# An Empirical Analysis of Feature Engineering for Four Regression Models

Jeff Heaton

Nova Southeastern University - Ft. Lauderdale, FL USA
Reinsurance Group of America (RGA) - St. Louis, MO USA

April 1, 2016

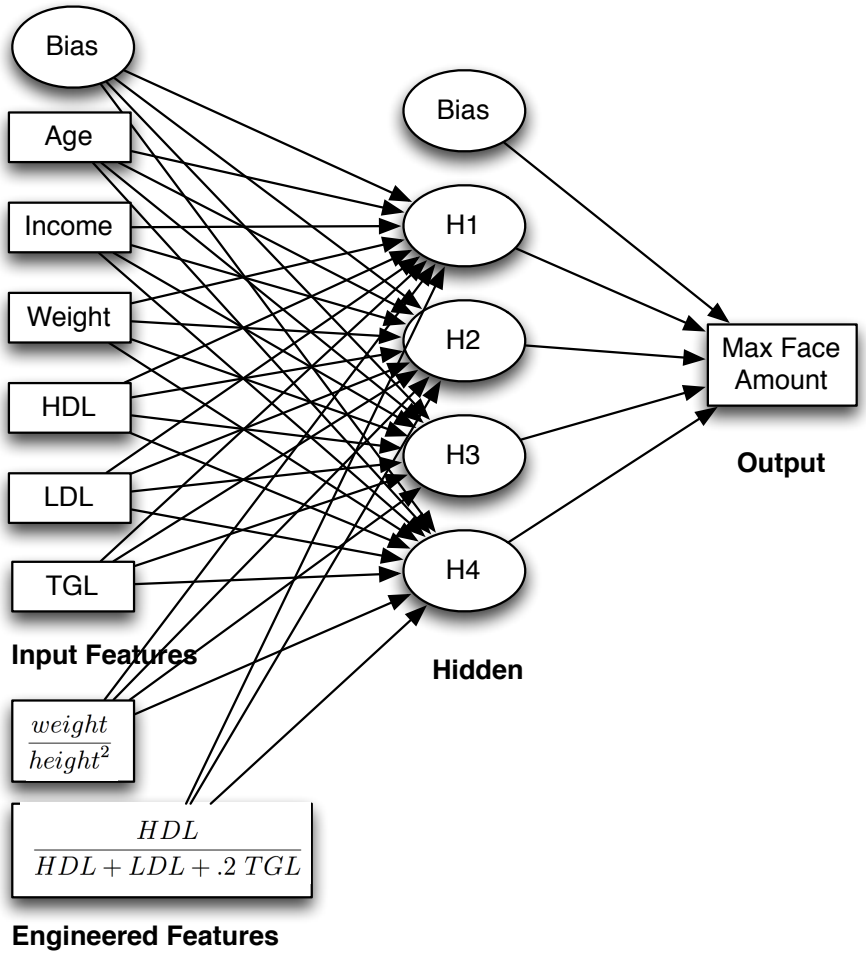
http://www.jeffheaton.com/

# What this Talk About?

- **Feature** - An observation, or single column of data, that is part of the input to a model.
- **Feature Vector** - A fixed-length collection of features.
- **Model** - An algorithm that accepts a feature vector and provides an outcome/prediction. Models covered in this talk: neural networks (deep and otherwise), support vector machines (SVMs), random forests and gradient boosted forests/machines (GBMs).
- **Feature Engineering** - Creating additional calculated features that help a predictive model achieve a more accurate outcome.

# Simple Example of Engineered Features

A simple toy neural network.




- Toy neural network to recommend a maximum face amount for an insurance applicant.
- Bias values are used by neural networks to provide a y-intercept.
- Regular features are: age, income, weight, HDL, LDL, and TGL.
- Two engineered features: BMI and cholesterol ratio.

# Outline

# Motivation and Problem Statement

Does the choice of model (neural network, support vector machine, etc.) also dictate the types of engineered features that should be investigated?

- Are certain types of engineered feature unnecessary for certain model types?
- Do other types of engineered feature have more potential for other model types?
- If certain engineered feature types are unnecessary for some model types, then their inclusion simply adds complexity to the preprocessing pipeline.

# Related Work

An important part of predictive modeling since the beginning.

- **Feature Engineering**
  - Linear regression with non-normally distributed features (Freeman and Tukey, 1950).
  - Feature engineering benefits deep learning for speech recognition, computer vision, & signal processing. (Bengio, 2013).
- **Automatic Feature Engineering**
  - Box-Cox Transformations (Box & Cox, 1964).
  - Alternating Conditional Expectation (ACE) (Breiman & Friedman, 1985).
- **Application of Feature Engineering**
  - Kaggle Algorithmic Trading Challenge (Ildefons & Sugiyama, 2013).
  - ACM KDD Cup 2010 (Yu, et al., 2011).
- *Others given in the conference paper...*

# Research Methodology

## Evaluating effectiveness of engineered features for models.

- How can the effectiveness for a type of engineered feature be evaluated for a type of model?
  - Not all datasets benefit from engineered features.
  - Models will often figure out an engineered feature for themselves.
- Evaluation method used in this research:
  - Generate 10 datasets where the outcome is the result of the engineered feature.
  - Fit 4 regression models to each of these 10 datasets and evaluate root mean square error (RMSE).
  - A low RMSE indicates that the model is capable of synthesizing the feature on its own, independent of any engineering efforts. A high RMSE indicates a more valuable type of engineered feature.
  - Keep expected outcome (Y) values in similar ranges so dataset RMSE values are approximately comparable.

# Research Methodology

## Models considered.

- **Deep Artificial Neural Network (DANN)**
  - 5 hidden layers (Input$\rightarrow$ 400 $\rightarrow$ 200 $\rightarrow$ 100 $\rightarrow$ 50 $\rightarrow$ 25 $\rightarrow$1 output).
  - Stochastic Gradient Descent (SGD) (learning rate: $1 \times 10^{-5}$, momentum: 0.9).
  - Rectified Linear Unit (ReLU) based transfer function.
- **Support Vector Machine (SVM)**
  - Grid search of $C$ values of 0.001, 1, and 100 in combination with $\gamma$ values of 0.1, 1, and 10.
  - Gaussian kernel.
- **Random Forest**
  - Estimators (trees): 100
  - Maximum tree depth: 10
- **Gradient Boosted Machine (GBM, XGBoost)**
  - Learning Rate: 0.1
  - Estimators (trees): 100
  - Maximum tree depth: 10

# Research Methodology

## Datasets generated.

- **Counts:** Count of 50 features above a threshold.
- **Differences:** $y = x_1 - x_2$
- **Logarithms:** $y = \ln(x_1)$
- **Polynomials:** $y = 1 + 5x + 8x^2$
- **Powers:** $y = x^2$
- **Ratios:** $y = \frac{x_1}{x_2}$
- **Rational Differences:** $y = \frac{x_1 - x_2}{x_3 - x_4}$
- **Rational Polynomials:** $y = \frac{1}{5x + 8x^2}$
- **Root Distance:** $y = \left| \frac{-b + \sqrt{b^2 - 4ac}}{2a} - \frac{-b - \sqrt{b^2 - 4ac}}{2a} \right|$
- **Square Roots:** $y = \sqrt{x}$

# Research Methodology

## Example dataset.

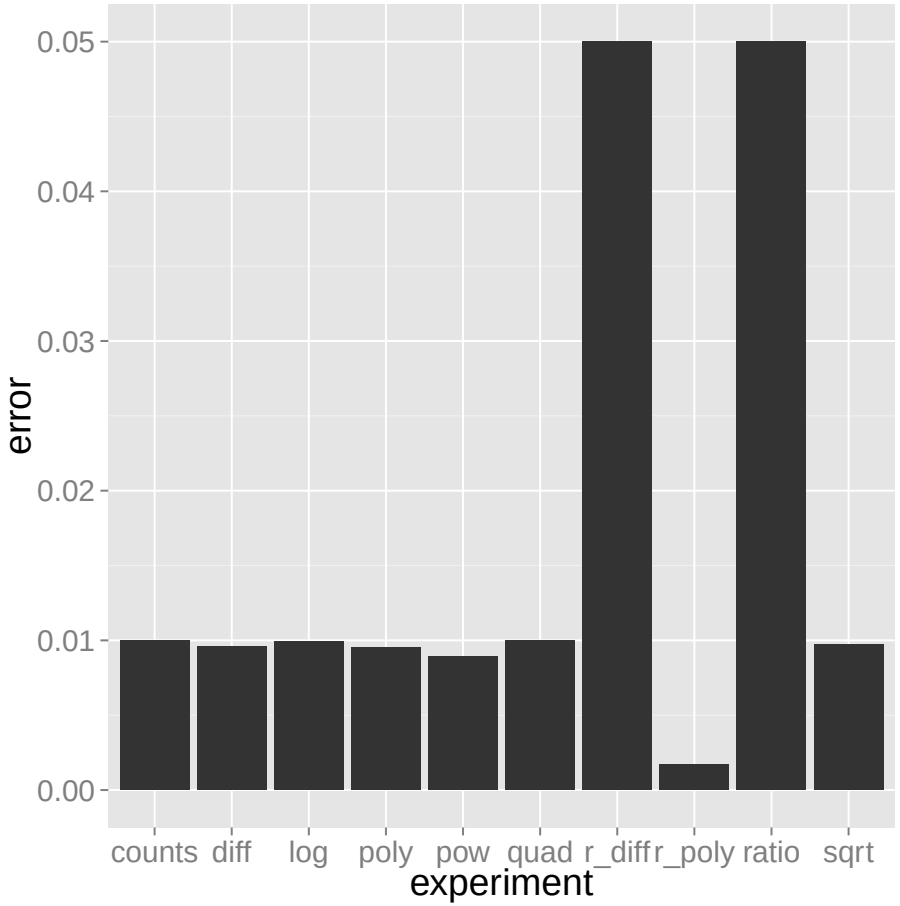
- **Rational Differences:** $y = \frac{x_1 - x_2}{x_3 - x_4}$
- Can the model learn the above equation, given the data below?

Table: Rational Difference Transformation

| $x_1$ | $x_2$ | $x_3$ | $x_4$ | $y_1$ |
|---|---|---|---|---|
| 6.30651 | 2.23126 | 6.95826 | 9.88415 | -1.39282 |
| 9.07714 | 6.21059 | 1.58401 | 1.97679 | -7.29794 |
| 5.02777 | 4.04626 | 3.90232 | 9.15452 | -0.18688 |
| 1.90746 | 7.76275 | 8.44665 | 7.09478 | -4.33127 |
| ... | ... | ... | ... | ... |

# Results

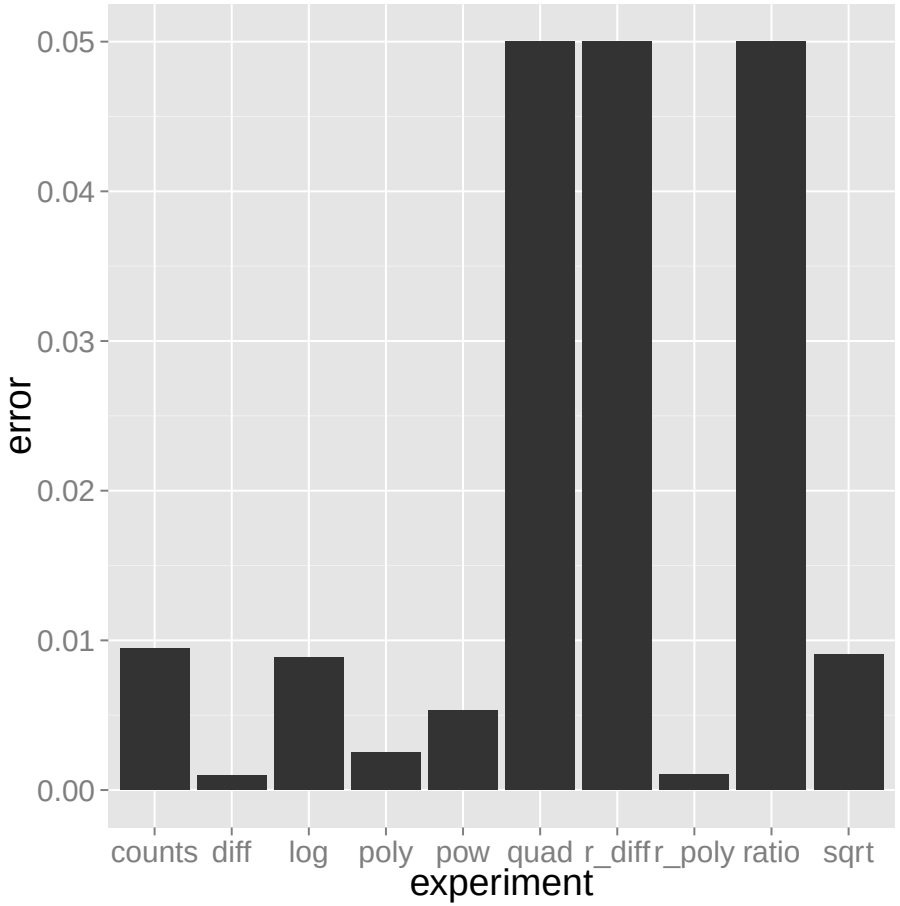
## Neural Networks.




- The neural network had difficulty with ratios and difference ratios.
- Neural network calculation: $f(x, w, b) = \phi\left(\sum_n (w_i x_i) + b\right)$
- Neural networks are function approximators that can sum and multiply effectively.

# Results

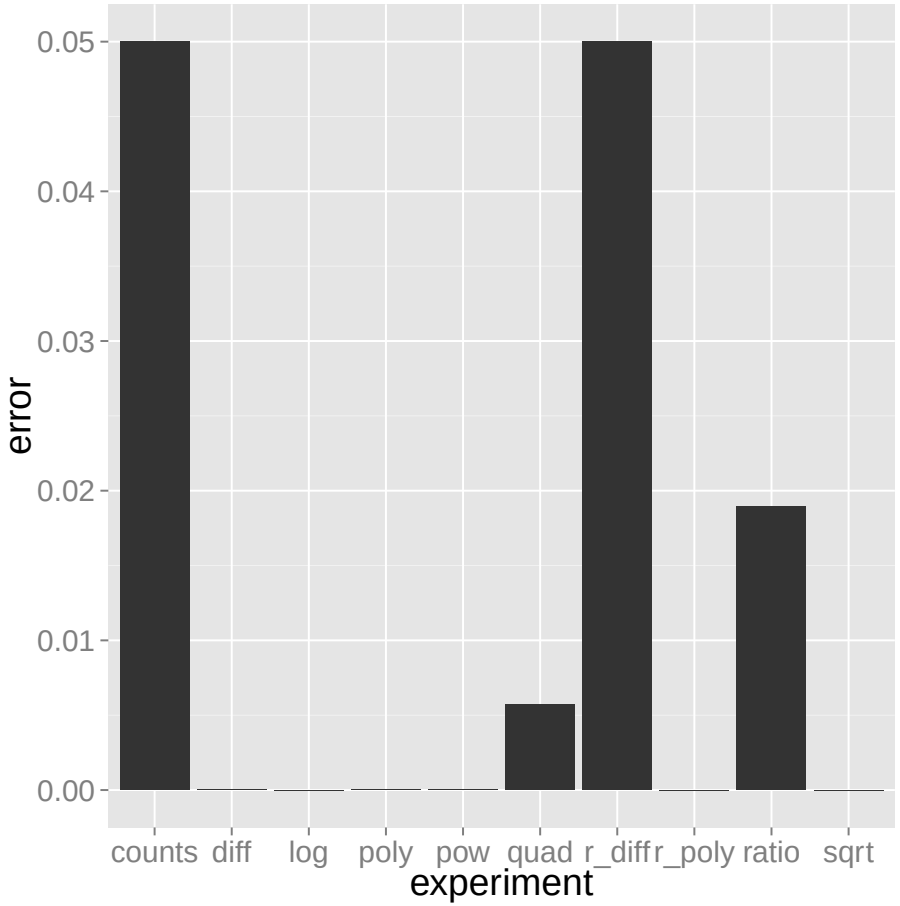
## Support Vector Machines (Regression) (SVR).




- The support vector regression (SVR) had difficulty with root difference (quadratic), ratios, and difference ratios.
- SVR decision function (calculation):
  $y = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) K(x_i, x) + \rho$
- Like neural networks, SVR's are function approximators that can sum and multiply effectively.

# Results

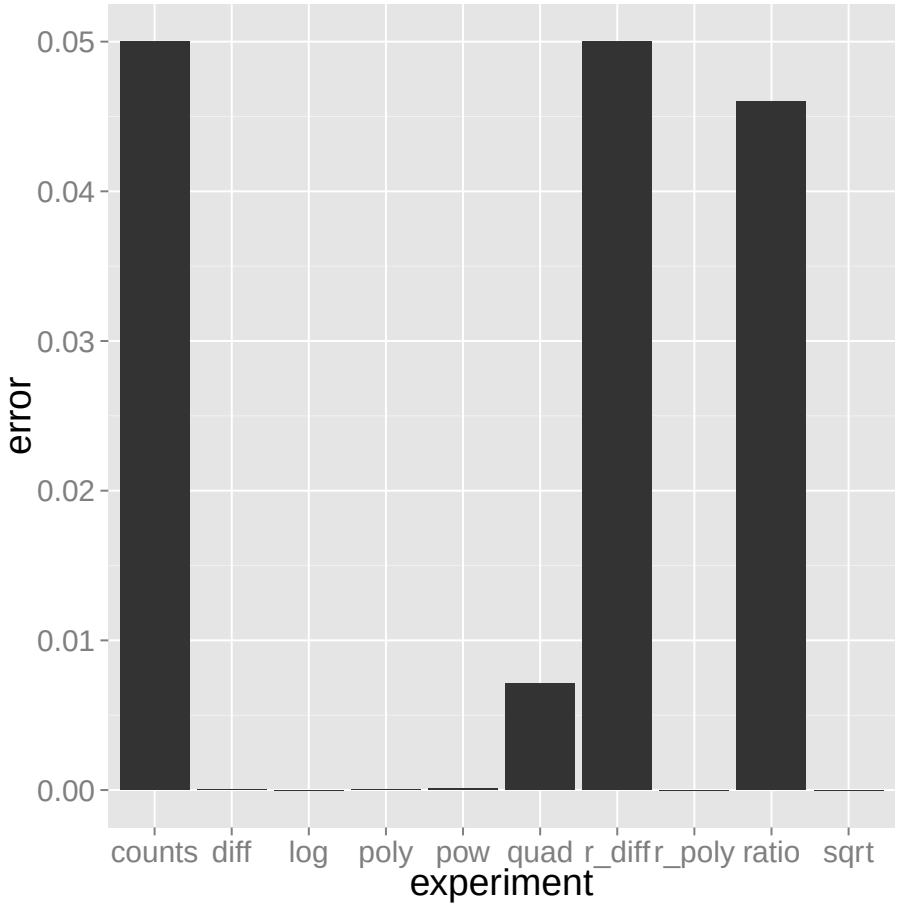
## Random Forests.




- The regression random forest had difficulty with counts and difference ratios.
- Random forests are calculated using trees of essentially local linear regressions and voting among the trees.
- Complex interactions among all features (such as a count) are difficult without great tree depth.

# Results

## Gradient Boosted Forests/Machines (GBM).




- The GBM had difficulty with counts and difference ratios.
- GBM's are calculated similar to random forests, but voting structure is decided with gradient descent.
- Like random forests, complex interactions among all features (such as a count) are difficult without great tree depth.

# Summary of Results

- No model (with provided hyper-parameters) was effective at the difference ratio feature. This implies that this feature type might be useful for these models (if supported by the dataset).
- All models evaluated did well with the simple (single feature) transformations. This implies that these feature types might not be as useful for these model types.
- Dot product based models (neural networks & SVR) have difficulty with similar engineered features.
- Tree-based models (random forest & GBM) have difficulty with similar engineered features.
- Because of their different approaches dot product and tree-based models are often used together in ensembles.

# Future Work

- Model RMSE results might improve with additional tuning (better hyper-parameters).
- Further explanation of why certain models perform better with particular equations.
- Try additional features and model types.
- Explore classification results of the four model types.
- Experiment with the effects of noise.
- Other engineered features: sum, mean, standard deviation, max and min.
- More than one engineered feature at a time.
- Automatic selection of engineered features.