House Price Prediction Report

Problem Statement:

To predict the purchase price of each home according to the market prices considering different features ranging from the basic amenities of the house to that of its proximity to public transport.

Description:

Buyers are not aware of factors that influence the house prices. They believe that it depends upon the square foot area, neighborhood, number of bedrooms. However, it also depends upon area outside the house, rooms on one floor, etc. Therefore, it is important to predict housing prices without bias to help both the buyers and sellers make their decisions and a data science technique is required.

Data Wrangling:

The raw dataset from Ames Housing dataset contained 79 explanatory variables (features) describing every aspect of residential homes in Ames, Iowa and there are 1460 observations and Sale Price feature is the target variable.

The dataset contained int, float and object types of data as features along with some null values in between. There are some feature columns which had null values more than 80 per cent so I dropped those columns since they have least effect on the result. Outlier data points were looked at individually to determine whether the number was an incorrect entry or legitimate. The former was either corrected or made null while the latter were kept in the dataset.

Null values were filled based on the data type of the particular feature. Null values in categorical feature were filled with the most frequently occurring category and integer as well as float features were filled with mean value of the particular feature column. Hence the final shape of the dataset was 1460 rows and 76 columns.

Exploratory Data Analysis:

Since the dataset contains the many numerical features then it's better to know the distribution of those numerical values with the target variable. The following histogram (Figure 1) shows the distribution.

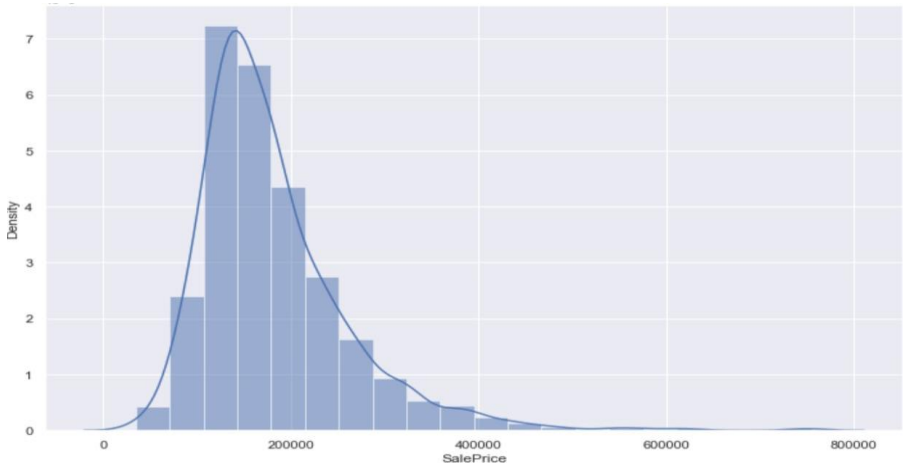


Figure 1: Distribution of numerical features with SalePrice

The above histogram is right skewed. This indicates there are some outliers. We can identify the outliers more precisely by the box plot. The following box plot (Figure 2) shows the outliers.

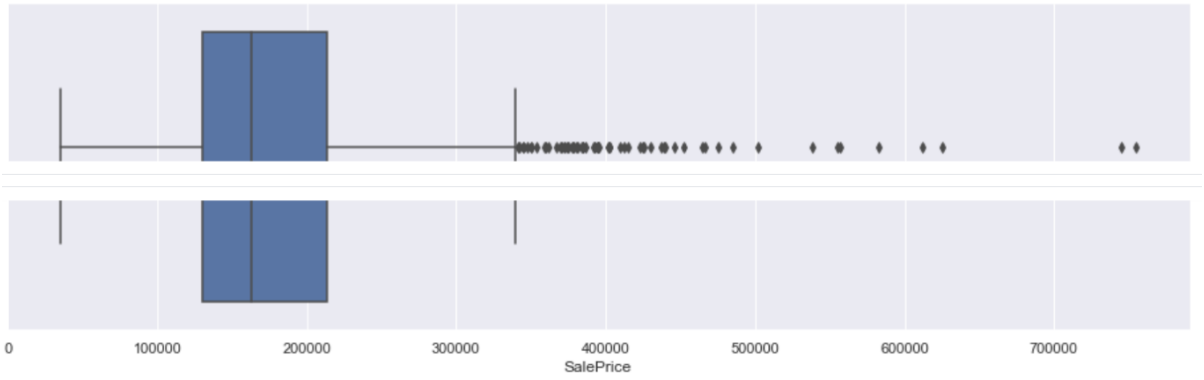


Figure 2: Outliers data points

Now to check the correlation between the independent features (76) and the target feature we plot the heatmap to identify positive or negative correlation. The following heatmap (Figure 3) shows that correlation.

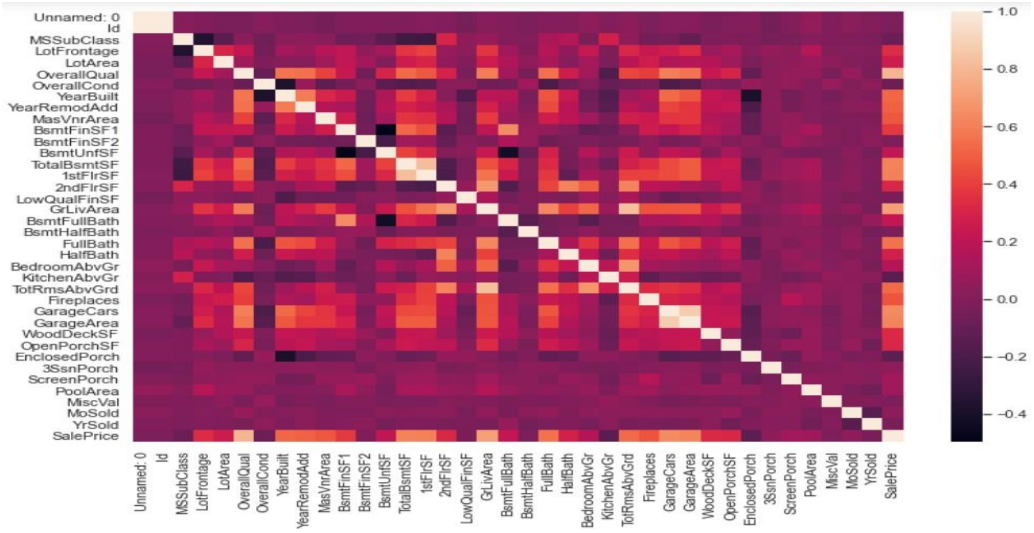


Figure 3: Heat Map to check the correlation.

The above heatmap shows the high positive correlation between OverallQual, GrLivArea, GarageCars, GarageArea, TotalBsmtSF, 1stFlrSF features and target feature SalePrice.
To get the clear picture about the correlation between the abovementioned features we plot the scatter plots between them. The following scatter pots (Figure 4) shows the positive correlation between the features.

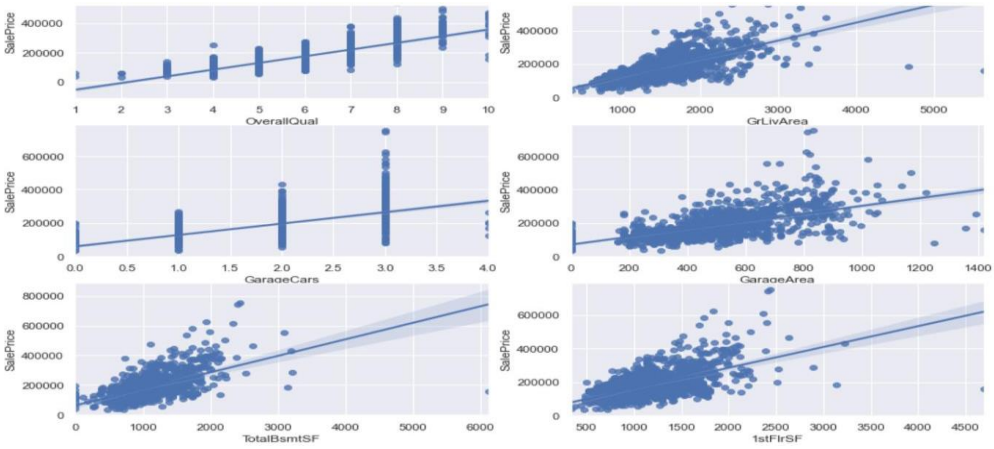


Figure 4: Scatter plots to find the correlations.

Initially it was felt that year feature (YearSold) is not affecting the target variable, but the following plot (Figure 5) shows its negative correlation on the target variable.

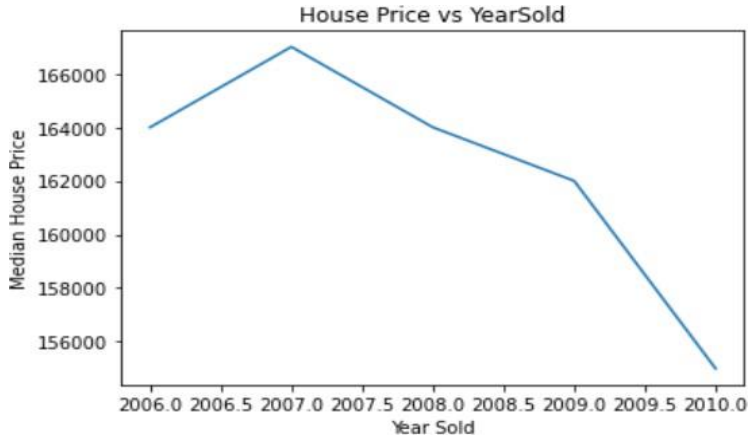


Figure 5

Feature Engineering:

In this step we filter the features and select only those features which have either positive or negative impact on target feature SalePrice. The features such as 'LotFrontage', 'LotArea', '1stFlrSF', 'GrLivArea', 'SalePrice' have skewed values (outliers) therefore converting them to log normal distribution. Similarly, YearBuilt and YearRemodAdd features are big number therefore converting them to small number by subtracting them from the YearSold feature.

The features which are correlated either positively or negatively on the target feature with the same amount (same value) then we are selecting any one among them. Hence here we dropped two more feature columns. We are left with 73 columns. Next step is to convert the categorical features into dummy features. We done this one with the help of dummies () function. It creates a separate column for each category in the feature with numerical values. Hence, we got 275 columns.

Data Normalization:

Data Normalization is a common practice in machine learning which consists of transforming numeric columns to a common scale. In machine learning, some feature values differ from others multiple times. The features with higher values will dominate the leaning process. However, it does not mean those variables are more important to predict the outcome of the model. Data normalization transforms multiscale data to the same scale. After normalization, all variables have a similar influence on the model, improving the stability and performance of the learning algorithm.

The dataset we use is usually split into training data and test data. The training set contains a known output, and the model learns on this data in order to be generalized to other data later on. We have the test dataset (or subset) in order to test our model's prediction on this subset. Except the target variable (SalePrice) consider the dataset as one variable called X and target variable as y. We do this using the Scikit-Learn library and specifically the train_test_split method.

Modeling:

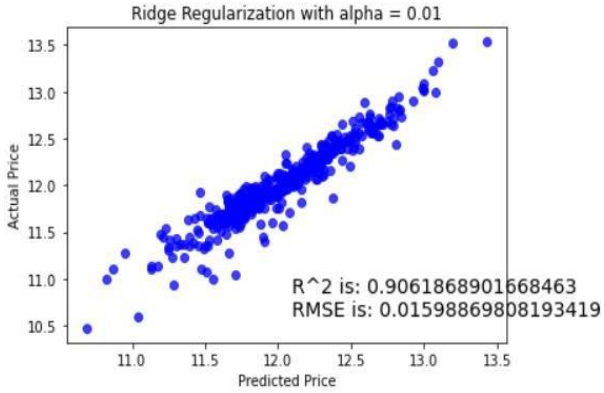
Here we are implementing three different regression models.

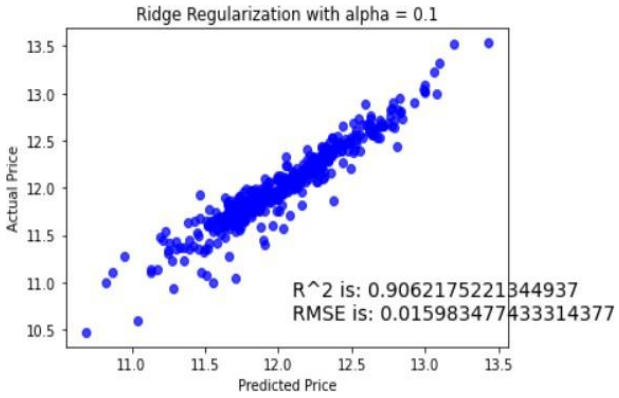
Model 1 (Linear regression):

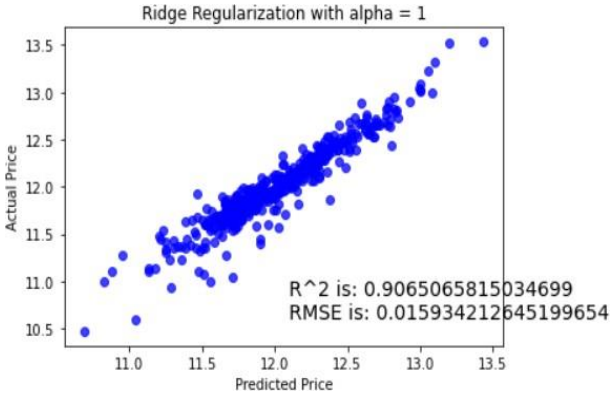
R^2 = -3.566569324628013e+19 RMSE = 6.078553436847325e+18

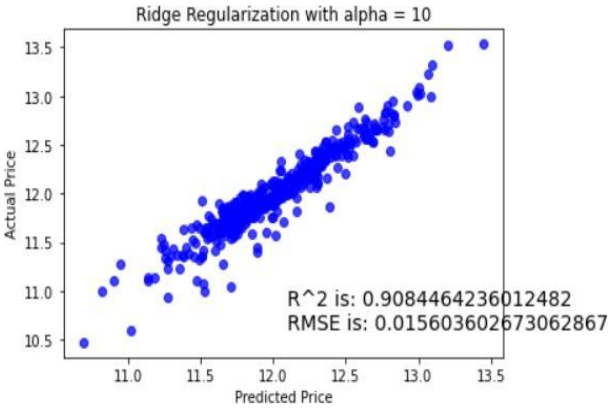
Model 2 (Ridge Regression):

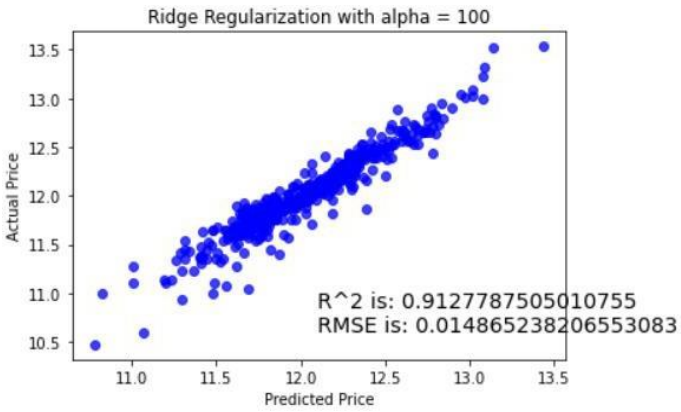
For five alpha values 0.01, 0.1, 1, 10, 100 we got the following results respectively











Model 3 (Lasso GridSearchCV):
R^2 is: 0.9086984269579725
RMSE is: 0.015560653392376344

Conclusion and Future Scope:
We have used machine learning algorithms to predict the house prices. We have mentioned the step-by-step procedure to analyze the dataset and finding the correlation between the parameters. Thus, we can select the parameters which are not correlated to each other and are independent in nature. These feature set were then given as an input to three algorithms and observed their performance. Hence, we calculated the performance of each model using RMSE metric and compared them and ridge regression came out with high percentage of success hence we selected that as a final model.
For future work, we recommend that working on large dataset would yield a better and real picture about the model. We have undertaken only few Machine Learning algorithms that are classifiers, but we need to train many other classifiers and understand their predicting behavior. By improving the error values these models can be useful for development of applications for various respective cities.