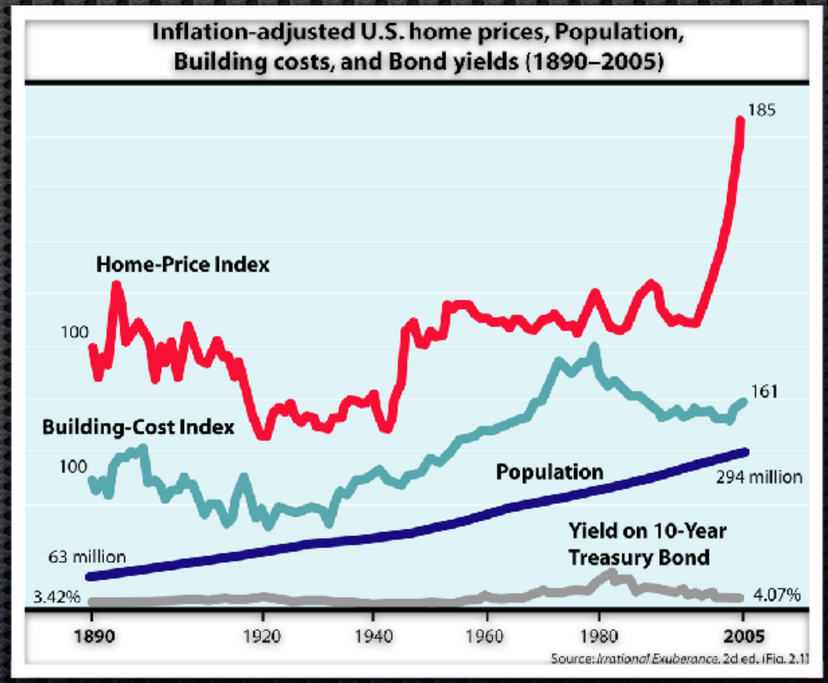
Predicting Sales Prices for the Kaggle Ames, Iowa Housing Dataset

Team Proxima Centauri - Wenchang, Kenny, James, Danny

Brief Overview of Data Set

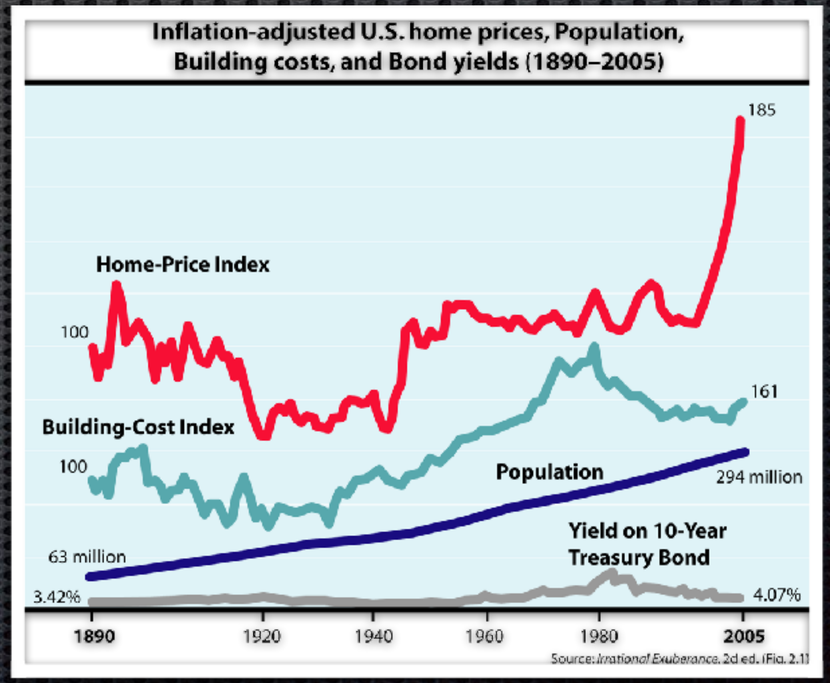
- 80 variables tied to price of home sales in Ames, Iowa between 2006-2010
- Initially obtained from Ames City Assessor's Office with 113 variables that described 3970 property sales -> cleaned and collated by Dean De Cock {Journal of Statistics Ed., **19**, (3) 2011}
- Reduced to 80 predictors:
  - 20 continuous variables {total dwelling sq. footage etc}
  - 14 discrete variables {number of bathrooms, kitchens}
  - 46 categorical variables - 23 nominal/23 ordinal {street pavement, neighborhood}

The US Housing Bubble



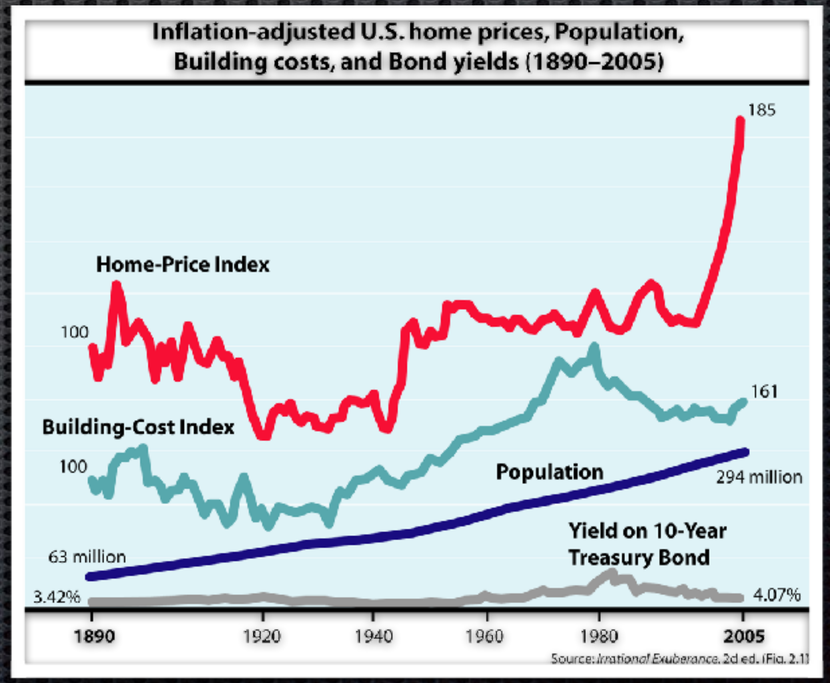
The US Housing Bubble

- What was going on between 2006-2010 in our country?
- Will there be a noticeable effect?



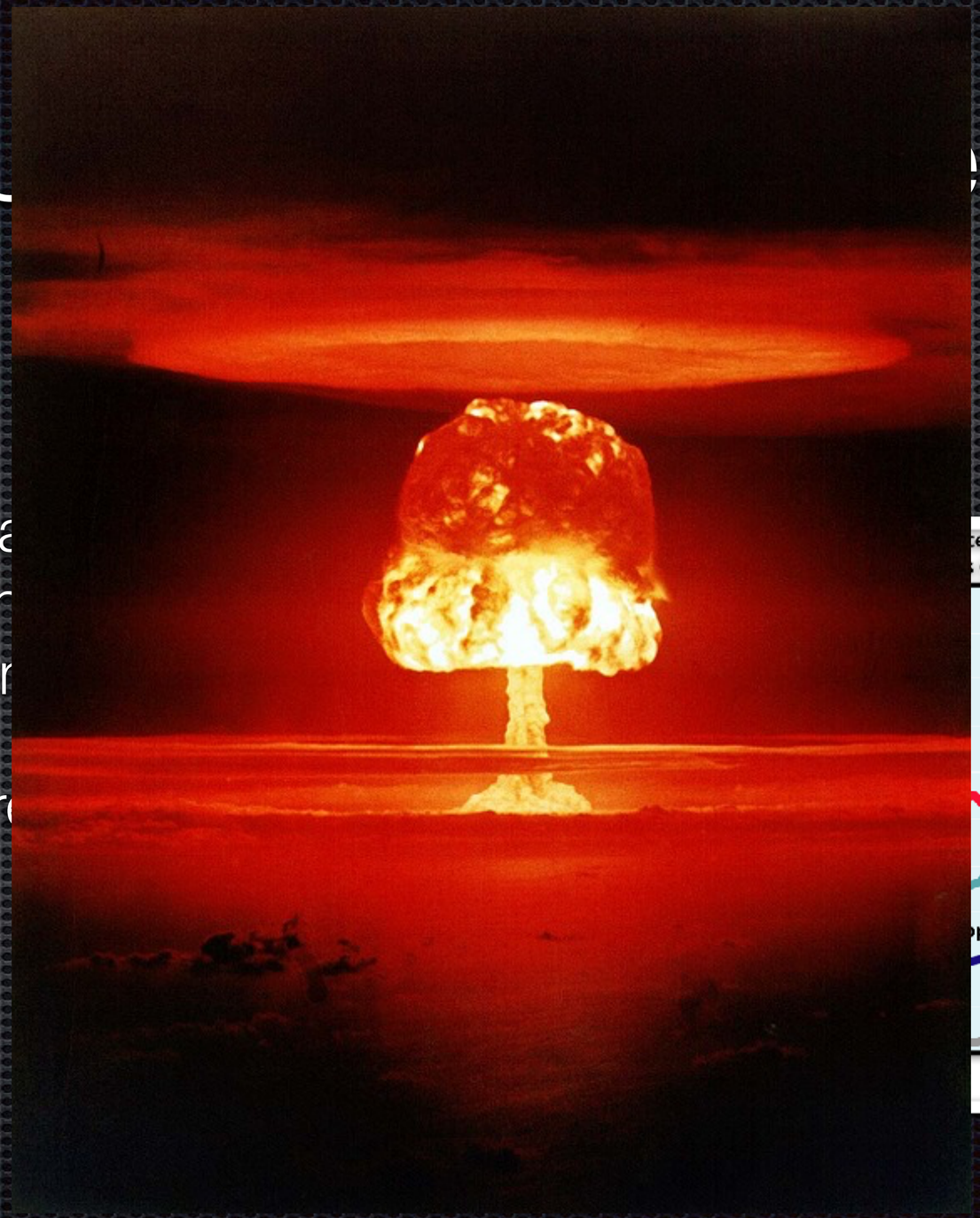
The US Housing Bubble

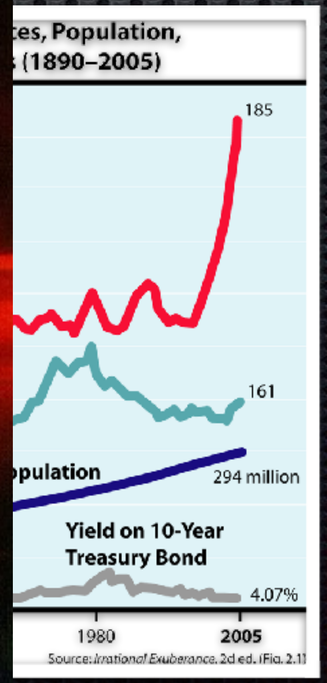
- What was going on between 2006-2010 in our country?
- Will there be a noticeable effect?



The U

- What wa
  betweer
  our cour
- Will ther
  effect?





Considerations Involving Home Prices

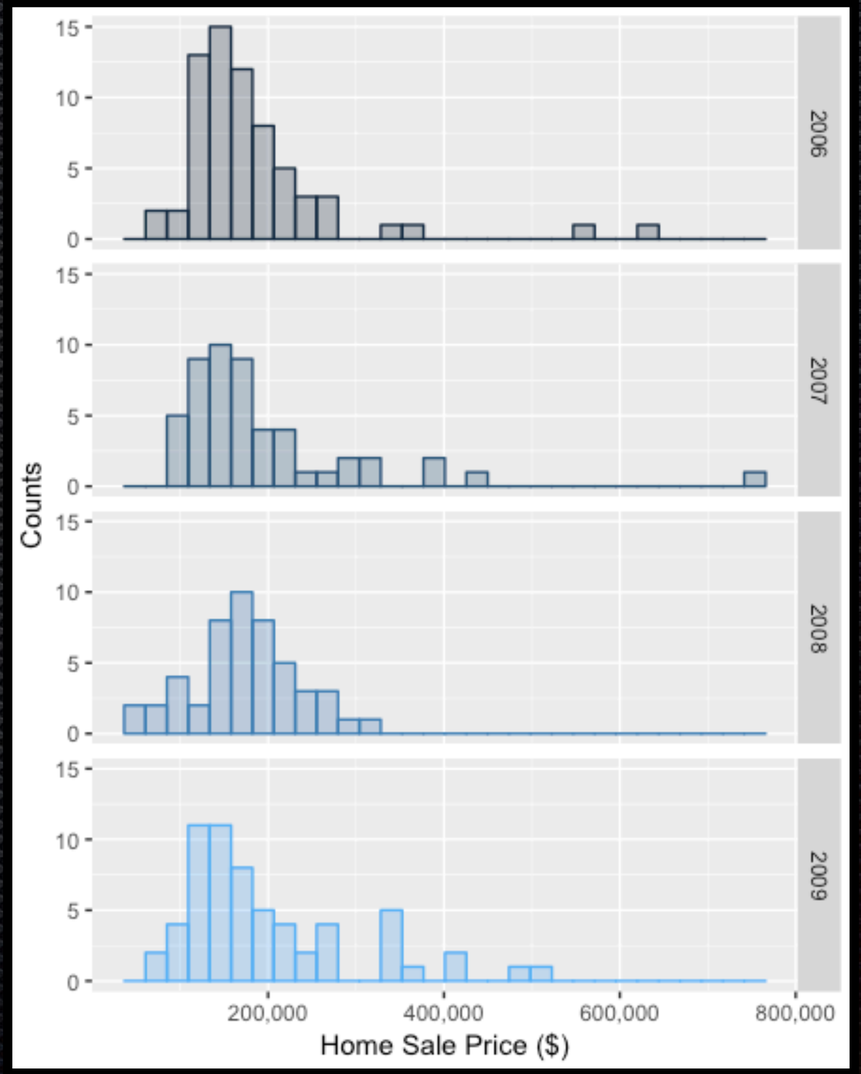
How did monthly home sales vary during the timespan represented in dataset?

https://plot.ly/~dbubb/13/

Buyer's market in Spring, Seller's in Fall

Kruskall-Wallis (~Anova): p=0.97, df=3, χ^2 = 0.24345

Levene(~Barlett's): p=0.97, df=3, F = 1.325



Considerations Involving Home Prices

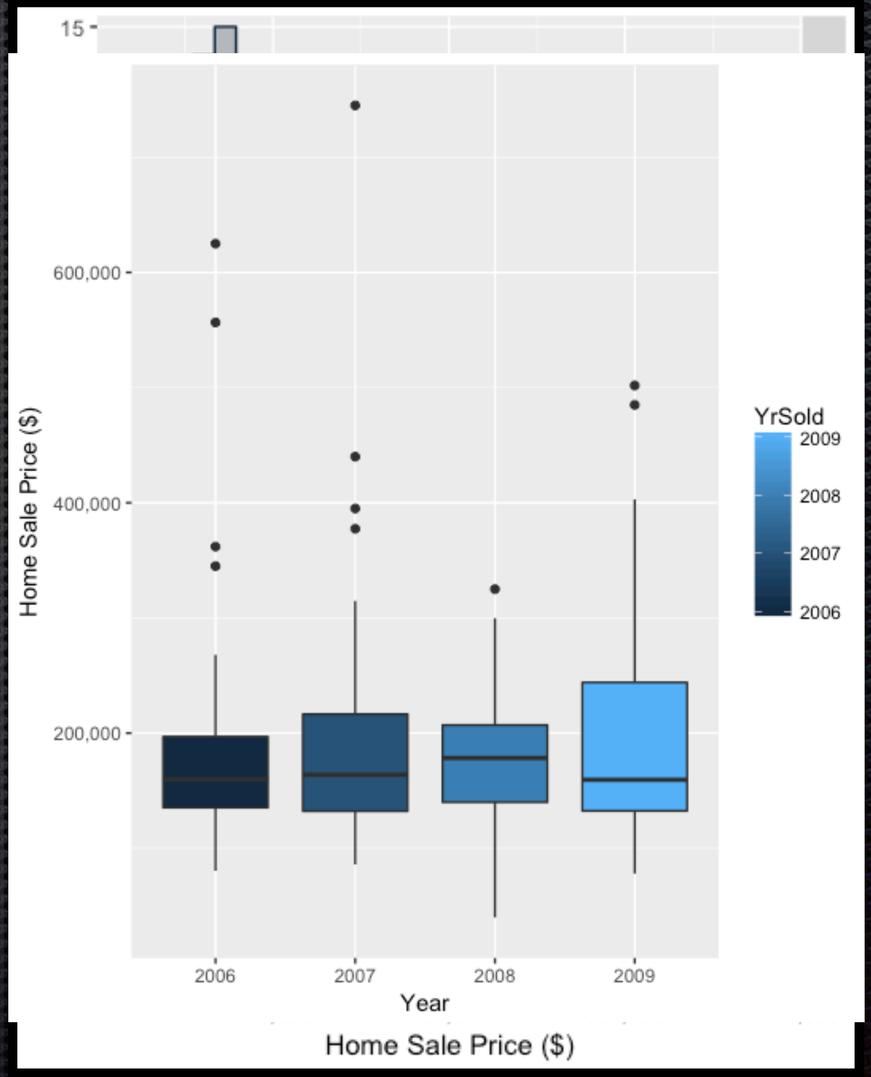
How did monthly home sales vary during the timespan represented in dataset?

https://plot.ly/~dbubb/13/

Buyer's market in Spring, Seller's in Fall

Kruskall-Wallis (~Anova): p=0.97, df=3, χ^2 = 0.24345

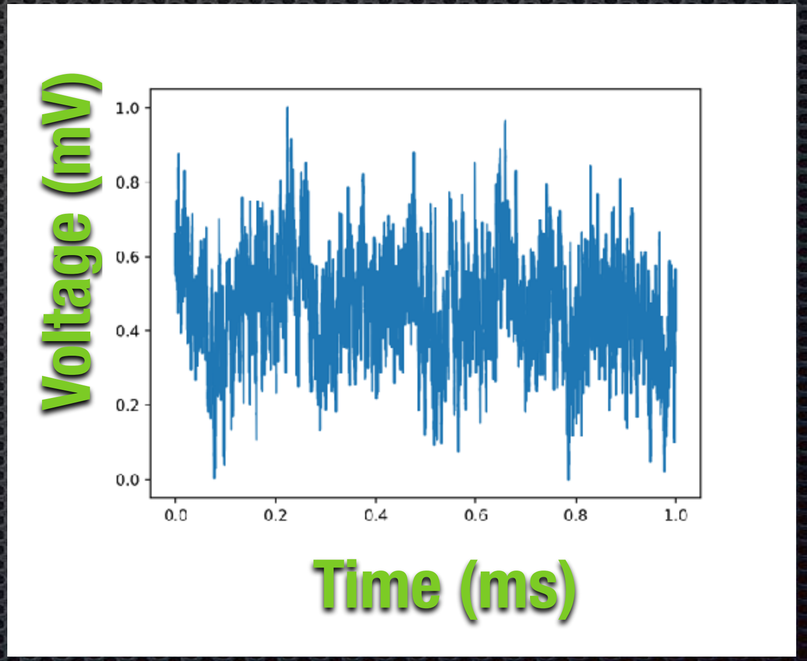
Levene(~Barlett's): p=0.97, df=3, F = 1.325



Hurst exponent

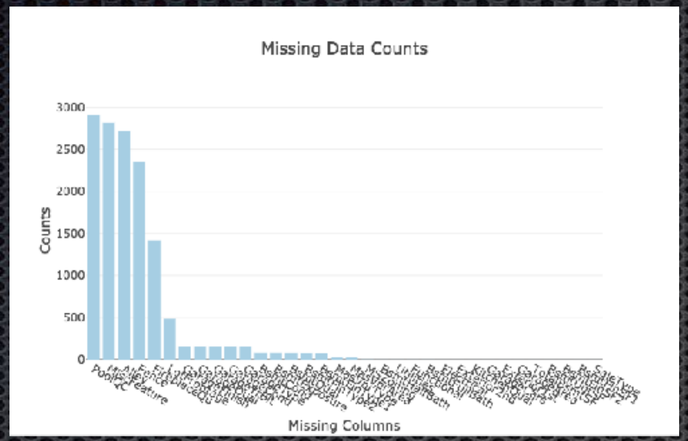
$$D_f = -\lim_{\epsilon \to 0} \frac{ln(N(\epsilon))}{ln(\epsilon)}$$

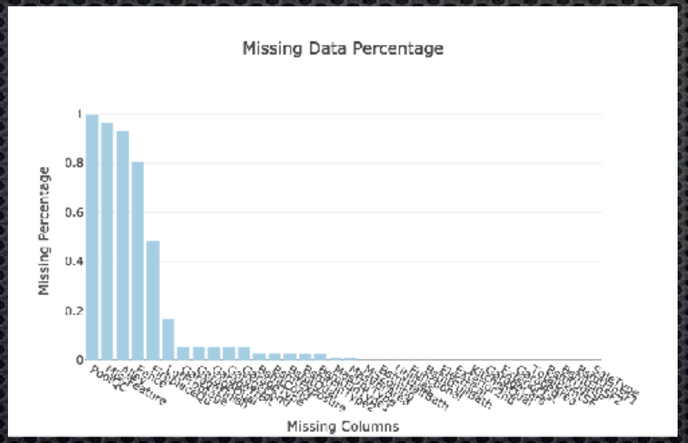
- H = 2 - D_f = 0.536 - ~ Brownian Motion
  - 0 < H < 0.5 - mean-reverting, anti-persistent
  - 0.5 < H < 1- persistent $\Longrightarrow$ increases tend to follow increases, same for decreases…



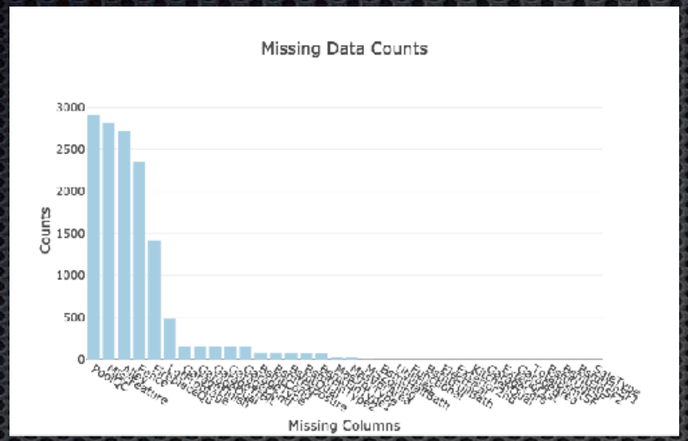
Good discussion here re: S&P 500

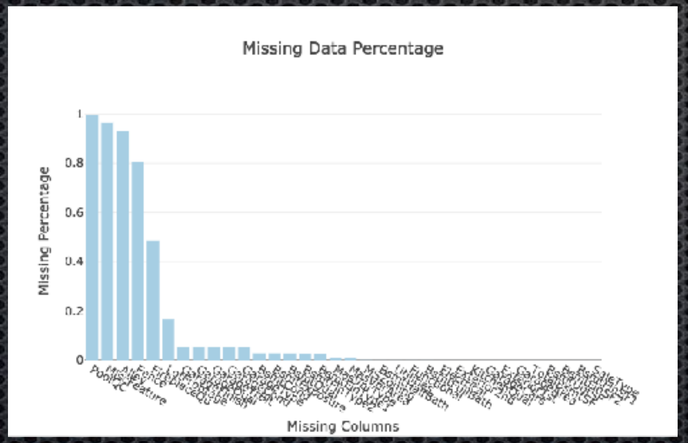
Data Cleaning, Feature Eng.





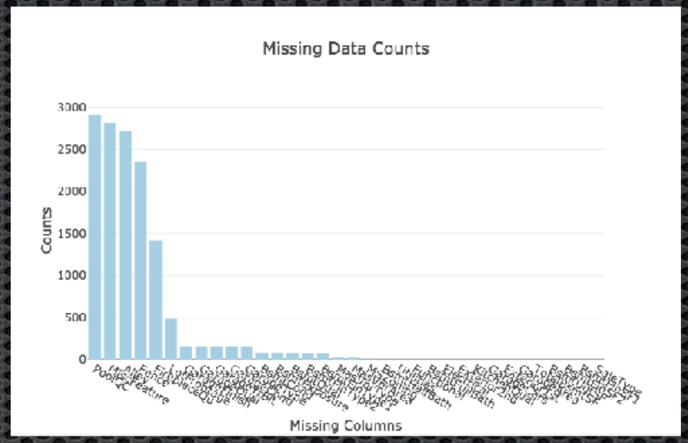
Data Cleaning, Feature Eng.

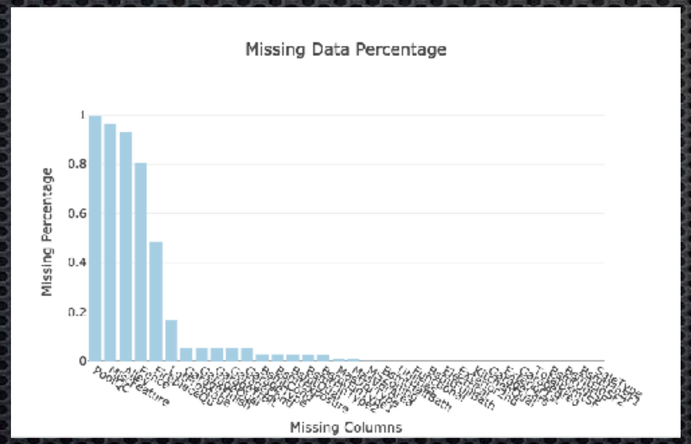




- Missingess
- Imputation
- Added features
- Normalization

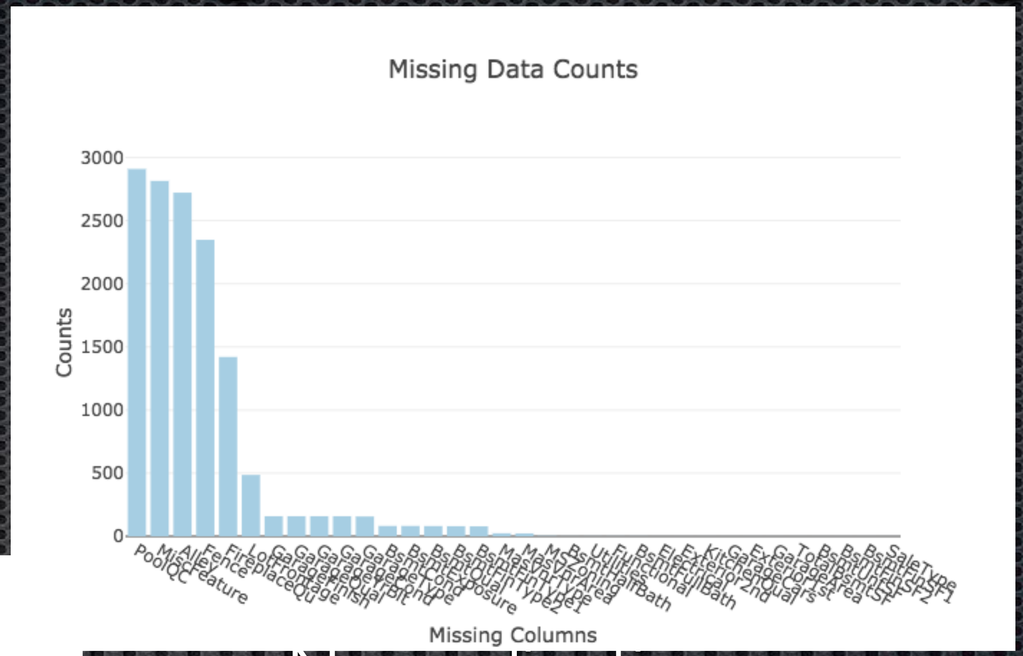
Data Cleaning, Feature Eng.

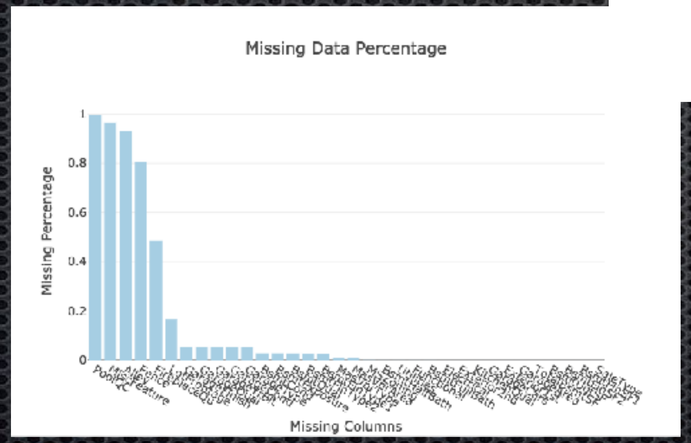




- Normalization

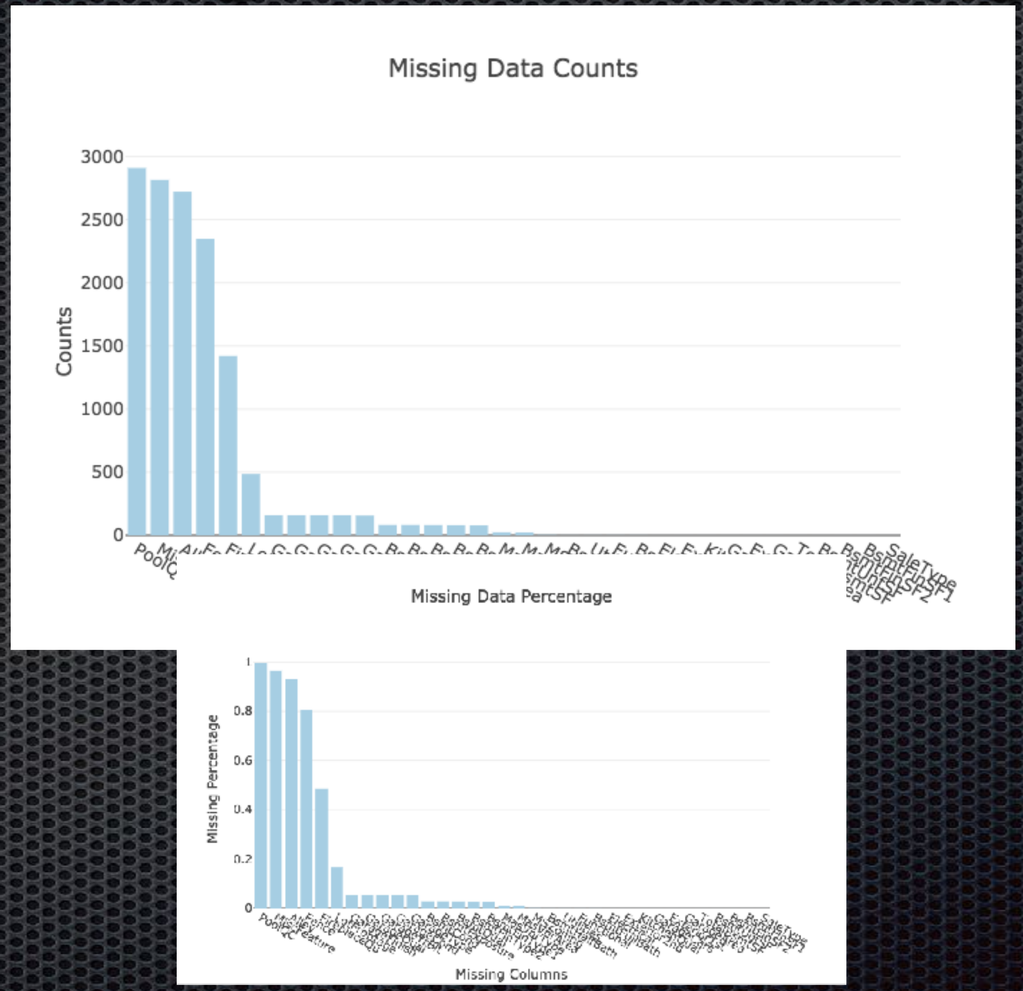
Data Cleaning, Feature Eng.





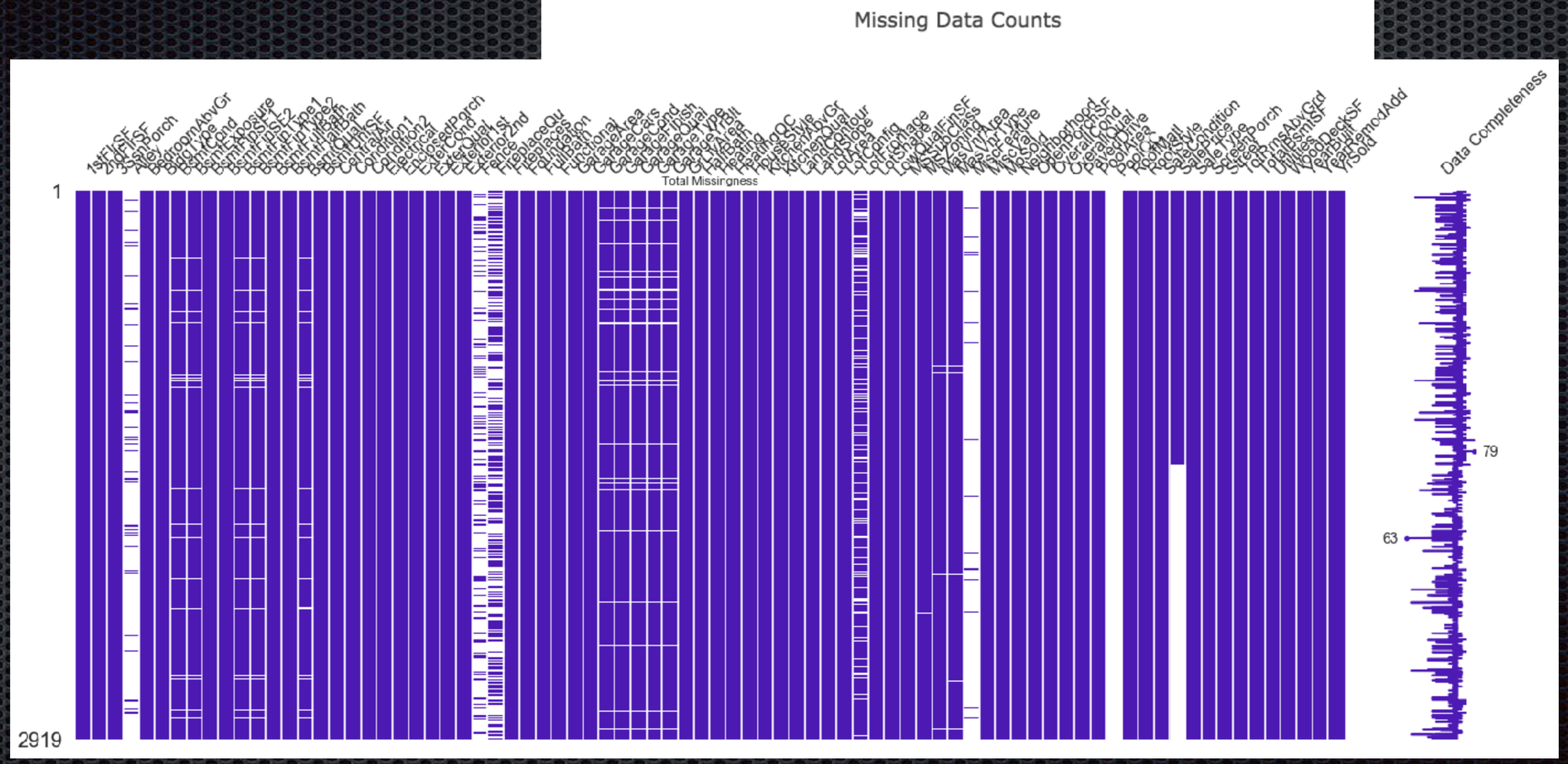
- Normalization

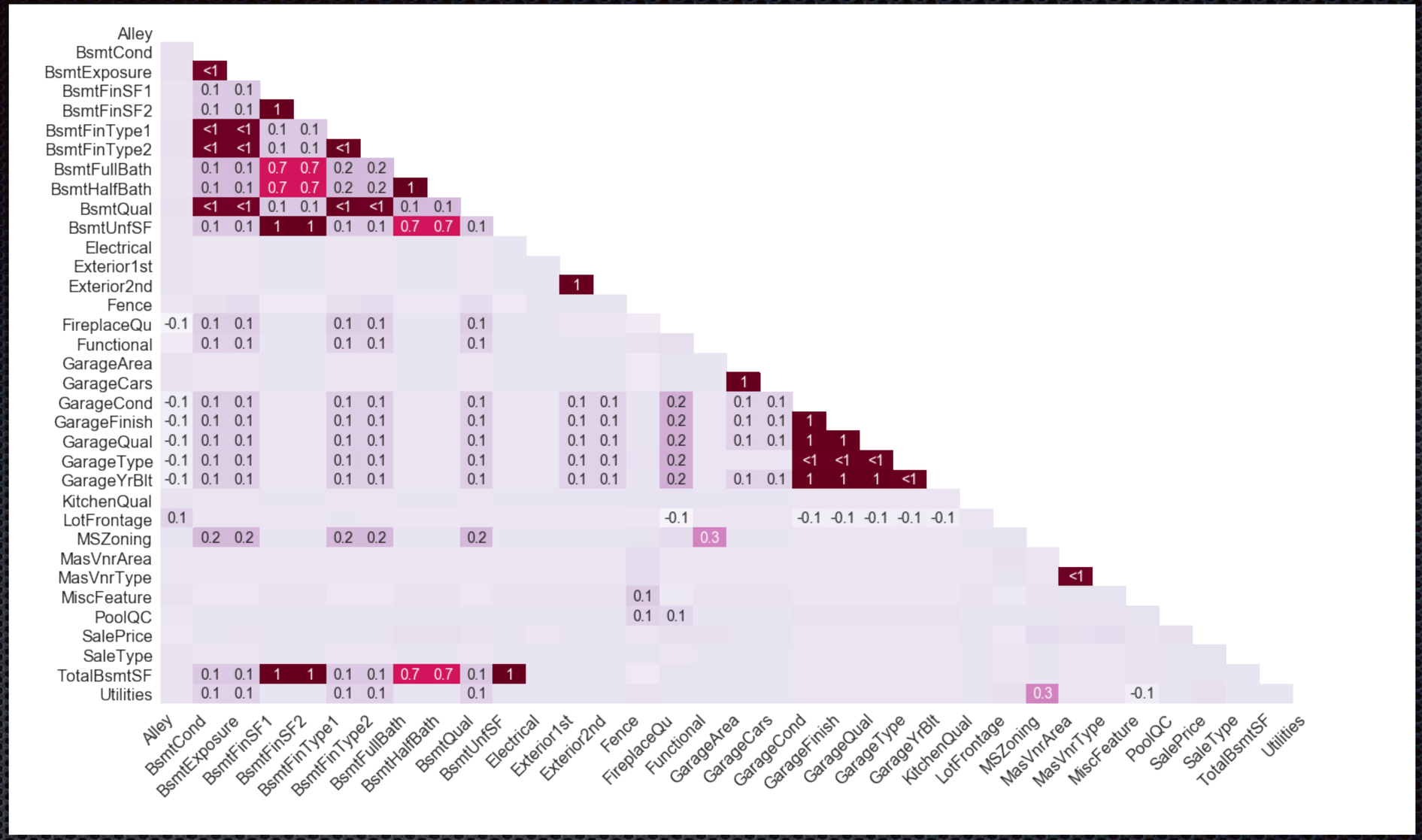
Data Cleaning, Feature Eng.



Data Cleaning, Feature Eng.

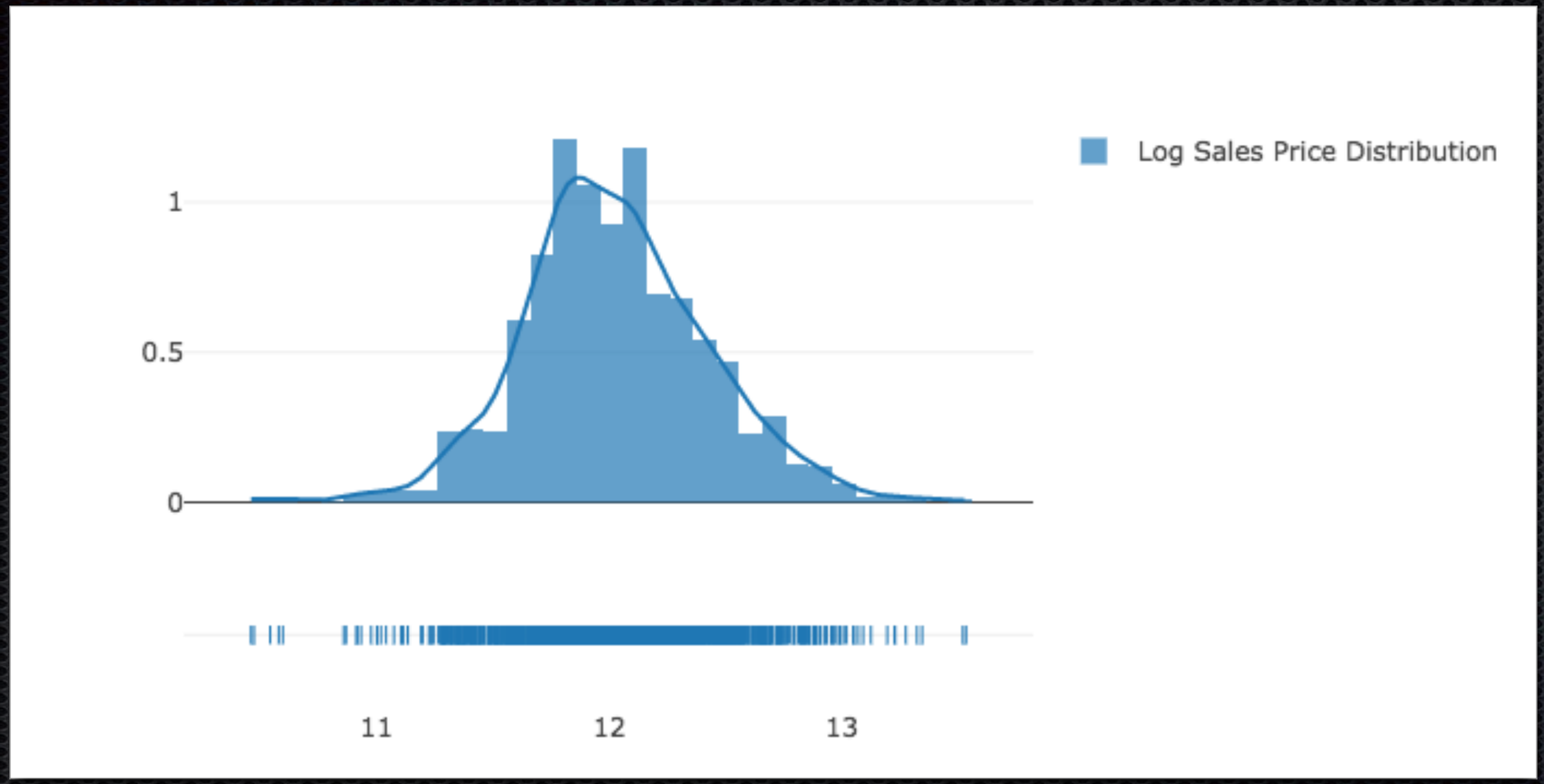
Data Cleaning, Feature Eng.





Correlated Missingness?

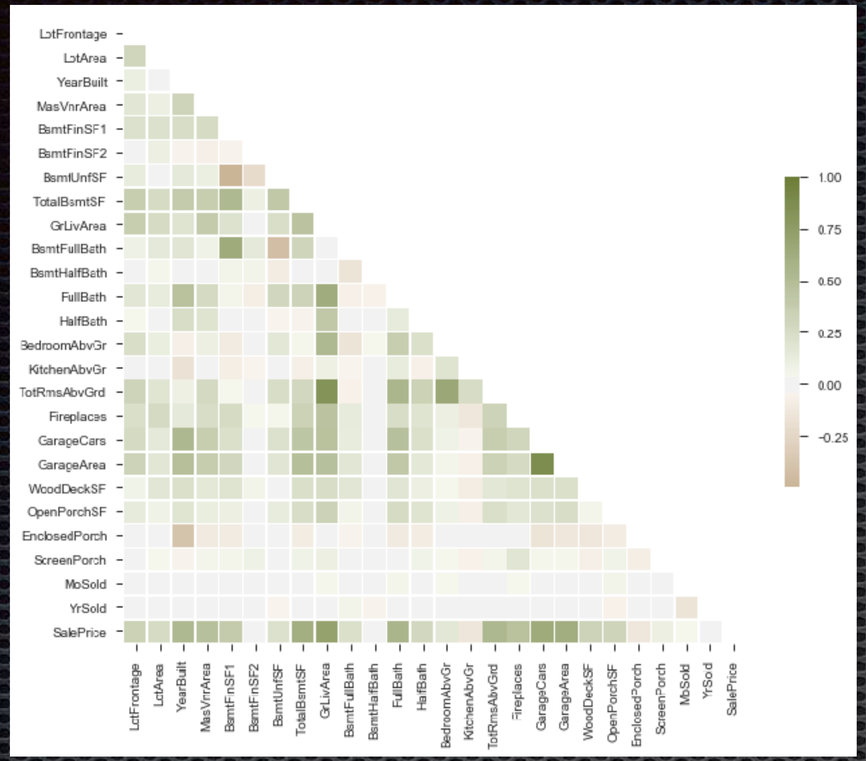
Dealing with Skew and ‘Dummifying’

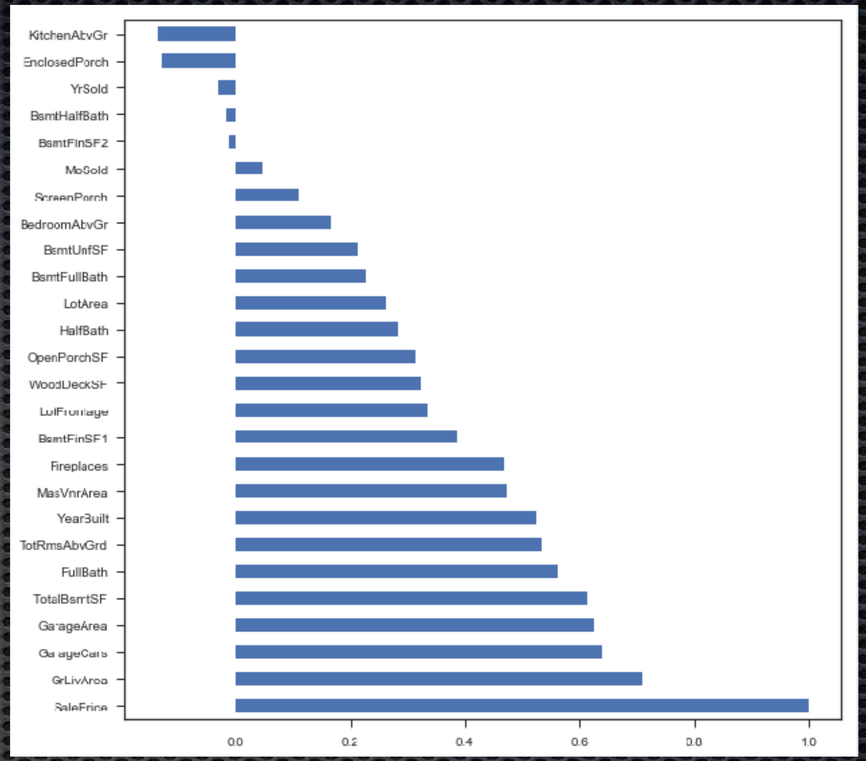


Many variables were highly skewed, so we chose to apply a log transformation (skew > 0.6)
‘One-hot’ encoding was used in conjunction with dummy variables to deal with categorical data

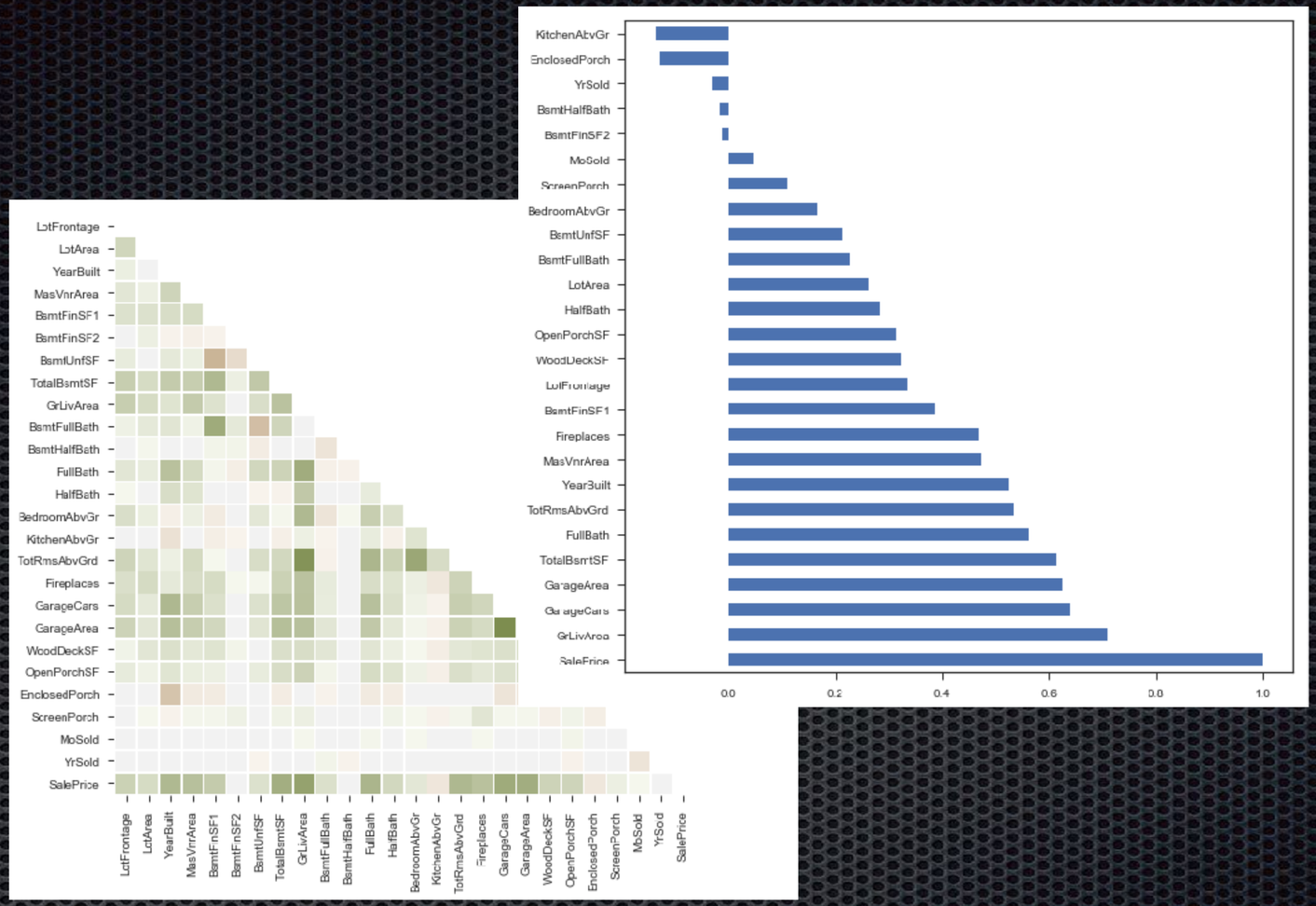
Feature Engineering

- Bathroom Capacitance
- Parking Capacitance



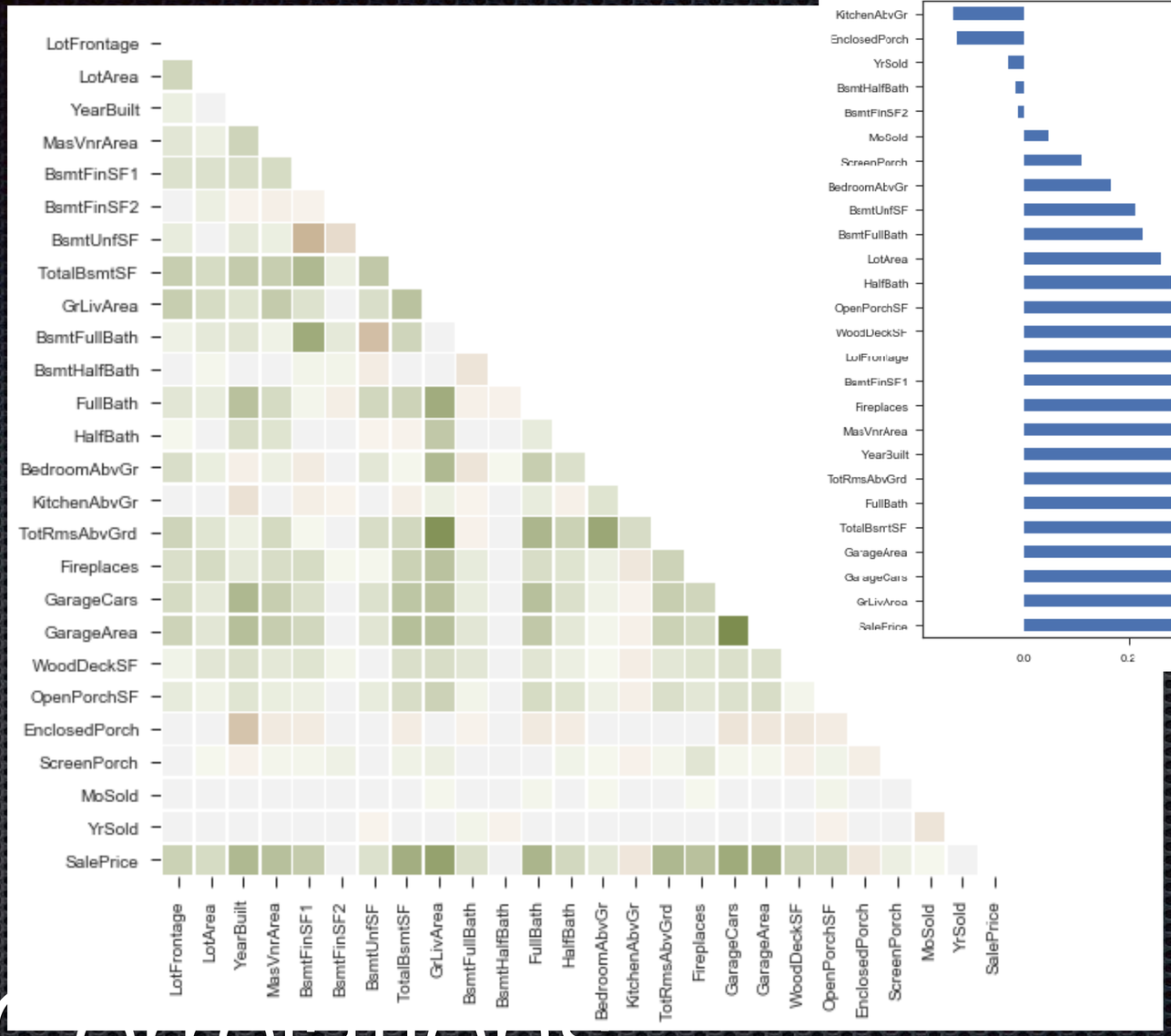


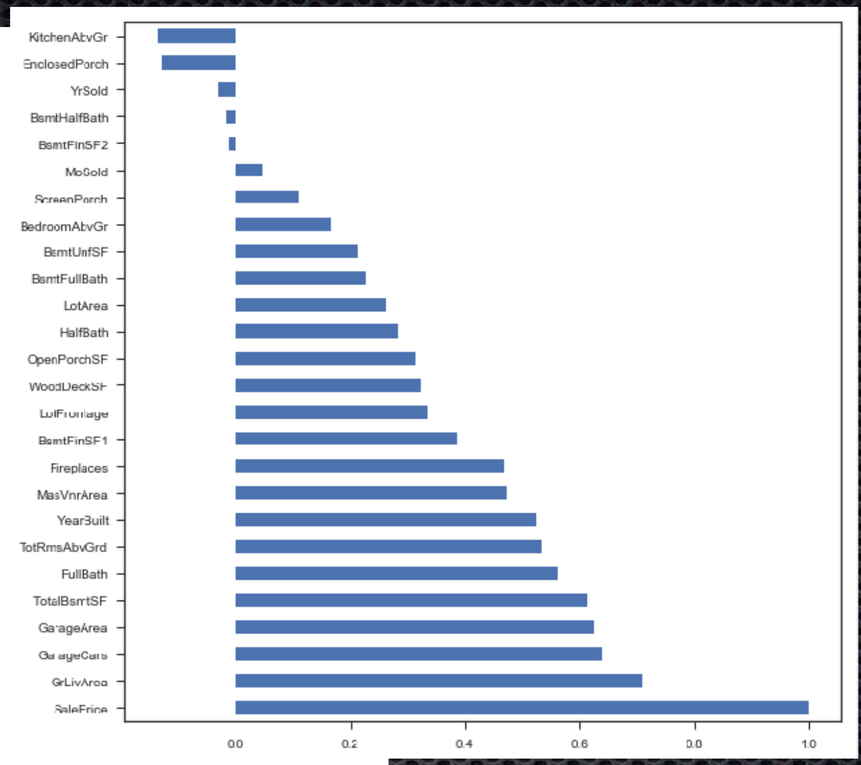
Correlations



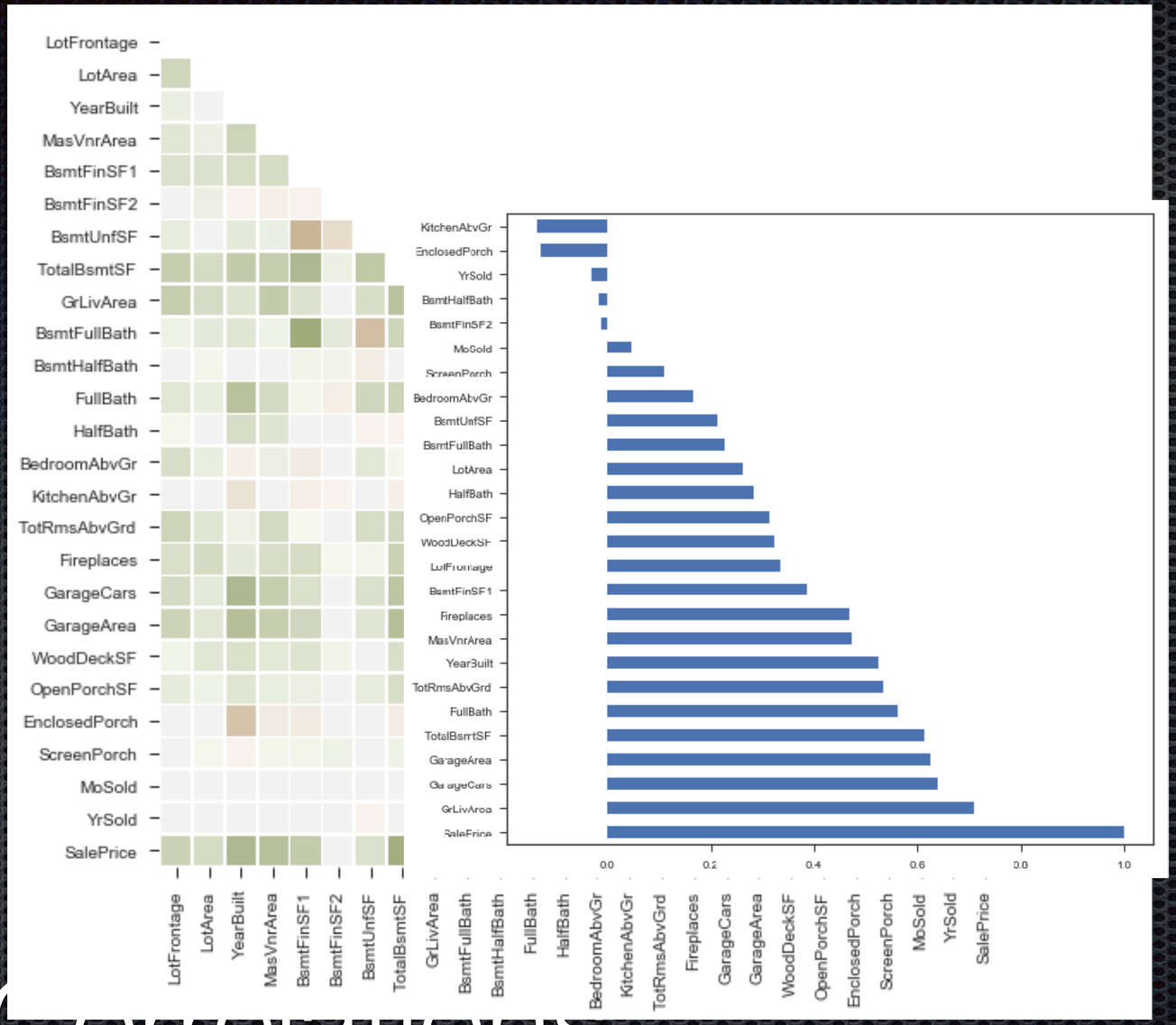
Correlations

Correlations

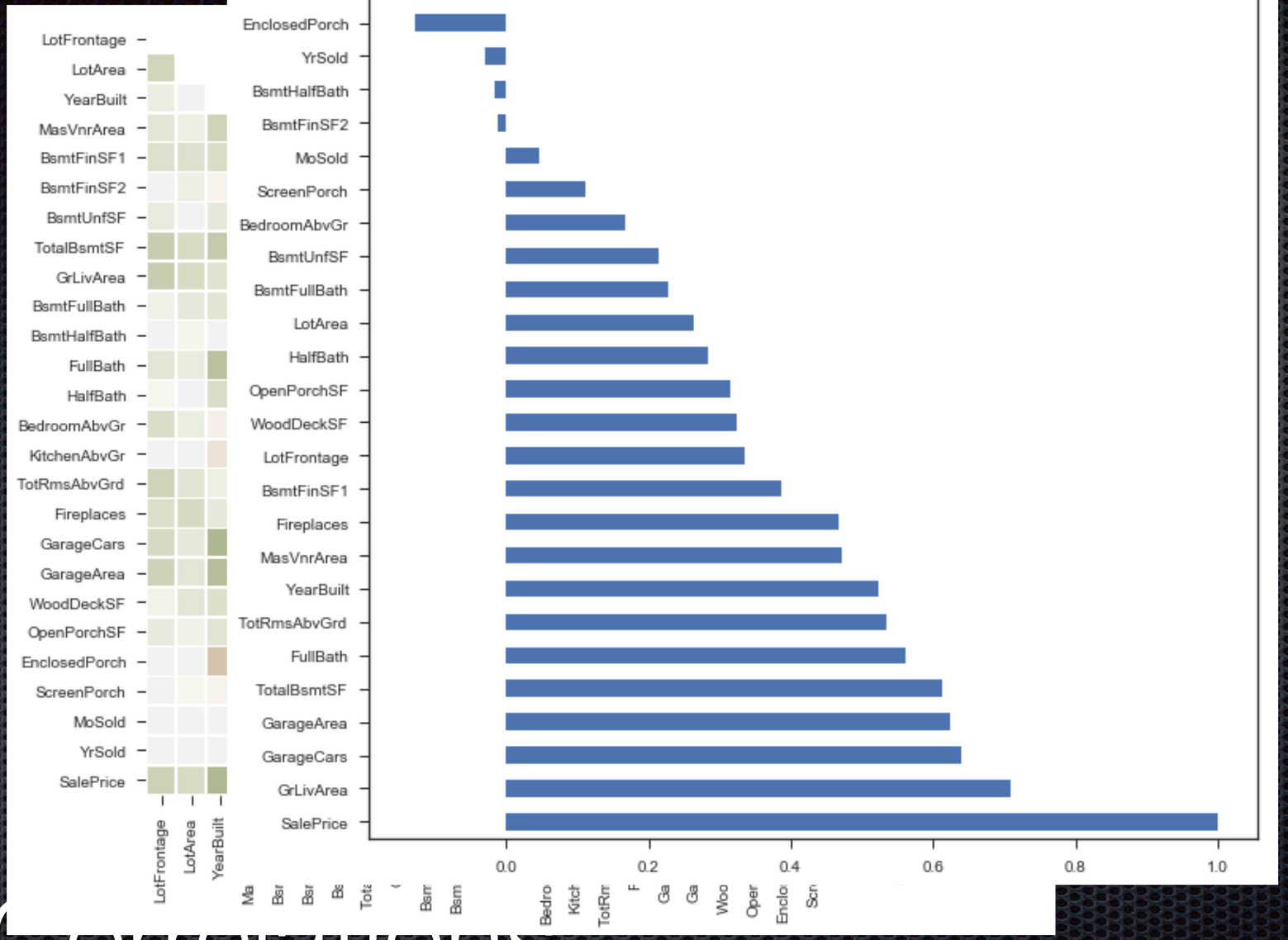




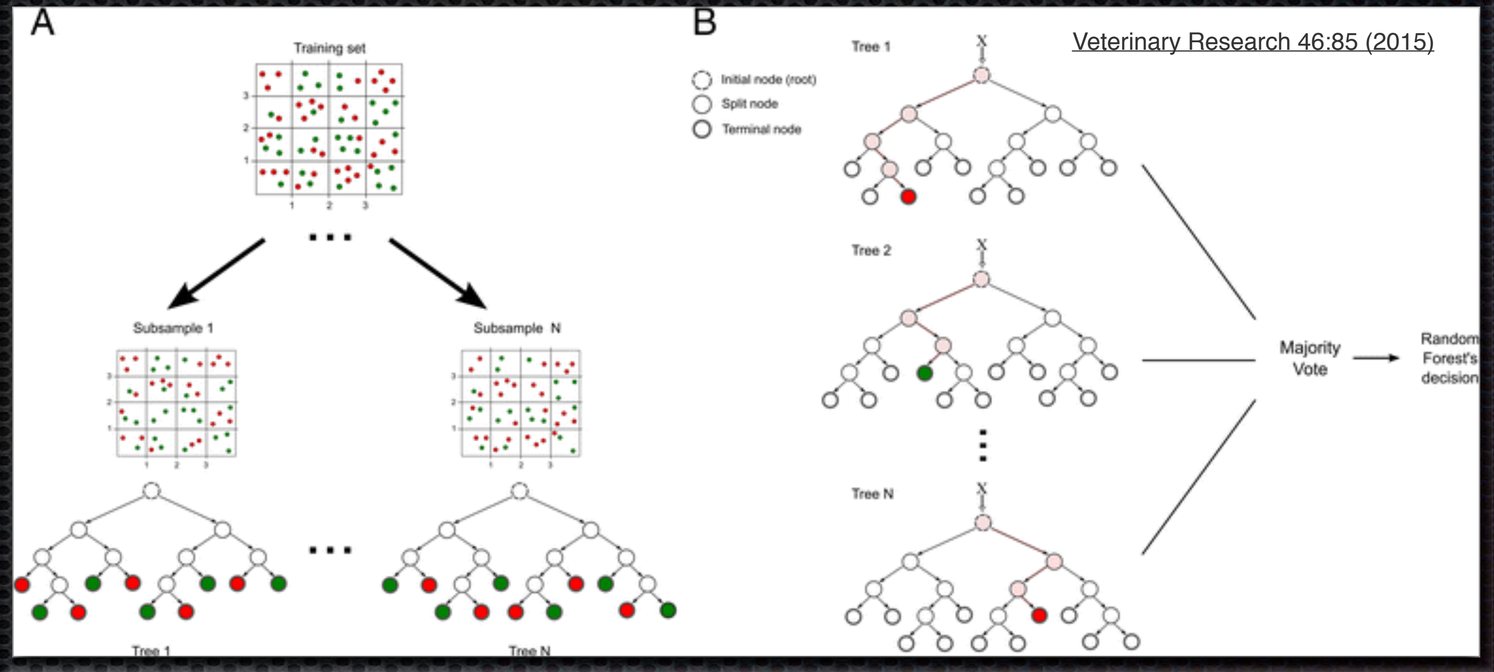
Correlations



Correlations



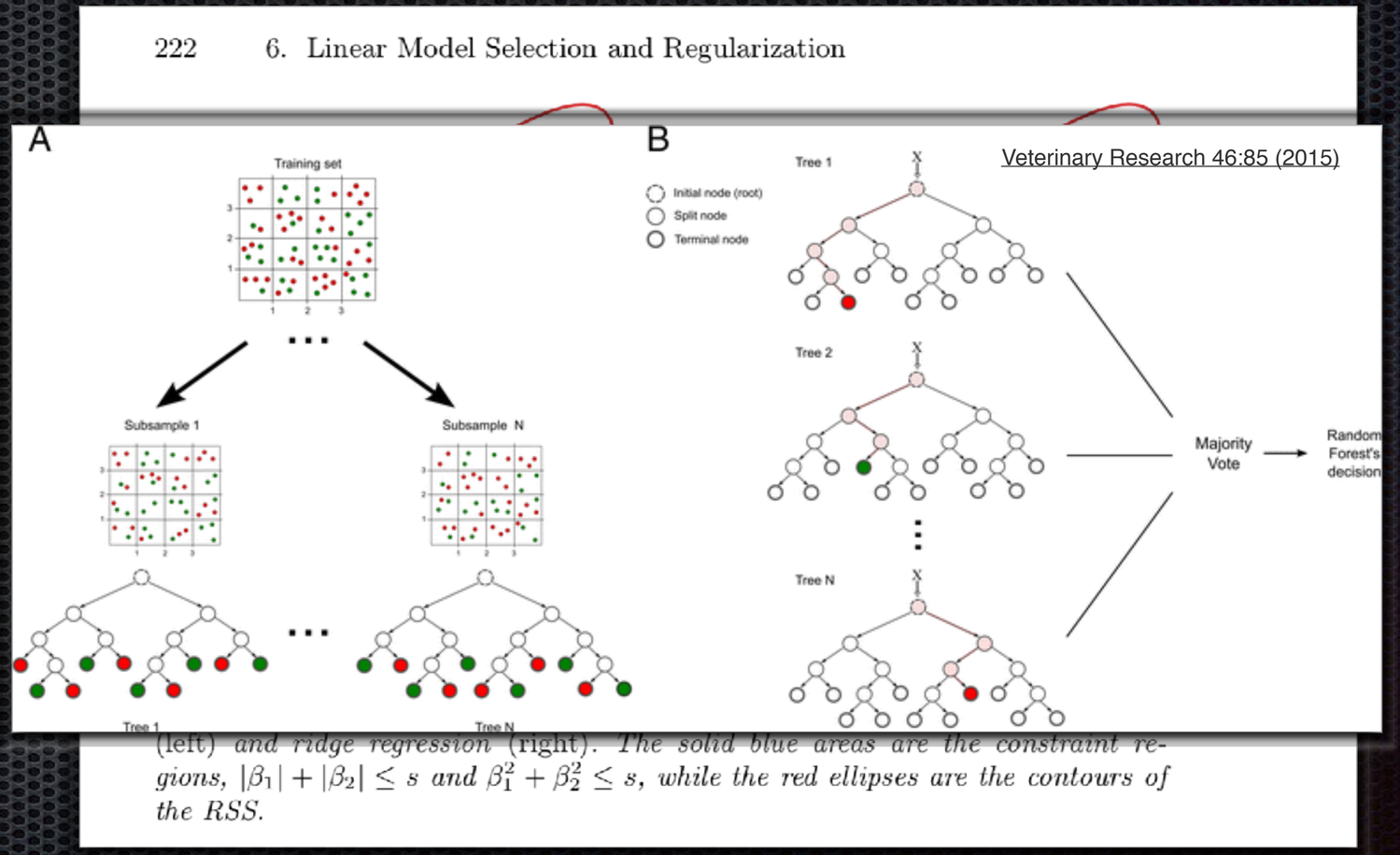
Models Investigated



Veterinary Research 46:85 (2015)

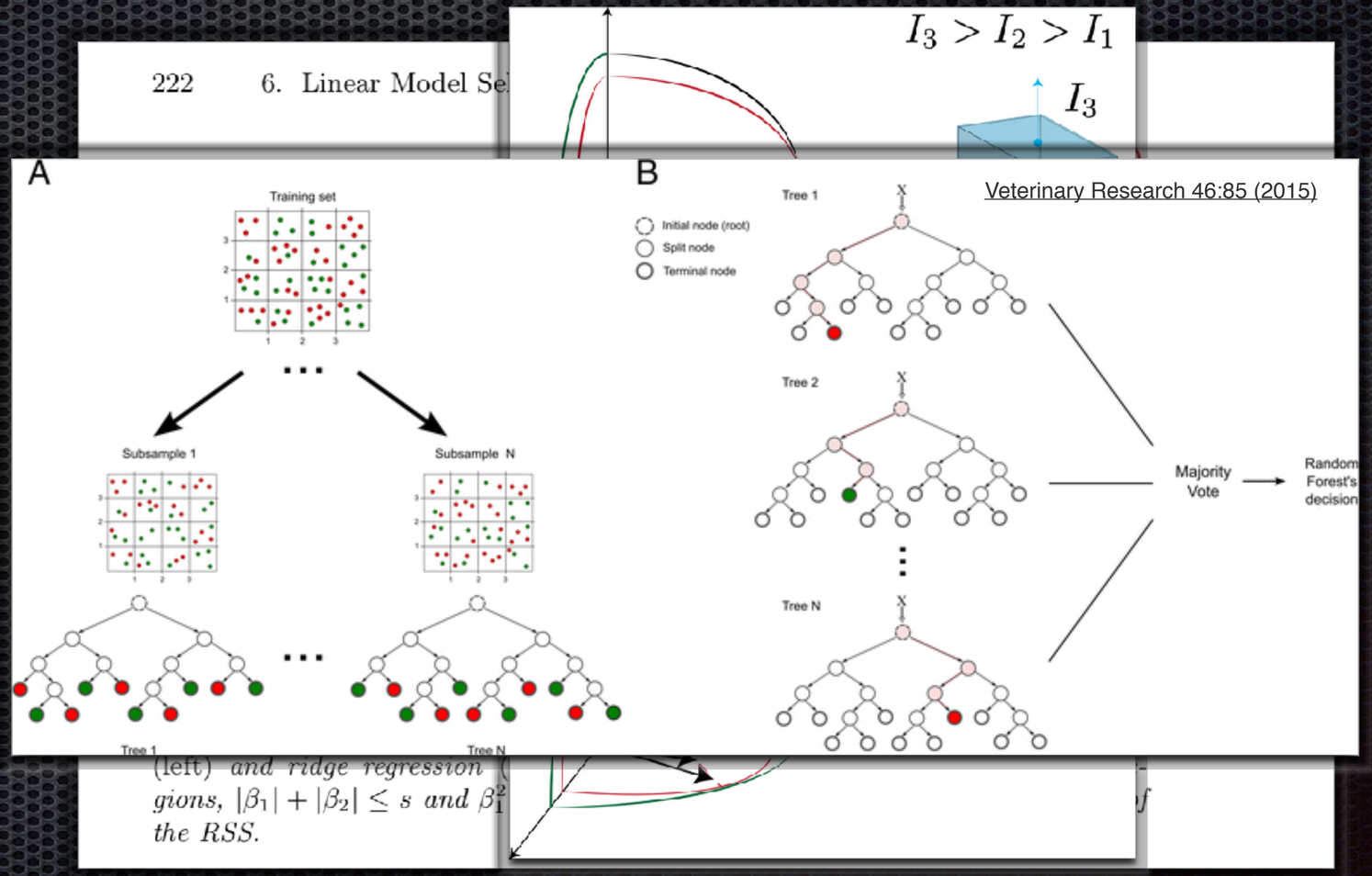
Models Investigated

- Regularized Regression
  - Ridge
  - Lasso



Models Investigated

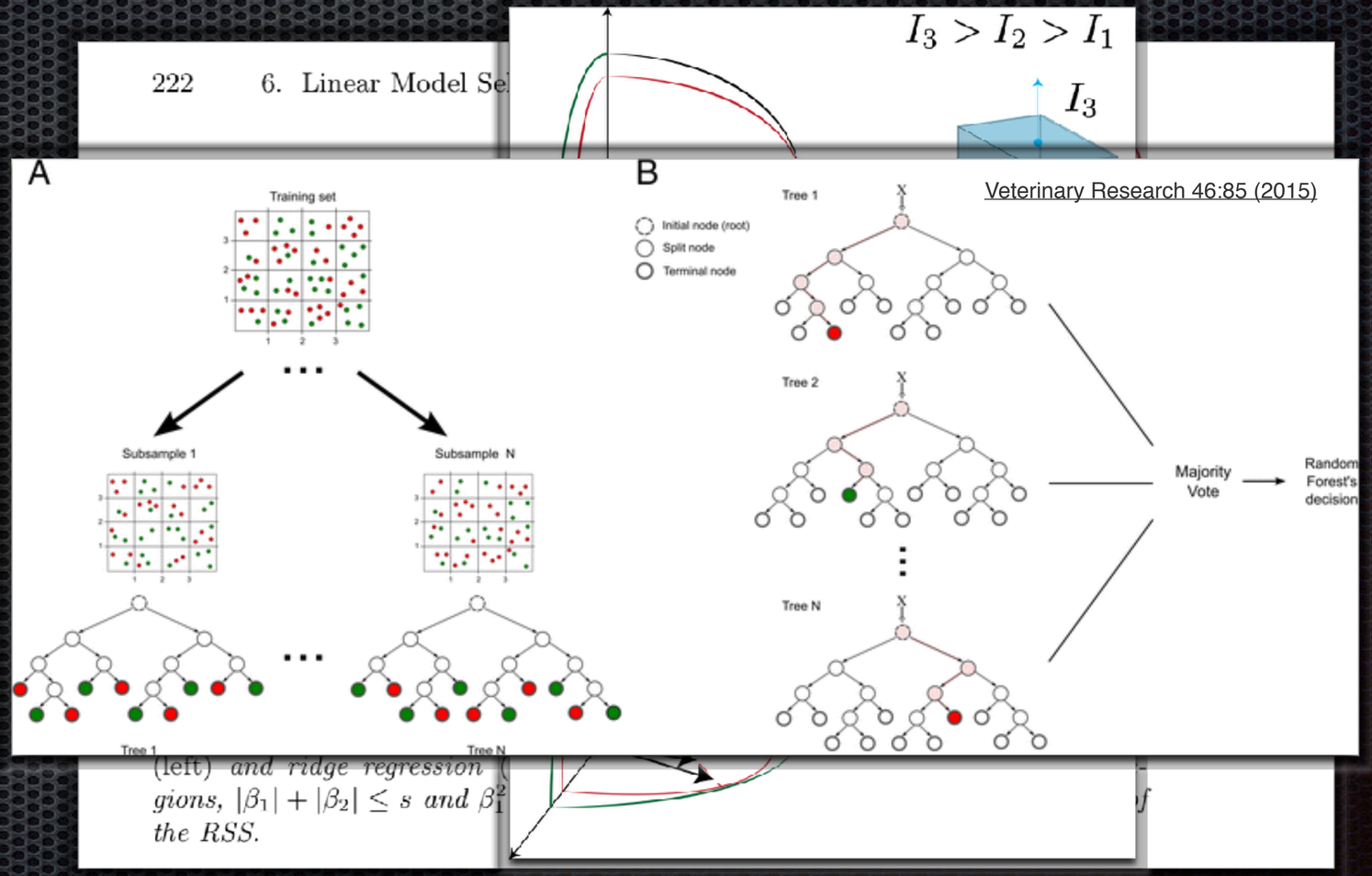
- Regularized Regression
  - Ridge
  - Lasso
- PCA - MLR, XGBoost



Veterinary Research 46:85 (2015)

Models Investigated

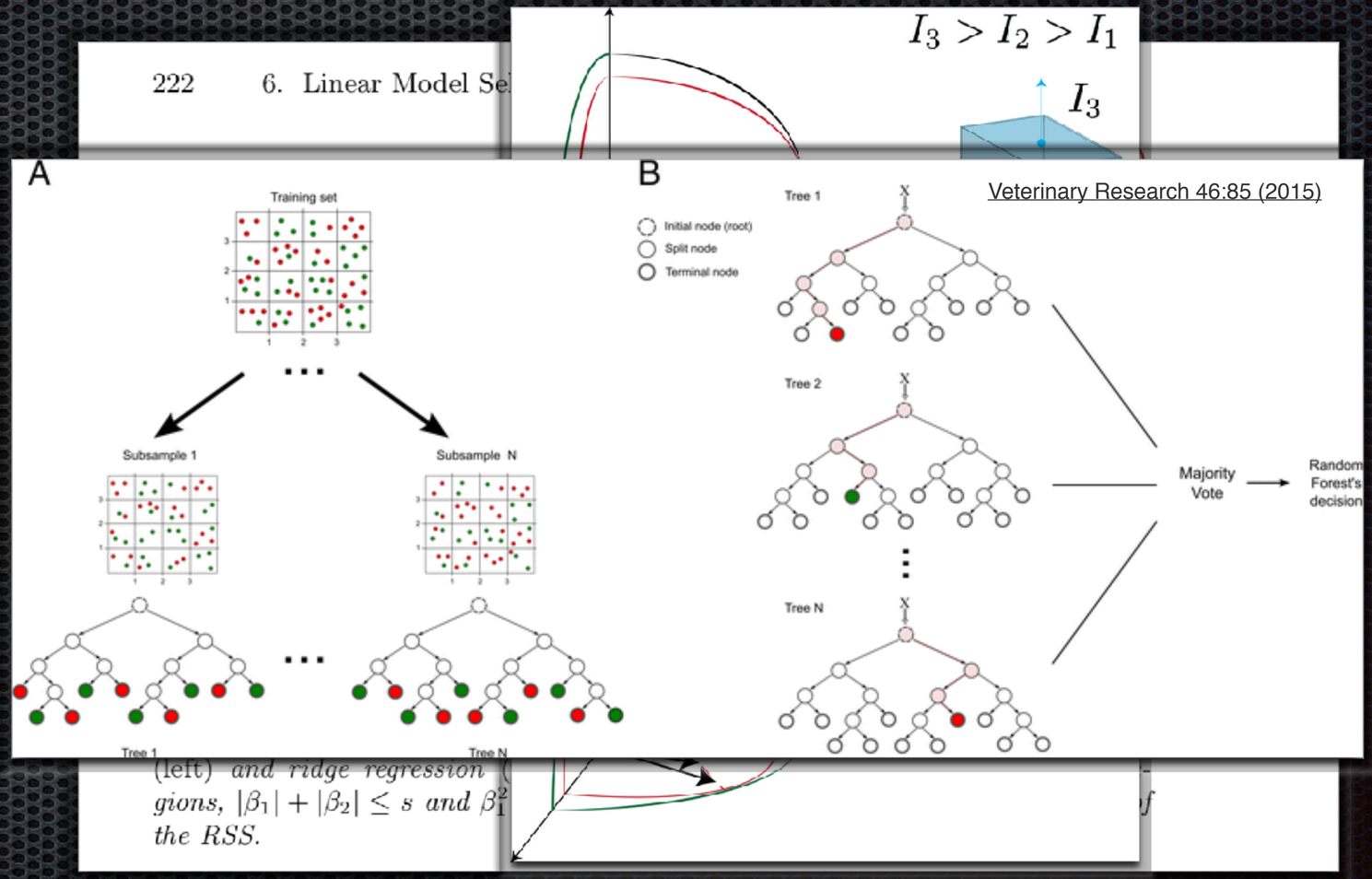
- Regularized Regression
  - Ridge
  - Lasso
- PCA - MLR, XGBoost
- Random Forest



Veterinary Research 46:85 (2015)

Models Investigated

- Regularized Regression
  - Ridge
  - Lasso
- PCA - MLR, XGBoost
- Random Forest
- XGBoost



Journey Through The Woods

- Decision Tree
- Tuned Decision Tree
- Bagged Tree
- Random Forest
- Tuned Random Forest
- Stochastic Gradient Boost

Decision Tree vs. Tuned Decision Tree

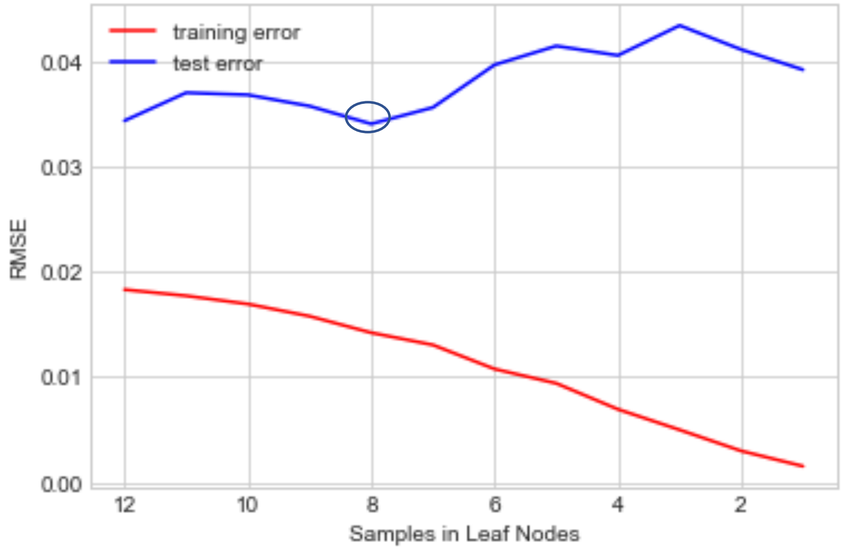
Decision Tree

- **Parameters:**
  - EVERYTHING defaulted
- **R^2**
  - Training: 99%
  - Test: 74%
- **RMSE**
  - Training: ~ 0
  - Test: .2195

Tuned Decision Tree

- **Best Parameters**
  - Max Depth: 11
  - Min Samples Leaf: 8
- **R^2**
  - Training: 91%
  - Test: 77%
- **RMSE**
  - Training: .1146
  - Test: .1861

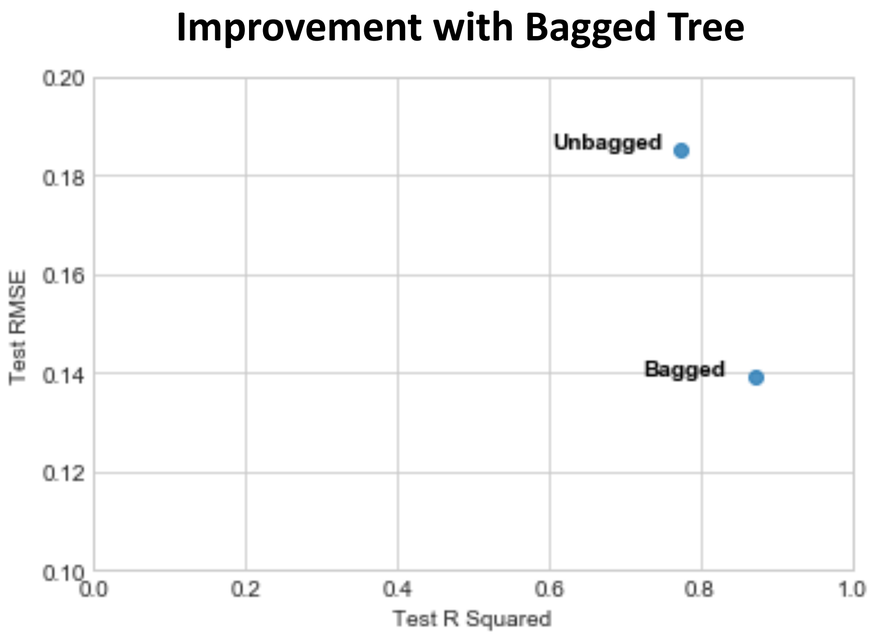
Tuning Samples in Leaf Nodes with GridSearch



THOUGHTS: We have only fit the tree on one set of observations. Lower the variance by fitting the tree on MULTIPLE sets of observations using a Bagged Decision Tree.

Bagged Tree Continues To Improve Score

- **Parameters**
  - Trees: 500
  - Max Samples: 783
    - (2/3rds of training set)
- **R^2**
  - Training: 97%
  - OOB: 87%
  - Test: 88%
- **RMSE**
  - Training: .0745
  - Test: .1370



THOUGHTS: Training and Test scores were improved but trees are too correlated as we are using the same set of parameters. Take subset of predictors at each split to decorrelate the trees using Random Forest.

Bagged Tree vs. Random Forest. No Change?!

Bagged Tree

- **Parameters**
  - Trees: 500
  - Max Samples: 783 (2/3rs of training)
- **R^2**
  - Training: 97%
  - OOB: 87%
  - Test: 88%
- **RMSE**
  - Training: .0745
  - Test: .1370

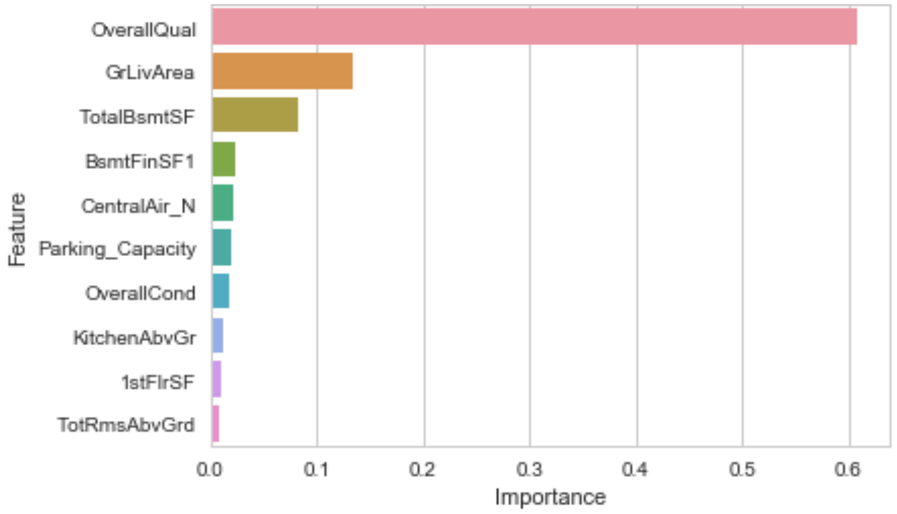
Random Forest

- **Parameters**
  - Trees: 500
  - Max features: 17 (*sqrt(p)*)
- **R^2**
  - Training: 98%
  - OOB: 86%
  - Test: 88%
- **RMSE**
  - Training: .0538
  - Test: .1339

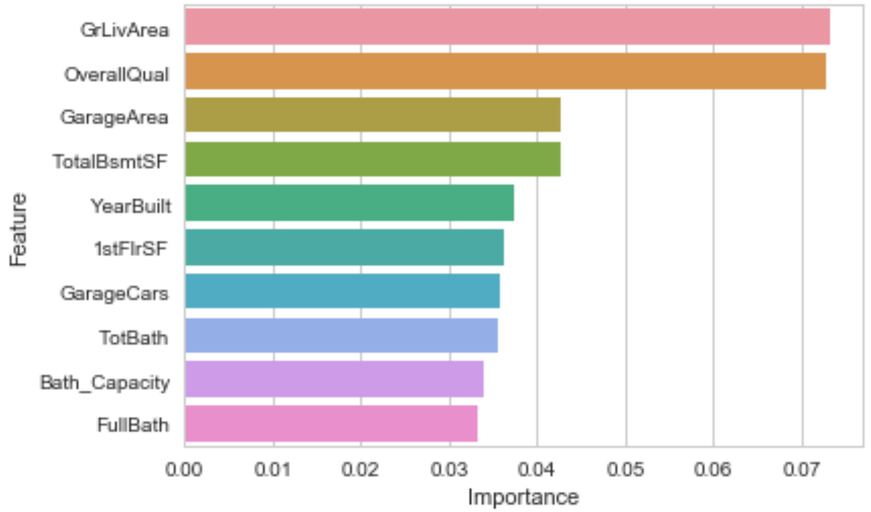
THOUGHTS: Things barely changed (.003 improvement in Test RMSE) in the results between the Bagged Tree and the Random Forest model. But what DID change were the importance to each predictor.

Top 10 Variable Importance Comparison

Correlated Trees (Bagged Tree)



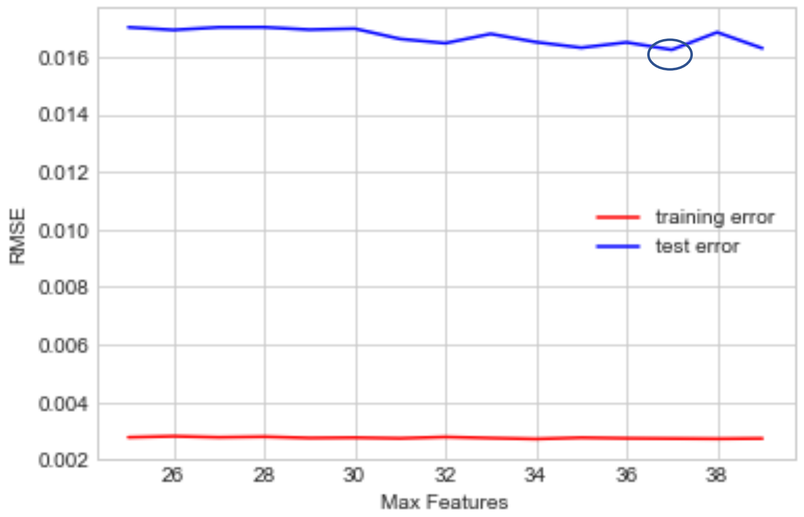
Non Correlated Trees (Random Forest)



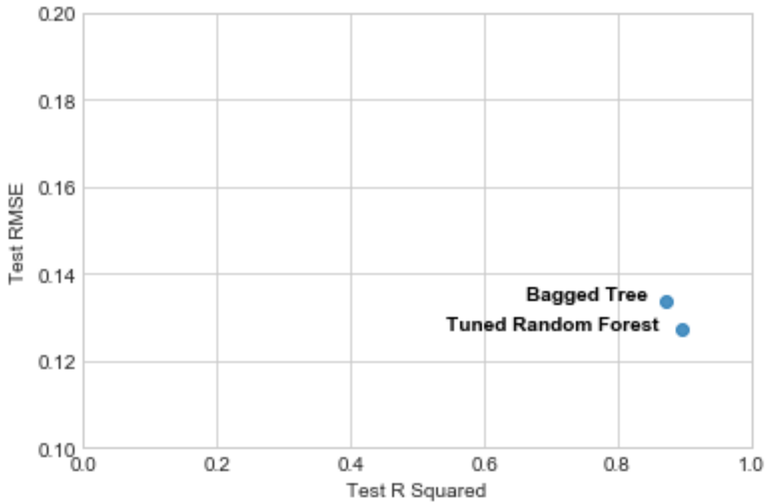
Tuned Random Forest – No Huge Improvements

- **Best Parameters**
  - Trees: 500
  - Max Features: 37
- **R^2**
  - Training: 98%
  - OOB: 87%
  - Test: 89%
- **RMSE**
  - Training: .0526
  - Test: .1274

Tuning Max Features with GridSearch



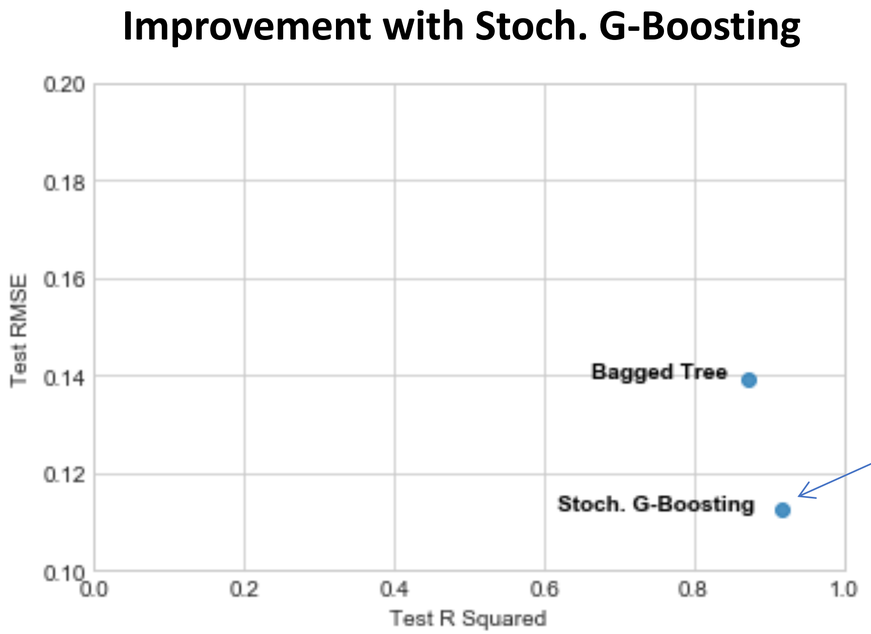
Improvement with Tuned Random Forest



THOUGHTS: Again, not much changed for the Training and Test R^2 . A .003 improvement in Test RMSE. What's going on?

Stochastic Gradient Boosting Shows Considerable Improvement To Score

- Parameters
  - Learning Rate: 0.1
  - Subsample: 2/3
  - Depth: 3
- R^2
  - Training: 96%
  - Test: 92%
- RMSE
  - Training: .0835
  - Test: .1125

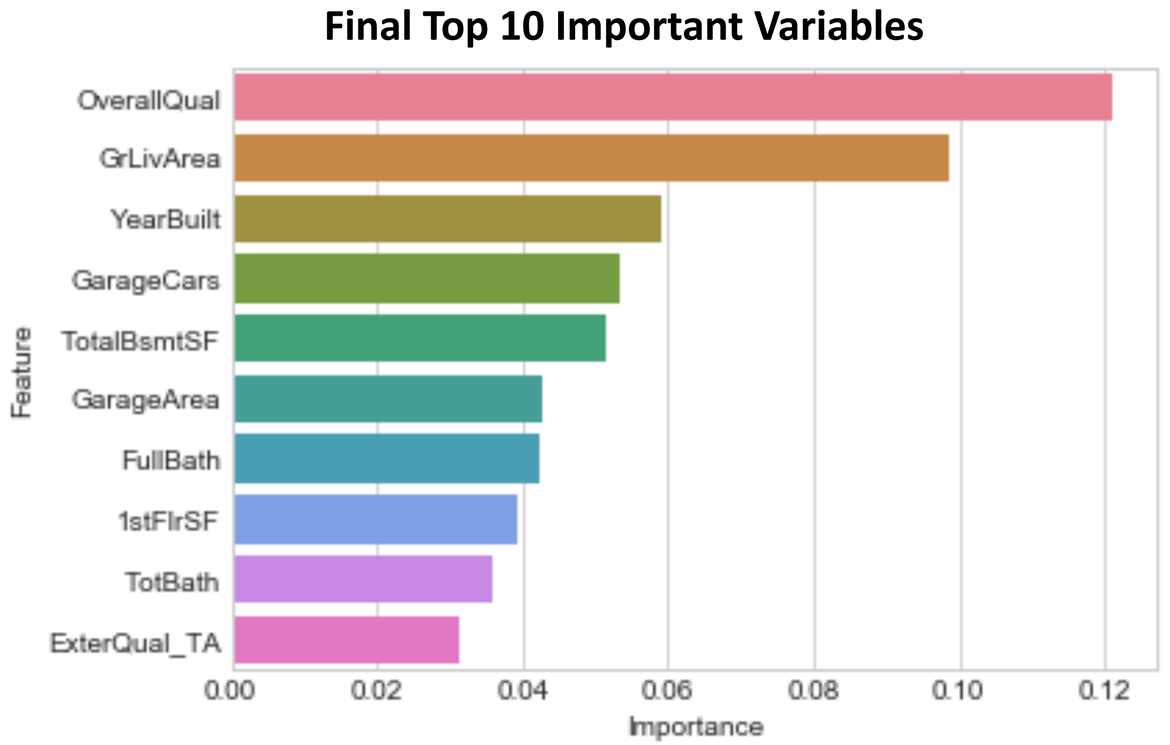


Test R^2 : 88% -> 92%

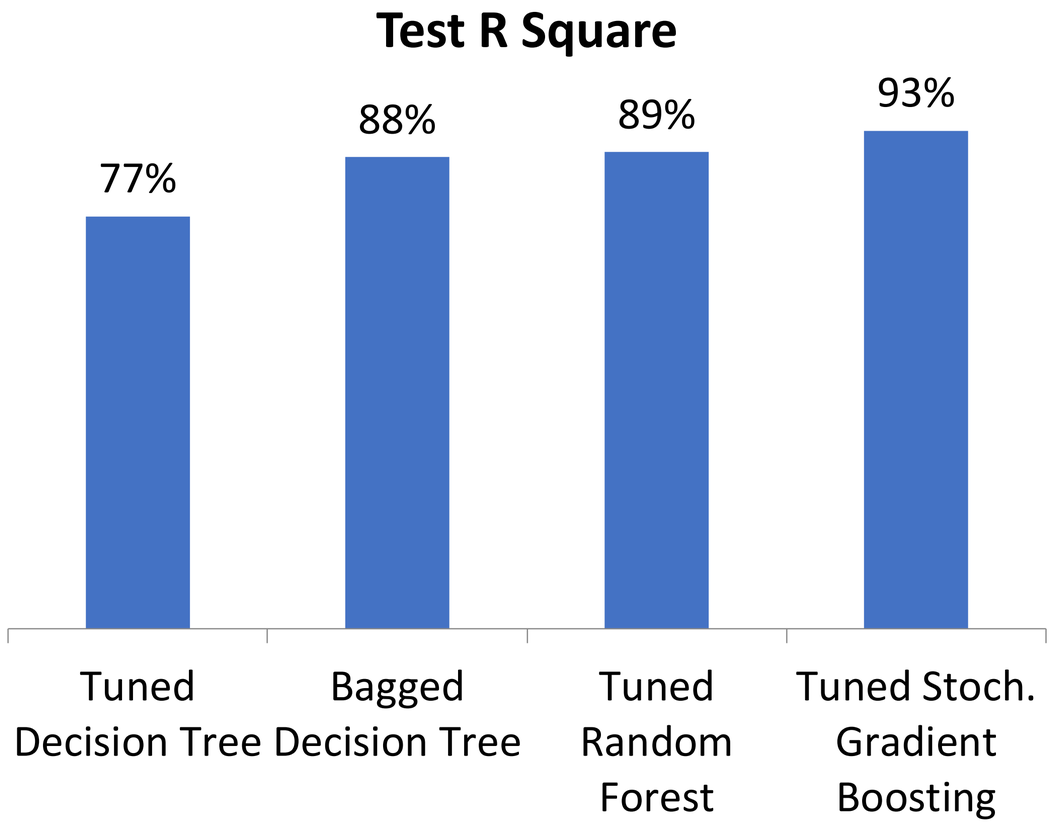
Test RMSE : .1361 -> .1125

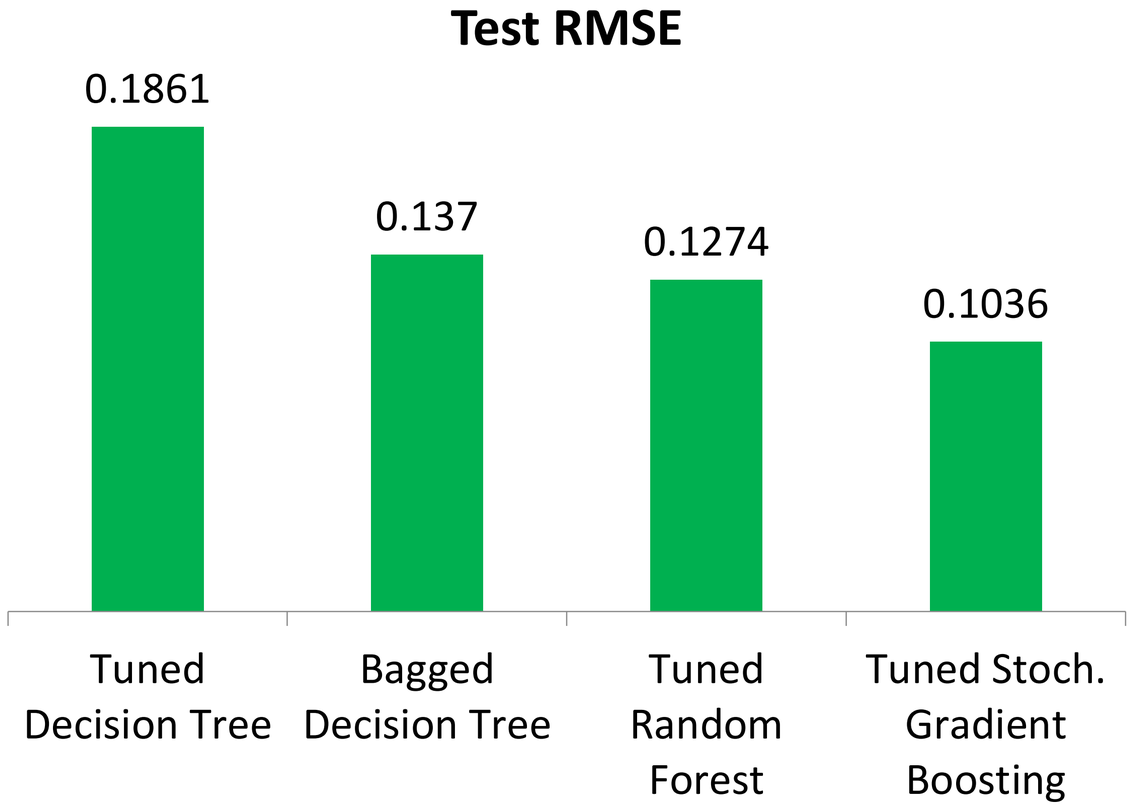
Tuned Stochastic Gradient Boosting (using GridSearchCV)

- **Best Parameters**
  - Trees: 650
  - Learning Rate: 0.05
  - Subsample = 2/3
- **R^2**
  - Training: 99%
  - Test: 93%
- **RMSE**
  - Training: .0454
  - Test: .1036

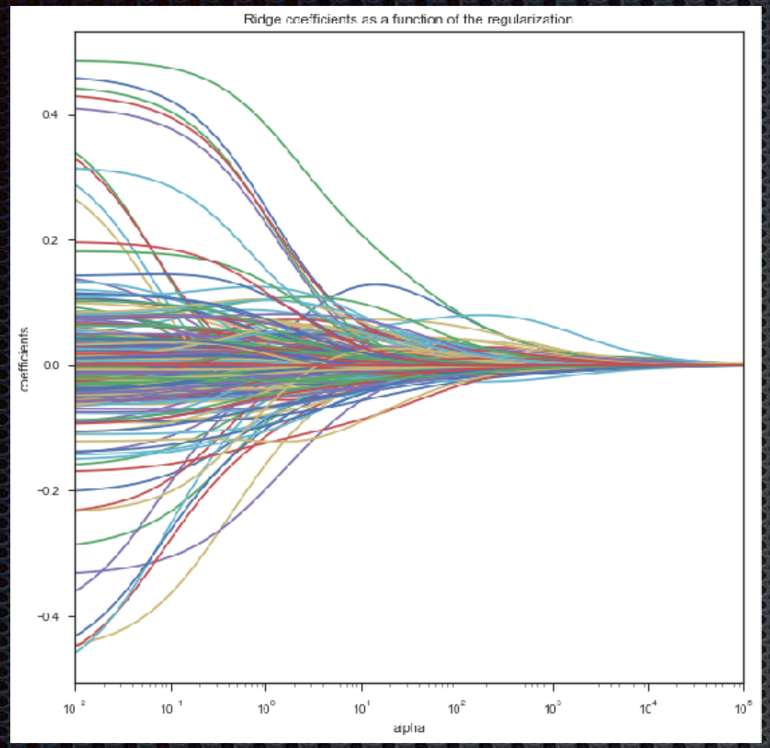


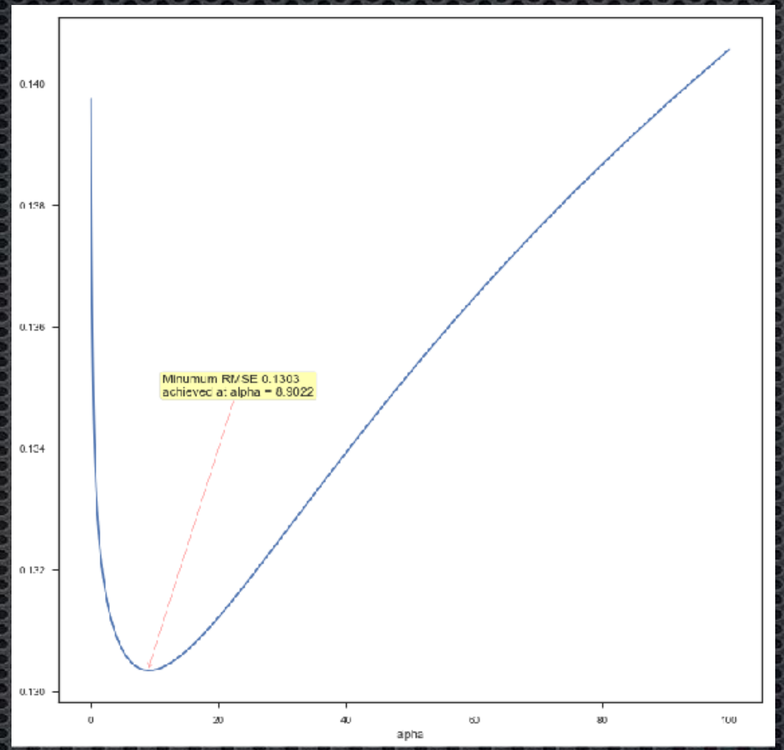
Tree Based Model Comparison

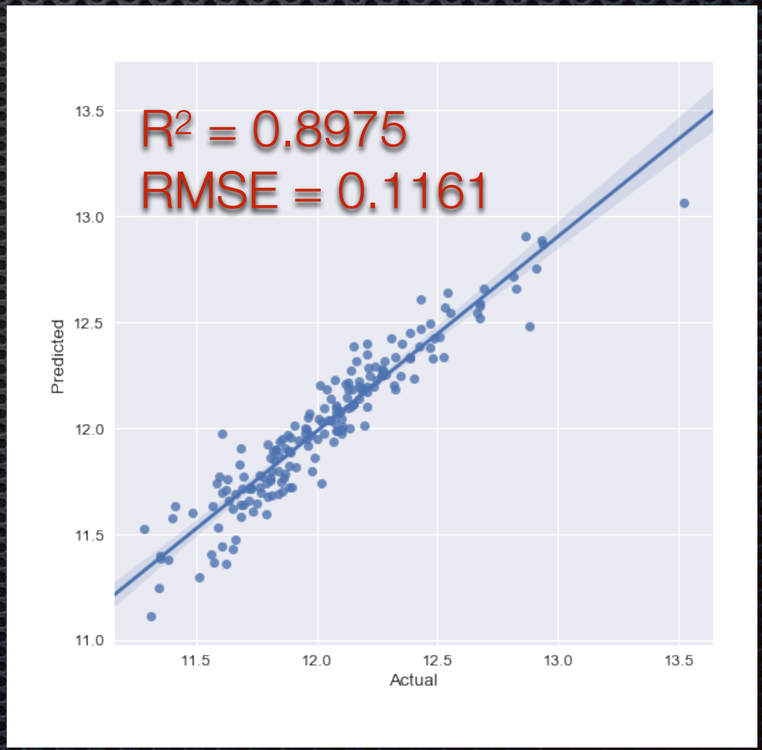




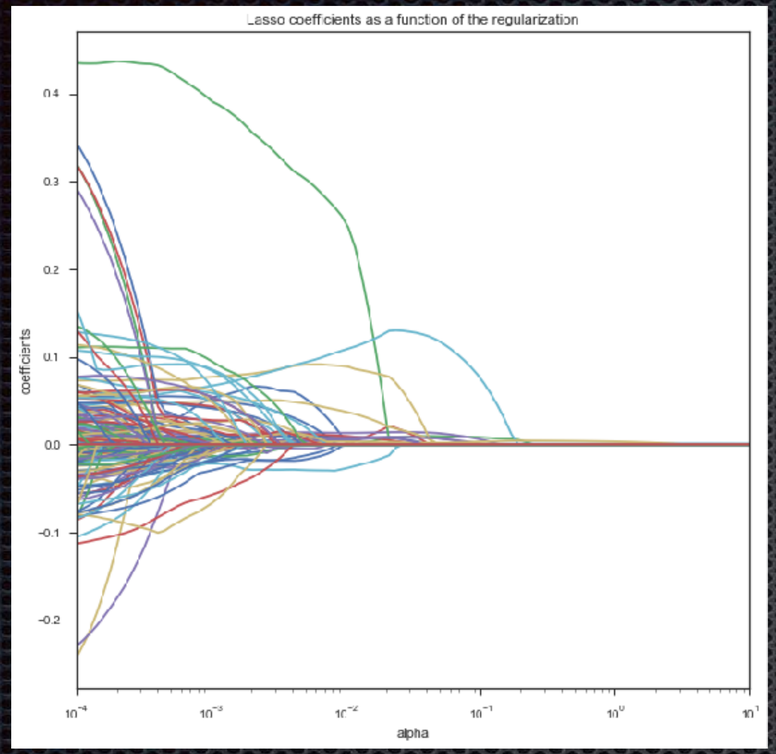
Regularized Regression - Ridge

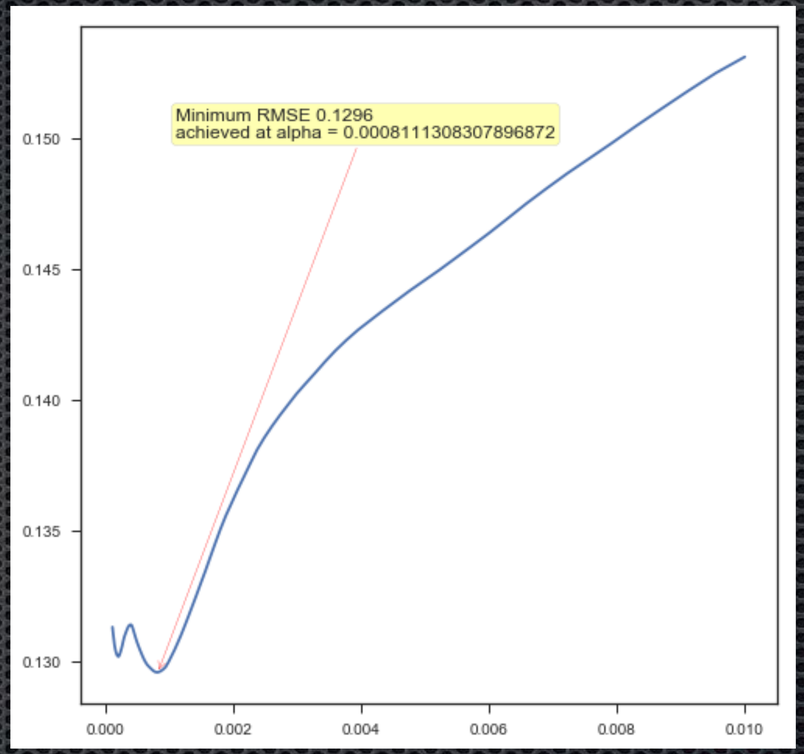




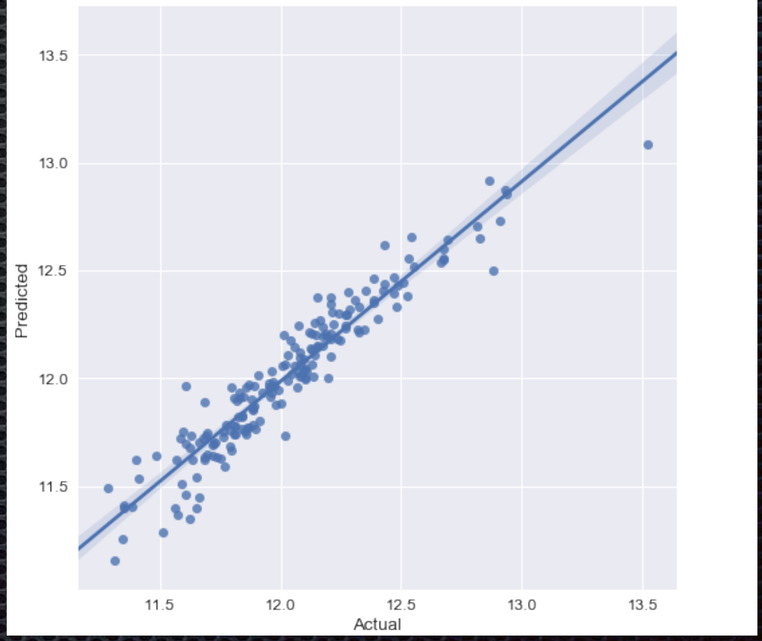


Regularized Regression - Lasso

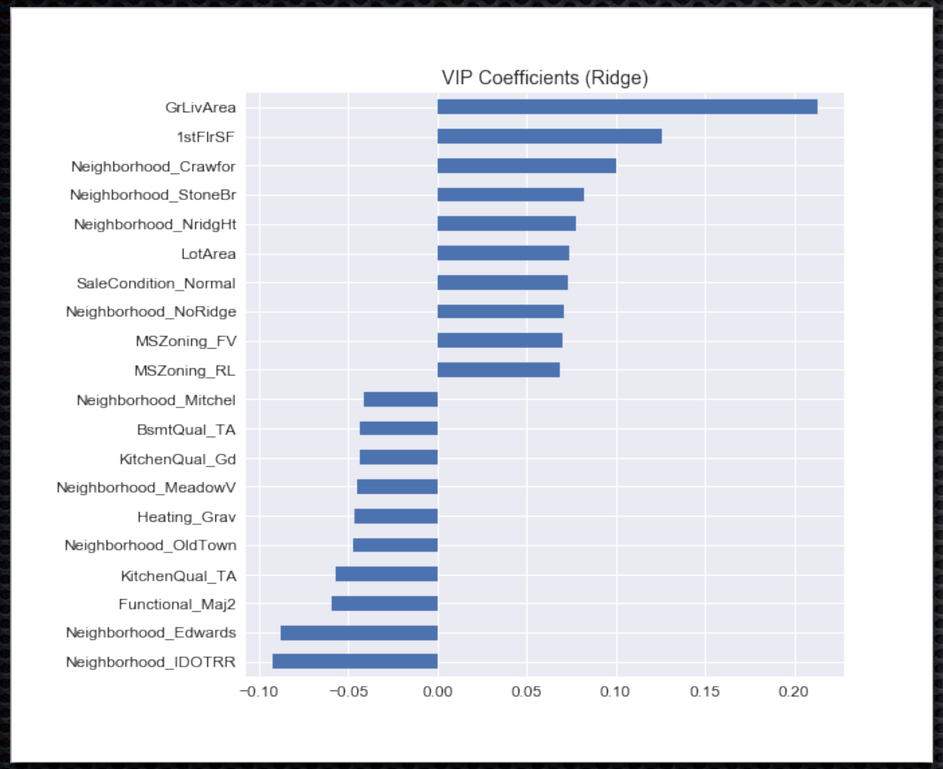


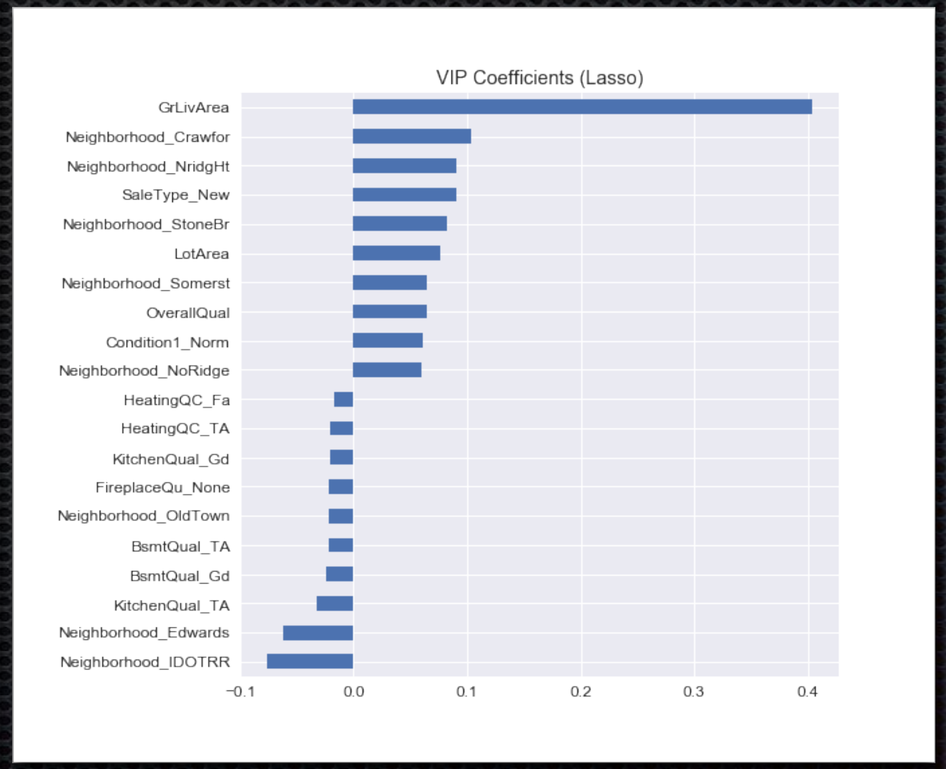


$R^2 = 0.9058$, RMSE = 0.1112



‘VIP’ Coefficients for Ridge and Lasso





XGBoost

- ???!!!$%^%$&% - many parameters to tune!
- Grid Search CV to the rescue (n_estimators, learning rate)
- Control overfitting with:
  - Gamma, max_depth, min_child_weight (complexity)
  - Subsample, col_sample_bytree (less sensitive to random fluctuations)

```
xgboost.XGBRegressor(
max_depth=3,
learning_rate=0.1,
n_estimators=100,
silent=True,
objective='reg:linear',
booster='gbtree',
n_jobs=1,
nthread=None,
gamma=0,
min_child_weight=1,
max_delta_step=0,
subsample=1,
colsample_bytree=1,
colsample_bylevel=1,
reg_alpha=0,
reg_lambda=1,
scale_pos_weight=1,
base_score=0.5,
random_state=0,
seed=None,
missing=None)
```

Putting it all together

- Ensemble Averaging?
- Stacking?