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 our Stacked model

Model	Cross Validation Score (RMSE)	Kaggle Score
Stacked	0.10895	0.11855

- Our final recommendation is based on if the deal's "asking" price is above or below our total valuation of **$261.5 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

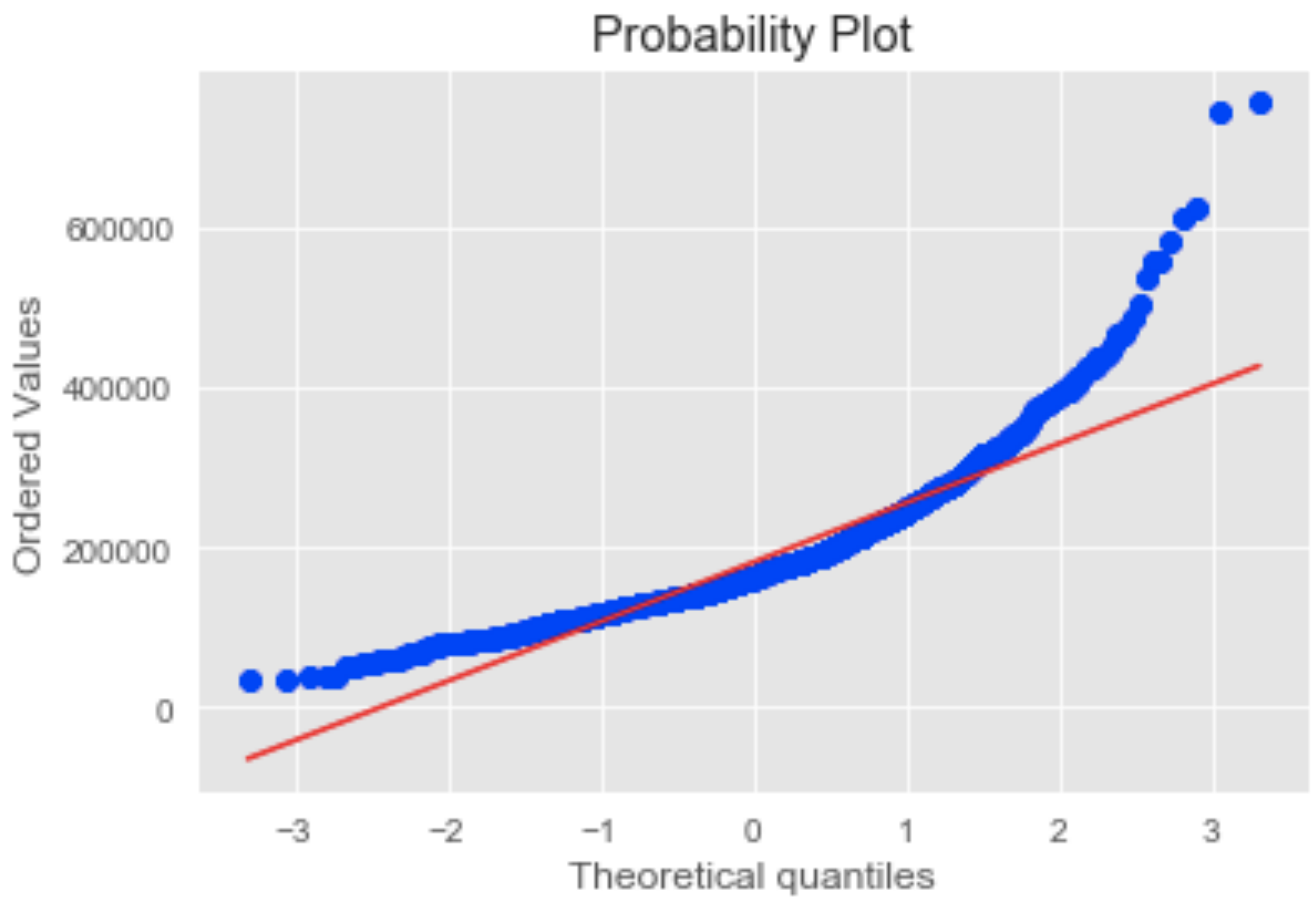
First Glance at the Kaggle Data

Two datasets provided

- Train set
  - 1,460 observations
  - 79 predictor variables (excluding ‘SalePrice, ‘Id’)
    - Numerical variables: 36
    - Categorical variables: 43
- Test set
  - 1,459 observations, 79 predictor variables (excluding ‘Id’)

Exploratory Data Analysis

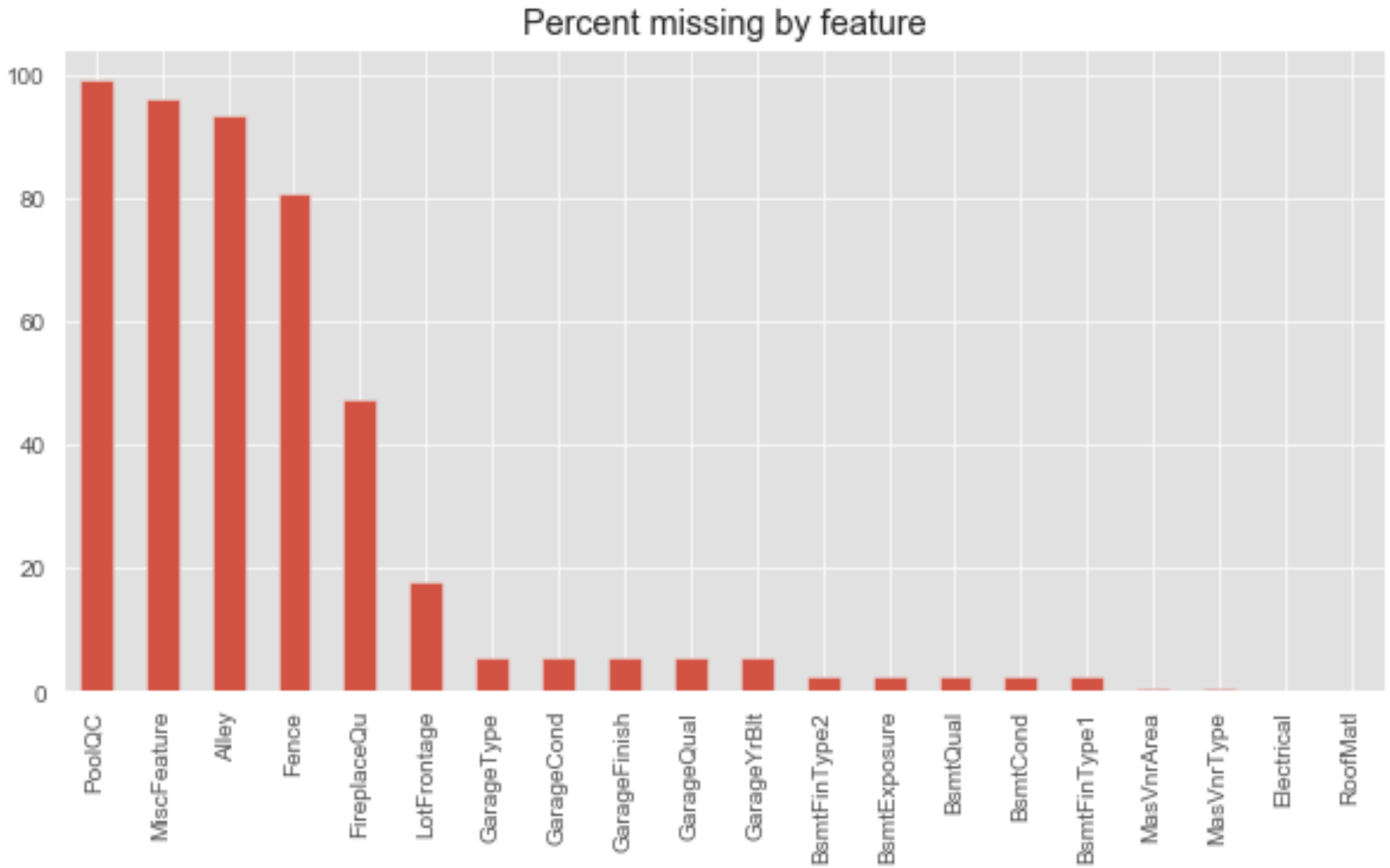
A quick look at our target variable: **SalePrice**



EDA: Examining Variable Distribution

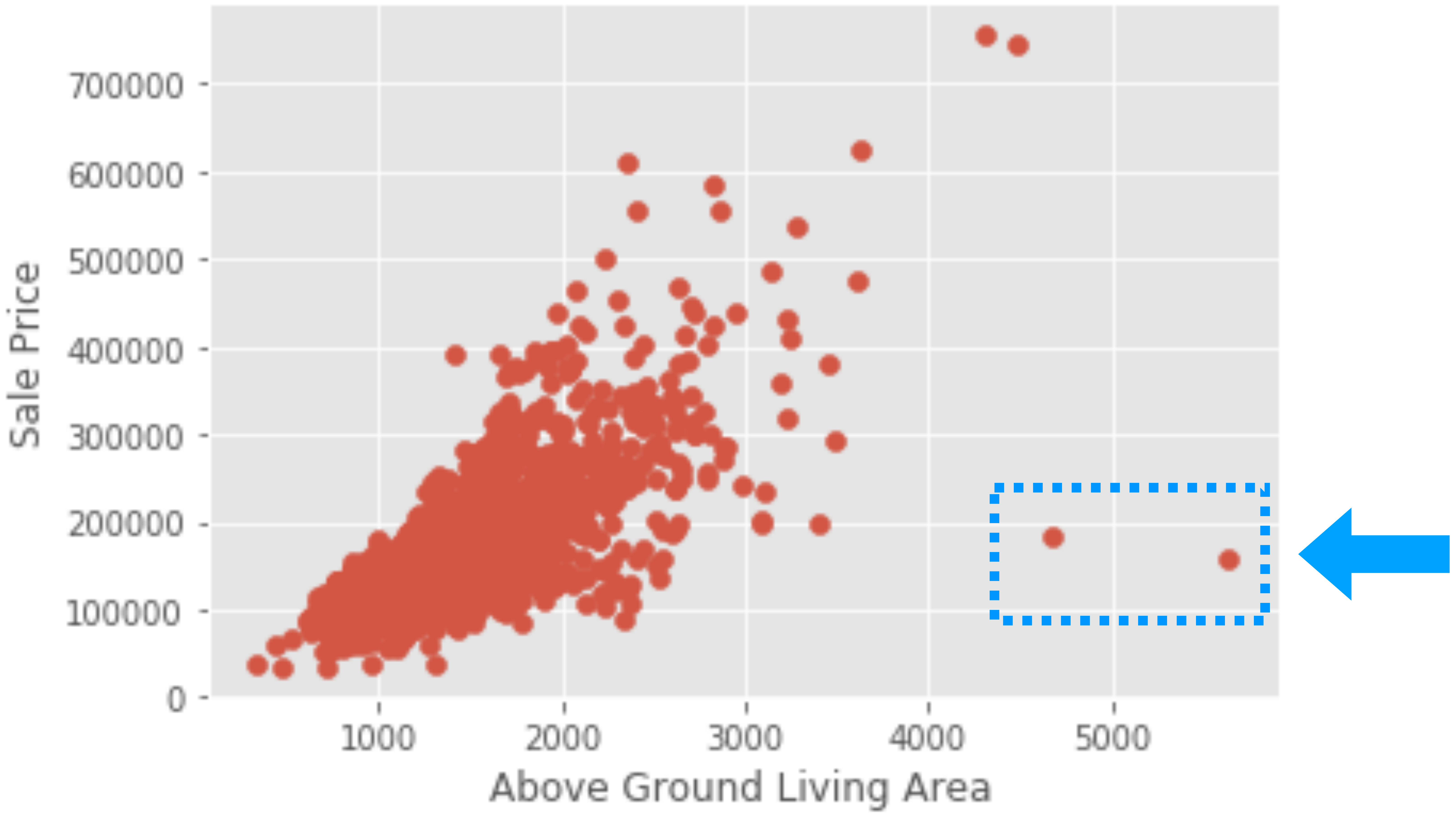
EDA: Looking for Missingness

Check the proportion of missing data by variable



EDA: Identifying Outliers

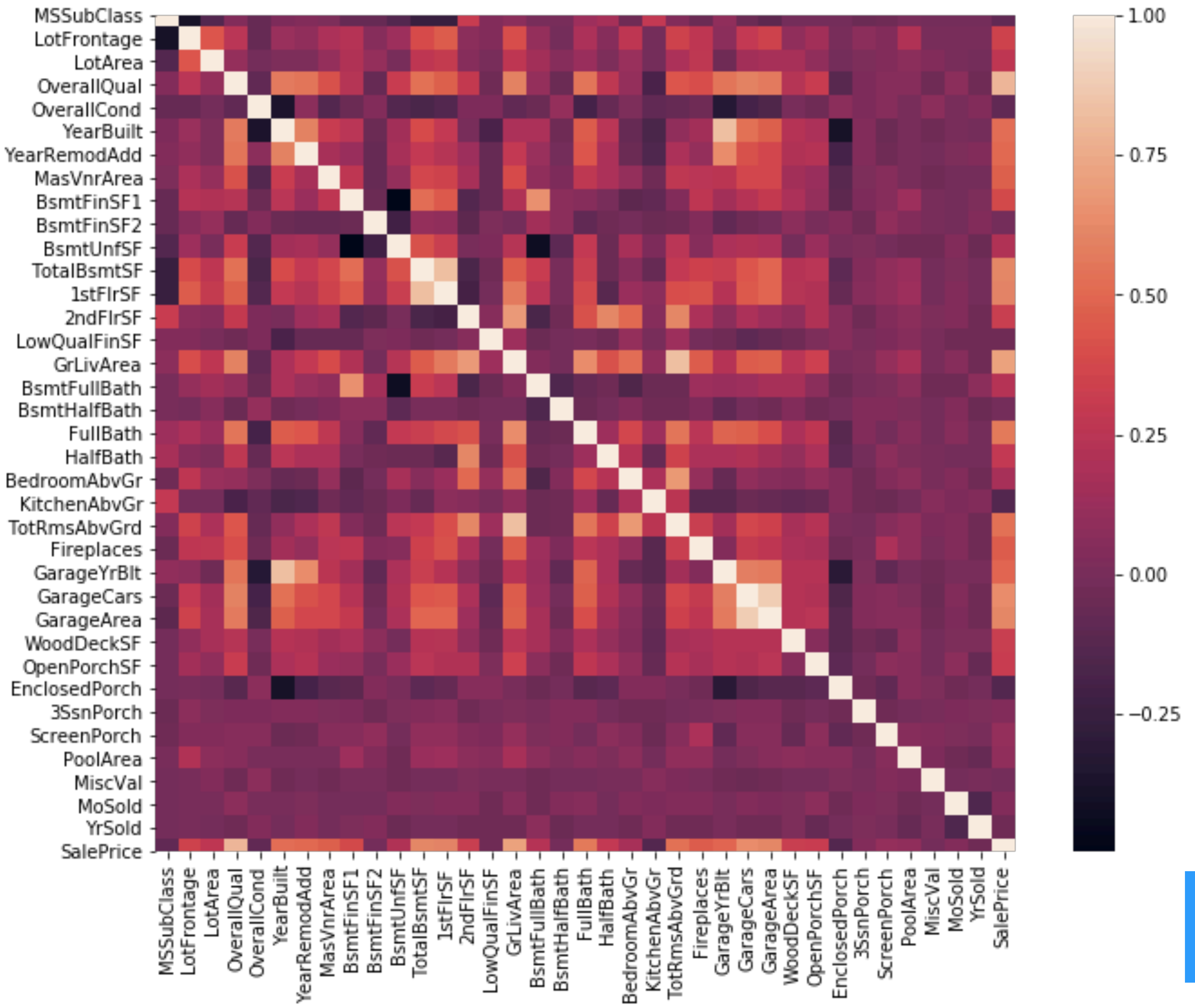
Isolating and removing outliers



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

- Several categorical features represented ordinal rankings, such as quality and condition
- These features' values were converted to numeric, ordinal rankings to reduce the need for dummification

Categorical to Ordinal Conversion

OverallQual, OverallCond, ExterCond, BsmtQual, BsmtCond, HeatingQC, KitchenQual, GarageFinish, GarageQual, GarageCond, BsmtFinType1

Feature Engineering (continued)

Generated New Features

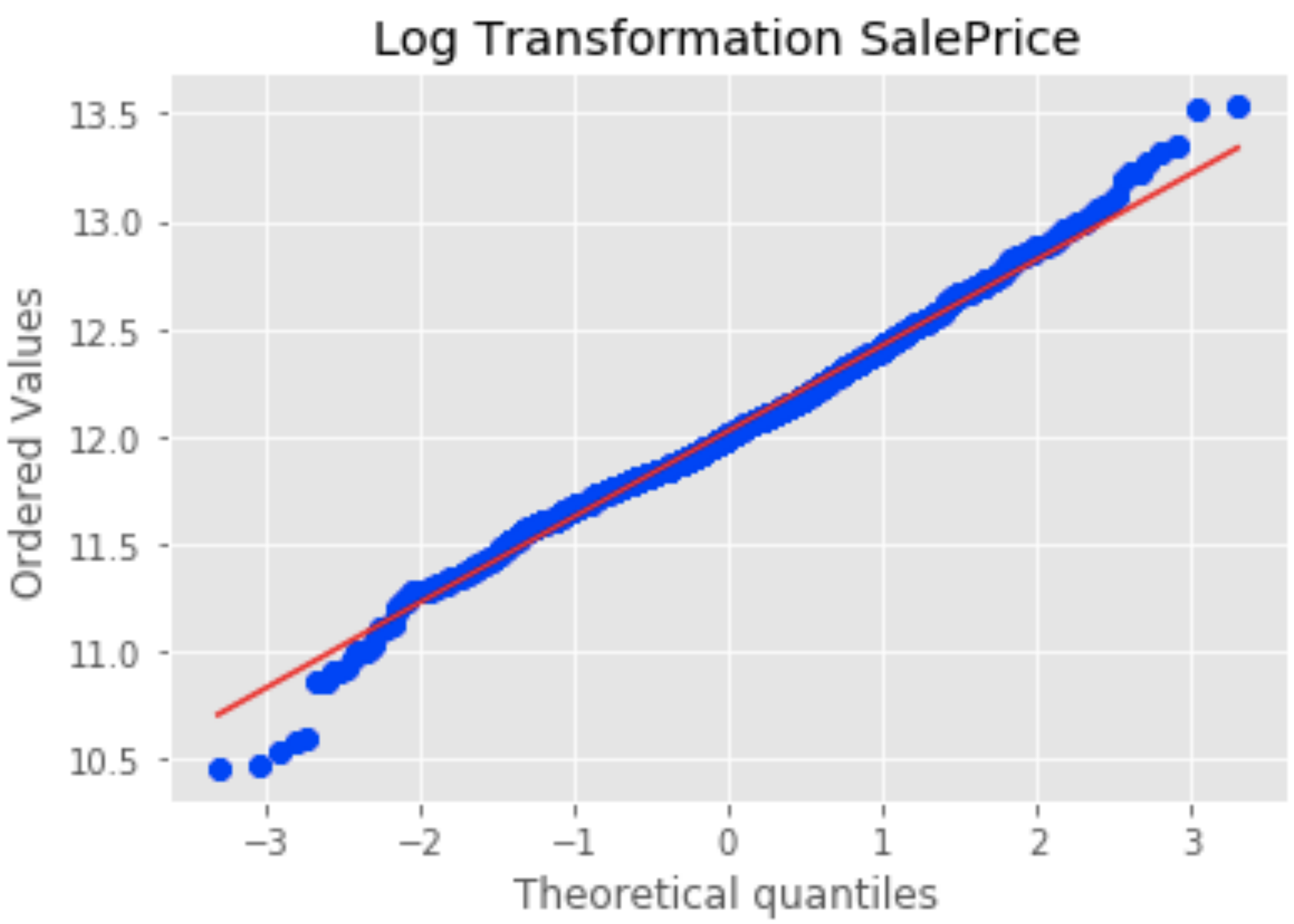
- Total SF = 1stFlrSF + 2ndFlrSF + TotalBsmtSF — Numeric
- TotalFullBath = BsmtFullBath + FullBath — Numeric
- TotalHalfBath = BsmtHalfBath + HalfBath — Numeric
- IsPool — Categorical
- IsGarage — Categorical

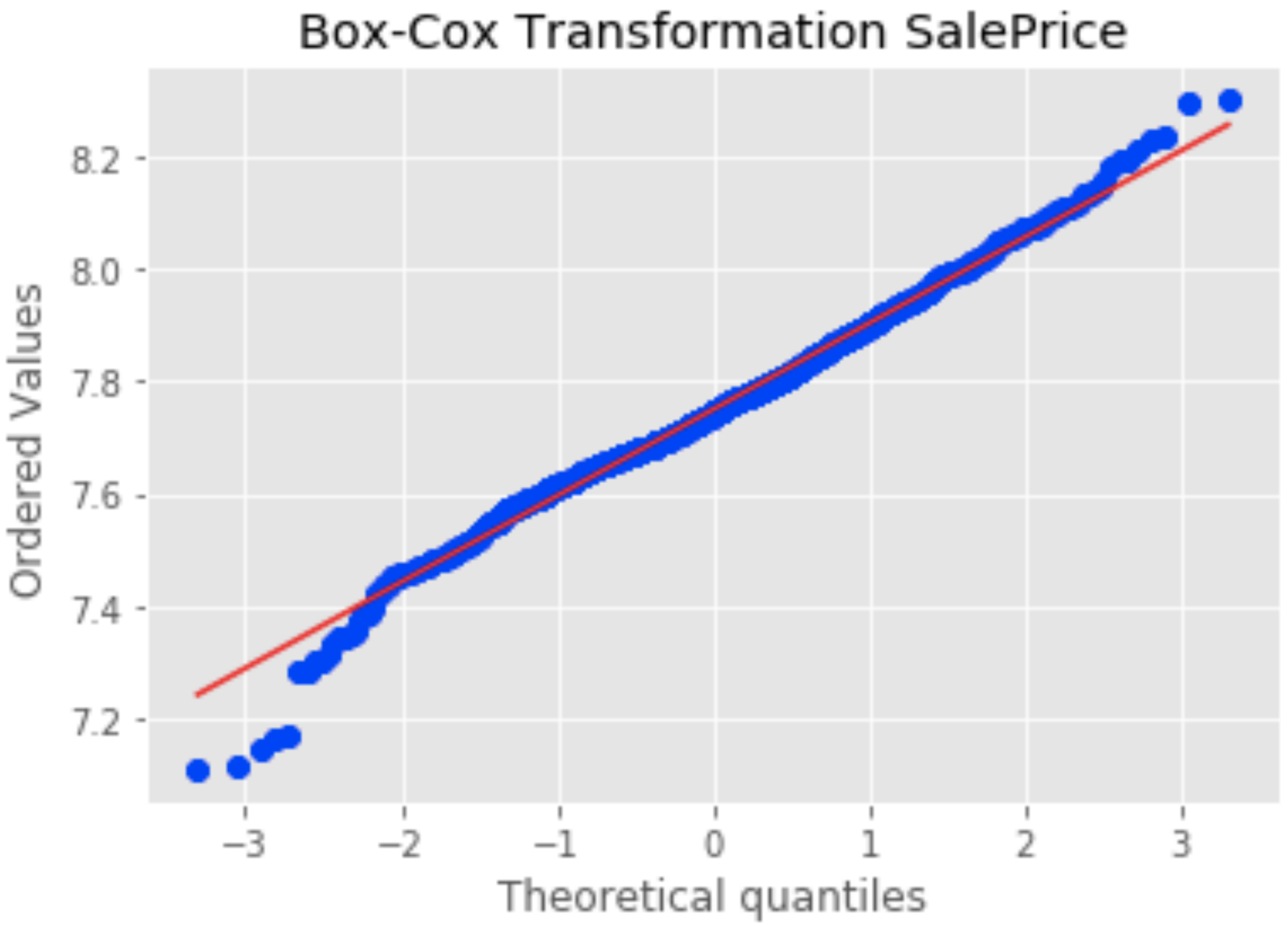
Dummified Categorical Feature Variables

Removed Redundant Variables

SalePrice Transformation: Log vs. Box-Cox

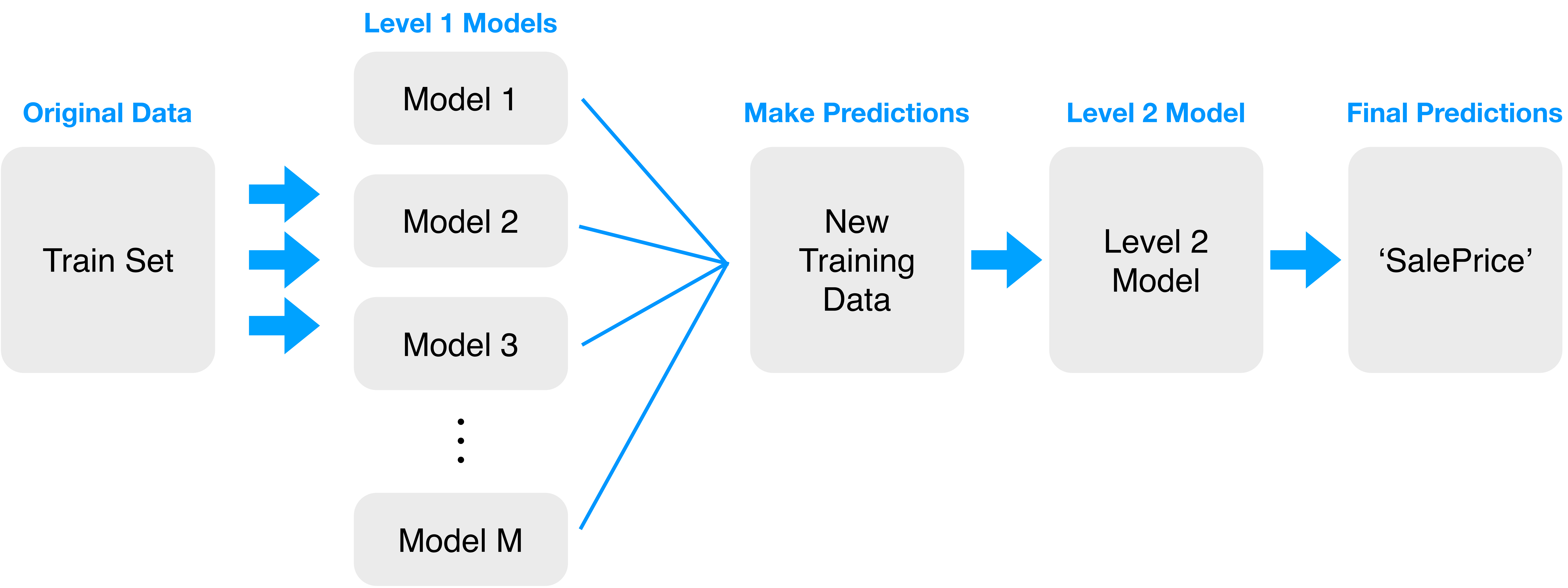
We decided to utilize a Box-Cox transformation over a Log transformation as it provided slightly better predictive power





Models

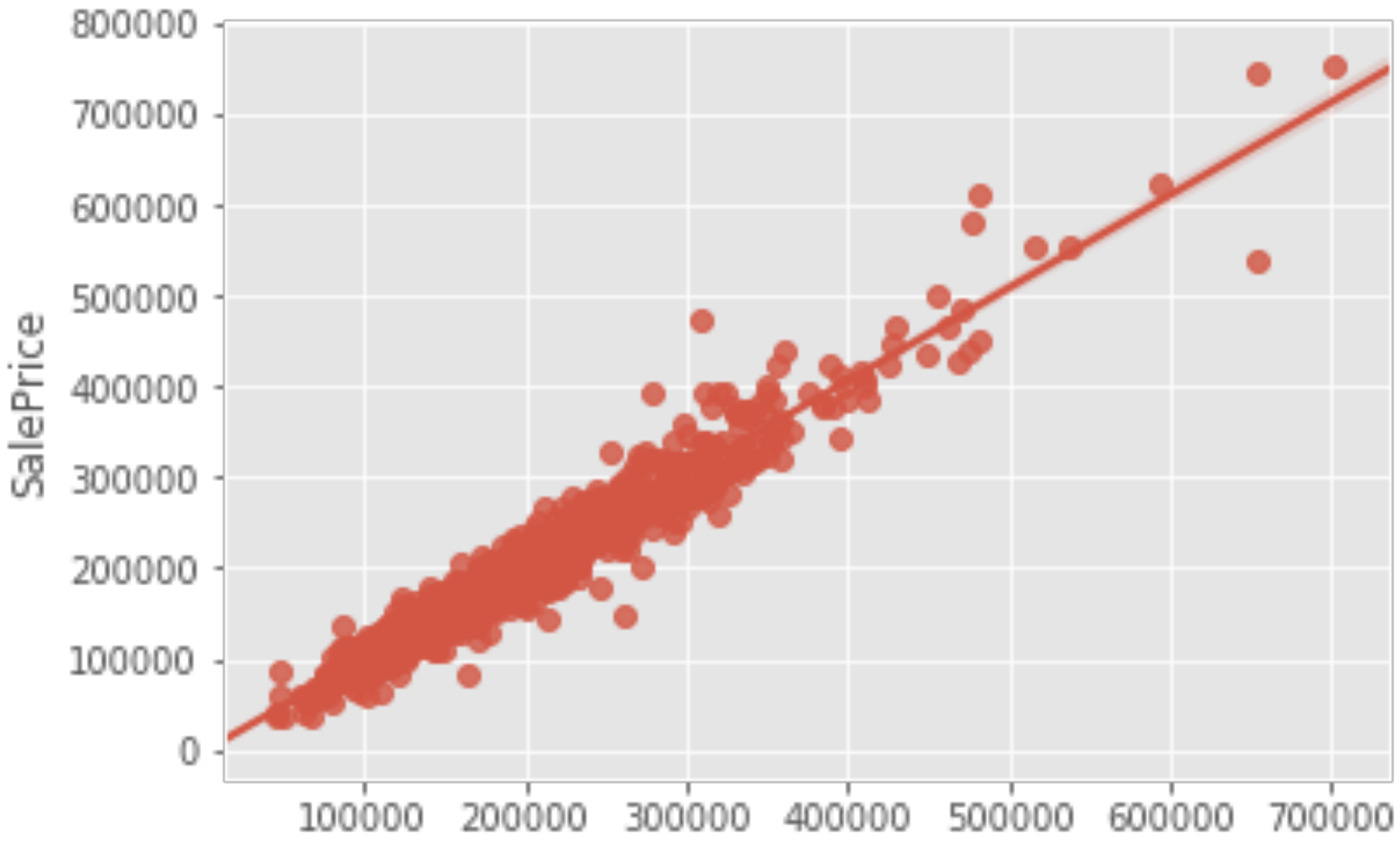
Modeling Methodology

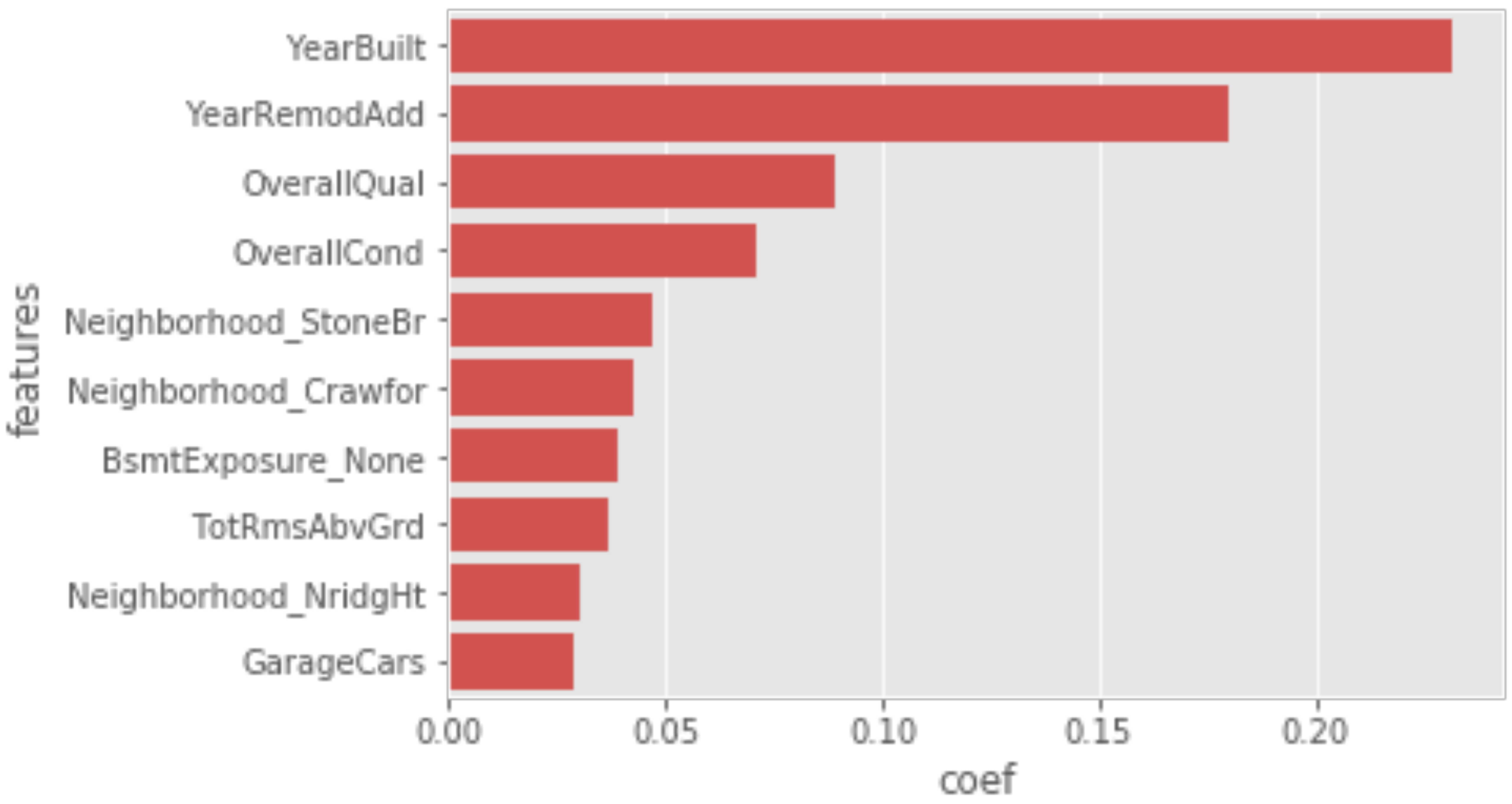


Models Tested

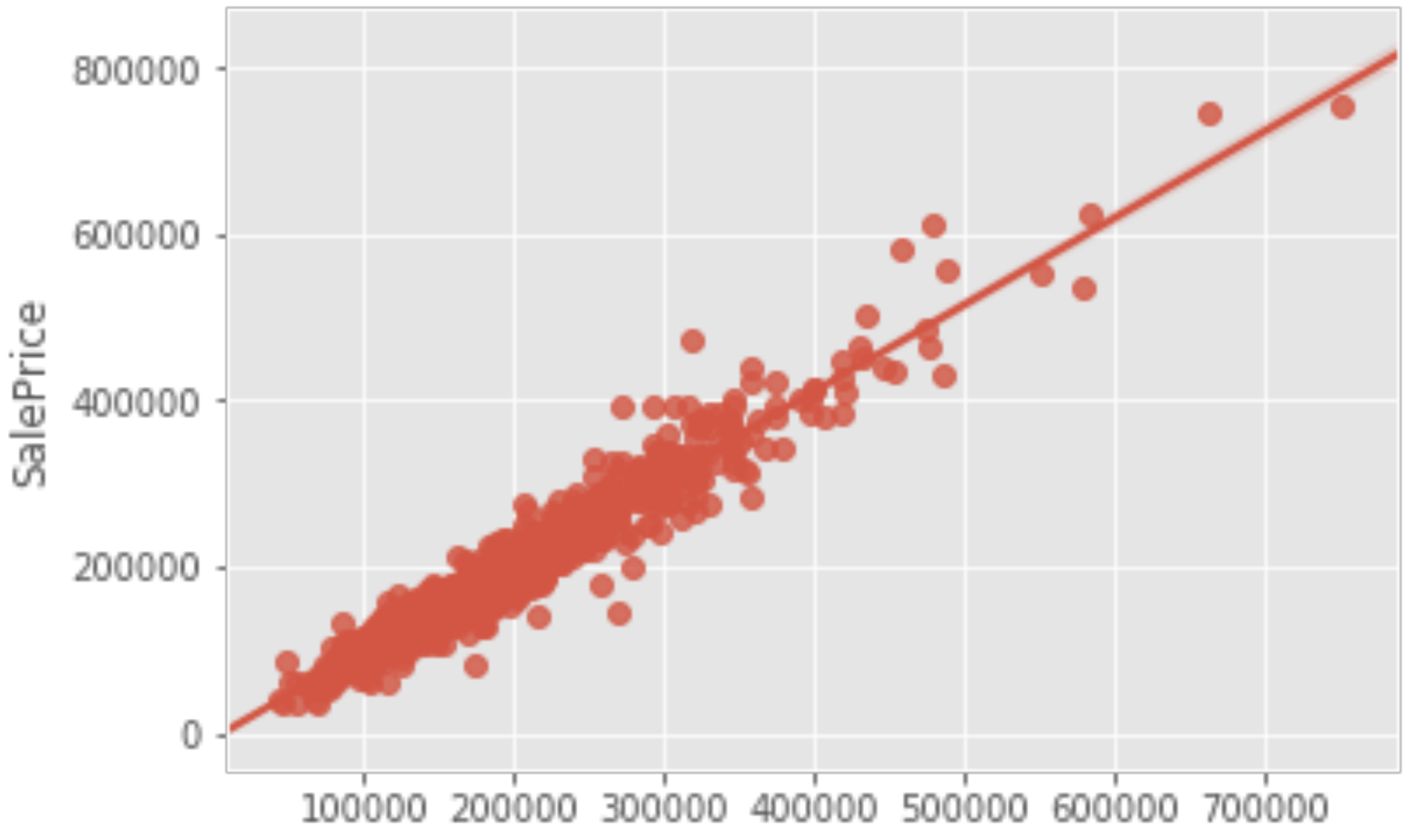
Model Level	Model	Cross Validation Score (RMSE)	Kaggle Score
Level I	Ridge	0.11513	0.12393
	Lasso	0.11355	0.12193
	Elastic Net	0.11355	0.12189
	Catboost	0.11291	0.12200
	Gradient Boost	0.11211	0.12360
	LightGBM	0.11573	0.12275
Level II	Simple Average	0.10921	0.11859
	Lasso	0.10895	0.11855

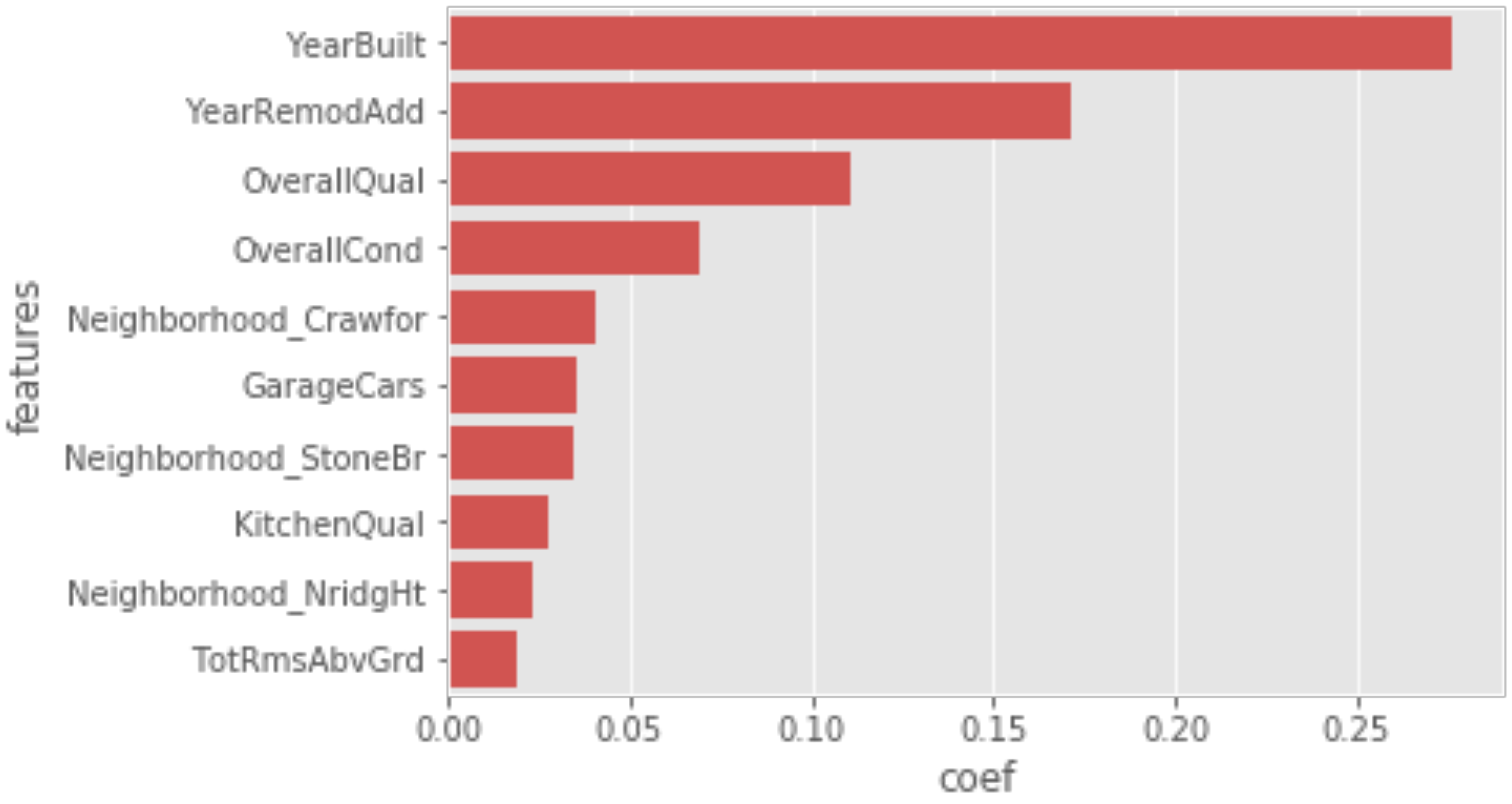
Model 1: Ridge



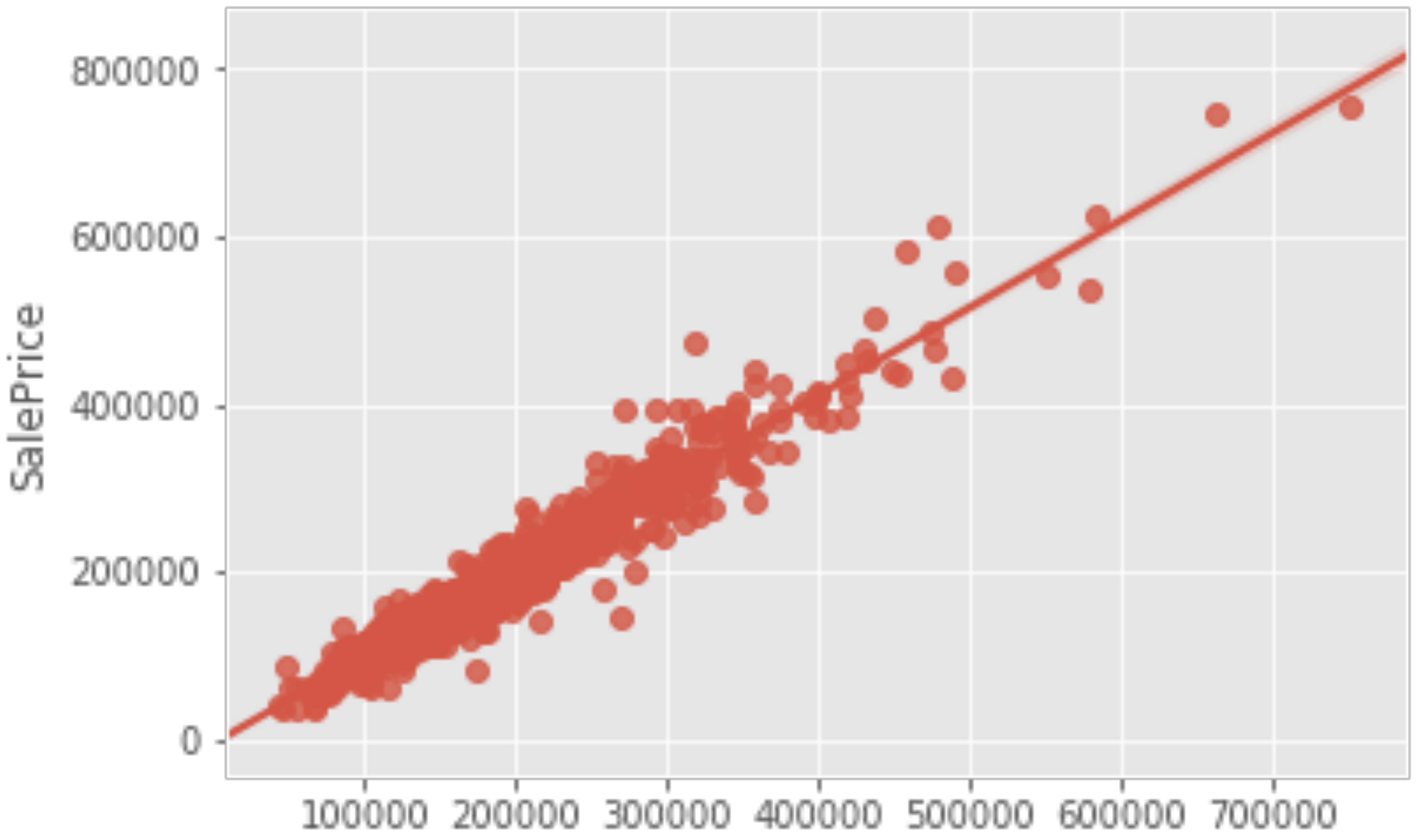


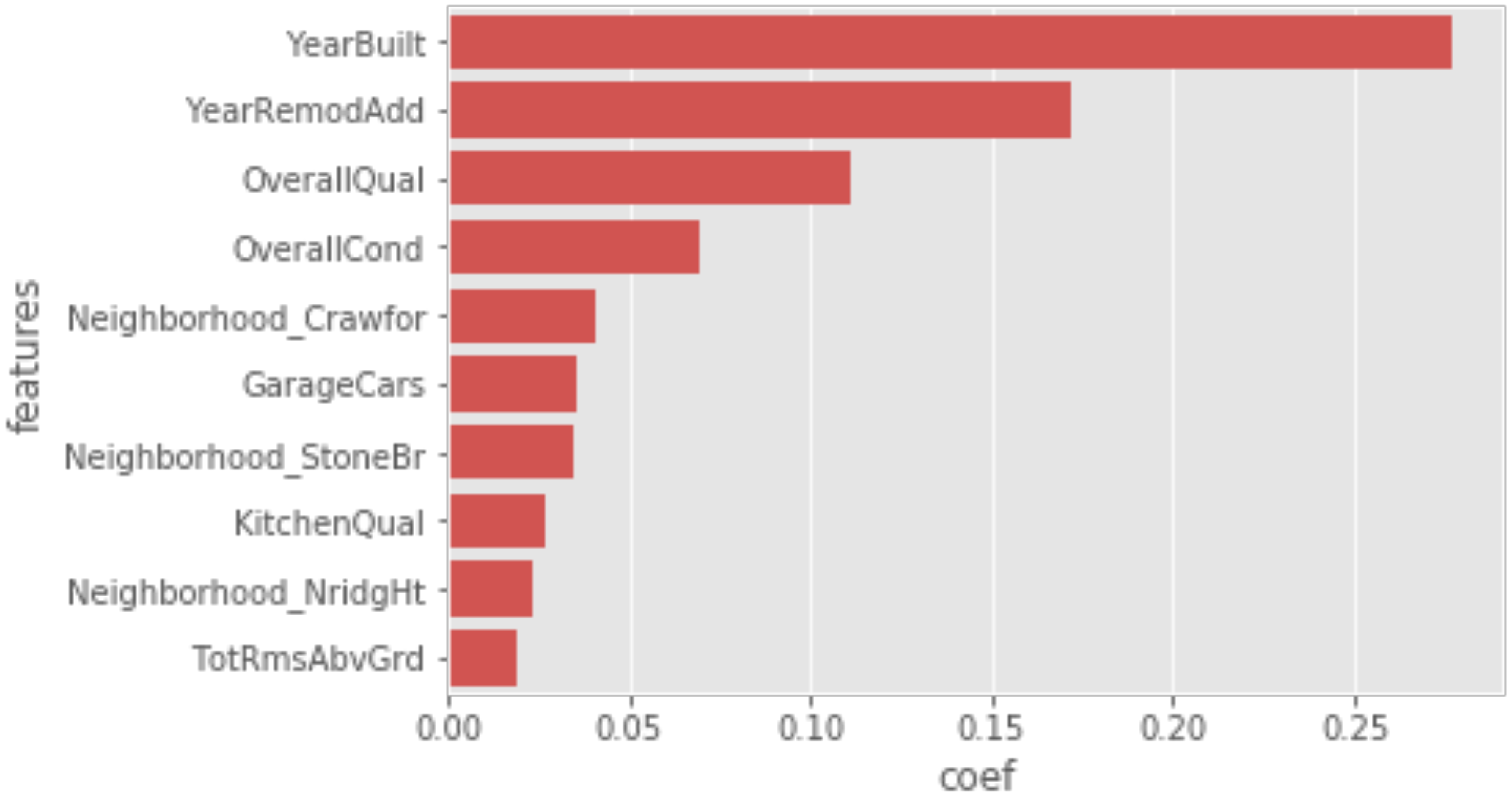
Model 2: Lasso



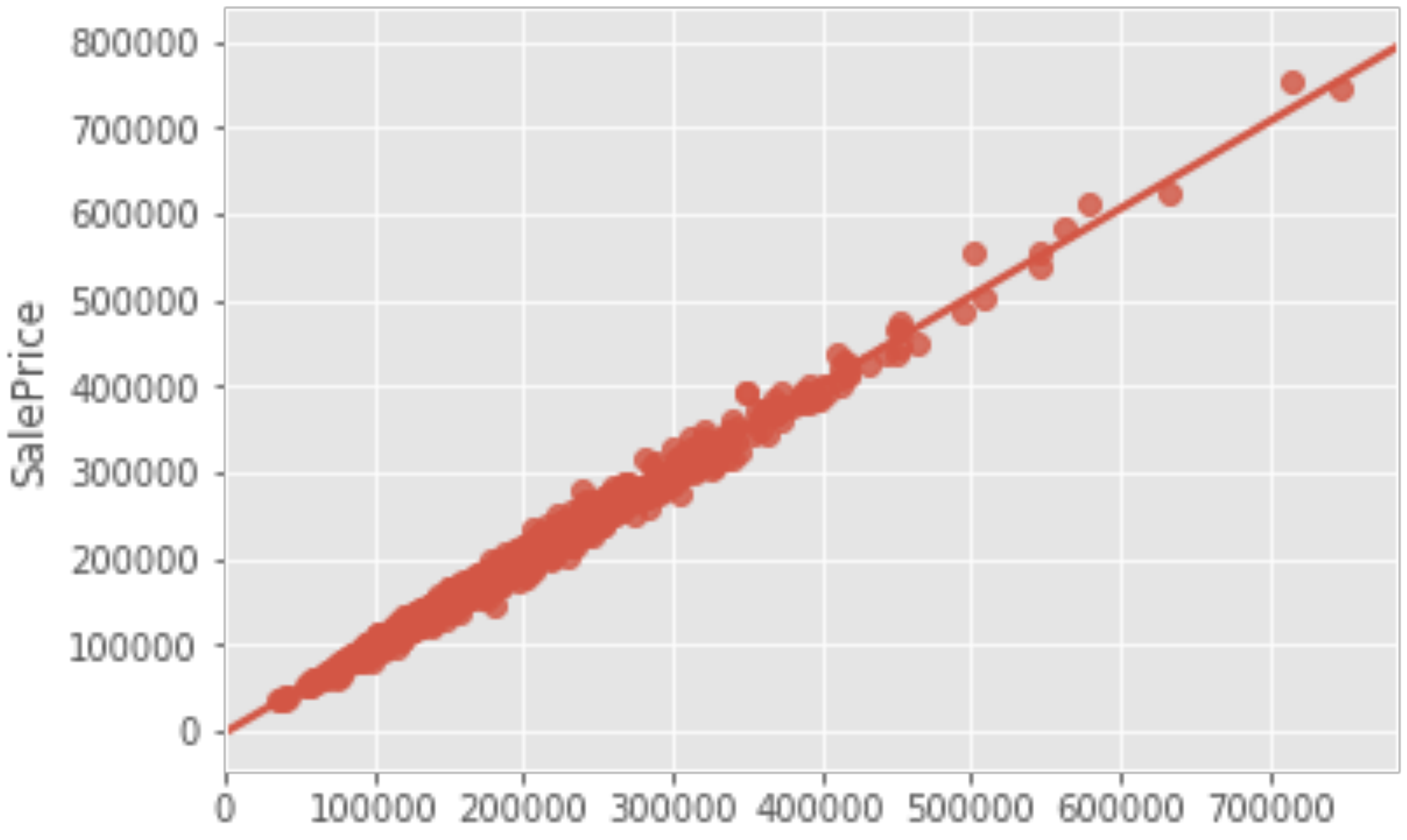


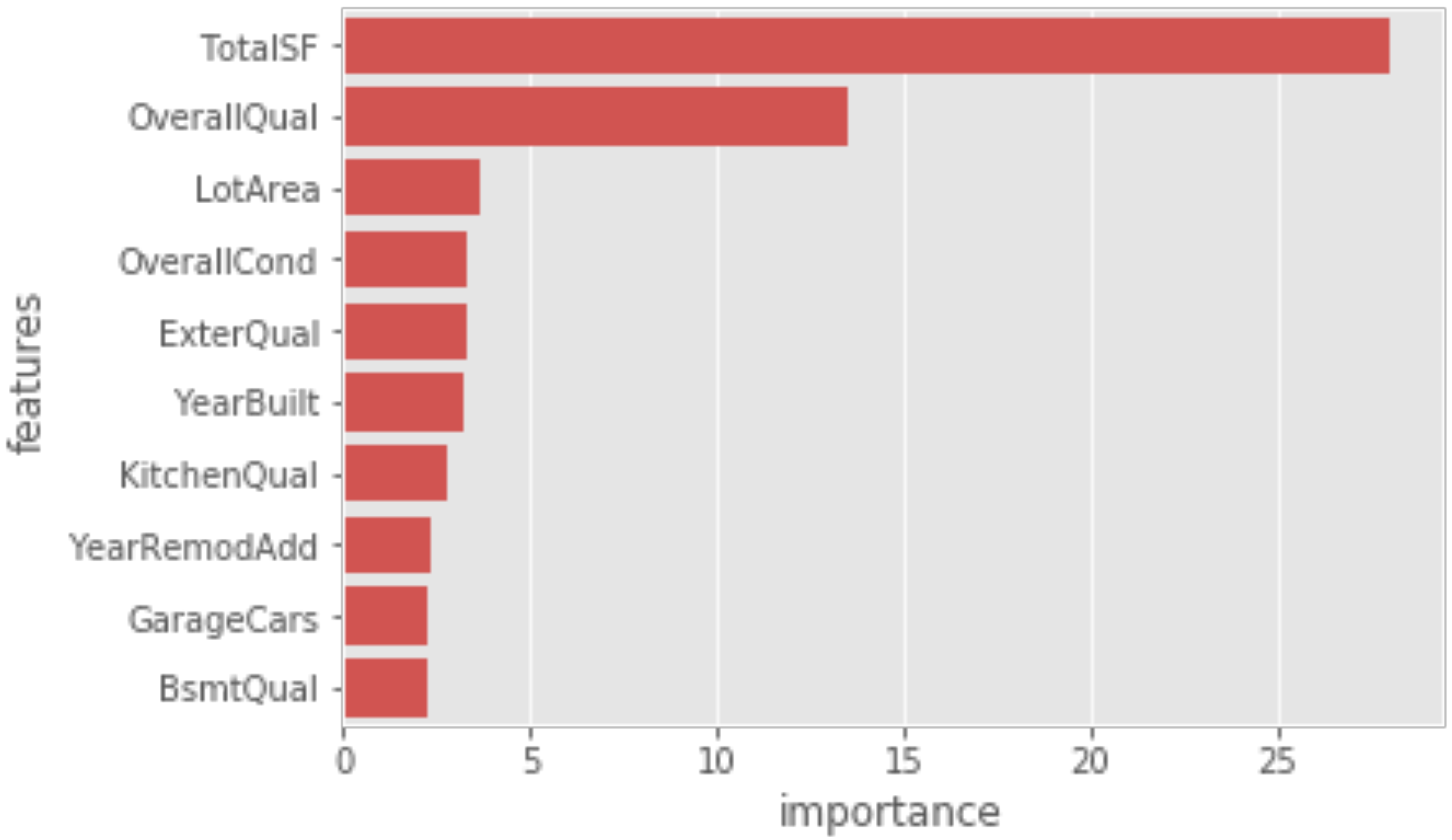
Model 3: Elastic Net



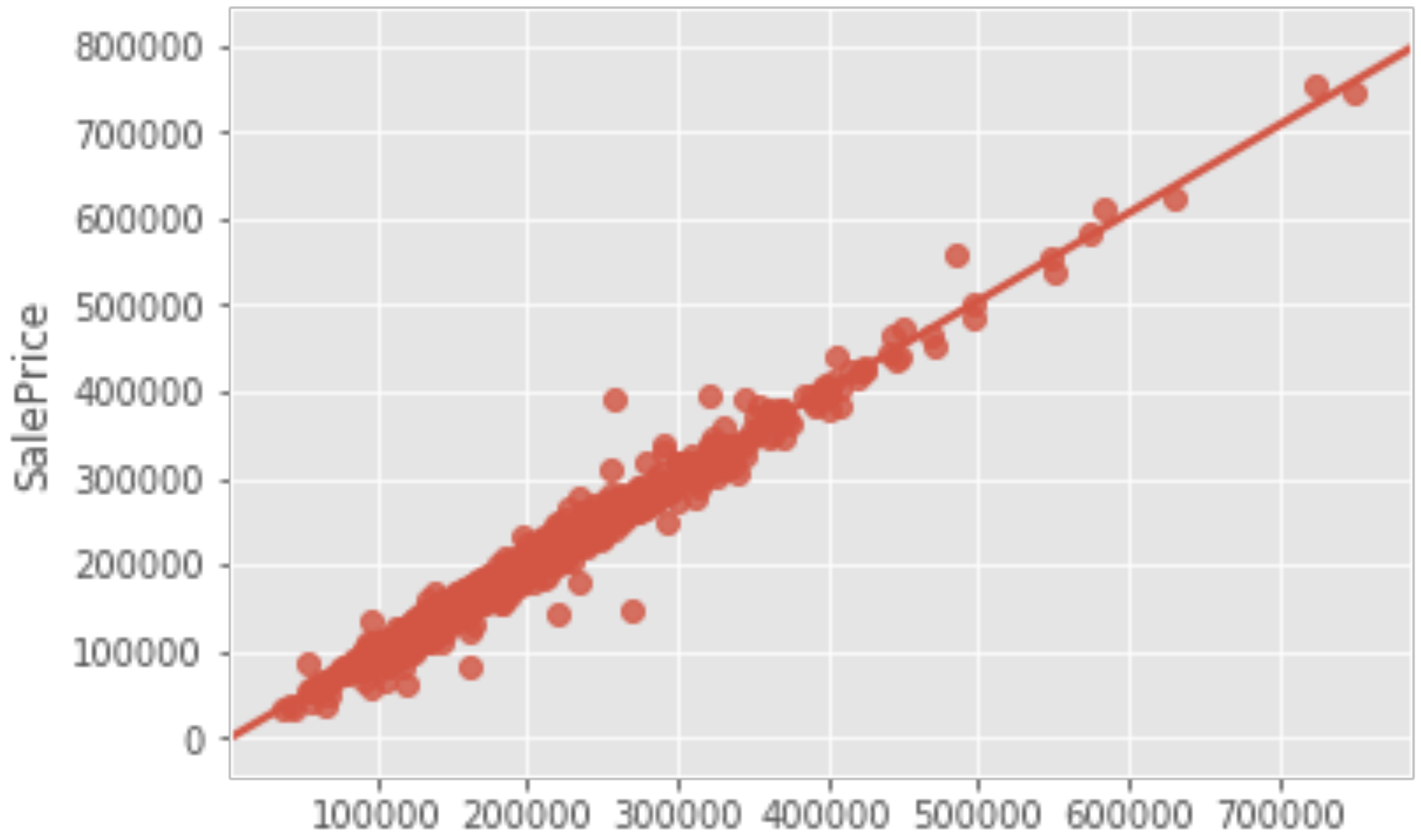


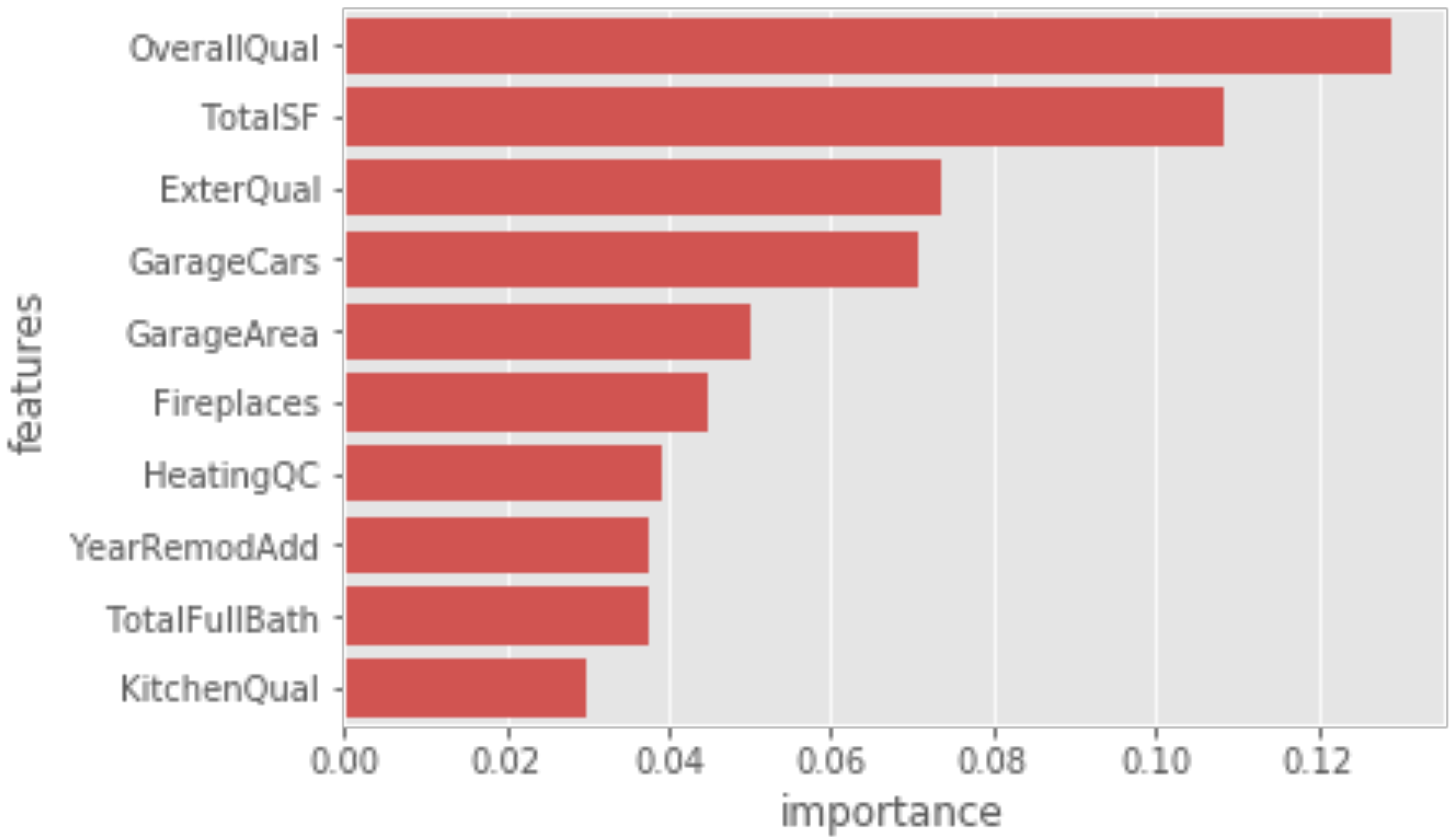
Model 4: Catboost



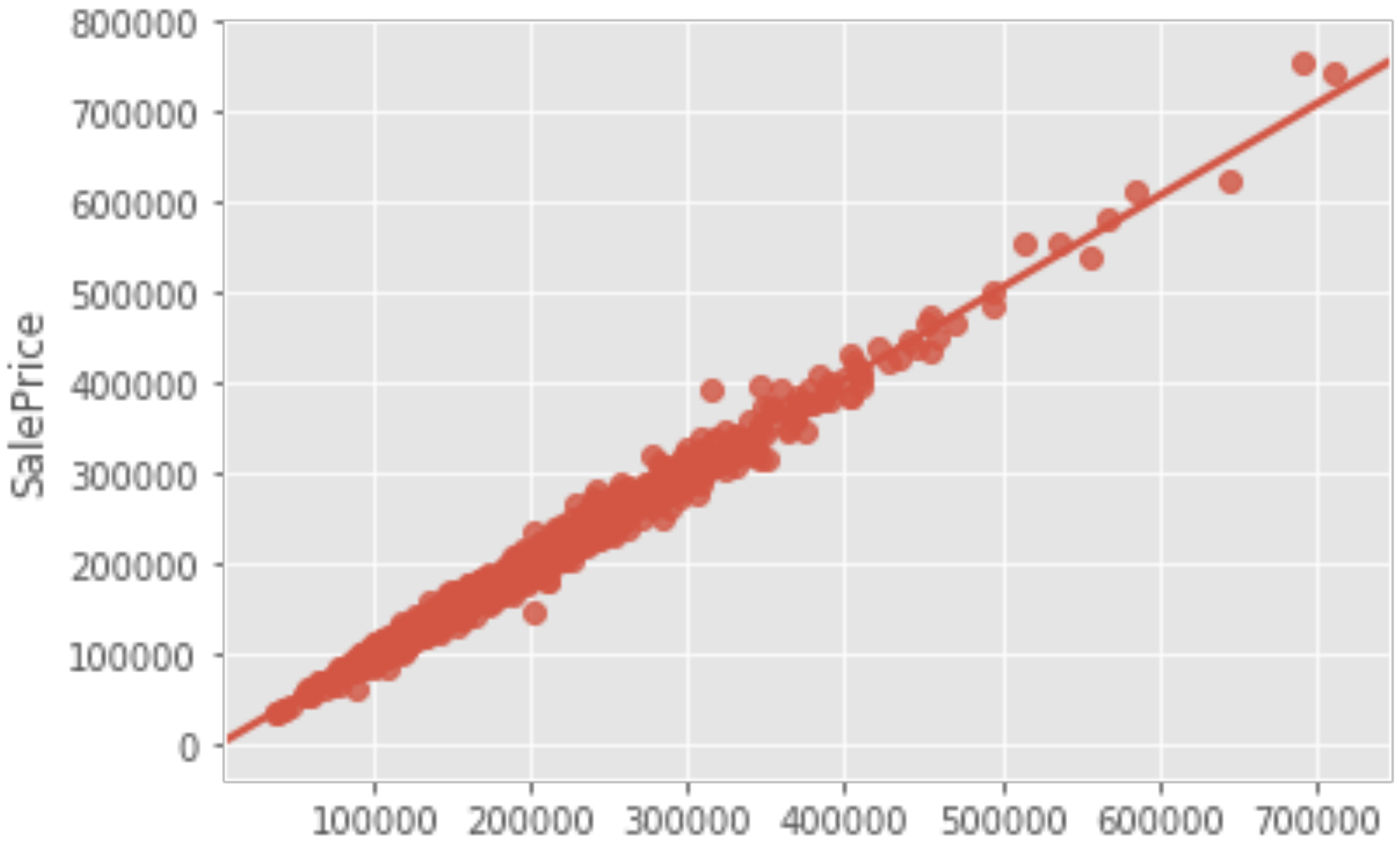


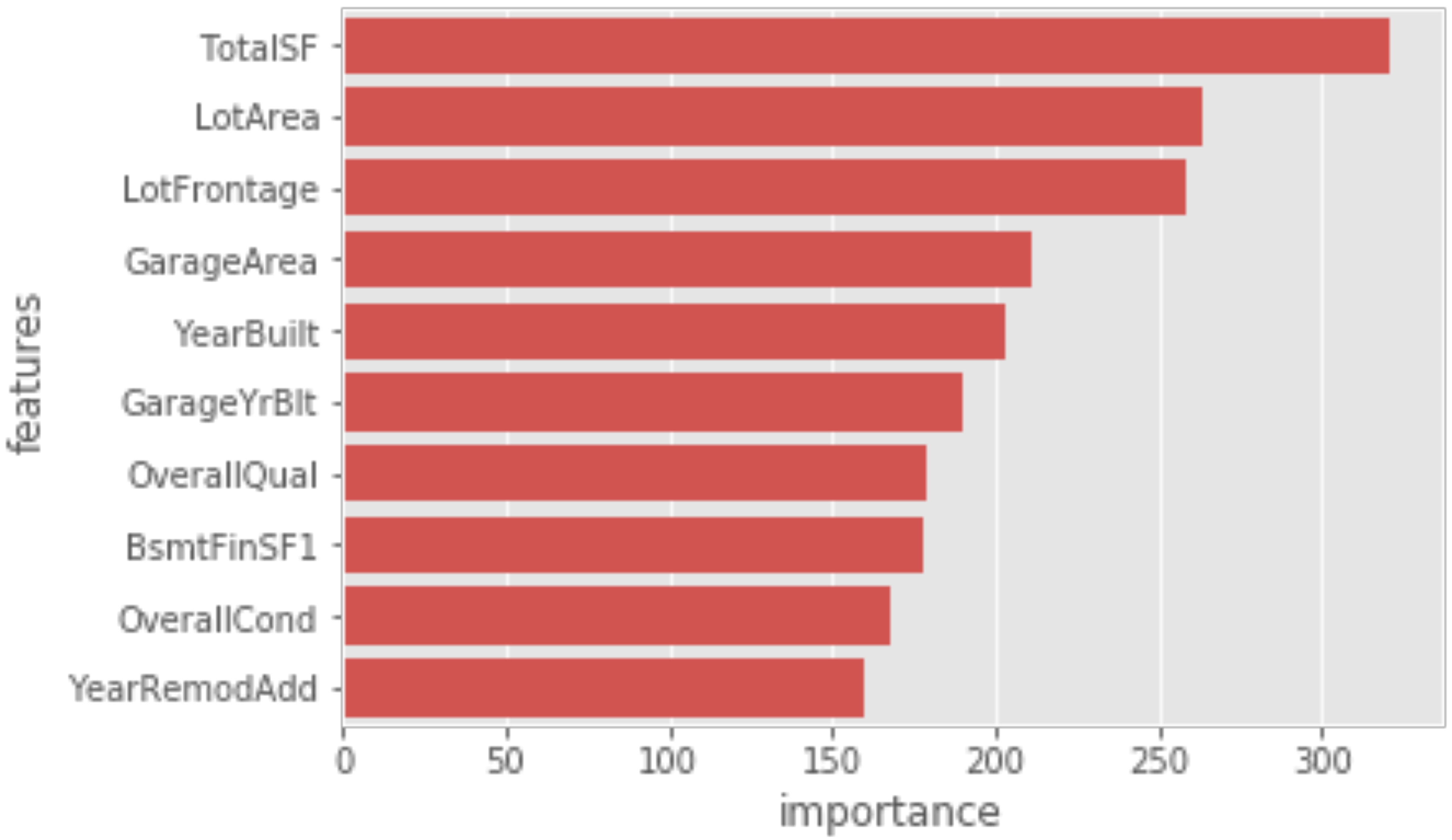
Model 5: Gradient Boost





Model 6: Light GBM





Conclusion

Recommendation:

- Our best predictions of home portfolio value came from our Stacked model

Model	Cross Validation Score (RMSE)	Kaggle Score
Stacked	0.10895	0.11855

- 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!