KAGGLE COMPETITION

# PREDICTING HOUSING PRICES

**Team MealPals**

Christian Opperman
Melanie Zheng
Paul Choi

# Introduction

### Challenge

- Predict home prices in Ames, Iowa using advanced machine learning techniques

### Why?

- You are a Real Estate Investor with the opportunity to invest in a portfolio of 1,459 homes
- This deal has a specific asking price for the entire package of homes and contains underlying data with each house's features
- You need help deciding if the transaction is worth the asking price — in other words, whether it represents a good enough return on investment ("ROI")
- *In this scenario, we assume* ***(1)*** *time is not a factor and* ***(2)*** *these homes will ultimately be re-sold at or near their predicted prices*

# Solving the Problem

## Approach:

- Build a machine learning model using historical housing data (the “train set”) to predict values on another group of homes (the “test set”)
- Use this information to determine a final investment recommendation on the purchase of this group of homes

# Our Solution

## Recommendation:

- Our best model for predicting the true values of the home portfolio came from a weighted average of our Stacked and Light GBM models

| Model | Details | Kaggle Score |
|---|---|---|
| Weighted Average of Best Models | 0.7 Stacked Model + 0.3 Light GBM | 0.12873 |

- Our final recommendation is based on if the deal's "asking" price is above or below our total valuation of **$262 million** based on the aggregated predicted prices for all 1,459 homes

- We include some margin of error in our analysis and exclude any further ROI requirements for the purposes of this example

# Exploratory Data Analysis

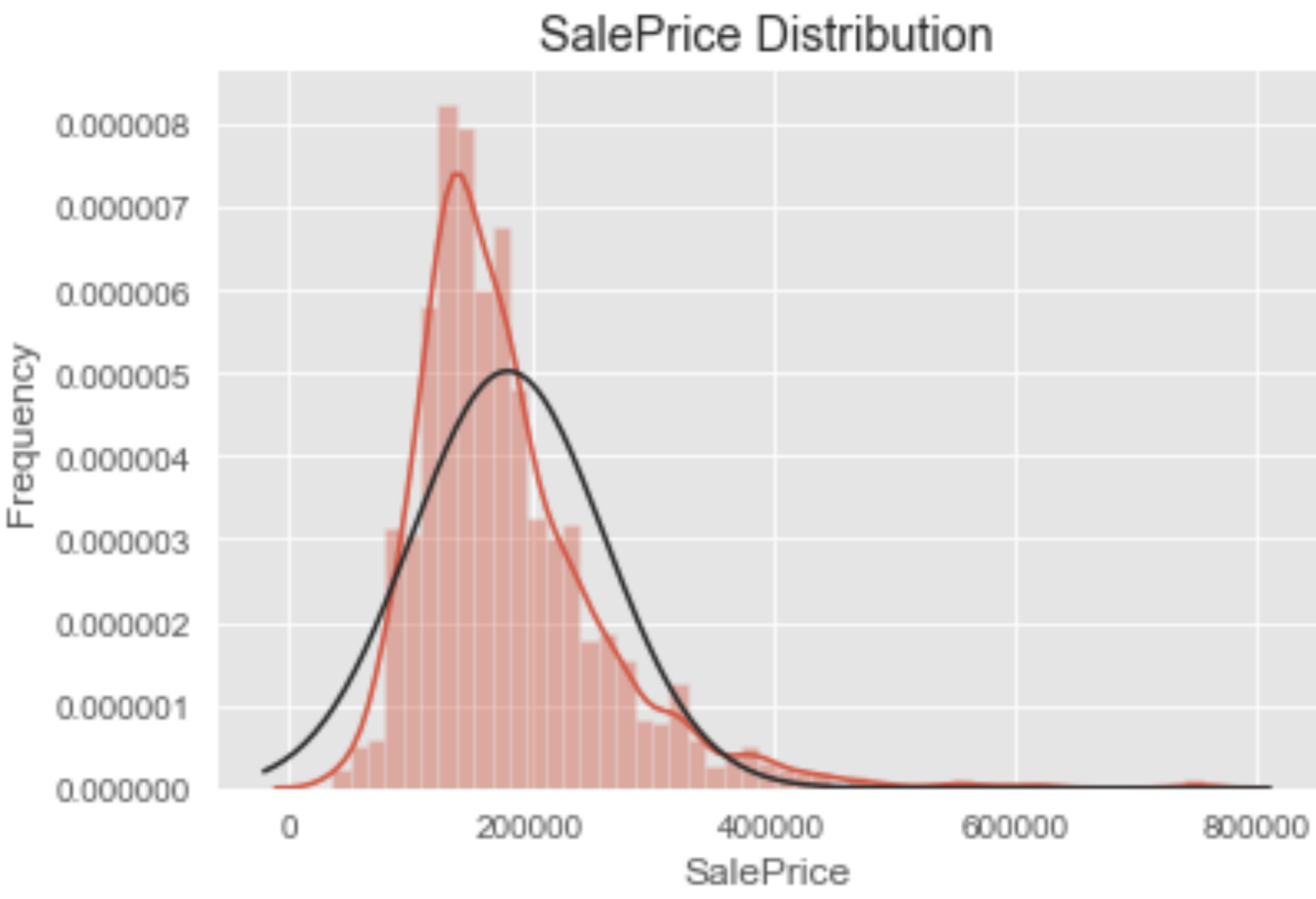
# First Glance at the Kaggle Data

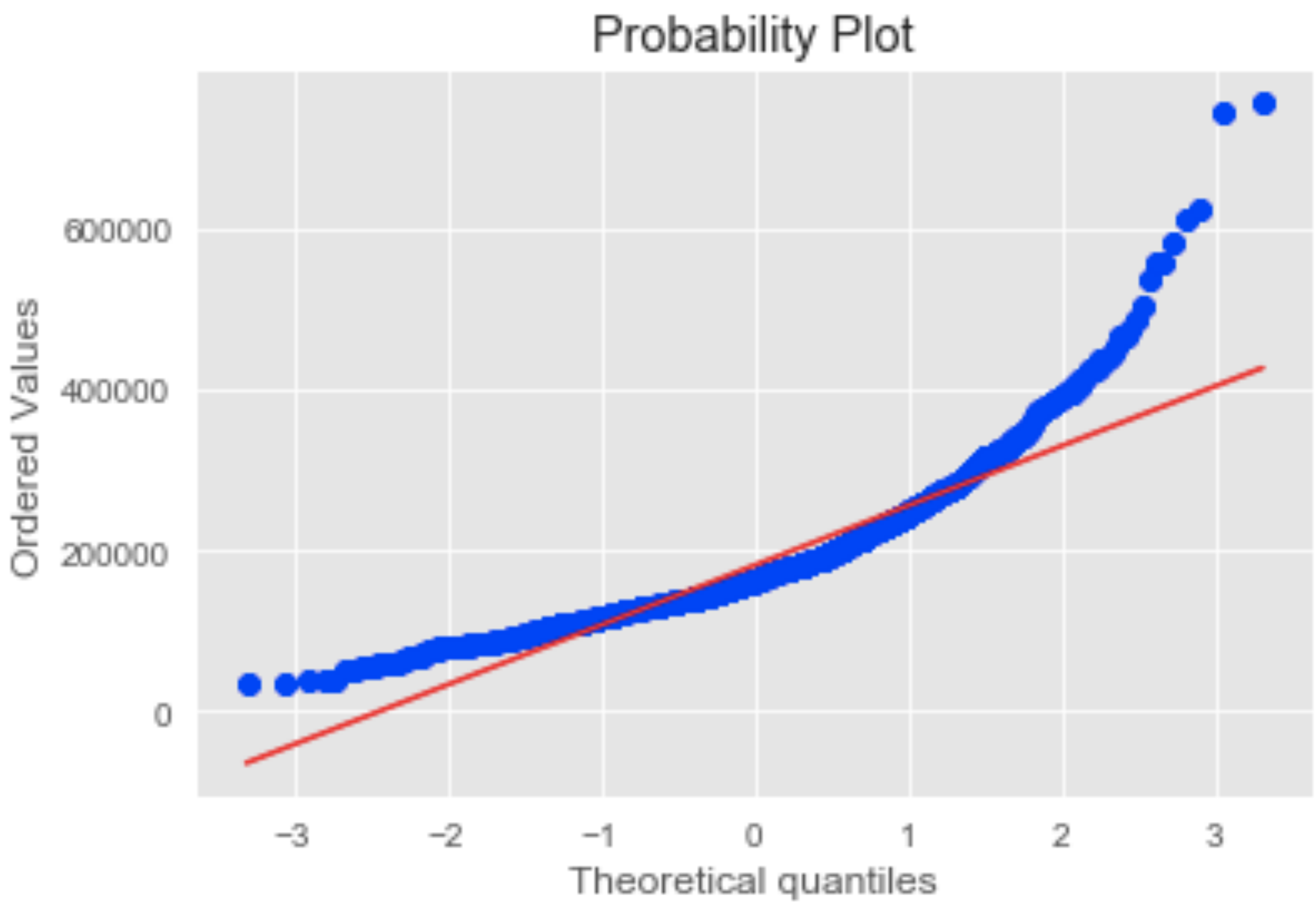
**Two datasets provided**

- Train set
  - 1,460 observations
  - 79 predictor variables (excluding ‘SalePrice, ‘Id’)
    - Numeric variables: 28
    - Categorical variables: 51
- Test set
  - 1,459 observations, 79 predictor variables (excluding ‘Id’)

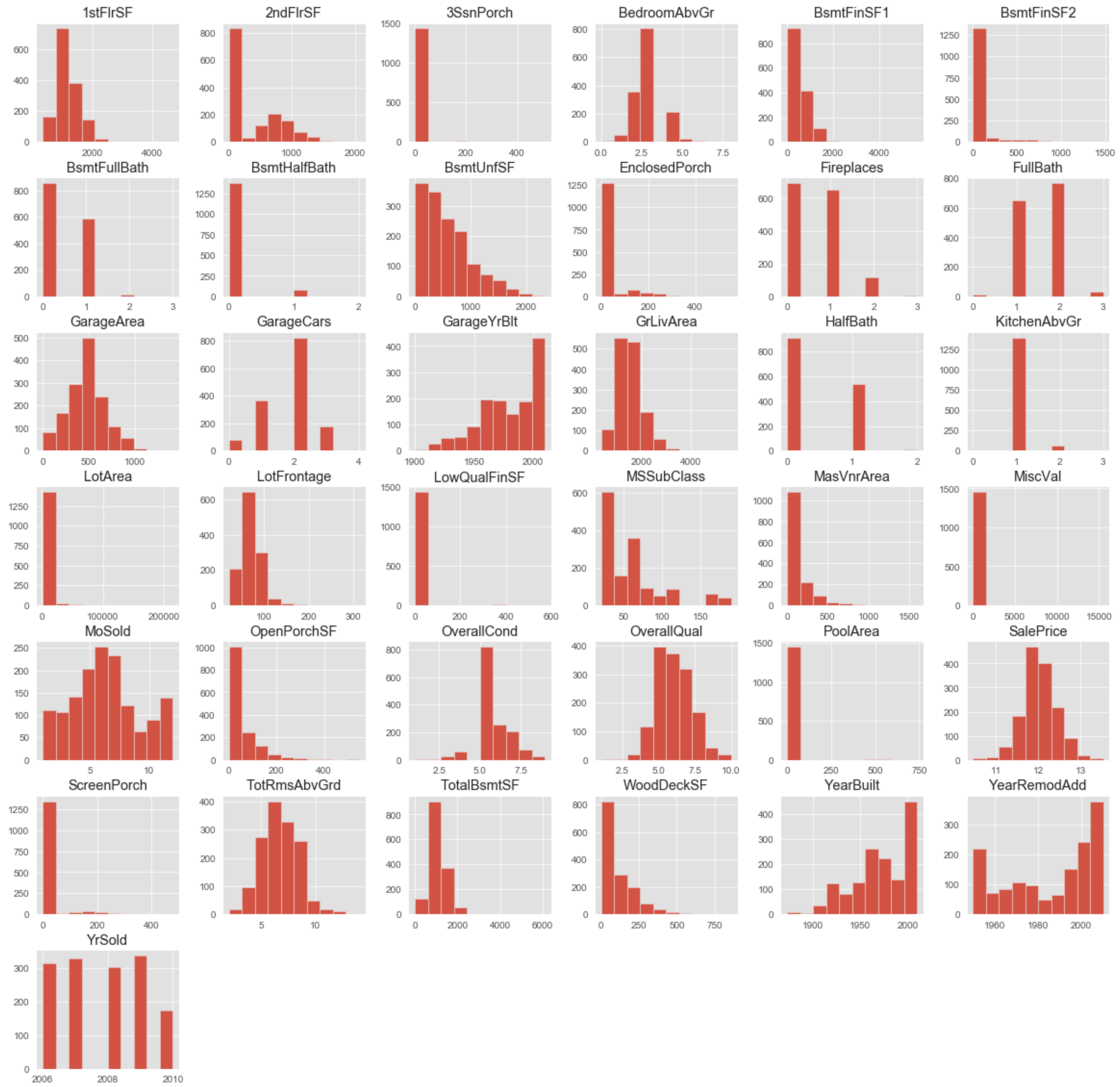
# Exploratory Data Analysis

A quick look at our target variable: **SalePrice**



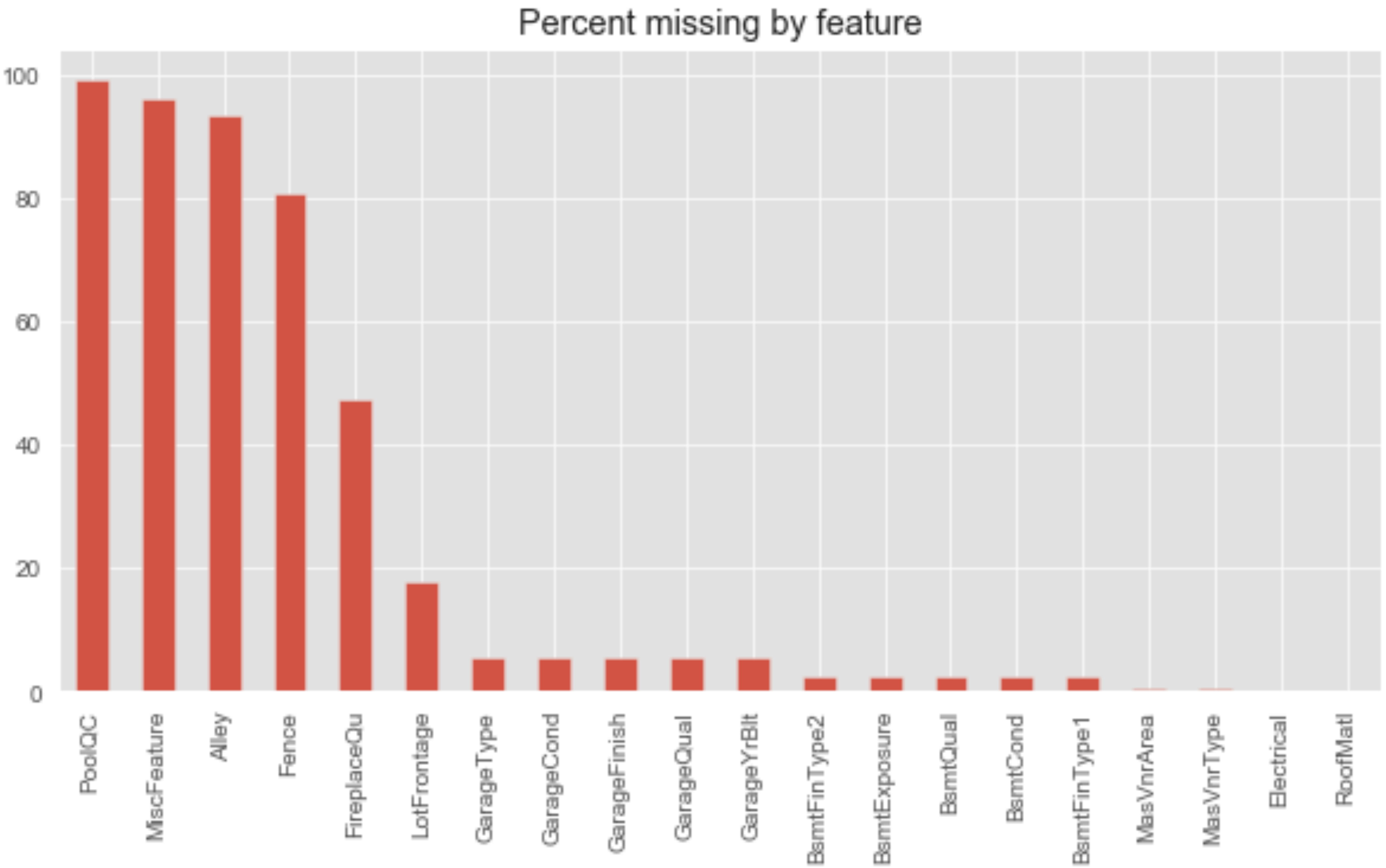


# EDA: Examining Variable Distribution



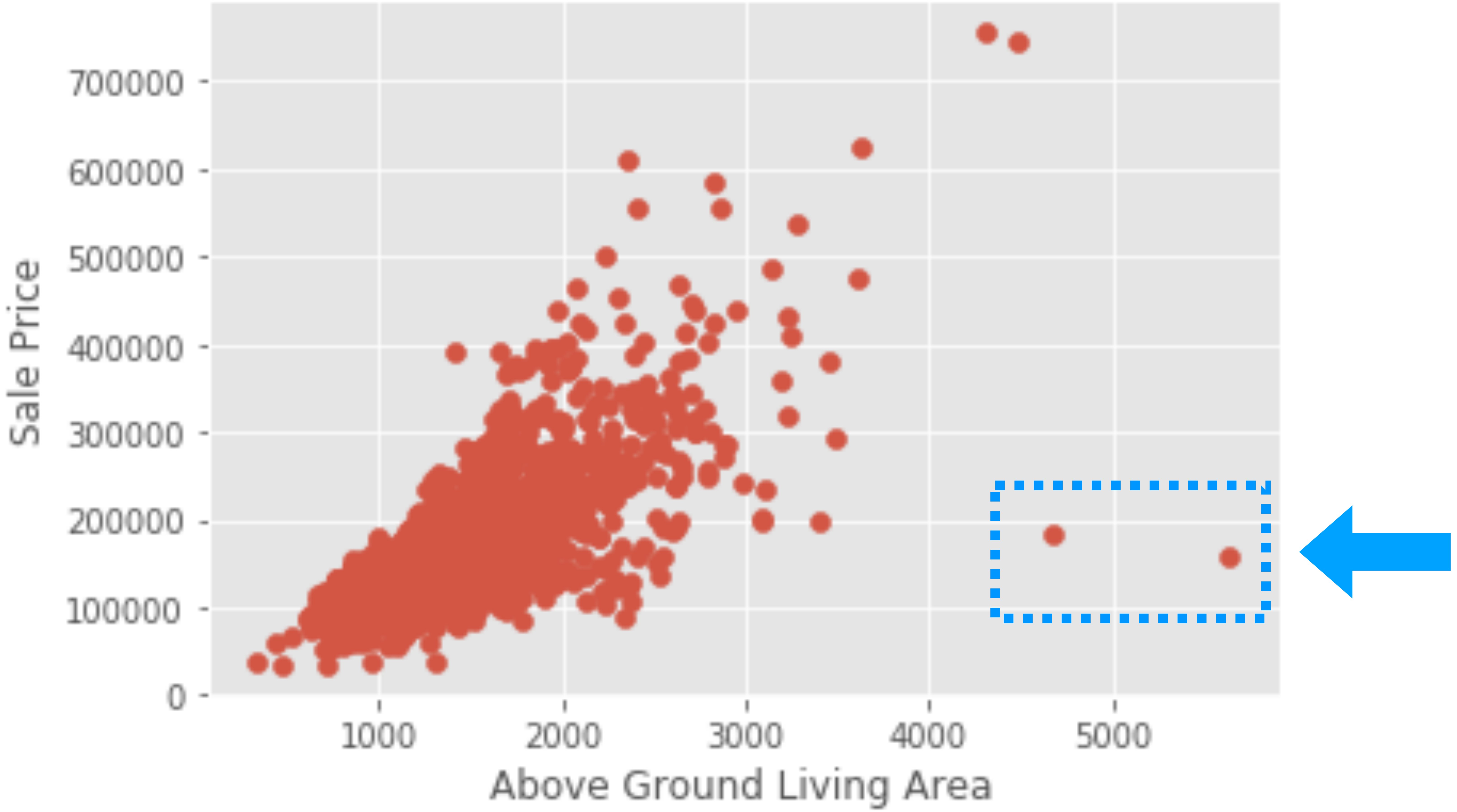
# EDA: Looking for Missingness

Check the proportion of missing data by variable



# EDA: Identifying Outliers

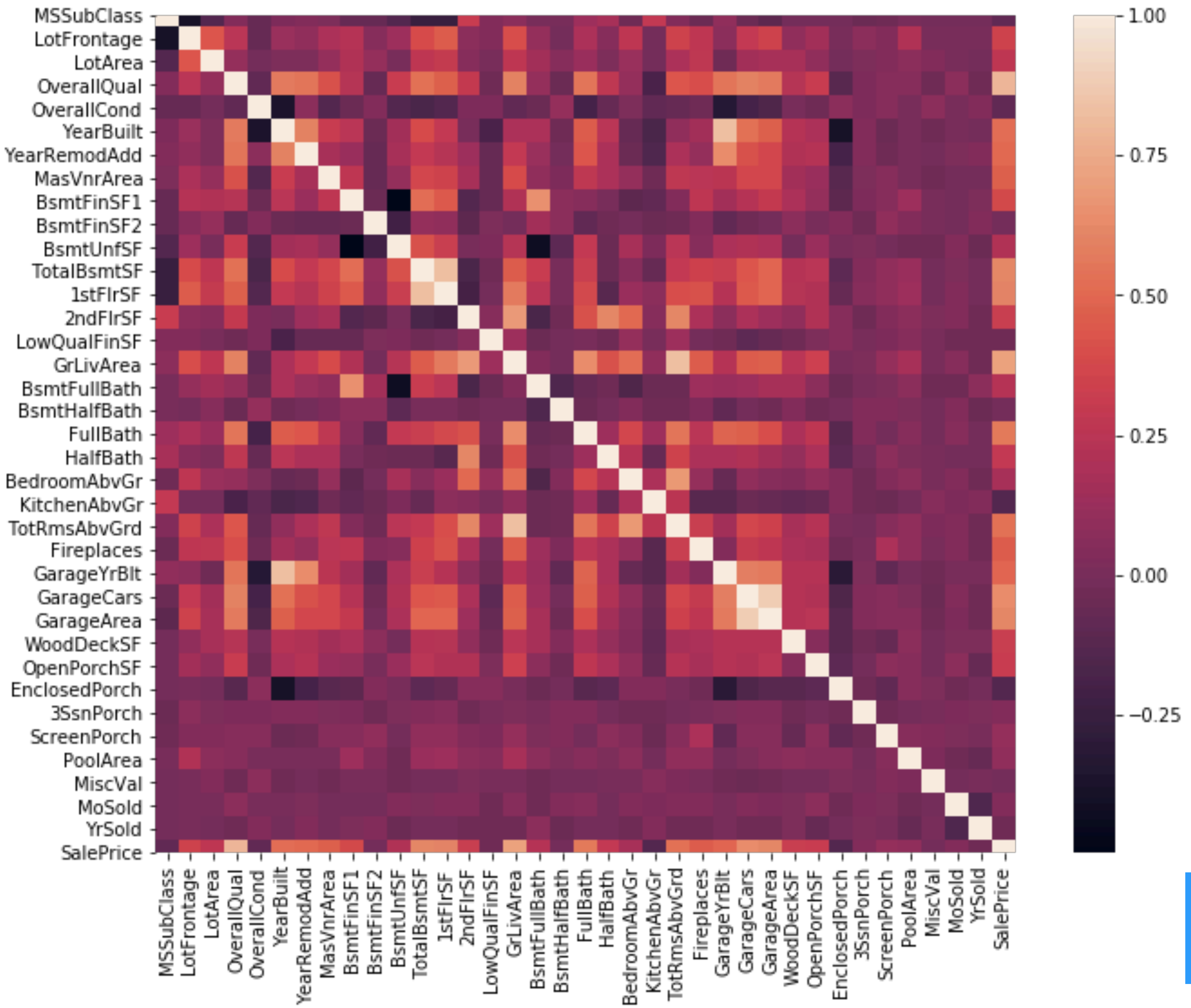
Isolating and removing outliers



# EDA: Examining Multicollinearity

Examples of variables with high correlation with each other:

- 1stFlrSF & TotalBsmtSF
- TotRmsAbvGrd & GrLivArea
- GarageArea & GarageCars
- YearBuilt & GarageYrBuilt



# Data Preprocessing

# Data Preprocessing Methodology

1. Remove outliers in training data
2. Impute data:
    - Impute pseudo-missing values
    - Impute true-missing values
    - Re-engineer categorical features as necessary
3. Add new feature variables
4. Dummify categorical feature variables
5. Remove redundant feature variables

# Imputation: Pseudo vs. Actual Missingness

- There were a number of feature variables that contained missing values that did not, in fact, represent missing data.
- The general philosophy for a given variable X (where X represents a housing feature such as a pool, a fireplace, etc.) with this “pseudo-missingness” was to impute missing values as “No X”

**Pseudo Missing Values**

Alley, BsmtCond, BsmtQual, BsmtFinType1, Fence, Fireplace, GarageCond, GarageFinish, GarageQual, GarageType, GarageYrBlt, MasVnrType, MiscFeature, PoolQC

**Actual Missing Values**

Electrical,
MasVnrArea,
LotFrontage,
BsmtExposure,
BsmtFinType2

# Feature Engineering

## Generated New Features

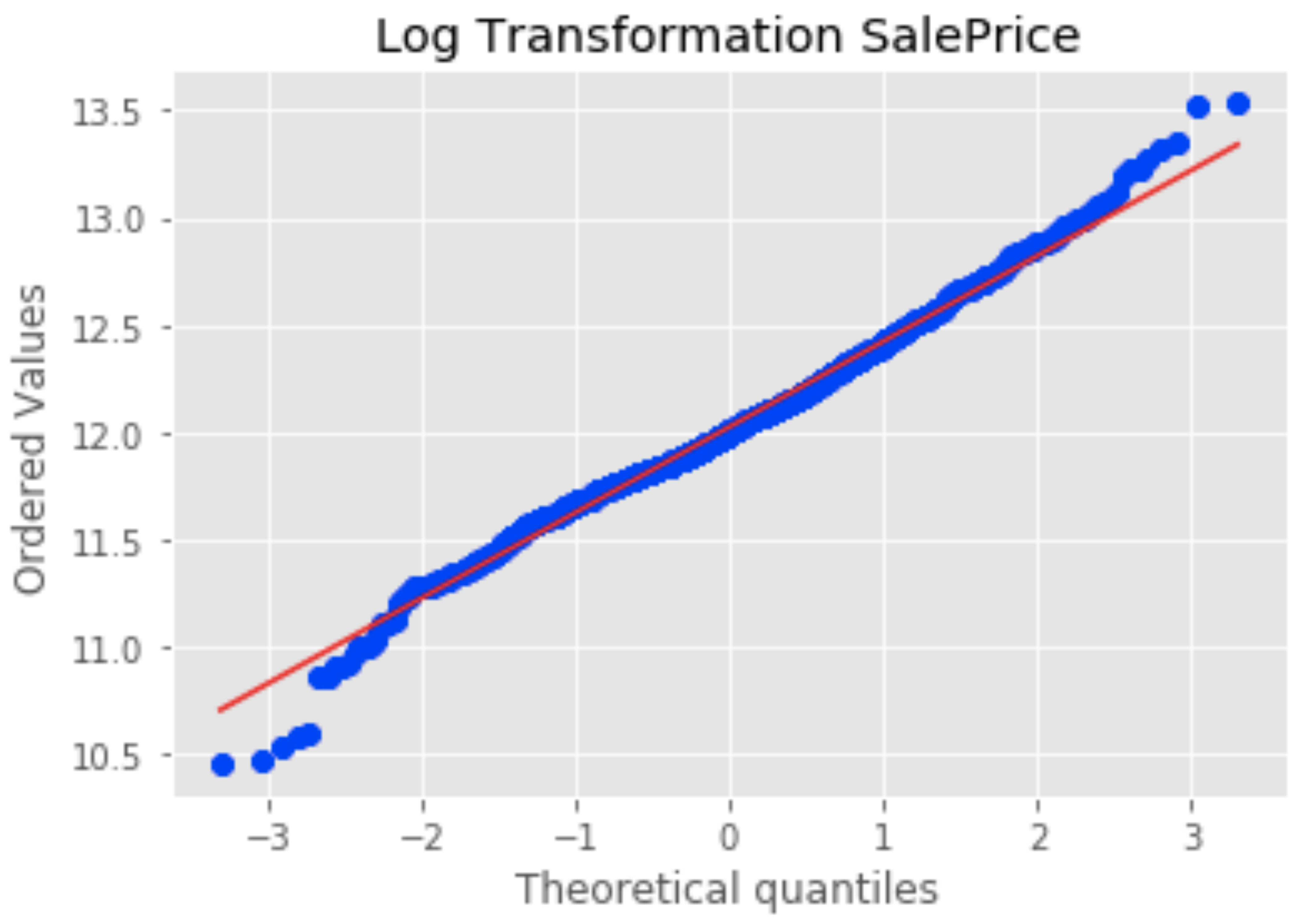
- Total SF = 1stFlrSF + 2ndFlrSF + TotalBsmtSF — Numeric
- TotalFullBath = BsmtFullBath + FullBath — Numeric
- TotalHalfBath = BsmtHalfBath + HalfBath — Numeric
- IsPool — Categorical
- IsGarage — Categorical

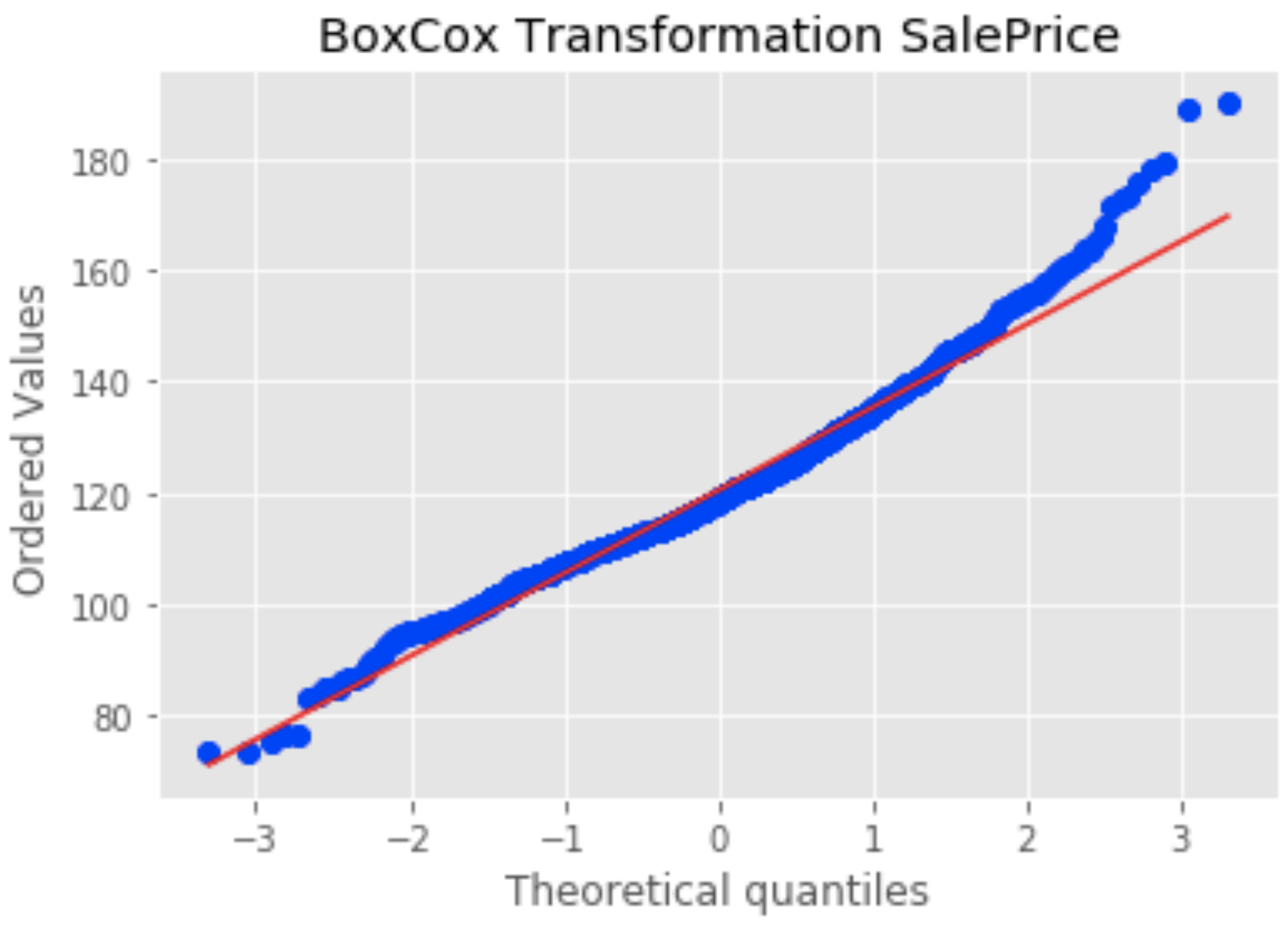
## Dummified Categorical Feature Variables

## Removed Redundant Variables

# SalePrice Transformation: Log vs. Box-Cox

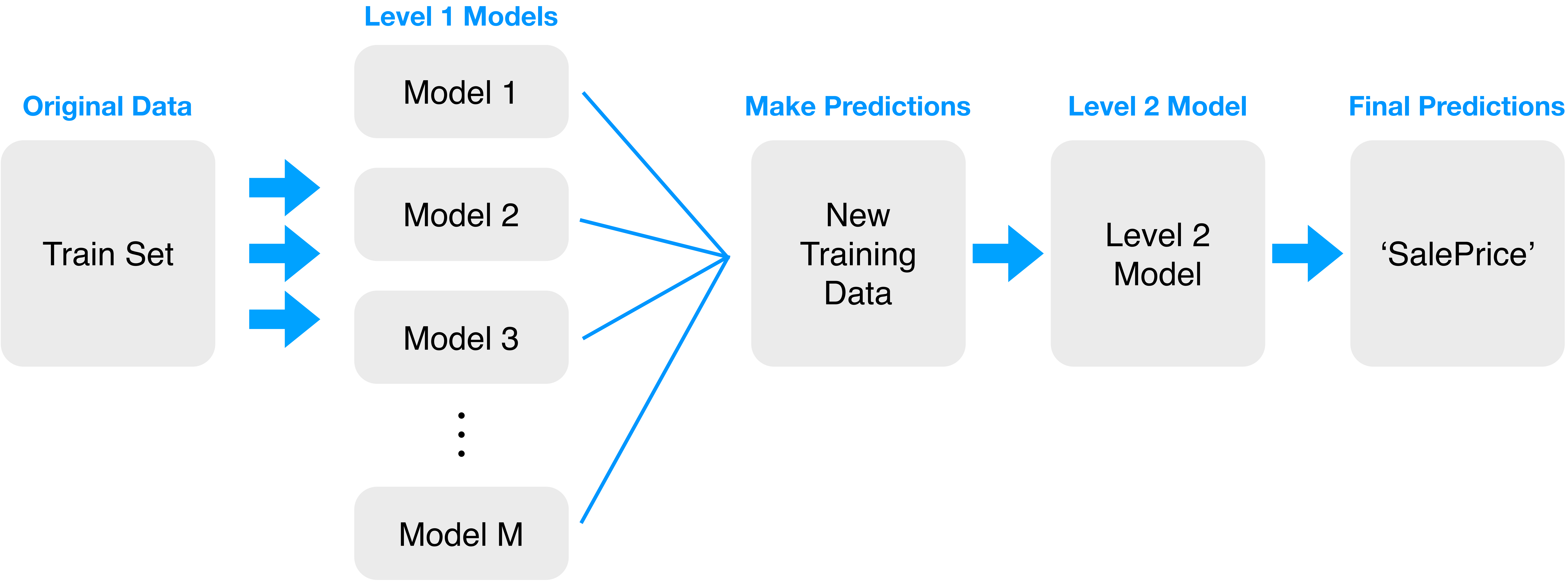
We decided to utilize a Box-Cox transformation over a Log transformation as it provided slightly better predictive power


Log Transformation SalePrice


BoxCox Transformation SalePrice

# Models

# Modeling Methodology

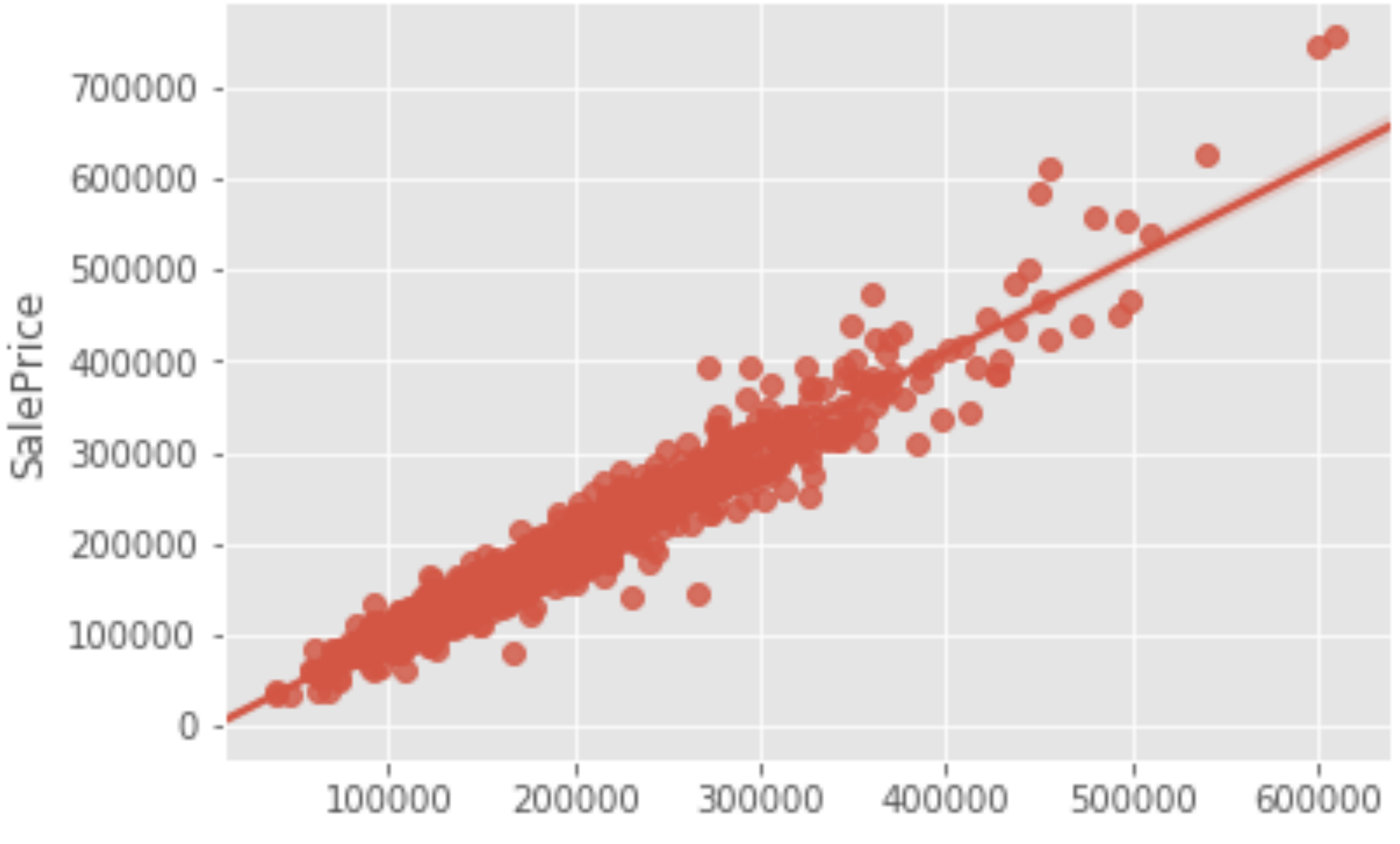


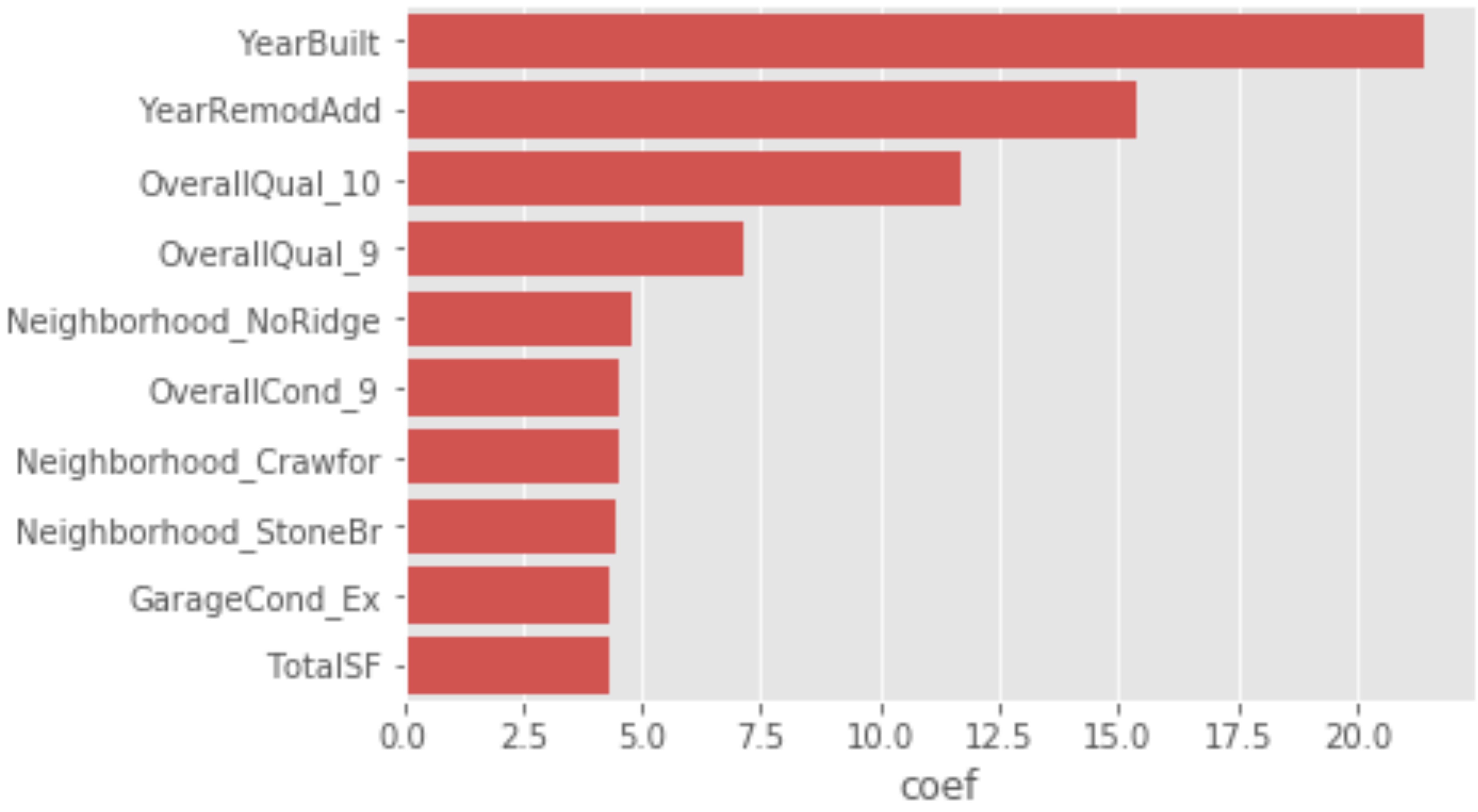
# Models Tested

Kaggle Competition I 2020

| Model Level | Model | Cross Validation Score (R^2) | Kaggle Score |
|---|---|---|---|
| Level I | Ridge | 0.11739 | 0.14049 |
| | Lasso | 0.11413 | 0.13839 |
| | Elastic Net | 0.11800 | 0.13974 |
| | Catboost | 0.11678 | 0.13343 |
| | Gradient Boost | 0.11426 | 0.13435 |
| | XGBoost | 0.11855 | 0.13528 |
| | LightGBM | 0.11853 | 0.13123 |
| Level II | Simple Average | 0.11021 | 0.13265 |
| | Stacked Models | 0.10989 | 0.13169 |
| | | 0.10959 | 0.13156 |
| Final Submission | 0.7 Stacked + 0.3 XGBoost | N/A | 0.12873 |

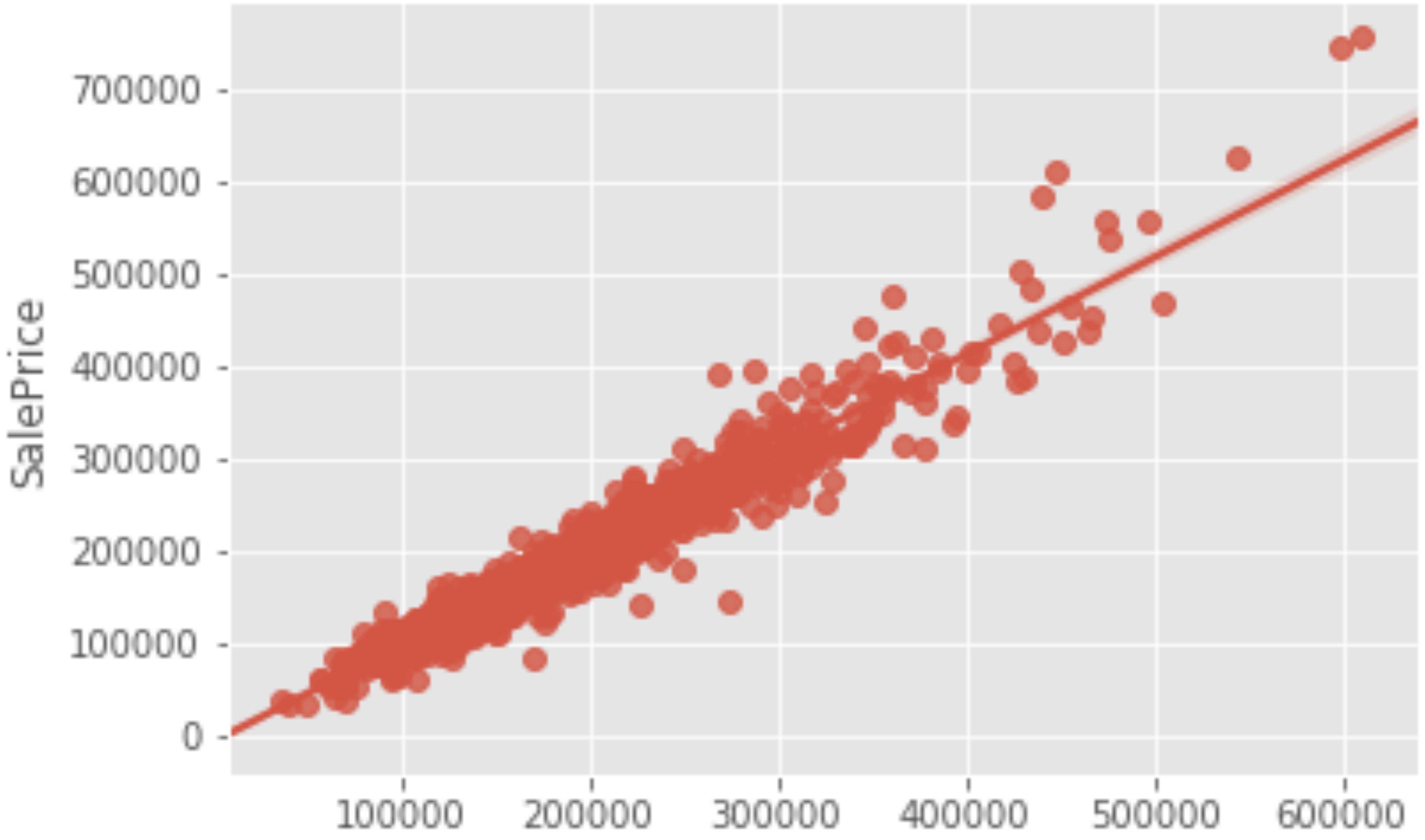
# Model 1: Ridge

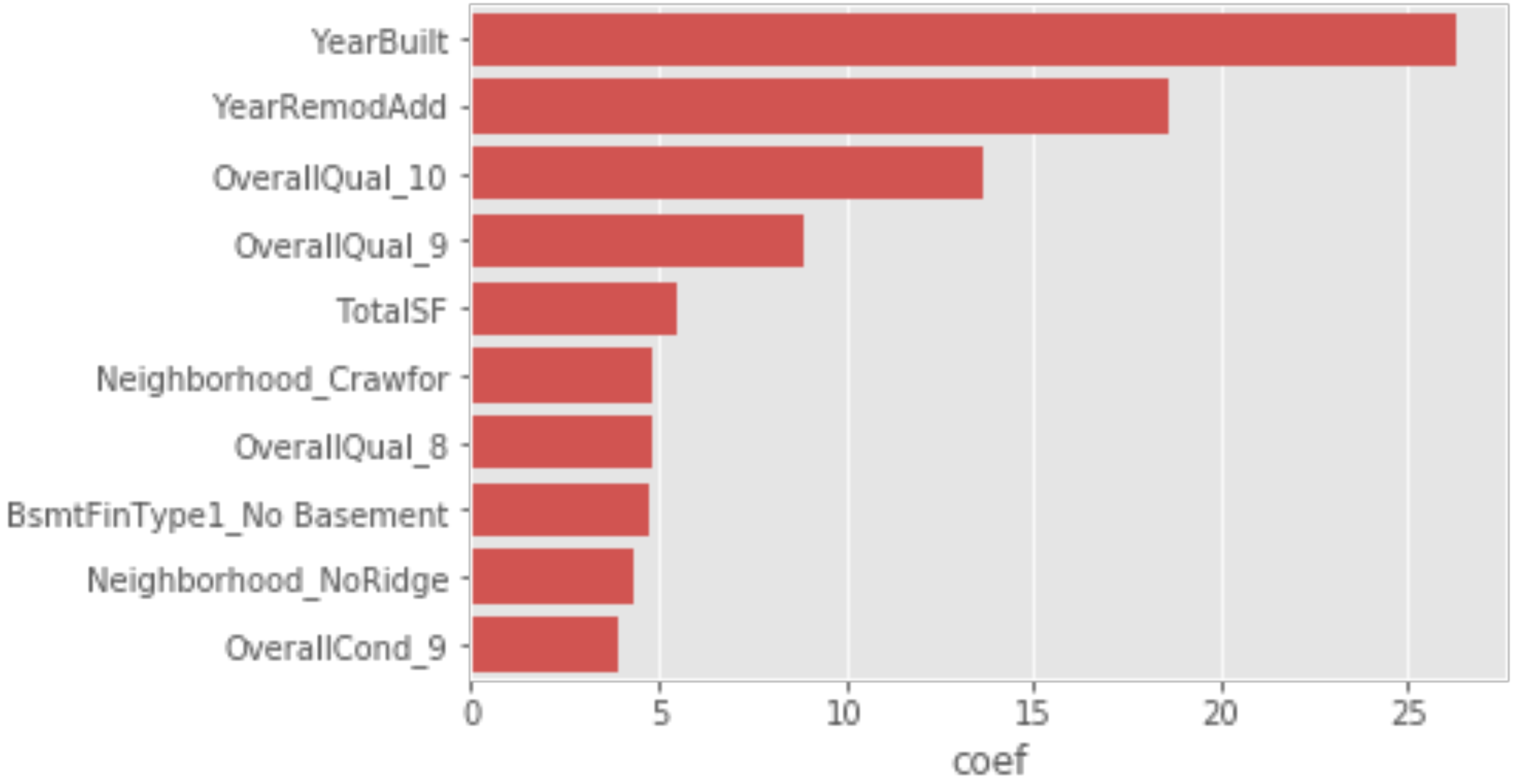




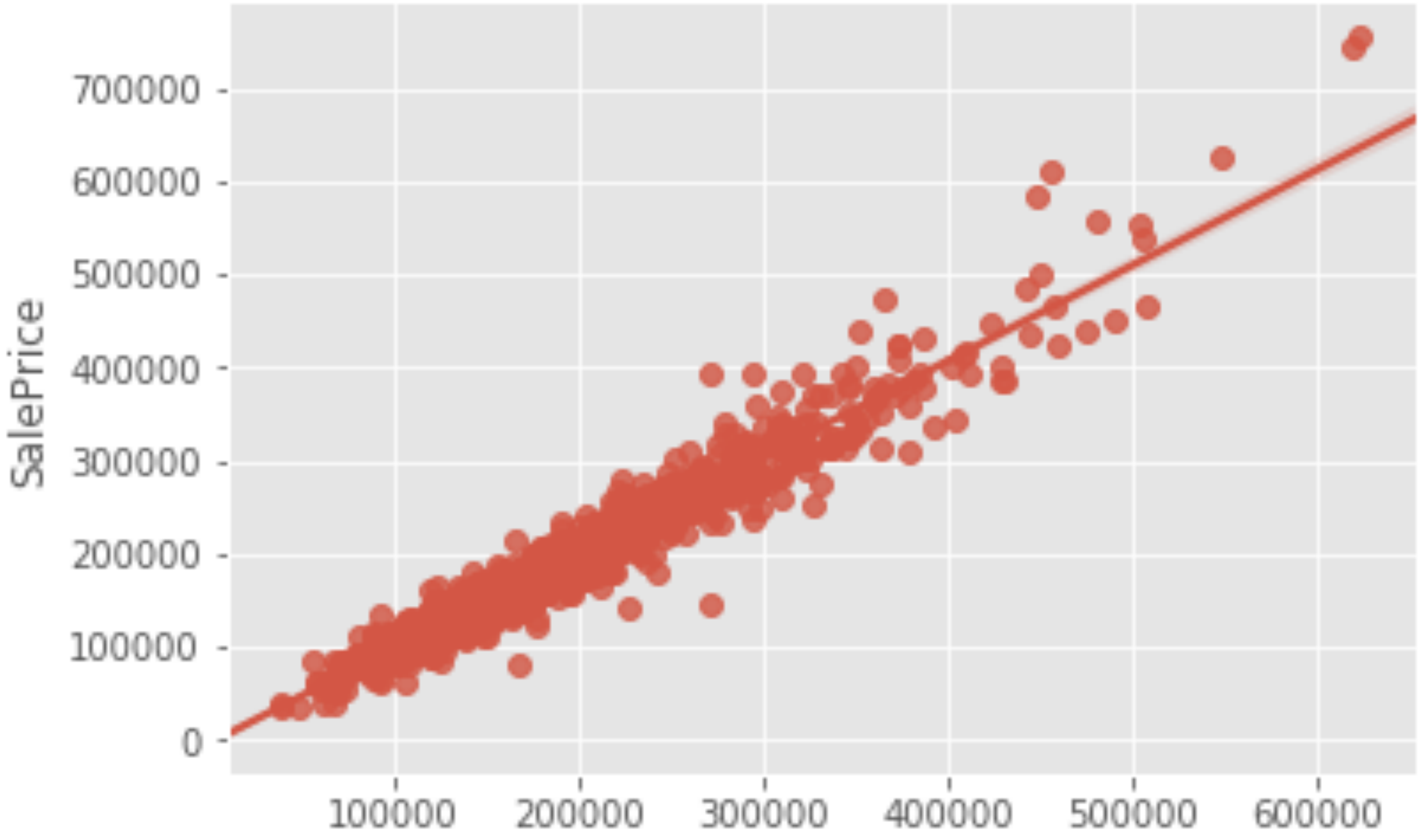
# Model 2: Lasso

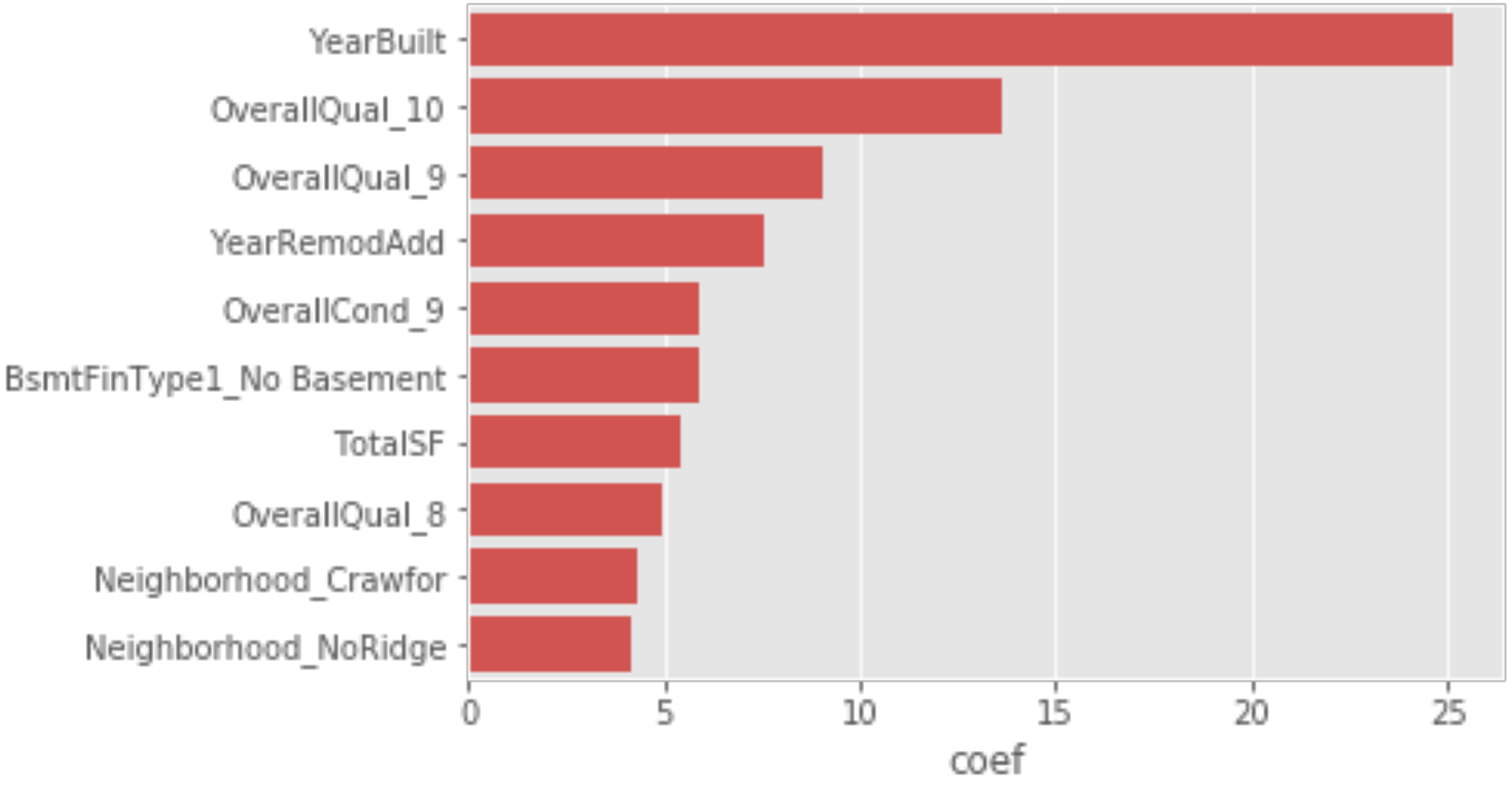
Kaggle Competition I 2020



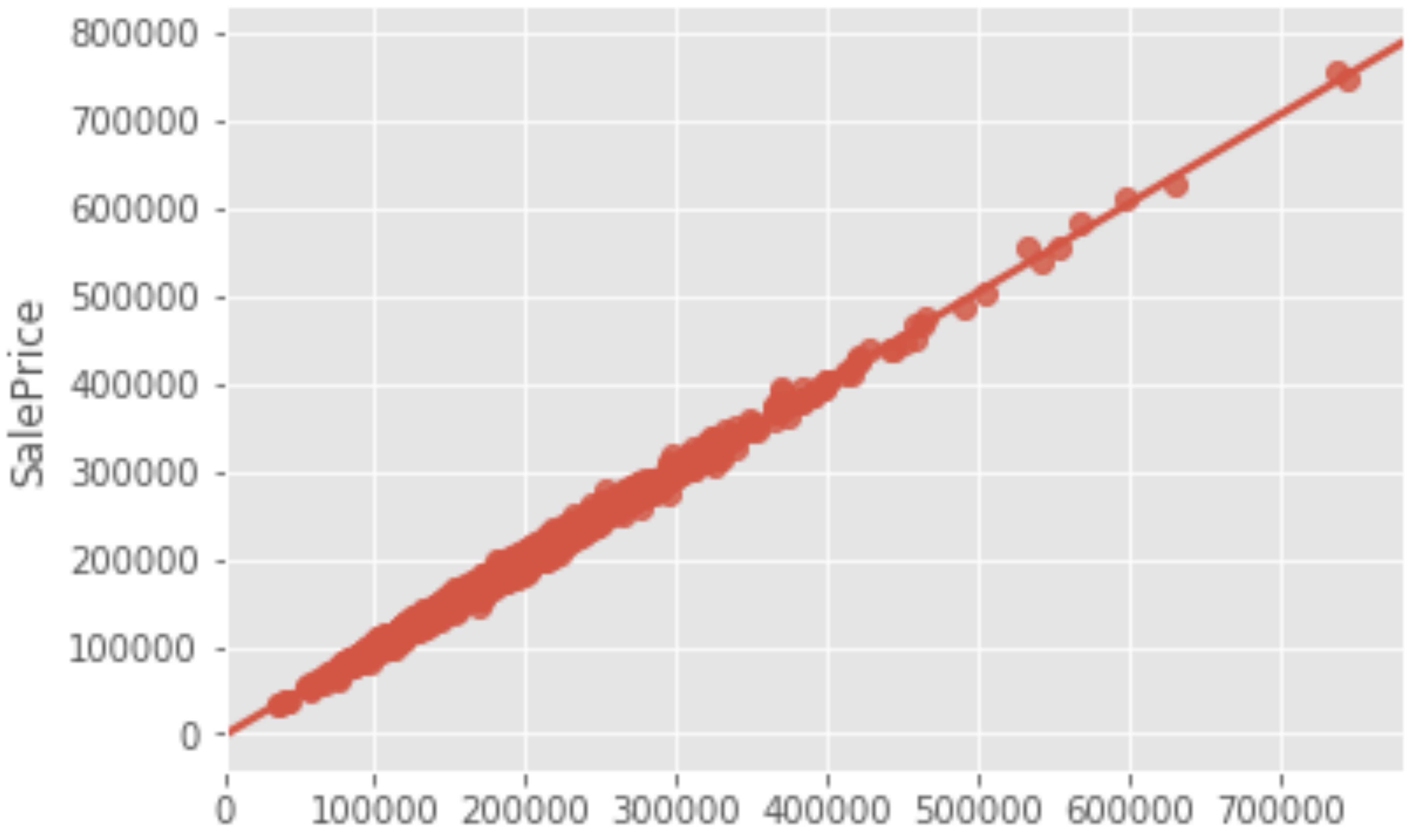


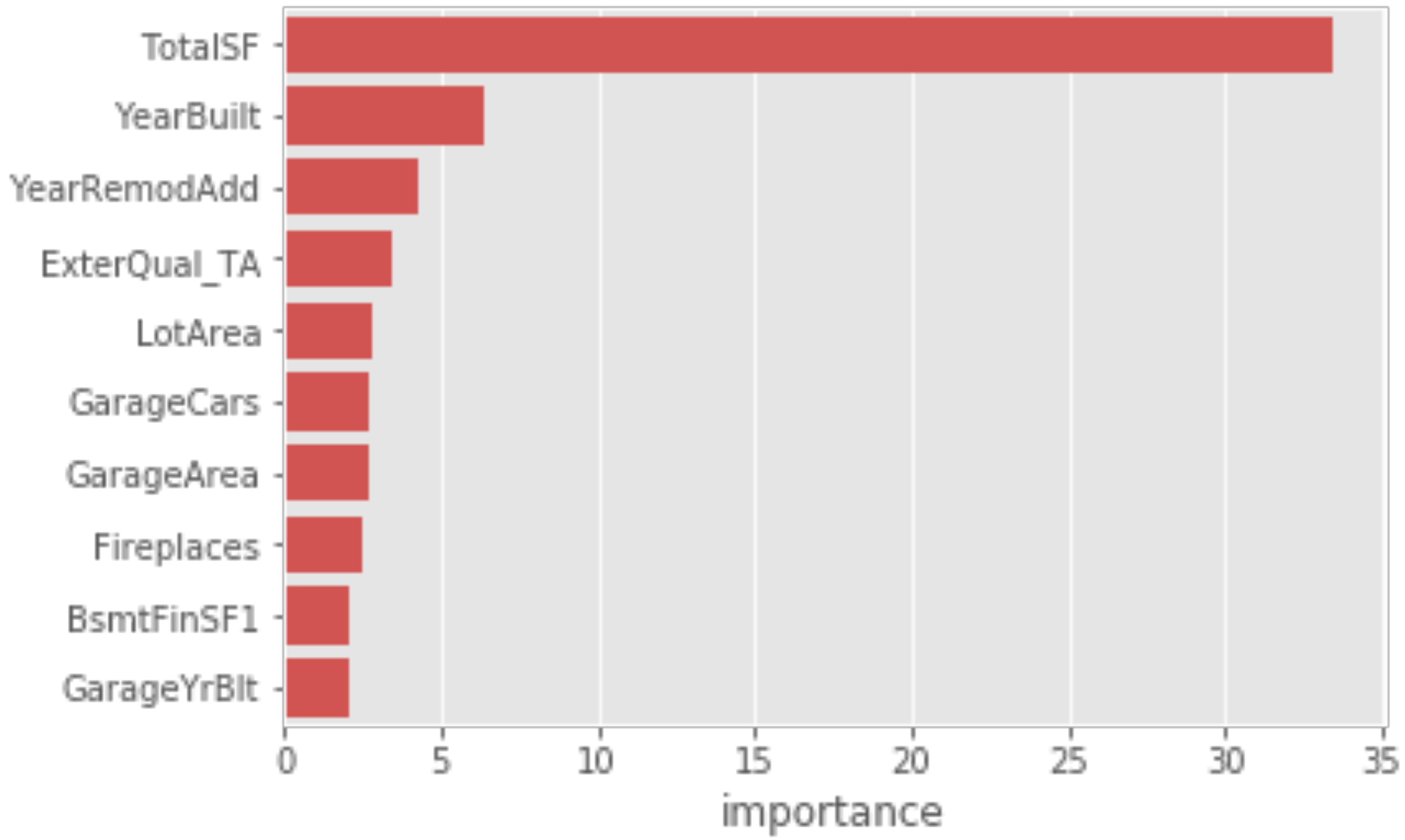
# Model 3: Elastic Net



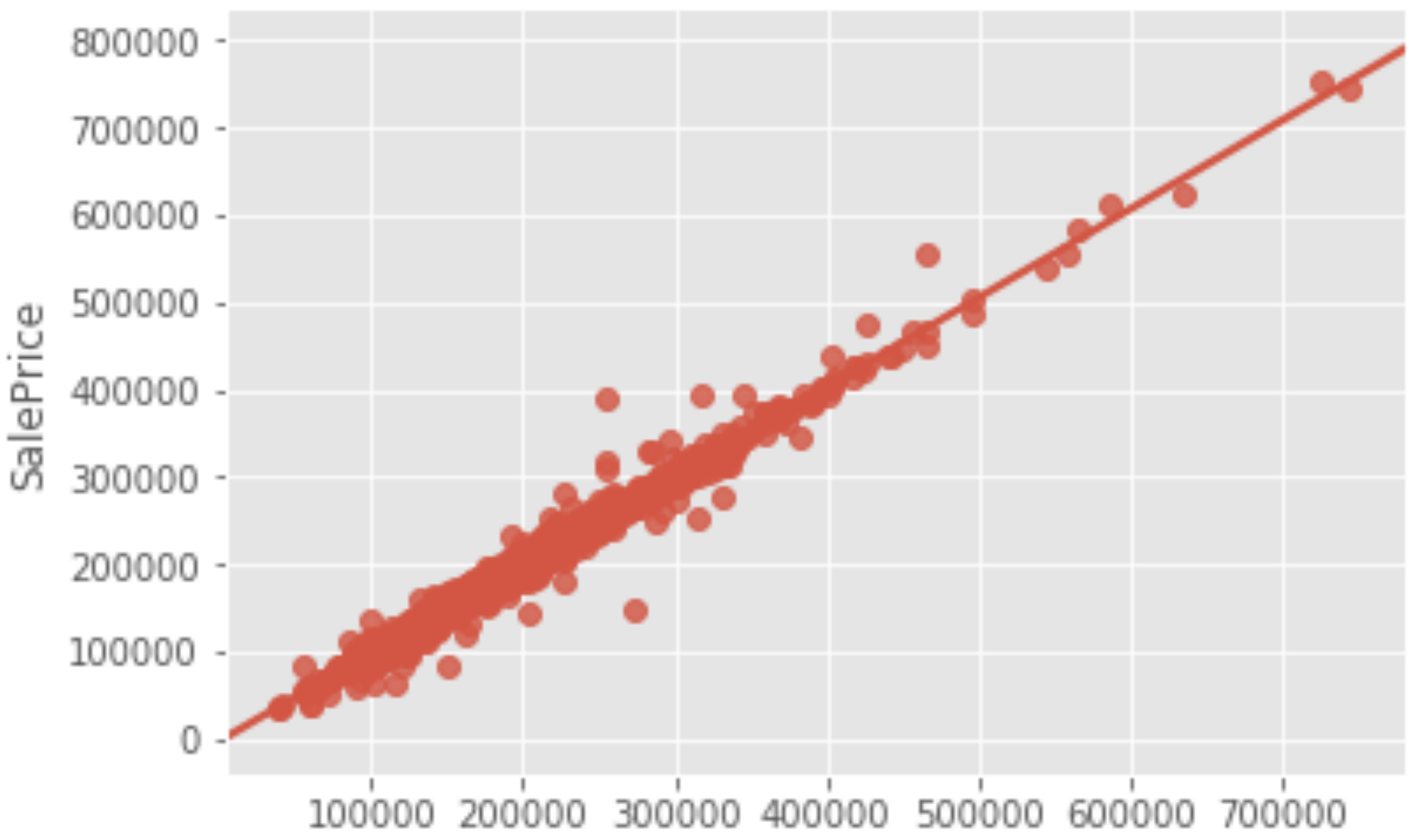


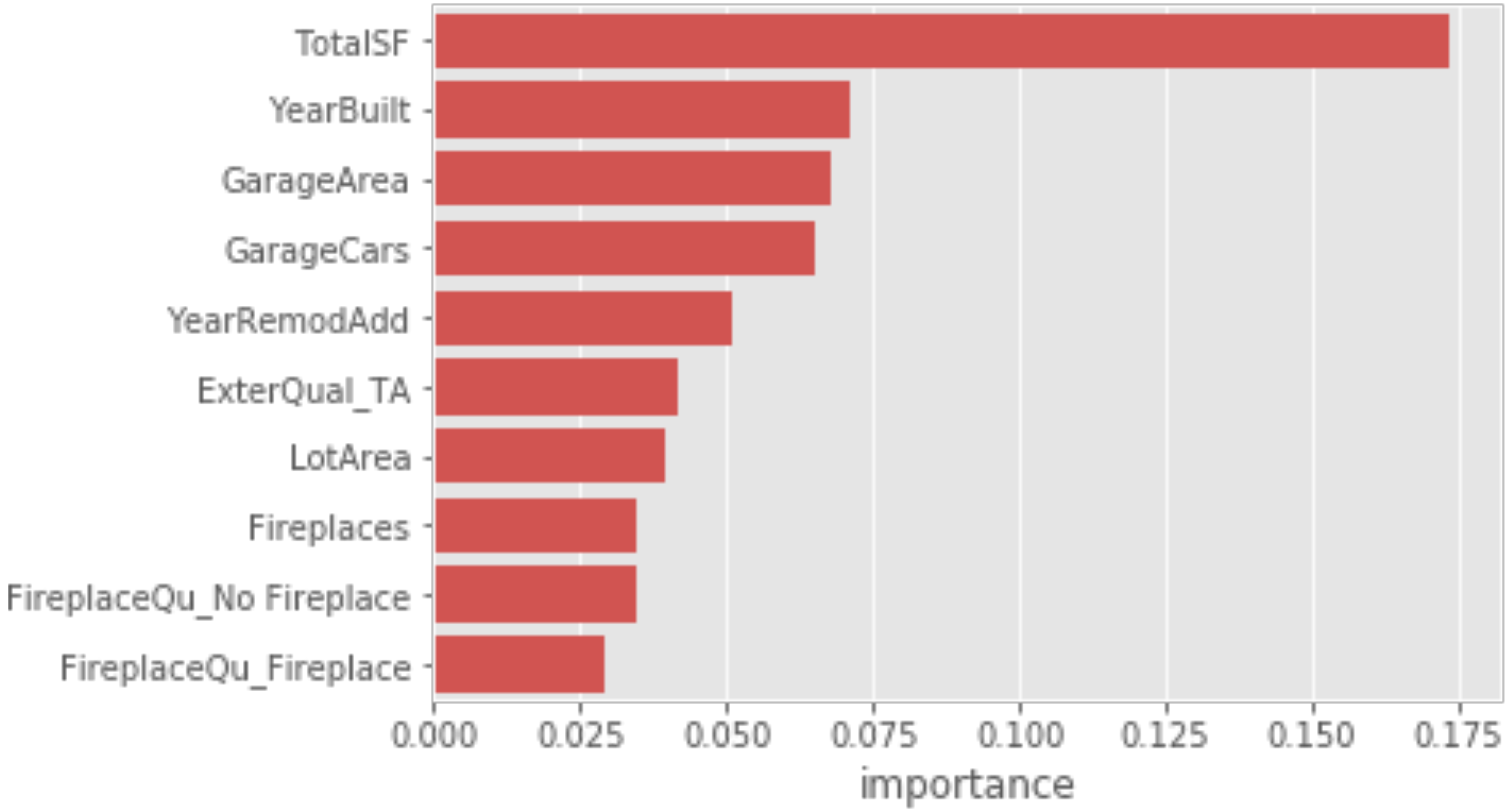
# Model 4: Catboost



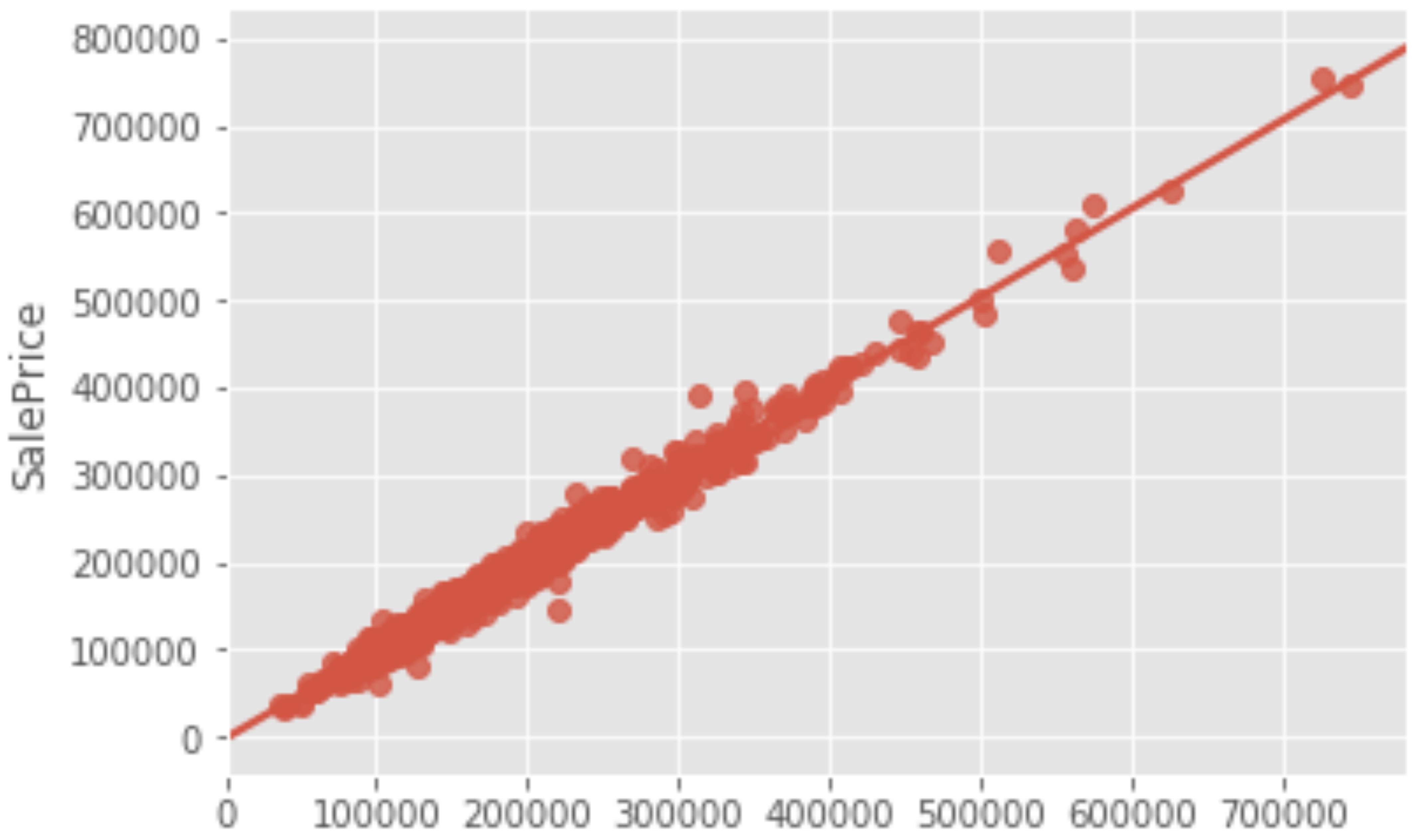


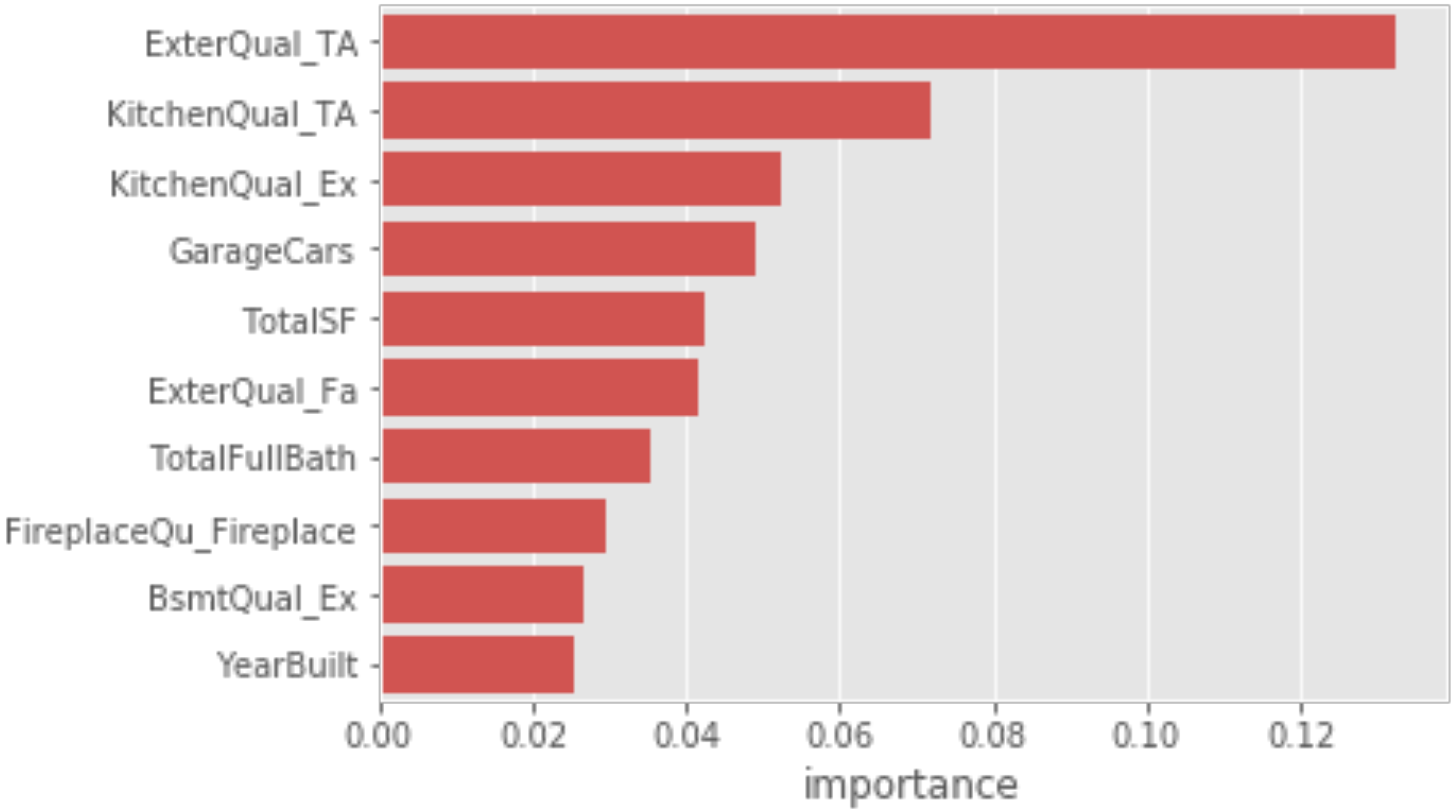
# Model 5: Gradient Boost



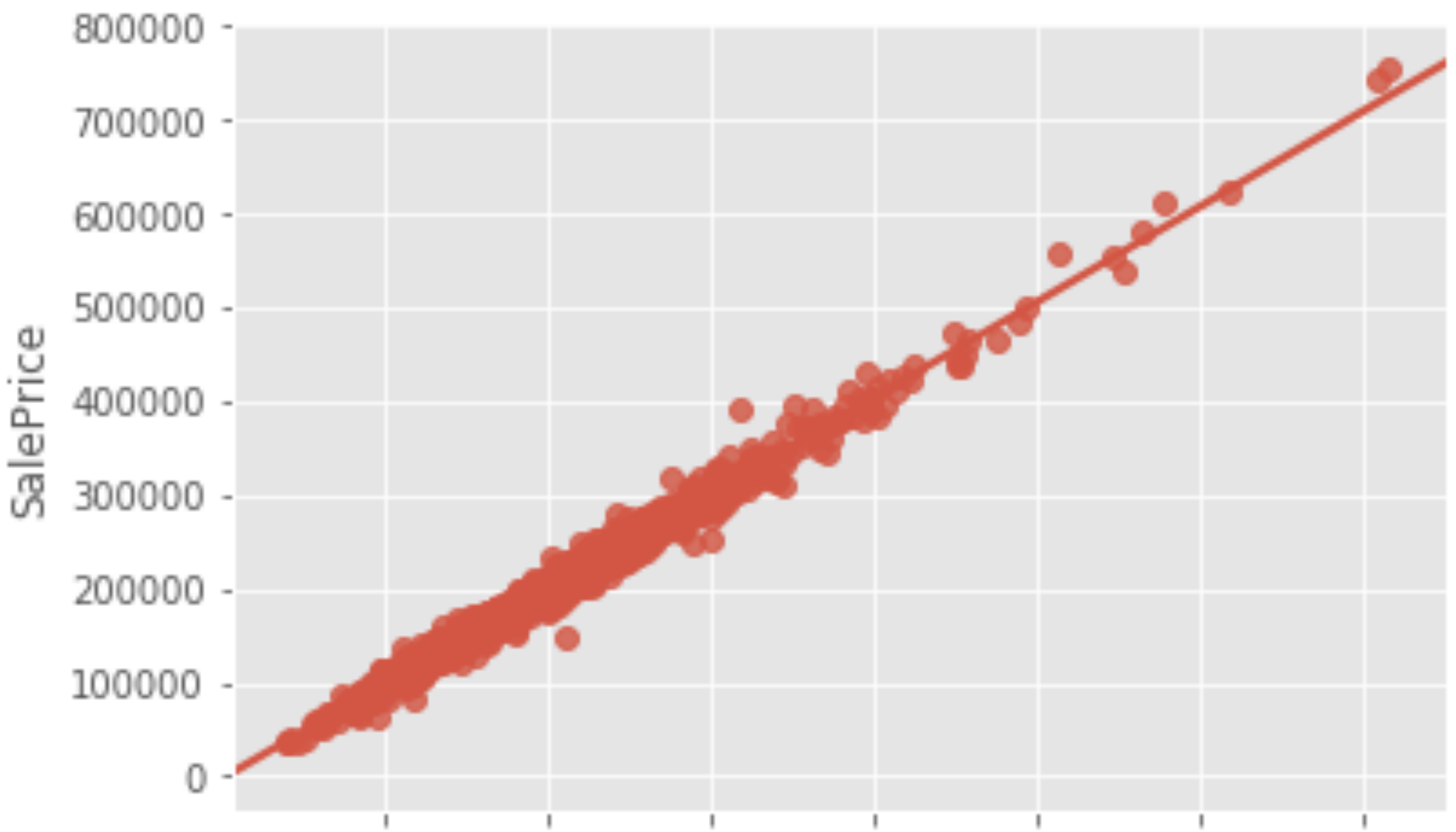


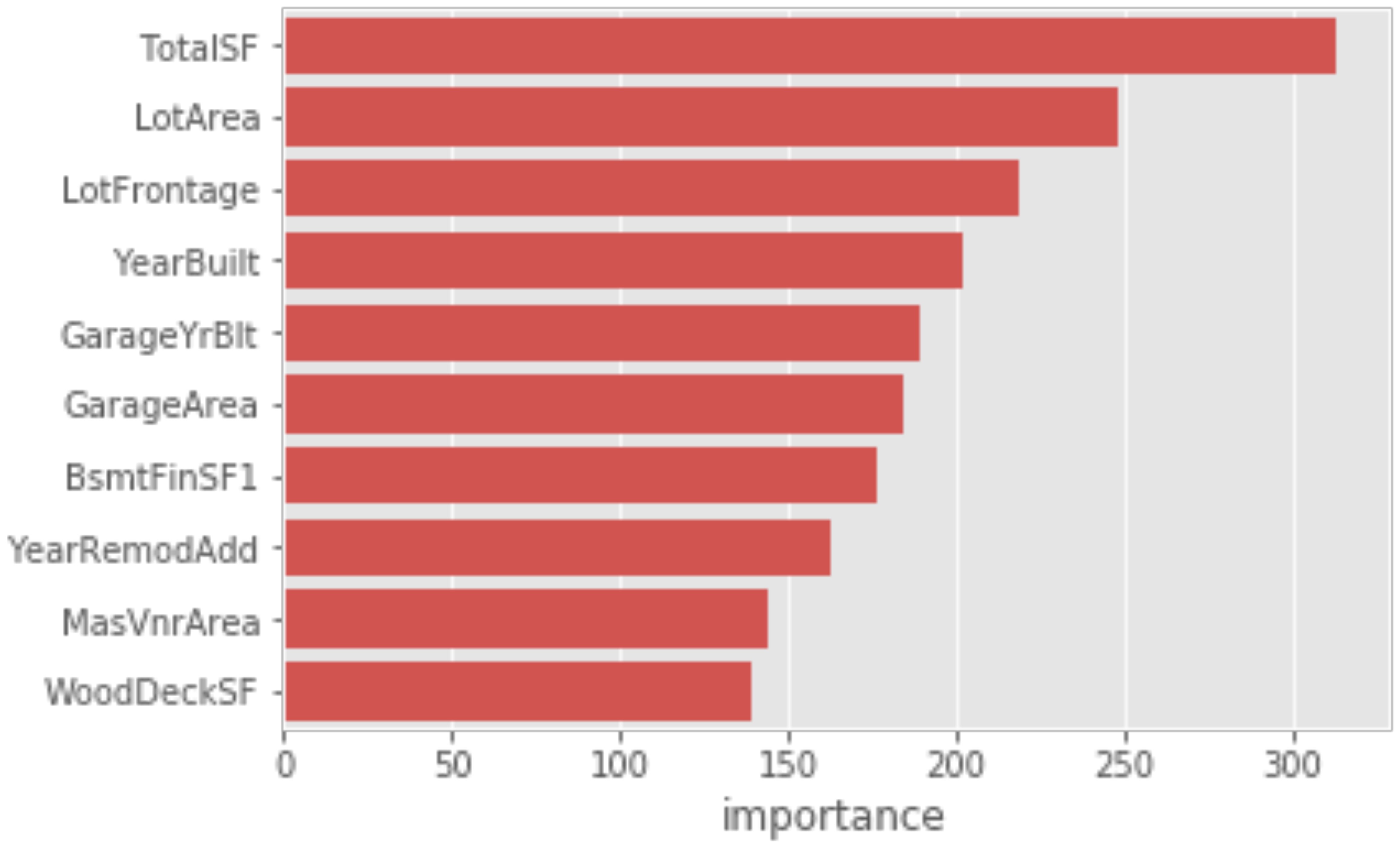
# Model 6: XGBoost





# Model 7: LightGBM





# Conclusion

## Recommendation:

- Our best predictions of home portfolio value came from a weighted average of our Stacked and Light GBM models

| Model | Details | Kaggle Score |
|---|---|---|
| Weighted Average of Best Models | 0.7 Stacked Model + 0.3 Light GBM | 0.12873 |

- This model performance is based on the Kaggle Score, which is the RMSE (root mean square error) of the log of the predictions and the log of the actual sales prices

## Questions?

Thank You!