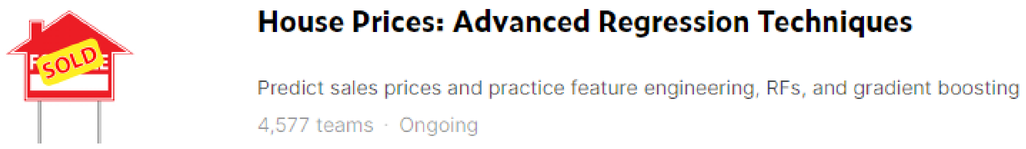
Advanced Regression: Exploring Lasso, Ridge, XGBoost, and Elastic Net Techniques in an Online Data Science Competition

Matthew Lister
matthew.lister@aggiemail.usu.edu
Utah State University
Logan, Utah

Jake Rhodes
jakerhodes@aggiemail.usu.edu
Utah State University
Logan, Utah

Chase Mortensen
chase.mortensen3@gmail.com
Utah State University
Logan, Utah


House Prices: Advanced Regression Techniques

Predict sales prices and practice feature engineering, RFs, and gradient boosting

4,577 teams · Ongoing

Figure 1: Kaggle competition information

ABSTRACT

The forerunner of testing regression techniques is the Boston Housing dataset, collected in 1978, is nearly 50 years old. In order to replace this valuable educational tool, Dr. Dean De Cock contacted the Ames county assessor's office in 2010 and collected the raw data which later became known as the 12 Ames Housing Dataset. The dataset was presented to his students as a semester project. [5], and eventually made its way into Kaggle as on open competition for aspiring data scientists. The authors took part in Kaggle's House Prices: Advanced Regression Techniques contest during the months of February and March 2020, achieving a top Root Mean Squared Logarithmic Error (RMSE) of 0.11984 , placing 694 out of 4577 competitors (top 15.1 percent).

The team's submissions underwent a progression of refinements during the contest. Fundamentally, all the categorical data had to be encoded into meaningful numerical data before a log transform could be performed. After reprocessing, different regression models were applied resulting in competitive scores. However, the final improvements were made after Cook's D test [6] was performed and the outliers removed. This allowed the author's regression models to more closely match the true home values and further reduce the root mean square logarithmic error. Then, after several attempts, a lowest scoring result was found by combining lasso with xgboost.

CCS CONCEPTS

• **Computer Science** → Data Mining; Data Visualization; Python; Data Encoding; Gradient Boost; XGBoost; one-hot vectors; R; • **Computer Science Education** → Competitive Coding; CS-6665; • **Statistics** → Ridge Regression; LASSO; Elastic Net; Data Visualization.

KEYWORDS

data sets, Ridge regression, LASSO, XGBoost, Elastic Net, Visualization, CS6665, tabular data, rmsle, one-hot

ACM Reference Format:
Matthew Lister, Jake Rhodes, and Chase Mortensen. 2018. Advanced Regression: Exploring Lasso, Ridge, XGBoost, and Elastic Net Techniques in an Online Data Science Competition. In *Proceedings of ACM Conference (CS 6665).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

By 2010, the Boston housing data set had been firmly establish as the main introduction to multivariate analysis and regression for more than a generation of students. By 2010, Dr. Dean De Cock believed the data had reached the limits of its usefulness and sought out a suitable replacement candidate. He found an amicable employee in the county assessor's office of Ames, Iowa, that provided him with their complete records of all property transactions from 2006 to 2010 [5].

Dr. De Cock further refined the data set to include only sales of single family homes and removed multiple listing. This left a data set of 2930 listings with 80 variable and a mixture of nominal, ordinal, discreet, and continuous values to for his students to analyze. While Dr. De Cock originally intended to use this data set to teach his traditional college statistics students, he did published a paper on his classroom methodologies and made the Ames housing data publicly available. The developers at Kaggle incorporated this data set into their "House Prices: Advanced Regression Techniques" contest [11].

However, the simple linear regression strategies that Dr. De Cock envisioned his students using would not place well for an open competition of aspiring data scientists. The authors quickly realized that due to col-linearity in the data set and stiff competition on the website, several advanced regression techniques would have

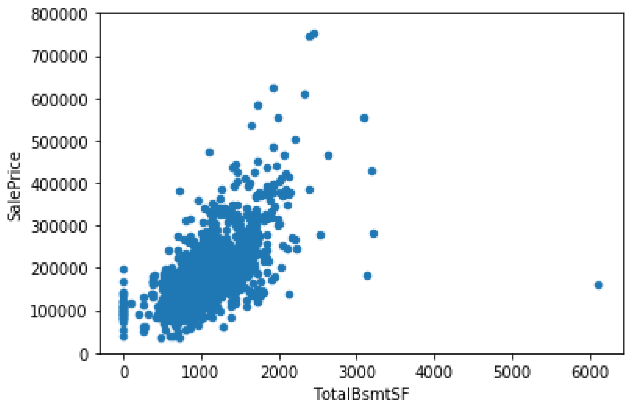


Figure 2: Data Visualization Hints at the Importance of Outliers [12]

to be employed. They tried multiple techniques including ordinary least-squares regression, ridge regression [10], Lasso [18], Random Forest [3], and gradient boost [4]. The highest performance was achieved by an ensemble of LASSO (Least absolute shrinkage and selection operator) and XGBoost [4, 15, 18].

The team's highest score was achieved after employing data visualization techniques to identify several outliers. The existence of extreme data points where initially hinted at during routine data exploration [12], however they where not clearly identified until Cook's D test [6] was performed. After removing extreme outliers, the team was able to the apply Papiu's combination of lasso and xgboost [15] to the reduced data set. Improving on both Papiu's results and achieving their smallest Root Mean Squared Logarithmic Error (the competition's scoring metric).

2 DATA PREPARATION

Arguable the biggest challenge of this Kaggle competition was cleaning the data set. The given Ames housing data set had many missing or incomplete values. The data imputation was quite a manual process; instead of imputing values using, say, random forests or by means of another machine learning algorithm, the data gave itself to imputation based on human knowledge. For instance, if a given house had no basement, it would naturally have no basement square footage. Instead of keeping the value as 'NA', we would enter a value of 0 square feet. Many such examples could be given. For categorical variables, it made sense in many cases to create another category as 'None' in place of 'NA'.

In order to run any of our regression models, all categorical variables needed to be encoded as one-hot vectors. This allowed for numerical coefficients to be meaningfully applied to the problem. For model stability, log-transformations were applied to numerical variables, including the response variable, Sale Price. This transformation alone accounted for a three- to five-hundredth decrease in the root-mean-squared error rates of preliminary models.

Several data visualization techniques were applied to the housing data to identify outliers. In regression analysis, outliers can have

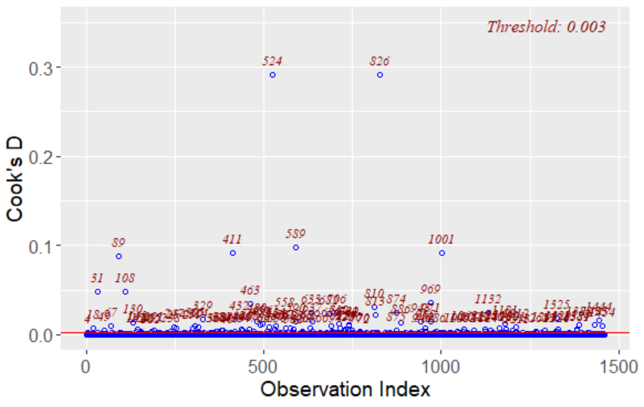


Figure 3: Cook's D Check for Outliers

a significant impact on the resulting value of the parameter being estimated. In addition to ordinary scatter plots and residual plots for outlier identification, Cook's distance test (Cook's D) was applied as a more robust measure of observation importance. Cook's distance is calculated for a given observation by computing the regression coefficients of a least-squares model after having removed the observation from the dataset. A distance (D) is calculated between the model given all observations and the model with the removed observation; the distance is given by

$$D_i \equiv \frac{\left(\hat{\beta}_{(-i)} - \hat{\beta}\right)' \mathbf{X}'\mathbf{X} \left(\hat{\beta}_{(-i)} - \hat{\beta}\right)}{ps^2}$$

Where $\hat{\beta}_{(-i)}$ is the vector of coefficients of the model with observation i removed. A threshold is determined by comparing the distance with the limits of a 95% confidence interval for the β coefficients [6]. The higher the Cook's distance, the more influential the point is in a linear regression model.

3 METHODS

3.1 Ridge Regression

Ridge regression is a technique for reducing variance in multidimensional data introduced by Hoerl and Kennard in 1970 [10]. By applying a penalty term to the variances of each term, bias is introduced to the estimator but the overall variance is reduced and the total error decreased. Ridge regression complements the Bayesian approach to parameter estimation because it is able be introduced as a Gaussian posterior distribution [19] . A lambda term is often called the shrinkage parameter [19] because it controls the size of the coefficients and the amount of regularization, optimizing the the mean square error. Lambda remains present in the resulting posterior distribution after calculating the Bayesian distribution of the variables.

Standard linear regression attempts to find a line which minimized the sum of the squared residuals, but suffers when data points are co-linear, or contain multiple dependant variables [7]. This is unfortunate when trying to regress the Ames housing data because a high value home will contain many desirable features (as well as the converse being true). A general equation for regression

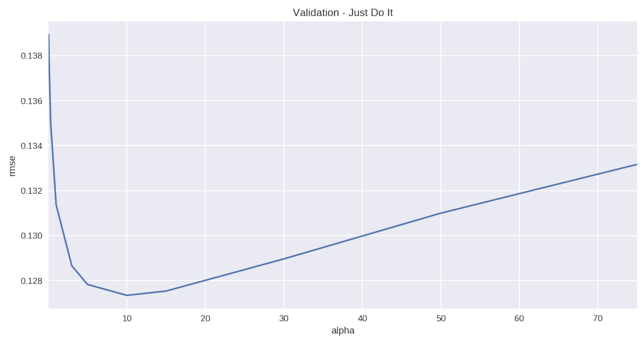


Figure 4: Picking Lambda to Minimize RMSE[12]

would be [17]

$$\underline{Y} = X\underline{B} + \underline{e}$$

Where B is the regression coefficients and e is the residuals And the resulting coefficients [17]

$$\underline{\hat{\mathbf{B}}} = \left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\underline{\mathbf{Y}}$$

This gives us an unbiased estimate of the regression coefficients but a potentially massive variance resulting in unacceptable large mean square errors [17].

$$E(\underline{\hat{\mathbf{B}}}) = \underline{\mathbf{B}}$$

$$V(\underline{\hat{\mathbf{B}}}) = \sigma^2\mathbf{R}^{-1}$$

Ridge regression alleviates this problem by introducing a biased estimator $\underline{\tilde{\mathbf{B}}}$ that Hoerl and Kennard [10] define as

$$\underline{\tilde{\mathbf{B}}} = \left(\mathbf{X}'\mathbf{X} + \lambda\right)^{-1}\mathbf{X}'\underline{\mathbf{Y}}$$

By choosing the values for lambda, the variance can be minimized. While this was once a mostly manual process, which Hoerl named a ridge trace [10], it has now been automated in modern programming libraries such as scikit-learn [16].While the authors achieved moderate results from this technique, those scores were quickly superseded by LASSO

3.2 LASSO Regression

LASSO regression can be seen as an improvement of, or compliment to, ridge regression. Like ridge regression, LASSO can penalize high regression coefficients, but LASSO can also ignore large coefficients should their importance fall below the threshold [14]. With LASSO, we are given a set of predictor d-dimensional variables $(x_{1d}, x_{2d}, \cdots, x_{nd})$ and a set of response variables $(y_1, y_2, \cdots, y_n)$, the ordinary least regression problem seeks to find a set of coefficients $(\beta_1, \beta_2, \cdots, \beta_j)$ such that the residual squared error is minimized. That is, it finds such coefficients that satisfy:

$$min_\beta \sum_{i=1}^{n}(y_i - (\beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_n x_{in})^2 = min_\beta \sum_{i=1}^{n}(y_i - \hat{y}_i))^2$$

Two issues raised by ordinary least-squares regression are brought up by Robert Tibshirani [18]:

(1) Low predictive accuracy due to large variance, and
(2) Difficult interpretation due to an excess number of variables

LASSO regression deals with both of these issues.

The issue of high variance and model overfitting is resolved by the formulation of the LASSO model. The revised model, after accounting for the bias term, or y-intercept, can be written in the form of a convex optimization problem, where all of the variables have been standardized:

$$\text{arg min}\left\{\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2\right\}$$

$$\text{subject to} \sum_{j=1}^{p}|\beta_j| \le t$$

This formulation forces the coefficients to be small which decreases the variance of the model and reduces overfitting. The constraint, which differs from the ridge regression constraint $\sum_j(\beta_j)^2$, forces the unimportant variables to zero, a feat that is not met by ridge regression.

3.3 Elastic Net

Elastic Net builds on the strengths of both LASSO and ridge regression and addresses some of their weaknesses. Hui Zou and Trevor Hastie point out three issues not fully addressed by the LASSO formulation [20]:

(1) The dimensionality of the data being greater than the number of observations only allows LASSO to select at most n variables
(2) Pairwise collinearity among predictors leads to only one of these variables to be selected
(3) Ridge regression outperforms LASSO when high correlations between predictors exist

The proposed solution by Zou and Hastie was the Elastic net. Elastic net makes use of both the L1 norm used by LASSO, and the L2 norm applied by ridge regression by weighting each of these penalty terms. The formulation minimizes over the coefficients:

$$\hat{\beta} = \underset{\beta}{\text{argmin}}\left\{\sum_{i=1}^{N}\left(y_i - \sum_{j=1}^{p}x_{ij}\beta_j\right)^2 + \lambda_1\sum_{j=1}^{p}\left|\beta_j\right| + \lambda_2\sum_{j=1}^{p}\beta_j^2\right\}$$

This formulation can be generalized to other loss functions as well.

3.4 XGBoost

Boosting is a machine learning technique that uses an ensemble of weaker models to create a single stronger model, and is primarily a result of Michael Kearns and Leslie Valiant [2] and also AdaBoost from Robert Schapire [8]. Its aim is to reduce variance and bias [1]. The high level algorithm follows a few steps:

(1) Training weaker learners and adding them to a stronger learners

```
Algorithm 1: Gradient_Boost
1  $F_0(\mathbf{x}) = \arg\min_\rho \sum_{i=1}^N L(y_i, \rho)$
2  For $m = 1$ to $M$ do:
3      $\tilde{y}_i = -\left[\frac{\partial L(y_i, F(\mathbf{x}_i))}{\partial F(\mathbf{x}_i)}\right]_{F(\mathbf{x})=F_{m-1}(\mathbf{x})}$, $i = 1, N$
4      $\mathbf{a}_m = \arg\min_{\mathbf{a},\beta} \sum_{i=1}^N [\tilde{y}_i - \beta h(\mathbf{x}_i; \mathbf{a})]^2$
5      $\rho_m = \arg\min_\rho \sum_{i=1}^N L(y_i, F_{m-1}(\mathbf{x}_i) + \rho h(\mathbf{x}_i; \mathbf{a}_m))$
6      $F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \rho_m h(\mathbf{x}; \mathbf{a}_m)$
7  endFor
   end Algorithm
```

Figure 5: Friedman's 1999 Gradient Boosting algorithm

```
Input  : Data set $\mathcal{D}$.
         A loss function $L$.
         The number of iterations $M$.
         The learning rate $\eta$.
         The number of terminal nodes $T$
1 Initialize $\hat{f}^{(0)}(x) = \hat{f}_0(x) = \hat{\theta}_0 = \arg\min_\theta \sum_{i=1}^n L(y_i, \theta)$;
2 for $m = 1,2,..,M$ do
3     $\hat{g}_m(x_i) = \left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x)=\hat{f}^{(m-1)}(x)}$;
4     $\hat{h}_m(x_i) = \left[\frac{\partial^2 L(y_i, f(x_i))}{\partial f(x_i)^2}\right]_{f(x)=\hat{f}^{(m-1)}(x)}$;
5     Determine the structure $\{\hat{R}_{jm}\}_{j=1}^T$ by selecting splits which maximize
      $Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L} + \frac{G_R^2}{H_R} - \frac{G_{jm}^2}{H_{jm}}\right]$;
6     Determine the leaf weights $\{\hat{w}_{jm}\}_{j=1}^T$ for the learnt structure by
      $\hat{w}_{jm} = -\frac{G_{jm}}{H_{jm}}$;
7     $\hat{f}_m(x) = \eta \sum_{j=1}^T \hat{w}_{jm} \mathrm{I}(x \in \hat{R}_{jm})$;
8     $\hat{f}^{(m)}(x) = \hat{f}^{(m-1)}(x) + \hat{f}_m(x)$;
9 end
Output: $\hat{f}(x) \equiv \hat{f}^{(M)}(x) = \sum_{m=0}^M \hat{f}_m(x)$
```

Figure 6: The Newton Boosting algorithm which is implemented in XGBoost

(2) The data is weighted or re-weighted based on accuracy of the learners (misclassified data is usually given a higher weight and correctly classified data loses weight)
(3) Steps 1 and 2 are repeated, with later iterations yielding learners targeted toward the misclassified data

Jerome Friedman built on the foundation of boosting with a Gradient boosting algorithm developed in 1999 [9]. It involves optimizing for a specific cost function, such as XGBoost's default squared error by following the negative gradient direction. Friedman's 1999 algorithm is shown in Figure 5.

Like most boosting algorithms, XGBoost utilizes tree boosting. However, instead of Gradient tree boosting, XGBoost implements Newton tree boosting - a similar algorithm to Gradient tree boosting, but it has a slight advantage in determining tree structure [13]. Nielsen also states that XGBoost has certain advantages over other gradient boosting libraries, such as many adjustable parameters, Newton boosting, and "higher-order approximation at each iteration." The Newton tree boosting algorithm which XGBoost implements is shown in figure 6.

4 RESULTS

The competition results were determined by computing the root-mean-squared error (RMSE) of the log of the sale prices, rounded to five decimal places. Ties were broken by timestamp. Our best overall RMSE was 0.11984, placing us in the top 15%. Below is a table of our overall results.

Method	RMSE Val.	RMSE Test
OLS	0.11428	6.06079
Ridge	0.11011	0.12647
Lasso	0.10341	0.12153
Elastic Net	0.10342	0.12086
Random Forest	0.13055	0.14195
Gradient Boosting	0.04911	0.12584
XGBoost	0.11160	0.11984

While these results are based on proven regression techniques the Kaggle competition has many higher results posted that involve non-regression based techniques. The Ames county assessors office has a web enabled API [5] that allows competitors to achieve zero or nearly zero scores. However, these scores and notebooks are routinely purged from the competition, but have the side effect of making the authors submissions fluctuate up and down several percentage points [11].

Personal Bests:

Contributor	Method	RMSE	Position
Chase	XGBoost	0.11984	683
Matt	ElasticNet	0.12086	764
Jake	LASSO	0.12153	824

The personal best scores from which the team members achieved their individual rank was a direct result of working closely with the successful submissions of the other team members. Similarly to how Kaggle contests drive their leader board from a review of submissions and notebooks, our team worked in a iterative process. Foundational knowledge of the analysis should be credited to Jake Rhodes, for which the other authors are grateful.

REFERENCES

[1] Leo Breiman. 1996. *Vertex Types in Book-Embeddings.* Technical Report. Berkeley, CA, USA.

[2] Leo Breiman. 1998. Arcing classifier (with discussion and a rejoinder by the author). *Ann. Statist.* 26, 3 (06 1998), 801–849. https://doi.org/10.1214/aos/1024691079

[3] Leo Breiman. 2001. Random Forests. *Mach. Learn.* 45, 1 (Oct. 2001), 5–32. https://doi.org/10.1023/A:1010933404324

[4] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, California, USA) *(KDD '16)*. ACM, New York, NY, USA, 785–794. https://doi.org/10.1145/2939672.2939785

[5] Dean De Cock. 2011. Ames, Iowa: Alternative to the Boston Housing Data as an End of Semester Regression Project. *Journal of Statistics Education* 19, 3 (2011), 1–14. www.amstat.org/publications/jse/v19n3/decock.pdf

[6] R. Dennis Cook. 1977. Detection of Influential Observation in Linear Regression. *Technometrics* 19, 1 (1977), 15–18. https://doi.org/10.1080/00401706.1977.10489493 arXiv:https://doi.org/10.1080/00401706.1977.10489493

[7] Peter Flom. [n.d.]. *The Disadvantages of Linear Regression.* Retrieved April 11, 2020 from https://sciencing.com/disadvantages-linear-regression-8562780.html

[8] Yoav Freund and Robert E. Schapire. 1996. Schapire R: Experiments with a new boosting algorithm. In *In: Thirteenth International Conference on ML*. 148–156.

[9] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001), 1189–1232.

[10] Arthur E. Hoerl and Robert W. Kennard. 2000. Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics* 42, 1 (Feb. 2000), 80–86. https://doi.org/10.2307/1271436

[11] kaggle.com. 2020. House Prices: Advanced Regression Techniques. Retrieved April 9, 2020 from https://www.kaggle.com/c/house-prices-advanced-regression-techniques

[12] Pedro Marcelino. 2019. Comprehensive data exploration with Python. Retrieved April 9, 2020 from https://www.kaggle.com/pmarcelino/comprehensive-data-exploration-with-python

[13] Didrik Nielsen. 2016. *Tree boosting with xgboost-why does xgboost win" every" machine learning competition?* Master's thesis. NTNU.

[14] @OfirChakon. [n.d.]. *Practical machine learning: Ridge Regression vs. Lasso.* Retrieved April 11, 2020 from https://hackernoon.com/practical-machine-learning-ridge-regression-vs-lasso-a00326371ece

[15] Alexandru Papiu. 2017. Regularized Linear Models. Retrieved April 9, 2020 from https://www.kaggle.com/apapiu/regularized-linear-models

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python . *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[17] NCSS Statistical Software. ND. *Ridge Regression.* Retrieved April 9, 2020 from https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Ridge_Regression.pdf

[18] Robert Tibshirani. 1996. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* 58, 1 (1996), 267–288. http://www.jstor.org/stable/2346178

[19] Rob Tibshirani. 2006. *Regularization: Ridge Regression and the LASSO.* Retrieved April 9, 2020 from http://statweb.stanford.edu/~tibs/sta305files/Rudyregularization.pdf

[20] Hui Zou and Trevor Hastie. 2005. Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 67, 2 (2005), 301–320. http://www.jstor.org/stable/3647580