Summary

**80 points.** Use Weka, R, or Python on this dataset to find a lower error rate. You may want to try the packages xgboost and nnet. The lower the error, the higher the mark.

My mean absolute error: **1148.4568** using xgboost

I install the following libraries in R:



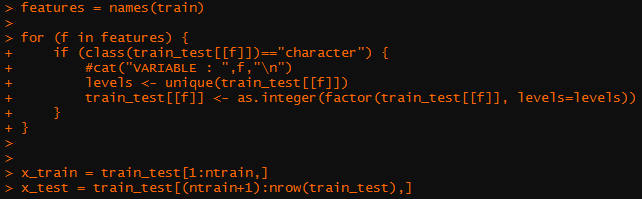
* I used data.table for fast aggregation of large data.
* Matrix for classes and methods for dense and sparse matrices and operations.
* xgboost which is an efficient implementation of gradient boosting framework. It includes efficient linear model solver and tree learning algorithms. The package can automatically do parallel computation on a single machine which could be more than 10 times faster than existing gradient boosting packages.
* Metrics a set of evaluation metrics that is commonly used in supervised machine learning.

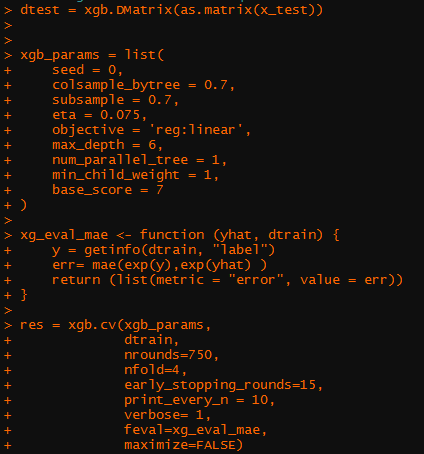
First, I read my csv files.



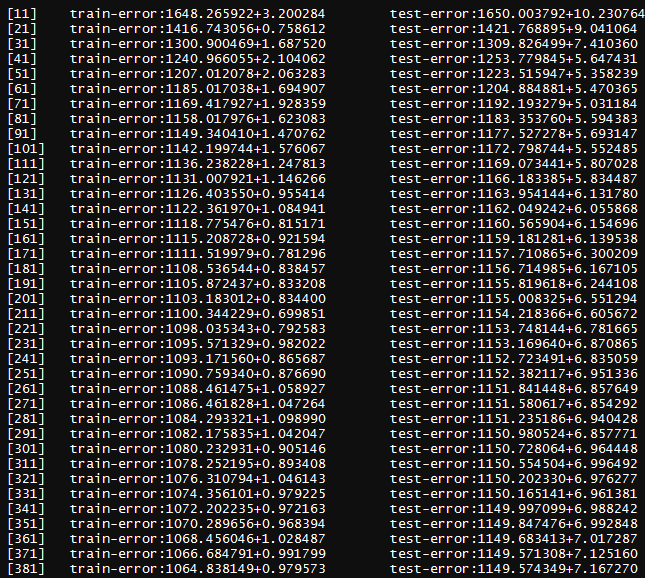
The allstate\_train.csv file is the original allstate\_1000\_131 file that is already provided in the course website. While the allstate\_test.csv file is my own file. Using WEKA, I created my test file by removing some attributes. I selected CfsSubsetEval for the attribute evaluator and best first for searching method. In my case the selected attributes total is 26. Then I saved the file as .csv file.

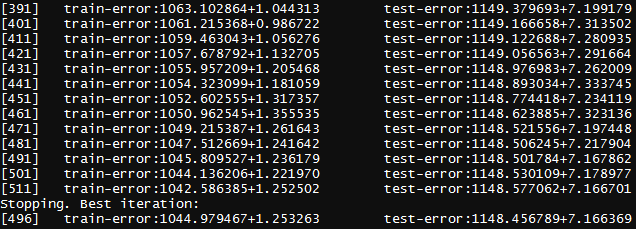
Then I write the following code to start the analysis.



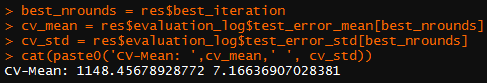


The output of the code is as follow:





Then I write another block of codes to get the mean absolute error.



My mean absolute error is now **1148.45678928772** or **1148.4568**

**20 points.** List 5 lessons that you have learned from this assignment.

* I learned that XGBoost is an implementation of gradient boosted decision trees designed for speed and performance.
* Execution speed and model performance are the main reasons why a lot of people uses this model.
* XGBoost dominates structured or tabular datasets on classification and regression predictive modeling problems.
* XGBoost is fast because of its capacity to do parallel computation on a single machine. This makes xgboost at least 10 times faster than existing gradient boosting implementations. It supports various objective functions, including regression, classification and ranking.
* XGBoost has additional features for doing cross validation and finding important variables.