Zillow Competition – Round One

@ Kaggle

Last Entry Allowed: Oct 16th, 2017

Scoring: Ended Jan 10th, 2018

Outline

- Competition Description
  - Overview
  - Scoring
  - Incentives
  - Data
- Feature Engineering
  - Basic work
  - Unusual methods
    - Neighbourhood
    - Constrained Similarity measures
  - Weak & discarded features
- Stacking
- Add-On Features
- Other Teams
- Thanks!

Overview

Scoring

Incentives

Data

COMPETITION DESCRIPTION

Overview

- Zillow has a proprietary model they use to predict home sales. The objective of the competition is to predict the log errors of their model.

  $$\text{logerror} = \log(\text{Zestimate}) - \log(\text{SalePrice})$$

  - Scoring is based on MAE.

- Economic rationale is important, but we might get counter-intuitive results.

Scoring

- Private scoring occurs using only data after the competition closes (Oct 16, 2017)
- Private scores were updated three times:
  - Nov 17
  - Dec 18
  - Jan 10 (final)

Incentives

- $50,000 in prizes for round one
- $1,150,000 in prizes for round two ($1M for first place)

- Round two:
  - Round one is a qualifier – top 100 teams only
    - Must submit code from round one
  - Much more data.
  - Instead of predicting residuals, you aim to have greater accuracy in predicting home sale prices
    - No prizes if you don’t beat Zillow’s model

- (Hopefully Zillow won’t use round one submissions to improve their model)
  - Edit: Zillow is now offering substantial prizes in the event teams do not beat the benchmark model!

Data

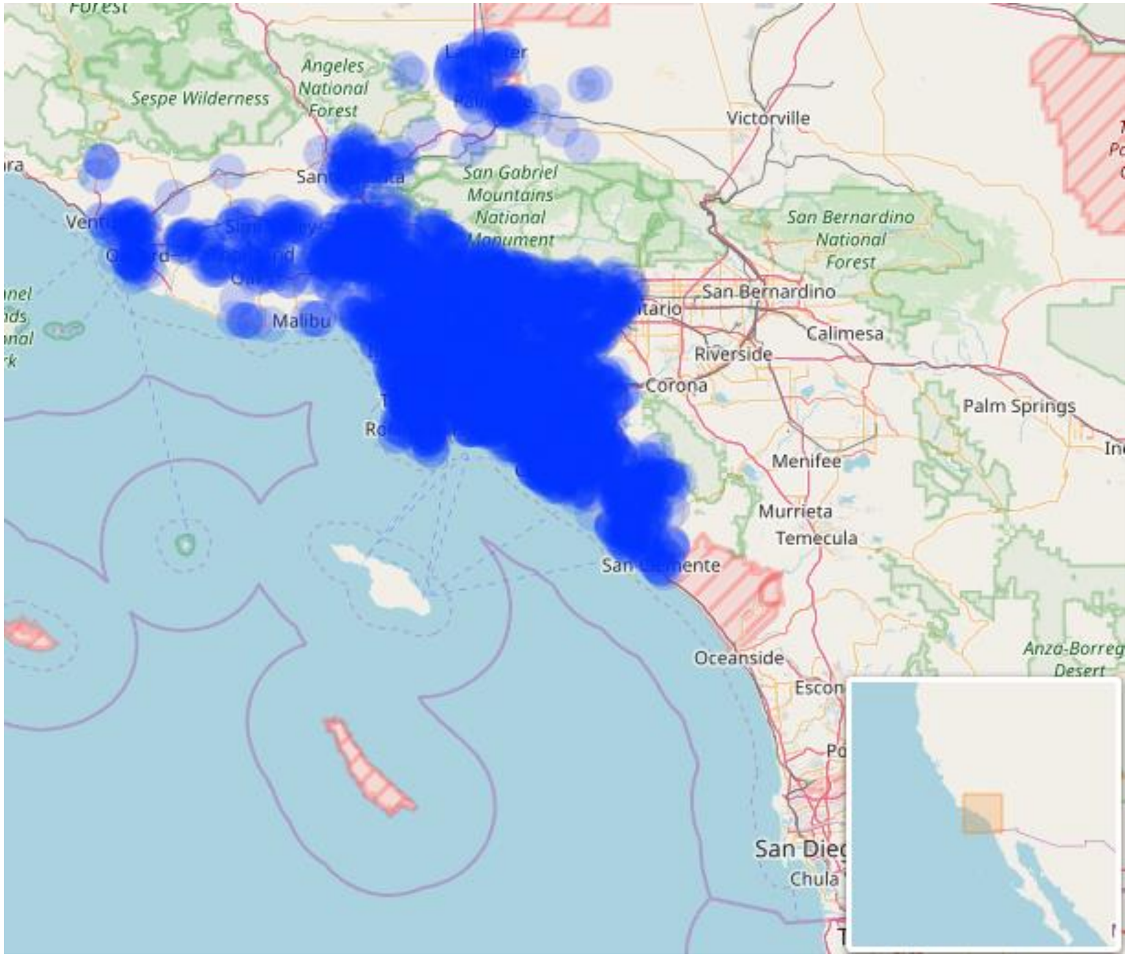
Roughly 150,000 observations with a valid Y value

Roughly 3,000,000 observations without a Y value

- Y Variable: Logerrors
- 58 X variables
  - substantial redundancy

Data Type	Fields
Home Quality	Build date, building quality, type of heating, type of air conditioning,
Property & Home Size	Square feet, number of rooms (by type)
Location	Latitude/longitude
Other	Garage, deck, pool, building type, number of units, taxes paid, assessment values

Map of Transactions

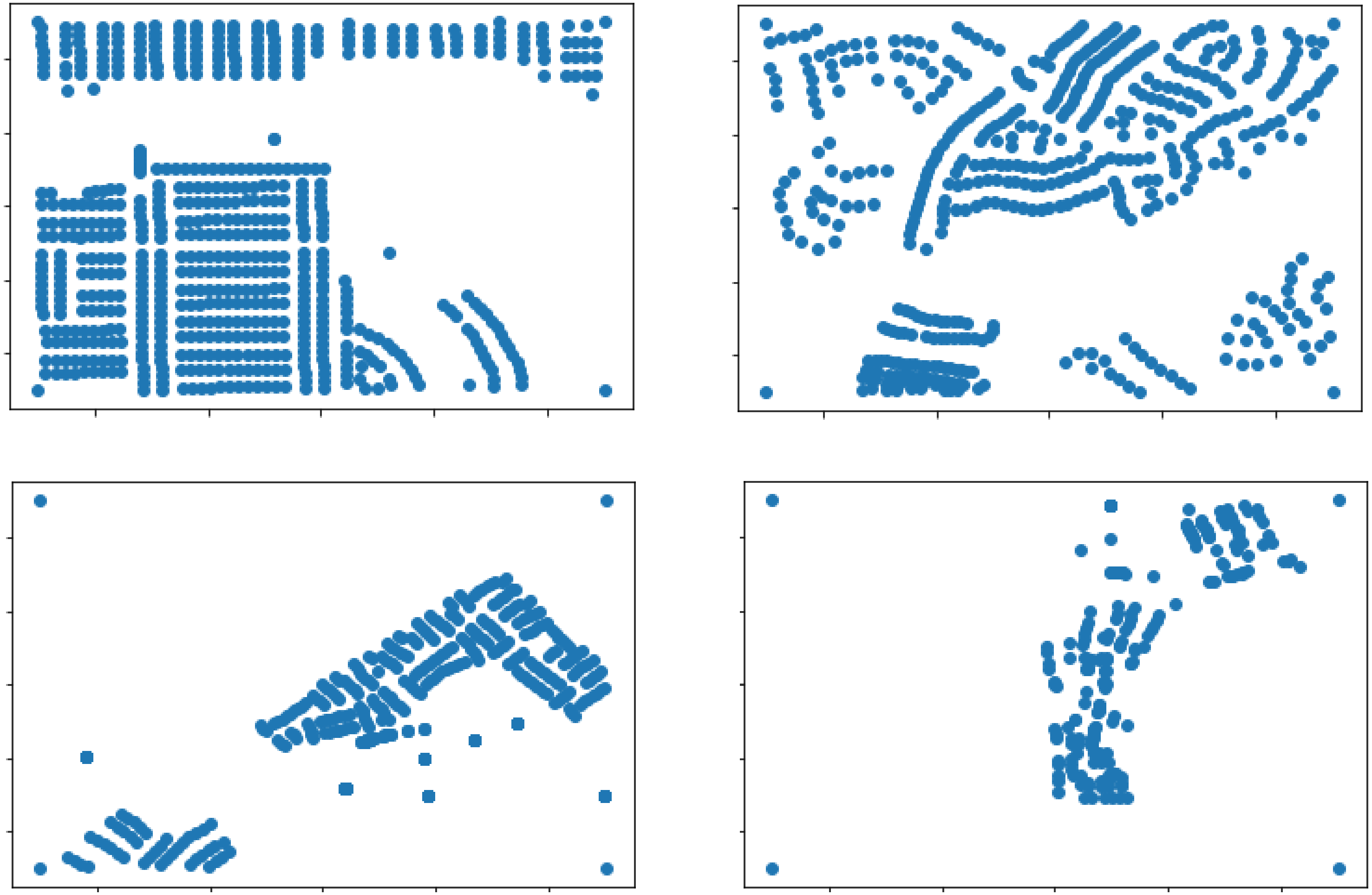


Philipp Spachtholz

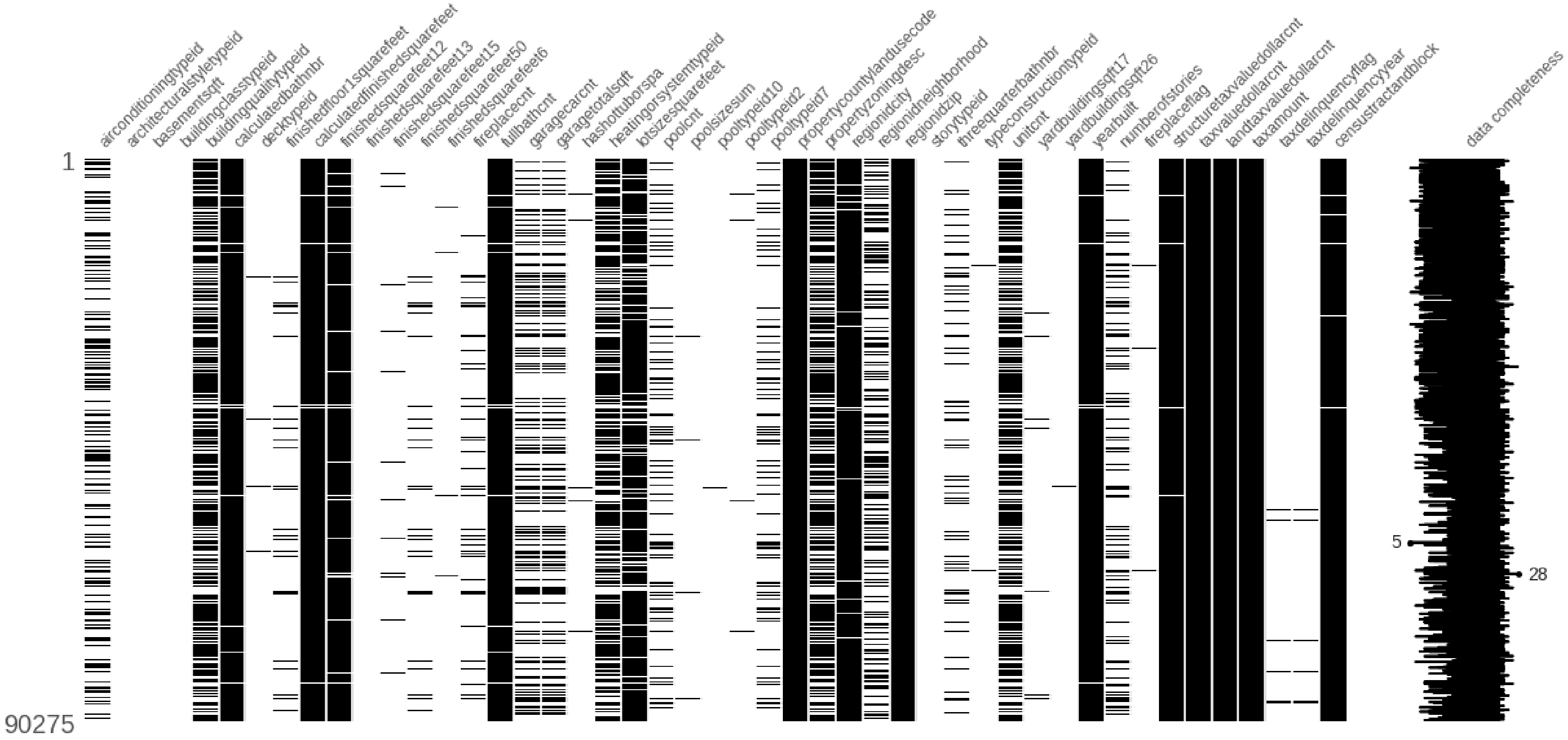
https://www.kaggle.com/philippsp

https://www.kaggle.com/philippsp/exploratory-analysis-zillow

Detailed Neighbourhood Maps



Feature Coverage
[White = Missing]

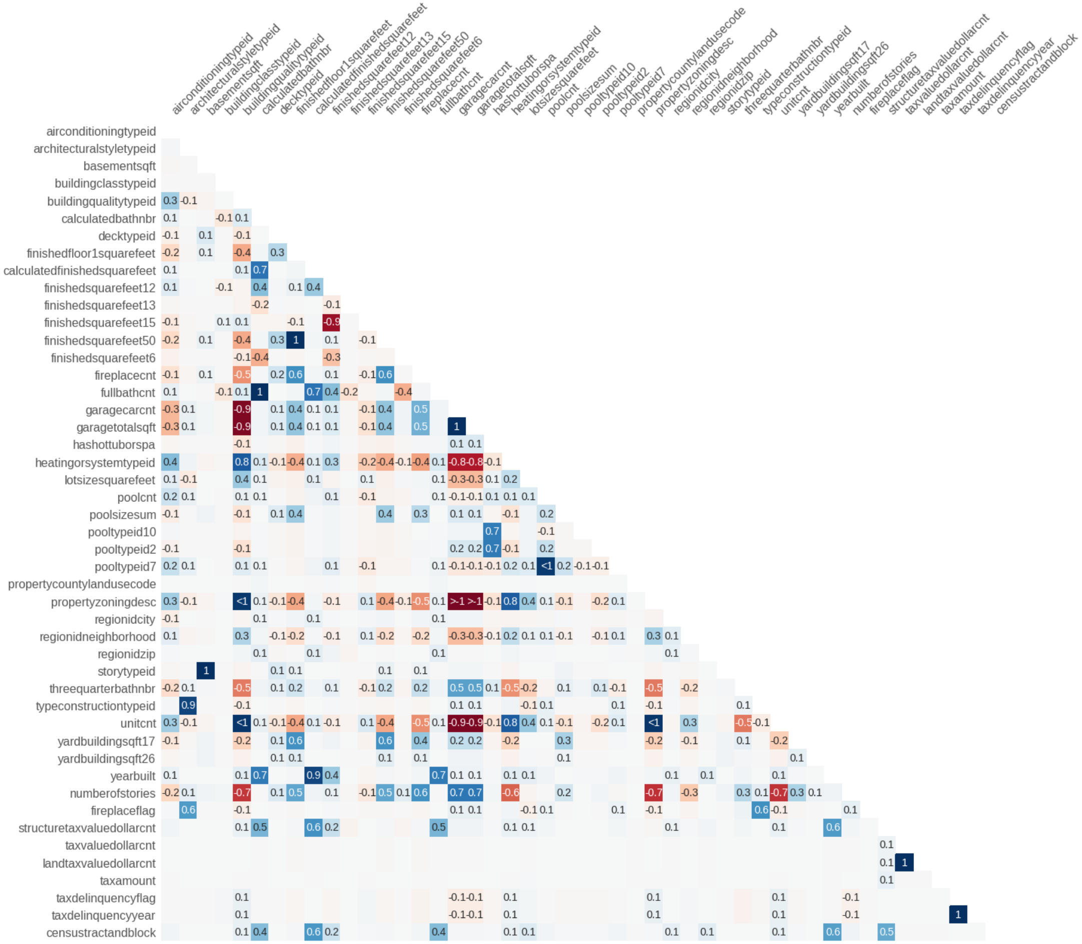


Vivek Srinivasan

https://www.kaggle.com/viveksrinivasan

https://www.kaggle.com/viveksrinivasan/zillow-eda-on-missing-values-multicollinearity

Feature Correlation

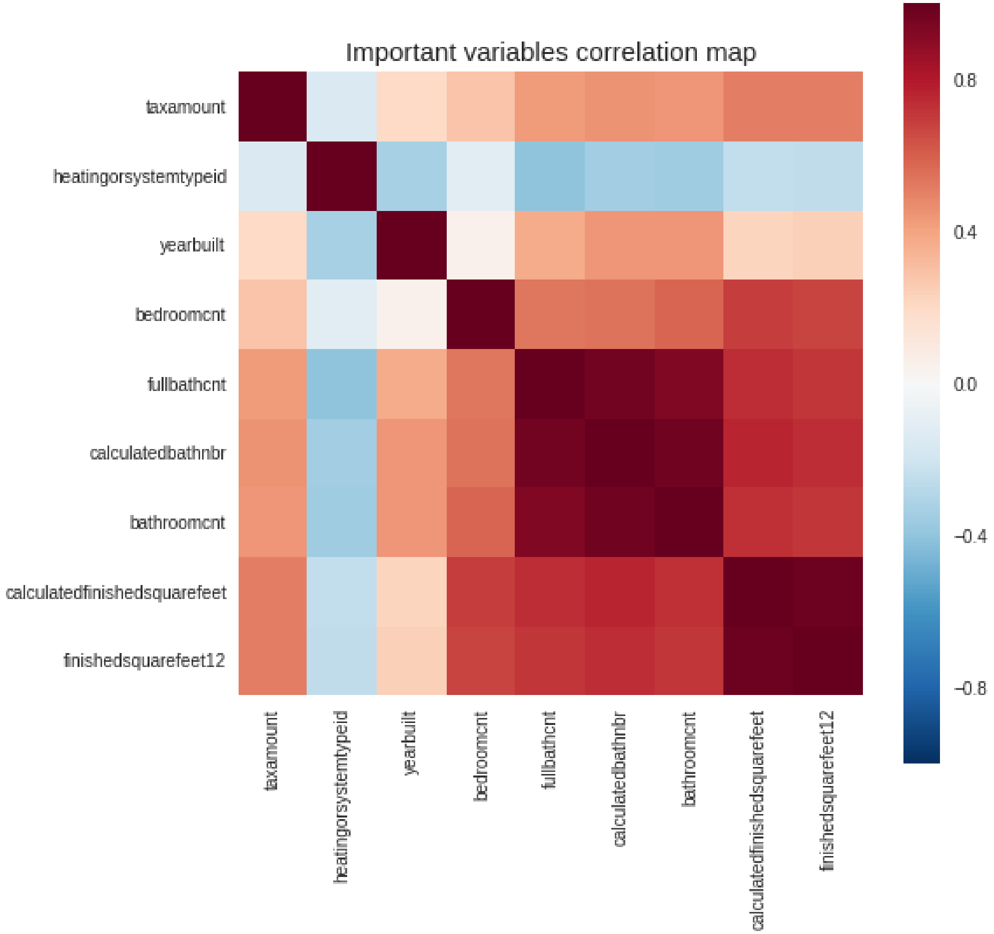


Vivek Srinivasan

https://www.kaggle.com/viveksrinivasan

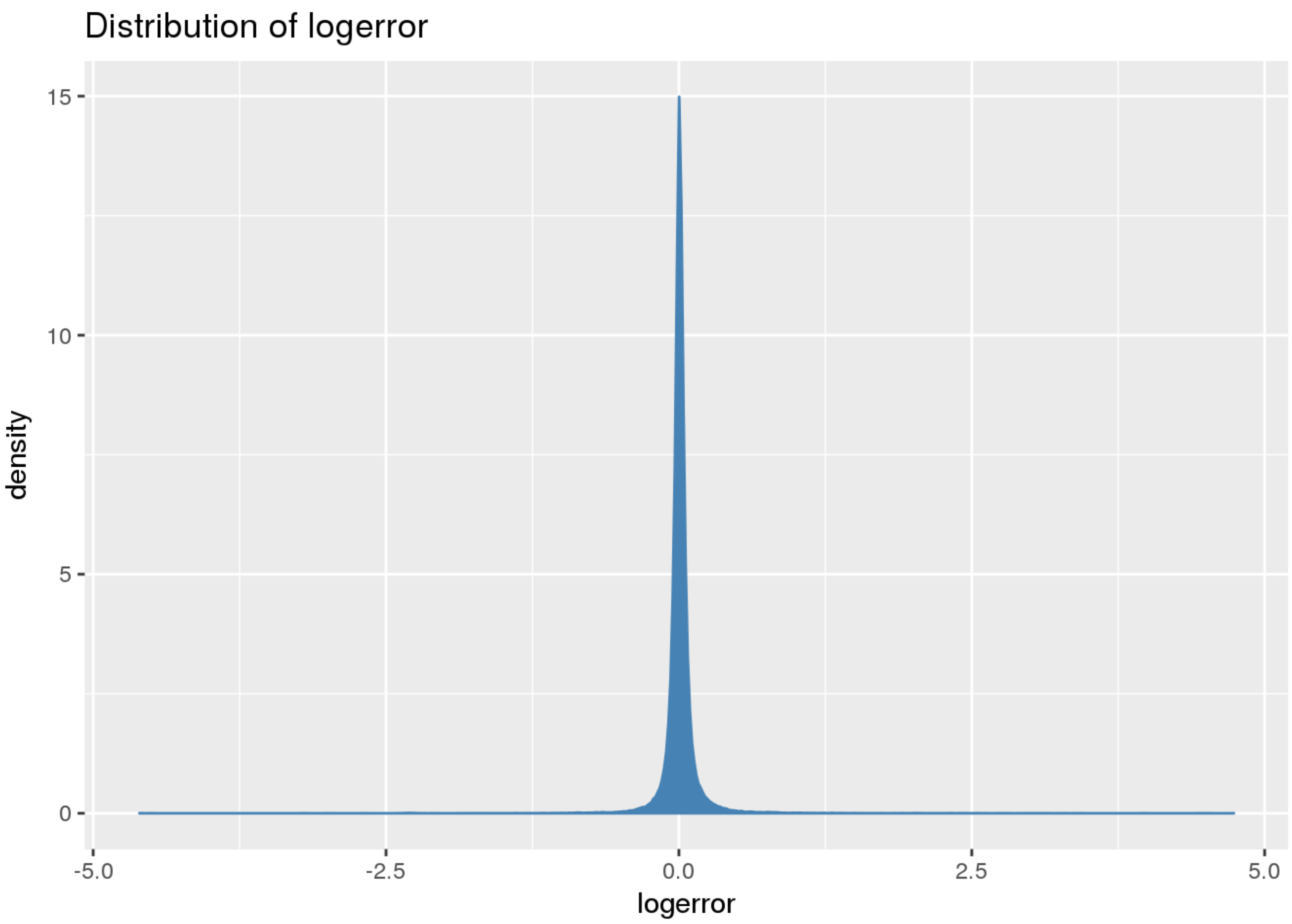
https://www.kaggle.com/viveksrinivasan/zillow-eda-on-missing-values-multicollinearity

Correlation of Most Important Features



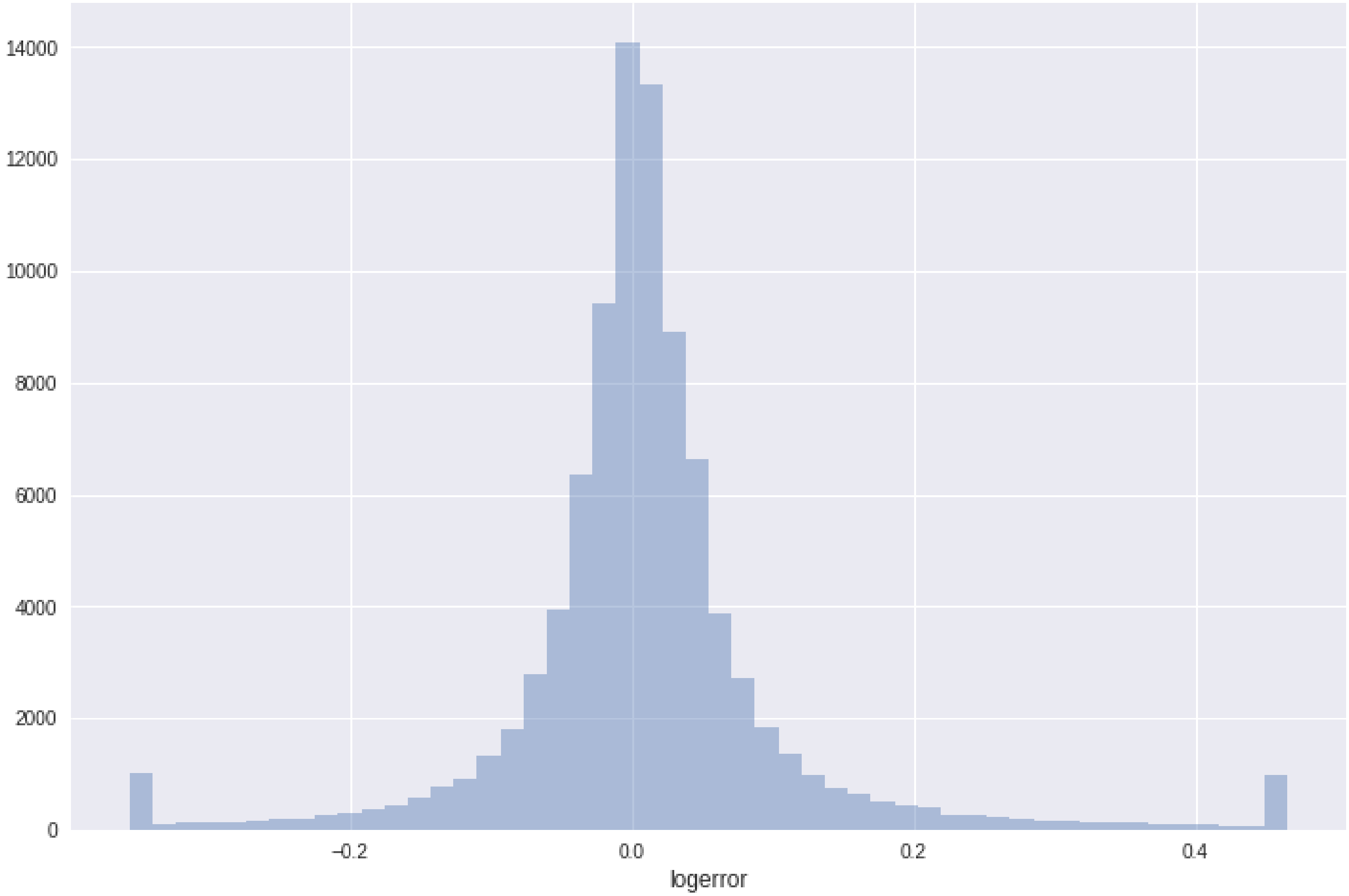
Sudalai Rajkumar
https://www.kaggle.com/sudalairajkumar
https://www.kaggle.com/sudalairajkumar/simple-exploration-notebook-zillow-prize

Distribution of Y Variable



Troy Walters
https://www.kaggle.com/captcalculator
https://www.kaggle.com/captcalculator/a-very-extensive-zillow-exploratory-analysis

Distribution of Y Variable

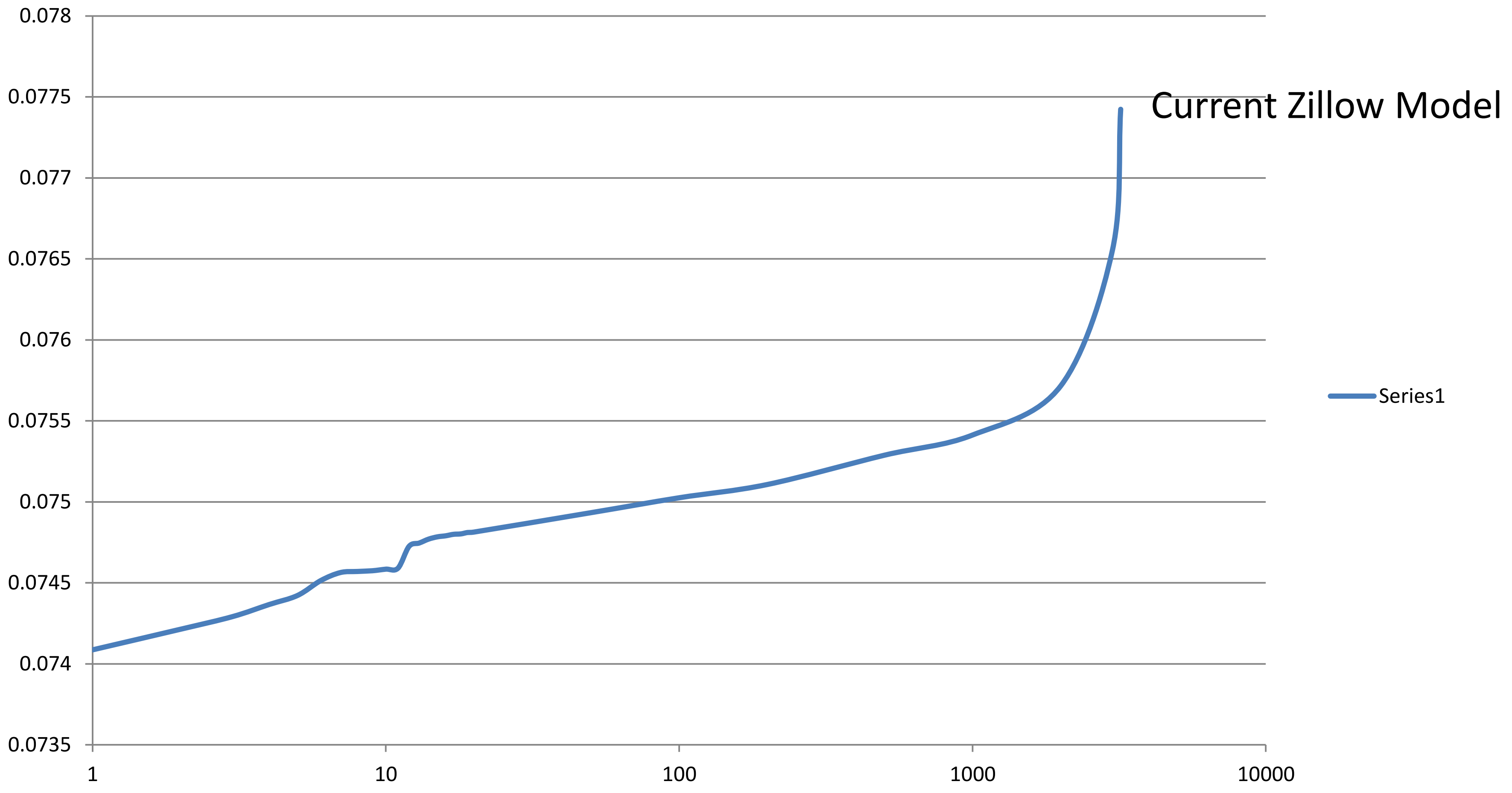


Sudalai Rajkumar
https://www.kaggle.com/sudalairajkumar
https://www.kaggle.com/sudalairajkumar/simple-exploration-notebook-zillow-prize

Solution Explored

- Rankings: (/3800)
  - Public #17
  - Private first month: #11
  - Private second month: #19
  - Private Final: #17
- Very limited disclosures so far – essentially all highly-ranked participants are withholding their strategies so they can be re-used in stage two.

MAE Private Score by Rank



Basic work

Unusual methods

Weak & discarded features

FEATURE DESIGN

Basic Feature Engineering

- New features:
  - Structure value per square foot
  - Average room size
  - Value of structure relative to land
  - & others

- Categorical variables treatment: use some intuition
  - If they have a natural order -> integers
    - For example: Air conditioning type – generate integer values based on a basic assessment of quality
  - If they are similar -> group

  (particularly if there are few observations)
    - For example: quadruplex, townhome

Neighbourhood (1/6)

Intuition:

- Location, location, location

➢ Use information other participants might overlook – the set of houses for which there is no corresponding Y-variable.

We used two methods to extract information from these houses:

1. Average Neighbour:
   - ➢ Average feature values for nearby homes
   - ➢ Average difference in feature values vs nearby homes
2. Each Neighbour:
   - ➢ This is a lot of variables. We will have to constrain the relationships.

Neighbourhood (2/6)

- First, model the log errors using information about the home. Using a regression as a representation of the problem, we have:

$$1)Y = \beta_1 X_1 + \cdots + \beta_n X_n + \varepsilon_1$$

Where:

– *X1* through *Xn* are features of the houses whose log errors we are modeling

- We think the characteristics of the neighbours also matter. If we were to include the features of the nearest house we could have:

$$2)Y = \beta_1 X_1 + \cdots + \beta_n X_n + \beta_{N1} X_{N1} + \cdots + \beta_{Nn} X_{Nn} + \varepsilon_2$$

Where:

– *XN1* through *XNn* are features of the neighbouring house

Neighbourhood (3/6)

- Otherwise we could obtain a slightly worse but similar solution to (2) by modeling the residuals of (1) as in:

$$3)\varepsilon_1 = \beta_1 X_1 + \cdots + \beta_n X_n + \beta_{N1} X_{N1} + \cdots + \beta_{Nn} X_{Nn} + \varepsilon_3$$

Neighbourhood (4/6)

- Although (1) and (3) are less efficient than (2), we can more think of the information in (3) as being incremental over equation (1), whereas in (2) the coefficients are interdependent. This simplicity is going to be useful as we will now go deeper…
  - We want to add more than the nearest home. We want to include information about **all** nearby homes.

Neighbourhood (5/6)

- So now we get:

$$4)\,\varepsilon_1 = \;\beta_1 X_1 + \cdots + \beta_n X_n + \beta_{SN1} X_{SN1} + \cdots + \beta_{SNn} X_{SNn} + \varepsilon_3$$

$$5)\,\varepsilon_1 = \;\beta_1 X_1 + \cdots + \beta_n X_n + \beta_{TN1} X_{TN1} + \cdots + \beta_{TNn} X_{TNn} + \varepsilon_3$$

- Where “SN” and “TN” refer to “second nearest” and “third nearest,” respectively.
  - We keep going until 500.

- Since we expect the coefficients of these equations to be highly related, we want to impose a little more structure…

Neighbourhood (6/6)

- We concatenate all 500 equations together, and to each one we add one more x variable – a number from 1 to 500, where the nearest gets a 1 and the furthest a 500.
  - Now for this to really work, we need to count on that last term to act as an interactor on all the other variables, so we don't use a regression, but instead a GBRT (many other options would also work).

$$6)\begin{bmatrix}\varepsilon_1\\ \cdots\\ \varepsilon_1\end{bmatrix} = \begin{bmatrix} X_1 + \cdots + X_n + X_{N1} + \cdots + X_{Nn} + 1 + \varepsilon_3 \\ \cdots \\ X_1 + \cdots + X_n + X_{500N1} + \cdots + X_{500Nn} + 500 + \varepsilon_3 \end{bmatrix}$$

- This gives us 500 estimates of each home's residual based on each of the 500 nearest neighbours.
  - Finally, we can take the 500 different estimates of each home's residual and use each as a new variable (or simply take a weighted average).

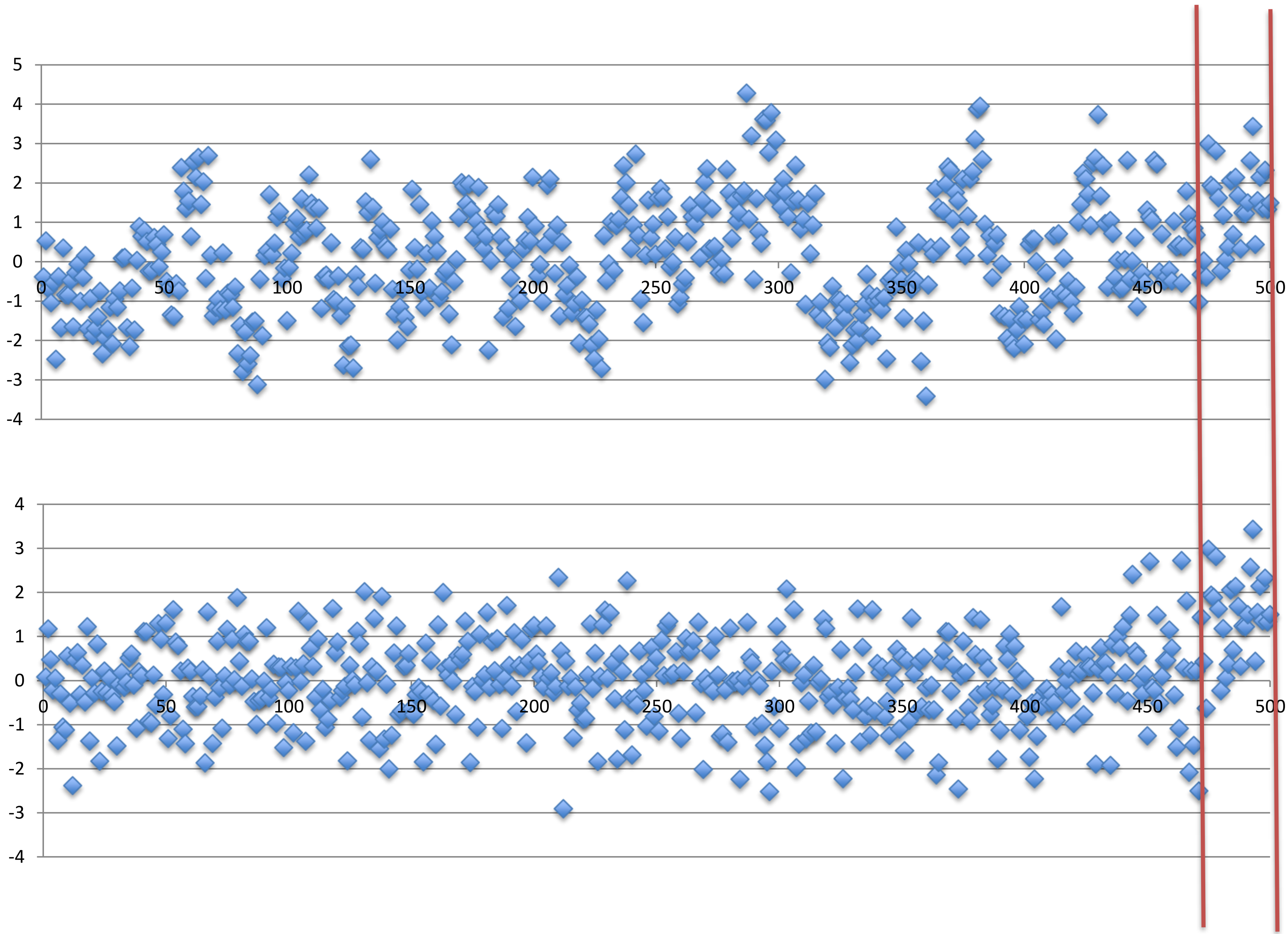
(New Topic)
Constrained Similarity Fitting

Given the limited data set, it may make sense to impose constraints on the fitting in order to obtain more robust results.

- ➢Create a preference for using features that are important throughout their full spectrum of values
  - More likely to be economically important (vs noise)
- ➢Weights determined by the whole dataset, not parts of the dataset.

Illustrating the Idea

- For which of the following two factors would you be more confident in using (as a forecast) the average value between the red lines?
  - (The numbers are identical)

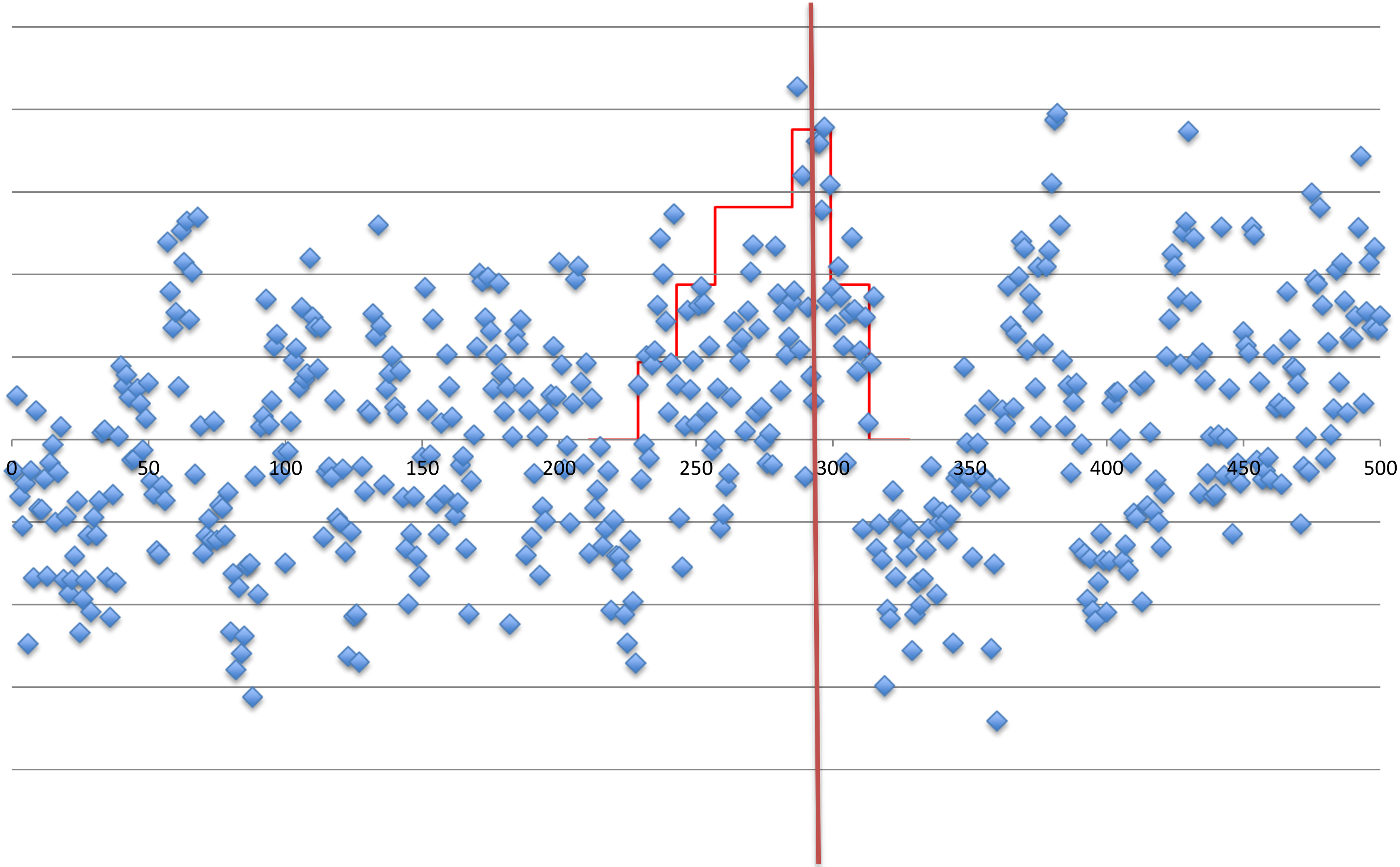

5
4
3
2
1
0
-1
-2
-3
-4
0
50
100
150
200
250
300
350
400
450
500
4
3
2
1
0
-1
-2
-3
-4
0
50
100
150
200
250
300
350
400
450
500

Argument #1

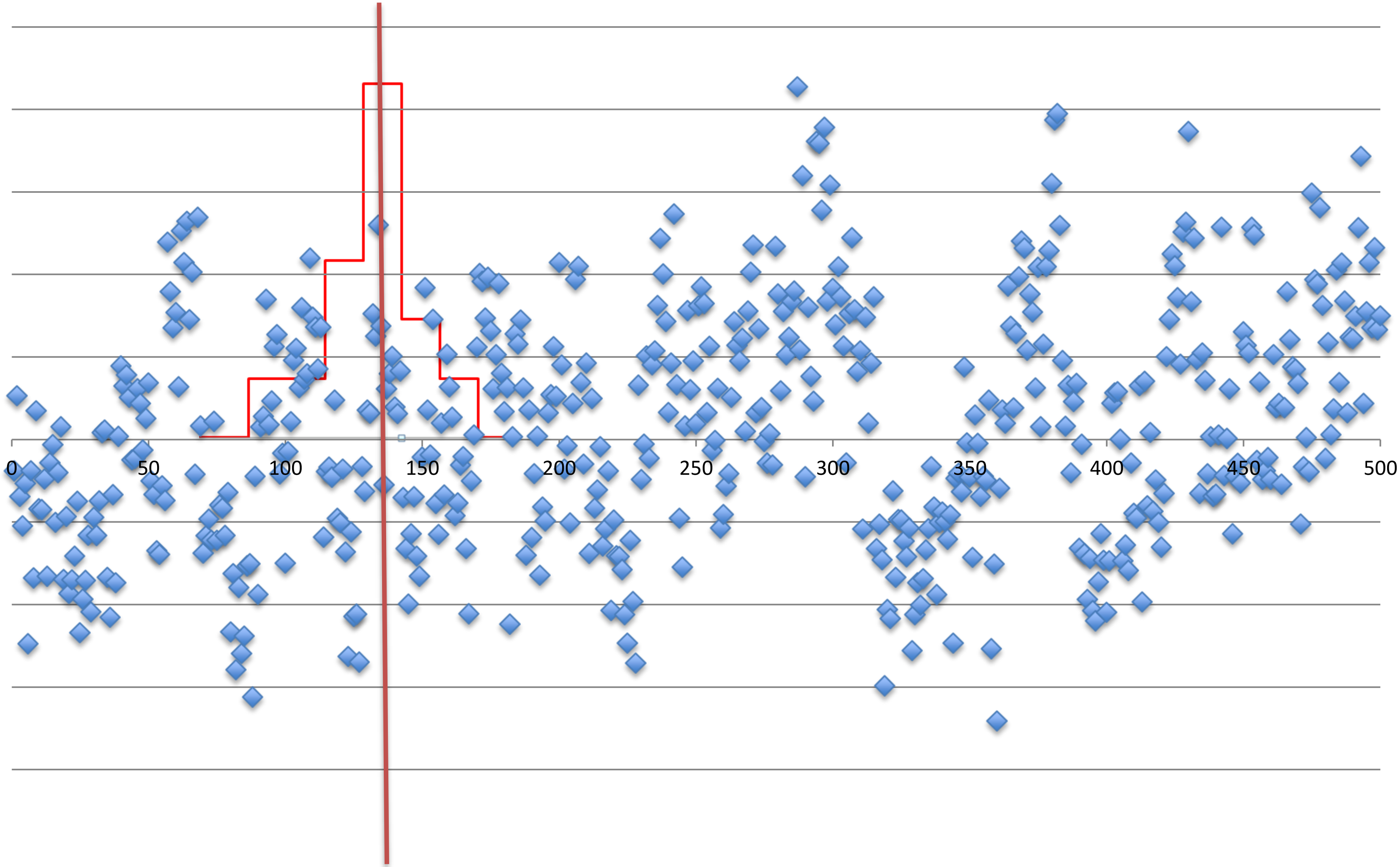
- The first feature is likely to have substantial economic importance, since it describes substantial variation across its entire spectrum.
- The second feature may have economic importance, but it is relatively likely to be noise.

- This would in general lead us to preference the split on the first factor over the second factor.

GB Decision Tree Weights



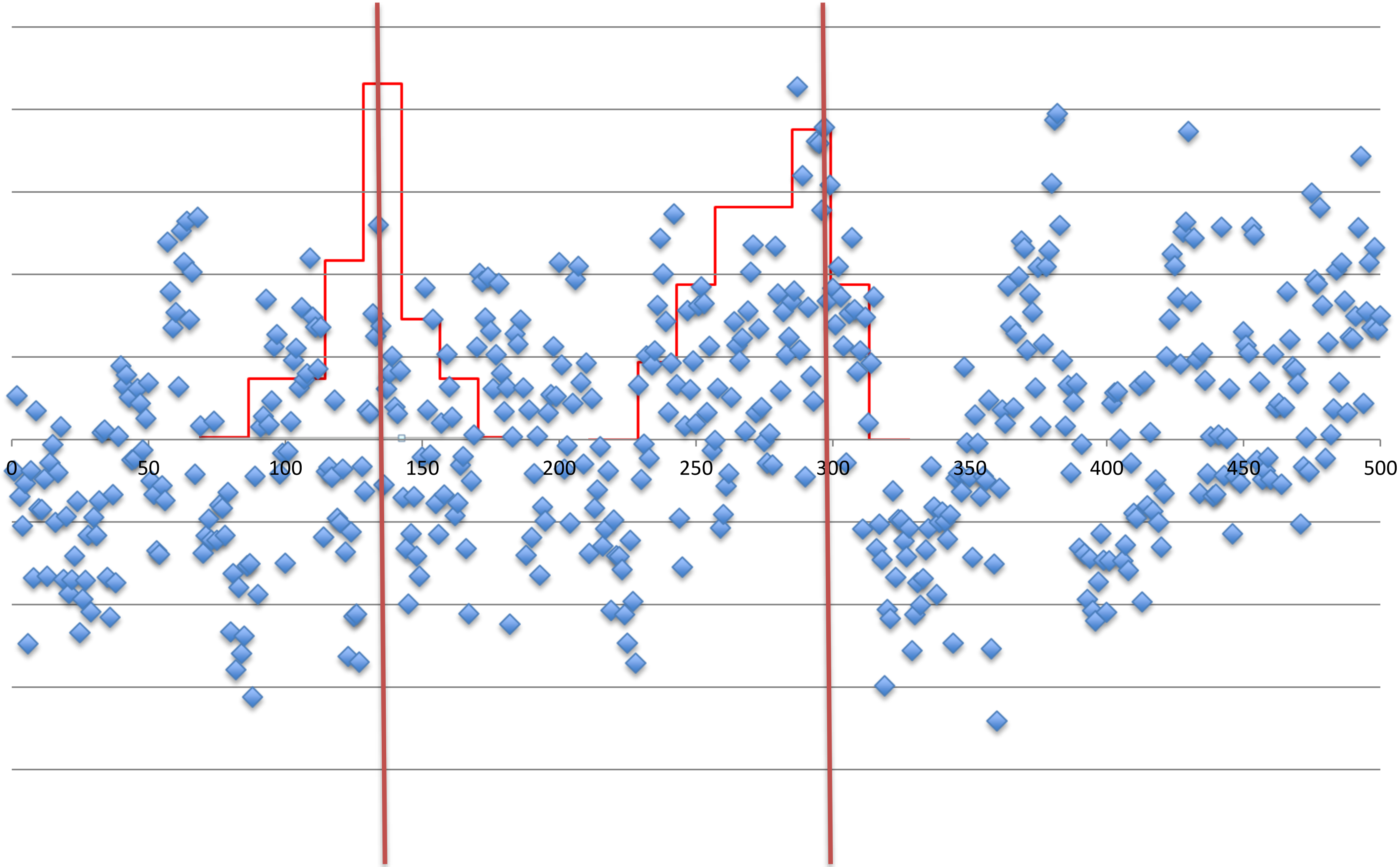
GB Decision Tree Weights



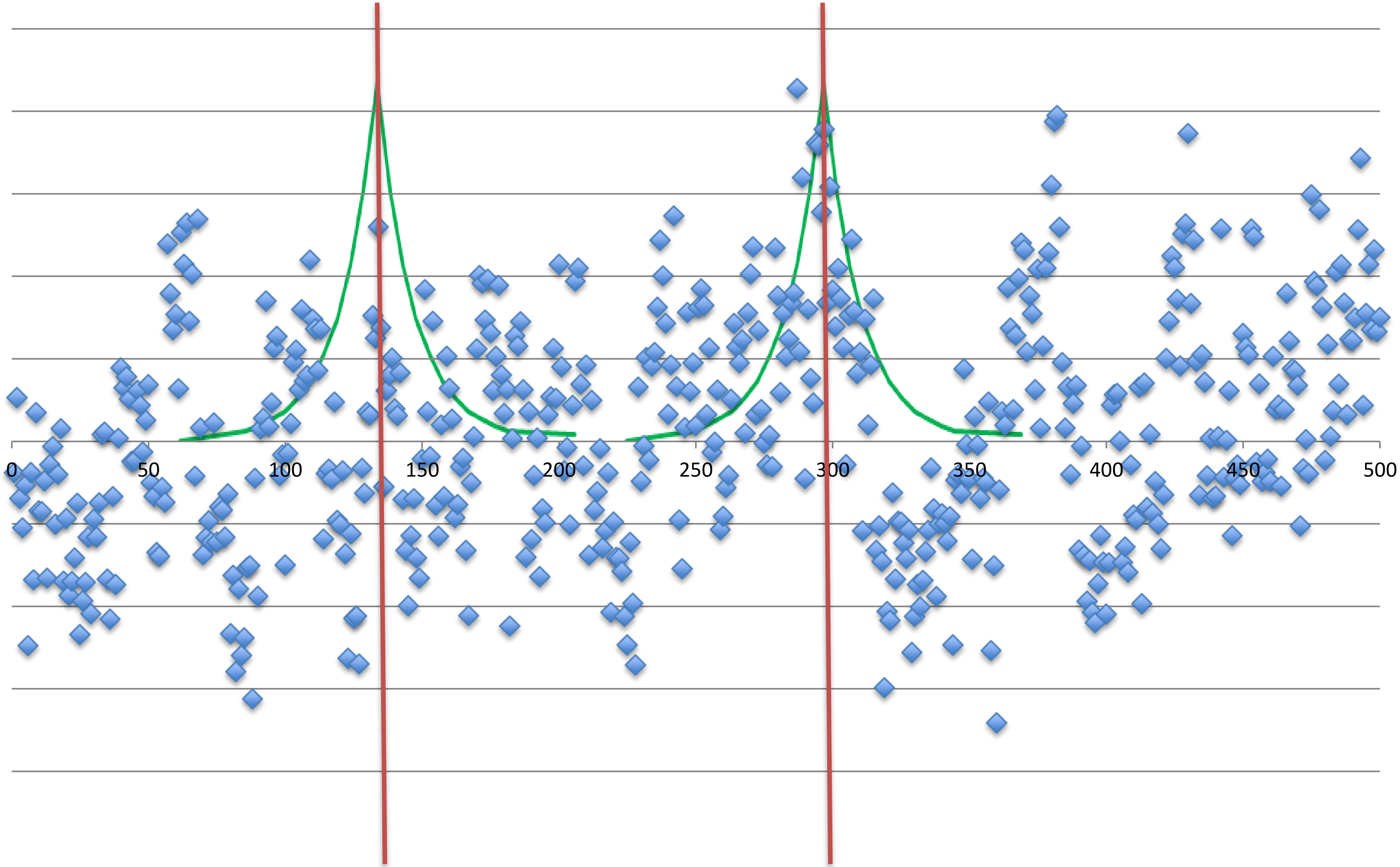
Argument #2

- We may want to force a similar importance to nearby observations across the entire spectrum of a feature
  - Rather than have discontinuities that are dependent on the local structure.

GB Decision Tree Weights



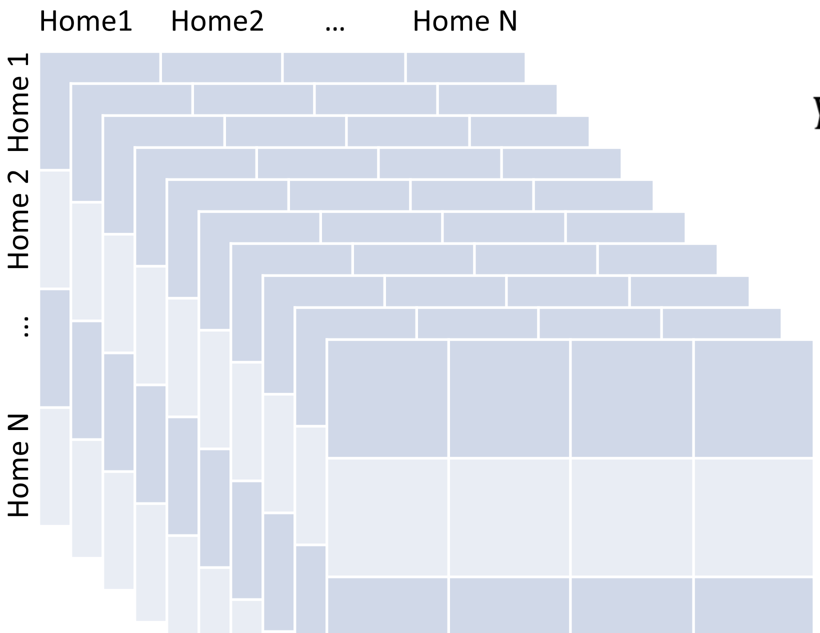
Alternative Weights
(equal functions of nearby observations)



Process
(Constrained Function)

1. Create measures of pairwise similarity for X and Y variables.
2. Fit new Y with new X.

Pairwise Relationships



$$YSimilarity = 2 * \frac{Y * Y'}{|Y| + |Y|}$$

$$X1Similarity = |X - X'|$$

$$X2Similarity = \frac{X}{X'} - 1$$

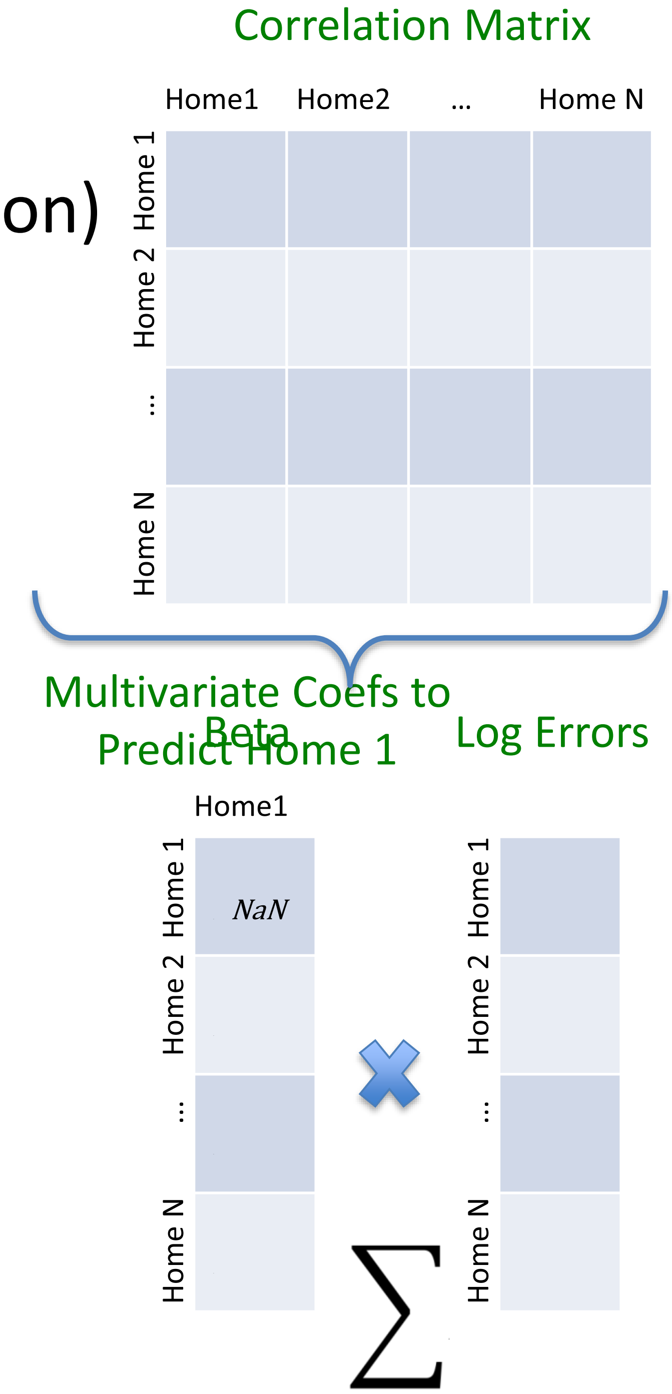
Process

(Constrained Function)

1. Create measures of pairwise similarity for X and Y variables.
2. Fit new Y with new X.
3. Using the model from (2), generate an expected correlation matrix between each home and all homes in the training set.
4. Extract multivariate coefficients from matrix (required Tihkonov regularization).
5. Multiply coefficients by training log errors, sum result.

➢ New feature (or into second layer of stacking)

Note: We prohibited matches with a time difference of less than 30 days; this was to mirror the test set prediction circumstances (so this wouldn’t take too much explanatory power from correlated variables that have more stable power over longer horizons).



Other Items we Explored

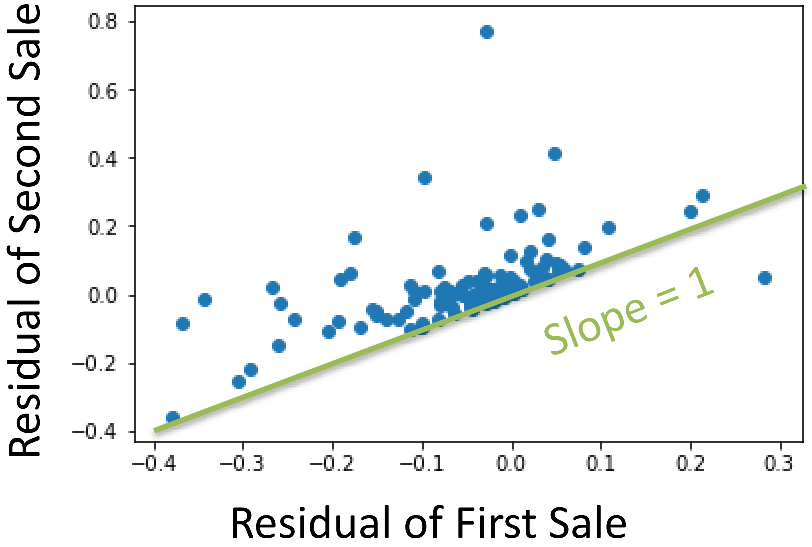
Some of these had explanatory power, but almost all were dropped due to tradeoffs between their limited benefits and computational cost / complexity.

Note: no outside data.

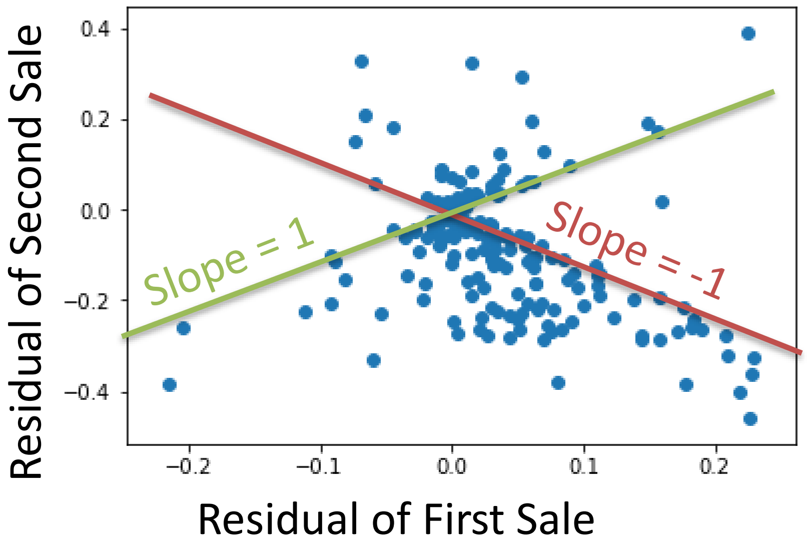
1. Fit assessment values with the other features. Use the residual as a new feature.
2. Width of the street (proxy for traffic).
3. The direction the home was facing.
4. Density of the neighborhood.
5. Near a “park” (empty space)
6. Proportion of nearby homes that were recently sold.
7. Prior same-home sales
    - This one was odd – there appears to be a structural break ~ on Jan 1st 2017.

Resales

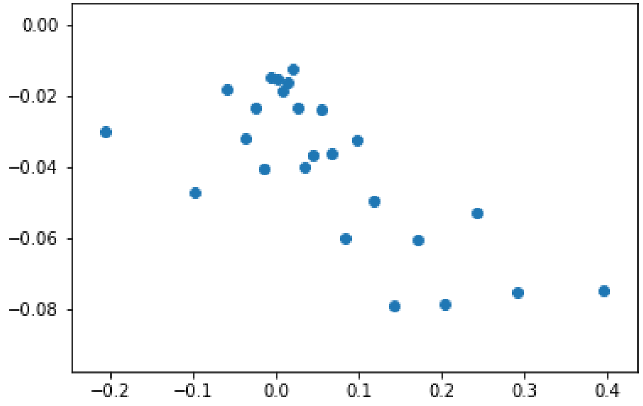
2016 for both sales



2017 for both sales



2016 for first sale; 2017 for second



STACKING

Stacking

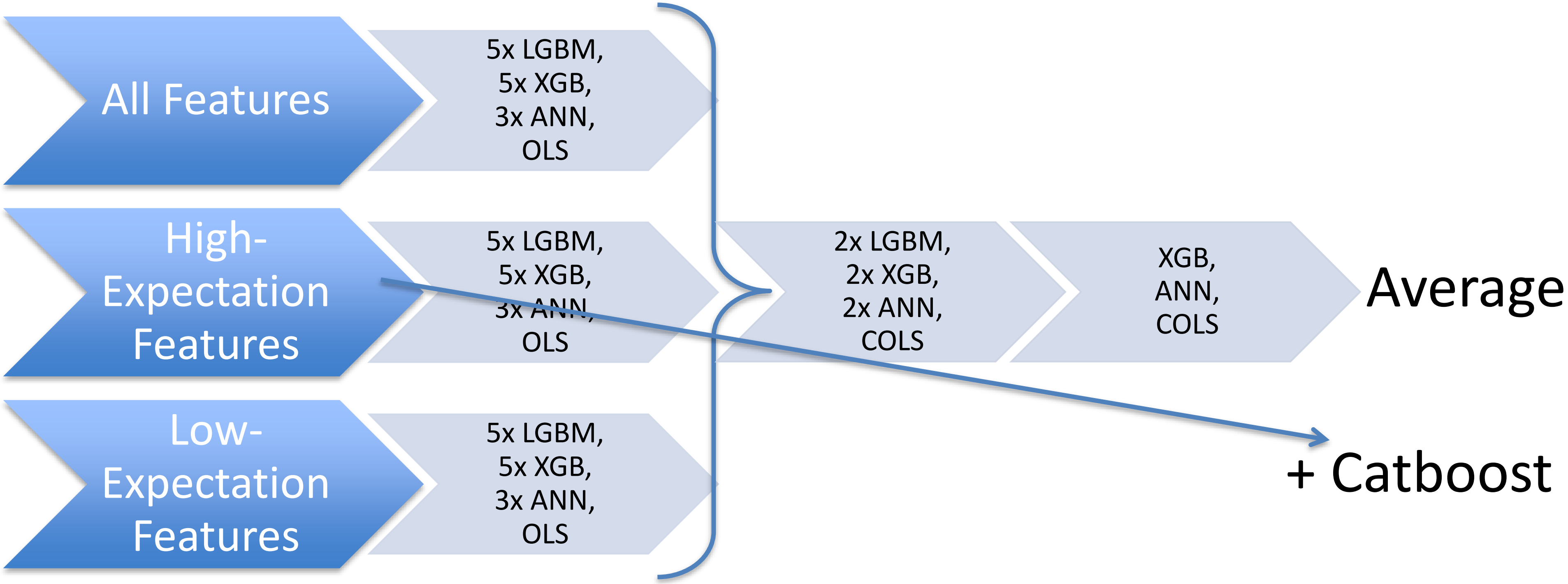
- First/first serious competition; we didn't have any code built up, so this part was pretty limited.
  - LightGBM, XGBoost, ANN (MLP), OLS, Constrained OLS.
  - Three layers
- Then last day: Add Catboost @ ~45% weight with no parameter tuning.

Stacking - Wrinkle

- If you have priors of different strength for your features, you would like to reflect this in your modeling via shrinkage of observed relationships.
  - Since most ML functions are not built to handle priors of different strength, we instead did this by modeling segregated feature sets in the first layer.

Stacking

Add training vs test



ADD-ON FEATURES

2017 Data Changes: Fit to Residuals of Overall Model on 2017 Data Only

Intuition:

- The aforementioned resale idea.
- A new bathroom is not the same as a bathroom (etc.).

- New Y variable: Residuals to our other model (using 2016 & 2017 data) – but using only 2017 values.
- New X variable: Changes in the data (only where the change was meaningful).

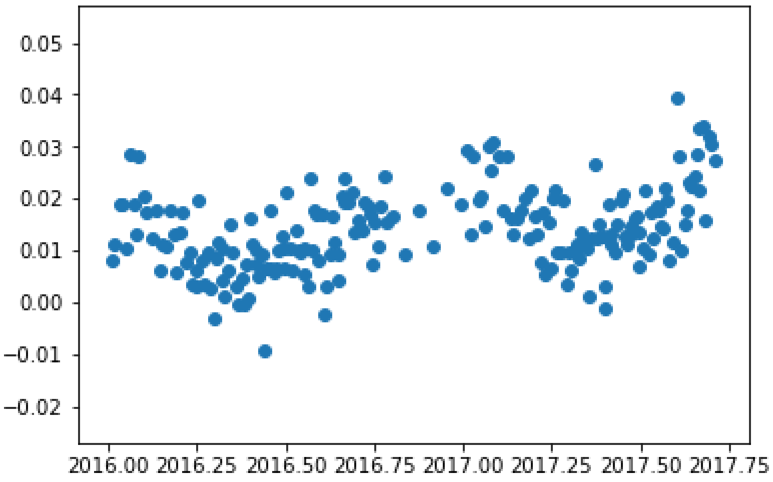
Notes:

- We cannot use these changes with 2016 data because it is forward-looking.
- We used these features on residuals of our general model because they are correlated with other features that existed for both 2016 and 2017, and we weren’t confident in the models’ ability to control for the inconsistent feature correlations across time.

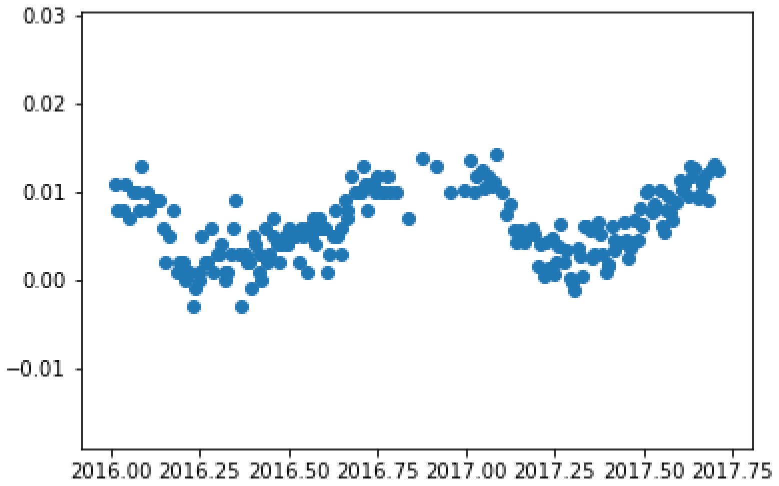
Average Forecast

- Estimating the future logerror is difficult:

Average LogError by Time



Median LogError by Time



  - Two final submissions are allowed. I entered one at 0.0115 and another at 0.0155.

OTHER TEAMS

Other Successful Solutions

- Some participants with little kaggle experience but decent scores got major boosts shortly after pairing with experienced kagglers – likely by running their features through well designed parameter tuning & model stacking code.
- Genetic algorithms.

Thanks!

- Morgan Gough:
  - Background: Research & modeling
- Dashiell Gough:
  - Background: Software, ML, & Marketing