# House Prices: Advanced Regression Techniques

Weijie Deng, Han Wang, Chima Okwuoha, and Shiva Panicker

# Workflow

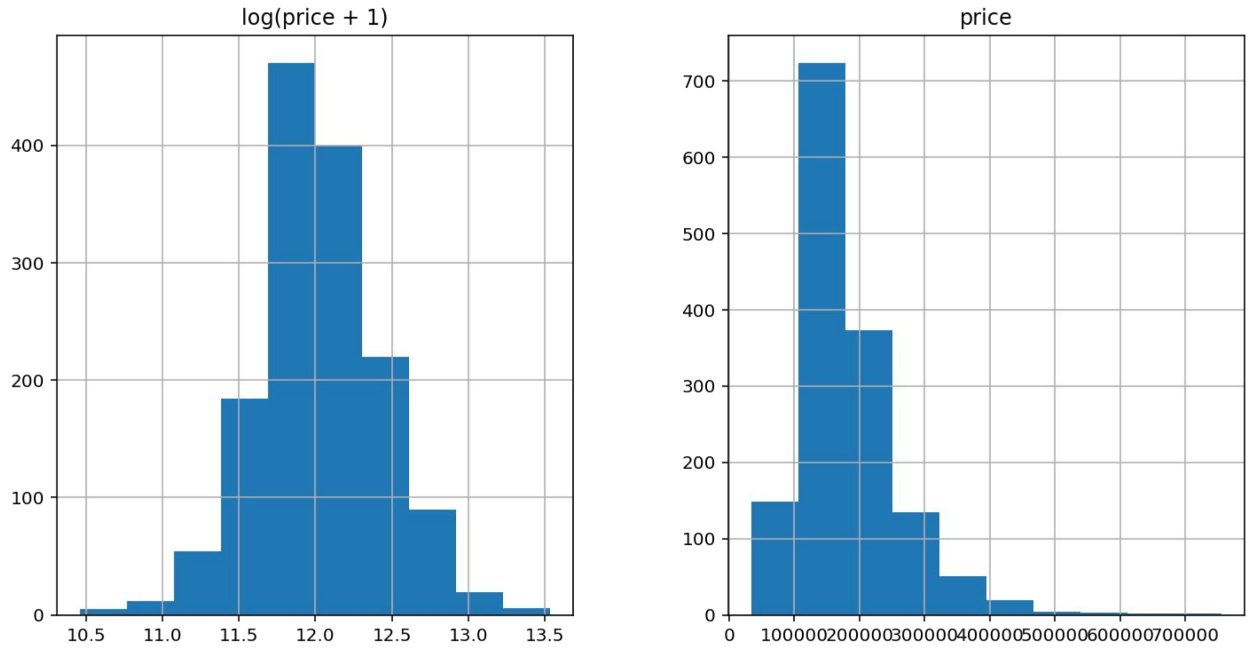
1. Data cleaning
2. Models
3. Stacking

# 1. Data Cleaning

# Features

- lot/land variables
- location variables
- age variables
- basement variables
- roof variables
- garage variables
- kitchen variables
- room/bathroom variables
- utilities variables
- appearance variables
- external features (pools, porches, etc.) variables

# Data Cleaning - Scaling the Sale Price



# Data Cleaning - Skewing

For numeric features, I checked whether they need to be skewed.

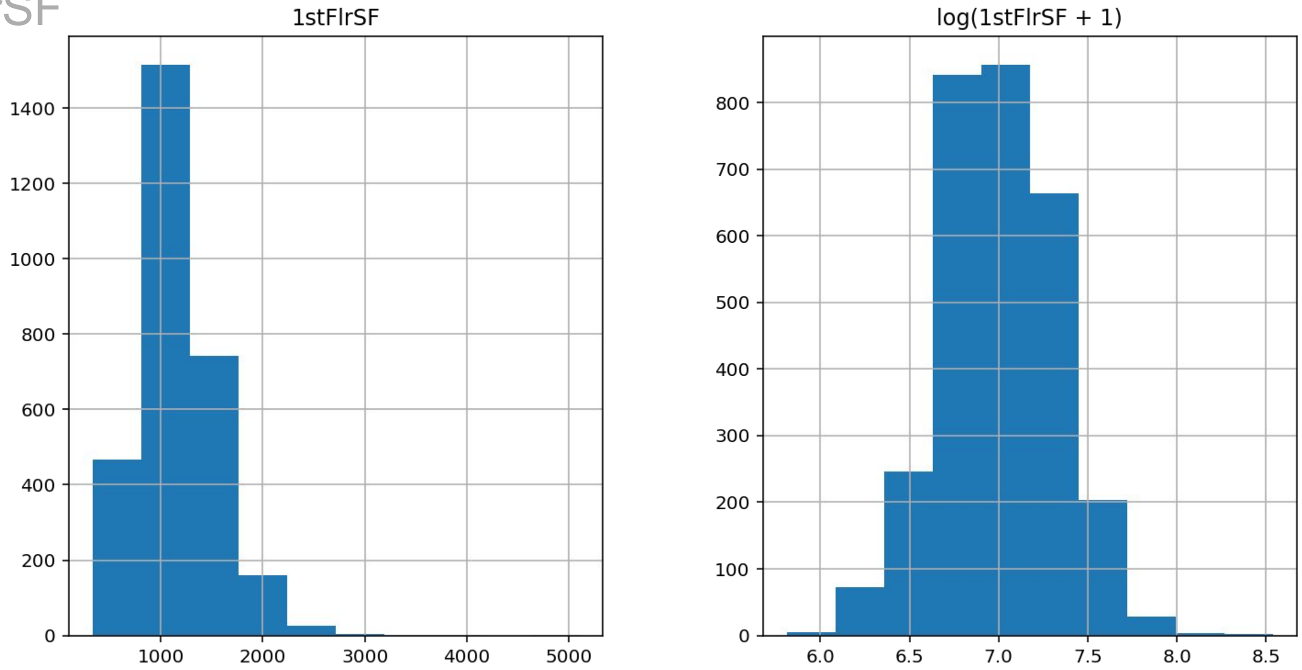
For those with degree of skewness larger than 0.75, we need to skew it to make the features obey normal distribution.

The following features need to be skew.

(1stFlrSF', '2ndFlrSF', '3SsnPorch', 'BsmtFinSF1', 'BsmtFinSF2',
'BsmtHalfBath', 'BsmtUnfSF', 'EnclosedPorch', 'GrLivArea',
'KitchenAbvGr', 'LotArea', 'LotFrontage', 'LowQualFinSF', 'MasVnrArea',
'MiscVal', 'OpenPorchSF', 'PoolArea', 'ScreenPorch', 'TotRmsAbvGrd',
'TotalBsmtSF', 'WoodDeckSF')

# Data Cleaning - Skewing

Example : 1stFlrSF



# Data Cleaning - Ordinal features

```
Dict = {"No": 0, "Po": 1, "Fa": 2, "TA": 3, "Gd": 4, "Ex": 5,
             "Mn": 2, "Av": 3,
             "Unf": 1, "LwQ": 2, "Rec": 3, "BLQ": 4, "ALQ": 5, "GLQ": 6,
             "Sal": 1, "Sev": 2, "Maj2": 3, "Maj1": 4, "Mod": 5, "Min2": 6, "Min1": 7, "Typ": 8,
             "RFn": 2, "Fin": 3,
             "MnWw": 1, "GdWo": 2, "MnPrv": 3, "GdPrv": 4,
             "N": 0, "Y": 1, np.nan:0
            }

for col in ['ExterQual', 'ExterCond', 'BsmtQual', 'BsmtCond', 'HeatingQC',
           'KitchenQual', 'FireplaceQu', 'GarageQual', 'GarageCond',
           'BsmtFinType1', 'BsmtFinType2', 'Functional', 'GarageFinish', 'Fence',
           'CentralAir']:

        all_data[col] = all_data[col].map(Dict).astype(int)
```

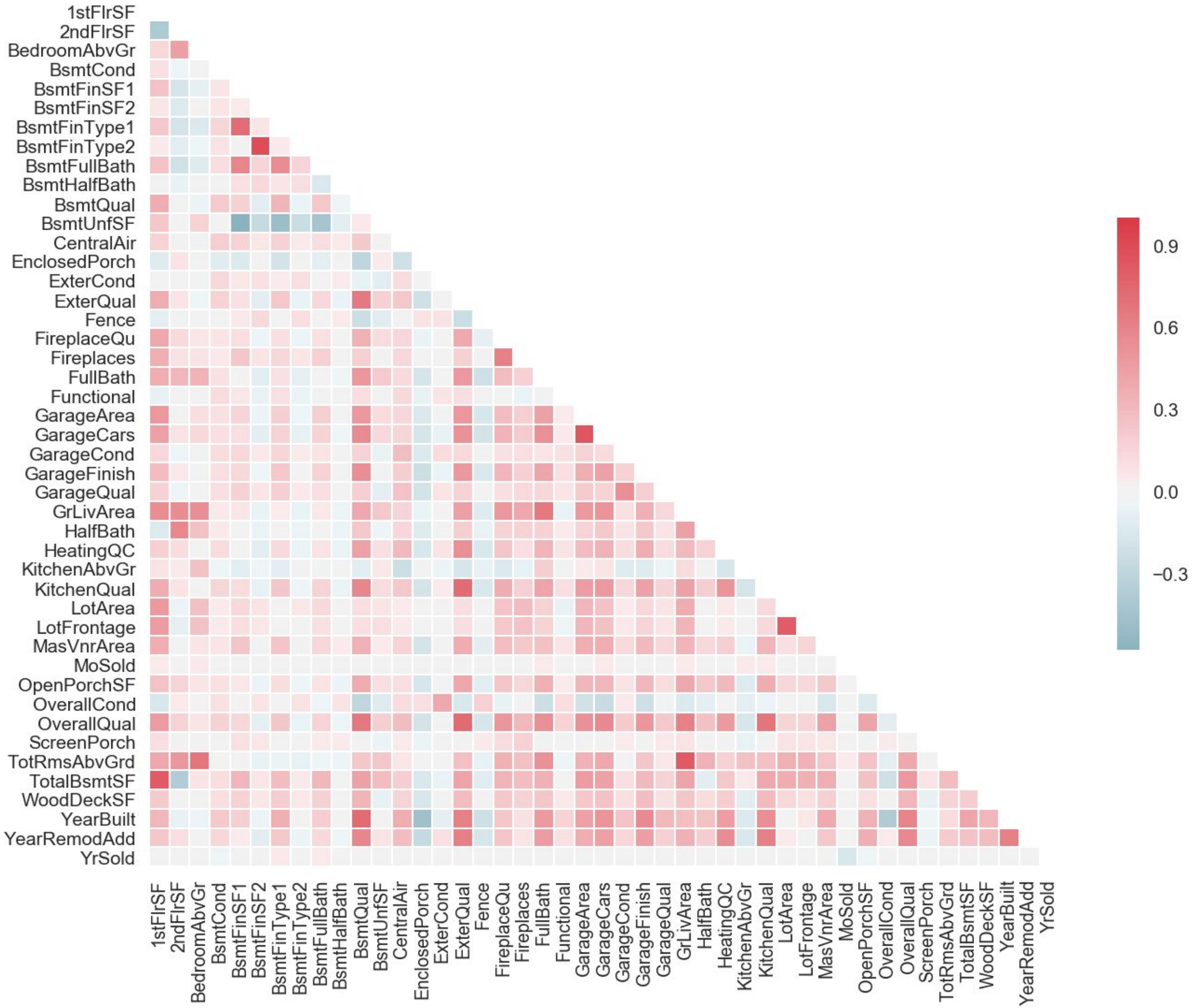
# Data Cleaning - One Hot Encoding

For the Categorical Features, we used one hot encoding to dummify the features.

| MiscFeature |
| --- |
| 'None' |
| 'Shed' |
| 'Gar2' |

| MiscFeatureNone | MiscFeatureShed |
| --- | --- |
| 1 | 0 |
| 0 | 1 |

# Multicollinearity



# 2. Models

# Gradient Boosting - Parameters

'learning_rate': 0.04,

'max_depth': 4,

'max_features': 'sqrt',

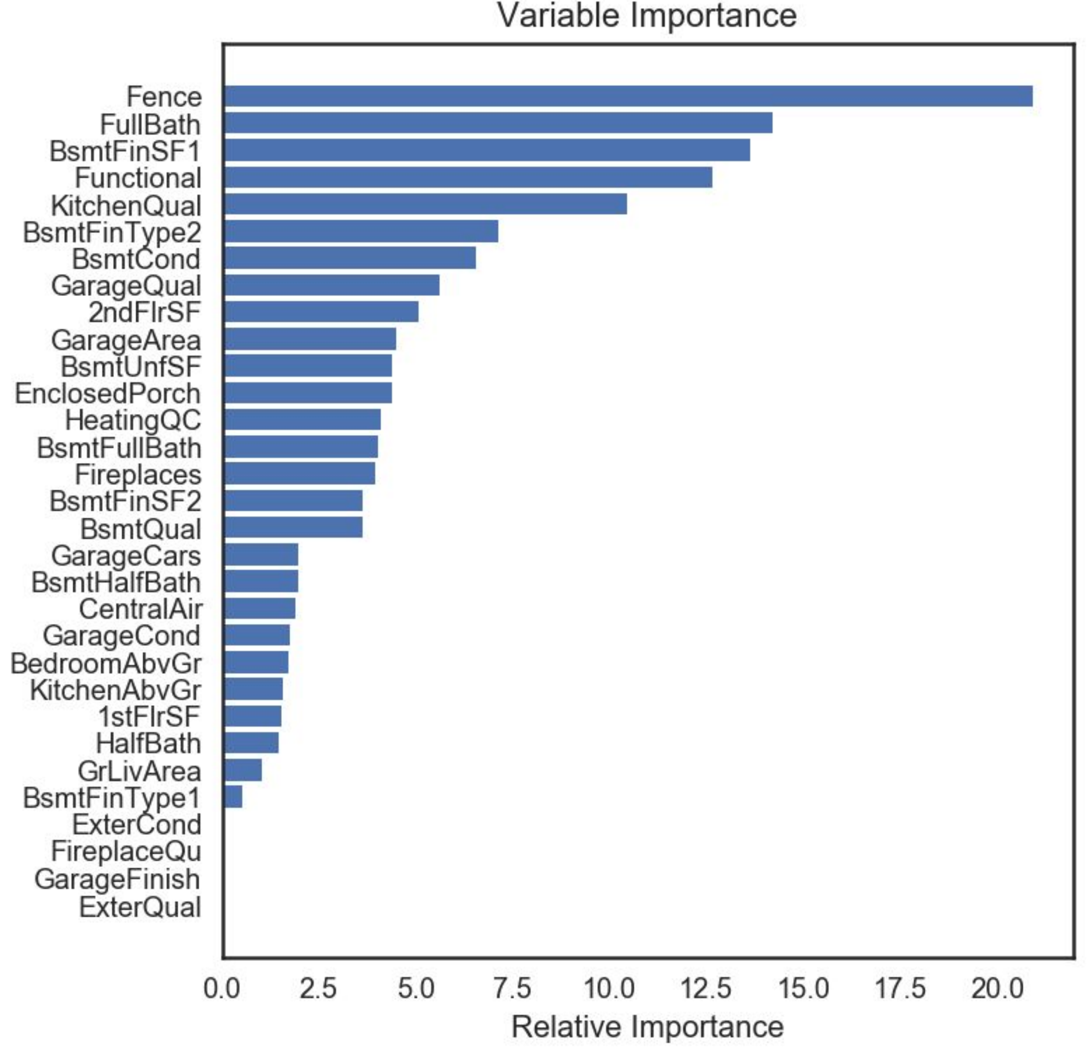
'min_samples_leaf': 2,

'min_samples_split': 10,

'n_estimators': 500,

# Gradient Boosting

-Variable Importance

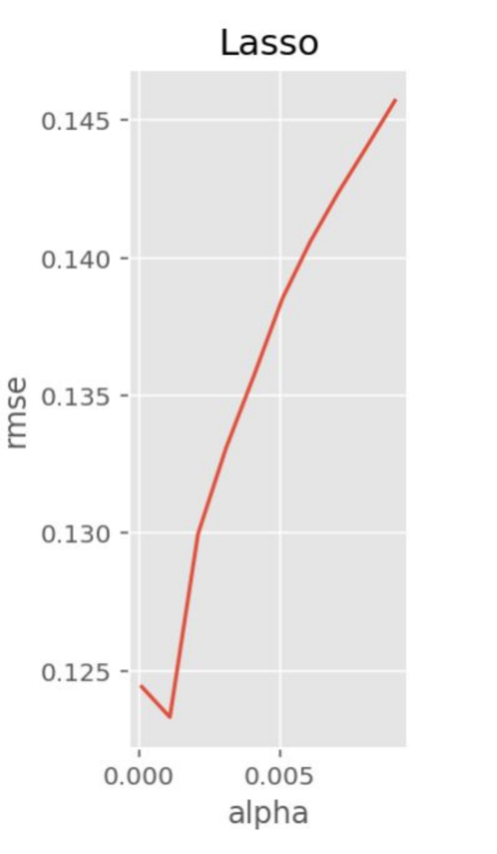


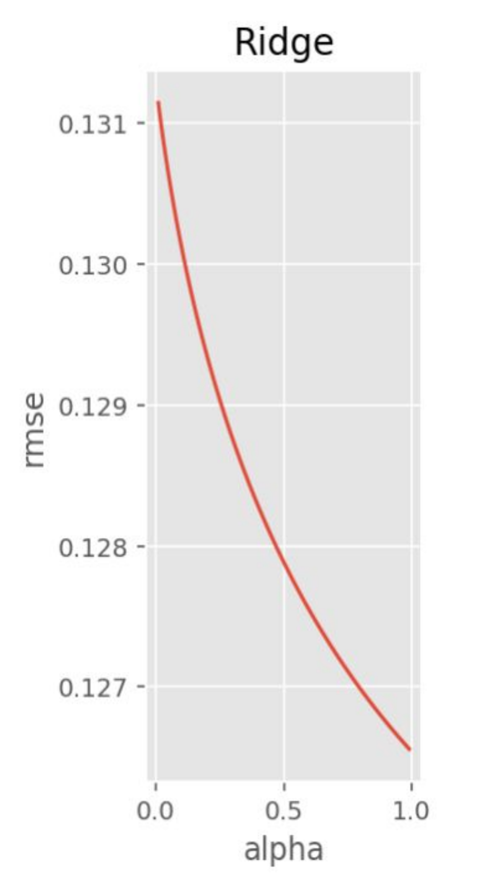
# Gradient Boosting - Overfitting in Kaggle

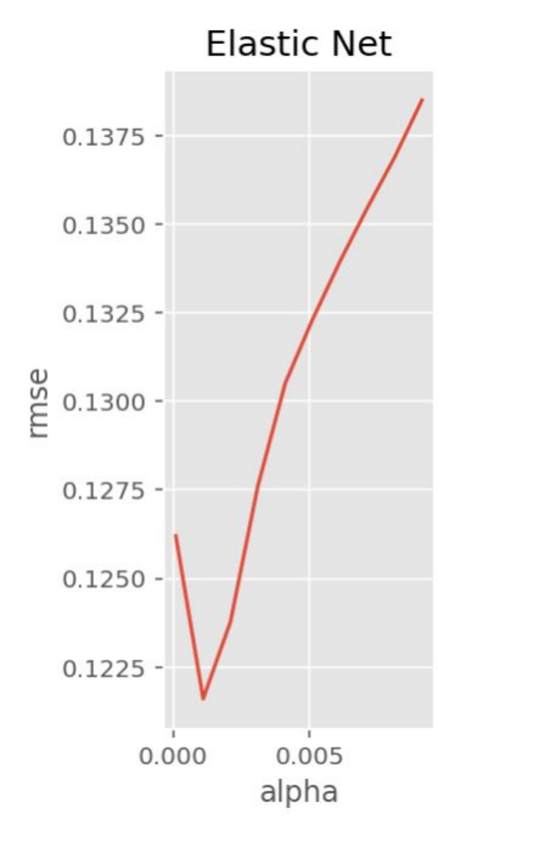
The Cross Validation Score: 0.11896

The Kaggle Score: 0.16694

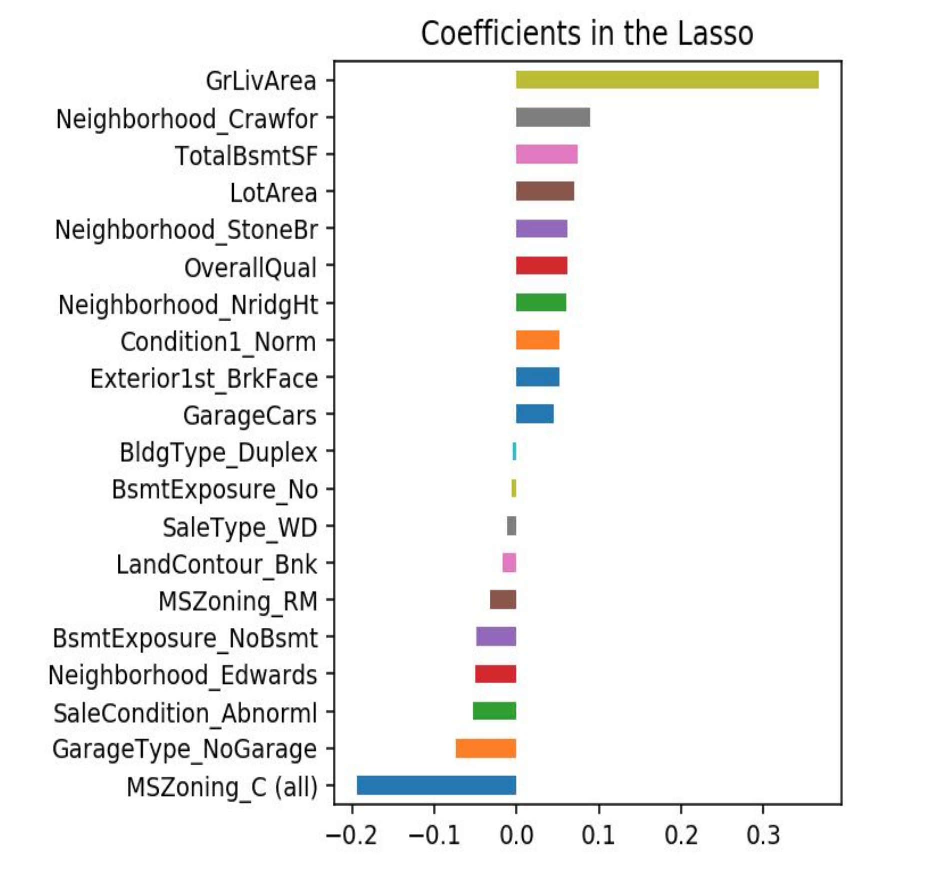
# Lasso, Ridge, Elastic Net Regression

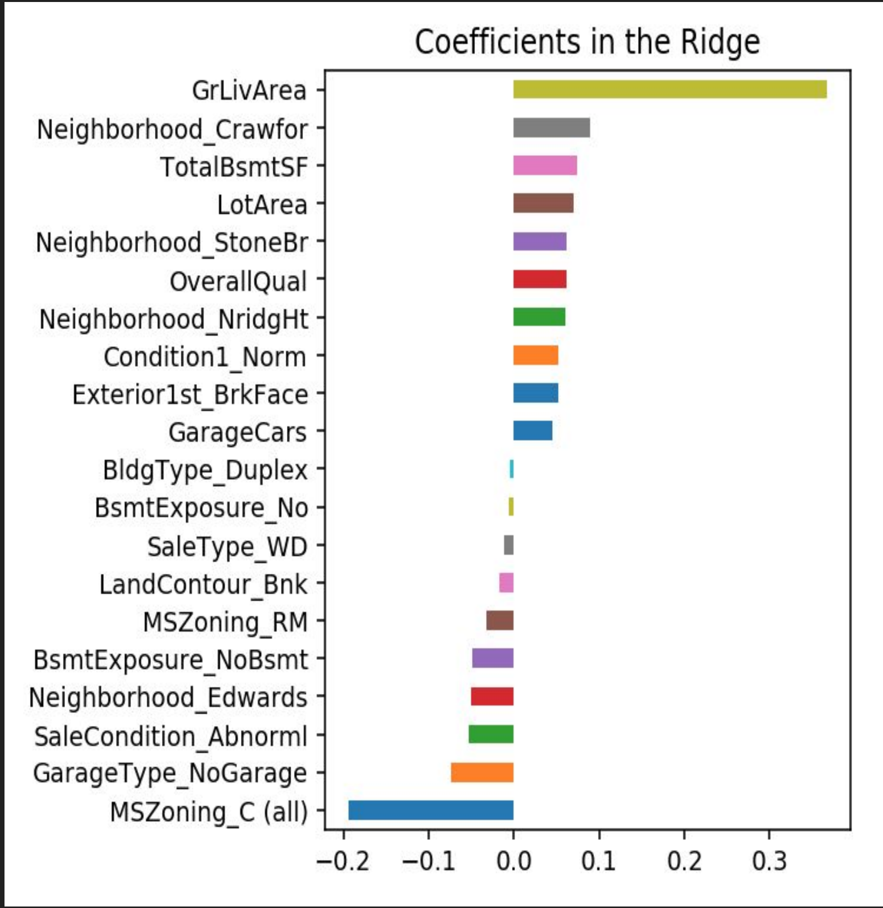


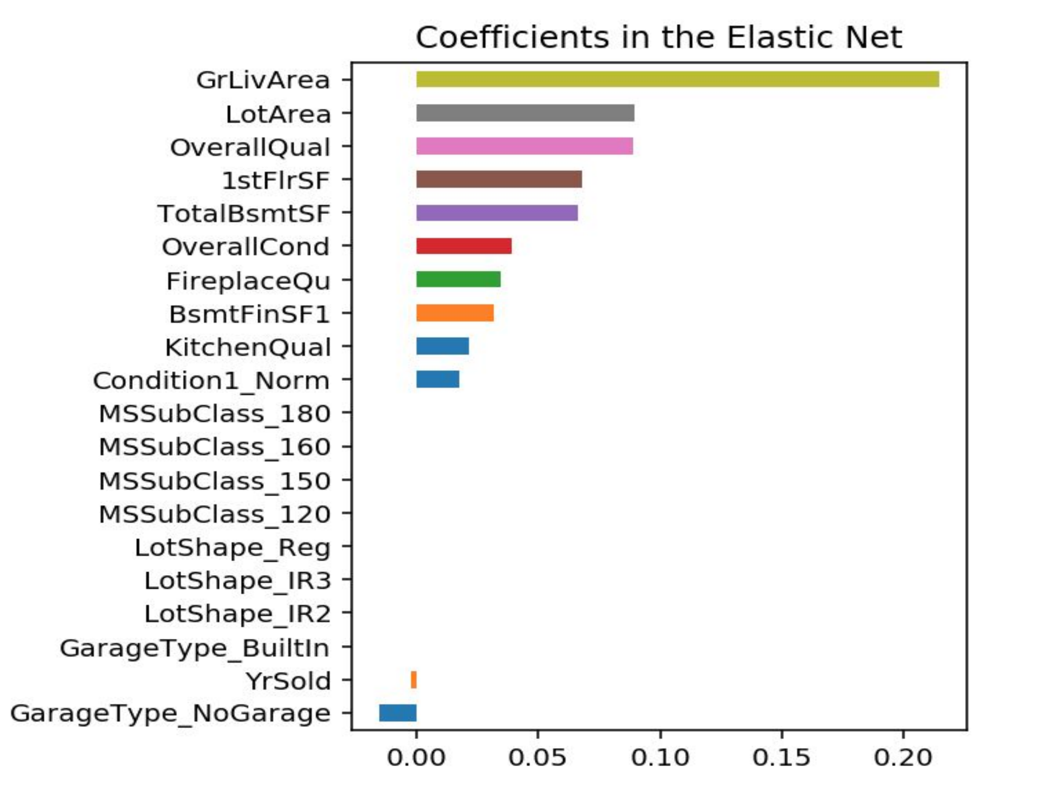



# Lasso, Ridge, Elastic Net Regression



# Lasso, Ridge, Elastic Net Regression

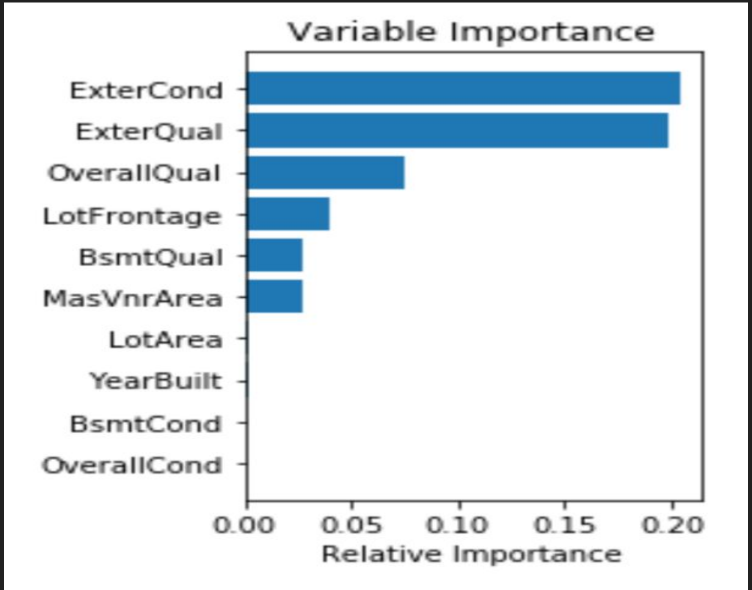


# Random Forest

Best parameters:

bootstrap: False, max_depth: 21, max_features: sqrt, min_samples_leaf: 1, min_samples_split: 2, n_estimators: 1000p

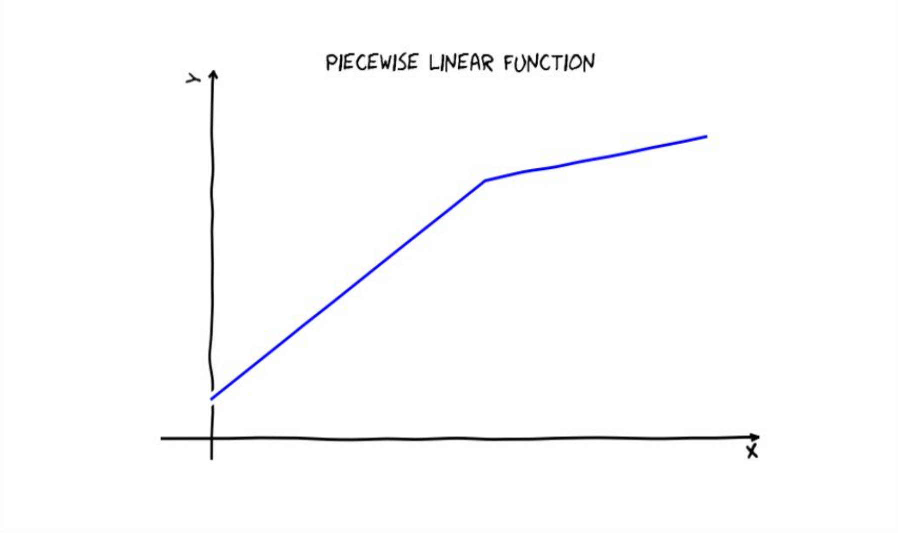
Variable importance

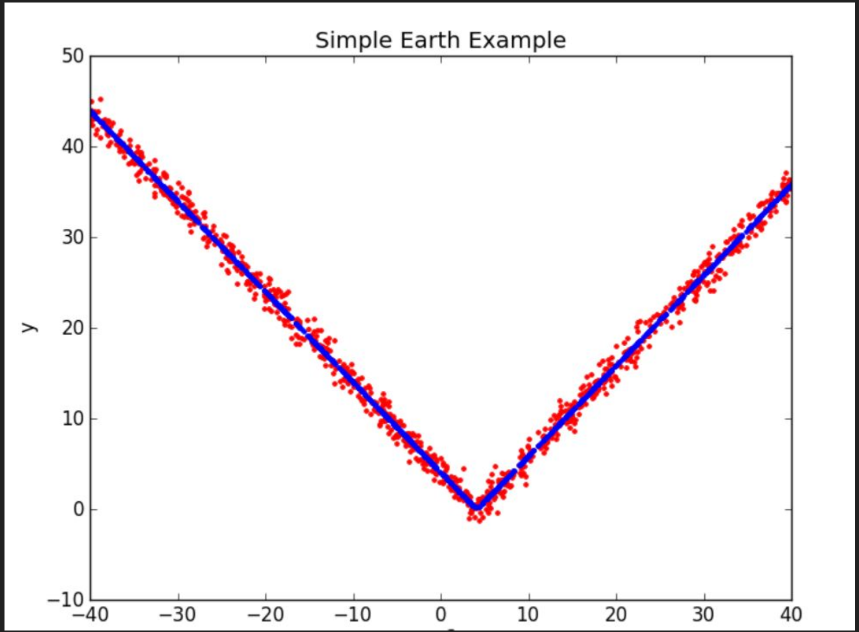


# Spline Regression

0.11357

$$y = 1 - 2\mathrm{h}(1 - x) + \frac{1}{2}\mathrm{h}(x - 1)$$



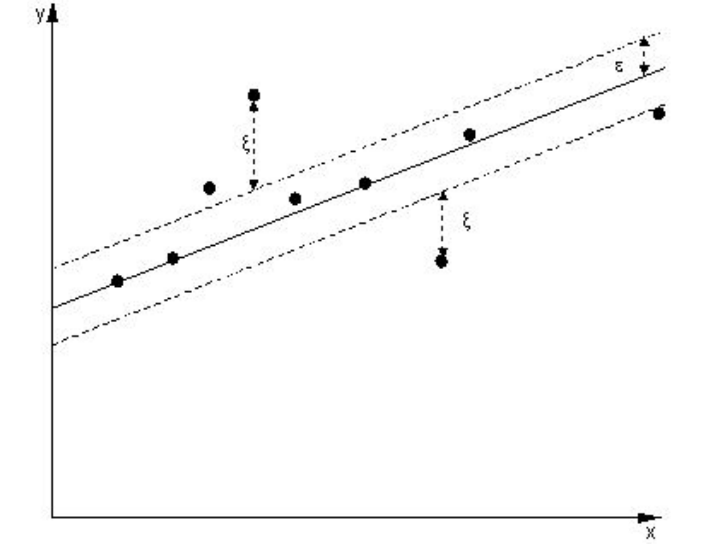



# Support Vector Regression

High dimensionality of data motivated the use of this regression

Kernel choice: Linear

Score: 0.1420617



# 3. Stacking

# Stacking

| Model | Test Score | Kaggle Score |
|---|---|---|
| Lasso | 0.122625 | 0.12249 |
| Ridge | 0.128345 | n/a |
| ElasticNet | 0.12554 | n/a |
| Spline | 0.11357 | n/a |
| Random Forest | 0.136219 | n/a |
| SVR | 0.14602 | n/a |
| Gradient Boosting | 0.11896 | 0.16694 |
| Stacking | n/a | 0.16356 |

# Lesson Learned

1. Data cleaning is very important and will take most of the time.
2. Give a hypothesis of which simple model may work best on the given data
3. Implement the simple model
4. UNDERSTAND the model, and why it gave the output it did
5. Update hypothesis
6. Repeat (2)