Machine Learning Predictive Modeling Competition

Blue team 12

Sophee Li, Chris Nobblitt, John Pamplin, Jerry Liao

Initial Modeling

We produced multiple models that managed to perform consistently as well as the baseline Mean Absolute Error. We tuned a lasso regression using 5-fold cross-validation and found that a value of alpha at 0.62 consistently performed at baseline MAE. At any alpha level between 0 and 1, the lasso regression would always select 4 variables: *num1, num30, num37, num59*. These four variables, range standardized, were used as the input to a 2-hidden layer feedforward neural network. All neurons were randomly initialized using a random normal distribution and ReLu activated with exception to the output neuron. To prevent overfitting, we utilized a 20% dropout rate in each hidden layer during training and weights were constrained to be no greater than 4. The target was also range standardized. This network performed very consistently at the baseline MAE with 5-fold cross-validation.

Transforming the Target Variable and Final Model

Taking a closer look at the data, we noticed that the target variable “target” was very leptokurtic with a kurtosis value of 9.17. The mean of the target was 20.04 with a minimum value of 8.48, a maximum value of 32.85, and values for the 1st and 3rd quartiles were 19.45 and 20.68 respectively. We think the concentration of the values and lack of signal in the middle hindered us from getting better prediction because our model could not capture the values at the tails. To deal with this, a Lambert W function was used to normalize the target. While successful at helping to normalize the data, the transformation did not improve predictability of the target. Our next thought was to reduce the influence of the middle 50% of the target by removing target values between 19.45 and 20.68, the 1st and 3rd quartile. This provided a significantly more normal distribution for the target variable which can be seen in figure A1 and figure A2.

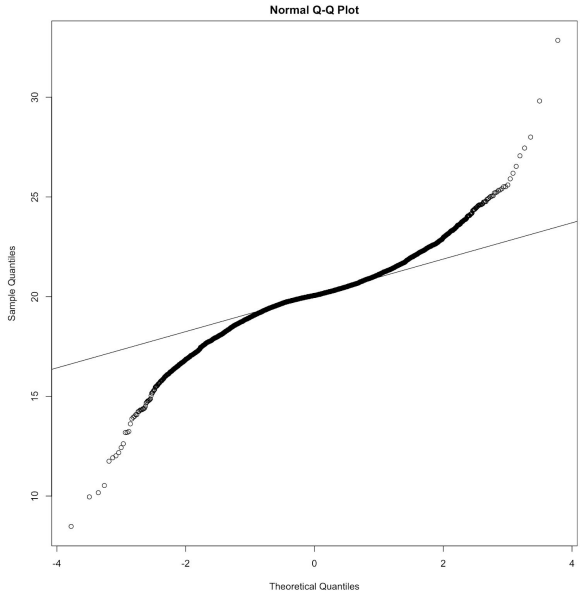

Figure A1. Original Target Data

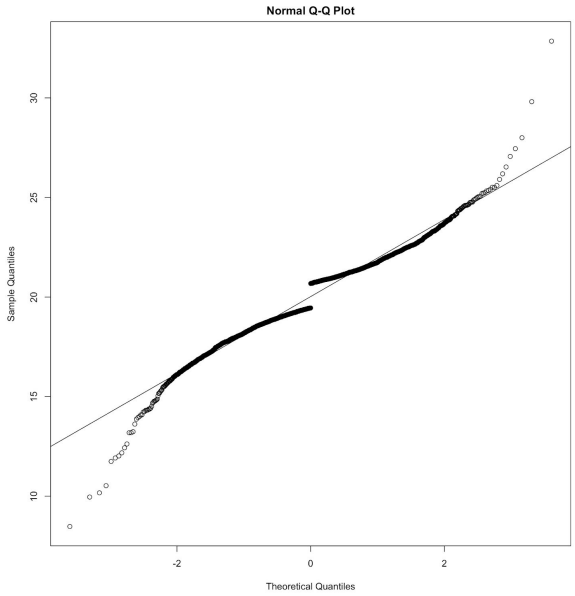

Figure A2. Middle 50% removed

The result of removing the middle 50% of the data is equivalent to weighting the tail values and underweighting the center of the distribution. Once this truncation was done the remaining 50% of the data was split into a test and training dataset with an 80-20 split. The training data was used to build the following models: Neural Network, Random Forest, Ridge Regression, Lasso Regression, AG Boost, SVM, and kNN. The random forest model with 26 trees, 10 attributes considered per split, and leafs with at least 2 observations performed the best on the test dataset and was the only model to continuously outperform the baseline mean estimate with an MAE of 1.563 compared to the baseline mean MAE of 1.597. To confirm the model was indeed better, the model was used to predict the full untruncated data set with a MAE of 0.379 compared to the baseline mean prediction with a MAE of 0.957. We believe this result is due to signal being present in the tails and not the middle of the data.

Recommendations

We believe that the method that we used to trim the middle of the data is not ideal and can be fine tuned. Keeping some or all of the middle 50% of the data and using a different weighting method might produce better results. We would recommend testing a series of methods for weighting the tails of the distribution with the random forest as the prediction model. We would also recommend reviewing the experimental setup since it appears that there is signal in the tails and not in the middle of the data. It may be that the experiment that produces this data is biased and forces data without signal towards the target mean.