Springboard-Data Science Career Program

Capstone Project # 2 – Predict house price

Final Report

Malleswari Gelli

March 2022

1. **Introduction**

With house prices near all-time high across United States since pandemic in 2020, I wanted to see if there are machine learning models for predicting house prices. When I asked many real estate agents how they determine the ideal sale price of the house before putting it on the market, the common answer I heard was we look at the comparable properties in the areas. However, they aren’t certain of what features of the house determines the house prices. My goal for this project is to build an end-to-end solution or application that can predict the house prices better than individuals/humans.

I will be building ML models to predict median house prices in Iowa considering 79 explanatory variables describing (almost) every aspect of residential homes in Ames, Iowa. This is supervised learning problem as the data set consists of labelled observations and it does looks like multivariate regression should be the option, but I will explore multiple ways of building the model and finally pick the one with lowest error rate RMSE (Root Mean Square Error) or MAE (Mean Absolute Error) or any other metrics to predict/validate the new datasets in future.

1. **Approach:**
   1. **Data Acquisition, Wrangling**

**Data Source**: House price dataset from Kaggle competition

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

1. Train.csv, contains several features including target variable SalePrice of house. Use this dataset to train the machine learning regression model.
2. Test.csv, contains all 80 features in train.csv file except the target variable. So, predict the SalePrice of the houses using best model trained on train.csv dataset.

**Data wrangling Summary**:

Train.csv contains 81 features including Target variable: SalePrice

Test.csv contains 80 features as train.csv but doesn't contain Target variable

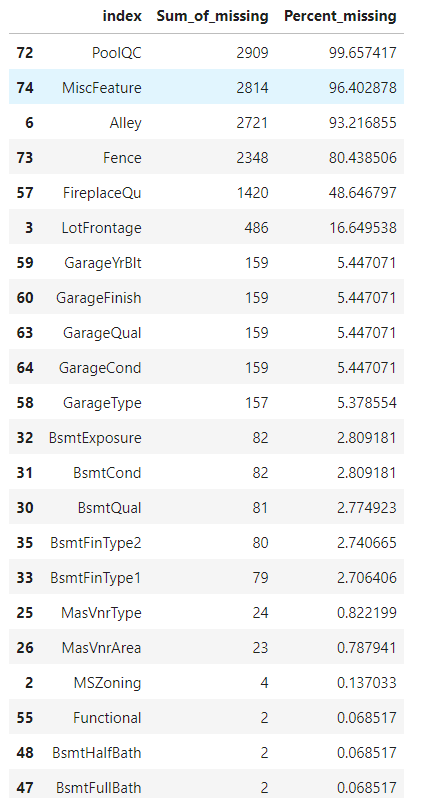
Since Train.csv and Test.csv contains same features and Id column is consecutive, I concatenated the two data frames by separating target variable from Train.csv into Target data frame. Target data frame contains Id, SalePrice columns from Train.csv. Having Id column helps later if data frame merging is required.

* Glad to see target feature had no missing values, no need to drop any rows or imputation of missing data is not needed.



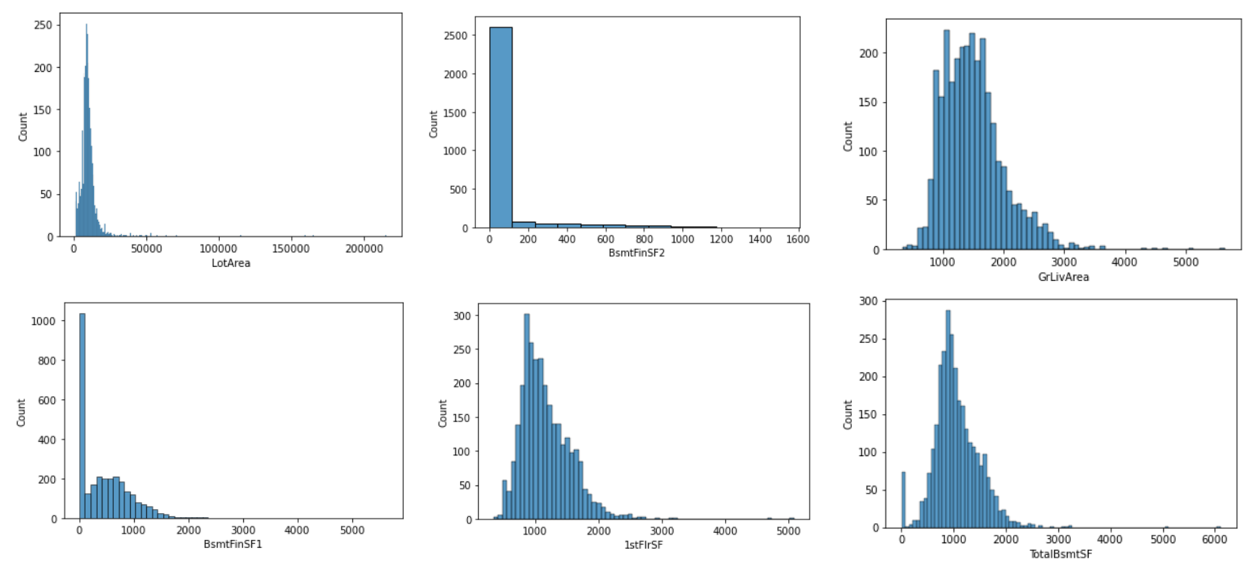
Concatenated Train.csv features except SalePrice column with same features of Test.csv from combined dataframe, Basement features ranges from 0 to 6110 sft; 0 means there is no basement in the house. Rather than removing the rows with 0 basement and/or imputing the 0's with column means; I replaced 0 with 1 (just adding 1 sft doesn't inflates the column values).

There are 18 features contain >23 missing values, also these are not important features determining house prices. So, dropped these 18 features from combined df and train df.

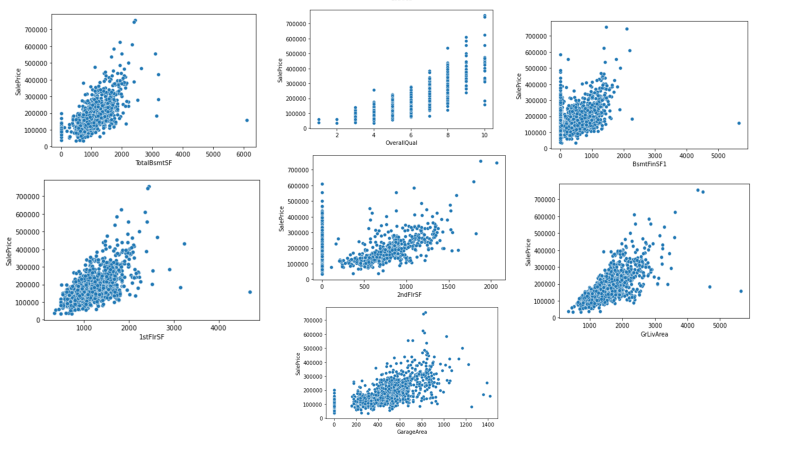


1. **EDA summary/Understand how variables are distributed and how they interact**

* Univariate plot/Histogram of numeric columns from dataset shows that LotArea, BsmtFinSF1, BsmtFinSF2, TotalBsmtSF, 1stFlrSF, GrLivArea, GarageArea columns had skewed distribution; may require log transformation

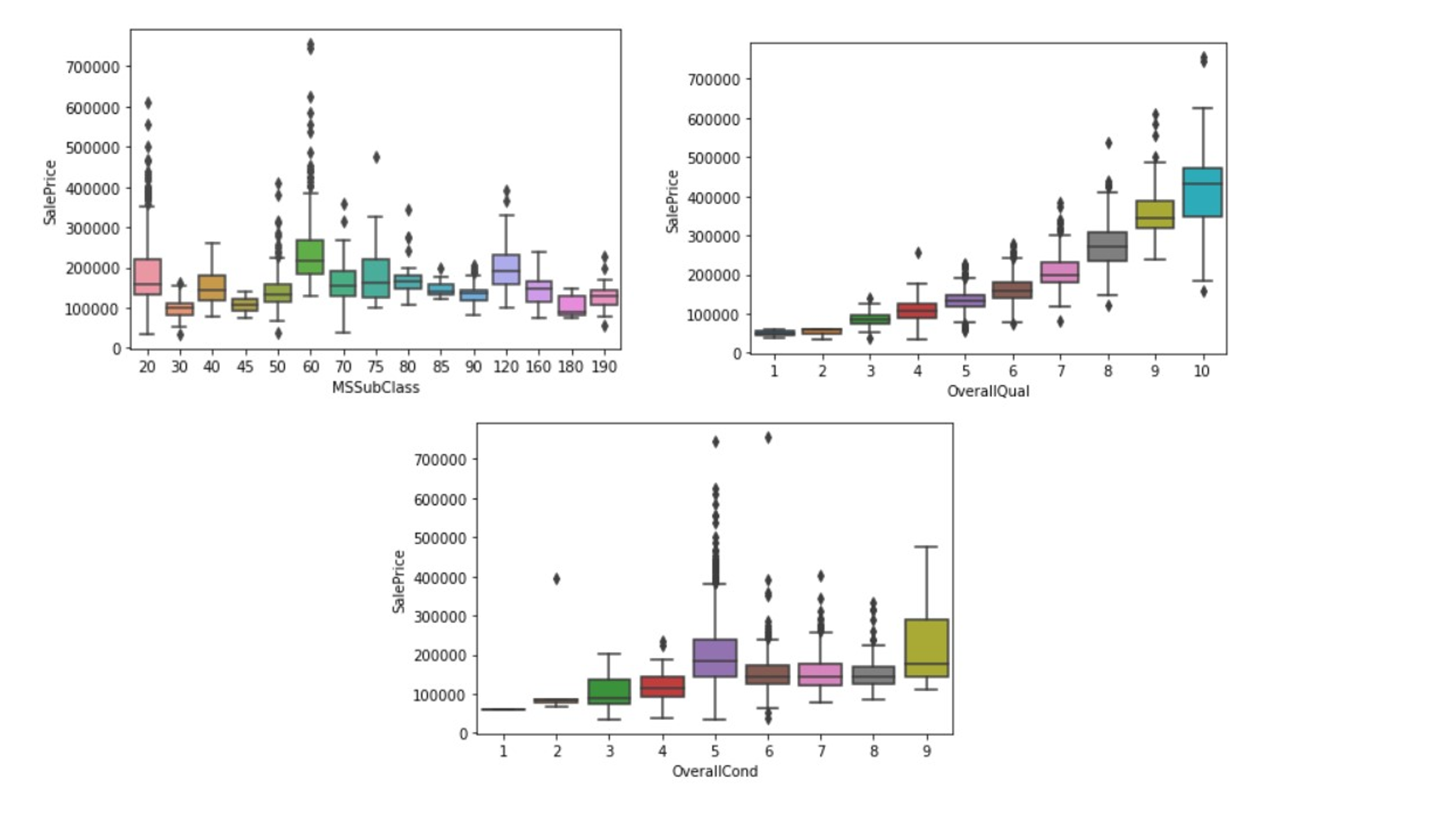


* Bivariate plots/scatter plots of numeric columns with SalePrice shows that TotalBmsSF, OverallQual, BsmtFinSF1, 1stFlrSF, 2ndFlrSF, GrLivArea, GarageArea related linearly with SalePrice

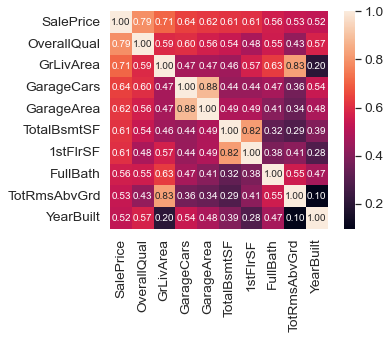


* Since some numeric features contain discrete values. BoxPlotting of those features with Sale price showed better distribution of SalePrice across different groups of features.

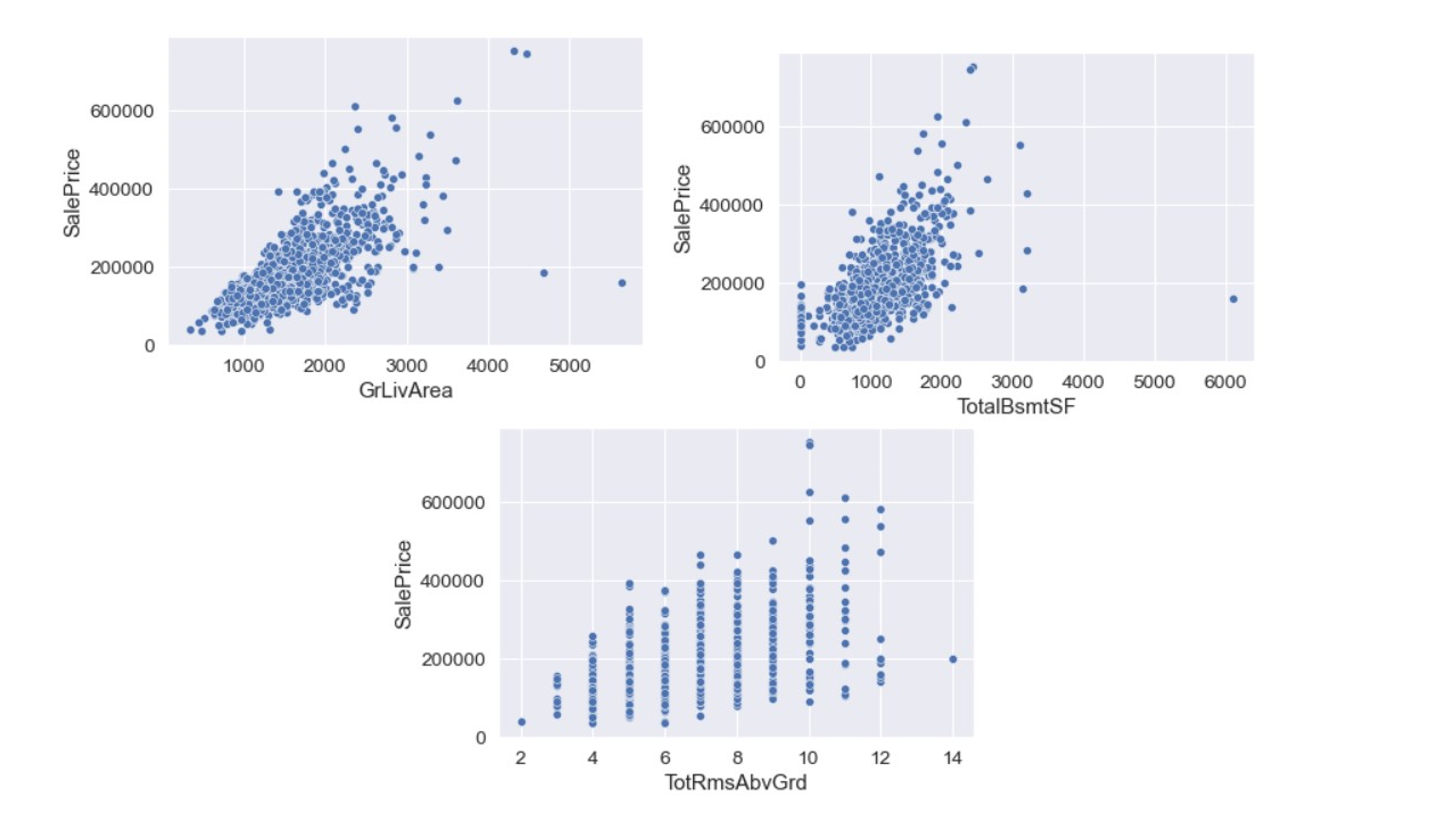
For example: MSSubClass, OverallQual and OverallCond had several discreate values; created new feature by grouping the values.



* With Correlation matrix, found top10 predictive features of SalePrice



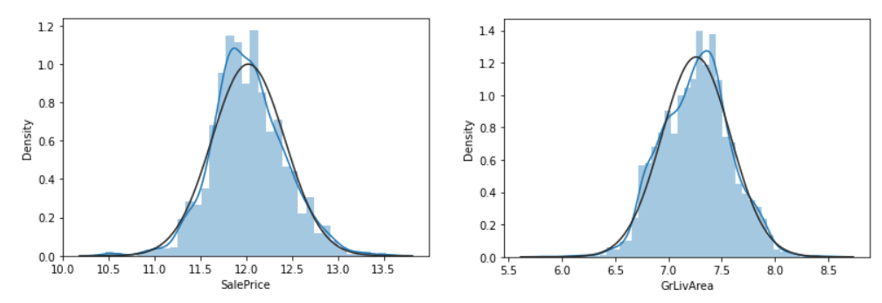
* Outliers are detected with GrLivArea >4500; TotalBsmtSF>6000; TotRmsAbvGrd>12. Dropped outliers from combined and trained\_data\_sets



1. **Preprocessing**

**Applied different transformations before training machine learning models**

* 1. **Log transformation**

Log transformation improved the distribution of features. For example: Distribution of SalePrice, GrLivArea features are normal with log transformation.

* 1. **Dummies:**

Since many machine learning algorithms cannot operate on label data directly and require all input variables and output variables to be numeric, I converted the categorical data to a numerical form using “Dummies function” from Pandas; this function converts categorical features to numerical values of 0 and 1; 0 being absence and 1 being presence. This increased the number of features to 220 from 63.

**4.3 Train-Test-Split:**

Then, I split the data into train and test set (30%) to evaluate the performance of models I am going to be tested. Because the data in the testing test already contains known values of the SalePrice (feature I want to predict), it is easy to determine whether the models used guesses and/or predictions are correct. Also, splitting data determines how well my model is generalizing and prevents model being overfitting or evaluating the model effectively.

**4.4 Scaling data**

Then, MinMaxScaler is fit to train data and transform the scaling to test dataset. Scaling is required to rescale the data and it’s used when we want features to be compared on the same scale for our algorithm. And, when all features are in the same scale, it also helps algorithms to understand the relative relationship better. Since dependent features were transformed to normality, scaling should be applied after transformation

1. **Modeling**
   1. **Multiple Linear Regression model is applied**

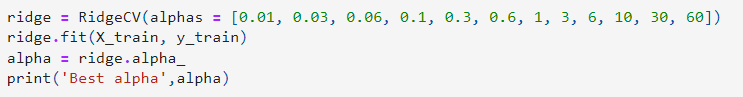
y= a1x1 + a2x2 + a3x3+....+b (linear regression minimizes a loss function (OLS) while choosing coefficients for fitting the model in sklearn.

* + R^2: -8.145738372417883e+20
  + Root Mean Square Error: 11235213346.399387

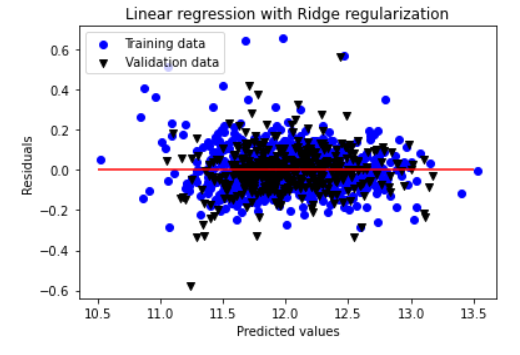
Five-fold cross validation is applied for evaluating the model. 5-fold cross val scores (i.e R^2 values):[0.91369528 0.90455213 0.90121937 0.91505553 0.90773841] with mean R^2 value: 0.9084521432489213. Large coefficients can lead to overfitting of Linear Regression; So, Regularization is required to penalize overfitting by handling collinearity

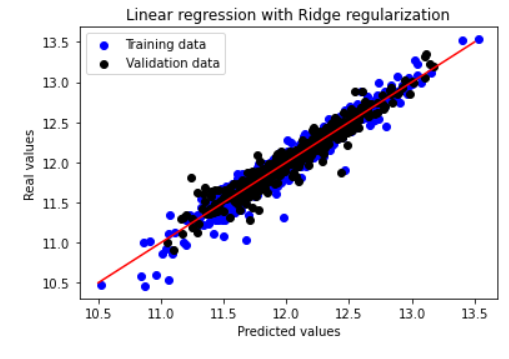
**5.2 Ridge Regression is applied**

With varied alpha values

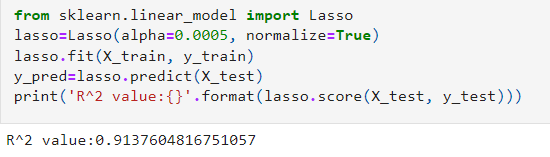


Best alpha 1.0, R^2 value:0.92668902660498 and RMSE: 0.10658610404310023





**5.3 Then, Lasso Regression is applied**



1. **Conclusions and feature work**

Among different regression models, Ridge regression had higher accuracy, so it is applied to predict the SalePrice of new test dataset.

Future work could include:

1. Size of the dataset can be increased by considering historical data from several years, it helps determining trends of house price fluctuations
2. More features of the data can be incorporated. For example: demographics, income of the household, tourism spots, schools, neighborhood etc to determine what features driving the house prices
3. Since the dataset I tested is just from Ames, Iowa and it’s difficult to say how high or low is the house price in Ames vs other cities of Iowa and/or other states in mid-west. So, these models can be validated in datasets from other cities and applied to predict the house prices.