1. Introduction

An accurate prediction on the house price is important to prospective homeowners, developers, investors, appraisers, tax assessors and other real estate market participants, such as, mortgage lenders and insurers. Traditional house price prediction is based on cost and sale price comparison lacking an accepted standard and a certification process. Therefore, the availability of a house price prediction model helps fill up an important information gap and improve the efficiency of the real estate market.

Real estate market is booming in the United States, every person’s dreams is to have a perfect house. As house market in the USA is thriving house price becomes a crucial factor for a home seeker. Research shows that important factors that influence the house price are housing site, housing quality, geographical location and the environment.

2. Client

This analysis report can be an interest to any Real estate company, Real estate investors, Mortgage lenders and Home insurers. This report helps make decisions easy for the businesses and home seekers.

3. Dataset

Dataset consists of historical house prices of residential homes in Ames, Iowa. The dataset consists of 81 exploratory features with 1460 observations. The dataset is extracted from Kaggle https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data

The data set contains every minute detail of the house. Some of the major features in this data set are:

1. Lot Area
2. Neighbourhood
3. House Style
4. Quality of the house
5. Overall condition of the house
6. Year built
7. Year remodelled
8. Foundation
9. Basement Condition
10. Total basement square feet
11. 1sty floor square feet
12. 2Nd floor square feet
13. Above ground living area in square feet
14. Full bathrooms above ground
15. Bedrooms above grade
16. Total rooms above grade
17. Garage size in square feet
18. Garage quality

However, it is good idea to explore the data set from Kaggle to get good idea on the data.

|  |  |
| --- | --- |
| Id | 1460 non-null int64 |
| MS Subclass | 1460 non-null int64 |
| MSZoning | 1460 non-null object |
| LotFrontage | 1201 non-null float64 |
| LotArea | 1460 non-null int64 |
| Street | 1460 non-null object |
| Alley | 91 non-null object |
| LotShape | 1460 non-null object |
| LandContour | 1460 non-null object |
| Utilities | 1460 non-null object |
| LotConfig | 1460 non-null object |
| LandSlope | 1460 non-null object |
| Neighborhood | 1460 non-null object |
| Condition1 | 1460 non-null object |
| Condition2 | 1460 non-null object |
| BldgType | 1460 non-null object |
| HouseStyle | 1460 non-null object |
| OverallQual | 1460 non-null int64 |
| OverallCond | 1460 non-null int64 |
| YearBuilt | 1460 non-null int64 |
| YearRemodAdd | 1460 non-null int64 |
| RoofStyle | 1460 non-null object |
| RoofMatl | 1460 non-null object |
| Exterior1st | 1460 non-null object |
| Exterior2nd | 1460 non-null object |
| MasVnrType | 1452 non-null object |
| MasVnrArea | 1452 non-null float64 |
| ExterQual | 1460 non-null object |
| ExterCond | 1460 non-null object |
| Foundation | 1460 non-null object |
| BsmtQual | 1423 non-null object |
| BsmtCond | 1423 non-null object |
| BsmtExposure | 1422 non-null object |
| BsmtFinType1 | 1423 non-null object |
| BsmtFinSF1 | 1460 non-null int64 |
| BsmtFinType2 | 1422 non-null object |
| BsmtFinSF2 | 1460 non-null int64 |
| BsmtUnfSF | 1460 non-null int64 |
| TotalBsmtSF | 1460 non-null int64 |
| Heating | 1460 non-null object |
| HeatingQC | 1460 non-null object |
| CentralAir | 1460 non-null object |
| Electrical | 1459 non-null object |
| 1stFlrSF | 1460 non-null int64 |
| 2ndFlrSF | 1460 non-null int64 |
| LowQualFinSF | 1460 non-null int64 |
| GrLivArea | 1460 non-null int64 |
| BsmtFullBath | 1460 non-null int64 |
| BsmtHalfBath | 1460 non-null int64 |
| FullBath | 1460 non-null int64 |
| HalfBath | 1460 non-null int64 |
| BedroomAbvGr | 1460 non-null int64 |
| KitchenAbvGr | 1460 non-null int64 |
| KitchenQual | 1460 non-null object |
| TotRmsAbvGrd | 1460 non-null int64 |
| Functional | 1460 non-null object |
| Fireplaces | 1460 non-null int64 |
| FireplaceQu | 770 non-null object |
| GarageType | 1379 non-null object |
| GarageYrBlt | 1379 non-null float64 |
| GarageFinish | 1379 non-null object |
| GarageCars | 1460 non-null int64 |
| GarageArea | 1460 non-null int64 |
| GarageQual | 1379 non-null object |
| GarageCond | 1379 non-null object |
| PavedDrive | 1460 non-null object |
| WoodDeckSF | 1460 non-null int64 |
| OpenPorchSF | 1460 non-null int64 |
| EnclosedPorch | 1460 non-null int64 |
| 3SsnPorch | 1460 non-null int64 |
| ScreenPorch | 1460 non-null int64 |
| PoolArea | 1460 non-null int64 |
| PoolQC | 7 non-null object |
| Fence | 281 non-null object |
| MiscFeature | 54 non-null object |
| MiscVal | 1460 non-null int64 |
| MoSold | 1460 non-null int64 |
| YrSold | 1460 non-null int64 |
| SaleType | 1460 non-null object |
| SaleCondition | 1460 non-null object |
| SalePrice | 1460 non-null int64 |
| dtypes: float64(3), int64(35), object(43) | |

4. Data Cleaning

Data cleaning is an extremely important step for any data analysis. It is very crucial for data to be organized. This process typically includes manually converting/mapping data from one raw form into another format to allow for more convenient consumption and organization of the data.

Data Cleaning steps carried out in this project are:

1. Handling missing data
2. Handling inconsistent data in a few variables

House Prices data set information:

The output above is produced from info() function. There are a few categorical and numerical variables with missing values.

1. **Handling Missing Data:** 
   * Categorical Data: The categorical variables with missing values are ‘MasVnrType’ and ‘Electrical’. Python provides many methods like fillna, forward/ backward filling, dropna etc. for handling missing data. I introduced another category called ‘missing’ to all the null values. This way I am retaining the original information of the data and not guessing anything.
   * Numerical Data: The most popular method to handle missing numerical data is Mean Imputation. I applied the same on my numerical data. Mean imputation is a method in which the missing value on a certain variable is replaced by the mean of the available cases. This is a reliable method for handling missing numerical data.
2. **Handling inconsistent data:** There are a few null values in the data set which are not actually nulls but are entered wrongly as nulls. Referring to the actual data set description file (data\_description.txt) from Kaggle, a few values were coded as ‘NA’ if a feature was not present in the house, but these NA values were entered as Nan in the .csv file. I decoded these misinterpreted values as ‘No feature\_name’ (feature\_name being name of the feature not present in the house).

5. **NewDataSet**

The data is now clean without any null/ inconsistent values. I transferred this data into a new csv file ‘house\_prices\_cleaned.csv’. I will use this data set for data exploration.

6. **Data Exploration**

Data exploration is the first step in data analysis and typically involves summarizing the main characteristics of a dataset. It is commonly conducted using visual analytics tools. Data Visualization is best way to explore the data because it allows users to quickly and simply view most of the relevant features of the dataset. By displaying data graphically scatter plots/ bar charts to name a few – users can identify variables that are likely to have interesting observations and if they are helpful for further in-depth analysis.

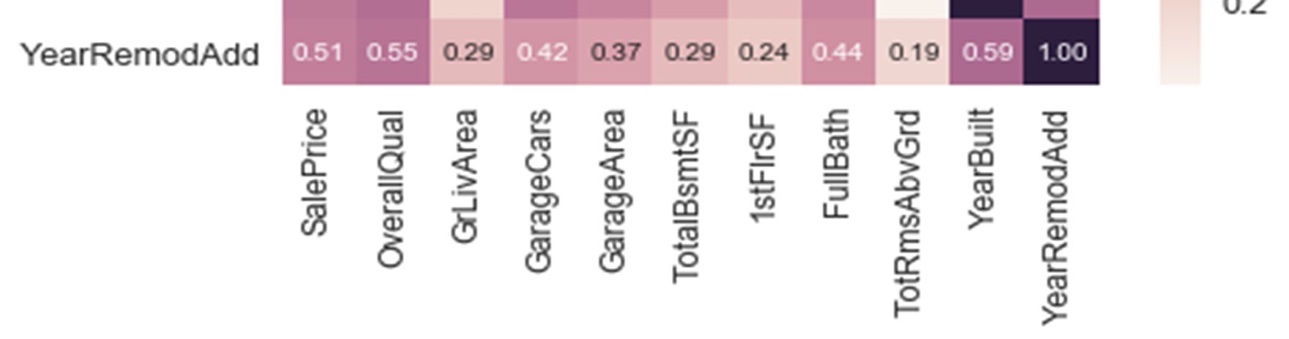
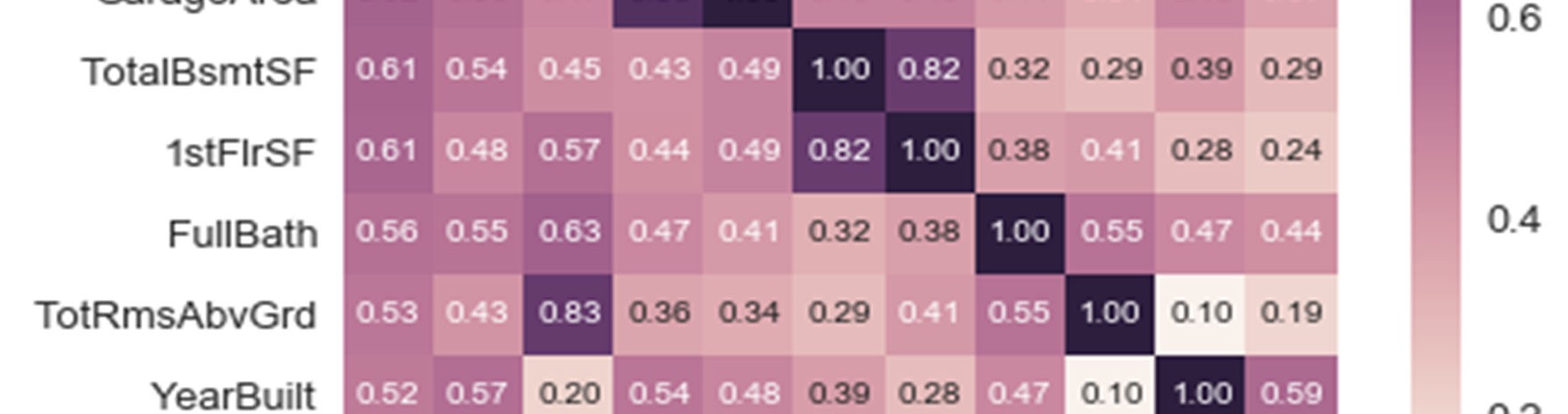
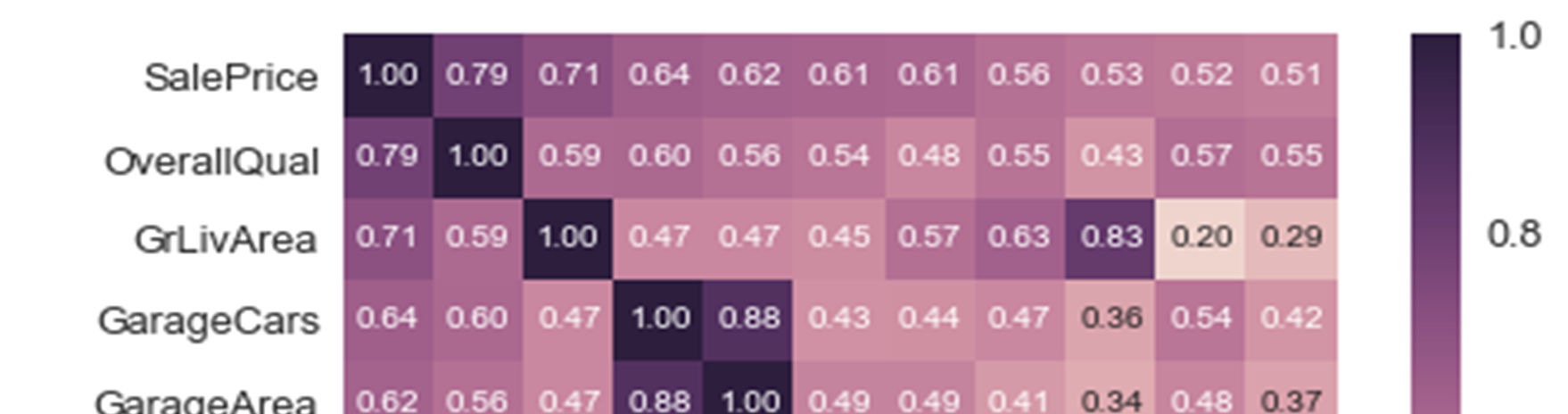
I used seaborn library provided by Python for my visualizations. I divided the data frame into numerical and categorical – containing quantitative and qualitative data respectively for the ease of analysis.

a. Multicollinearity: Multicollinearity exists when two or more of the predictors highly correlated, this might lead to an increase in the variance of the coefficient estimates and make the estimates very sensitive to minor changes in the model. I used Heat map to find out highly correlated independent variables. From the graph, we can see that features like:

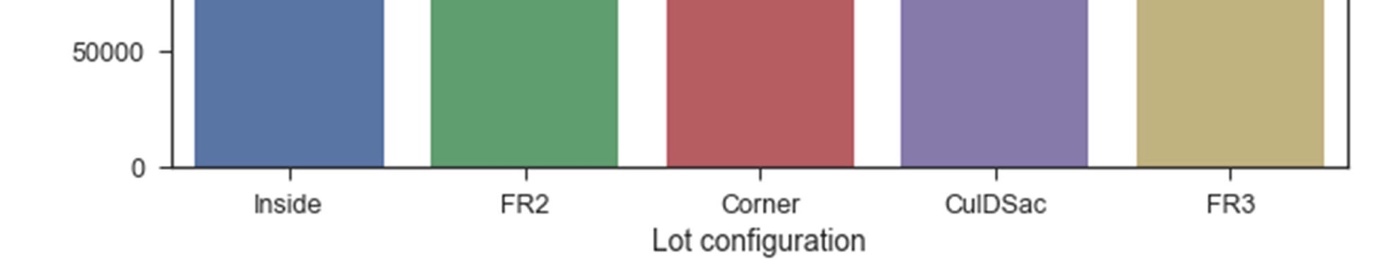
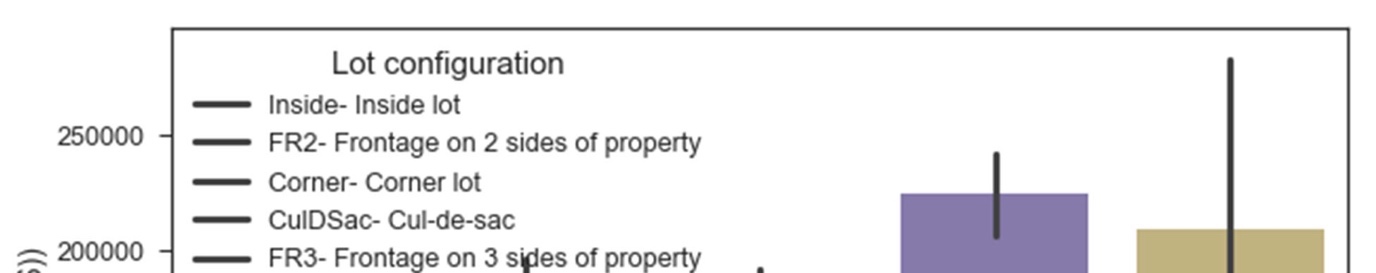
* -  'GarageCars' and 'GarageArea',
* -  'Total Basement square footage' and '1st floor square footage',
* -  'Above grade(ground) area' and 'Total no. of rooms above grade(ground) are highly correlated with each other.

The issue with Multicollinearity can be addressed through Machine Learning algorithms such as Ridge and Lasso Regression.

Other than that, the highly correlated independent variables with the target variable Sale Price are Overall Quality, Above Ground Living area and Garage cars.

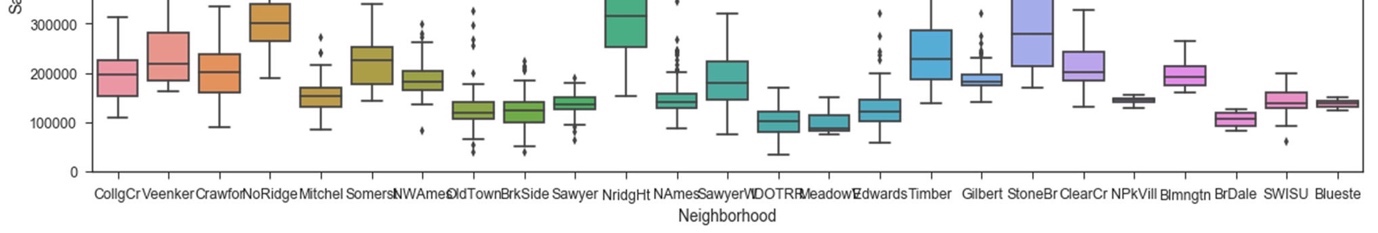
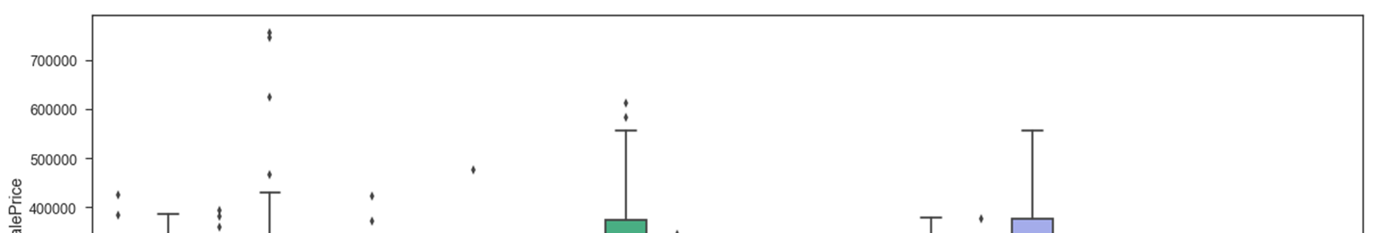


b. Some interesting questions:  
**1. What type of lots tend to have higher prices?**



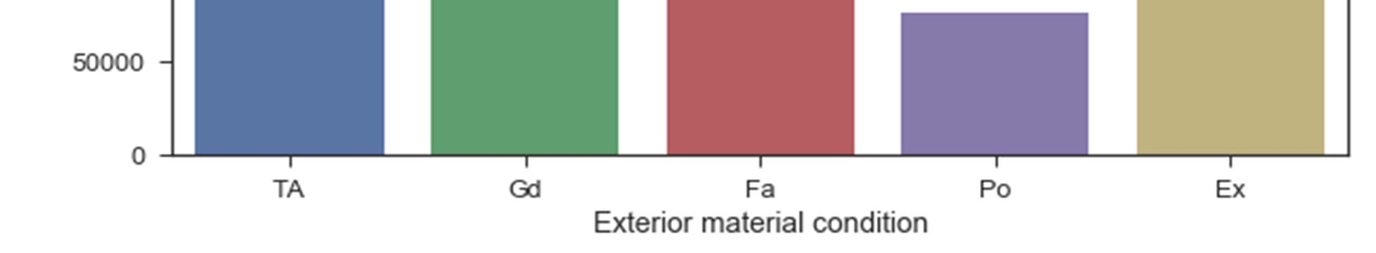
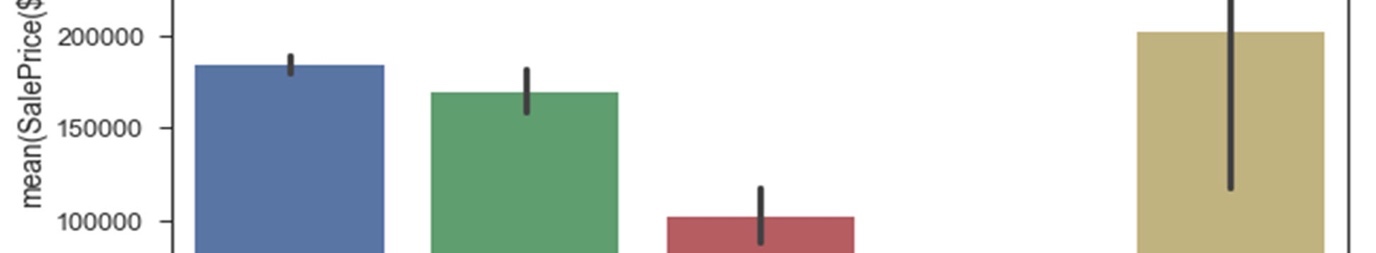
Cul-de-Sac lots tend to have higher prices followed by houses that have frontage on 3 sides of property. Cul-de-sac houses usually have more lot area, this might be a reason for a spike in a Cul-de-Sac site.

2. **Which neighborhoods are most and least expensive?**



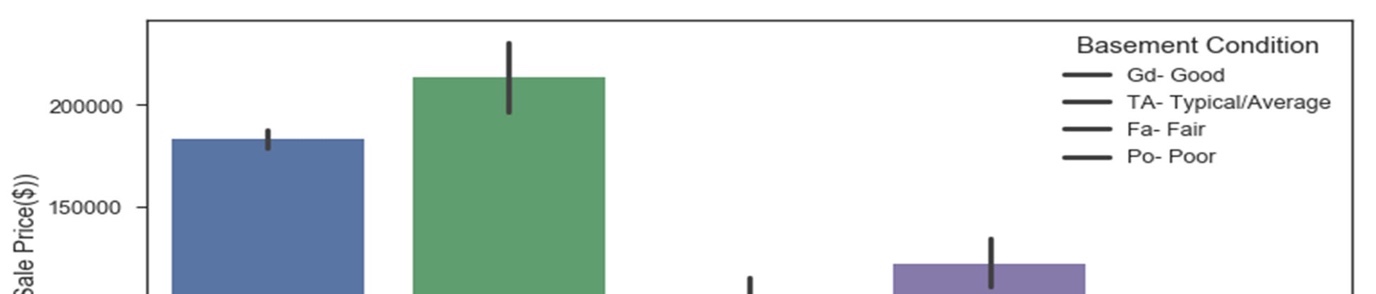
Northridge Heights and Stone Brook have the most expensive houses and Old Town, Brook Side, Sawyer, North Ames, Edwards, Iowa DOT and Rail Road, Meadow Village and Briardale are least priced houses among all the neighborhoods.

3. **Does external look of the house effect Sale Price?**



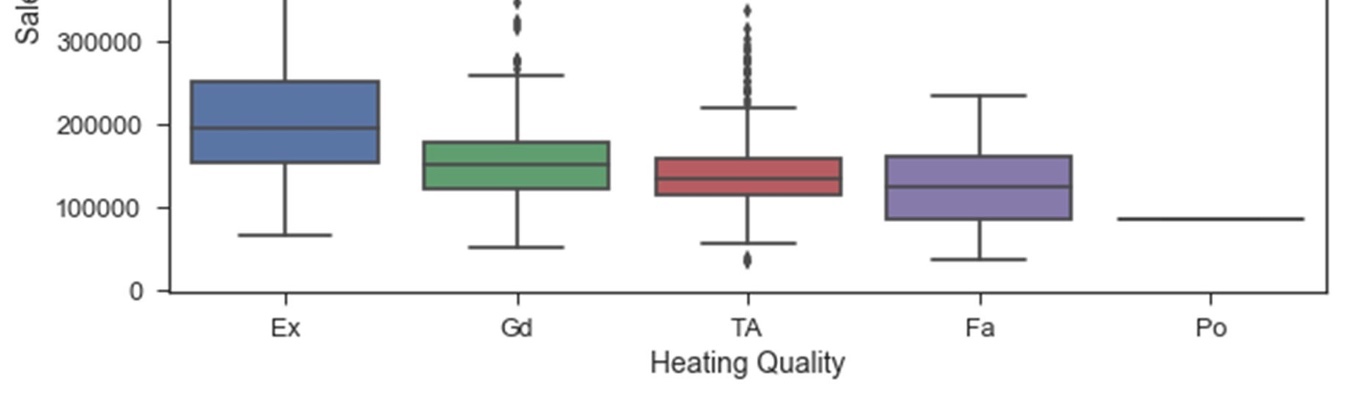
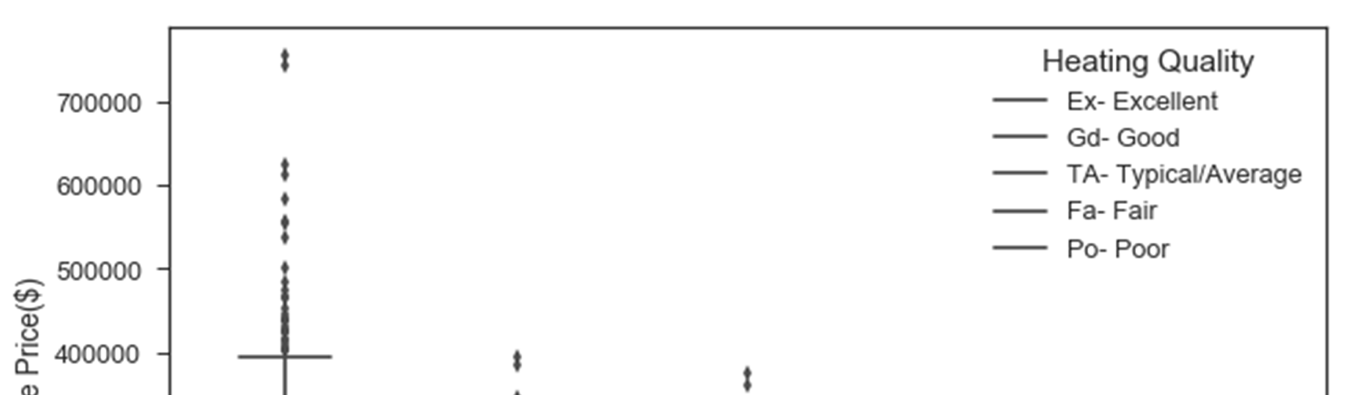
Looks like the exterior of the house is as important as the interior. The better the exterior quality the higher the house price is.

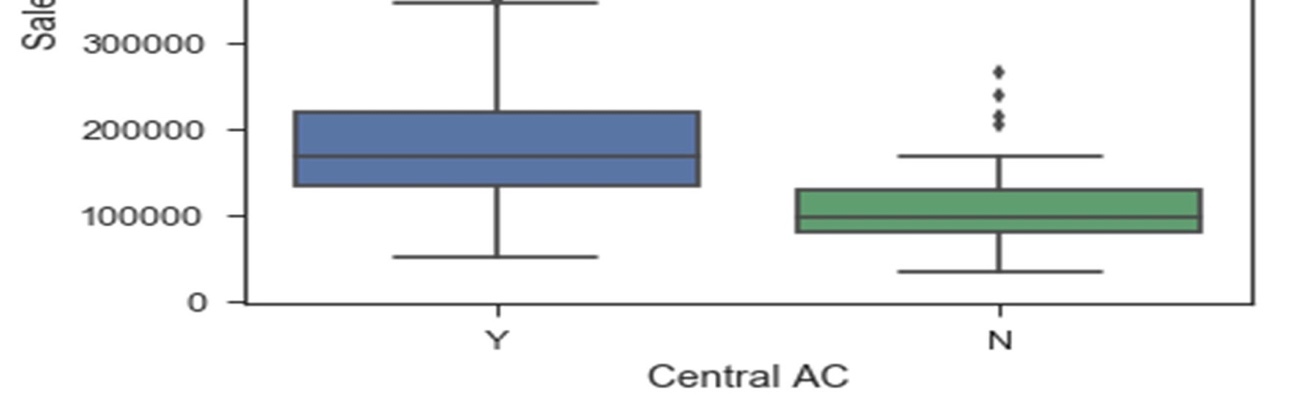
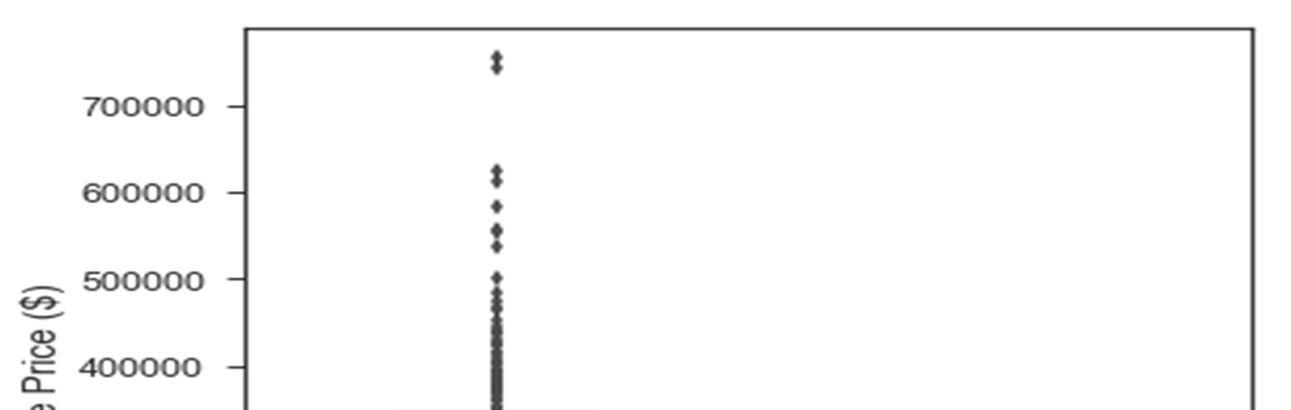
4. **What effect does Basement Condition have on house price?**



Basement condition has a linear effect on Sale Price, the better the quality of basement the more the price of the house.

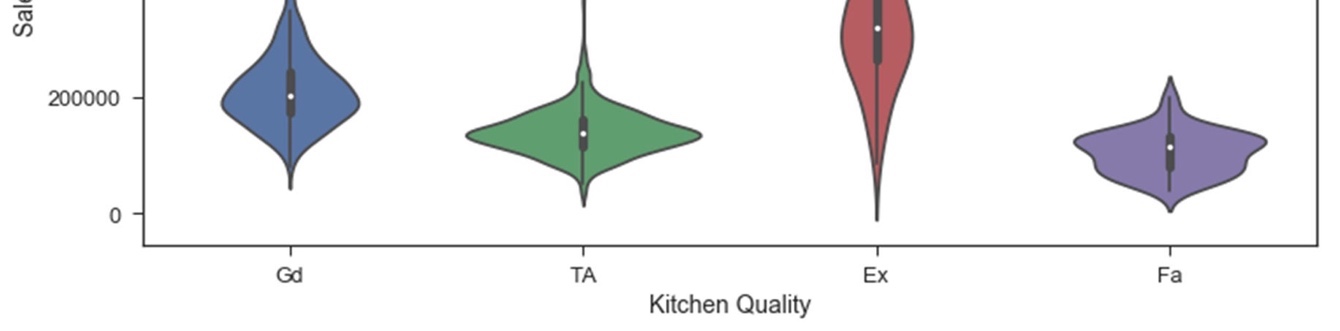
5. **What is the relationship between HVAC system and Sale Price?**





HVAC is one of the major component every house owner should consider before buying the house. HVAC has a positive correlation with Sale Price.

**6. How does Kitchen Quality effect the final Sale price of a house?**



Kitchen is the heart of the house. It is evident from the graph that an improvised/remodelled kitchen doesn’t come cheap.

7. Data Standardization

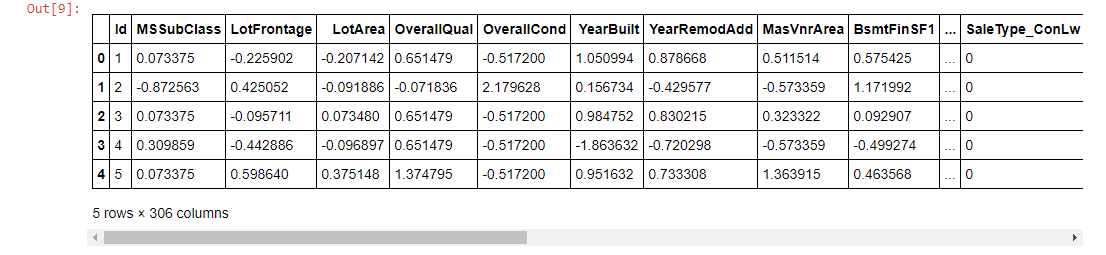
Before applying any Algorithms, it is extremely important to standardize the data. Data Standardization should be performed to make sure that all the features are on the same scale so that they can be compared for analyzing results. Data Standardization (or Z-score normalization) is the process where the features are rescaled so that they’ll have the properties of a standard normal distribution with μ=0 and σ=1, where μ is the mean (average) and σ is the standard deviation from the mean. I used functions from Scikit-learn to standardize the data.

8. Encoding CategoricalData

Regression Analysis only takes numerical data as input, the model doesn’t consider categorical data, because it is not possible to fit a least squares line with non-numerical data. Therefore, it is common practice in Machine Learning to transform the categorical data into numerical data. Scikit- learn offers two methods to achieve this task – Label Encoding and One Hot Encoding.

I used One Hot Encoding to convert the categorical data into binary form of representation. This resulted in enormous increase in the number of features from 81 to 306 features in the resultant matrix. With the data fully–– prepared the next step is to apply Machine Learning algorithms on data.

The data frame after performing Standardization and One Hot Encoding is below

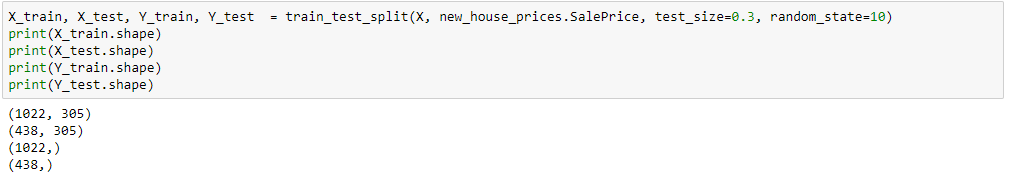


9. TrainandTestSets

Before applying ML algorithm, it is essential to split the data into train and test sets, so that there will be an untouched data set to assess the performance of the model. I split data the into train (70% of the entire data) and test (30% of the entire data).

X\_train – contains all the predictors of train data set Y\_train – the target variable in train set  
X-test – all predictors in test set  
Y\_test – target variable in test set

Note: Target Variable – ‘SalePrice’



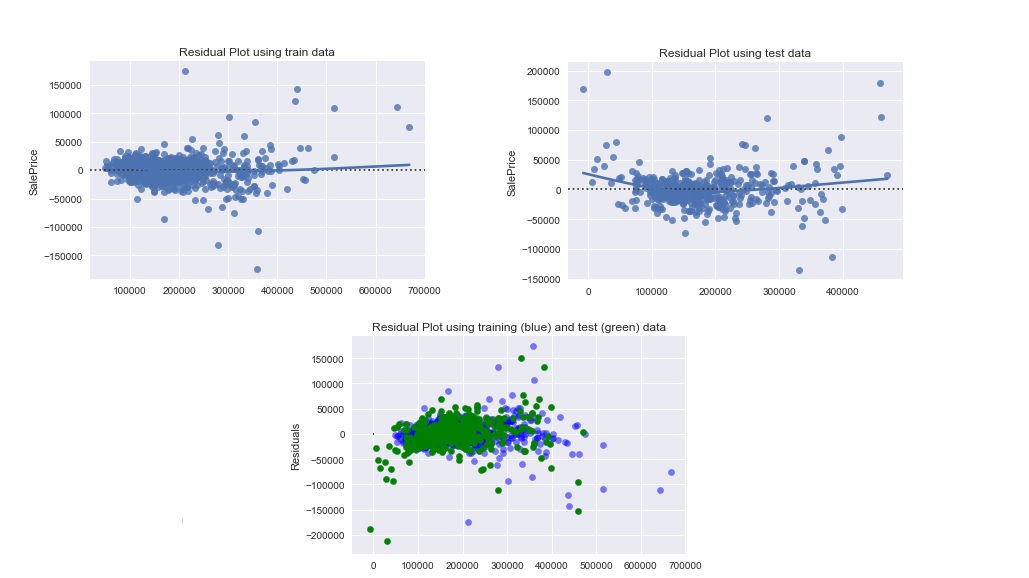
Notice that train data set is a matrix with all predictors and test data is a vector with only target variable.

10. Regression:

I performed Multiple Linear Regression first and then moved to more advanced algorithms. Regression plot plotted between the actual and predicted prices produced a good fit of a line for the data.

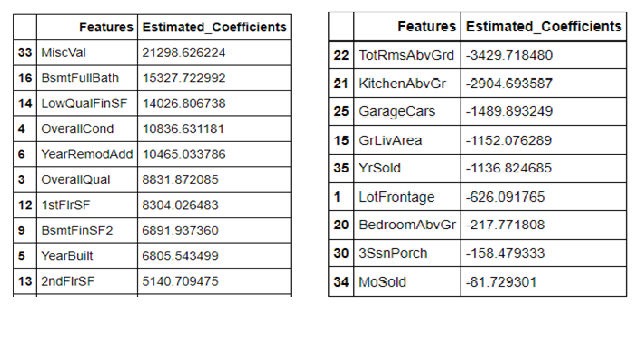


The analysis can be further strengthened by making residual plots. There are three different residual plots for train, test, train and test together. They all are surrounded along the reference line. The data range in between $ -50,000 and $ +50,000, this is very much comparable to real estate market. A house that has most the positively correlated features in a house will be at least $ 50,000 to $ 80,000 higher than the houses with negatively correlated features.



The regression analysis produced a bunch of positively and negatively correlated coefficients with the Sale Price. The top ten positive and negative coefficients are

Positive Coefficients Negative Coefficeints



11. Regularization

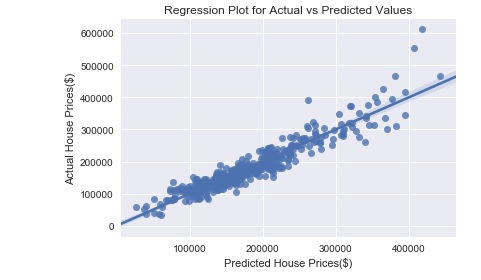
To overcome the problem of ‘Overfitting’ which usually occurs because the model learns the train data and noise in the data too hard Regularization is used. Regularization allows to shrink the coefficients to zero by introducing a tuning parameter 'lambda' or 'alpha'. This ensures:

* Shrinking of parameters, therefore it is mostly used to prevent multicollinearity.
* Reduces the model complexity by coefficient shrinkage.

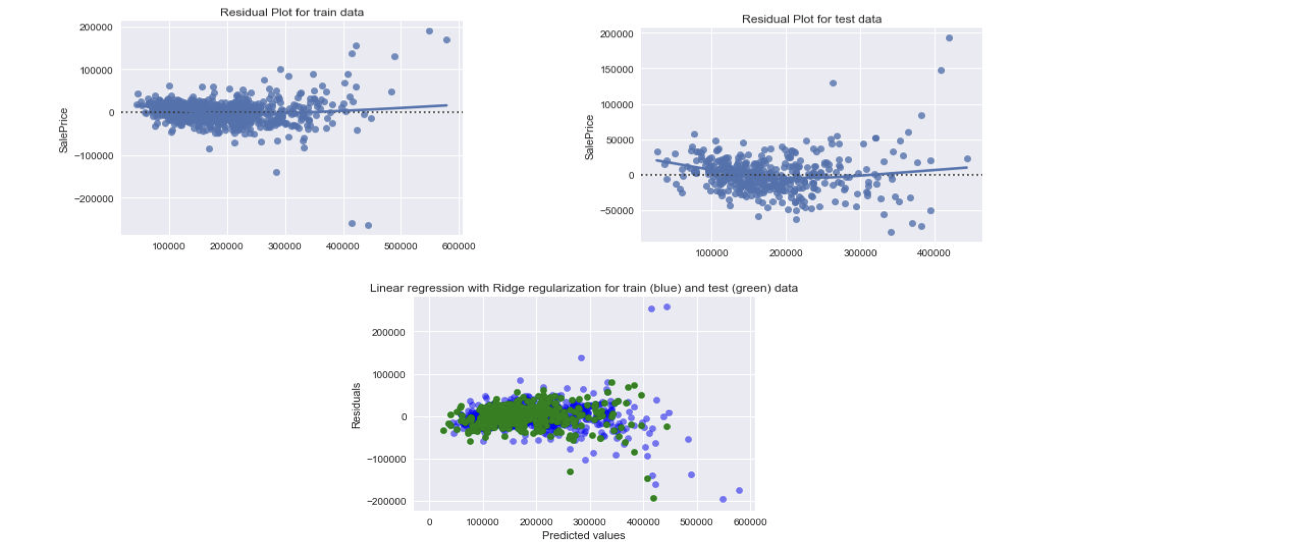
Ridge and Lasso Regression techniques are used in Regularization process.

**a. Ridge Regression:**

- Regression Plot:

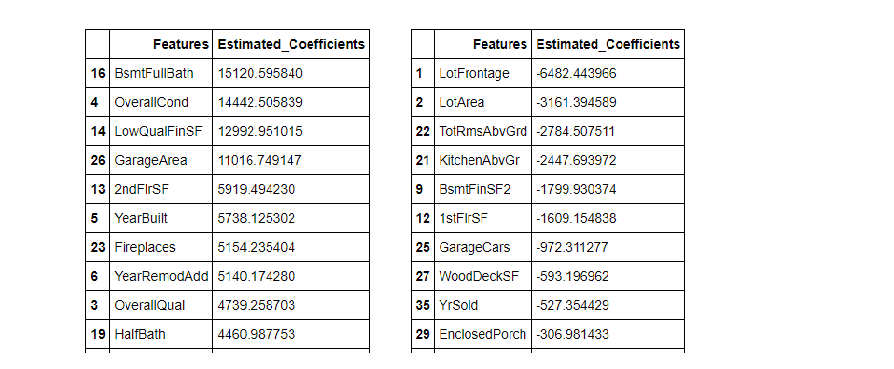


The least squares line looks to be a good fit for the data. - Residual Plots:



The graphs look similar to the Regression residual graphs. The plot representing Train and Test data tells that model is performing good on test data too.

- Coefficients:  
Positive Coefficients Negative Coefficients

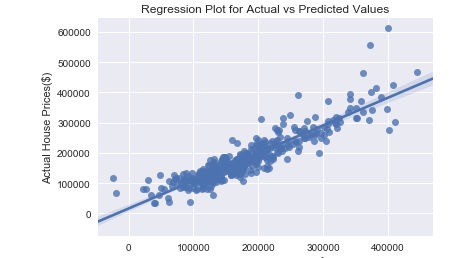


There is a change in the coefficients of features. A few features now are more positively/ negatively correlated with the target variable than in Multiple Regression.

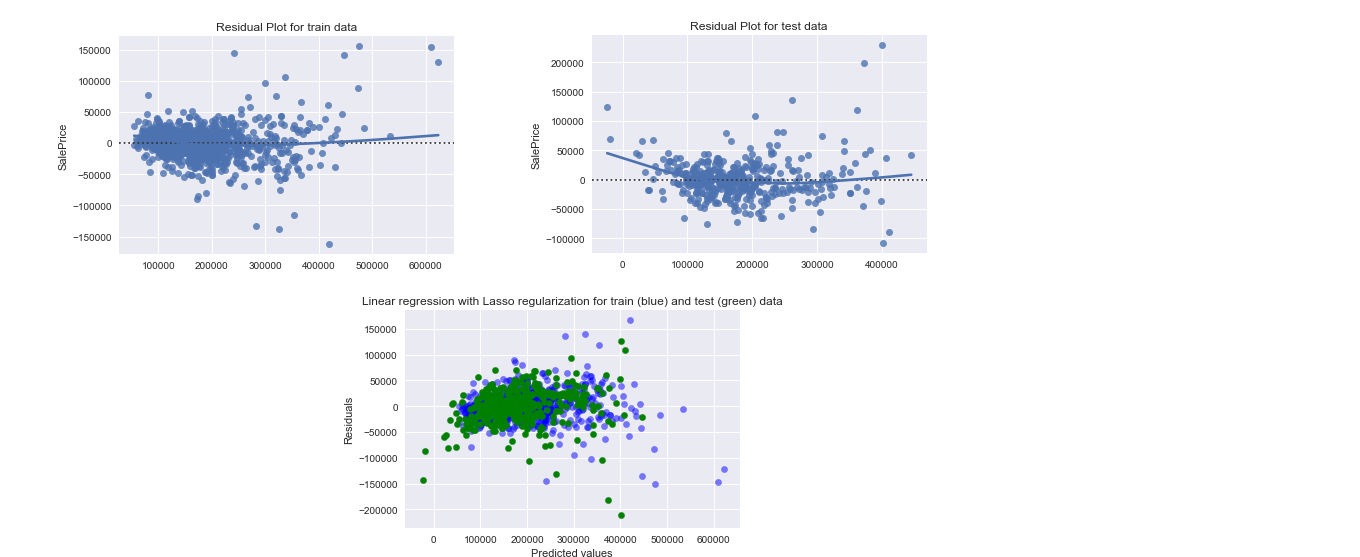
**b. Lasso Regression:** Lasso introduces a tuning parameter to shrink the coefficients to zero, this is an advantage over Ridge regression.

- Regression Plot:

From the plot above the regression line is a good fit for data

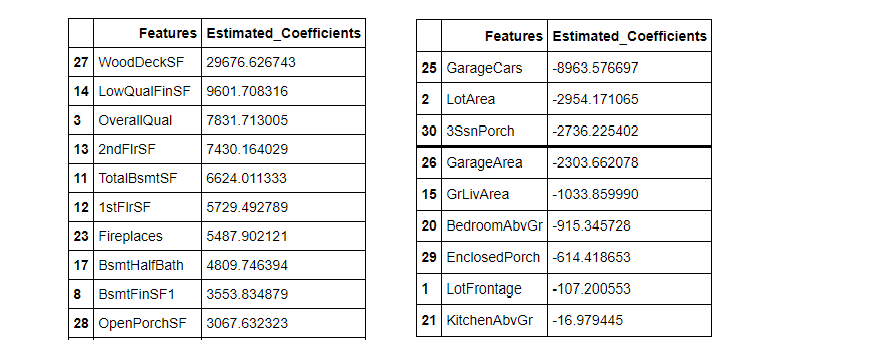


- Residual Plots:



The data is spread within $ -50,000 and $ +50,000 of the reference line. The test data is spread similarly as train data in the third plot. This is a sign that the model works well outside of the train data (test data).

- Coefficients  
 Positive Coefficients Negative Coefficients



There is a difference between in the coefficients when Lasso regression is applied on data.

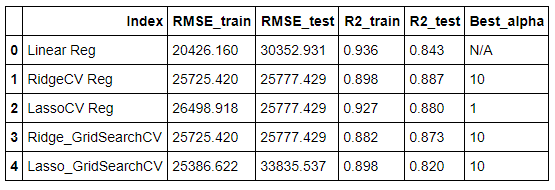
12. CrossValidation

When evaluating different hyperparameters for estimators, such as the alpha is this setting that must be manually set for an Ridge, there is still a risk of overfitting on the test set because the parameters can be tweaked until the estimator performs optimally. To solve this problem, yet another part of the dataset can be held out as a so-called “validation set”: training proceeds on the training set, after which evaluation is done on the validation set, and when the experiment seems to be successful, final evaluation can be done on the test set. GridSearchCV is used in Scikit-learn library to achieve this task.

13. Scores

* -  GridSearchCV with Ridge: Alpha values considered are - alphas = [1e-15, 1e-10, 1e-8, 1e-5,1e-4, 1e-3,1e-2, 1, 5, 10]
* -  GridSearchCV produced best alpha value as: 10 and scores R2 as 0.873 and RMSE as 25777.429.
* -  GridSearchCV with Lasso: Same alpha values are considered in Lasso too - alphas = [1e-15, 1e-10, 1e-8, 1e-5,1e-4, 1e-3,1e-2, 1, 5, 10]
* -  GridSearchCV produced best alpha value as: 10 and scores R2 as 0.812 and RMSE as 33835.537.

With these many models applied on data how can we conclude the best model for data. For this I compared the R2 and RMSE scores produced by all the models.



Comparing the train and test scores (R2 and RMSE), Ridge regression with Cross Validation (Ridge\_GridSearchCV) seems to best suited for the data, because there is not much difference between the scores of train and test data sets. The regression and residual plots from Ridge regression using Cross Validation also seem to be a good fit for data.

14. Summary

Houses with Full bath in Basement, Good condition, More Low quality finished area (sqft), Bigger Garage area (sqft), More Square footage in 2nd floor, lesser age, more number of fireplaces, recent remodeling, more number of Half baths above basement/ ground floor (for houses without basement) are priced high.

Houses with bigger front yard (more than back yard), bigger lot area, more number of rooms above basement/ ground floor (for houses without basement), Kitchen above ground floor, bigger finished square footage of second basement, bigger area in 1st floor (sqft), Garage capacity, bigger area in Wooden Deck, Year Sold and Enclosed Porch decreases the house price.

15. Recommendations

* Businesses/ home owners can quote a price for based on some of the important features such as the overall condition of the house and basement (if any), bigger garage, extra square footage in 2nd floor, recently built/ remodelled house, and more number of bathrooms.
* Since 96% of the predicted values range between -50,000 to 50,000 when compared to actual values, house prices in this location (Ames, Iowa) differ by $50,000, from the base price, based on the quality and features present in a house.
* Parties interested should decide the price of a house based on important features pointed out in the analysis that increase/ decrease the house price.