

|  |
| --- |
| AI Algorithms |
|  |
| December 18,2020  House Prices Prediction  Submitted by: Muhammad Imran (100805394) |

Executive Summary

Artificial intelligence has become the breakout technology, utilizing huge amounts of computing power to learn and identify patterns in data.  Real estate industry has been driven by personal preferences and human interactions, primarily among buyers, sellers, and real estate agents. But now, changes to the buyer-agent-seller paradigm are happening. Real estate companies adopting Artificial Intelligence will be critical to sustaining and enhancing their competitive advantage to grow in this rapidly evolving industry.

Rationale Statement

Property price influences many different transactions such as sales and loan. Traditionally, property estimate is determined by professional. The risk is that human tend to be biased due to interest from lender, buyer, or seller. Therefore, a predicted system based on Artificial intelligence can serve as an independent and less biased system. For first time or less experienced buyers of real estate properties, an automated price prediction system can be useful to suggest underpriced or overpriced properties in the market.

## **Data Acquisition**

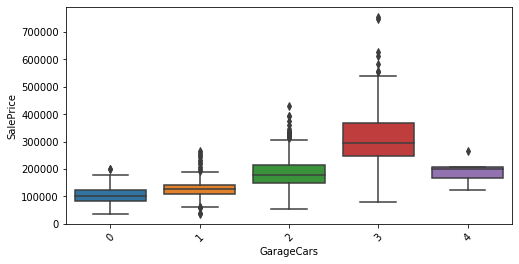
Data has been collected from Kaggle competition which is based on [Ames Housing dataset](http://www.amstat.org/publications/jse/v19n3/decock.pdf) , it was compiled by Dean De Cock for use in data science education.

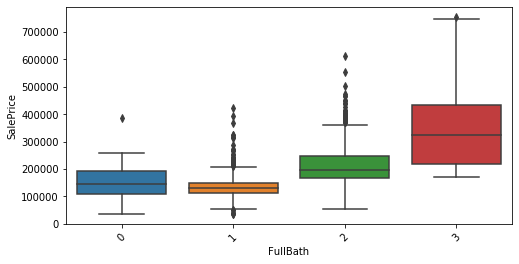
## **Exploratory Data Analysis**

* Understand the problem. We'll look at each variable and do a philosophical analysis about their meaning and importance for this problem.
* Univariate study. We'll just focus on the dependent variable ('Sale Price') and try to know a little bit more about it.
* Multivariate study. We'll try to understand how the dependent variable and independent variables relate.
* Basic cleaning. We'll clean the dataset and handle the missing data, outliers and categorical variables.
* Test assumptions. We'll check if our data meets the assumptions required by most multivariate techniques.

Multivariate study:

###### 





EDA – Summary:

* SalePrice is right skewed and not normally distributed. Transformed Salesprice using log transformation.
* 'SalePrice' has high correlation with 'OverallQual', 'GrLivArea', 'GarageCars', 'TotalBsmtSF', 'FullBath', 'YearBuilt' feature.

###### *TotalBsmtSF' and 'GrLiveArea' showing linear relation - drawing a linear line.*

###### *SalePrice' and 'YearBuilt' showing booming real estate market.*

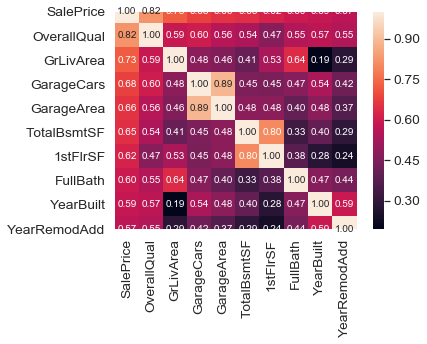
###### *In conclusion, these highly correlated feature seems to be good contributor for predictions.*

* Outliers can cause underfitting and increase in cost function. Therefore, GrLivArea > 4000 and sales price 300000 has been removed.
* Heat Maps showing correlation between target variable and with other features. Though correlation is not necessarily tells the importance of independent features but it can provide a good idea for further analysis.
* Higher correlation among features could be multicollinearity problem.

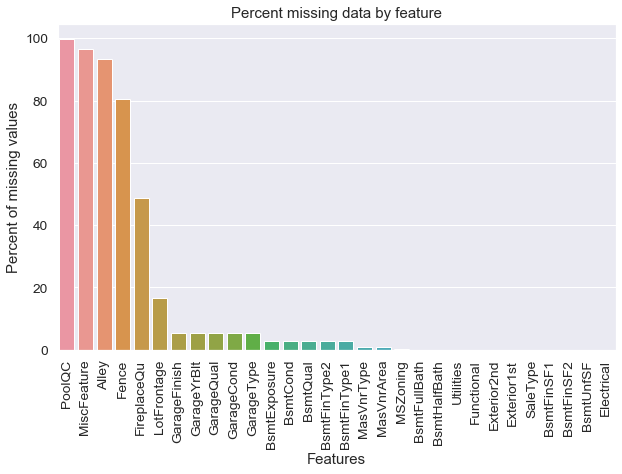
Outliers:

|  |  |
| --- | --- |
|  |  |

Correlation:



Data Processing



Handling Missing values

* Categorical values which are Missing , replaced with none
* Numerical features’ missing values , replaced by median and Zeors

Transforming some numerical variables that are really categorical

\*\* refer to note book for details

Feature Engineering

##### Creation of new features from numerical features

Combining some features based on the real life experience . This might help model to predict accurately. Also, there are other techniques like in some cases where linear relation is not able to separate different values, we use higher degree functions to find patterns

all\_data['YearsSinceBuilt'] = all\_data['YrSold'].astype(int) - all\_data['YearBuilt']

all\_data['YearsSinceRemod'] = all\_data['YrSold'].astype(int) - all\_data['YearRemodAdd']

all\_data['OtherRooms'] = all\_data['TotRmsAbvGrd'] - all\_data['BedroomAbvGr'] -

all\_data['TotalBathrooms'] = all\_data['FullBath'] + (0.5 \* all\_data['HalfBath']) + all\_data['BsmtFullBath'] + (0.5 \* all\_data['BsmtHalfBath'])

all\_data['LotDepth'] = all\_data['LotArea'] / all\_data['LotFrontage']

all\_data['TotalSF'] = all\_data['TotalBsmtSF'] + all\_data['1stFlrSF'] + all\_data['2ndFlrSF']

Model Performance:

Following results are based on KFOLD Cross validation (k = 5) .

|  |  |
| --- | --- |
| Model | RMSE (5-fold mean)/std |
| *LASSO Regression* | 0.1111 / (0.0068) |
| Kernel Ridge Regression | 0.1154 / (0.0053) |
| XGBoost | 0.1178 / (0.0062) |
| LightGBM | 0.1159 / (0.0059) |

LASSO Regression proved to be best based on above CV values.

Further Steps:

## Stacking models