**Kaggle Competition: Predicting Housing Prices in Ames, Iowa, USA**

**About the Dataset**

The Ames Housing Dataset was introduced by Professor Dean De Cock in 2011.

Kaggle provided separate CSV file for train and test. The train dataset contains

1460 rows and 81 columns, whereas test dataset contains 1459 rows and 80 columns.

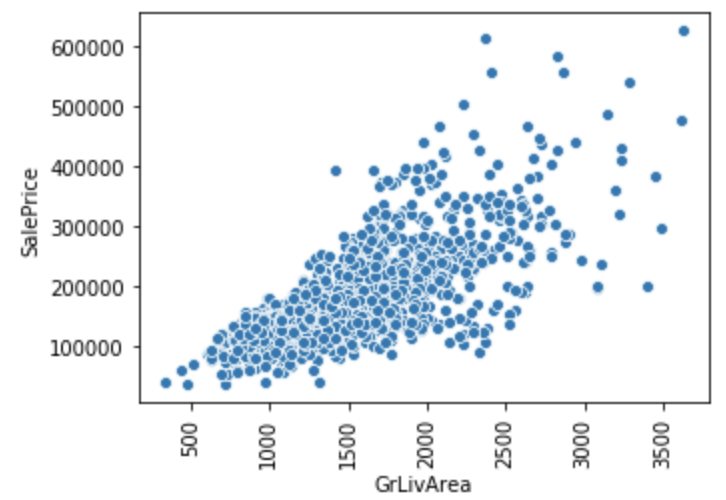
The dataset contains house features as column for individual house.

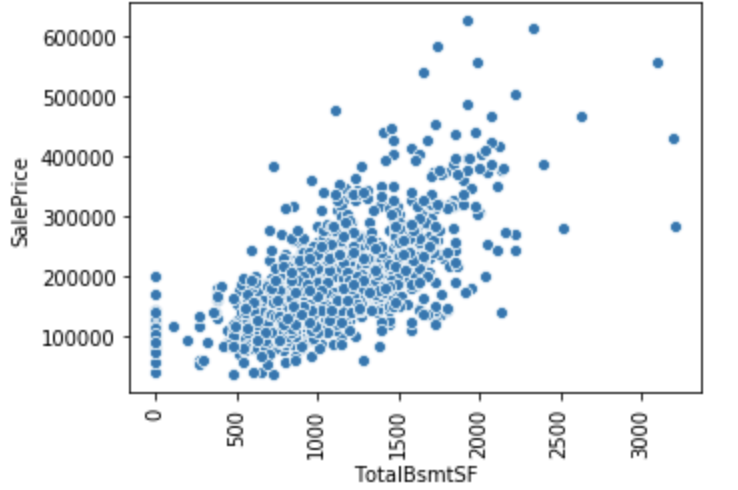
**1.Data Exploration**

The dataset contains house features like Area, number of bed rooms, Bath rooms, house style, lot shape, Neighborhood, year built etc. Since this project is more concern about prediction, I concentrate more on how features are related to target variable (SalePrice).

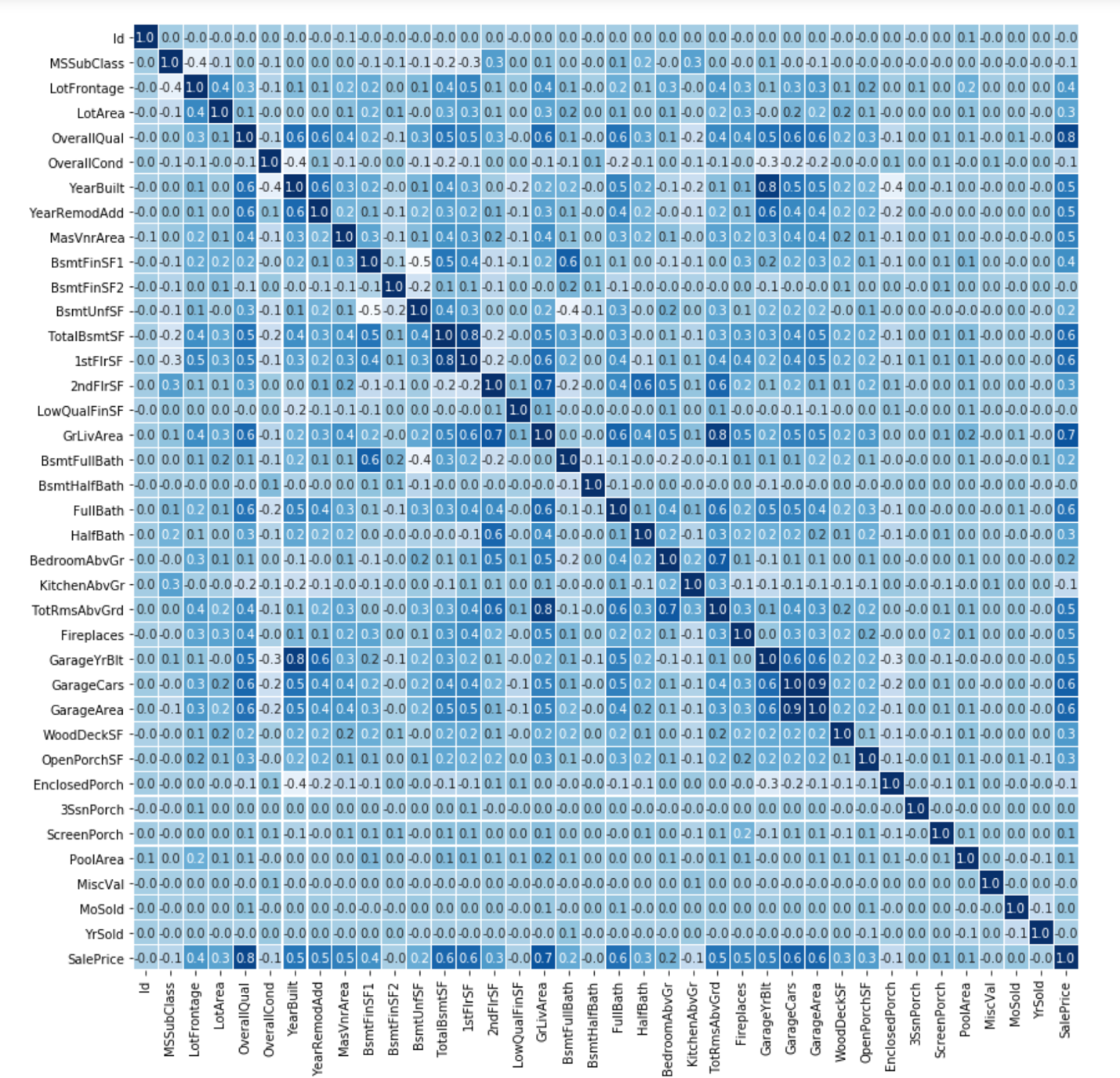
**Numerical Features**

There are lot of numerical features that are correlated with target variable, among them GrLivArea and TotalBsmtSF shows high correlation (0.7) and (0.6) respectively





To get good idea of all numerical feature, I use heat map



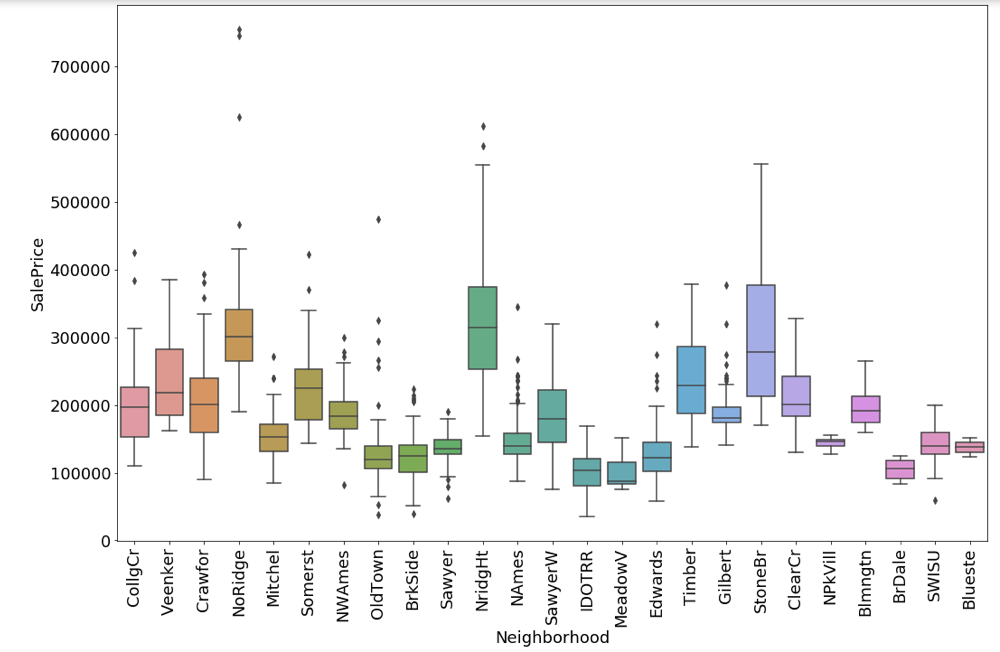
Heat map clearly shows correlation, between target variable and other features, last row in the figure shows all correlation, we can also see high correlation between features like (GarageArea Vs GarageCars) and (GarageYrBlt Vs YearBuilt ).

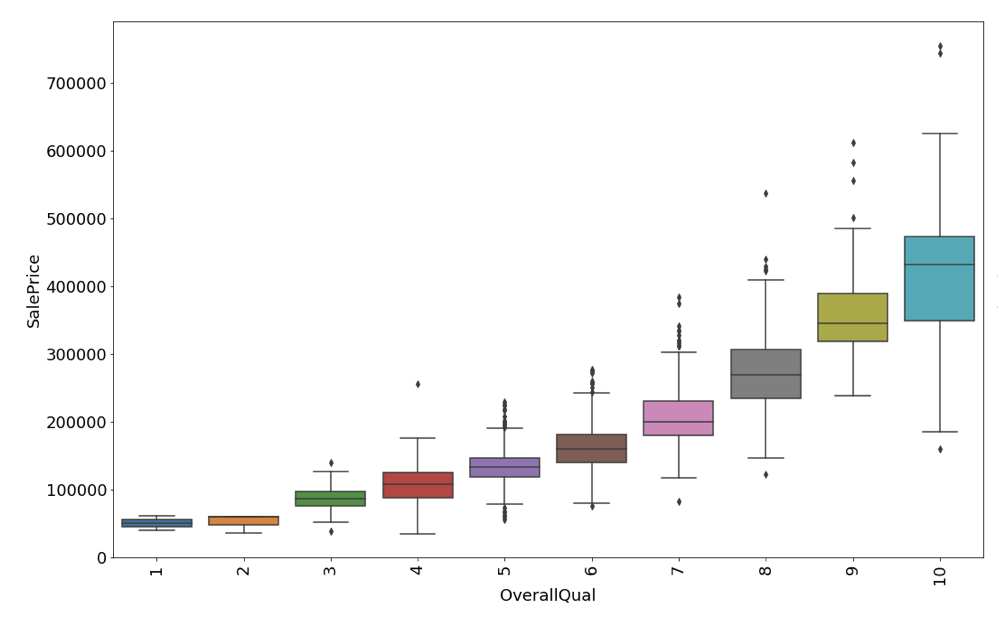
There are some features that we should consider as categorical, even though it’s not defined as object type by default dataset.

1. MSSubClass
2. OverallQual
3. OverallCond
4. YearBuilt
5. YearRemodAdd
6. BsmtFullBath
7. BsmtHalfBath
8. FullBath
9. HalfBath
10. BedroomAbvGr
11. KitchenAbvGr
12. TotRmsAbvGrd
13. Fireplaces
14. GarageCars
15. MoSold
16. YrSold

**Categorical Features**

There are lot of interesting patterns to see with categorical vs SalePrice, few notable variables are Neighborhood, OverallQual

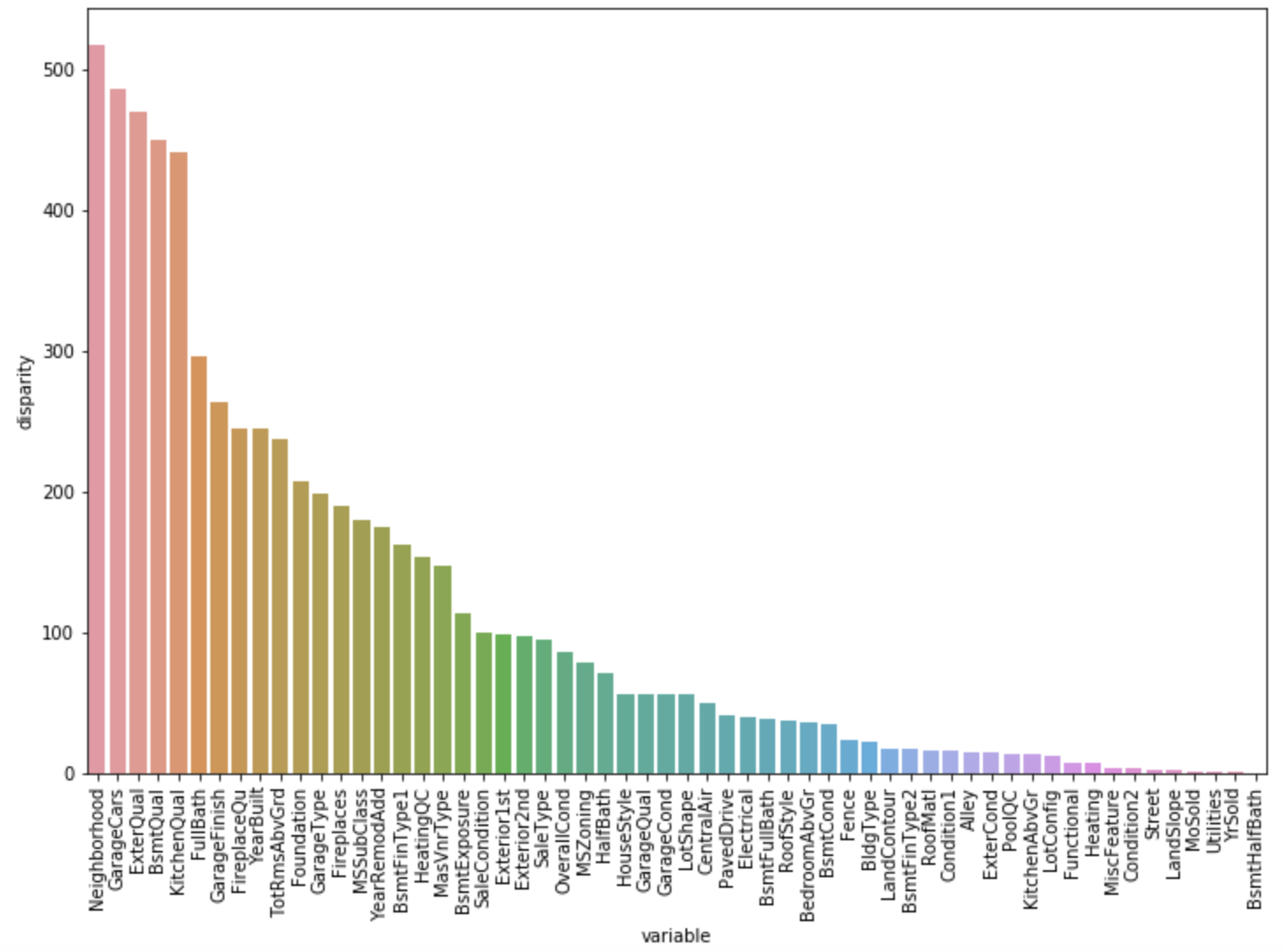




There is clear variation in medium price, if we look at Neighborhood variable NridgHt, StoneBr has high median SalePrice compare with all values in the group. So do OverallQual as Quality increases median value increases.

**Categorical Variable Significance (ANOVA)**

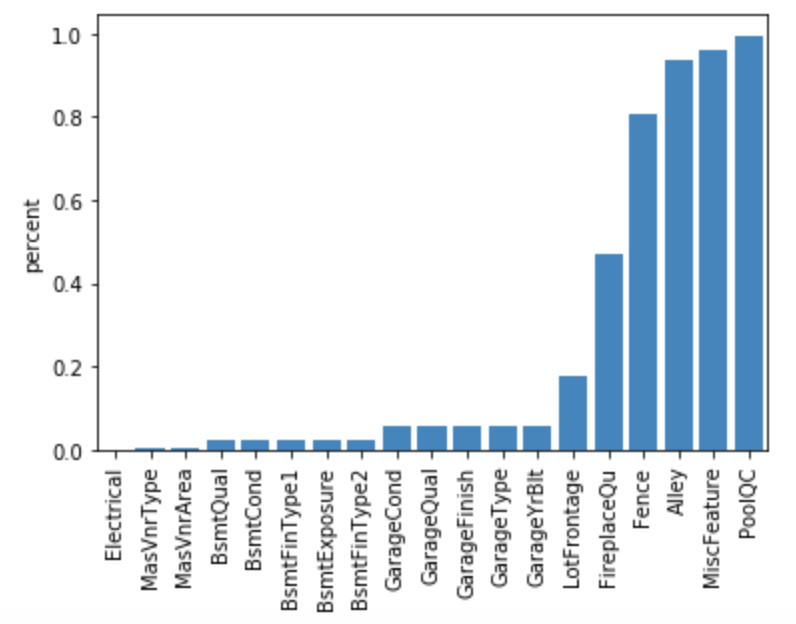
H0 (Null Hypothesis) - There exists no significant difference between the groups.   
Ha (Alternate Hypothesis) - There exists a significant difference between the groups.



This graph shows Neighborhood, GarageCars, ExterQual etc. features group has significance with respect to SalePrice.

**2. Handling NULL/Missing Values**

This dataset contains NA values, but it’s not random, the data description shows those NA values are on purpose to shown that particular house does not have that feature. So I did not remove NA values instead treat them as no feature.



For example PoolQC has more than 90% NA values, which indicates Pool not present in those houses.

But Electrical features has NA values, but description shows those values are random, so I choose to fill those kind of values group by Neighborhood and take mode for categorical variable and mean for numerical variable.

**3. Feature Engineering**

Sometimes combining or breaking variables helps to improve model predication. Below are the variable that I created

**Numerical:**

TotalArea1st2nd = 1stFlrSF + 2ndFlrSF

TotalBath= BsmtFullBath + BsmtHalfBath+ FullBath + HalfBath

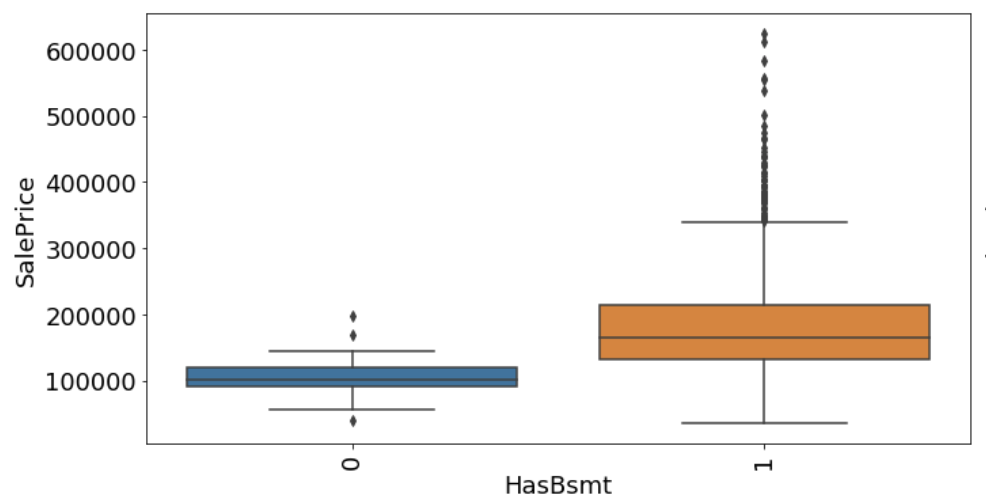
**Categorical:**

HasBsmt = TotalBsmtSF > 0

HasPool = PoolArea > 0

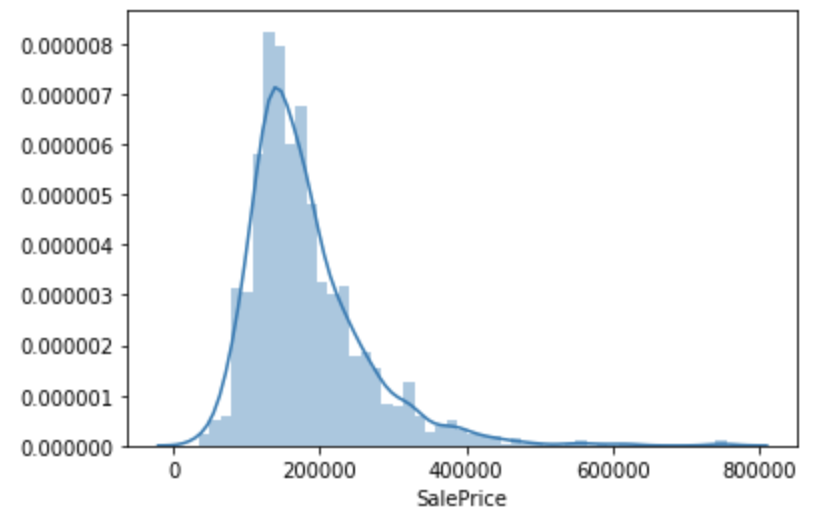
HasFirePlace = FirePlace > 0

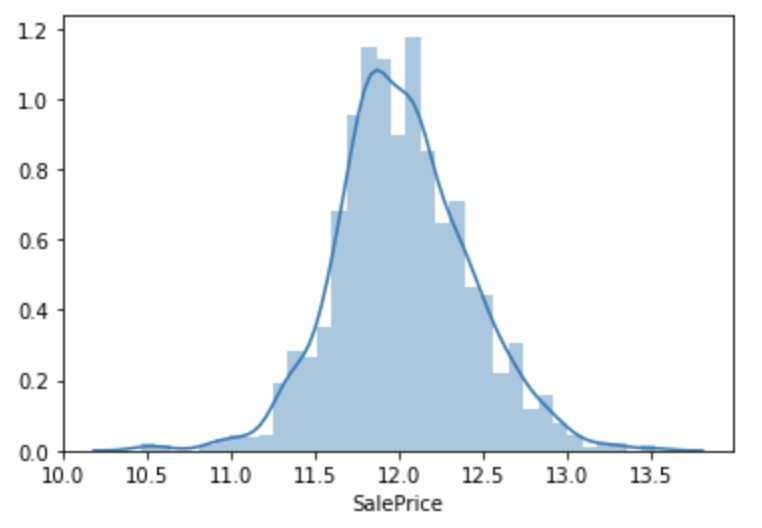
HasGarage = Garage > 0



**4.Model Fitting**

Before modeling it’s better to normalize target variable, below are two figure shows before and after normalization respectively.

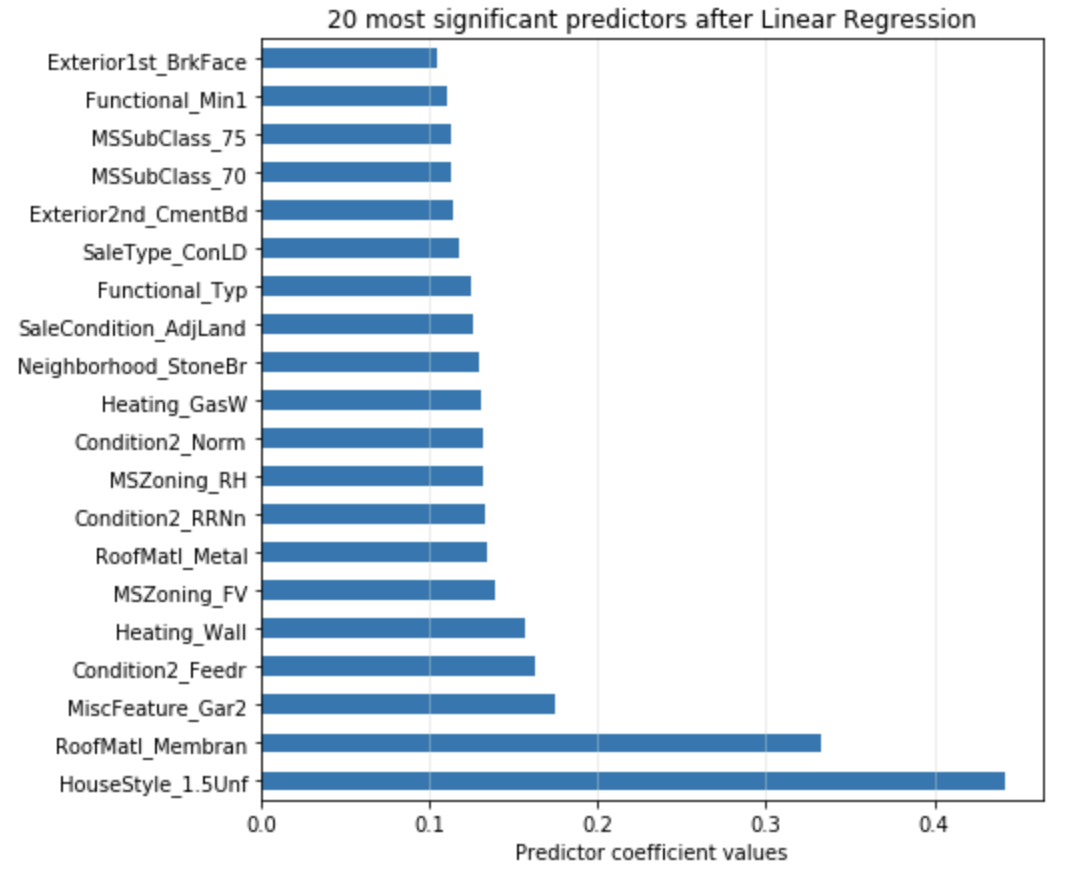
****

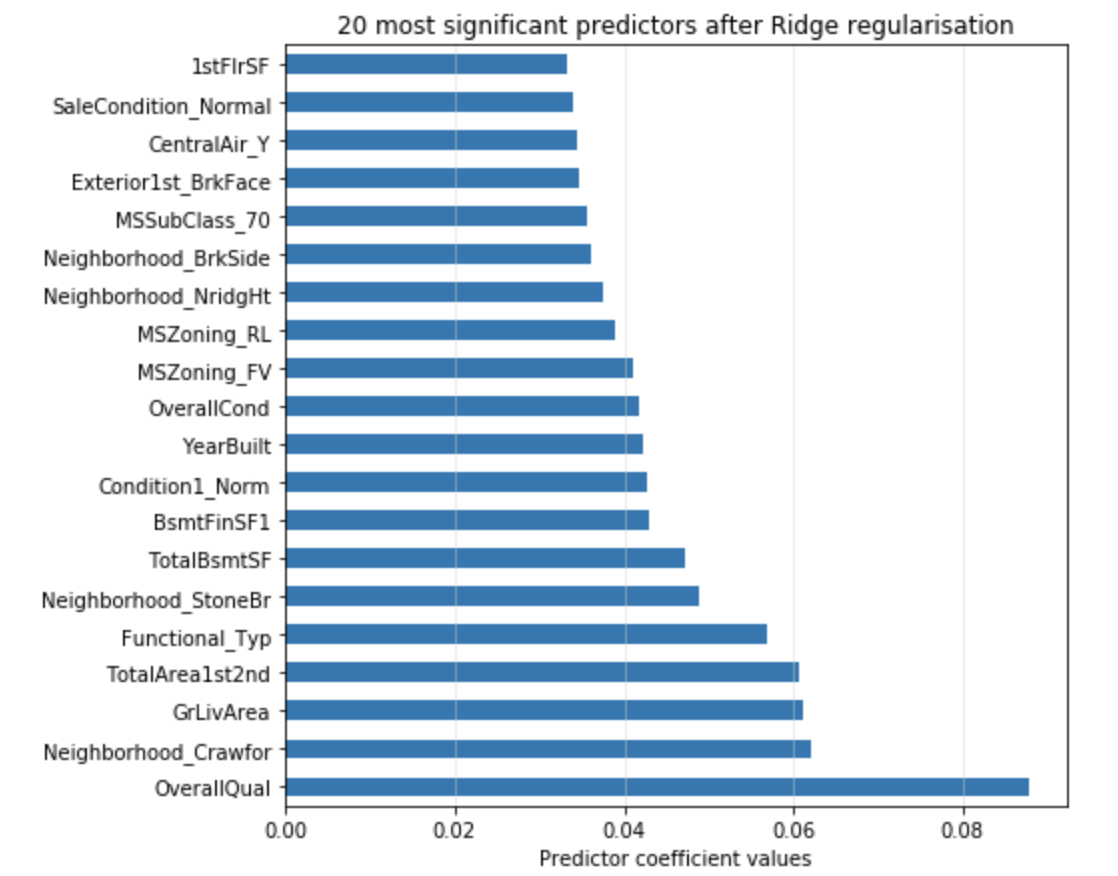
****

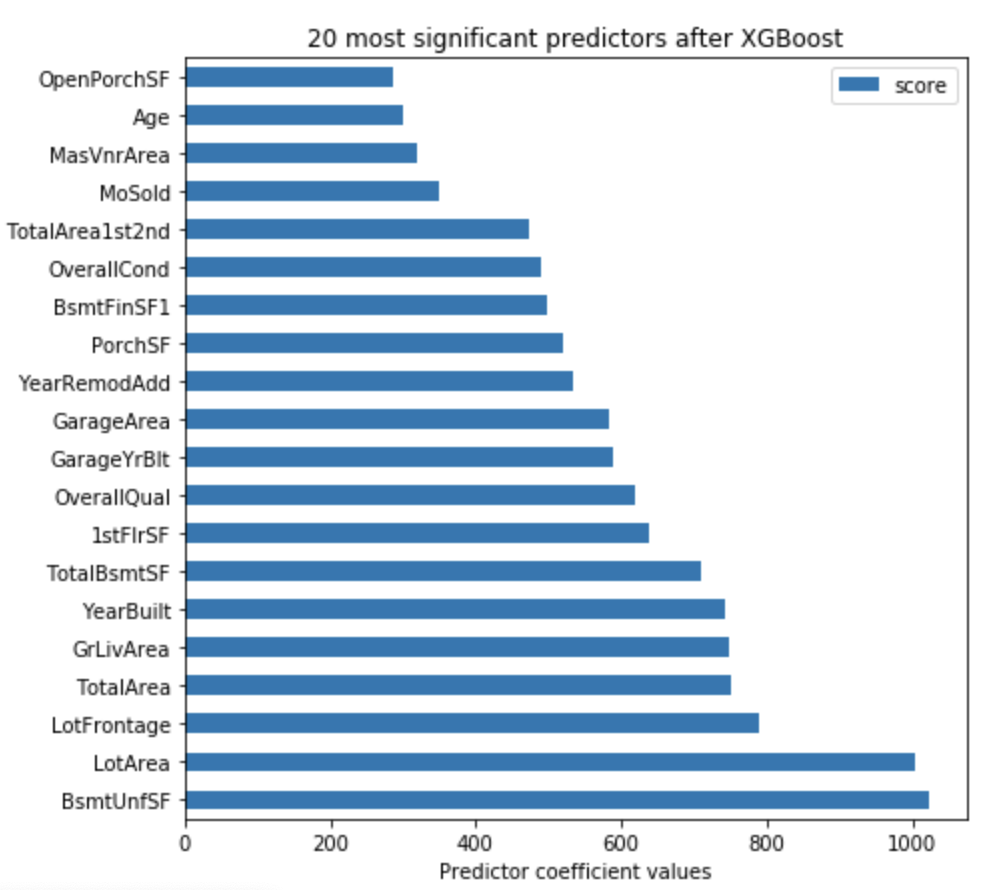
Below are the models that is tried to fit the dataset

* Linear Regression
* Ridge Regression
* XGBoost
* Light Gradient Boosting
* Stacking models
* Blended models

After fitting models below are the Best features choose by few models







Below are few house features that choose by most of ML models.

* Surface Area (TotalBsmtSF, GarageArea, ...)
* Neighborhood
* YearBuilt
* HouseStyle
* MSZoning

I used RMSE to determine performance of each model in training dataset

Mean Predictor 🡪 0.3925287056767283

Linear Regressor 🡪 0.12547247570078915

XGBoost Regressor 🡪 0.11188695657683541

Light Gradient Boosting Regressor 🡪 0.11344042661577285

Ridge Regressor 🡪 0.11484707745530867

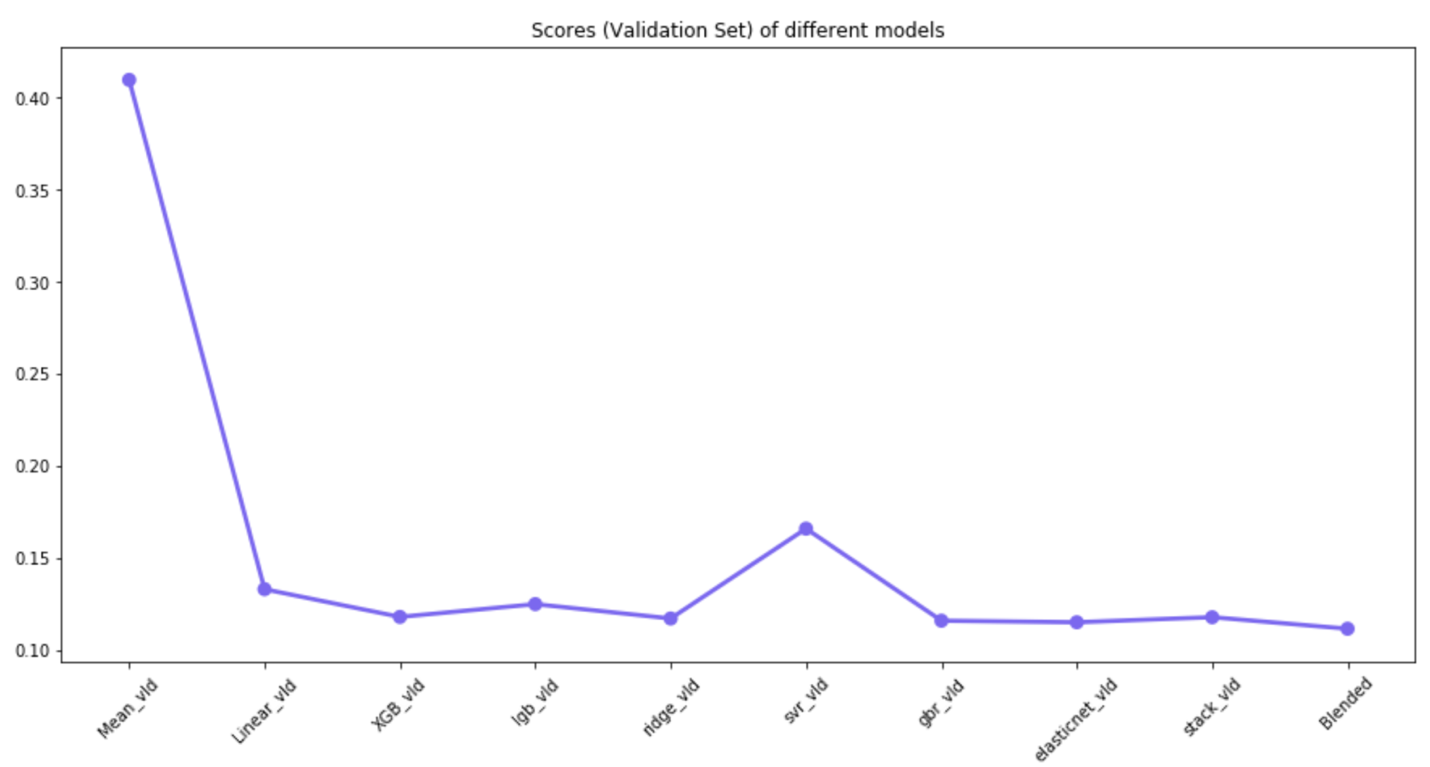
Support Vector Regressor 🡪 0.17811431394431906

Gradient Boosting Regressor 🡪 0.11226530074097474

ElasticNet Regressor 🡪 0.11366619563827776

Stacked Regressor 🡪 0.11674449588385769

Blended Regressor 🡪 0.11111402140120144



The Blended model works well with this dataset, I used it to predict SalePrice for test dataset and below is the score from Kaggle.

Test Data Set 🡪 0.12372 (Kaggle Score)

**Conclusions**

To summarize, this project shows the strength of linear model, tree-based models, stacked and blended models. It also shows how one-hot encoding, ordinal encoding, box-cox transformation, and parameter tuning improved our models. Although different model select differ feature for its predictions, if we check all model certain variables were chosen by most of the model.

For a Kaggle competition, very little score improvement is an great achievement, combined models worked very well, taking advantage of the strengths of different models. However, if we want to build a model --both to make a prediction and to get information about the response variable linear models works well, but if we care about better prediction then depending on dataset tree based model and combined model works well.