**Housing Price Predictions Using Advanced Regression Techniques**

Akash IndaniHarsh Barot

**Declaration**

We certify that:

The submitted work is me and my teammates' original work and not copied from the work of someone else.

Each use of existing work of others in the submitted is cited with proper reference.

Name: Akash Indani

Date: 08/12/19

Name: Harsh Barot

Date: 08/12/19

**Problem Statement**

Ask a home buyer to describe their dream house, and they probably won't begin with the height of the basement ceiling or the proximity to an east-west railroad. But this playground competition's dataset proves that much more influences price negotiations than the number of bedrooms or a white-picket fence.

With 79 explanatory variables describing (almost) every aspect of residential homes in Ames, Iowa, it’s a good challenge to predict the final price of each home.

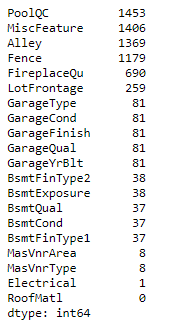
**Introduction**

The data set used in this project describes the sale of individual residential property in Ames, Iowa from 2006 to 2010. The data set contains 1460 observations and a large number of explanatory variables (23 nominal, 23 ordinal, 14 discrete, and 20 continuous) involved in assessing home values.

We have implemented linear and advanced regression techniques like Linear Regression, Random Forest Regressor, Lasso, Gradient Boosting and SVM regression to achieve more precise and accurate predictions with multiple independent variables. Also, we have compared the predicted results with actual prices to find out which model provides the best accuracy by using various accuracy measurement factors. The goal of this project is to create a regression model that can accurately estimate the price of the house given the features.

**Preliminary Data Analysis**

Before moving towards the data analysis let’s draft some information about the data. As mentioned above we have around 1460 observations of house sales prices. First, let’s check the missing values in the data.



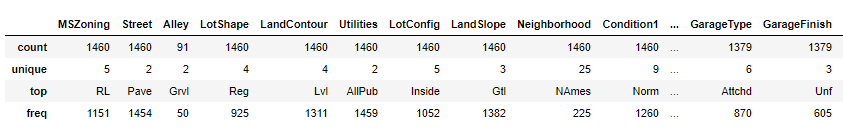
Null Values

We have handled all these attributes by imputing default values which is described at later stage. We have also removed features who has more than 70% null values as they are not helpful in predicting the Sales Price.

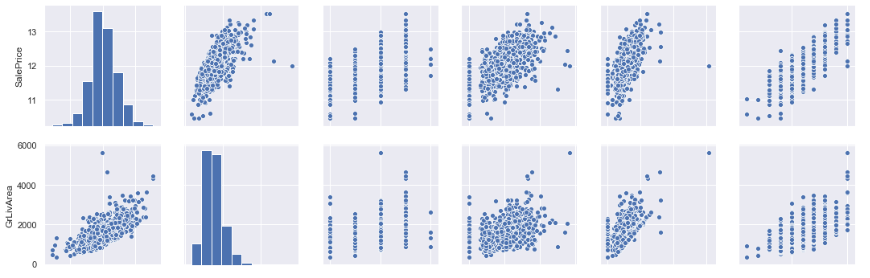
The following table shows the statistics for all the numerical features.

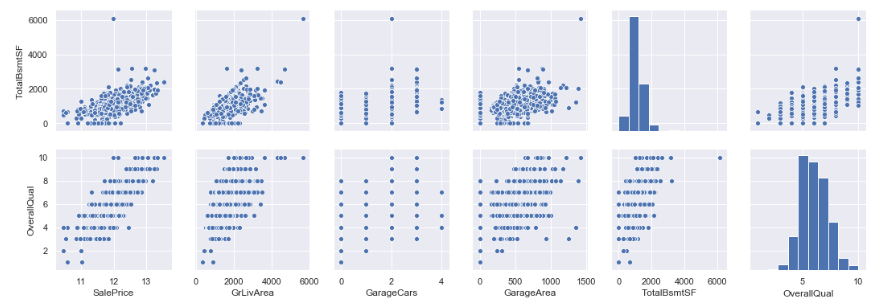


The following table shows the statistics for all the categorical features.

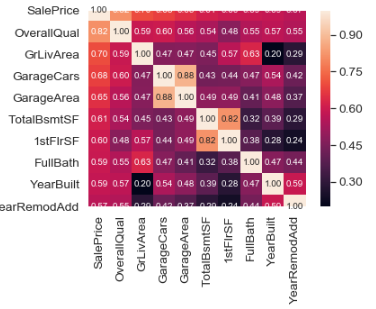


Next, we conduct bivariate analysis to find out outliers in this data set as our final goal is to predict the sale price of given house. To see the outliers visually, we can use the Seaborn library for Python to draw different plots. The following figure shows the plot on a selection of features: price, garage, bathrooms, bedrooms and overall quality. We can see clearly, there exist one or more outliers in these features. In addition, we can see the general trends for price over different features.

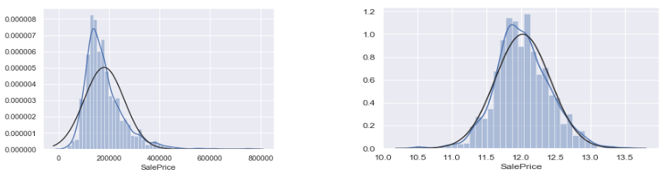




We have also plotted heatmap to understand the correlation among important features with Sales Price.



Skewness is another problem we are concerned about and we have handled it using log transformation. In the below figure you can see the skewness in the dependent variable Sales Price which had to be normally distributed before using it in our models.



Similarly, we have also transformed remaining numeric attributes which is covered in the Data Preprocessing section.

**Data Preprocessing**

1. Imputing Missing Values:- Missing values in each of the attribute is handled differently by first differentiating it into quantitative and qualitative fields and then imputing None, Mean, Median, Mode and 0. Before imputing with above values we did a thorough understanding of each feature and then finalized on the values.



1. Clean Outliers: - Using scatter plot we identified outliers in some of the attributes and looking at the plots we computed the constraints.



These outliers were removed from the dataset to make it as clean as possible which can help us build more accurate models.

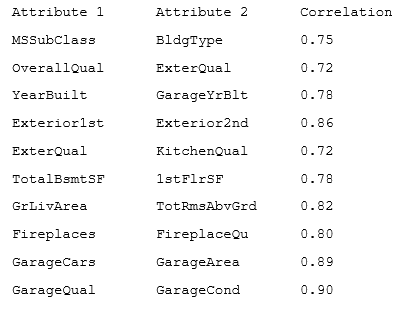
1. Transform Skewed Attributes: - As we transformed Sales Price shown above we transformed all the attributes whose skewness was more than 0.75.



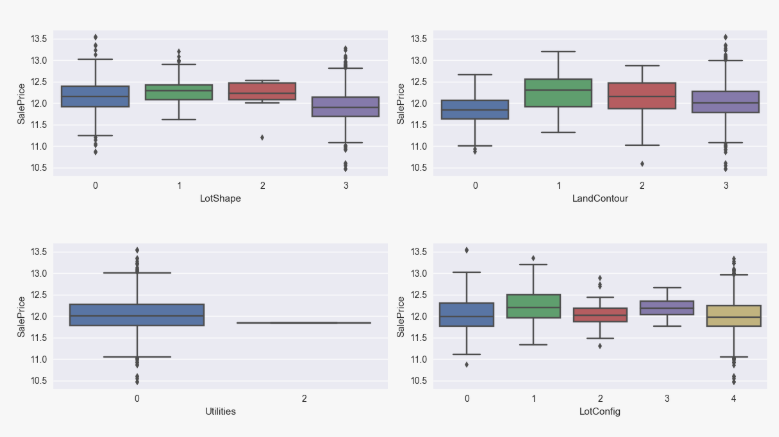
**Feature Selection**

1. Correlation Numerical Attributes: - Now let's think of the correlations of remaining attributes. The following code cell tells us attributes pairs whose correlation values are more than 0.7.

Therefore, we dropped attributes: BldgType, ExterQual, GarageYrBlt, Exterior2nd, KitchenQual, 1stFlrSF, TotRmsAbvGrd, FireplaceQu, GarageCars, and GarageCond.



1. Categorical Attributes: - Even though we applied log transformation on numeric attributes, the above scatter plot demonstrates that some attributes will perform bad in training, because they are not normal distributed even after transformation. In addition, we also plot the relationship among categorical attributes and 'SalePrice' as shown in the following figure for couple of the attributes.



1. Backward Elimination: - Backward elimination is considering all the features and keep on deleting one max p-value column on each iteration. We are using hypothesis testing for finding the relevant columns using backward elimination as shown below

H0: There is no relation between Y and X

Ha: Y independent on X

Now, if p-value is less than significance level then reject null hypothesis (i.e. if p-value > 0.05 we do need the columns)

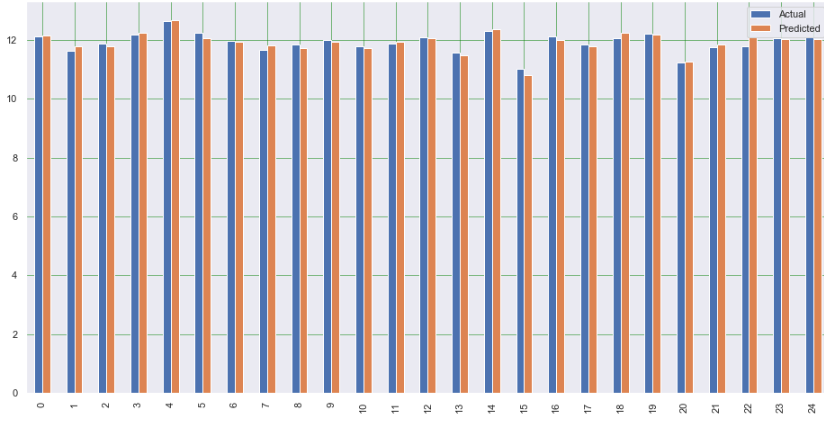
1. Lasso Feature Importance: - LASSO provides a principled way to reduce the number of features in a model. LASSO involves a penalty factor that determines how many features are retained; using cross-validation to choose the penalty factor helps assure that the model will generalize well to future data samples. If you need to cut down on the number of predictors for practical reasons, LASSO is a good choice. In the below figure we found out that out of 79 attributes at the beginning of the data preprocessing Lasso has come up to 8 most important features for predicting Sales Price which is quite a drastic reduction in the number of features.



**Implementing Models & Comparing Results**

After finishing our preprocessing we came up with a set of most important features that can now be used in our learning models. As mentioned in the introduction we implemented linear and advanced regression techniques like Linear Regression, Random Forest Regressor, Lasso, Gradient Boosting and SVM regression. We found out that Linear, Random Forest and Gradient Boosting performed well with R2\_score > 0.85. On the other end SVM regression performed bad.

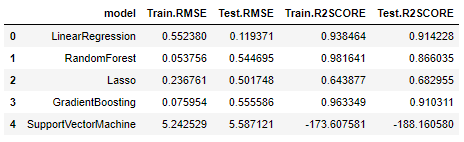
We have also plotted the difference between the actual and predicted Sales Price for all the models.



Comparison between actual and predicted Sales Price (Linear regression)

**Conclusion:**

Please find below the dataframe which has performance statistics for each of the learning models. We have computed Mean Squared Error (MSE), Root Mean Squared Error (RMSE), R2 Score for training and test data. Algorithm like gradient boosting seem to be the best algorithm suitable on dataset as the R2 value is above 90% for both training as well as testing.



As we can see, we are getting very high R2 scores which define the efficiency of the models. One of the reasons we are getting such high R2 scores which thus results in accurate predictions is because we only had around 1500 records to deal with. If we had a greater number of records the accuracy number might had changed which would have created a stable model.

**References**

We have referred mainly below online sources to understand the concepts and reference code:

1. <https://towardsdatascience.com/>
2. <https://stackoverflow.com/>
3. <https://www.kaggle.com/>