STAT 444 FINAL PROJECT PAPER

BY ANGELO CARREON [1,*]

[1] *Department of Statistics and Actuarial Science, University of Waterloo,* [*] *jaccarre@uwaterloo.ca*

This paper answers the question: which regression method best predicts the prices of houses given their characteristics? To do this, four different regression techniques were fitted to the Ames Housing dataset, a popular collection of house sales that was prepared for an end-of-semseter regression project. This dataset was processed to address missing values, perform dimensionality reduction, and remove outliers. A multiple linear regression, ridge regression, lasso regression, and generalized additive model were then fitted to this processed data and their root mean squared errors were compared using cross-validation. It was found that a Generalized Additive model best fit the data, due to the non-linear relationships present in the covariates.

1. Introduction to our chosen dataset. Our project aims to assess the feasibility of utilizing regression techniques for interpolating and modeling housing prices. We have selected a dataset of housing prices in Ames, Iowa, along with their features from the *Journal of Statistics Education* (Cock, 2011). This dataset was prepared by Dean De Cock for use as an end-of-semester regression project. His intent was to provide data of substantial size ($n = 2930$) with easy-to-understand variables that are known to affect the final sale price such as build date, lot size, and living space square footage. Applying this to today's real estate market, we wish to answer the following question: *which regression method best predicts the prices of houses given their characteristics*?

2. Exploratory Data Analysis. This dataset contains 2930 rows and 80 columns. There are 80 explanatory variables, consisting of 23 nominal, 23 ordinal, 14 discrete, and 20 continuous variables. Several columns contain many missing values and will be dropped before we begin fitting models. As highlighted in De Cock's paper, several unusual outlier house sales exist in the data; these will also be removed. There are no duplicate rows.

As seen in figure 1, the distribution of `Sale Price` is significantly right-skewed. The sale prices range from \$12,789 to \$755,000 with a mean of \$180,796 and a standard deviation of \$79,886.69. To achieve a more normal distribution, we can apply a log transformation on the dependent variable, as seen below:

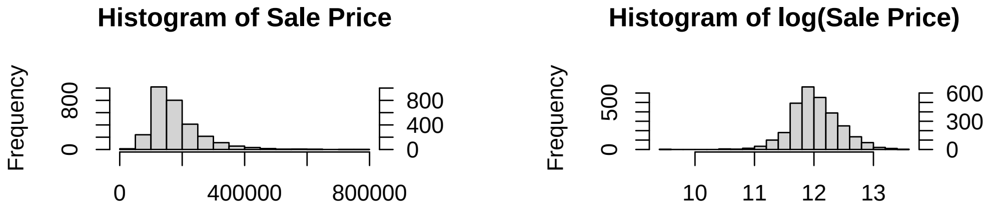


FIGURE 1. *Histograms of the response variable, Sale Price*

Keywords and phrases: housing, advanced regression.

Some variables of interest are `Neighbourhood` and `Lot.Area`, where we see in figure 2 to have significant differences in the average property sale price. One way in which these neighborhoods could differ is in the size of the lots of the houses that reside there. Our team would need to consider variable associations such as these to deal with collinearity.

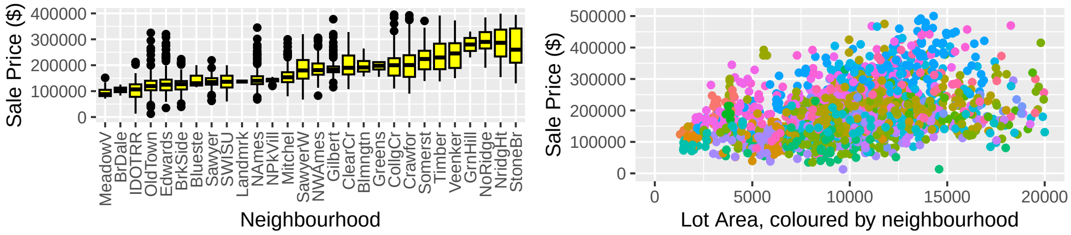
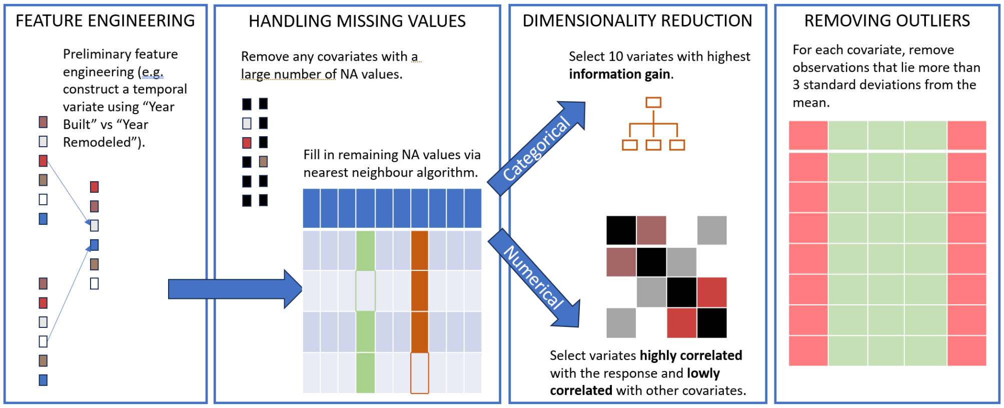


FIGURE 2. *Plots showing the distribution of final Sale Price across the different Neighborhoods*

3. Methods. There were three steps in the process of determining the most effective regression model for predicting house prices: data preprocessing, model fitting, and cross-validation. Preprocessing was used to clean the data, upon which models were fitted and evaluated against each other using a cross-validation scheme.

3.1. *Data Preprocessing and Dimensionality Reduction.* Below is the pipeline we used to process the data:



Preliminary feature engineering (e.g. construct a temporal variate using “Year Built” vs “Year Remodeled”).

Remove any covariates with a large number of NA values.

Fill in remaining NA values via nearest neighbour algorithm.

Select 10 variates with highest **information gain**.

Select variates **highly correlated** with the response and **lowly correlated** with other covariates.

For each covariate, remove observations that lie more than 3 standard deviations from the mean.

FIGURE 3. *Ames Housing data processing pipeline*

- First, we did some feature engineering in order to differentiate the `Year.Built` and `Year.Remodeled` covariates. We replaced the `Year.Remodeled` covariate with the difference in years between renovation and construction, so that the rest of our processing steps can correctly differentiate between these two covariates.
- Then, we addressed missing values in the data by removing covariates that had majority null values. A nearest neighbor algorithm was used on the remaining covariates to impute the remaining missing values.
- Next, we employed dimensionality reduction, reducing the number of covariates to 19.

  - For categorical variables, we selected the top 10 that had the highest information gain. The specific calculation of this score is defined in Quinlan (1986). In summary, information gain measures the reduction in uncertainty about the target variable when the data is split based on a specific feature, helping decision tree algorithms identify the most valuable features for prediction.
  - For numeric variables, one covariate from each group of highly correlated features was kept, and any covariates that showed weak linear relationships were removed.
- In order for our models to correctly interpret the categorical variables, their levels were factorized and converted into numeric values. Some categorical variables were ratings ranging from poor $\rightarrow$ excellent. These were converted to ordinal covariates, whose values increase as the rating increases.
- Lastly, to address outliers, we removed any observations that had a covariate whose values lie more than 3 standard deviations from the mean.

After the data preprocessing step, we were left with 19 covariates: 5 nominal, 5 ordinal, 5 discrete, and 4 continuous. Before any models were fit, the data was partitioned into training and testing splits at an 80:20 ratio.

3.2. *Model Fitting.* Four regression techniques were compared in our analysis: Multiple Linear Regression, Ridge Regression, Lasso Regression, and a Generalized Additive Model.

- The MLR model was fit using R's built-in `lm` object, which solves Ordinary Least Squares:

$$\min_{\beta} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2$$

- Both the Ridge and Lasso regression models were fit using the `glmnet` package, which uses cyclical coordinate descent to efficiently solve

$$\min_{\beta} \frac{1}{2N} \sum_{i=1}^{N} (y_i - x_i^T \beta)^2 + \lambda[(1-\alpha)||\beta||_2^2/2 + \alpha||\beta||_1]$$

where $0 \leq \alpha \leq 1$ is the elastic net penalty and $\lambda \geq 0$ controls its strength (Friedman, Hastie and Tibshirani, 2010). $\alpha = 0$ was used to obtain a ridge penalty, and $\alpha = 1$ was used to obtain lasso penalty. For both ridge and lasso regression, the optimal λ value was chosen to minimize mean-squared error via grid search under a 10-fold cross-validation scheme. An example of the output of the hyperparameter fit of λ for both ridge and lasso regression using the training set is below:

TABLE 1
Sample output of lambda that minimizes MSE under Cross Validation

	Selected Lambda	Index	MSE	SE
Lasso	0.001382	60	0.02186	0.002074
Ridge	0.033450	100	0.02213	0.001429

- A GAM was fitted using `mgcv`. A smooth was created for each covariate using the default *thin plate regression spline*, which minimize

$$||y - f||^2 + \lambda J_{md}(f)$$

where y is the vector of y_i data, $f = (f(x_1), f(x_2), \ldots, f(x_n))$, $J_{md}(f)$ is a penalty function that measures the “wiggliness” of f, and λ is the hyperparameter that controls the trade-off between data fitting and smoothness of f. The “wiggliness” penalty is defined in Wood (2003).

Smooths for continuous covariates were generated automatically, but categorical/ordinal variables required k (the dimension of the basis used to represent the smooth term) to be explicitly set to the number of categories in the covariate. This was required since the categorical variables often had unique values less than the default degrees of freedom $k = 10$ set by `mgcv`, and attempting to construct the smooth would result in an error.

As seen in figure 4, ordinality is captured fairly well by the smooths (usually trends upwards), but unordered categorical variables such as `Neighborhood` must be interpreted with caution, as the trends in the smooth are essentially meaningless - we are only interested in the predicted value at a specific neighbourhood.

3.3. *Cross-validation.* These models were assessed using a 20-fold cross-validation scheme, and the primary metric used to compare the different models was RMSE:

$$CV(\hat{f}) = \frac{1}{N}\sum_{i=1}^{N} L\left(y_i, \hat{f}^{-k(i)}(x_i)\right), \text{where } L = RMSE = \sqrt{\sum_{i=1}^{n}\frac{(\hat{y}_i - y_i)^2}{n}}$$

defined in Hastie, Tibshirani and Friedman (2009). This metric was chosen due to its interpretability, and due to the fact that the number of covariates were preprocessed at the beginning rather than during each fold. The reason for this decision was because we were unable to develop a way to run the data preprocessing steps within each fold, as creating the GAM model with custom combinations of smooths proved difficult. This meant that metrics such as AIC, R^2, etc. were not needed to generate and evaluate bias-variance trade-off under different combinations of covariates; we were only interested in the raw predictive power of traditional linear regressions vs. additive models on the final subset of covariates in the preprocessed data. Hence, RMSE was used to evaluate how close a model’s prediction was to the actual value.

4. Results.

4.1. *Final fitted models.* The Cross Validation and Test RMSEs for the models are below:

TABLE 2
RMSEs for the fitted models

	Average Cross-Validation RMSE	Test RMSE
MLR	0.1483174	0.2160399
Lasso	0.1537585	0.2106195
Ridge	0.1554260	0.2105215
GAM	0.1426394	0.1803288

We observe that the CV RMSE appears to be substantially lower than the testing RMSE. In ESL section 7.10.2, Hastie explains that our method of preprocessing the data and then fitting the models results in information leak, as the subset of predictors were chosen on the basis of *all of the samples* (Hastie, Tibshirani and Friedman, 2009). Hastie goes on to say lots of high ranking journals make this same mistake, so at least we have something in common! In the end, CV scores seemed to be fairly representative of the test scores, and we can attribute any differences in generalization error to the information leak.

In the end, the GAM approach outperformed the traditional linear regression methods in both CV and Test scores, so it was chosen as the final fitted model. To understand why, we examine the generated plots of the smooths:

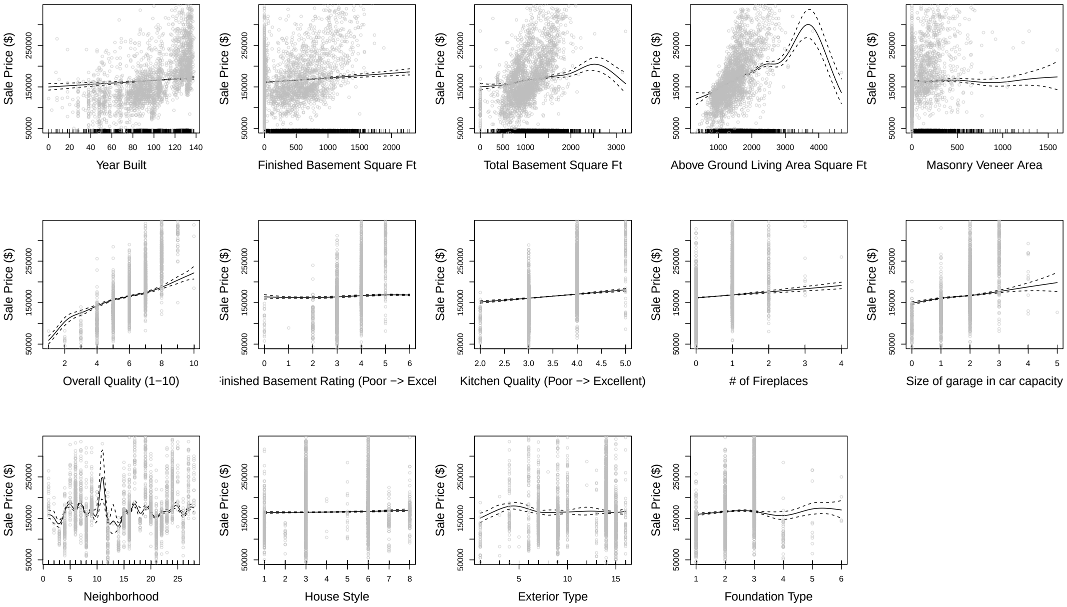


FIGURE 4. *Selected smooths plotted against residuals. Row 1 displays continuous covariates, row 2 displays ordinal/discrete covariates, and row 3 displays categorical covariates.*

We clearly see that there are lots of non-linear trends in several continuous and ordinal covariates that the GAM handles gracefully. This is likely why the GAM outperformed the linear regression models, and best predicted the prices of houses given their characteristics. Since Lasso, Ridge, and MLR require that the predictors are linear in order to perform the best, they struggle to fit the least-squares solutions and see higher RMSE scores. Moreover, Lasso and Ridge approaches only serve to deal with multicollinearity, not nonlinearity. Ridge shrinks correlated predictors towards each other, whereas Lasso selects one correlated predictor and ignores the rest (Friedman, Hastie and Tibshirani, 2010). Since most of the multicollinearity was addressed in the data preprocessing step, it makes sense that the MLR, Ridge, and Lasso RMSEs are close to one another.

4.2. *Model Assumptions.* The following graphs depict the performance of the final model on the testing set, fitted using the training set.

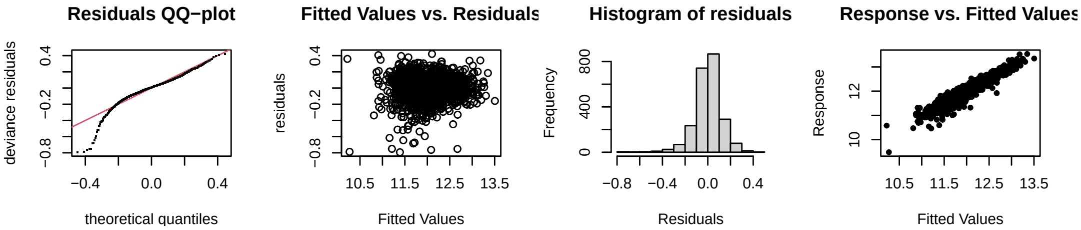


FIGURE 5. *Residual QQ-plot, Residual histogram, and Fitted Values vs. Residuals plot*

We can see that the normality assumption appears to be violated, as the tails on the left side of the QQ-plot appear to be quite heavy. Taking a look at the histogram of residuals, we can see that the model appears to be undershooting sale prices when the true value is large. This may be the result of the GAM function smoothing away the wiggliness that might be required at the tail ends of the sale price values, or outliers in the data that our preprocessing steps failed to recognize. Thus we have to be careful when the predicted values lie in the range of the tails of the sale price values, since we can visually see that there are large amounts of error at these regions.

5. Conclusions. In conclusion, we find that a generalized additive model produced the best predictions of house sale prices given its physical characteristics, with a RMSE of $\$26338.29$, which is acceptable given the sale price ranges from $\$12,789$ to $\$755,000$ with mean $\$180,796$ and standard deviation $\$79,886.69$. This model was able to correctly capture the non-linearity in the data, and consistently produce predictions with the lowest RMSE score in both the cross-validation and testing scenarios. Although there was evidence of information leak due to the data preprocessing being done before cross-validation, the GAM still outperformed traditional regression methods in the test scenario.

One limitation of this analysis was that, in retrospect, of course the GAM would fit nonlinear data better than the linear regression models. However, the benefit of using a GAM is that much less time and effort was required to preprocess the data. While are many ways to transform the covariates such that they are linear with the response and produce better predicted values under linear models, these methods can be computationally intensive and sacrifice interpretability. Simply fitting a GAM allowed a group of undergraduate students to quickly generate accurate predictions without worrying too much about the usual linear regression assumptions.

Therefore, given the limited resources our group had to generate these regression models, the GAM was the best choice for quickly producing accurate predictions. In the future, it would be worthwhile to compare a GAM against linear models that better address the nonlinearity, to make a more fair comparison. It would also be interesting to see how this model performs on a more recent dataset, as these housing prices are quite low for today's standards.

REFERENCES

COCK, D. D. (2011). Ames, Iowa: Alternative to the Boston Housing Data as an End of Semester Regression Project. *Journal of Statistics Education* **19**.

FRIEDMAN, J. H., HASTIE, T. and TIBSHIRANI, R. (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software* **33** 1–22.

HASTIE, T., TIBSHIRANI, R. and FRIEDMAN, J. (2009). *The elements of statistical learning: data mining, inference and prediction*, 2 ed. Springer.

QUINLAN, J. R. (1986). Induction of Decision Trees. *Machine Learning* **1** 81–106.

WOOD, S. N. (2003). Thin Plate Regression Splines. *Journal of the Royal Statistical Society Series B: Statistical Methodology* **65** 95-114.