

Housing Price Prediction

Submitted by:

Anusha K

**INTRODUCTION**

Houses are one of the necessary needs of each and every person around the globe and therefore housing and real estate market is one of the markets which is one of the major contributors in the world’s economy. It is a very large market and there are various companies working in the domain. Data science comes as a very important tool to solve problems in the domain to help the companies increase their overall revenue, profits, improving their marketing strategies and focusing on changing trends in house sales and purchases. Predictive modelling, Market mix modelling, recommendation systems are some of the machine learning techniques used for achieving the business goals for housing companies. Our problem is related to one such housing company.

A US-based housing company named Surprise Housing has decided to enter the Australian market. The company uses data analytics to purchase houses at a price below their actual values and flip them at a higher price. For the same purpose, the company has collected a data set from the sale of houses in Australia.

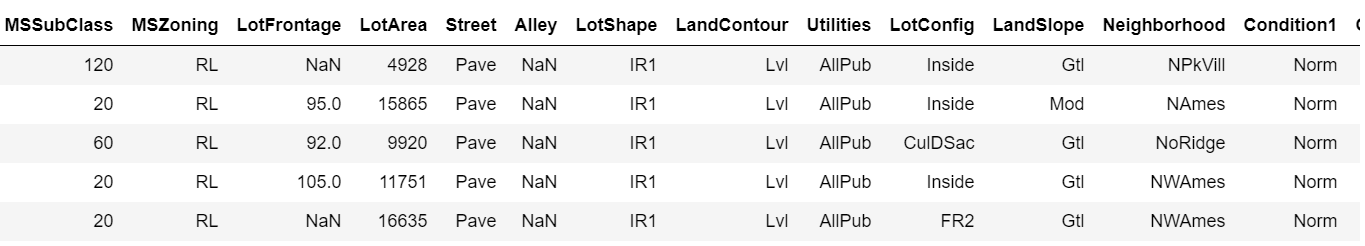
Some of the factors that are directly affect the house price are age of the house, location of the house, how many floors does the house have, water/electricity connections, Parking space availability, construction material, transportation, security and many more. Let’s analyse the data and build the predictive model to predict the price of a house.

.

**Analytical Problem Framing**

For this project, the data is provided by the client database. The data set has 81 variables, SalePrice is our target variable. The dataset consists of both numeric and object data types. The object type variables are of categorical variables. Numeric variable contains both continuous and discrete values. We will have to separate these in

# The sample snapshot of the data is given below:



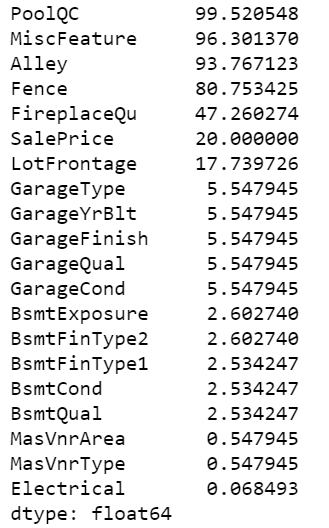
**Data Pre-processing:**

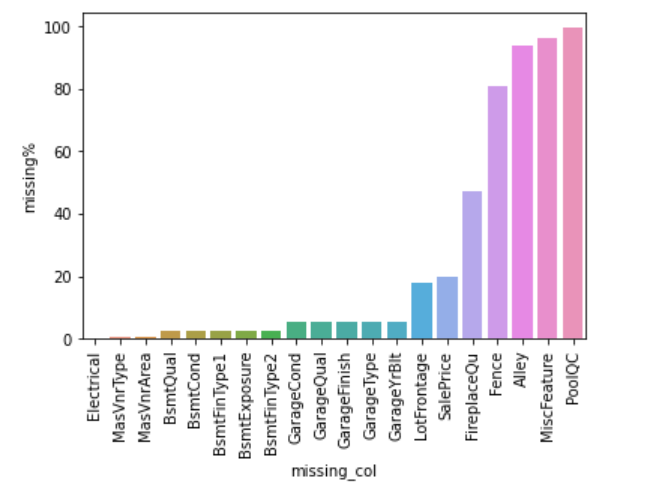
Describe method provides us the statistical summary of the data, we can identify whether the dataset has null values, Outliers or the data is skewed.



From the above stats we can infer that the data is skewed and there are outliers in some of the columns. Also, the dataset has Null values.

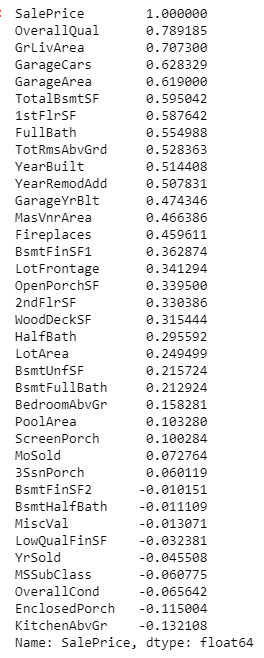
Out of 80 feature columns, 19 columns have the null values. Below table shows the percentage of the missing values of each column.





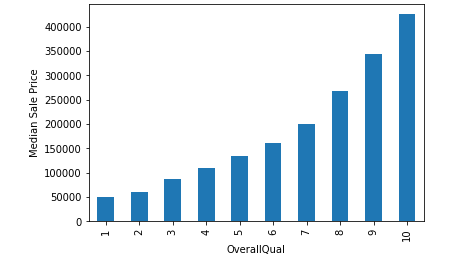
From the above bar chart, we can clearly see the highest number of missing values from the column PoolQC followed by MiscFeature, Alley, Fence, FireplaceQu and so on. These missing values will be treated during the feature engineering.

Now the data must be divided into numerical and object type data to understand the features and to identify the correlation with the target variable.  
Below table shows the numeric variable and the correlation percentage with the target variable SalePrice:



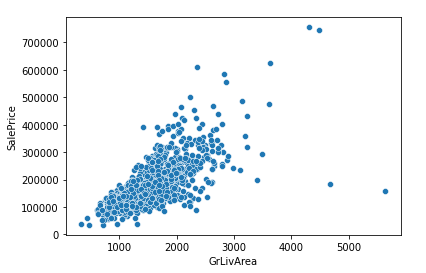
OverallQual, GrLivArea are highly correlated with the saleprice that is 79% and 71% respectively. Well, this makes sense. OverallQual is nothing but the overall quality of the house, everyone will look into this feature when they buy one. GrLivArea is Above grade (ground) living area square feet. GarageCars and GarageArea are Size of garage in car capacity and Size of garage in square feet, the correlation is 63% and 62% respectively. TotalBsmtSF is Total square feet of basement area which has 60% correlation with the dependent variable.

The OverallQual has 10 rating values, 5 is most frequest value follwed by 6,7,8. These are Average, above average, good and very good rating values. 9 and 10 appears less frequently being excellent and very excellent due to the higher cose of the houses.1,2,3,4 are poor and below average rated houses which are sold very less due to the quality of the house.



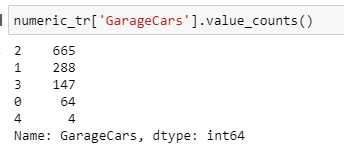
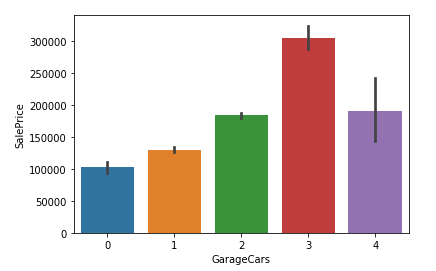
Here we have chosen median values of SalePrice as it is robust to outliers and the skewness. We can clearly see as the quality of the house increases, the price is also increasing.

'GrLivArea' is a continuous variable, we can plot a scatter plot to visualize the trend with the SalePrice.



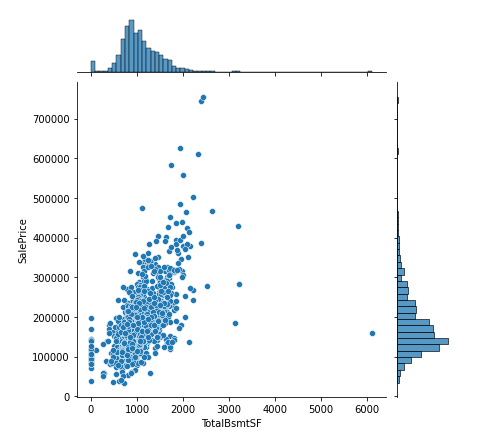
We can see, there is a direct correlation with the GrLivArea, most of the data points lie between the 1k to 2k also there are some outliers above 4000. we will have to treat these outliers as it will affect our model's performance.

GarageCars is the car capacity in the Garage, this variable has more than 60% correlation with the target variable SalePrice.



3 car capacity garages is costlier compared to others even though the 2 car capacity was sold higher followed by 1 and 3. 0 and 4 are much rare. value 2 and 1 comes with average price. value 3 comes with higher price. 4 car size is brought very rare; hence the price is bit lower.

TotalBsmtSF is the Total sq ft of basement area. The density is higher when it is from 500 sq ft to 1500 sq ft. Here also we see 3 outliers, 1 which is greater than 6000 sq ft and other 2 has more than 700000 SalePrice, will be handled later.



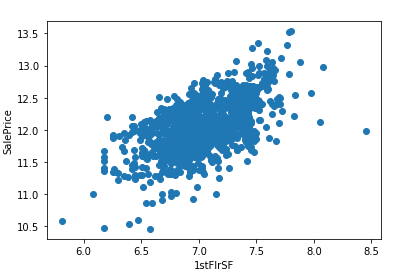
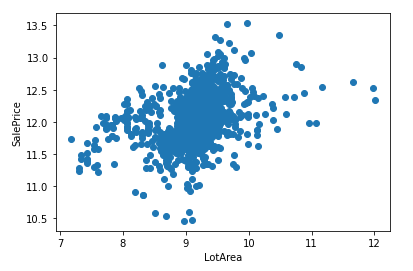
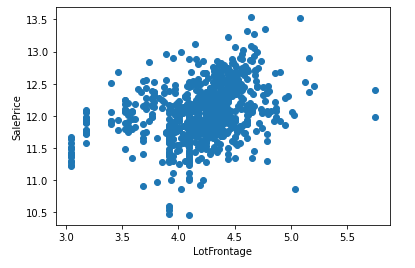
As the Total basement area increase the price also increasing. These all are the main aspects from continuous variables.

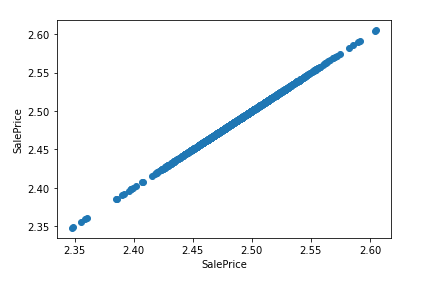
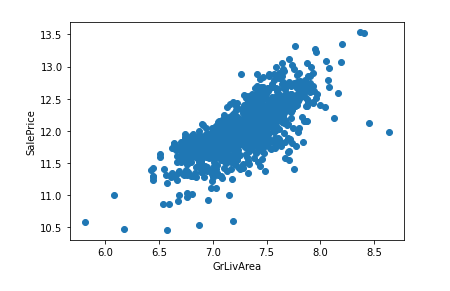
Let’s now look into the continuous data which has discrete or categorical values.



There are 13 variables which has less than or equal to 15 categorical values. These all will alter the SalePrice positively.

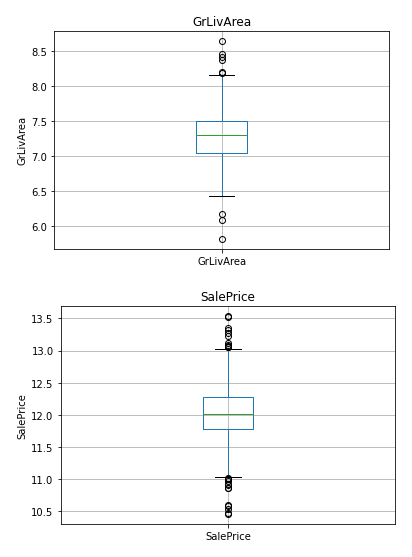
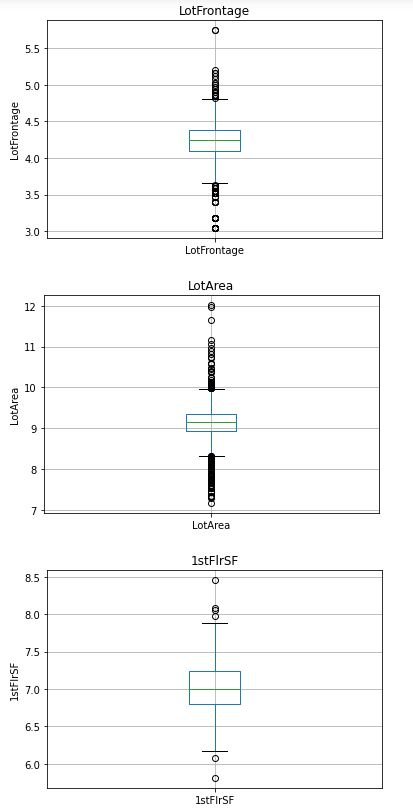
Many of the numerical variables are right skewed data, hence lets apply logarithmic transformation on the numerical variables and try to remove the skewness of the data.





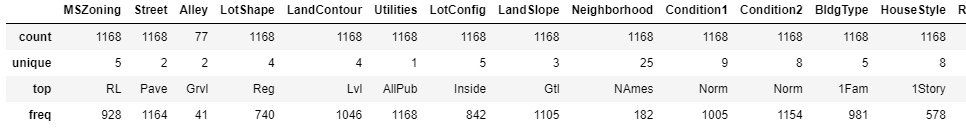
This is how they appear after the log transformation.

Outliers Detection: Again, there are outliers in the numerical variables.  
We can plot boxplot to identify the outliers in the numerical dataset.



Here we can clearly see the outliers and these needs to be treated well.

Let’s now focus on the categorical variables. There are 43 categorical columns, and these can be described as below:

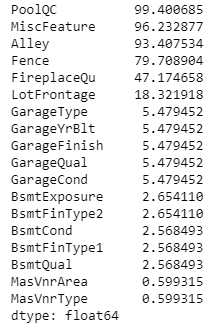


By describing the categorical variables, we can now see the number of unique categories, top category, and the frequency. In general, when we observe the categorical trend with SalePrice, wherever the categorical value count is more, the SalePrice is affordable.

**Feature Engineering:**

Let’s apply the feature engineering techniques one by one separately to train data set and test data set to avoid data leakage.

Initially the missing values must be treated. Below is the table shows missing percentage. This will give an insight on how we can treat these missing values.



Let’s not consider PoolQC, MiscFeature, Alley and Fence as they have more that 80% of missing values. But let’s replace all the missing values in these 4 variables as 'missing' to identify the missing values.

All the missing values are filled by the appropriate values by understanding the data and the problem.

Once the missing values are treated well, the skewness of the numerical variables must be transformed using logarithmic transformation to normalize the data.

Also, the categorical object type data must be encoded and standardised. We have used LabelEncoder library to encode all the categorical values.

The data range will look very large; hence we will use Standard Scaler technique to scale our data and bring all the data points to one scale.

**Feature Selection:**

In feature selection technique, the highly correlated and highly influenced columns will be extracted to get the best performance of the model.

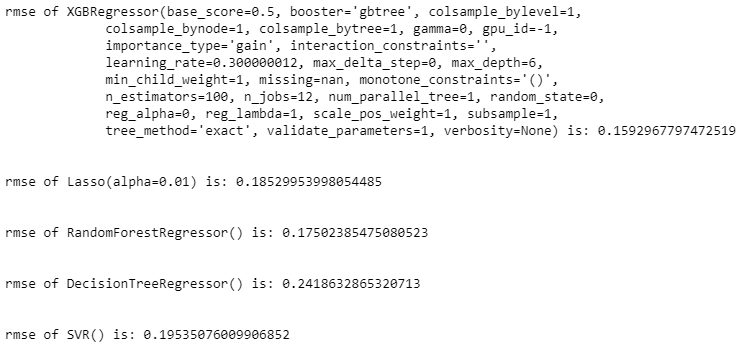
Here we are using feature\_selection.SelectFromModel library to select the features. This is a Meta-transformer for selecting the features based on the importance weights.

Using the Lasso Regression and with alpha value 0.01, we have got 30 features that affects the SalePrice.

As the alpha value increased the number of selected features will be reduced. We have finalised the Lasso features as our final features for model building.

**Model Training and Evaluation:**

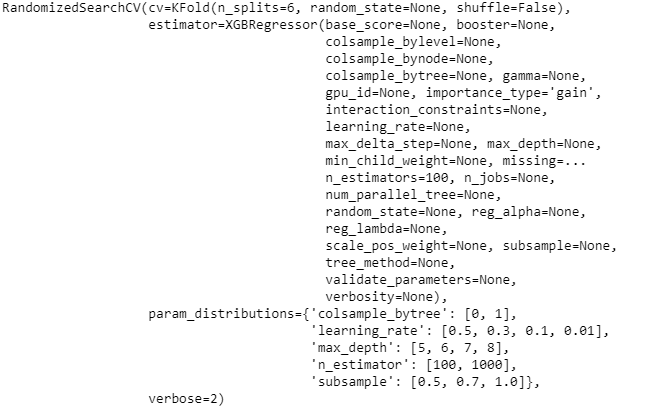
Now our data is ready and its time to train the model. Here we have considered XGBoost, LassoRegression, RandomForestRegressor, decision TreeRegressor and SVR algorithms.



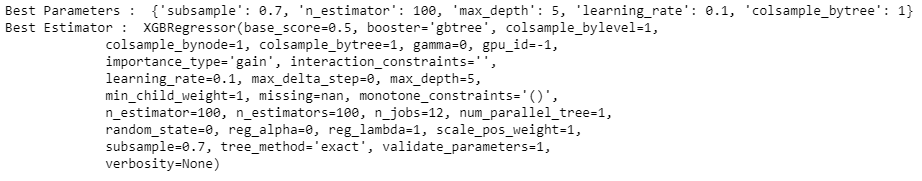
Above screenshot we can see the root mean squared error value for each of the algorithms. We found the XGBoost algorithm has lowest RMSE value which is 0.159%. Therefore, we can now finalize the XGBoost algorithm as our final algorithm.

**Hyper Parameter tuning and Cross Validation:**

With the below parameters and the cross validation, we were able to get the rmse score to 0.157%.



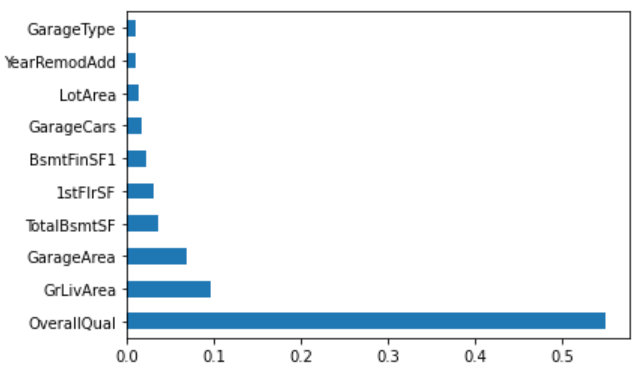
Below are the Best Parameters and Best Estimators:



Tried multiple cross validation and at cv=8 the RMSE value was very less compared to other cv values.

**CONCLUSION**

When we look at the feature importance of the dataset, we see that OverallQual contributed the most towards the best accuracy followed by GrLivArea, GarageArea and TotalBsmtSF and so on.



Above graph shows the important features that affect the house price directly.

XGBoost is a powerull approach for building supervised Regression Models. RMSE is one of the key members of XGBoost models. This model works best on this dataset with a minimal root mean squared error of 0.157. XGBoost expects to have the base learners which are uniformly bad at the remainder so that when all the predictions are combined, bad predictions cancels out and better one sums up to form final good predictions.