Ashtin Cheng
Daniel Meier
Ziv Schwartz

Stats 101C Final Project
Team DAZ

Abstract

For this assignment we were given a large dataset from the Los Angeles Fire Department. With these data we were tasked to predict the elapsed time it takes for an emergency responder to arrive after being contacted. The training dataset includes the following variables: Row ID, Incident ID, Year, First in District, Emergency Dispatch Code, Dispatch Sequence, Dispatch Status, Unit Type, PPE Level, Incident Creation Time, and Elapsed Time. We were also given a testing dataset, with the elapsed time (what we are predicting) removed. We fit several models using this training dataset and checked daily how well each model was performing by submitting our predictions for the testing dataset on Kaggle.

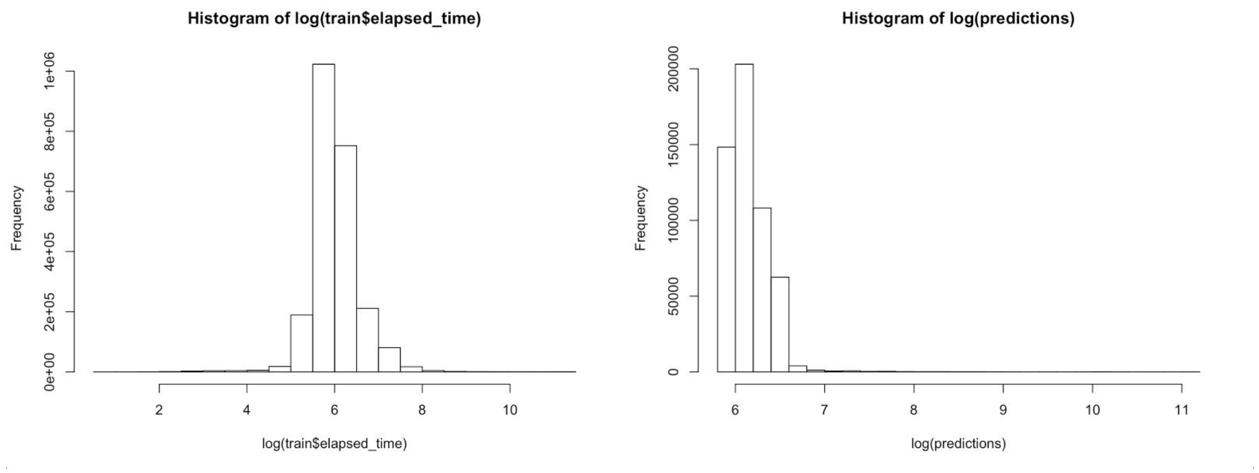
Data Preparation

Our group noticed that there were several missing values in both of the LAFD training and testing datasets. To address this, we determined how many N/A's there were in each column and for our training dataset we found the mean of the known values of that column and filled in the N/A's with that value. For the missing values (all of which were under Dispatch Sequence) in our testing dataset we used the R package GBM, to boost and predict the missing values. We predicted the missing values using all the other predictors of testing data.

Best Model

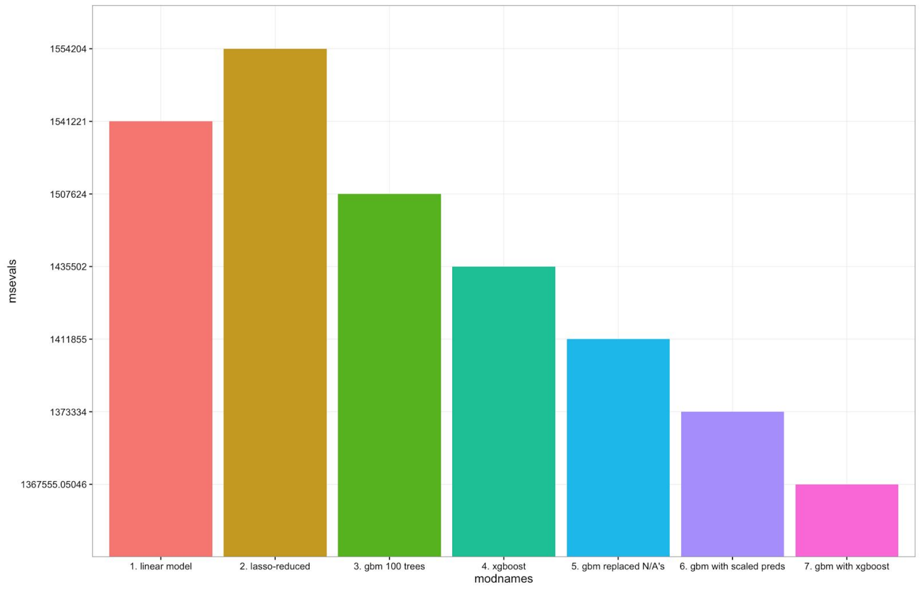
Our best model consists of averaging our predictions from a Generalized Boosted Regression Model and a XGBoost Model with the following variables: Year, First in District, Dispatch Sequence, Dispatch Status, Unit Type, and PPE Level. The variables Row ID, Incident ID, Emergency Dispatch Code, and Incident Creation Time were all excluded from our model. Row and Incident ID were removed because they are unique for each incident, Emergency Dispatch Code was removed because every incident had the same value, "Emergency." Incident Creation Time is solely the timestamp of when the incident was reported.

After creating the model for XGBoost and GBM, we examined the predicted values and compared their distribution to that of the training dataset "Elapsed Times." We noticed that the predicted values were not as skewed as the training data, so we decided to scale the predicted values above 10,000 by a factor of 1.6. We arrived at 1.6 by testing a few different scaling factors until our MSE was at its lowest. After this, we saw that our models were still over and under predicting values so we decided to combine our GBM and XGBoost models to further mimic the distribution of the training dataset.



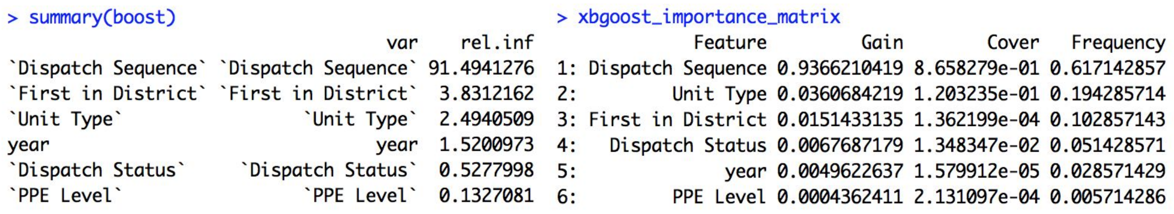
The above histogram shows the distributions of the response variable from our training dataset and the one on the right shows the distribution of our predicted response values for the testing dataset.

Before we reached our best model, we tested several other models. These models include: Simple Linear Regression, Lasso Regression, Generalized Boosted Regression Model (GBM) before scaling, Extreme Gradient Boosting (XGBoost), and GBM after scaling. *Below represents the testing MSE obtained from each.*



Best Mean Squared Error with Scaled GBM Combined with Scaled XGBoost

Our best Mean Squared Error with our Scaled GBM + XGB model was **1,367,555.05046.**



```
> summary(boost)                                          > xbgoost_importance_matrix
                                        var    rel.inf               Feature         Gain        Cover   Frequency
`Dispatch Sequence` `Dispatch Sequence` 91.4941276  1: Dispatch Sequence 0.9366210419 8.658279e-01 0.617142857
`First in District` `First in District`  3.8312162  2:         Unit Type 0.0360684219 1.203235e-01 0.194285714
`Unit Type`                 `Unit Type`  2.4940509  3: First in District 0.0151433135 1.362199e-04 0.102857143
year                               year  1.5200973  4:   Dispatch Status 0.0067687179 1.348347e-02 0.051428571
`Dispatch Status`     `Dispatch Status`  0.5277998  5:              year 0.0049622637 1.579912e-05 0.028571429
`PPE Level`                 `PPE Level`  0.1327081  6:         PPE Level 0.0004362411 2.131097e-04 0.005714286
```

The summaries above represent the relative importance of each factor in the boosted model. The left output shows the significant predictors from the GBM model while the right output shows that of the XGBoost model. Dispatch Sequence seems to be the most significant predictor in both boosting models.

Why Boosting Models Works

Boosting allows for us to grow trees while having them be dependent on the previous tree. This allows us to build an algorithm which "boosts" weak learners to strong ones. Because of this, we decided to use a combination of the GBM and XGBoost packages within R. We chose the parameters of our model based on suggestions from our textbook and from the professor. We then further tuned our GBM and XGBoost models and their parameters based on the MSE performance it had on the Kaggle website. At first, our model severely overpredicted elapsed time values. Because of this we decided to test some different parameters to prevent overfitting. For example, in our gbm model, to do this we increased our shrinkage value all the way up to 0.2, interaction depth to 4, and our number of trees to 27, where we saw that our model was not overfitting anymore. While in our xgboost model, we changed max depth to 3, minimum child weight to 8, and number of rounds to 25.