Random Forest Fires

Zachary Meeks
Kaylin Dee

Variables Used

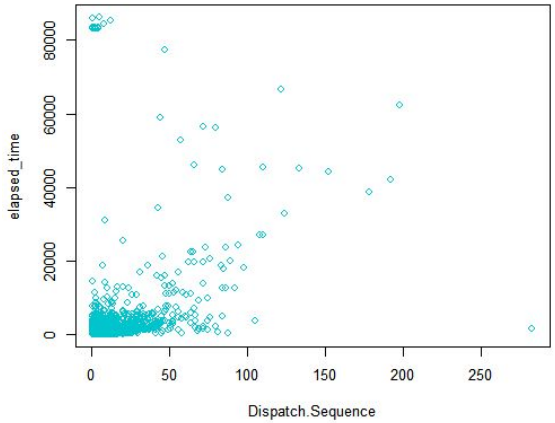
- Dispatch Sequence
- Dispatch Status
- Unit Type
- Incident Creation Time
  - Transformed to seconds → 18 categories
    (via as.numeric(), dividing by 4800, floor() function, and then casting to character)

Response: Elapsed Time

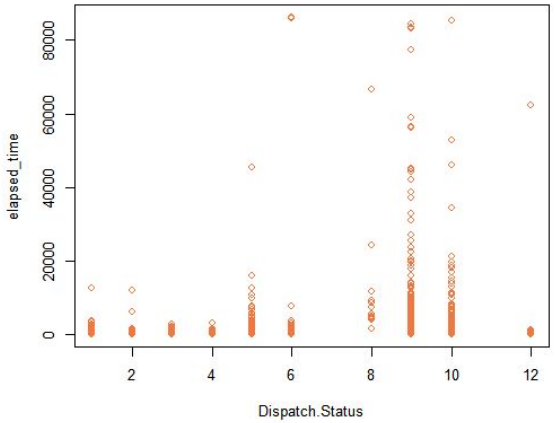
Plots of Variable-interactions (EDA)

(1) Dispatch Sequence
(2) Dispatch Status (as numerical)
(3) Unit Type (as numerical)
(4) Incident Creation Time

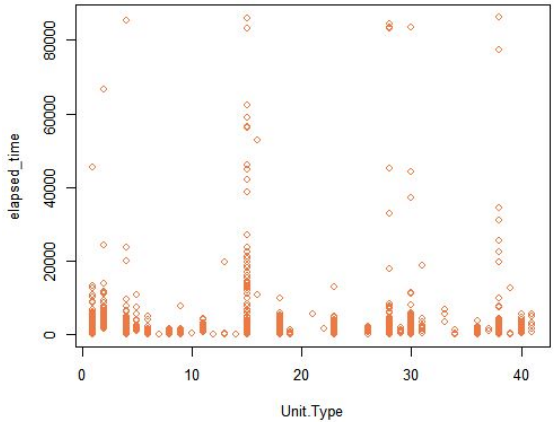
1



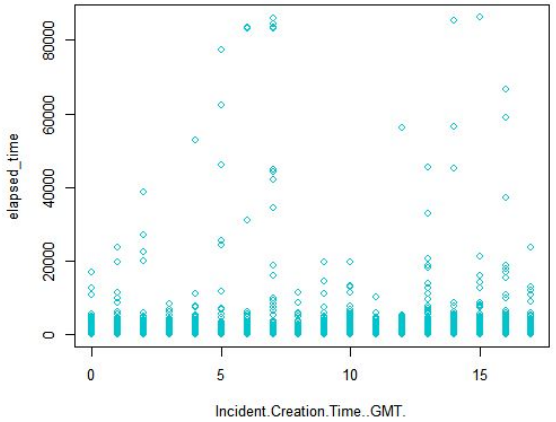
2



3



4



Cleaning the data

- Training data had over 400 dispatch sequence levels
  - Kaggle data only had 155
- Took out responses (elapsed time) over 75600
- Omitted the rest of the NA values

The Model

- Used XGBoost package
  - Linear model solver and tree learning algorithms
  - Parallel computation → 10 times faster than gradient boosting techniques
- Parameters:
  - Eta: step size shrinkage
  - Gamma: minimal loss reduction required to make a further partition on a leaf node
    - The larger, the more conservative
  - Max_Depth: max depth of a tree
  - Nround: number of trees
  - Subsample: subsample ratio of training
  - Colsample_bytree: subsample ratio of columns when creating a tree
  - Seed: set seed
  - Eval_metric: evaluation metrics for validation data (regression MSE in our case)
  - Nthread: number of threads used

Preparing the data for XGBoost

- 70% testing, 30% training
- Used full data for kaggle submission
- Had to convert the matrices for testing, training, and full data to Sparse Model Matrices
    - Most values are 0

-

Building the Model

```
xgb <- xgboost(data = full_matrix,
          label = labels,
         ta = 0.0173,
         gamma = 0.2,
         max_depth = 3,
         nround=185,
         subsample = 0.425,
         colsample_bytree = 0.5,
         seed = 17,
         eval_metric = "rmse",
         nthread = 2)
```

1400735.87

Resulting MSE (pre kaggle final holdout score)

How to improve

Use a scripted grid search instead of manual hill-climbing search.

Exploratory analysis suggests that factor 10's signal beneath the noise is with respect to very notably higher elapsed_time around 10am. However, factor engineering around 10am didn't lead to significant decreases in mse. This may be due to other factors with more categories playing a magnified role in the xgb algorithm such that overfitting was still at play. In other words, more factor engineering could be done. Could use random forest to verify.

In preliminary models we made gains by ensembling mildly different models. It would behoove us to engineer models that can make use of ensembling tricks

Use 5 fold CV instead of 70/30 split validation

Should have spent more time analyzing the data of our results so as to understand the data better so as to make bigger gains faster.