Feature Engineering and Selection…
Chapter 7: Detecting Interaction Effects

Interactions

- “For many problems, additional variation in the response can be explained by the effect of two or more predictors working in conjunction with each other. As a simple conceptual example of predictors working together, consider the effects of water and fertilizer on the yield of a field corn crop. With no water but some fertilizer, the crop of field corn will produce no yield since water is a necessary requirement for plant growth. Conversely, with a a sufficient amount of water but no fertilizer, a crop of field corn will produce some yield. However, yield is best optimized with a sufficient amount of water *and* a sufficient amount of fertilizer. Hence water and fertilizer, when combined in the right amounts, produce a yield that is greater than what either would produce alone.”
- Use expert knowledge if possible… (will not be discussing though)

Detecting Interaction Effects

Continuous x categorical

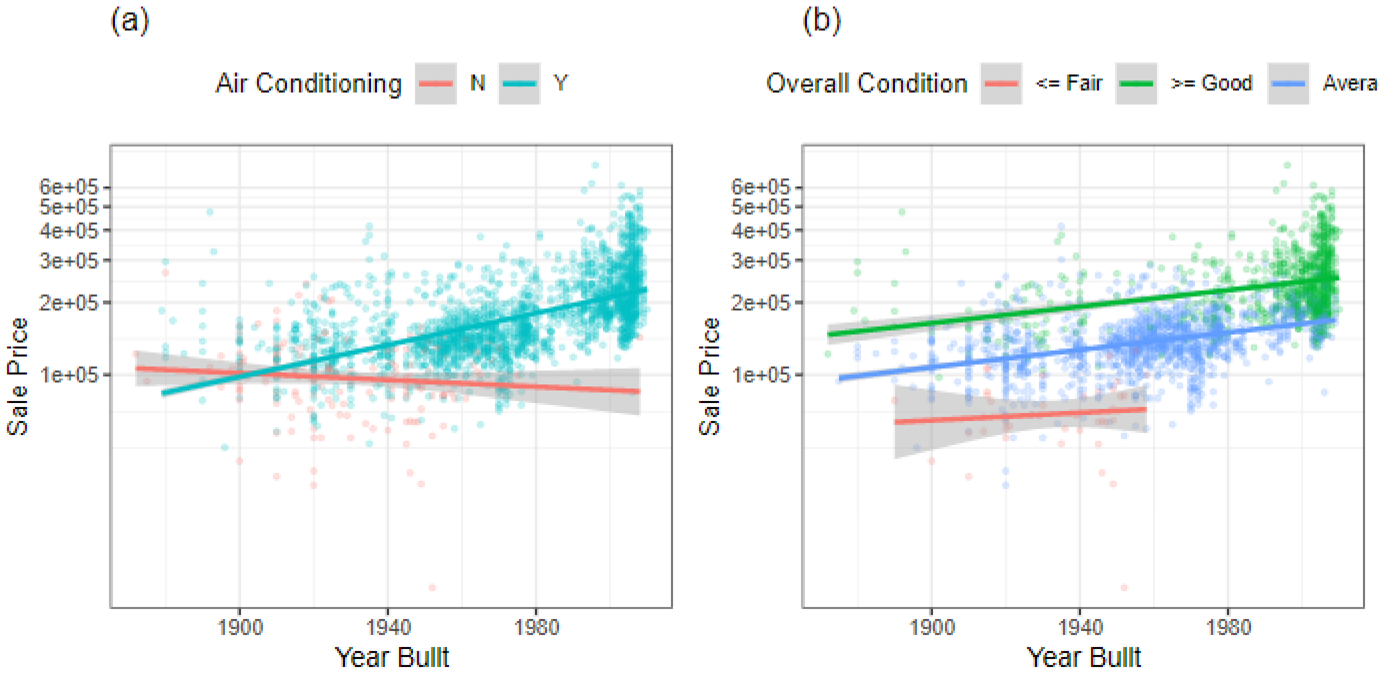
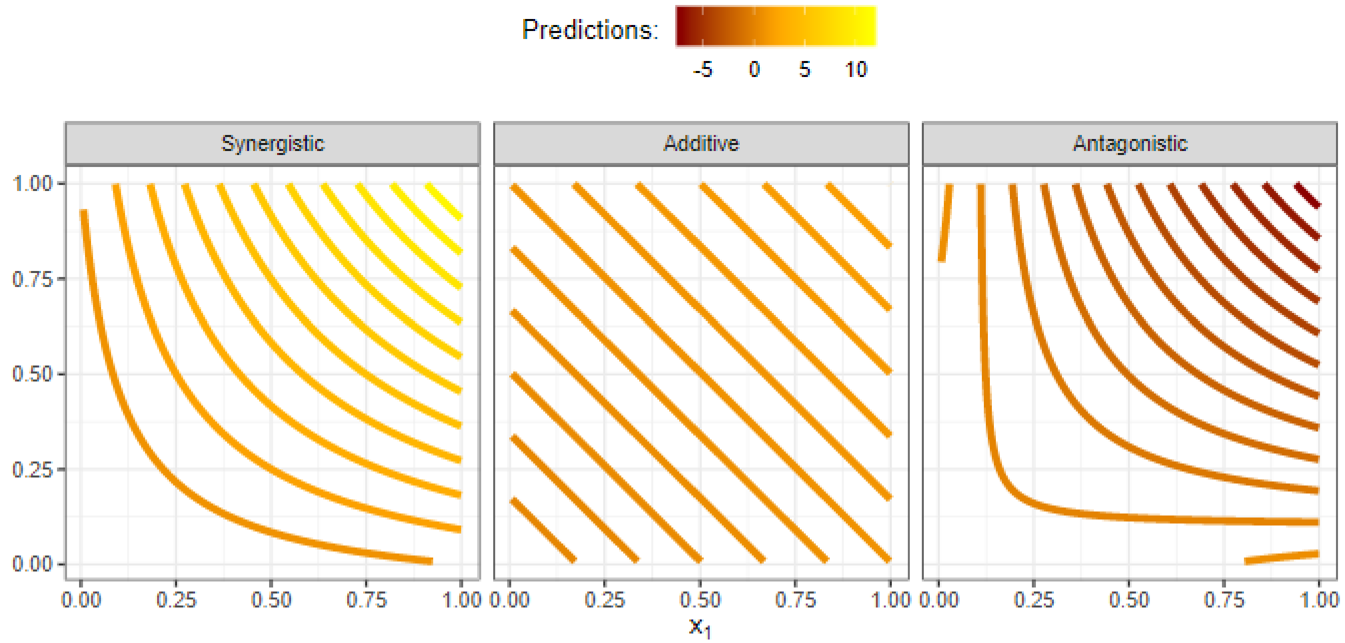


Figure 7.1: Plots of Ames housing variables where the predictors interact (a) and do not interact (b).

Detecting Interaction Effects

Continuous x continuous

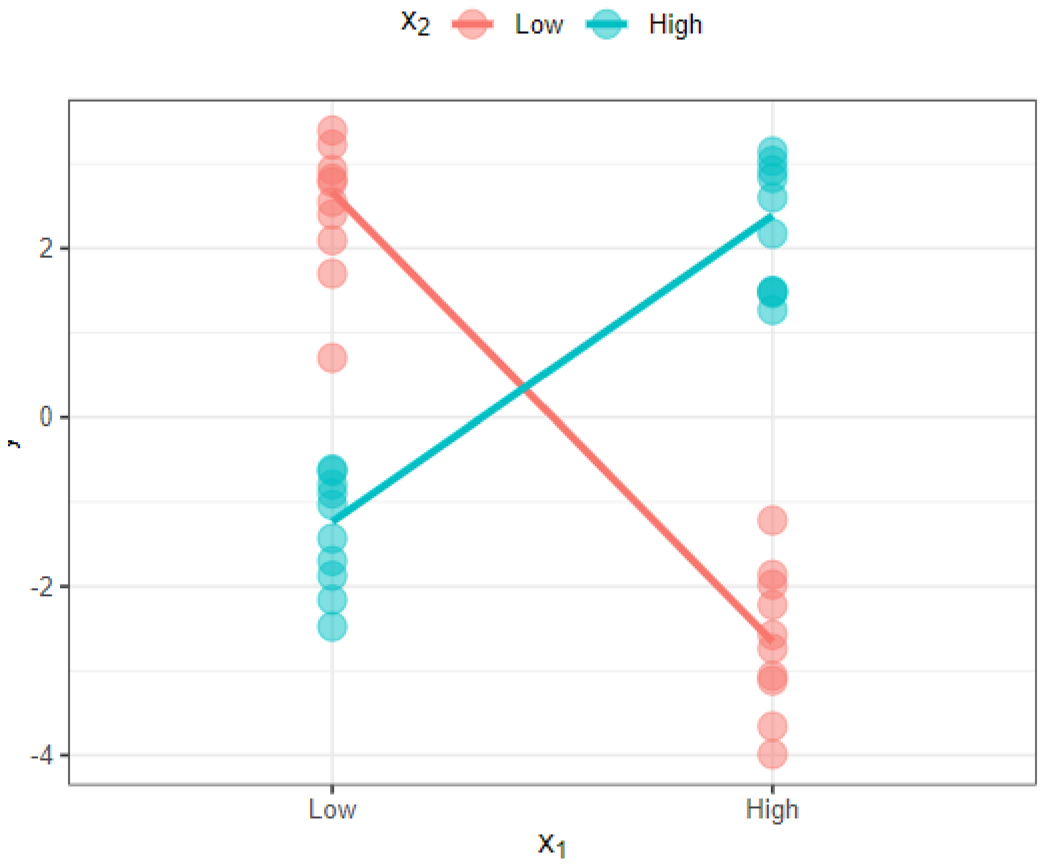


Encoding interaction effect:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + error$$

Detecting Interaction Effects

Categorical x categorical

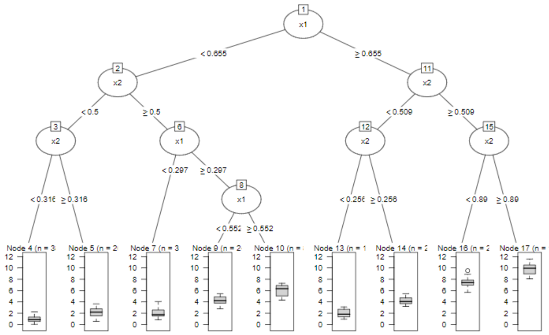


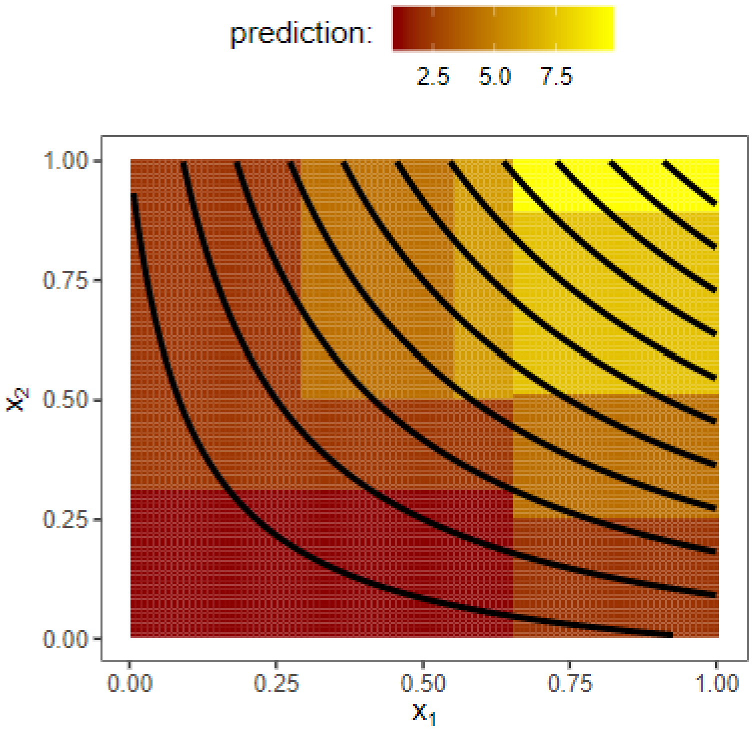
Why not just use trees?

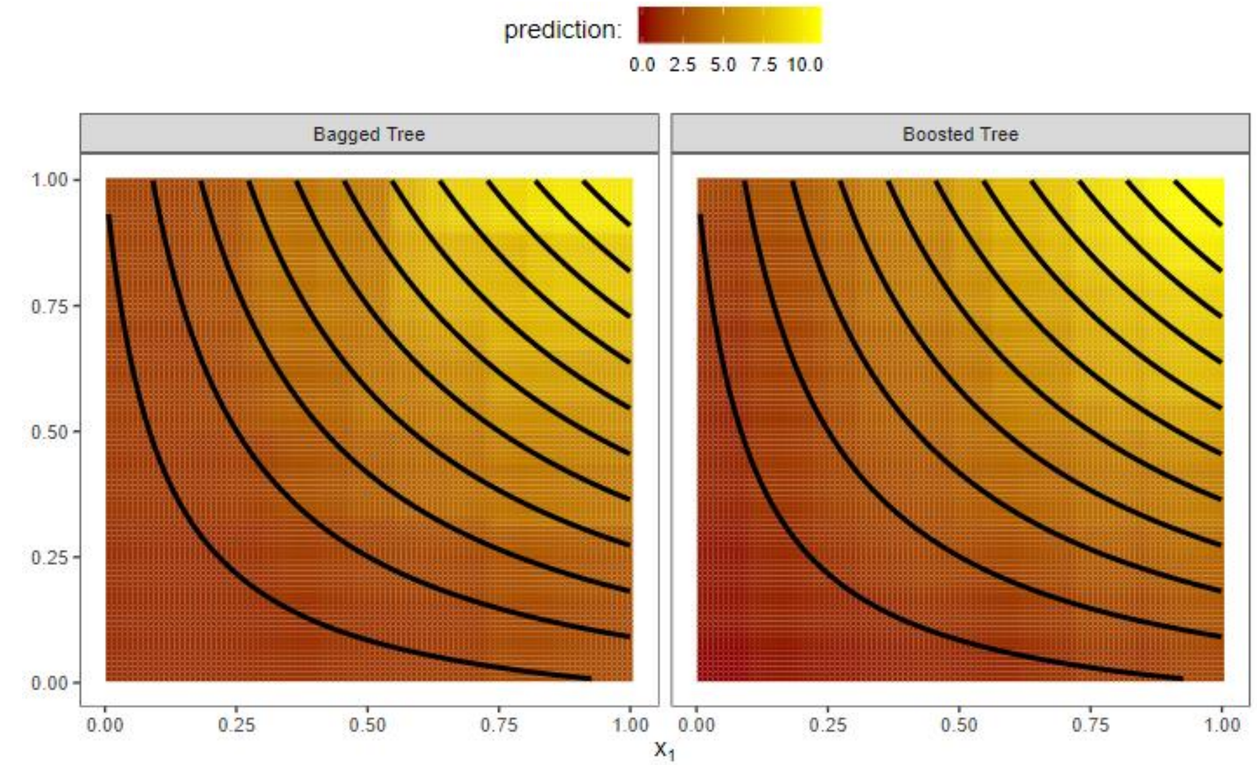
- “ These findings prompt the question that if sophisticated predictive modeling techniques are indeed able to uncover the predictive information from interactions, then why should we spend any time or effort in pinpointing which interactions are important? The importance comes back to the underlying goal of feature engineering, which is to create features that improve the effectiveness of a model by containing predictively relevant information. By identifying and creating relevant interaction terms, the predictive ability of models that have better interpretability can be improved.”

Why not just use trees?

Decision tree; bagged tree; boosted tree



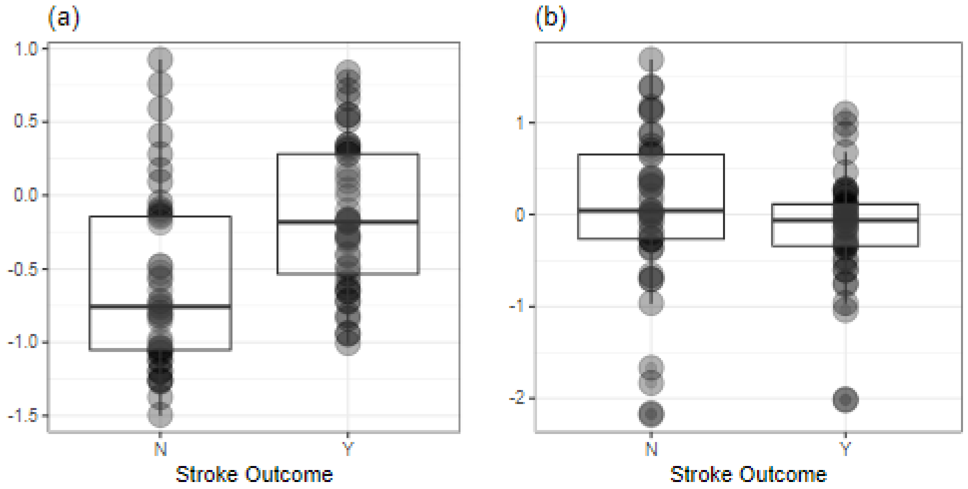




Also, if an interaction exists, will not capture it quite as granularly. More work to interpret. Though will come back to…

Preprocess before or after?

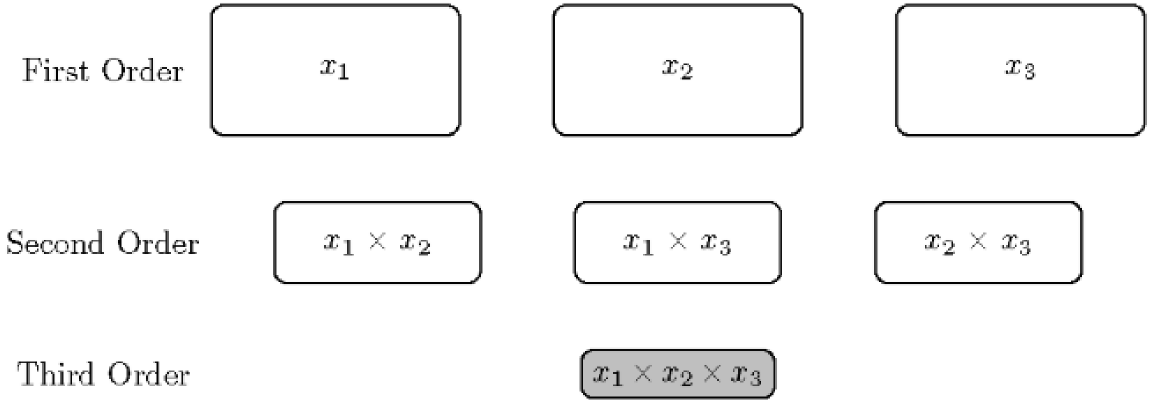
- “For these data, the preprocessing steps were centering, scaling, and individually transformed… the interactive predictive signal is almost completely lost when the original predictors are preprocessed prior to creating the interaction term.”



Some principles / heuristics in search

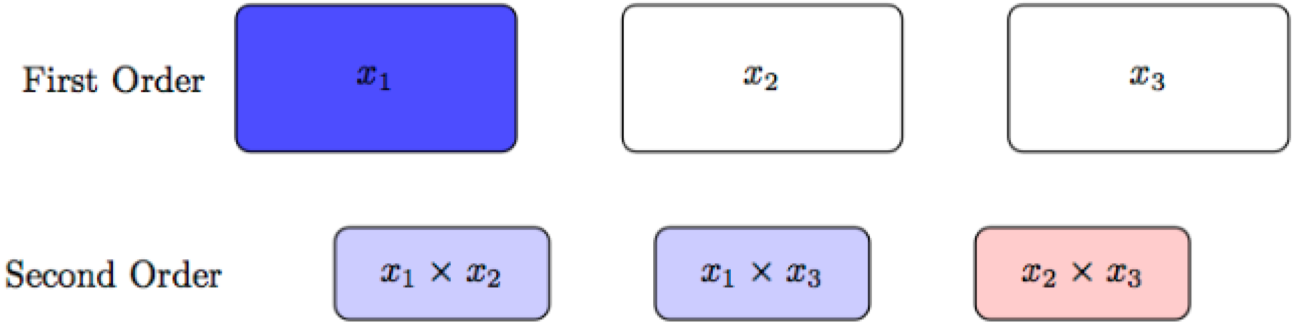
- “The interaction **hierarchy principle** states that the higher degree of the interaction, the less likely the interaction will explain variation in the response.”
- “The second principle, **effect sparsity**, contends that only a fraction of the possible effects truly explain a significant amount of response variation”
- PROBLEM: possible interactions increases exponentially
- “With as few as 100 original predictors, complete enumeration requires a search of 4,950 terms. A moderate increase to 500 predictors requires us to evaluate nearly 125,000 pairwise terms!”

Some principles / heuristics in search



First Order	x_1	x_2	x_3
Second Order	$x_1 \times x_2$	$x_1 \times x_3$	$x_2 \times x_3$
Third Order		$x_1 \times x_2 \times x_3$	

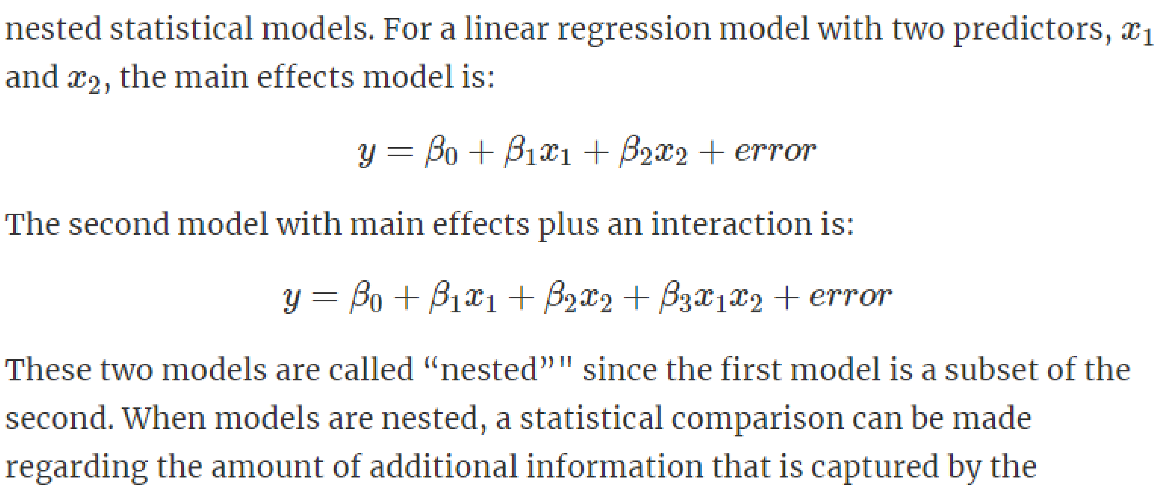
- Example of “weak” hereditary principle



First Order	x_1	x_2	x_3
Second Order	$x_1 \times x_2$	$x_1 \times x_3$	$x_2 \times x_3$

Brute force approach

Simple screening



nested statistical models. For a linear regression model with two predictors, x_1 and x_2, the main effects model is:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + error$$

The second model with main effects plus an interaction is:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + error$$

These two models are called “nested”" since the first model is a subset of the second. When models are nested, a statistical comparison can be made regarding the amount of additional information that is captured by the

- False discovery rate is also major concern.
- P-value; False Discovery Rate adjustment; Bonferroni adjustment
- Ideally, use resampling techniques

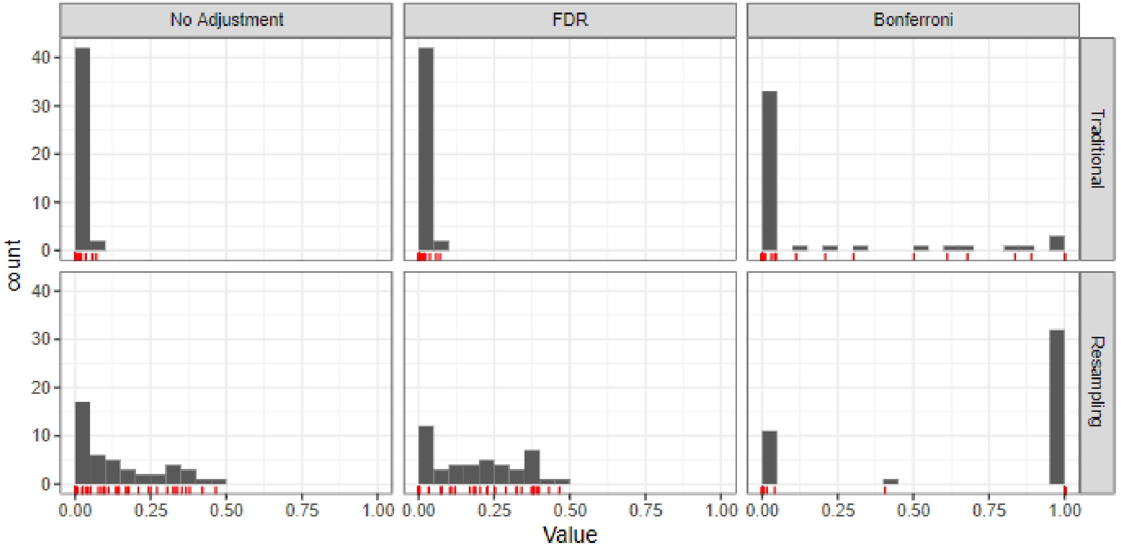


Figure 7.7: A comparison of the the distribution of p-values and adjustment method for p-values between traditional and cross-validated estimates for interactions among predictors that produce a reduction in the RMSE.

Penalized Regression

Regular linear regression:

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

where

$$\hat{y}_i = \widehat{\beta}_1 x_1 + \widehat{\beta}_2 x_2 + \ldots + \widehat{\beta}_p x_p$$

Penalized regression:

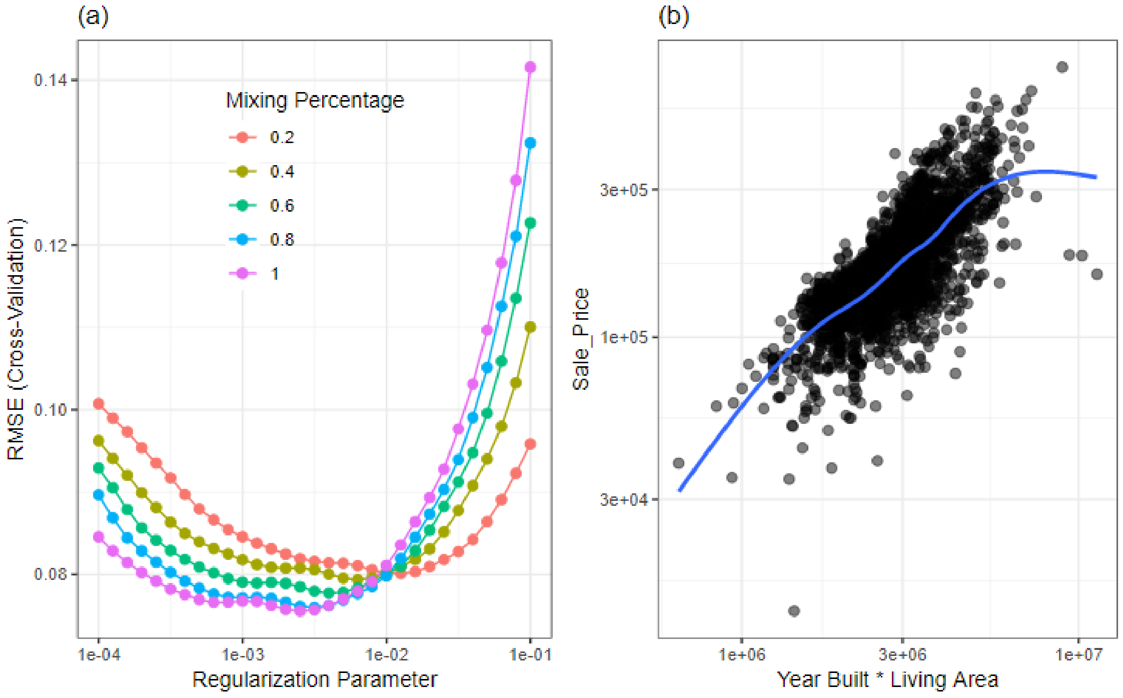
$$SSE_{L_2} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda_r \sum_{j=1}^{P} \beta_j^2$$

$$SSE_{L_1} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda_\ell \sum_{j=1}^{P} |\beta_j|$$

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + (1 - \alpha)\lambda_r \sum_{j=1}^{P} \beta_j^2 + \alpha\lambda_\ell \sum_{j=1}^{P} |\beta_j|$$

Penalized regression example

1033 predictors, 115 terms selected in best model
Bonus: interpretable, and coefficients are more meaningful



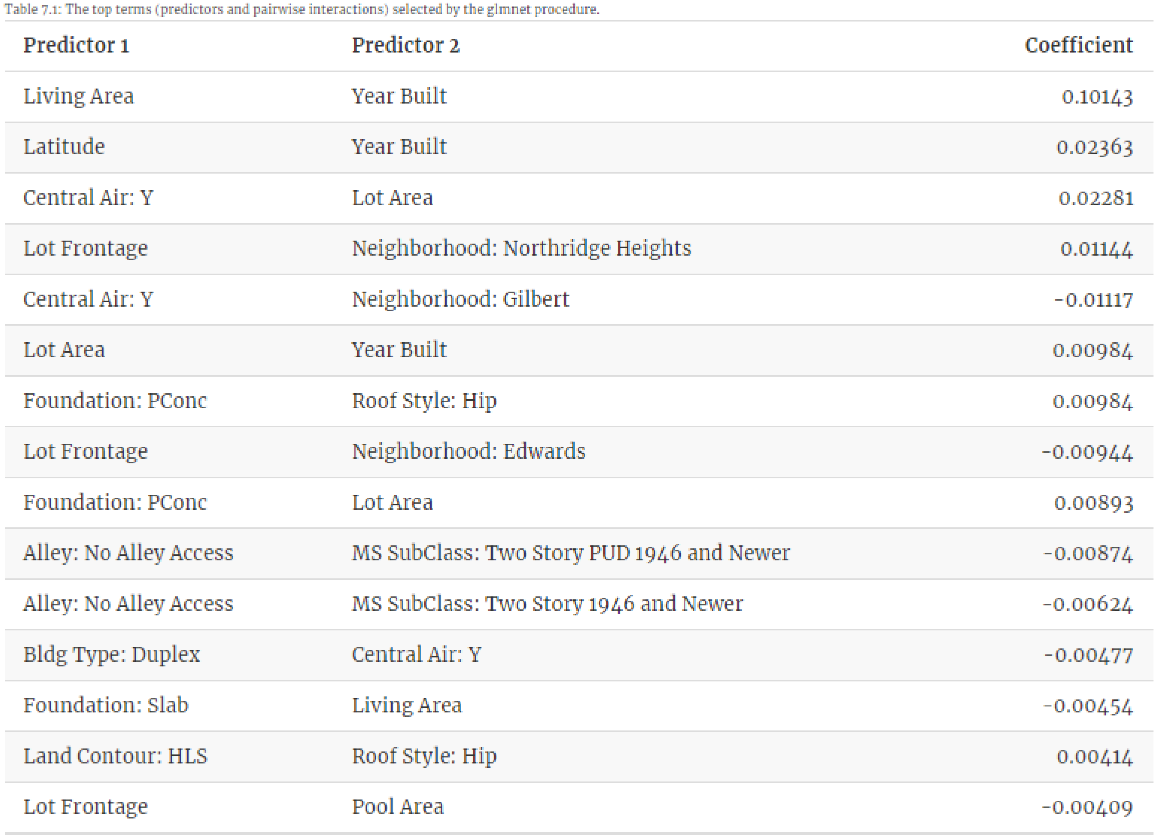


Table 7.1: The top terms (predictors and pairwise interactions) selected by the glmnet procedure.

Predictor 1	Predictor 2	Coefficient
Living Area	Year Built	0.10143
Latitude	Year Built	0.02363
Central Air: Y	Lot Area	0.02281
Lot Frontage	Neighborhood: Northridge Heights	0.01144
Central Air: Y	Neighborhood: Gilbert	-0.01117
Lot Area	Year Built	0.00984
Foundation: PConc	Roof Style: Hip	0.00984
Lot Frontage	Neighborhood: Edwards	-0.00944
Foundation: PConc	Lot Area	0.00893
Alley: No Alley Access	MS SubClass: Two Story PUD 1946 and Newer	-0.00874
Alley: No Alley Access	MS SubClass: Two Story 1946 and Newer	-0.00624
Bldg Type: Duplex	Central Air: Y	-0.00477
Foundation: Slab	Living Area	-0.00454
Land Contour: HLS	Roof Style: Hip	0.00414
Lot Frontage	Pool Area	-0.00409

Two-stage modeling

Example

- Use hereditary and SPARSITY principles to limit size of investigation, steps:
  1. Identify, *using only base variables as inputs*, which variables are important
  2. Use either the weak or strong hierarchy principle to create all associated pairwise interactions
  3. Build new model with base inputs and all interaction effects
- For step 1, you might use any of the following:
  - Selected base variables from lasso (or elastic net) regression
  - Most “important” variables as found by Random Forest; gradient boosting; etc.
  - Any method that will do feature selection for you…

P.s. I was a little confused on a minor point here so opened up a question online: https://community.rstudio.com/t/two-stage-modeling-example-in-feature-engineering-kuhn-johnson/42889

Tree Based methods

- Tree based models naturally produce interactions – however they are “local” interactions so require multiple of them to effectively capture an interaction effect and will be somewhat partitioned. Though random forests and gradient boosting can do a better job, an interaction (if captured properly) can be better (and more simply) explained in a linear model (see slide 7 for images)
- However LOCALIZED interactions may be captured better or at least more simply with tree-based methods.

More notes on trees

- Feature weighted random forests:
  - “The approach presented by Basu et al. (2018) uses a form of a random forest model (called feature weighted random forests) that randomly selects features based on weights that are determined by the features’ importance. After the ensemble is created, a metric for each co-occurring set of features is calculated which can then be used to identify the top interacting features.”
- Method based on partial dependence:
  - “compares the joint effect of two (or more) predictors with the individual effect of each predictor in a model. If the individual predictor does not interact with any of the other predictors, the difference between the joint effect and the individual effect will be close to zero.”
  - Produces an H statistic
- Feature Importance metrics can also be used to select “main effects”

Feasible solution algorithm

- Forward and Backward selection algorithms can be used but suffer the challenge of not having consistent or necessarily optimal final predictor sets
- The Feasible Solution algorithm is a method that should produce optimal (though potentially locally optimal) predictor sets

Other potentially useful tools

- MARS/FDA (Flexible discriminant analysis)
- Cubist (builds tree with each node having a linear model)