Text Analytics using High Performance SAS® & SAS® Enterprise Miner™

Edward R. Jones, Ph.D.
Texas A&M Statistical Services, LP

Abstract: The latest release of SAS® Enterprise Miner™, version 13.1, contains high performance modules, including new modules for text mining. This paper compares the new High Performance Text Mining modules to those found in SAS Text Miner. The advantages and disadvantages of HP Text Miner are discussed. This is illustrated using data from the National Highway and Transportation Safety Administration for the GMC recall of 258,000 SUVs for a potential fire hazard .

INTRODUCTION

In August 2014, General Motors announced the third recall to correct a life-threatening hazard. The new recall was for six models: Cobalt's, Pontiac Solstices, Pontiac G5s, Saturn Sky's, Chevrolet HHRs and Saturn Ions. GM claimed 13 deaths were related to faulty ignition switches after reports of 28 fires in the power windows and locks of these vehicles from 2005-2007. The NHTSA, National Highway Traffic Safety Administration, maintains a public database of consumer complaints for these and other vehicles. The question is whether these complaints could have been used to identify this hazard earlier than 2014.

Over 1 million complaints have been registered and are available for download from the NTHSA. To examine this question, both SAS Text Miner and SAS HP Text Miner, version 13.1, were used to analyze over 11 thousand complaints recorded by the NTHSA for this latest recall. These data include not only written complaints but also information on the vehicle, the driver and whether or not the vehicle was involved in a fire, a crash, and the number of injuries and deaths. This paper describes how SAS can be used to analyze these data. The analysis is done using both the original text miner procedures and also the new SAS HP Text Miner. The purpose is to compare and contrast these procedures, and to illustrate the advantages of High Performance Text Miner.

This application illustrates the utility of incorporating unstructured data into a traditional analysis of structure data. SAS Enterprise Miner combined with the new SAS HP Text Miner allows analysts to not only discover what customers are complaining about but why.

SAS® TEXT MINER

SAS® Text Miner is incorporated within SAS® Enterprise Miner™ and shares the same syntax. However, now SAS analysts have two options for text mining, each with a different character:

SAS Text Miner and SAS High Performance (HP) Text Miner. The first option has been available for some time, but the second only became available with Versions 9.4 of SAS and 13.1 of SAS Enterprise Miner.

SAS Text Miner consists of seven nodes, but in most applications, five are used in the following order.

1. Text Import – reads text files stored in a single directory
2. Text Parsing – creates the Term/Document matrix
3. Text Filter – allows analysts to eliminate non-useful words
4. Text Topic – identifies document groups associated with topic words
5. Text Cluster – identifies document clusters from their word structures.

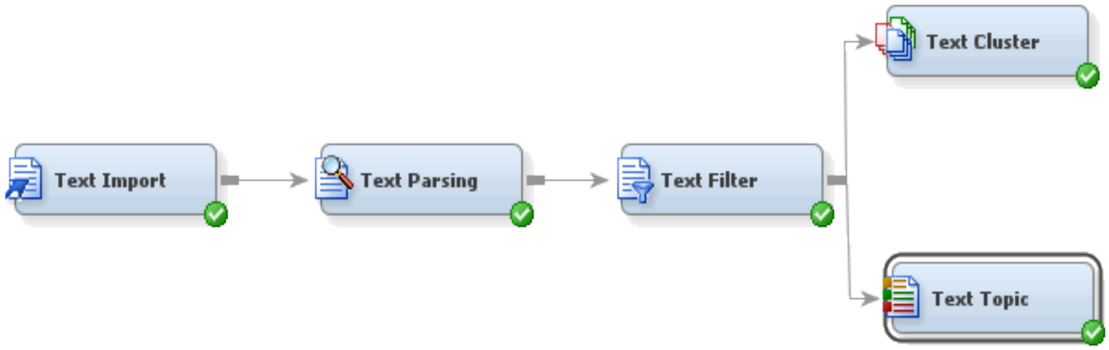


Figure 1. The Text Mining Process using SAS® Text Miner

The text import node requires all documents to be placed in individual files. Unfortunately, the NTHSA data file contained all 11,363 complaints in one file. These had to be extracted and placed into 11,363 files. The process of extracting individual comments and placing them into separate files was accomplished using a small Java file described at the end of this paper. The Java program reads a tab-delimited file containing the comment identifier and the comment. Each comment is written to a separate text file with a unique name. The name incorporates the comment ID, which makes it easy to associate individual comments with cluster and topic groupings from the text mining process.

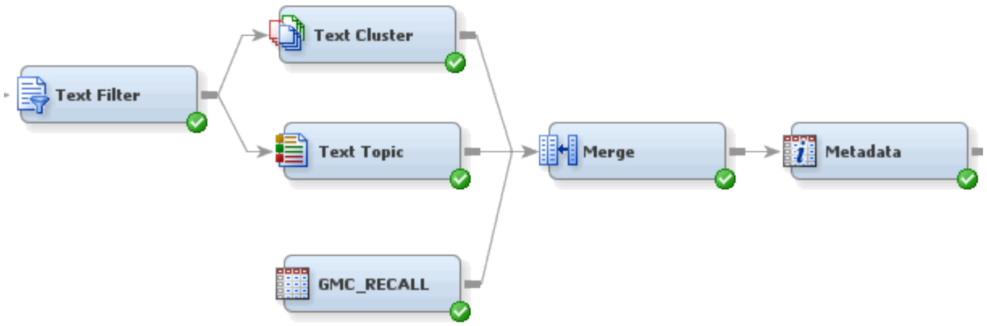
The analytics process of for text mining these complaints is described in Figure 1 above. The default document size in text import is 100 characters. This had to be increased to 2100 characters. Text parsing was done using the default settings. Text filtering was done using inverse document frequency term weights, and the filter viewer was used to review terms to identify those that were not useful in this analysis. This consisted mainly of the initials of the NTHSA recorder.

Using the default settings, the text cluster node identified 10 clusters and 37 SVD vectors. The text topic node was used iteratively to identify 9 user word clusters including one word cluster for 1,416 complaints about a key and ignition problem, 1,500 complaints about a brake problem, and 7 others for various mechanical problems such as air bag and steering problems.

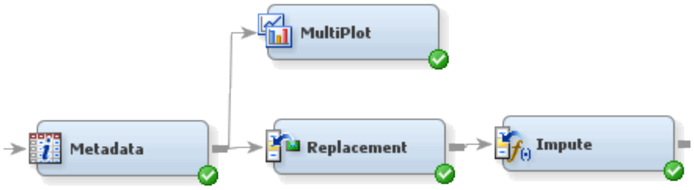
Table 1. Topic Groups Identified by Topic Node

Topic Group	Complaints	Description
1	1374	Air Bag Problem
2	1500	Brake Problem
3	996	Door Problem
4	903	Electrical
5	911	Fuel System
6	1416	Key & Ignition
7	1744	Recall
8	1435	Start Problem
9	1794	Steering Problem

The output from text mining process consisted of 9 binary attributes associated with the 9 user developed topics and 3 interval attributes consisting of the estimated probabilities for the three text clusters. These were merged into the original structured data to produce a single file for modeling whether fires in these vehicles might be related to some of these complaint topic groups or text clusters.



After the merge, the roles of all attributes were modified as either rejected, input or target. The text topic groups and text cluster estimated probabilities were all marked as inputs. The SEMMA process in Enterprise Miner was followed. After preparing this integrated file, outliers were set to missing using the replacement node, and missing values were imputed.



The model step was implemented using the logistic regression node, the decision tree node and neural network node. Experience with problems involving many input attributes has found it's prudent to reduce the number of input attributes using variable selection.

The logistic regression and decision tree models all had the same misclassification error rate for predicting whether the vehicle was involved in a crash based upon the topic of the complaint and the driver and vehicle characteristics. The logistic regression model developed using stepwise selection had the smallest Validation Average Squared Error (VASE). It incorporated only one of the input attributes: text topic 2, which was the topic related to a door switch problem. The GMC recall reported that the fires were related to shorts in the door switch.

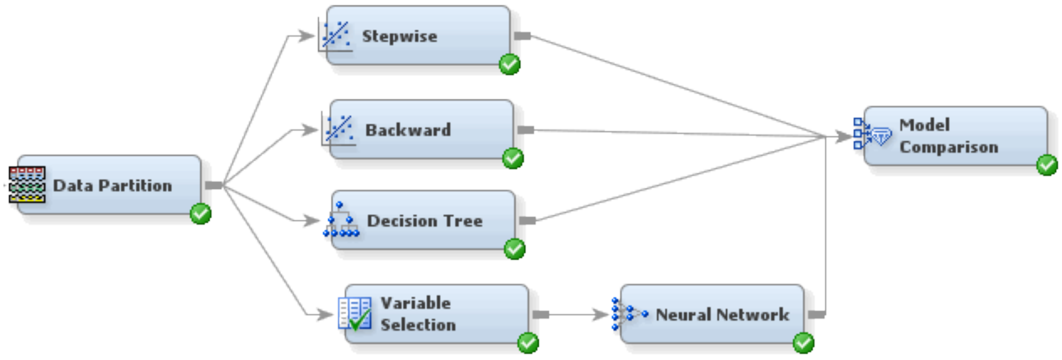
The complaints to point to problems related to the key & ignition problem. We now understand that this problem caused steering and braking problems related to vehicle crashes. Combining those three categories suggests the NHTSA received 4710 complaints related to the key and ignition problem. That's over 40% of all complaints for these vehicles.

SAS® HIGH PERFORMANCE TEXT MINER

SAS® High Performance Text Miner is also incorporated within SAS® Enterprise Miner™ since version 12.1. The latest version, 13.1 is an improvement and expansion over 12.1.   There are several features that differentiate High Performance Text Miner from the original version of SAS Text Miner. First, all High Performance nodes are designed to run faster by using high-speed algorithms and multi-core and cluster capabilities.

Second, the SVD process in the HP Text Miner not incorporates a rotation of the principal vectors from the SVD, which helps with the interpretation of topics and document groupings. This sets HP Text Miner above its competitors.

Another important difference in text mining is that unlike SAS Text Miner, High Performance Text Miner does not require all documents to be stored in individual files inside a single directory.
Instead, High Performance Text Miner will accept any SAS file that contains two attributes, one with the role ***key*** and a second

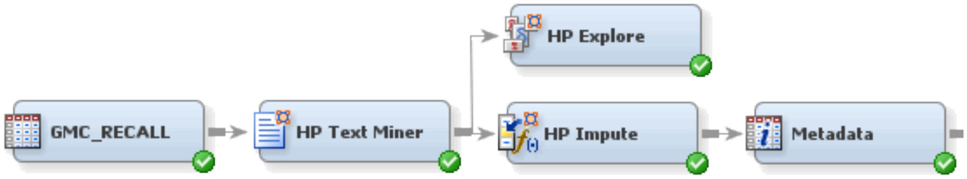


with the role ***text***. The key field should be a unique interval value associated with each of the text fields. This type of input is ideal for the NTHSA data, surveys and other data containing written comments.

In SAS 9.4, these features are implemented using `Proc HPTMINE` and Proc HPTMSCORE. These were described in the SAS references at the end of this paper.

In this example, the key field was the unique NTHSA ID assigned to each complaint and the text field was the written comment. In these data it was restricted to a maximum of 2100 characters, but High Performance Text Miner can accept written comments up to 32,000 characters.

The Sample-Explore-Modify process in SEMMA using High Performance Text Miner is much simpler than what was described for SAS Text Miner. First, there is no need to separate the text mining from the original data. In addition, at this time there is no equivalent to the Text Filter and Text Topic nodes. Instead, High Performance Text Miner by default exports topic groups defined by the SVD analysis. These will be orthogonal to one another.



Using the default setting for the High Performance Text Miner node produced 32 topic groups. These were marked automatically as ***interval inputs***. Version 13.1 of High Performance Enterprise Miner also contains a new ***Impute node*** that was used to estimate missing values, such as vehicle mileage.

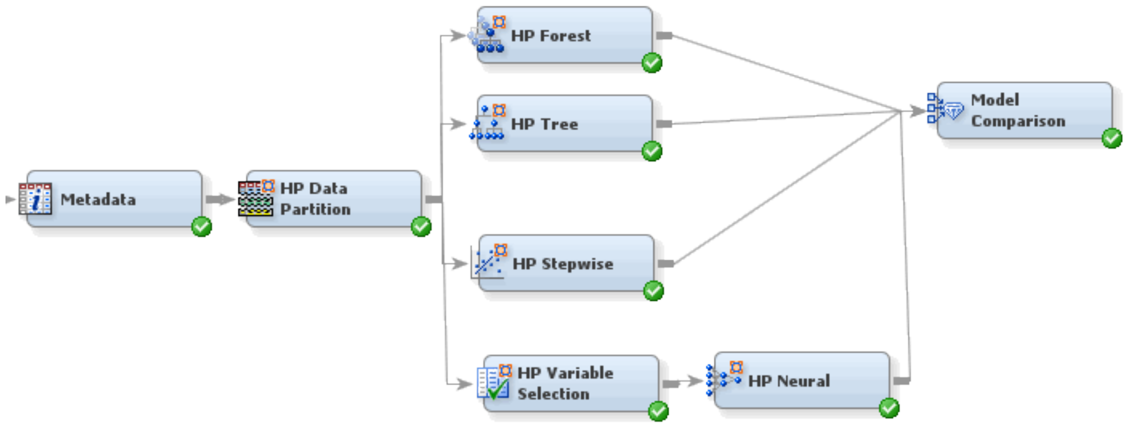
Table 2. Topic Groups Identified by HP Text Miner

Topic Group	Complaints	Description
1	1374	Deploy, bag, air bag, air, airbag
2	1500	Police, Report, Police Report, file, sustain
3	996	Accident, cause, sustain, result, avoid
4	903	Airbag, passenger, seat, side, seatbelt
5	911	Hit, cause, curb, curb, side, in front of
6	1416	Driver, side, side, passenger, roll
7	1744	Break, break, pedal, apply, stop
8	1435	mph, vehicle, approximately, fail, wheel
9	1794	Crash, pole, curb, tree, cause

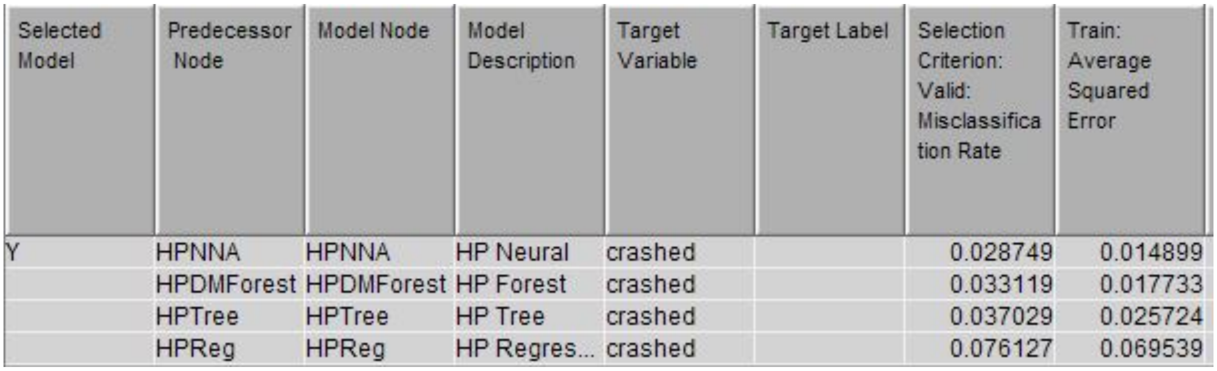
Notice that the topic attributes are named COL1…COL20, but High Performance Text Miner puts the names of the words associated with each attribute in its label. Notice that COL4 is associated with a comment about a vehicle fire. COL1 and COL4 are talking airbag deployment. None have an obvious relationship to keys or ignition.

The modeling approach is almost idential to that used with SAS Text Miner. Stepwise logistic regression is used, along with decision tree and neural network.

However there is one additional modeling node that was used this is only available in the High Performance nodes: High Performance Random Forests.



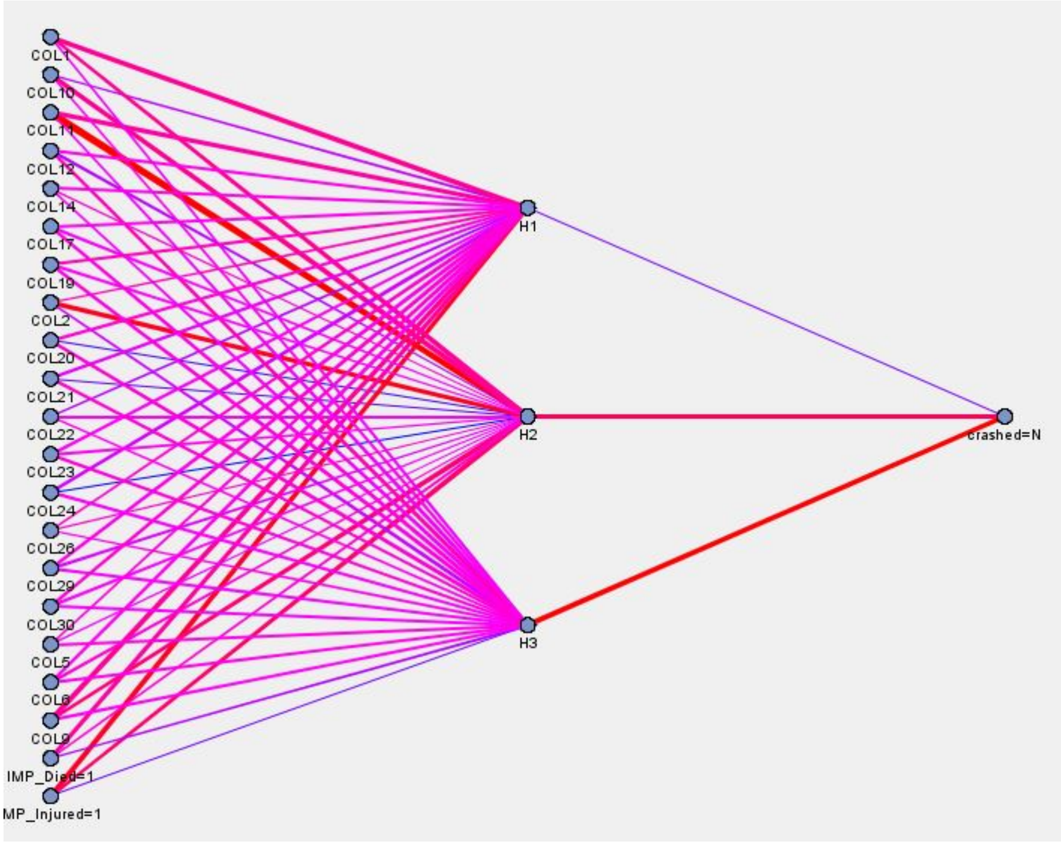
In this case, the model that produced the smallest misclassification error is the Neural Network Model.



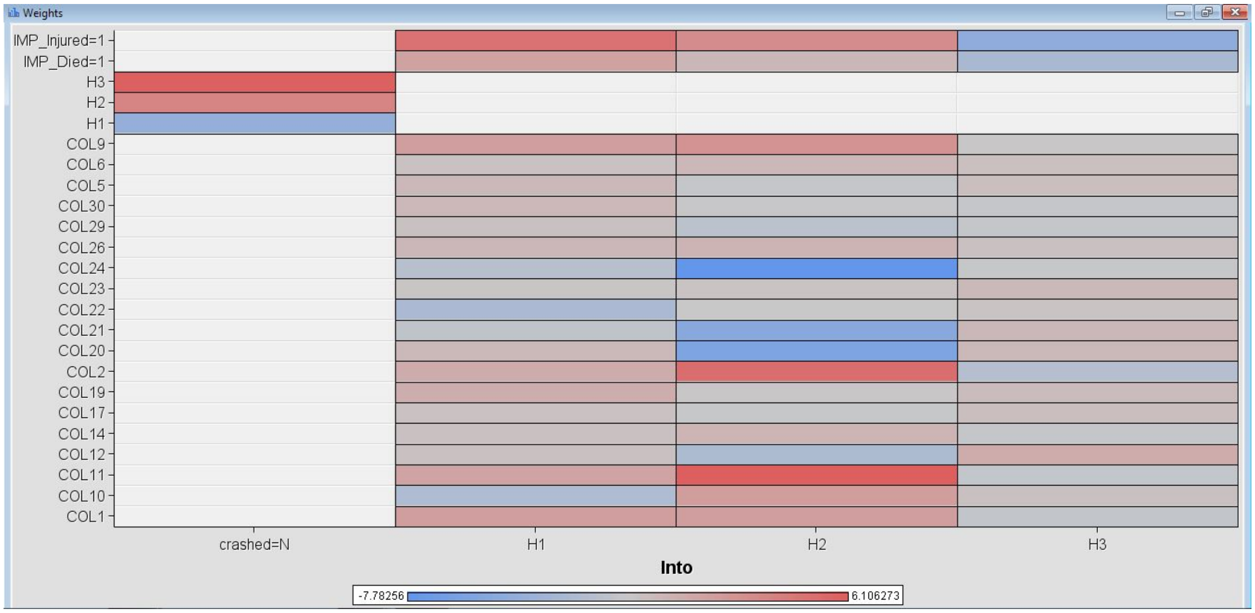
Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Target Label	Selection Criterion: Valid: Misclassification Rate	Train: Average Squared Error
Y	HPNNA	HPNNA	HP Neural	crashed		0.028749	0.014899
	HPDMForest	HPDMForest	HP Forest	crashed		0.033119	0.017733
	HPTree	HPTree	HP Tree	crashed		0.037029	0.025724
	HPReg	HPReg	HP Regres...	crashed		0.076127	0.069539

It was 2.8%, lower than that seen with the non-High Performance nodes. The new High Performance Neural Network provides clues about the relative importance of attributes in the model. First the network is displayed with colorful lines. The darker the line, the larger the weight for that attribute.

The new HP Neural Network Node also produces a visual display of the network. In this case, the inputs to the network were pre-screened using the HP Variable Selection Node. In general a neural network tends to converge faster if the variables are pre-screened.



In this case the attributes with the highest weights were text topics. Another plot that describes the relative size of the network weights is the weights plot. From this it's apparent the the topics with the larger weights are COL2, COL11, COL24, COL21 and COL20.

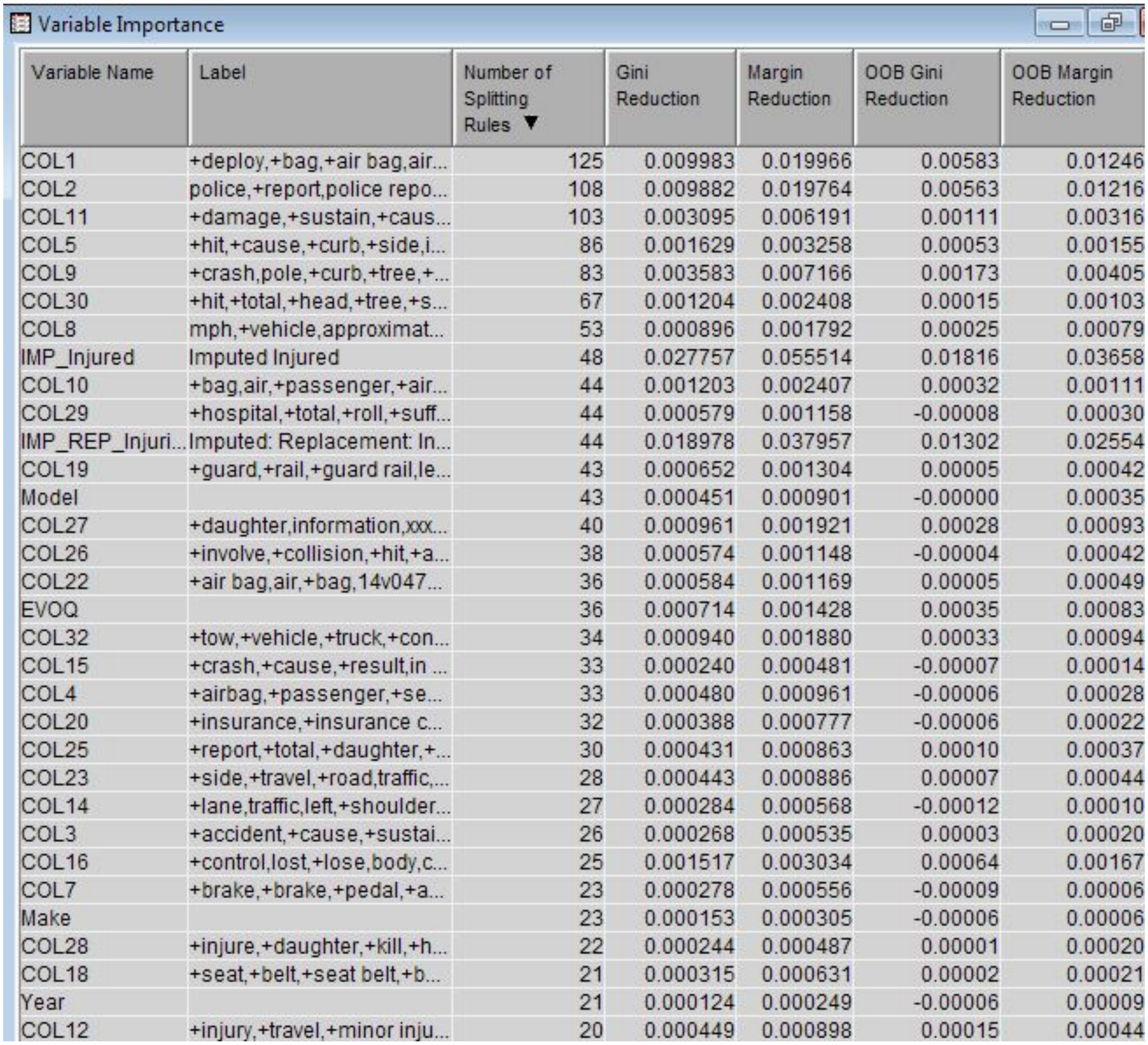


These correspond to the following toics:

- COL2 – Police Report
- COL11 – damage
- COL24 – Impact, collision
- COL21 – File Claim
- COL20 – Insurance Company

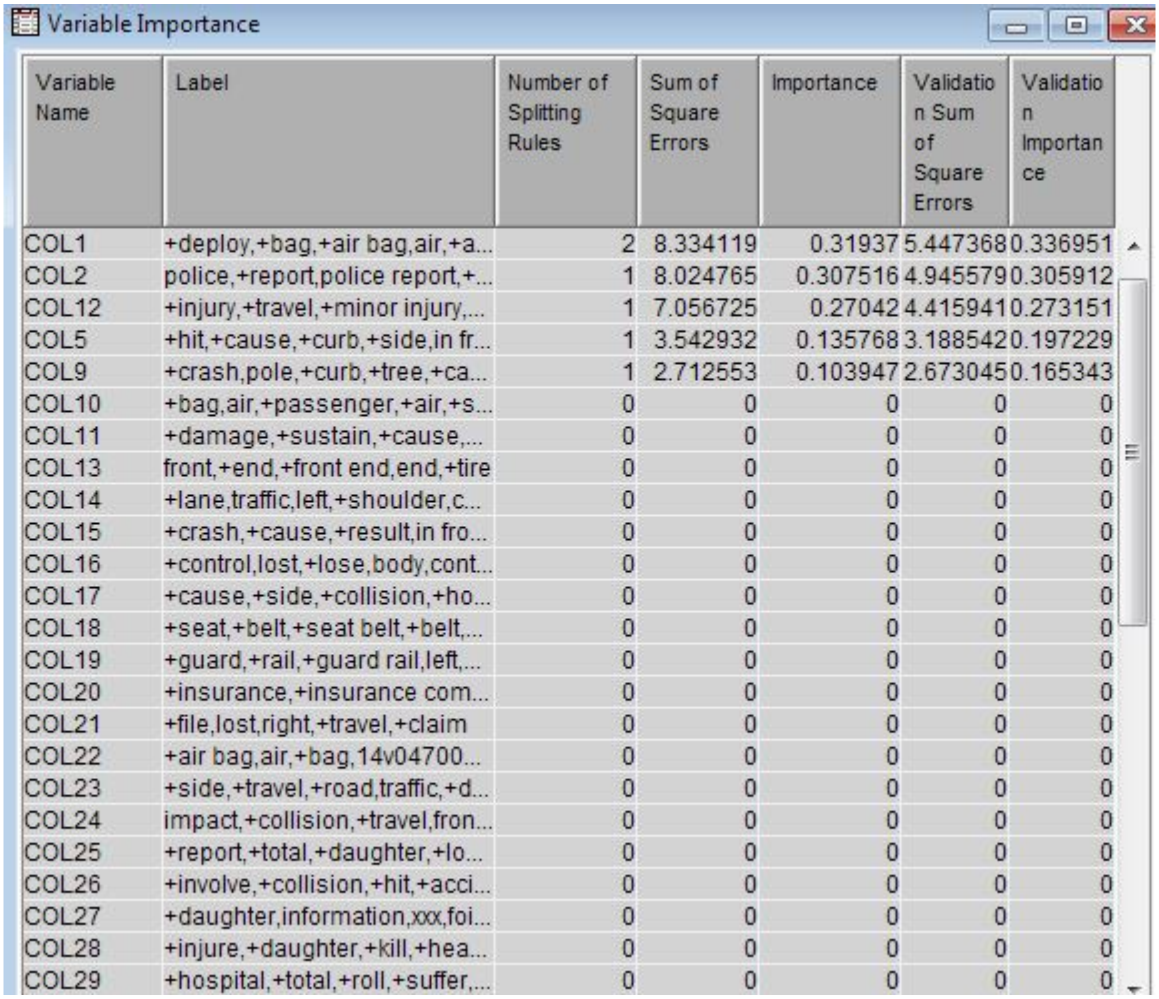
These make sense as indicators of an accident (crash). However, in this analysis, the key and ignition are not one of selected topics.

This is also seen in the importance statistics from the new HP Random Forest node .



Variable Name	Label	Number of Splitting Rules ▼	Gini Reduction	Margin Reduction	OOB Gini Reduction	OOB Margin Reduction
COL1	+deploy,+bag,+air bag,air...	125	0.009983	0.019966	0.00583	0.01246
COL2	police,+report,police repo...	108	0.009882	0.019764	0.00563	0.01216
COL11	+damage,+sustain,+caus...	103	0.003095	0.006191	0.00111	0.00316
COL5	+hit,+cause,+curb,+side,i...	86	0.001629	0.003258	0.00053	0.00155
COL9	+crash,pole,+curb,+tree,+...	83	0.003583	0.007166	0.00173	0.00405
COL30	+hit,+total,+head,+tree,+s...	67	0.001204	0.002408	0.00015	0.00103
COL8	mph,+vehicle,approximat...	53	0.000896	0.001792	0.00025	0.00079
IMP_Injured	Imputed Injured	48	0.027757	0.055514	0.01816	0.03658
COL10	+bag,air,+passenger,+air...	44	0.001203	0.002407	0.00032	0.00111
COL29	+hospital,+total,+roll,+suff...	44	0.000579	0.001158	-0.00008	0.00030
IMP_REP_Injuri...	Imputed: Replacement: In...	44	0.018978	0.037957	0.01302	0.02554
COL19	+guard,+rail,+guard rail,le...	43	0.000652	0.001304	0.00005	0.00042
Model		43	0.000451	0.000901	-0.00000	0.00035
COL27	+daughter,information,xxx...	40	0.000961	0.001921	0.00028	0.00093
COL26	+involve,+collision,+hit,+a...	38	0.000574	0.001148	-0.00004	0.00042
COL22	+air bag,air,+bag,14v047...	36	0.000584	0.001169	0.00005	0.00049
EVOQ		36	0.000714	0.001428	0.00035	0.00083
COL32	+tow,+vehicle,+truck,+con...	34	0.000940	0.001880	0.00033	0.00094
COL15	+crash,+cause,+result,in ...	33	0.000240	0.000481	-0.00007	0.00014
COL4	+airbag,+passenger,+se...	33	0.000480	0.000961	-0.00006	0.00028
COL20	+insurance,+insurance c...	32	0.000388	0.000777	-0.00006	0.00022
COL25	+report,+total,+daughter,+...	30	0.000431	0.000863	0.00010	0.00037
COL23	+side,+travel,+road,traffic,...	28	0.000443	0.000886	0.00007	0.00044
COL14	+lane,traffic,left,+shoulder...	27	0.000284	0.000568	-0.00012	0.00010
COL3	+accident,+cause,+sustai...	26	0.000268	0.000535	0.00003	0.00020
COL16	+control,lost,+lose,body,c...	25	0.001517	0.003034	0.00064	0.00167
COL7	+brake,+brake,+pedal,+a...	23	0.000278	0.000556	-0.00009	0.00006
Make		23	0.000153	0.000305	-0.00006	0.00006
COL28	+injure,+daughter,+kill,+h...	22	0.000244	0.000487	0.00001	0.00020
COL18	+seat,+belt,+seat belt,+b...	21	0.000315	0.000631	0.00002	0.00021
Year		21	0.000124	0.000249	-0.00006	0.00009
COL12	+injury,+travel,+minor inju...	20	0.000449	0.000898	0.00015	0.00044

The HP Tree Node produces a more compact solution.


Variable Importance

Variable Name	Label	Number of Splitting Rules	Sum of Square Errors	Importance	Validation Sum of Square Errors	Validation Importance
COL1	+deploy,+bag,+air bag,air,+a...	2	8.334119	0.31937	5.447368	0.336951
COL2	police,+report,police report,+...	1	8.024765	0.307516	4.945579	0.305912
COL12	+injury,+travel,+minor injury,...	1	7.056725	0.27042	4.415941	0.273151
COL5	+hit,+cause,+curb,+side,in fr...	1	3.542932	0.135768	3.188542	0.197229
COL9	+crash,pole,+curb,+tree,+ca...	1	2.712553	0.103947	2.673045	0.165343
COL10	+bag,air,+passenger,+air,+s...	0	0	0	0	0
COL11	+damage,+sustain,+cause,...	0	0	0	0	0
COL13	front,+end,+front end,end,+tire	0	0	0	0	0
COL14	+lane,traffic,left,+shoulder,c...	0	0	0	0	0
COL15	+crash,+cause,+result,in fro...	0	0	0	0	0
COL16	+control,lost,+lose,body,cont...	0	0	0	0	0
COL17	+cause,+side,+collision,+ho...	0	0	0	0	0
COL18	+seat,+belt,+seat belt,+belt,...	0	0	0	0	0
COL19	+guard,+rail,+guard rail,left,...	0	0	0	0	0
COL20	+insurance,+insurance com...	0	0	0	0	0
COL21	+file,lost,right,+travel,+claim	0	0	0	0	0
COL22	+air bag,air,+bag,14v04700...	0	0	0	0	0
COL23	+side,+travel,+road,traffic,+d...	0	0	0	0	0
COL24	impact,+collision,+travel,fron...	0	0	0	0	0
COL25	+report,+total,+daughter,+lo...	0	0	0	0	0
COL26	+involve,+collision,+hit,+acci...	0	0	0	0	0
COL27	+daughter,information,xxx,foi...	0	0	0	0	0
COL28	+injure,+daughter,+kill,+hea...	0	0	0	0	0
COL29	+hospital,+total,+roll,+suffer,...	0	0	0	0	0

SUMMARY

Both SAS Text Miner and High Performance Text Miner were able to uncover 4,710 complaints (41%) that were associated with the key and ignition problem in selected models. The latest version of High Performance Text Miner, version 13.1, is easier to use since it processes a single file containing multiple comments. In addition, the modeling output from High Performance Neural Network models and Random Forest Models makes it easier to identify key complaint topics associated with the target.

CONTACT INFORMATION

Comments and questions are valued and encouraged. Please feel free to email the author:

Ed Jones –ejones –at- stat dot tamu dot edu
Texas A&M University, Dept. of Statistics: http://www.stat.tamu.edu
Texas A&M Statistical Services: http://www.tamstatservices.com

REFERENCES

Text Mining & Big Data

Azevedo, A. and Santos M.F. (2008) KDD, SEMMA and CRISP-DM: A parallel overview. In the proceedings of the IADIS European Conference on Data Mining 2008, pp 182-185. Available from www.researchgate.net . After joining, search on the title for a pdf copy.

Berry, M.W. and Kogan, B. (2010) Text Mining: Applications & Theory; John Wiley & Sons.

Chapman, P., Clinton, J., et.al. (2000) CRISP-DM 1.0: Step-by-step data mining guide. Available from IBM at: ftp://ftp.software.ibm.com/software/analytics/spss/support/Modeler/Documentation/14/UserManual/CRISP-DM.pdf

Fayyad, U.M. et.al. (1996) From data mining to knowledge discovery: an overview. In Fayyad, U. M et.al. Advances in Knowledge Discovery and Data Mining. AAAI Press/MIT Press.

Ingersoll, G.S., Morton, T.S. and Farris, A.L. (2013) Taming Text; Manning Publications.

Liu, B. (2011) Web Data Mining, 2nd Edition; Springer.

Russell M. A. (2011) Mining the Social Web; O'Reilly.

Sathi, A. (2012) Big Data Analytics: Disruptive Technologies for Changing the Game; MC Press Online.

Weiss, S.M., Indurkhya, N. and Zhang, T. (2005) Text Mining: Predictive Methods for Analyzing Unstructured Information; Springer.

Weiss, S.M., Indurkhya, N. and Zhang, T. (2010) Fundamentals of Predictive Text Mining; Springer.

Linguistics

Bird, S., Klein, E. and Loper, E. (2009) Natural Language Processing with Python; O'Reilly.

Cambria, E. and Hussain, A. (2012) Sentic Computing: Techniques, Tools, and Applications; Springer.

Indurkhya, N. and Damerau, R. J. (2010) Handbook of Natural Language Processing; CRC Press.

Liu, B. (2012) Sentiment Analysis and Opinion Mining; Morgan & Claypool Publishers.

Manning C.D. and Schutze, H. (1999) Foundations of Statistical Natural Language Processing, MIT Press.

Pustejovsky, J. and Stubbs, A. (2012) Natural Language Annotation for Machine Learning; O'Reilly.

Sogaard A. (2013) Semi-Supervised Learning and Domain Adaption in Natural Language Processing; Morgan & Claypool Publishers.

Wilcock, G. (2009) Introduction to Linguistic Annotation and Text Analytics

Computer Science

Lin, J. and Dyer, C. (2010) Data-Intensive Text Processing with MapReduce; Morgan & Claypool Publishers.

Holmes, A. (2012) Hadoop in Practice; Manning Publishers.

Lam, C. (2011) Hadoop in Action; Manning Publishers.

SAS

SAS Technical Paper Enabling High-Performance Text Mining Procedures, https://support.sas.com/resources/papers/Tips-High-Performance-Text-Mining-Procedures.pdf

Zhao, Z., Albright, R., Cox, J. and Bieringer, A. (2013) Big Data Meets Text Mining, SAS Global Forum 2013, http://support.sas.com/resources/papers/proceedings13/400-2013.pdf

Statistics

Bartlett, Randy (2013) A practitioner's Guide to Business Analytics, McGraw-Hill.

Japkowicz, N. and Shah, M. (2011) Evaluating Learning Algorithms: A Classical Perspective; Cambridge Press.

Text Analytics Software

Chakraborty, G., Pagolu, M., and Garla, S. (2013) Text Mining and Analysis: Practical Methods, Examples, and Case Studies Using SAS; SAS, Inc.

Cunningham, H. et.al. (2012) Developing Language Processing Components with GATE Version 7 (a User Guide); online http://gate.ac.uk/userguide .

Elder, J., Miner, G. and Nisbet, R. (2011) Practical Text Mining and Statistical Analysis for Non-Structured Text Data Applications Academic Press.

Nisbet, R., Elder, J. and Miner, G. (2009) Statistical Analysis and Data Mining Applications; Academic Press.

Voice Recognition

Rabiner, L and Jung, F (1993) Fundamentals of Speech Recognition, Prentice Hall.

Jelinek, F. (1998) Statistical Methods for Speech Recognition, MIT Press.

ACRONYMS & SOFTWARE TOOLS

AAP	Advanced Analytics Platform
BA	Business Analytics
BI	Business Intelligence
CRISP-DM	Cross Industry Standard Process for Data Mining
DMP	Data Management Platform
DSP	Demand Side Platform (also Digital Signal Processor)
DW	Data Warehouse
ETL	Extract Load Transform
EM	(SAS) Enterprise Miner
GATE	General Architecture for Text Engineering (software)
Hadoop	Java Class for HDFS (distributed system for text files)
HP	High Performance
HDFS	Hadoop Distributed File System
HMM	Hidden Markov Model (used in POS tagging)
IBM Modeler	IBM's software for data & text mining, also called IBM SPSS Modeler
IE	Information Extraction
IR	Information Retrieval
KPI	Key Performance Indicator
LDC	Linguistic Data Consortium (www.ldc.upenn.edu)
MapReduce	Java Class for Hadoop designed to perform analytics on HDFS files
MPP	Massively Parallel Platform
NER	Named Entity Resolution
NLP	Natural Language Processing
NLTK	Natural Language Toolkit (Python software kit)
PCA	Principal Components Analysis
PII	Personally Identifiable Information
POS	Parts-of-Speech. (Sometimes used for point of sale.)
RapidMiner	R-based package for data mining (software)
ROC	Receiver Operating Characteristic curve.
SAS	Statistical Analysis System (software)
SAS TM	SAS Text Miner (software)
SEMMA	Sample, Explore, Modify, Model and Assess (SAS Acronym)
SPSS	Statistical Package for the Social Sciences (software)
SSP	Supply Side Platform
STATISTICA	Statistical Analysis Package (software)
SVD	Singular Value Decomposition - algorithm
SVM	Support Vector Machine – algorithm for multi-classification problems
UIMA	Unstructured Information Management Architecture

JAVA PROGRAM FOR POPULATING A SELECTED DIRECTORY WITH TEXT FILES

```java
package fileprocess;

import java.io.*;
import java.util.*;
public class FileProcess {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        System.out.println("Text Mining Comments Start -->");
        System.out.println("Opening Input File         -->");
        /* Set path to directory containing the text file, and set the
            second argument to the name of that file
        */
        File inFile =
                new File("/Users/Home/Desktop/Java/","gmc_ignrecalls.txt");
        if (inFile.exists()) {
            System.out.println("\tFound comments file "+inFile);
        }
        else {
            System.out.println("\tDid not find comments file "+inFile+"\n");
            System.exit(0);
        }
        /* Set path to directory used to store the individual file */
        File outDir = new File("/Users/Home/Desktop/Java/gmc_ignrecalls/");
        if (outDir.exists()) {
            System.out.println("\tFound output directory "+outDir+"\n");
        }
        else {
            System.out.println("\tDid not find output directory"+outDir+"\n");
            System.exit(0);
        }

        try {
            FileReader     inComments = new FileReader(inFile);
            FileWriter    outComments = null;
            BufferedReader breader    = new BufferedReader(inComments);
            BufferedWriter bwriter    = null;
            String str        = null;
            String surveyID   = null;
            String surveyDate = null;
            String comment    = null;
            String fileName   = null;
            StringTokenizer strFields = null;
            int nLine        = 1;
            int noComment    = 0;
            int nComment     = 0;
            int nCommentPlus = 0;

            System.out.println("Reading Input File 1st Line-->");
            str = breader.readLine();
            System.out.println("\tLine "+nLine+" "+str+"\n");
            nLine++;

            System.out.println("Reading Comments           -->");
            while ((str = breader.readLine())!= null)
            {
                strFields = new StringTokenizer(str, "\t\n");
                int nFields = strFields.countTokens();
                if (nFields > 0) surveyID     = strFields.nextToken();
                /* if (nFields > 1) surveyDate = strFields.nextToken(); */
                if (nFields > 1)
                {
                    str = strFields.nextToken();
                    comment = str.trim();
                    if (comment.length() > 0) {
                        nComment++;
                        /* set first string to the path of the directory
                            you are using to store the individual text files
                        */
      fileName = "/Users/Jones/Desktop/Java/gmc_ignrecalls/Survey"+surveyID+".txt";
                        outComments = new FileWriter(fileName);
                        bwriter     = new BufferedWriter(outComments);
                        outComments.write(comment);
                        outComments.close();
                    }
                }
                if (nFields > 3) {
                 System.out.println("ERROR - Extra Token in Line: "+nLine+" "+str);
                 nCommentPlus++;
                }
                nLine++;
            }
            inComments.close();
            System.out.println("Closed Input File-------------------------------");
            noComment = nLine-2-nComment-nCommentPlus;
            System.out.println("Total Comment Files            -->"+(nLine-2));
            System.out.println("Files without a comment        -->"+noComment);
            System.out.println("Files with one or more comments-->"+nComment);
            System.out.println("Files with two or more comments-->"+nCommentPlus);
        }
        catch (IOException e) {
            System.out.println(e);
            System.exit(1);
        }
    }
}
```