# Predicting House Prices with Regression Techniques

A Kaggle Project

Tyler Wilbers
**NYC Data Science Academy**

# Outline

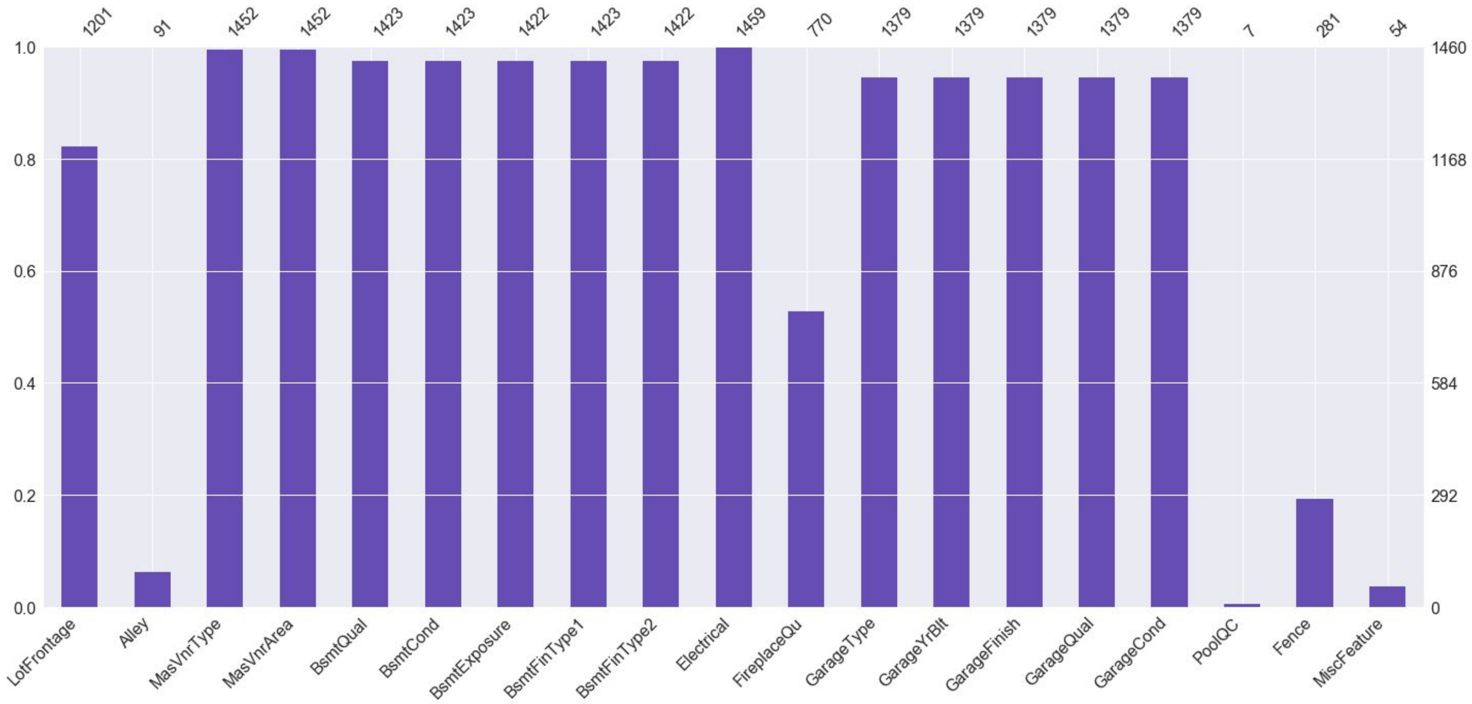
- EDA
    - Missingness
        - Imputations
    - Other interesting EDA
        - Neighborhoods vs.  Price Exploration
- Preprocessing
    - Skewness and Outliers
        - Box Cox Transformations
        - Standardize Predictors
    - Correlation
    - Multicollinearity
    - Selection
- Modeling
    - Tree methods
    - Linear
    - What we did wrong
    - What we did right
- Future  Improvement
    - Kaggle Score
    - Future improvement

# Exploratory Data Analysis (EDA)

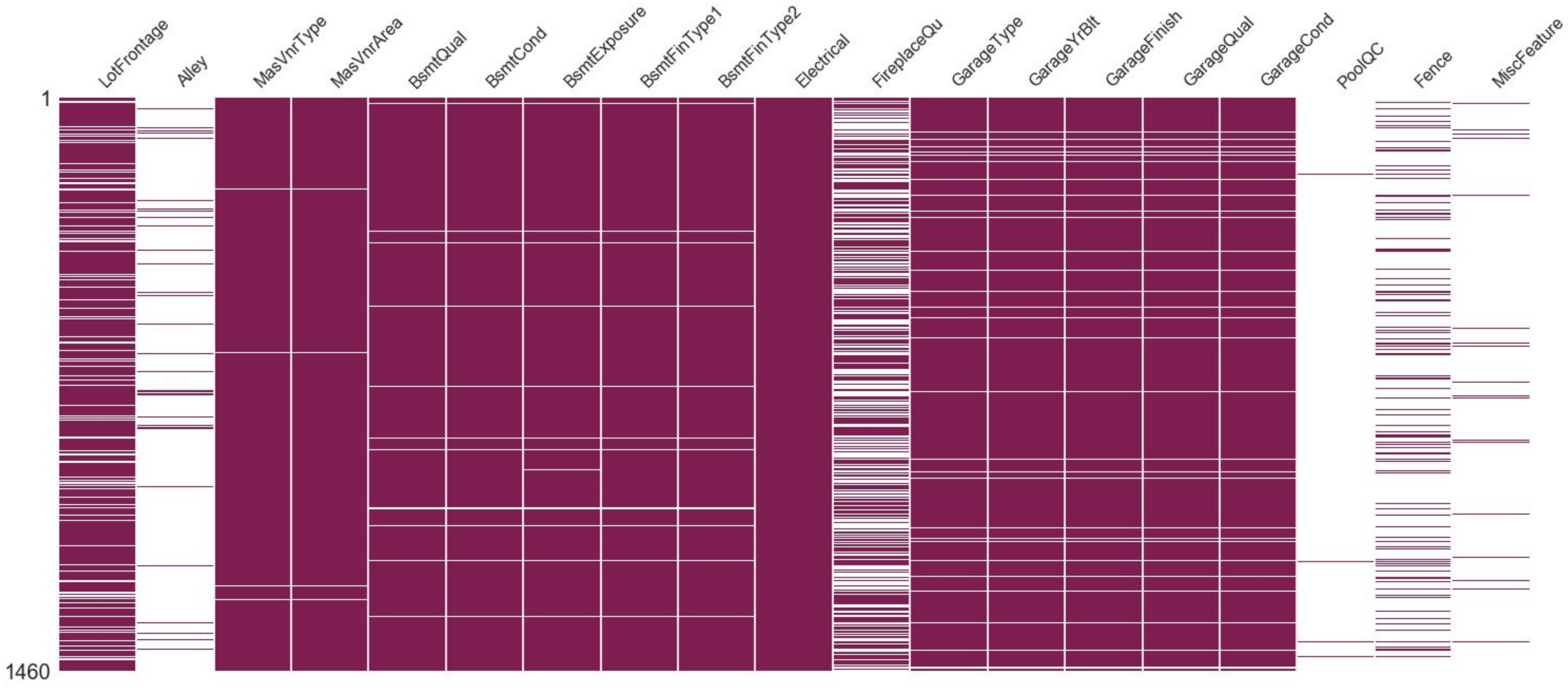
# Findings

- 80 variables(Exclude: ID)
    - Categorical → 63
    - Continuous → 17
- Counts of all NAs : 6965
- Number of Variables that contains missing values : 19

# Amount of Missingness of each column



# Relationship of missingness b/w each variable



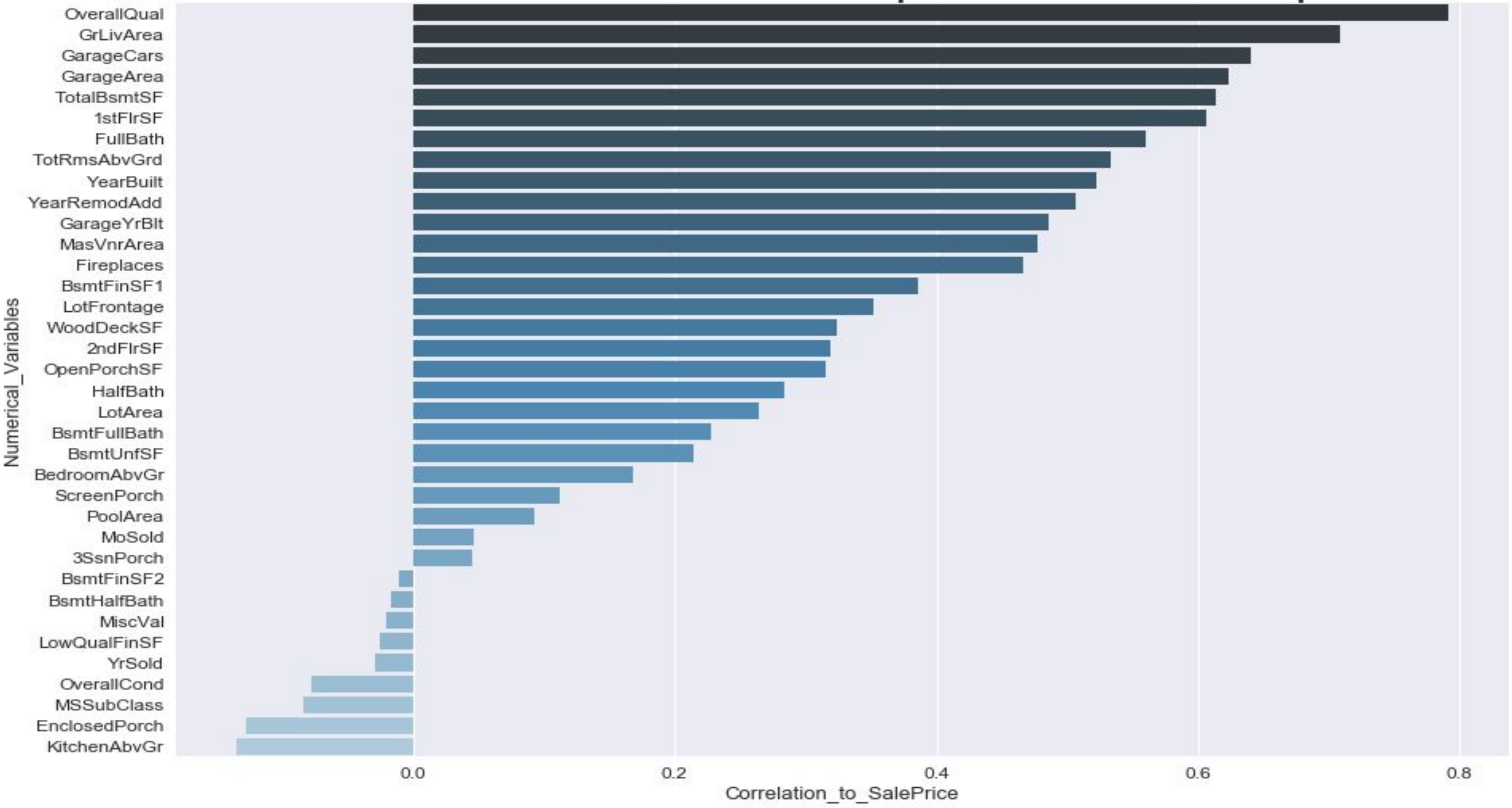
# Imputations

- We took a two part approach:
  - For continuous variable, Group by = neighborhood and impute mean.
  - For categorical variable, we just **dummified;**

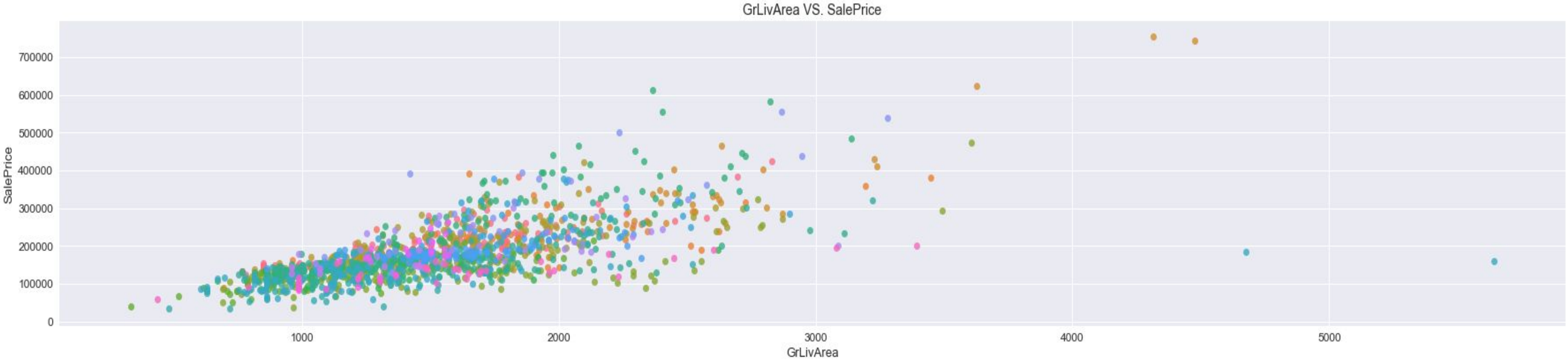
| | MiscFeature |
|---|---|
| **0** | NaN |
| **1** | NaN |
| **2** | NaN |
| **3** | NaN |
| **4** | NaN |
| **5** | Shed |

| | Gar2 | Othr | Shed | TenC |
|---|---|---|---|---|
| **0** | 0 | 0 | 0 | 0 |
| **1** | 0 | 0 | 0 | 0 |
| **2** | 0 | 0 | 0 | 0 |
| **3** | 0 | 0 | 0 | 0 |
| **4** | 0 | 0 | 0 | 0 |
| **5** | 0 | 0 | 1 | 0 |

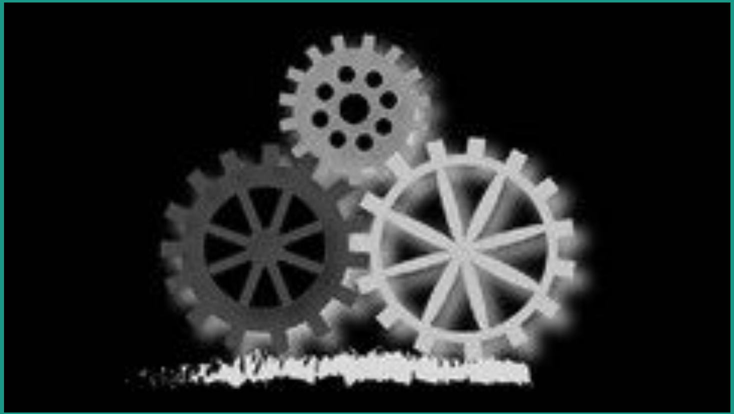
# Correlation of each numerical predictor to the response



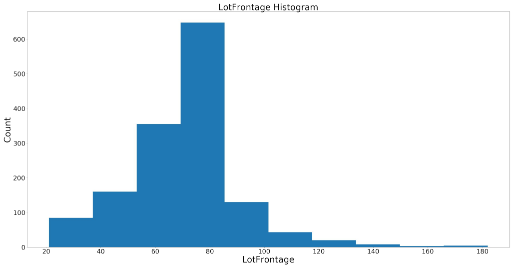
# Further Exploration - Group by Neighborhood

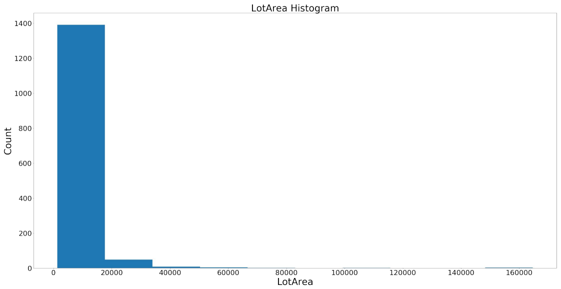


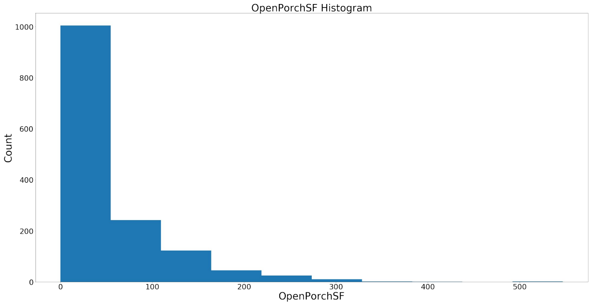

# Data Preprocessing

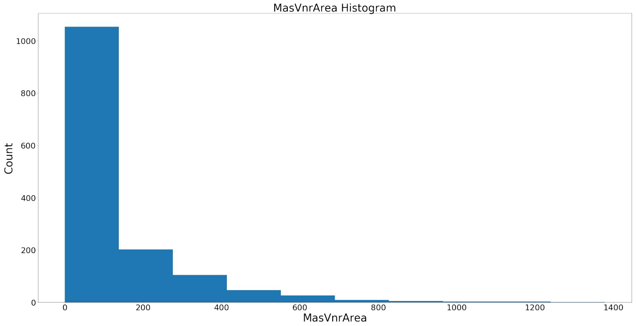


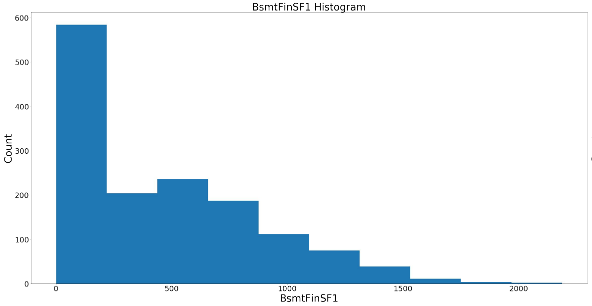
# Skewness

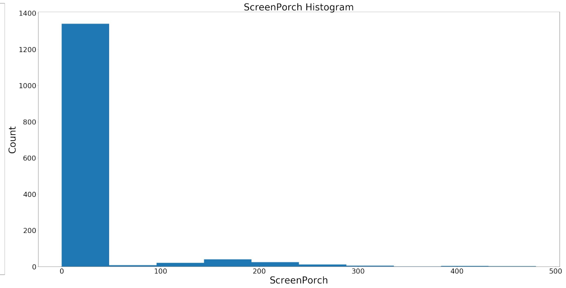

LotFrontage Histogram
Count
LotFrontage
0
100
200
300
400
500
600
20
40
60
80
100
120
140
160
180


LotArea Histogram
Count
LotArea
0
200
400
600
800
1000
1200
1400
0
20000
40000
60000
80000
100000
120000
140000
160000


OpenPorchSF Histogram
Count
OpenPorchSF
0
200
400
600
800
1000
0
100
200
300
400
500


MasVnrArea Histogram
Count
MasVnrArea
0
200
400
600
800
1000
0
200
400
600
800
1000
1200
1400


BsmtFinSF1 Histogram
Count
BsmtFinSF1
0
100
200
300
400
500
600
0
500
1000
1500
2000


ScreenPorch Histogram
Count
ScreenPorch
0
200
400
600
800
1000
1200
1400
0
100
200
300
400
500

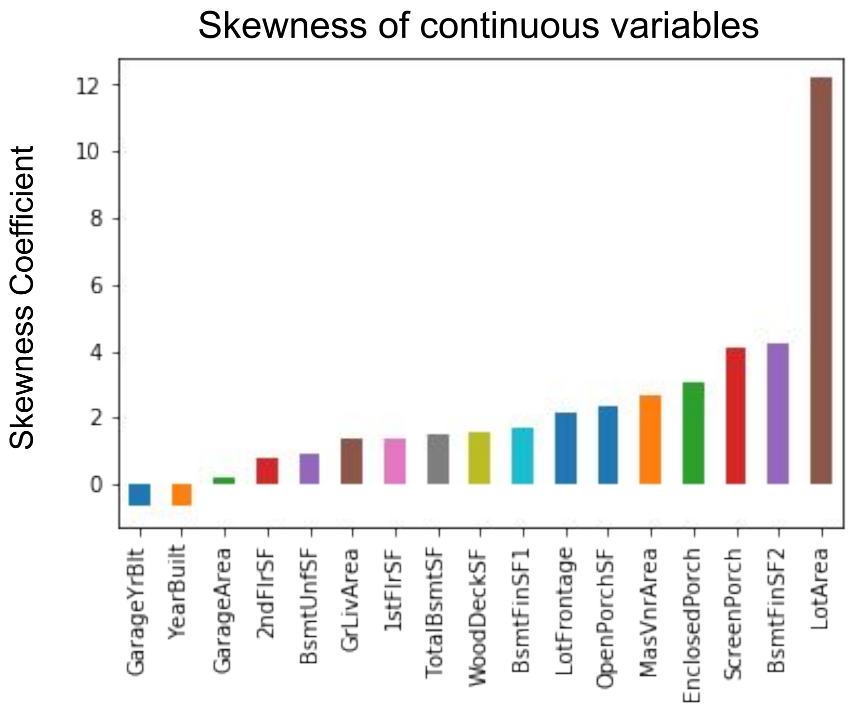
# Skewness (cont.)

- We used **scipy.stats.skew** which calculates the coefficient of skewness:

$$\frac{\mu_3}{\mu_2^{3/2}}$$

Where $\mu_i$ is the central moment.

- Negative skew usually indicates that the tail is on the left side of the distribution, and positive skew indicates that the tail is on the right.
- For normally distributed data, the skewness should be about 0.



# Box cox transformation

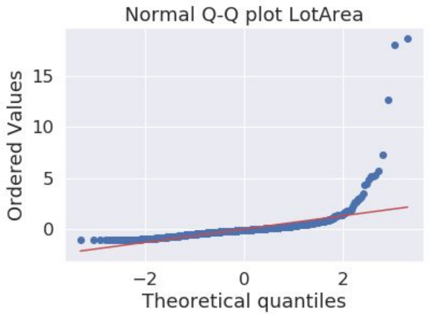
- We decided to perform a one parameter box cox transformation to the skewed variable that have a skewness ***s*** such that *2* < ***s*** < -2

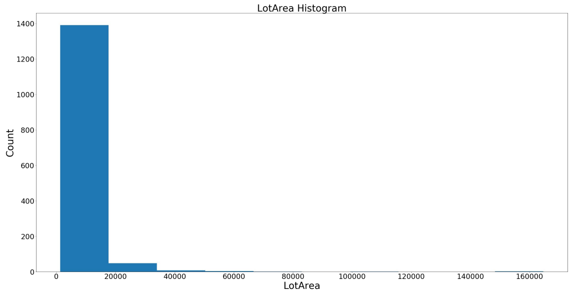
$$y_i^{(\lambda)} = \begin{cases} \dfrac{y_i^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0, \\ \ln y_i & \text{if } \lambda = 0, \end{cases}$$

- Making this transformation with **scipy.stats.boxcox1p** helped us achieve symmetry, normality, or independence of the error terms.
  - It will also help us stabilize the variance of the distributions and improve the validity of association measures (e.g. correlation).

# Example -  Lot Area

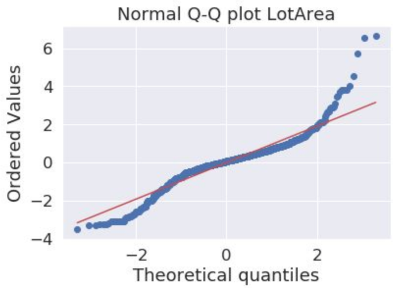
skew(LotArea) = 10.94

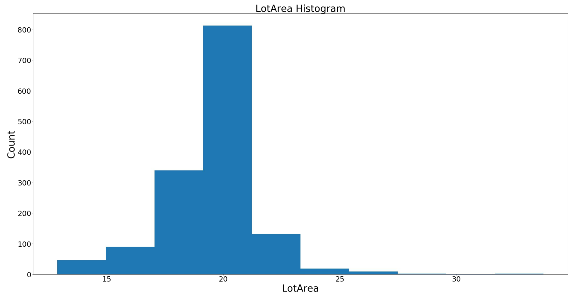


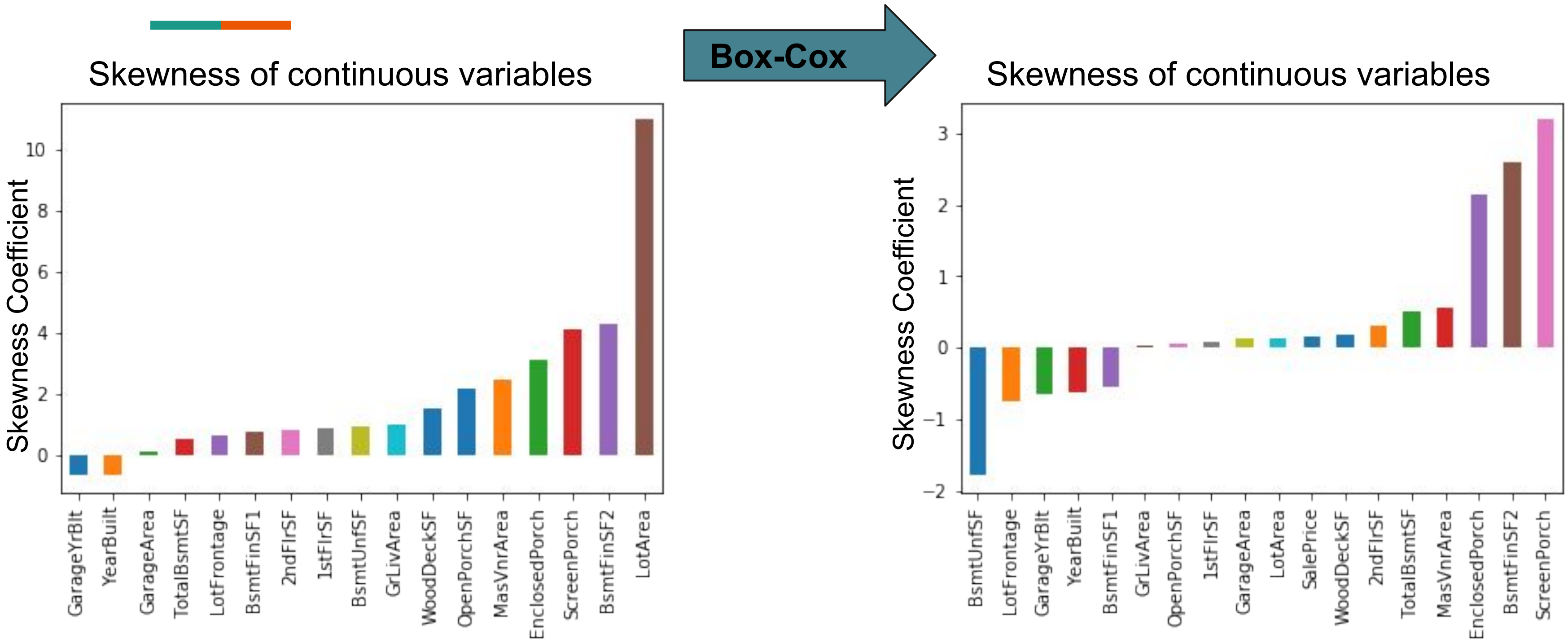

**Box-Cox**

skew(LotArea) = 0.42

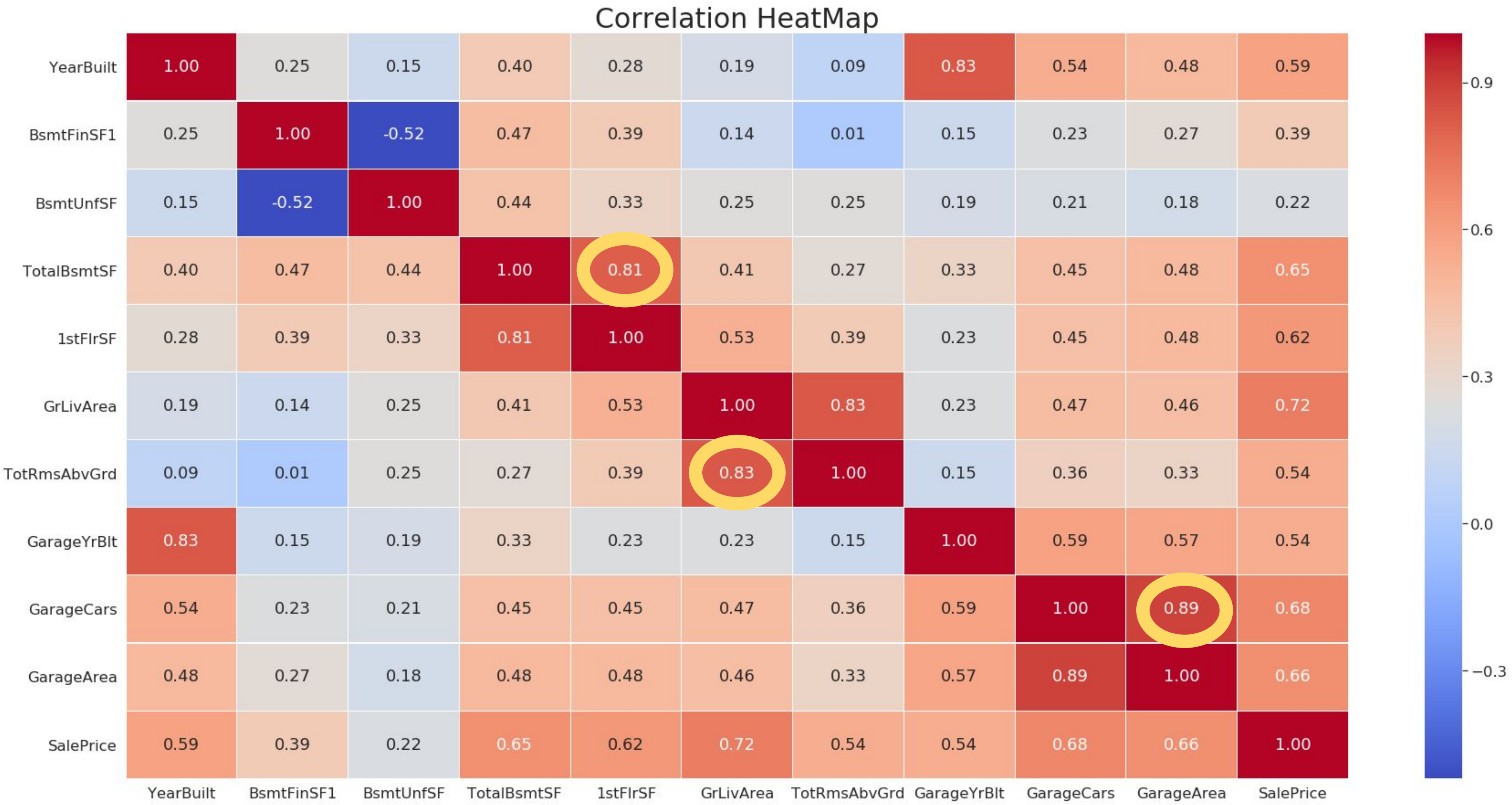


# Skewness (After box cox)



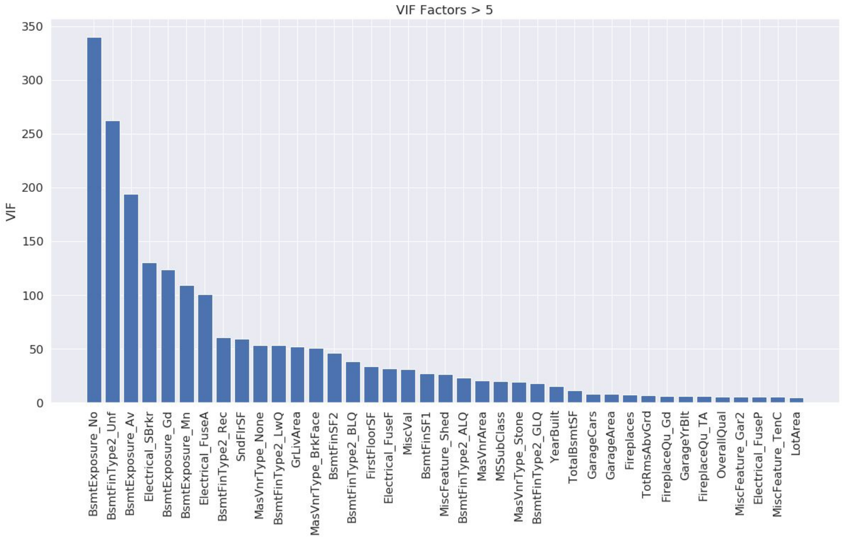
# Feature Selection - Identifying Multicollinearity


Correlation HeatMap
YearBuilt 1.00 0.25 0.15 0.40 0.28 0.19 0.09 0.83 0.54 0.48 0.59
BsmtFinSF1 0.25 1.00 -0.52 0.47 0.39 0.14 0.01 0.15 0.23 0.27 0.39
BsmtUnfSF 0.15 -0.52 1.00 0.44 0.33 0.25 0.25 0.19 0.21 0.18 0.22
TotalBsmtSF 0.40 0.47 0.44 1.00 0.81 0.41 0.27 0.33 0.45 0.48 0.65
1stFlrSF 0.28 0.39 0.33 0.81 1.00 0.53 0.39 0.23 0.45 0.48 0.62
GrLivArea 0.19 0.14 0.25 0.41 0.53 1.00 0.83 0.23 0.47 0.46 0.72
TotRmsAbvGrd 0.09 0.01 0.25 0.27 0.39 0.83 1.00 0.15 0.36 0.33 0.54
GarageYrBlt 0.83 0.15 0.19 0.33 0.23 0.23 0.15 1.00 0.59 0.57 0.54
GarageCars 0.54 0.23 0.21 0.45 0.45 0.47 0.36 0.59 1.00 0.89 0.68
GarageArea 0.48 0.27 0.18 0.48 0.48 0.46 0.33 0.57 0.89 1.00 0.66
SalePrice 0.59 0.39 0.22 0.65 0.62 0.72 0.54 0.54 0.68 0.66 1.00
YearBuilt BsmtFinSF1 BsmtUnfSF TotalBsmtSF 1stFlrSF GrLivArea TotRmsAbvGrd GarageYrBlt GarageCars GarageArea SalePrice
0.9
0.6
0.3
0.0
−0.3

# Variance Inflation Factor

$$\mathrm{VIF_i} = \frac{1}{1 - R_i^2}$$

where R2i is the R2 from a regression of Xi onto all of the other predictors. If R2, then collinearity is present, and so the VIF will be large.



# Multicollinearity

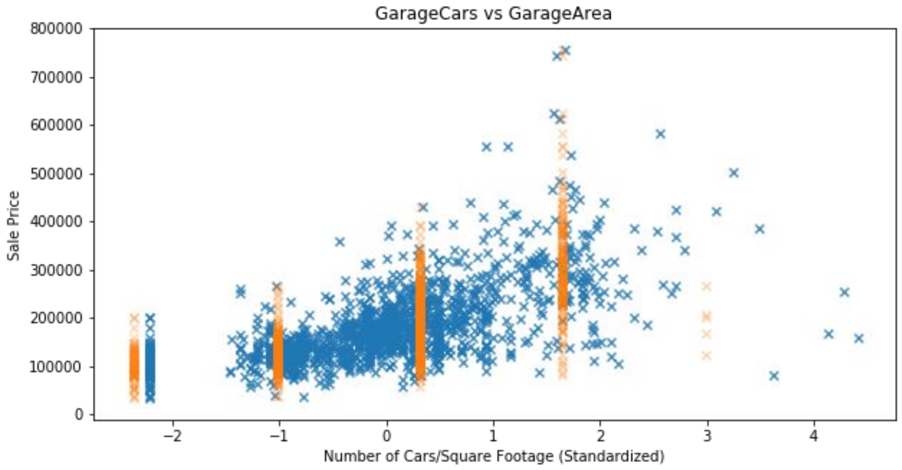
- We found that **TotalBsmtSF** is co-linear with **1stFlrSF** ($\rho$ = 0.82)
    - VIF (**TotalBsmtSF) =** 11.8922
    - **VIF(1stFlrSF) =** 36.4134

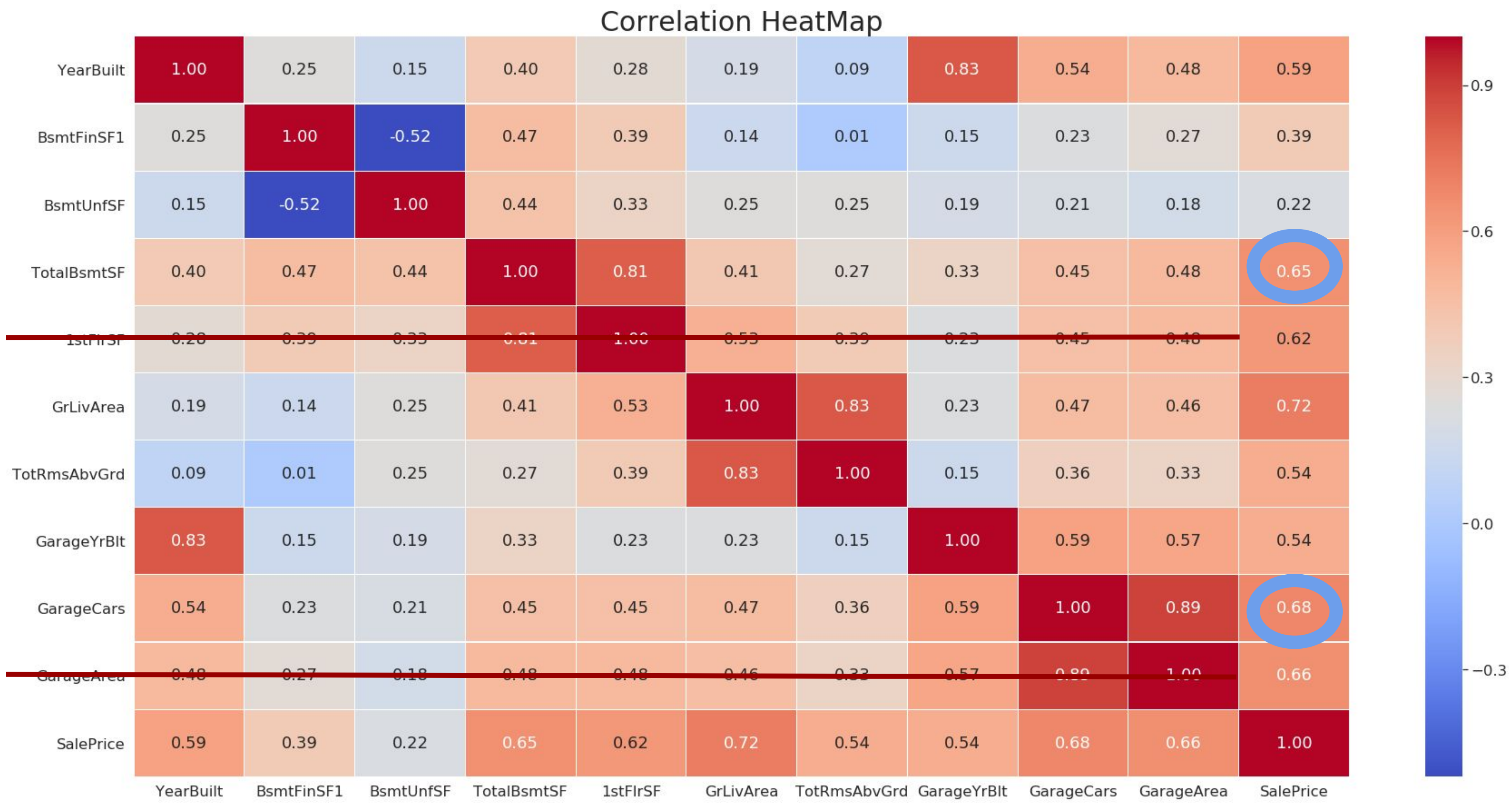
- Due to the higher VIF we opted to drop **1stFlrSF**.

# Multicollinearity

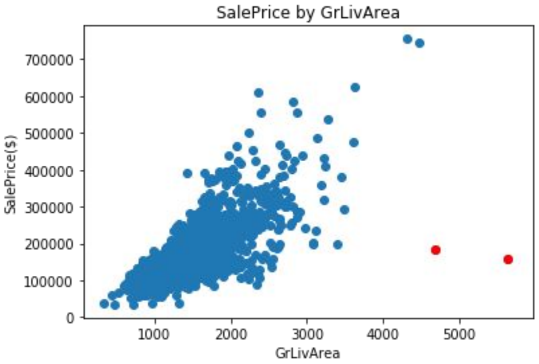
- We found that **GarageArea** is co-linear with **GarageCars** ($\rho$ = 0.89).
    - VIF(**GarageCars**) = **8.28583**
    - VIF(**GarageArea**) = **8.18866**

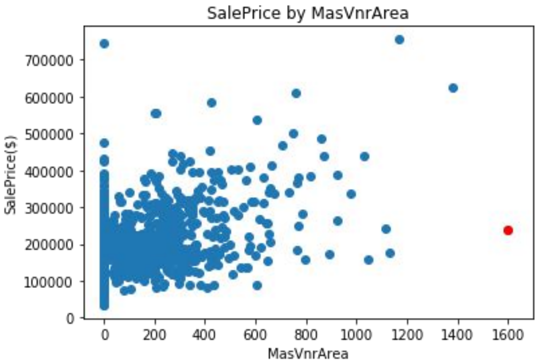
- We opted to drop **GarageArea**.

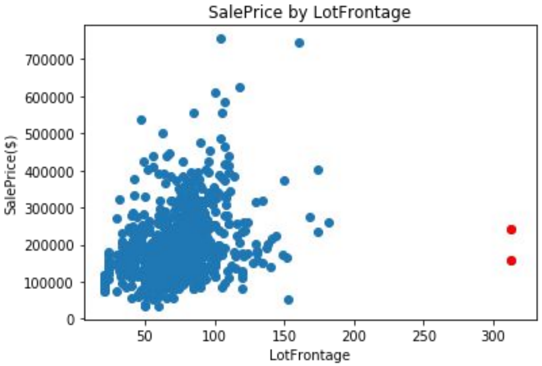


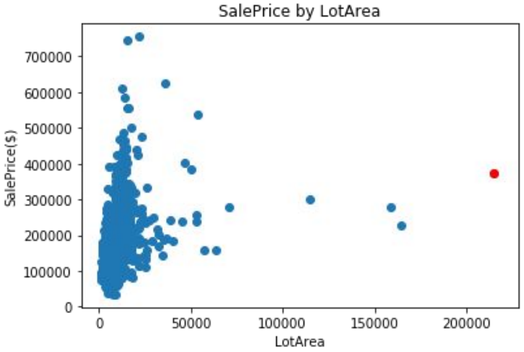

Correlation HeatMap
YearBuilt 1.00 0.25 0.15 0.40 0.28 0.19 0.09 0.83 0.54 0.48 0.59
BsmtFinSF1 0.25 1.00 -0.52 0.47 0.39 0.14 0.01 0.15 0.23 0.27 0.39
BsmtUnfSF 0.15 -0.52 1.00 0.44 0.33 0.25 0.25 0.19 0.21 0.18 0.22
TotalBsmtSF 0.40 0.47 0.44 1.00 0.81 0.41 0.27 0.33 0.45 0.48 0.65
1stFlrSF 0.28 0.39 0.33 0.81 1.00 0.53 0.39 0.23 0.45 0.48 0.62
GrLivArea 0.19 0.14 0.25 0.41 0.53 1.00 0.83 0.23 0.47 0.46 0.72
TotRmsAbvGrd 0.09 0.01 0.25 0.27 0.39 0.83 1.00 0.15 0.36 0.33 0.54
GarageYrBlt 0.83 0.15 0.19 0.33 0.23 0.23 0.15 1.00 0.59 0.57 0.54
GarageCars 0.54 0.23 0.21 0.45 0.45 0.47 0.36 0.59 1.00 0.89 0.68
GarageArea 0.48 0.27 0.18 0.48 0.48 0.46 0.33 0.57 0.89 1.00 0.66
SalePrice 0.59 0.39 0.22 0.65 0.62 0.72 0.54 0.54 0.68 0.66 1.00
YearBuilt BsmtFinSF1 BsmtUnfSF TotalBsmtSF 1stFlrSF GrLivArea TotRmsAbvGrd GarageYrBlt GarageCars GarageArea SalePrice
0.9
0.6
0.3
0.0
−0.3

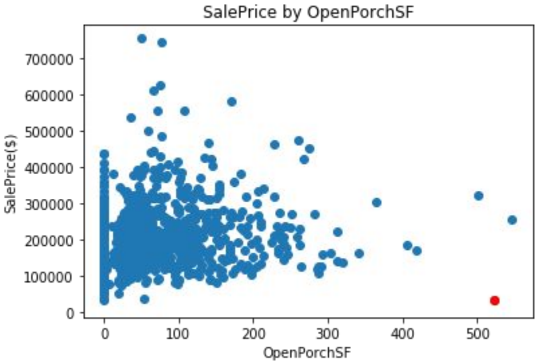
# Removing Outliers



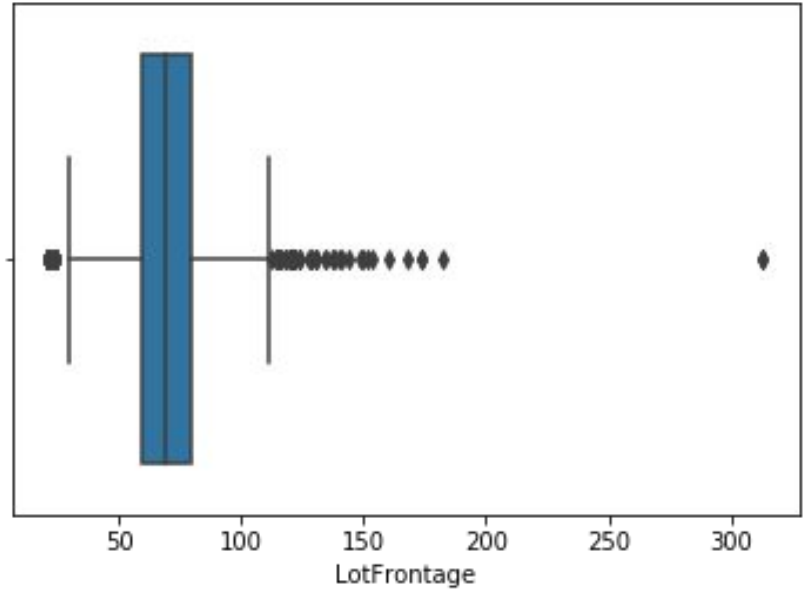



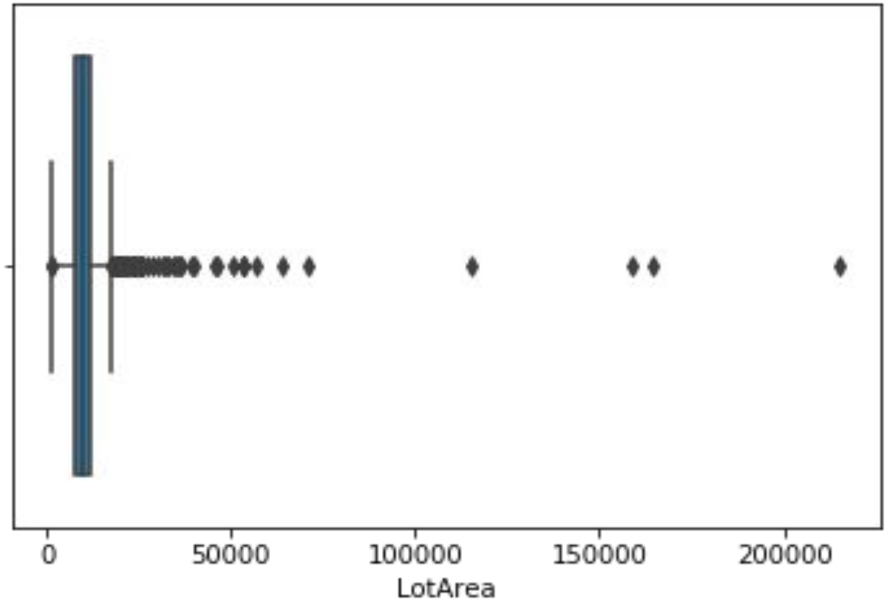



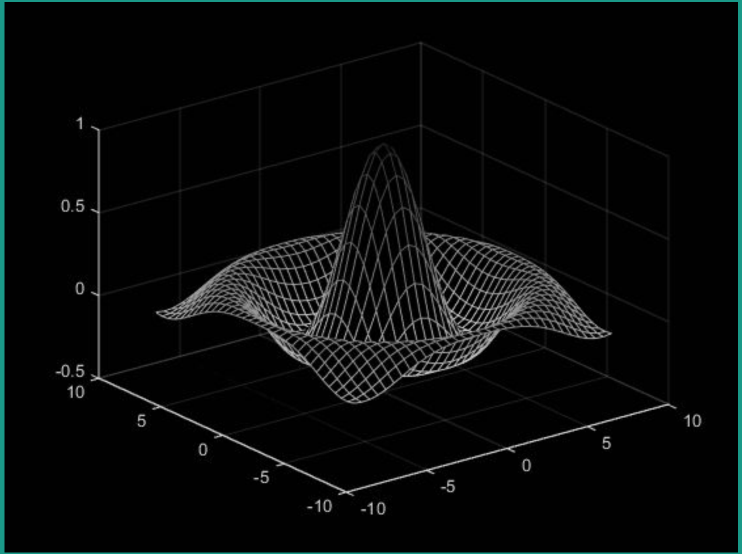



# Outliers Discovery - Examples



# Modeling



# Approach

Tree Models:

- Random Forest Regressor
- XGBoost

Linear Models:

- Lasso
- Ridge
- ElasticNet

# Tree Methods and Hyperparameter optimization

Randomized Search

+ low cost

- low precision

Grid Search

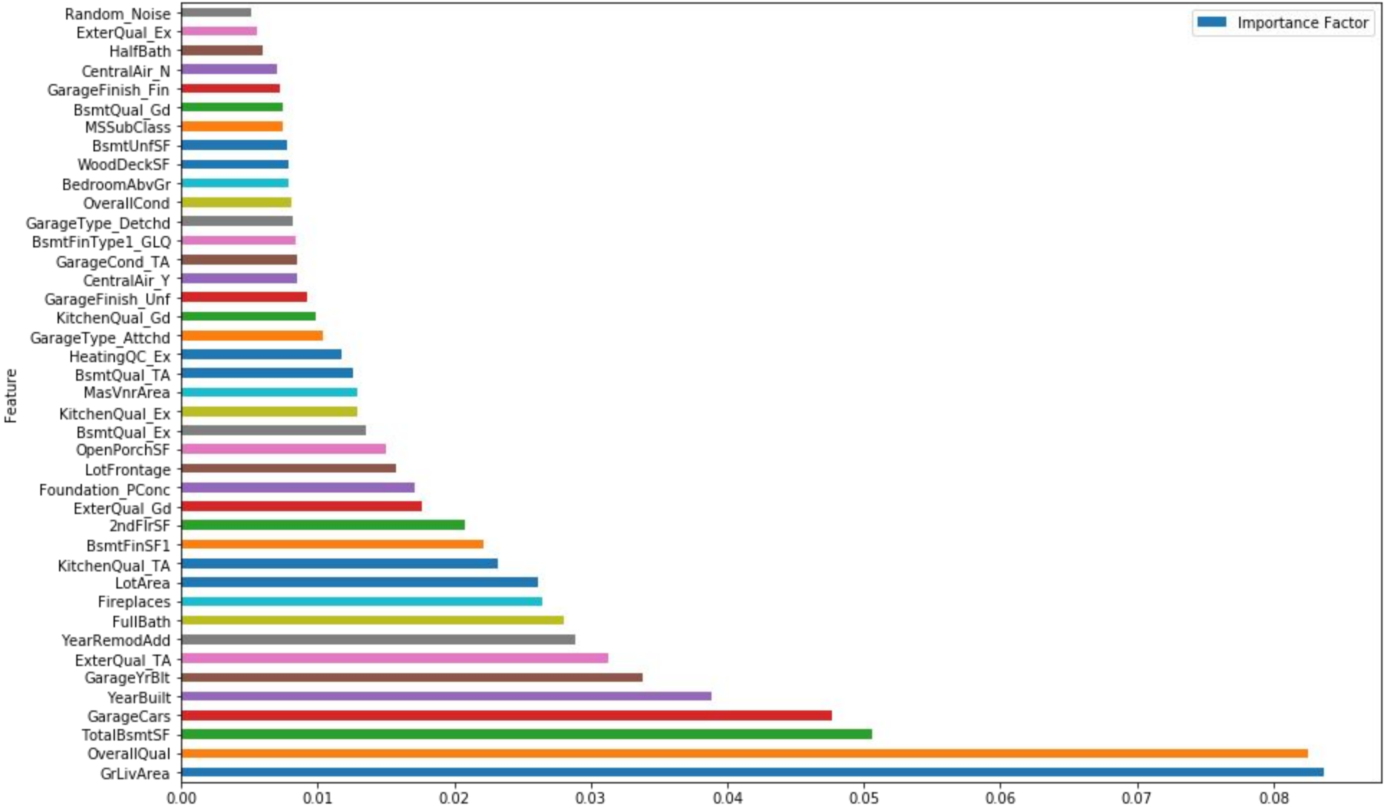
+ high  precision

*Repeat*

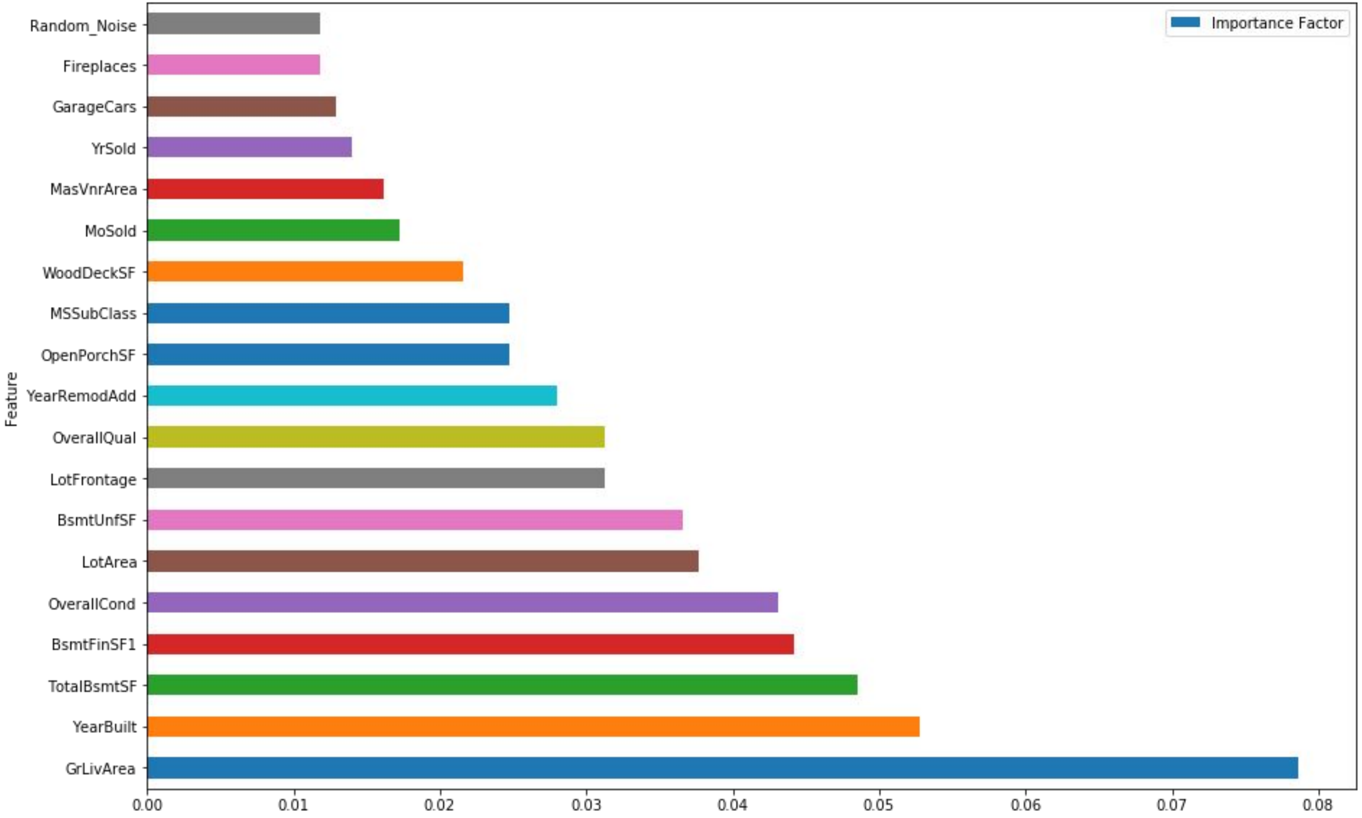
- high cost

Optimal Estimators

# Random Forest - Feature Importance



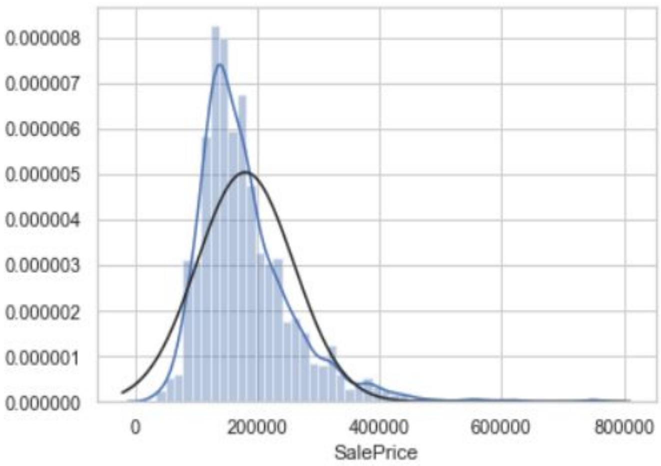
# XGBoost - Feature Importance



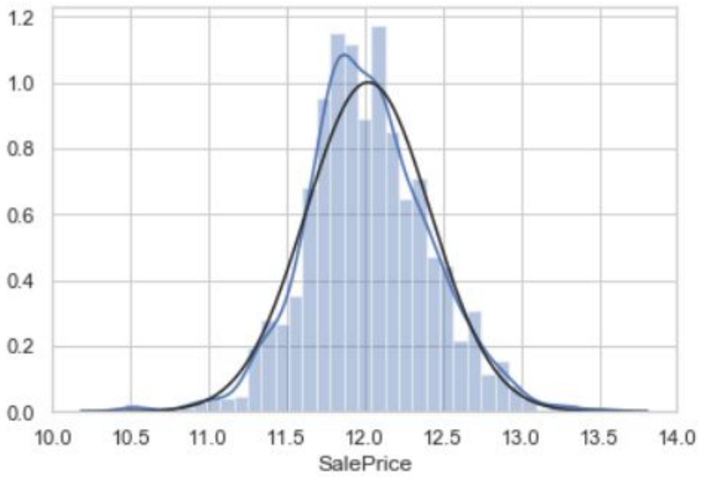
# Tree Method Results

| Model | Train CV Scores (RMSE) |
|---|---|
| Random Forest | 0.1361 |
| XGBoost | 0.1258 |

# Distribution of Sale Price with Transformation



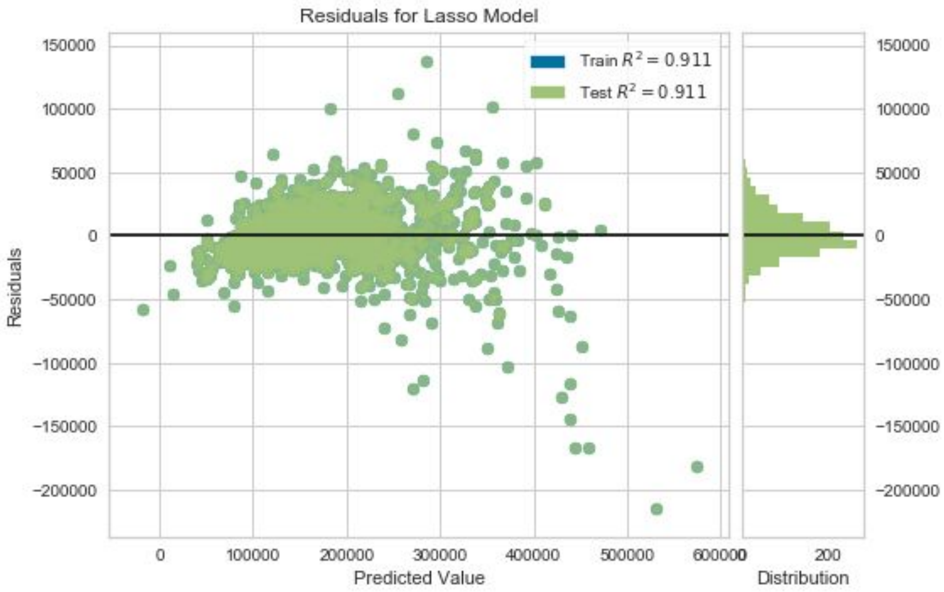

Before transformation



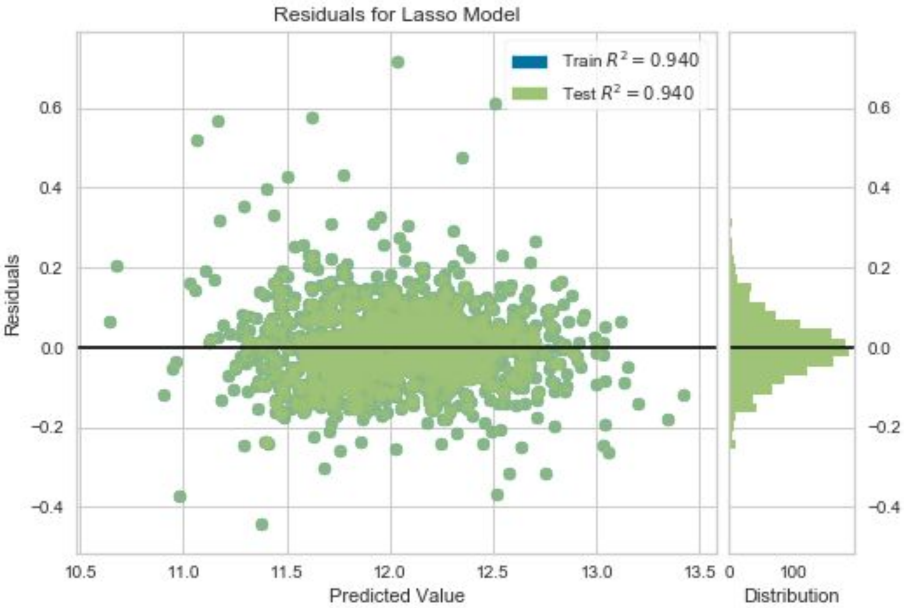

After log transformation

# Log Transform for Normally Distributed Residuals

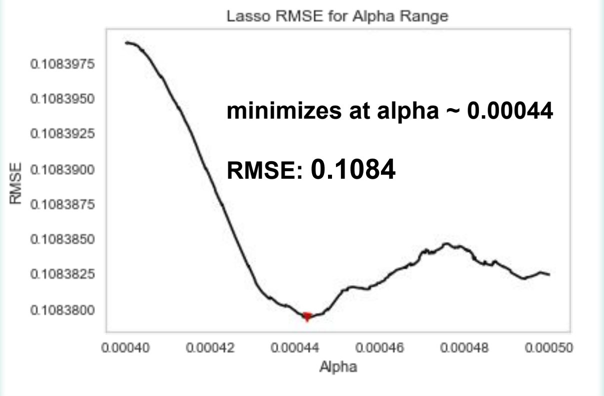
Standard target variable

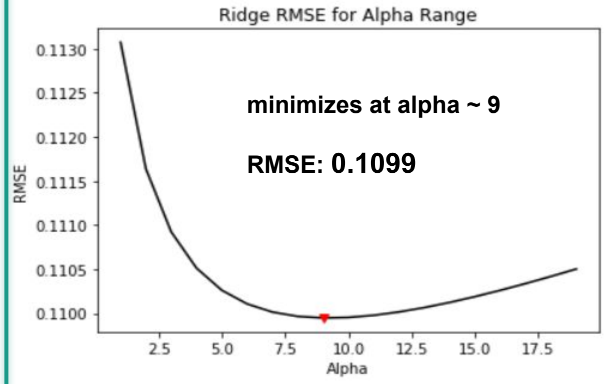


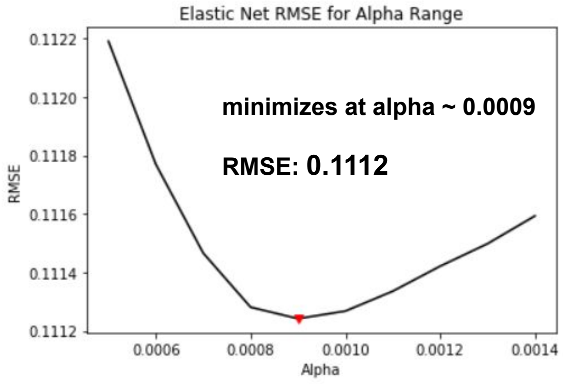

Log Transform



# Regularized Regression: Hyperparameter Testing w/ GridSearch



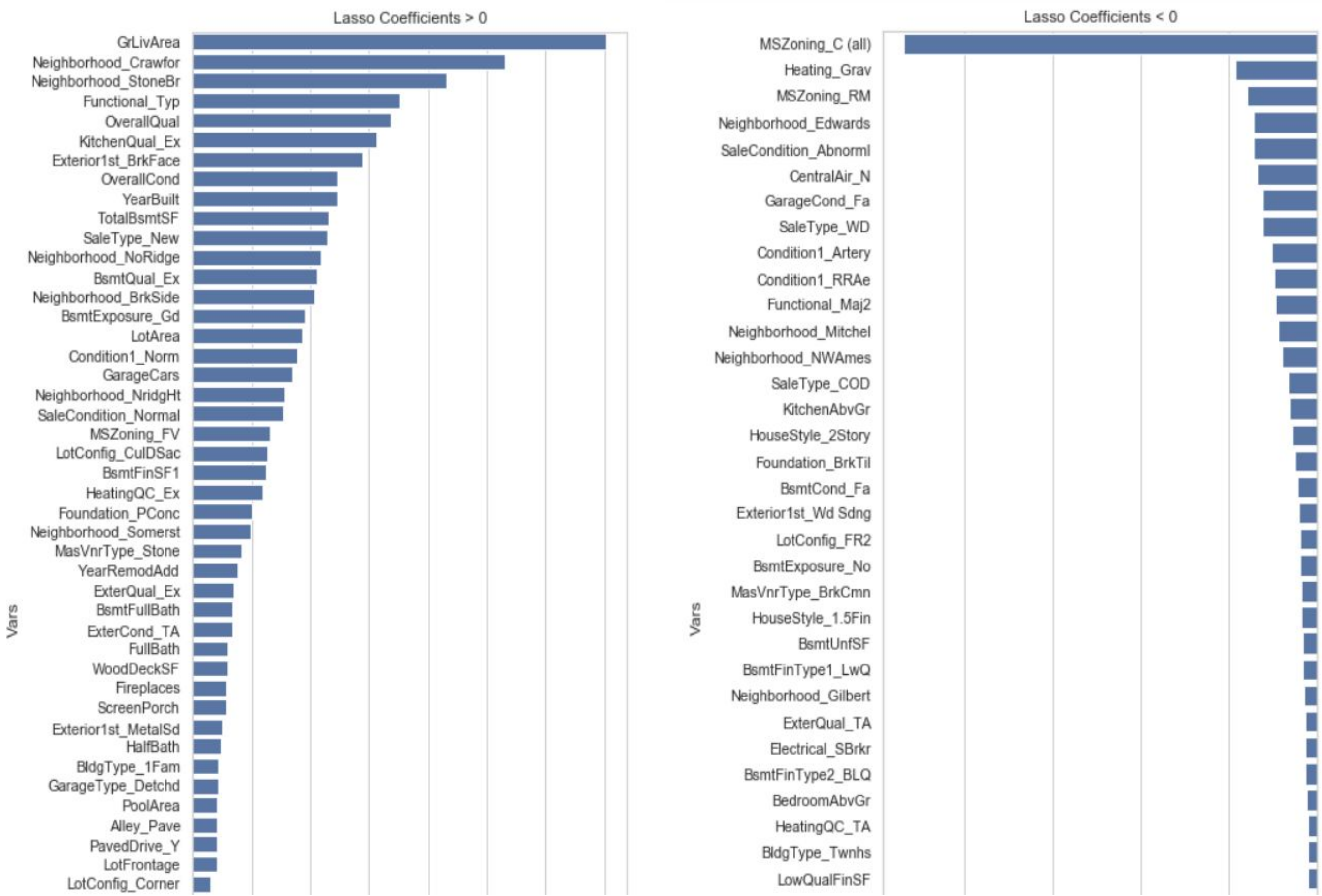



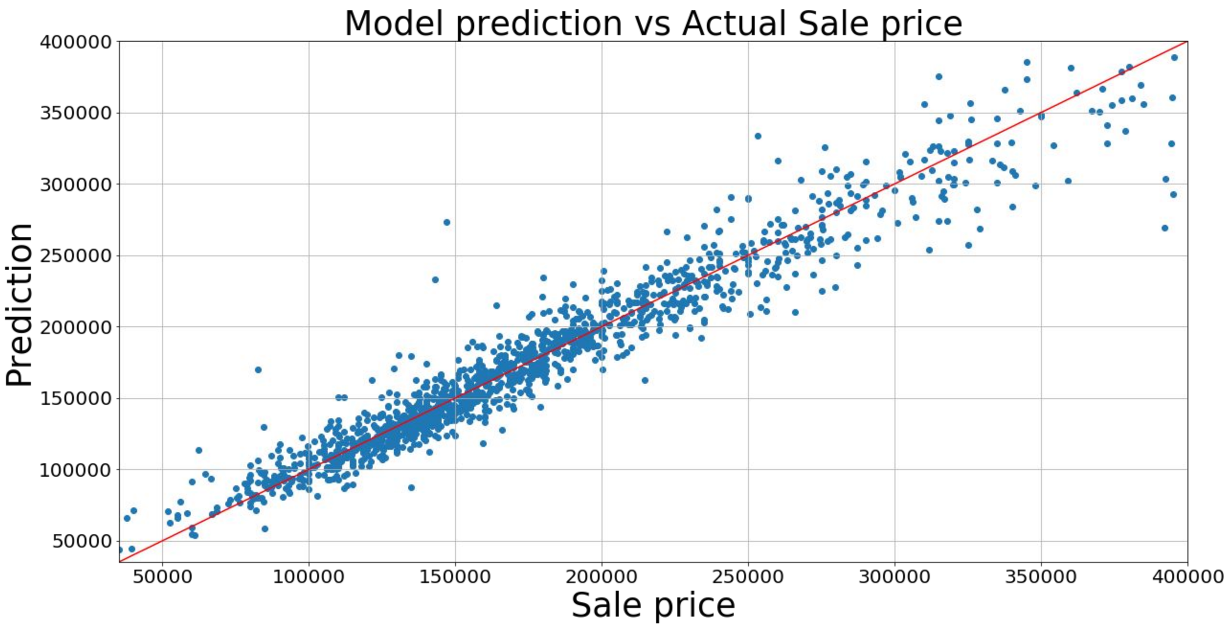
# Lasso Coefficients

Original # of Features:
**285**

# of Features After Lasso:
**105**



# Lasso Predictions on Train



# Model Cross Validation Scores

| Model | Train CV Scores (RMSE) |
| --- | --- |
| Random Forest | 0.1361 |
| XGBoost | 0.1258 |
| **Lasso** | **0.1084** |
| Ridge | 0.1099 |
| Elastic Net | 0.1112 |

# Future Improvement

# Conclusion

- We scored multiple models, but in the end Lasso scored the highest on Kaggle.
    - RMSLE = 0.11544

# Q & A