Statistics101C Final Report

[Winning]

Victor Shih(kyleshihv@gmail.com)

Instructed by: Akram Almohalwas

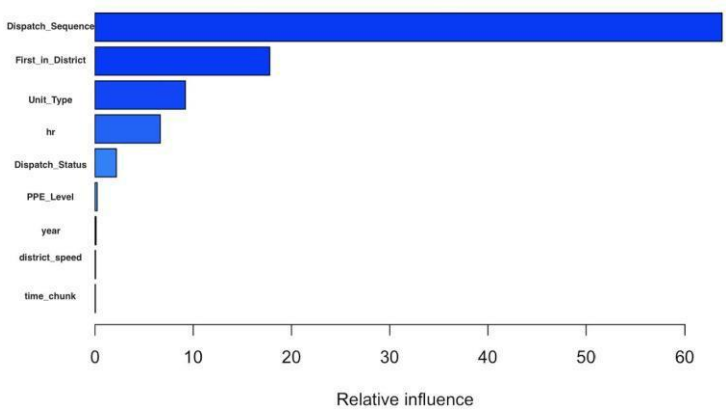
Spring 2017

1. Data Cleaning and Transformations

- **Data Cleaning:** We deleted the variable “Emergency Dispatch Code” from the dataset, because it only had one level, Emergency. We also removed the variable “incident.ID”, since it was just a useless random ID number. There were missing values in both the training dataset and the testing dataset. Alongside, since most NAs in the training dataset were contained in the response variable, we simply deleted those observations. After removing NAs in the training data, there was only around 0.1% of data containing missing values for predictors. We removed those observations because of the low percentage. For the testing data, there are missing values in the predictor “Dispatch Sequence”. We used boosting algorithm to predict the dispatch sequence for these observations.
- **Data Transformation:** We created a new categorical variable called “hr” by extracting the hour of the incident from the predictor “Incident Creation Time (GMT)”. After grouping the average elapsed time by hour, we created a new predictor called “time_chunk”, which classifies the observations into one of the four levels: fastest, fast, slow or slowest. We also changed the predictor “year” and “First in District” into categorical variables. By grouping the median elapsed time by “First_in_District”(the fire station), we created “district_speed” which ranked each observation with a numerical score from 1 to 10, indicating the speed of that station. We took the log transformation on the predictor “Dispatch_sequence”, since it represented the “waiting time” of vehicles to go to incident places and may follow an exponential distribution. For features “Dispatch Status”,“Unit Type” and “PPE Level” and “row.id”, we just leaved them as they are.

2. Our Model

```
bst3<-gbm(elapsed_time~.-row.id, interaction.depth = 3,
data=l[-idna,],distribution="gaussian",n.trees =300,verbose =F, shrinkage =0.15)
```



We used a boost model with the 9 predictors mentioned above. We tuned the parameters manually by creating a testing set of the actual size of the testing data and tried for almost 20 possible models. The model “bst3” with a shrinkage

	var	rel.inf
Dispatch_Sequence	Dispatch_Sequence	63.79636340
First_in_District	First_in_District	17.78617035
Unit_Type	Unit_Type	9.19992917
hr	hr	6.63489763
Dispatch_Status	Dispatch_Status	2.18093911
PPE_Level	PPE_Level	0.21522791
year	year	0.10084843
district_speed	district_speed	0.04839488
time_chunk	time_chunk	0.03722912

value of 0.15, 300 trees and an interaction depth of 3. Together this usually produced the smallest MSE for the testing set we separated from the training data. After realizing bst3 was the best model, we used all of the training data to fit this model, thereby producing the final model shown above. The result on the left is the relative importance of our predictors for the model. As is seen, “Dispatch_Sequence” is the most important predictor with a relative influence of 64%.

3. Best MSE

Our best score on Kaggle is 1392287.13605.

Please note that our team name in Kaggle is cyc00, NOT Winning, because our team members could not submit results using the name Winning

4. Strengths and Weaknesses of the Model

- **Strengths:** Compared to other models that we tried(PCA, PLS, random forest, tree, Lasso, and ridge), the gradient boosting model had the highest predictive power, since it minimized the MSE. Furthermore, it is a relatively computational and inexpensive in comparison to random forest, making it possible for us to use all of the training data to build and tune our model. The boosting model also avoids overfitting, since we tuned the parameter shrinkage, creating a penalty for less useful predictors. Furthermore, the boosting model allows us to use categorical predictors with lots of levels. Therefore, we avoided losing information by reducing the number of levels in predictors (i.e. replacing levels of very low frequency with levels of high frequency).
- **Weaknesses**: We searched online for possible external data, such as the location of the fire station and the average response time for each station. But it turned out none of them are helpful in terms of decreasing MSE. If we could find any useful external information, the predictive power would have increased. Furthermore, we tuned the parameters manually by trying some possible combinations. (Because it would take a very long time to use package caret to find the best combination) Due to the limited number of combinations we tried, the combination we used for parameters may be a set of values that just produced a “local minimum” MSE. If we

actually tuned the parameters using functions in caret, we may have gotten better results.