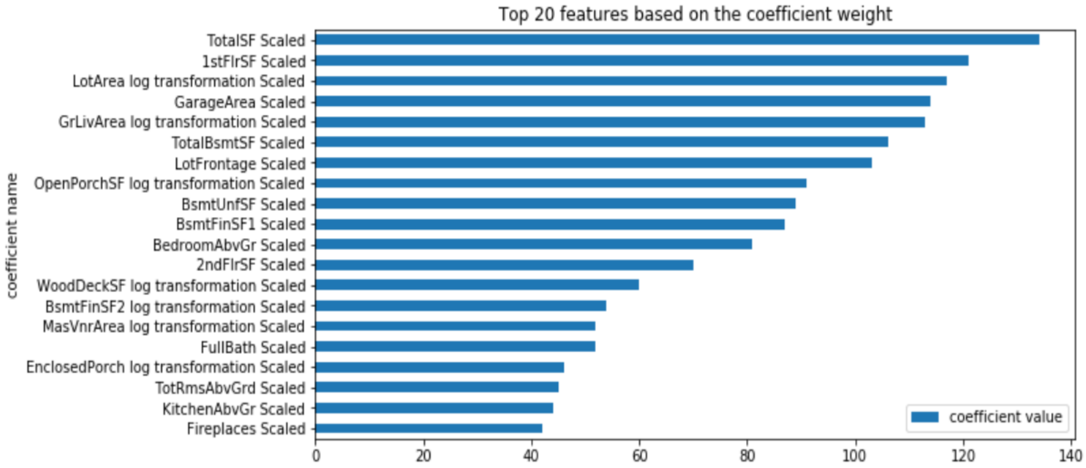
Residual plot:

For any linear regression-based model like Ridge, lasso and elastic net, the residual would be expected to be randomly and normally distributed

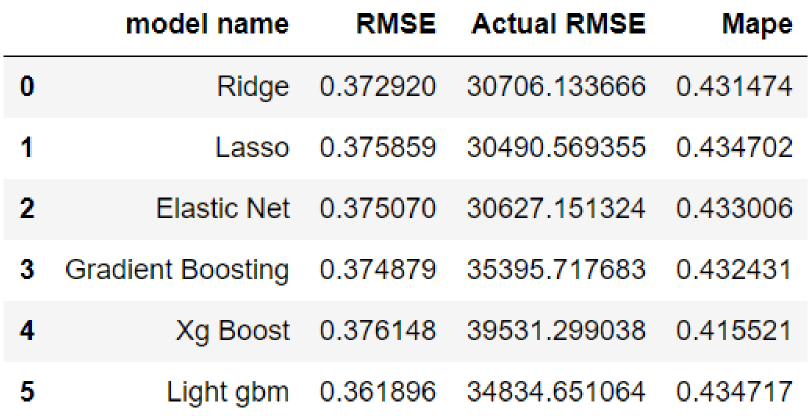




Top features based on the coefficient/ feature importance weight and feature importance:



Below is the comparison on the performance across the base models, as you can see Light GBM is performing better than the other models:



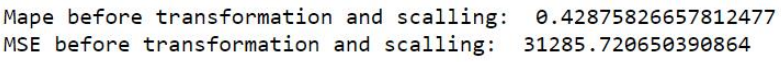
	model name	RMSE	Actual RMSE	Mape
0	Ridge	0.372920	30706.133666	0.431474
1	Lasso	0.375859	30490.569355	0.434702
2	Elastic Net	0.375070	30627.151324	0.433006
3	Gradient Boosting	0.374879	35395.717683	0.432431
4	Xg Boost	0.376148	39531.299038	0.415521
5	Light gbm	0.361896	34834.651064	0.434717

- Stacked Model:

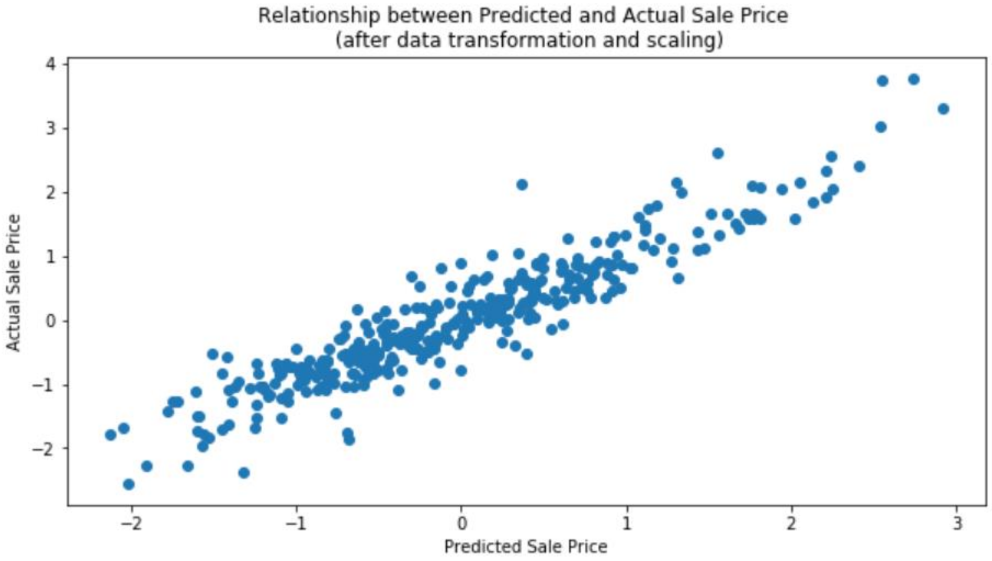
Other than just leveraging the base models, I have ensembled the base models I have trained above into a stacked model. Basically, the stacked model takes in the predictions from all the base models for one single record and compute the mean of the predictions as one single prediction.

By comparing on the RMSE (before transformation and scaling), the stacked model seems to work better than one single model.

Below is the screenshot of the performance:

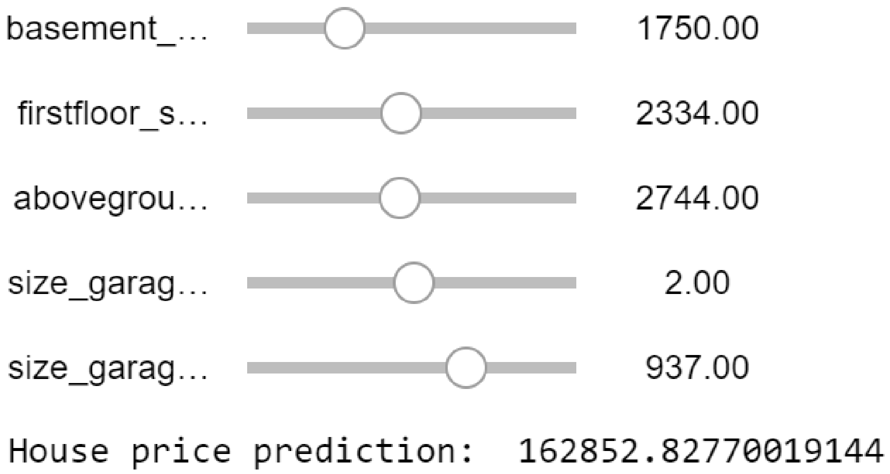


```
Mape before transformation and scalling:  0.42875826657812477
MSE before transformation and scalling:  31285.720650390864
```



Prediction Application

I have built a prediction demo to take in the change of the variables below like basement size in square feet, first floor area in square as input to for predicting the house price. Note that since there are too many variables in the model, for demonstration purpose, I'm just using default values for the rest of the attributes that are not shown in the prediction application below.



basement_… 1750.00

firstfloor_s… 2334.00

abovegrou… 2744.00

size_garag… 2.00

size_garag… 937.00

```
House price prediction:  162852.82770019144
```