Final Project – STAT 656

Team Members

Divyank Garg

Mayank Mishra

Pawan Dixit

Sajjaat Muhemmed Reyath

Shivam Solanki

Index

Section 1: Overview – General objectives:

National Highway Traffic Safety (NHTSA) is a US governmental organization dedicated to prevent, protect, and reduce vehicle related crashes. The problem statement given to us contains dataset collected from NHTSA with 5330 consumer complaints from an automobile maker Honda in the year 2001 to 2003. It contains not only the complaints but also other information about the reason for their complaint and the type and condition of their automobile. The NTHSA uses these complaints to identify potential needs for automobile recalls.

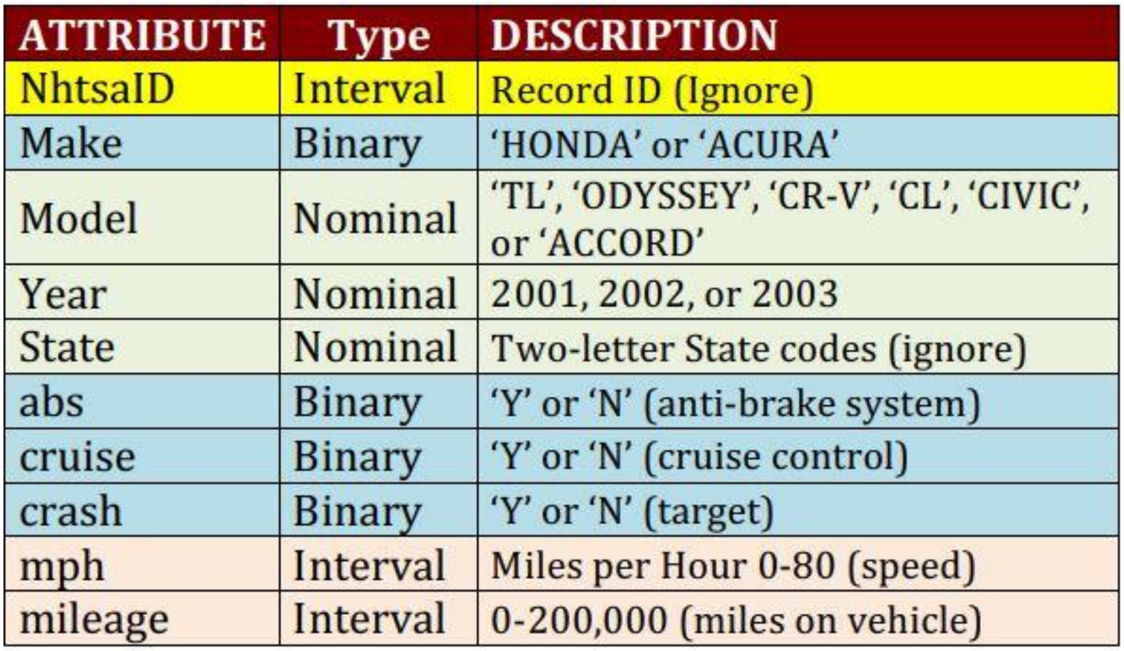
The objective is to build a predictive classification model to predict the probability of crash i.e. to classify crash into “Y” (crash) and “N” (no crash) based upon the topic and sentiment analysis from the description provided in the dataset as well as using other attributes such as Make, Model, Cruise, Crash, mph and mileage.

Both SAS Enterprise miner 14.3 and Python have been used to analyze the data and predict the probability of crash.

Section 2: Data – Descriptive statistics:

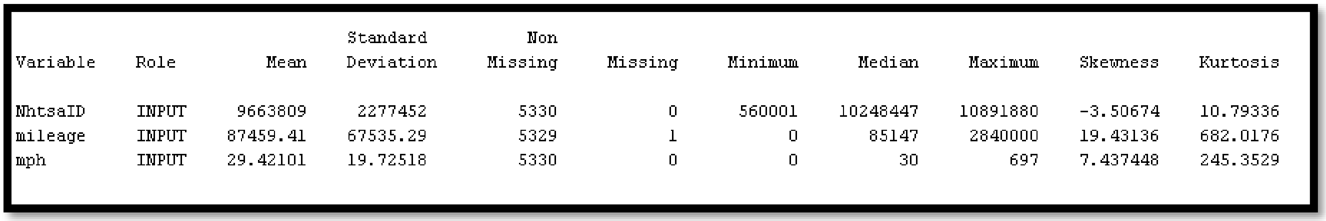
The original data contains 5330 observations and 11 columns out of which there are 3 interval attributes, 3 nominal attributes, 4 binary attributes and one column containing text description of the crash provided by the consumer.

The following table shows various attributes, its type and the description associated.



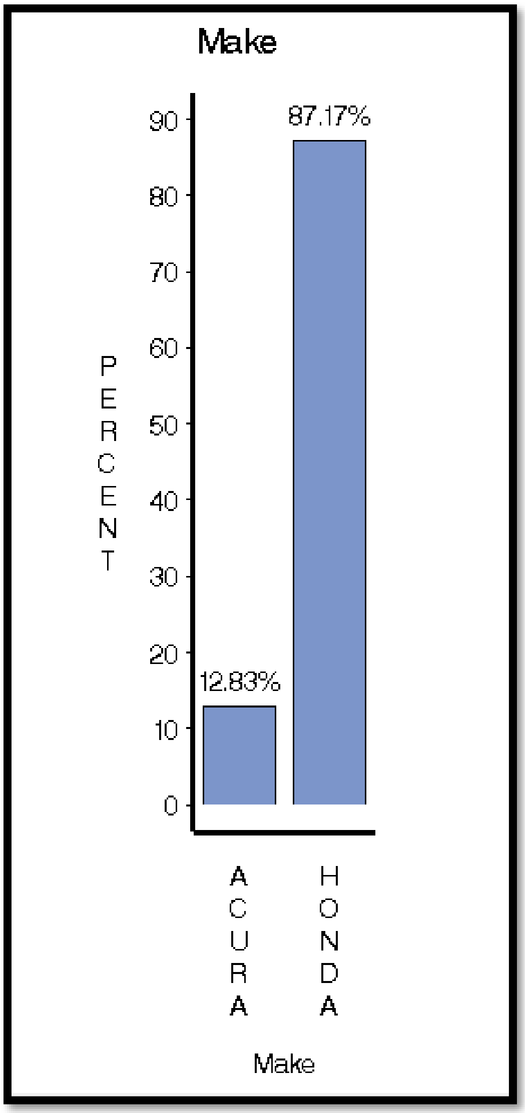
ATTRIBUTE	Type	DESCRIPTION
NhtsaID	Interval	Record ID (Ignore)
Make	Binary	‘HONDA’ or ‘ACURA’
Model	Nominal	‘TL’, ‘ODYSSEY’, ‘CR-V’, ‘CL’, ‘CIVIC’, or ‘ACCORD’
Year	Nominal	2001, 2002, or 2003
State	Nominal	Two-letter State codes (ignore)
abs	Binary	‘Y’ or ‘N’ (anti-brake system)
cruise	Binary	‘Y’ or ‘N’ (cruise control)
crash	Binary	‘Y’ or ‘N’ (target)
mph	Interval	Miles per Hour 0-80 (speed)
mileage	Interval	0-200,000 (miles on vehicle)

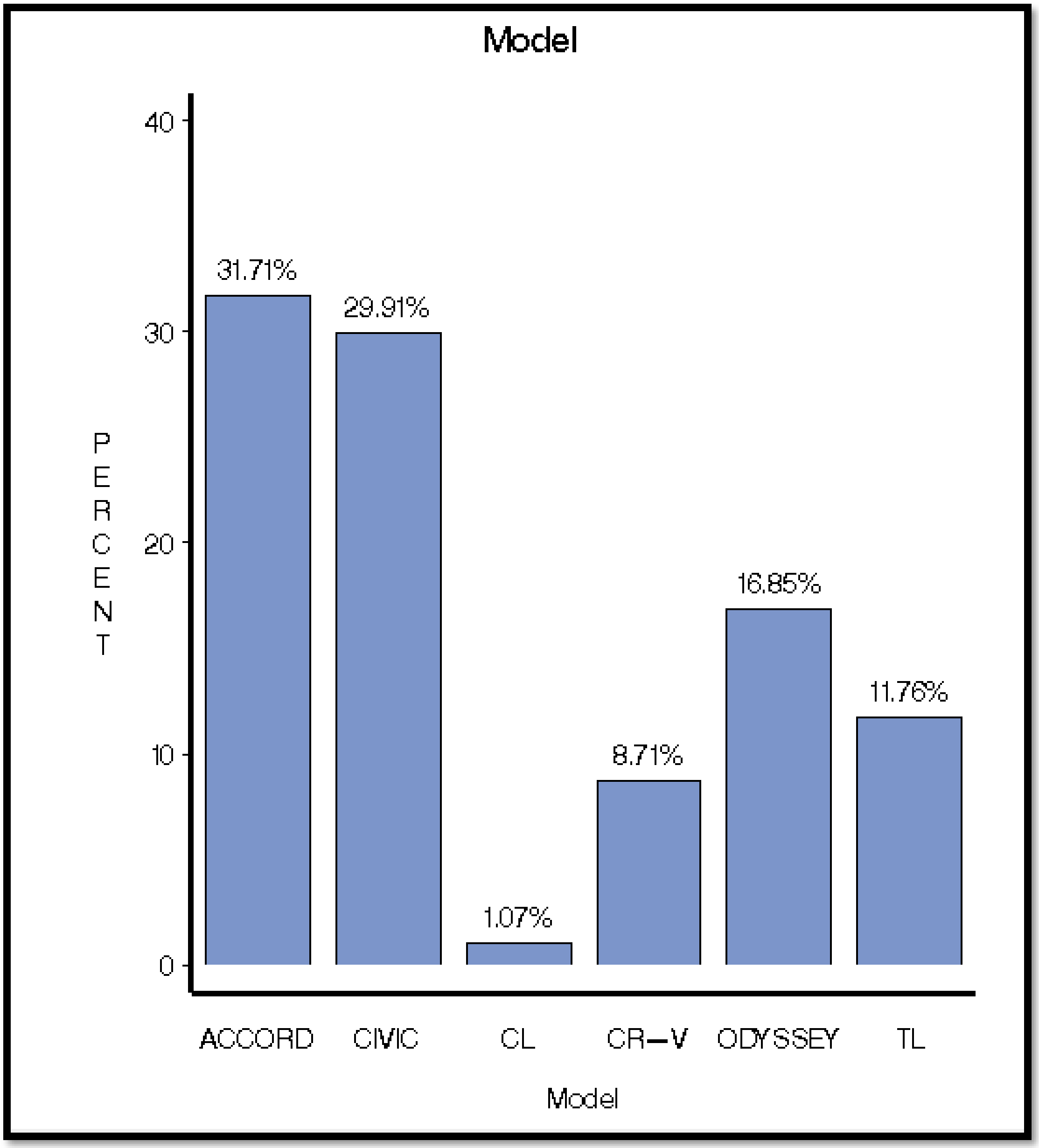
In SAS-EM, the Stat explore node gives a good understanding of the given data with minimum and maximum values of the data. The positive values of skewness for the variables mileage and speed indicates that the data is skewed to the right.



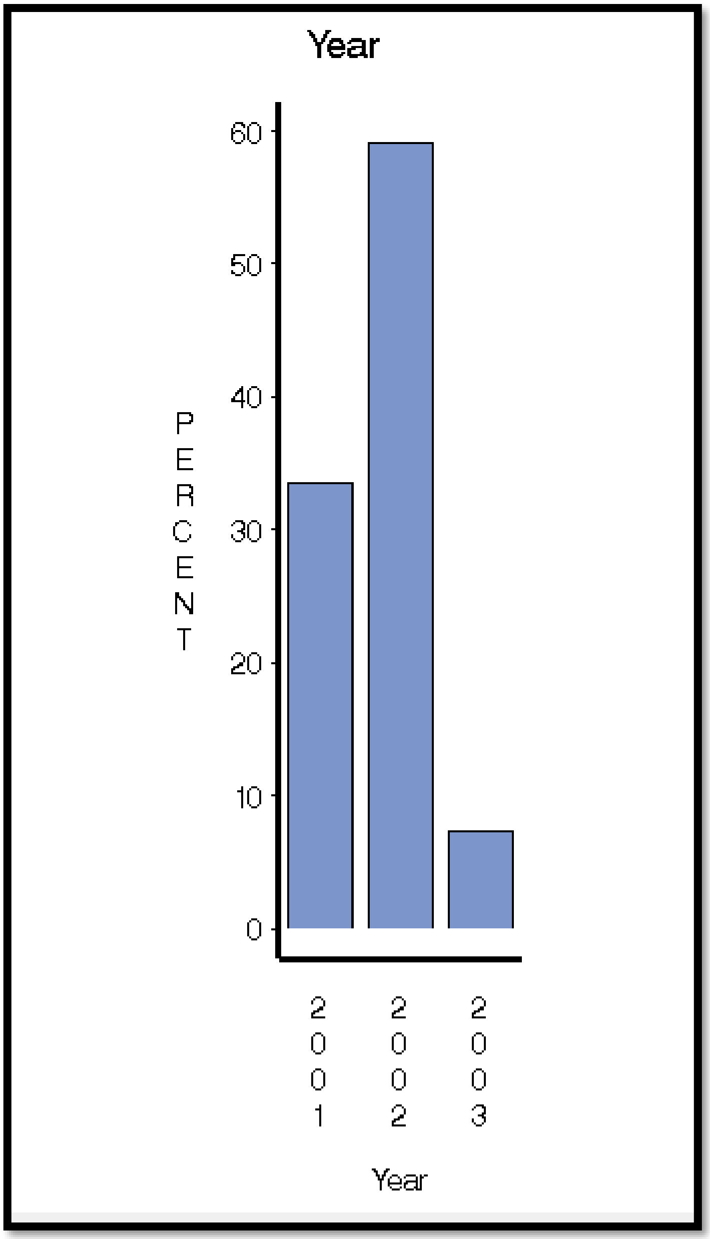
```
                                   Standard        Non
Variable     Role        Mean     Deviation      Missing     Missing     Minimum      Median     Maximum    Skewness    Kurtosis

NhtsaID      INPUT     9663809      2277452         5330           0      560001    10248447    10891880    -3.50674    10.79336
mileage      INPUT    87459.41     67535.29         5329           1           0       85147     2840000    19.43136    682.0176
mph          INPUT    29.42101     19.72518         5330           0           0          30         697    7.437448    245.3529
```

Since visualizing is one of best ways in expressing and efficiently deriving insights from the data. The Multiplot node is an excellent node in visualizing the given data. From the below diagram, we can see that the 87% of the complaints are from Honda and remaining 13% are from the luxury brand of Japanese automaker Honda.

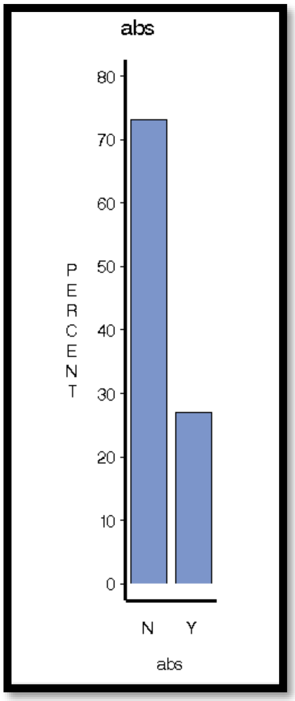




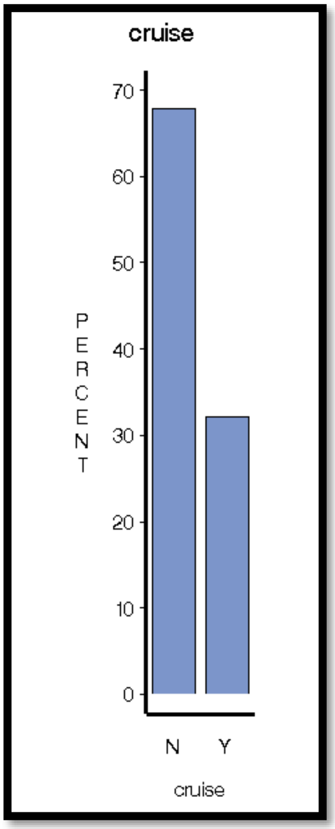
The diagram above shows that percentage distribution of complaints received from different models of Honda.



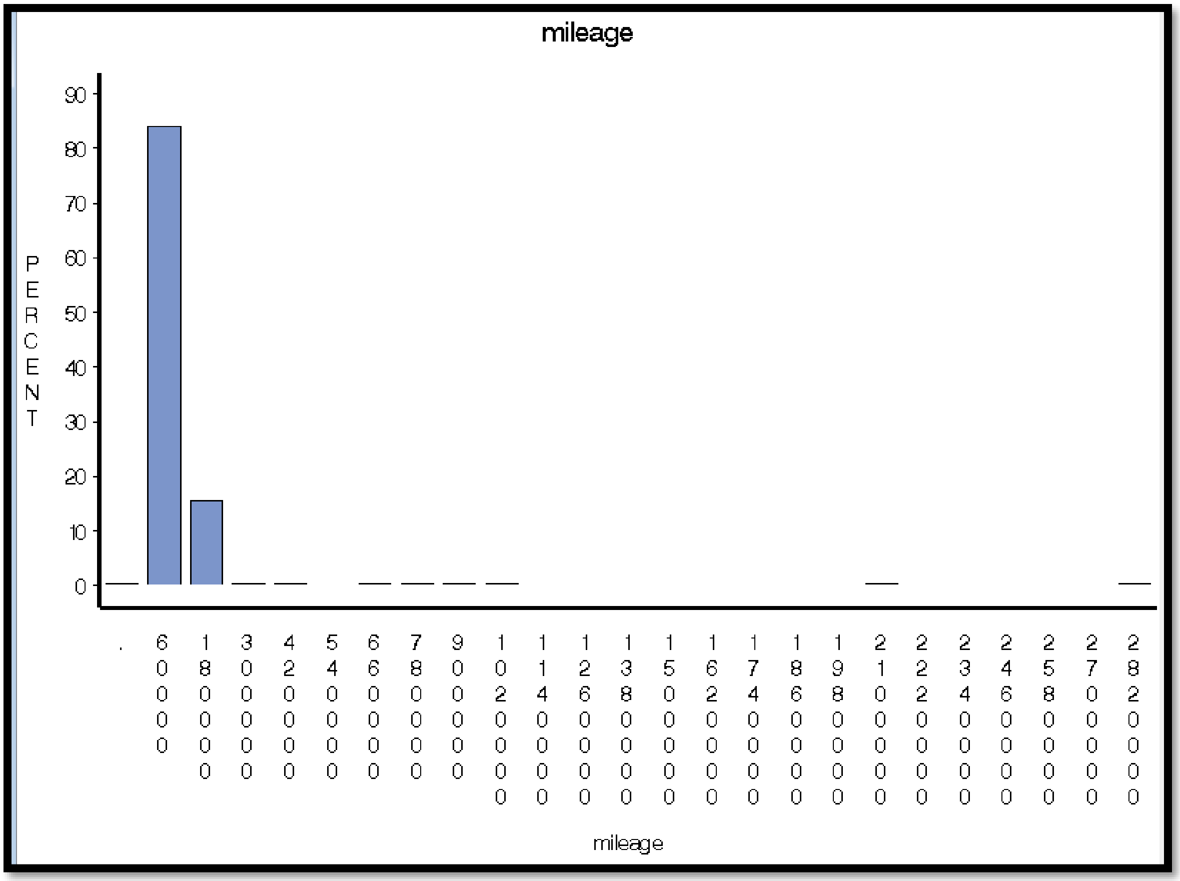
The above figure shows that number of complaints distributed during the years 2001 to 2003.

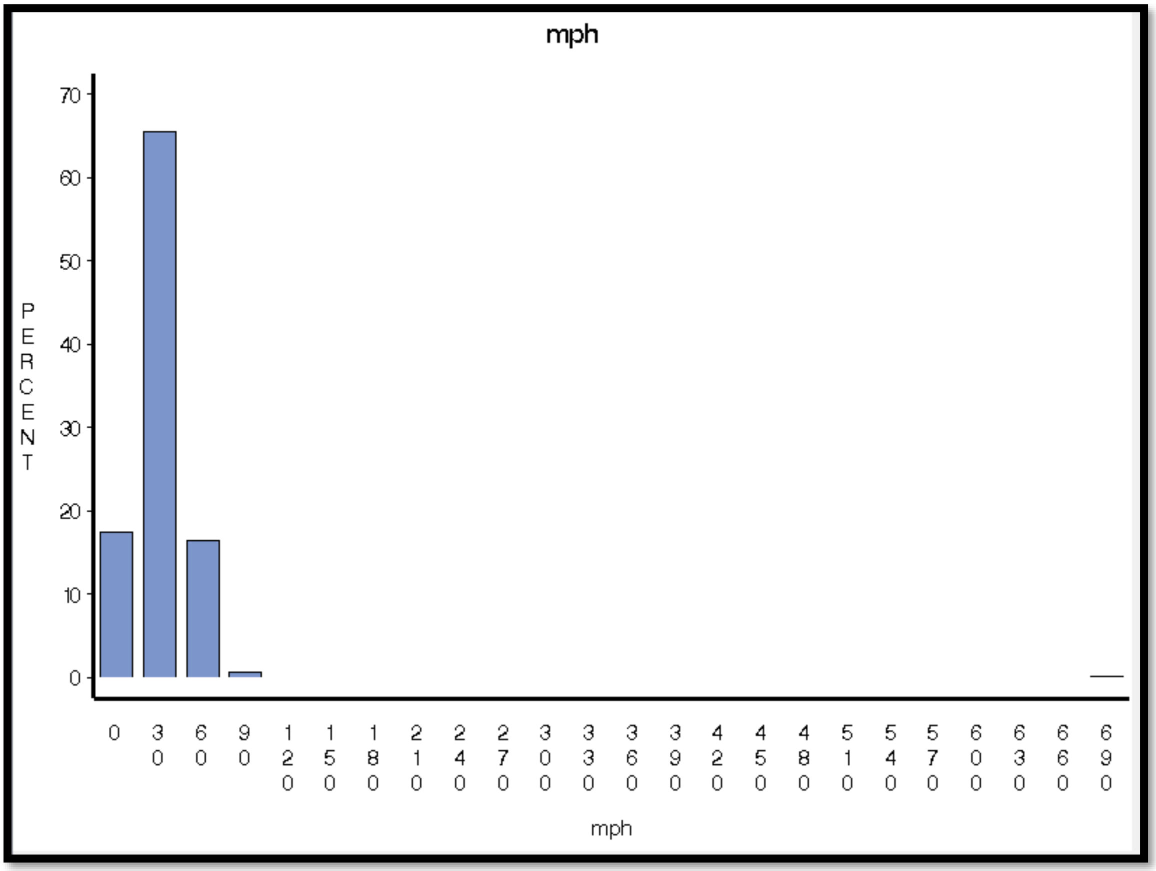


Anti-lock braking system(ABS) technology allows the car to maintain a tractive contact with the road surface there by preventing the car from uncontrolled skidding. From the above figure, we can see that 70% of the complaints received from the cars with no ABS.

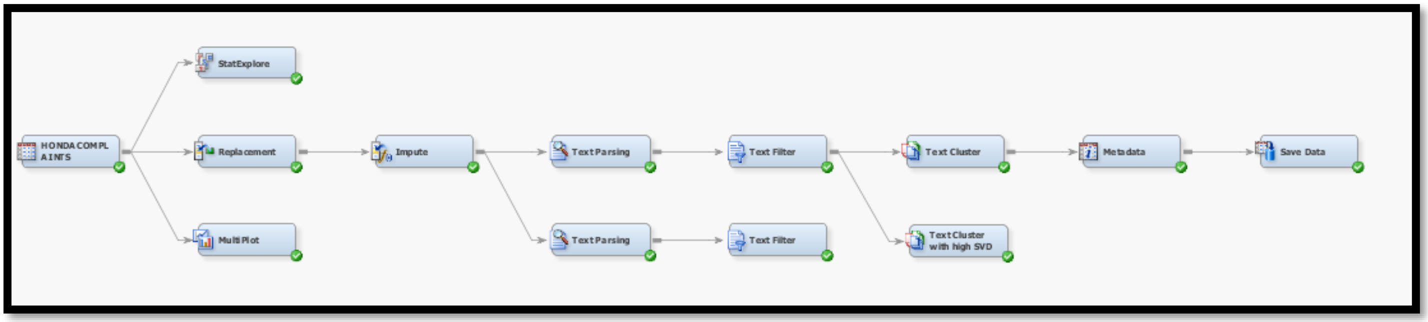


Cruise control is a technology that is used to maintain a constant speed on the road with using accelerator, the above figure shows that close to 70% of complaints received are from the car with no cruise control in it.


mileage
PERCENT
90
80
70
60
50
40
30
20
10
0
.
6000
18000
30000
42000
54000
66000
78000
90000
102000
114000
126000
138000
150000
162000
174000
186000
198000
210000
222000
234000
246000
258000
270000
282000
mileage

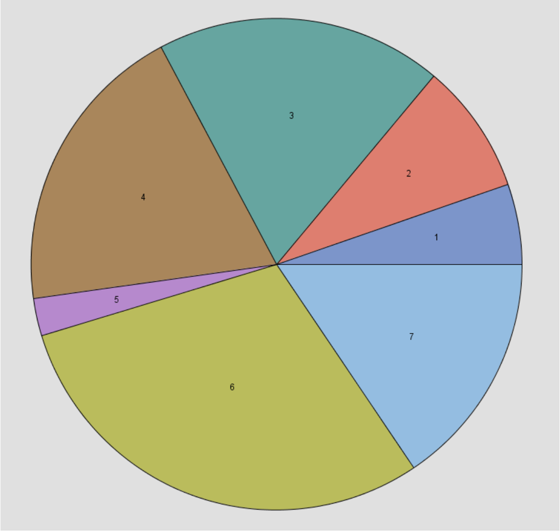

mph
PERCENT
70
60
50
40
30
20
10
0
0
30
60
90
120
150
180
210
240
270
300
330
360
390
420
450
480
510
540
570
600
630
660
690
mph

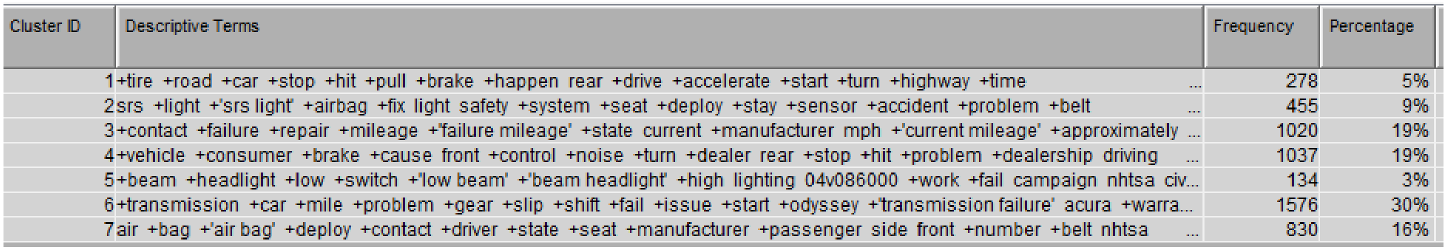
Section 3: Analytics approach:



SAS EM

With the given data dictionary, the outliers in the data are cleaned with the replacement and impute nodes. The text in the data is parsed with parsing node with its default setting. The parsed data is then filtered with inverse document frequency term weights. Using the text cluster node, the filtered data is then modelled into seven topics. These 7 topics were created using low SVD resolutions setting. The data is saved under the file name Topics.

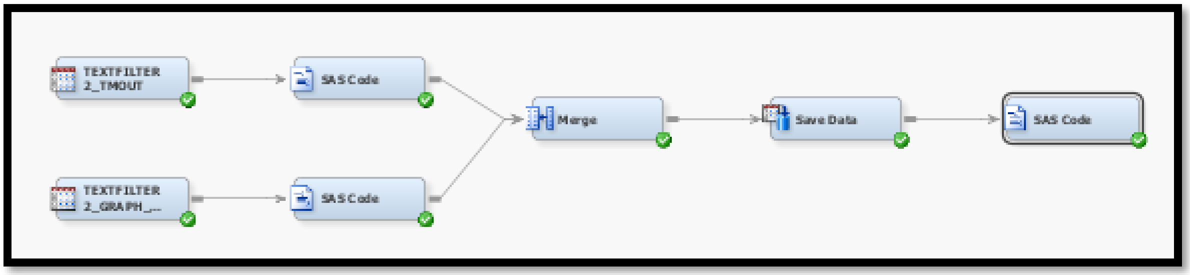




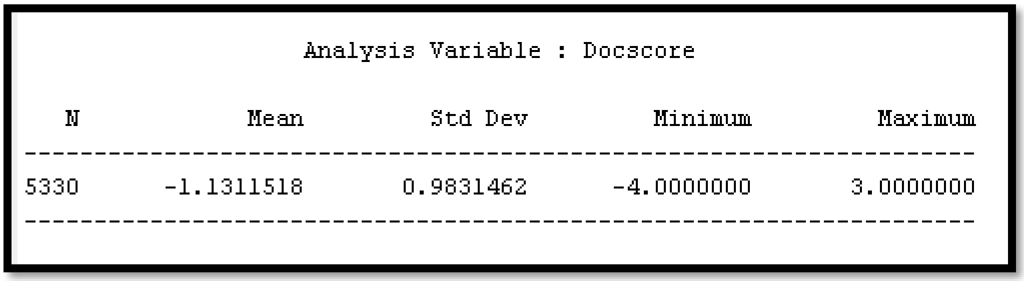
Cluster ID	Descriptive Terms	Frequency	Percentage
1	+tire +road +car +stop +hit +pull +brake +happen rear +drive +accelerate +start +turn +highway +time ...	278	5%
2	srs +light +'srs light' +airbag +fix light safety +system +seat +deploy +stay +sensor +accident +problem +belt ...	455	9%
3	+contact +failure +repair +mileage +'failure mileage' +state current +manufacturer mph +'current mileage' +approximately ...	1020	19%
4	+vehicle +consumer +brake +cause front +control +noise +turn +dealer rear +stop +hit +problem +dealership driving ...	1037	19%
5	+beam +headlight +low +switch +'low beam' +'beam headlight' +high lighting 04v086000 +work +fail campaign nhtsa civ...	134	3%
6	+transmission +car +mile +problem +gear +slip +shift +fail +issue +start +odyssey +'transmission failure' acura +warra...	1576	30%
7	air +bag +'air bag' +deploy +contact +driver +state +seat +manufacturer +passenger side front +number +belt nhtsa ...	830	16%

The above table shows the frequency and percentage of each topics in the document. It can be clearly observed that Topic Cluster 6 is the dominant topic contained in 1576 of the documents covering 30% of the total document.

Sentiment Analysis

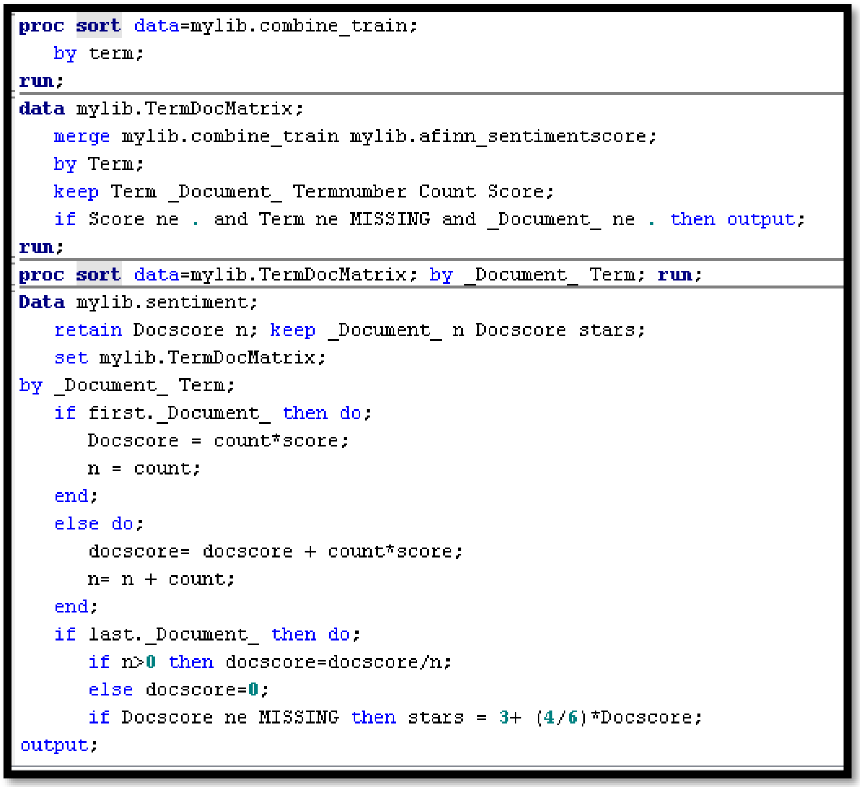


In order to create the sentiment of all the complaints, the given data is parsed and filtered without any stemming, parts of speech tagging, and stop words. The AFINN start list is used to parse the complaints. So, only the words that are included in the start list appear in the parsing results. In the filter node, term weight and frequency weighting were set to none. After renaming and sorting the documents, the sentiment of the entire corpus was found to be **-1.13** which is quite obvious since the document being a consumer complaint would have a negative overall sentiment.

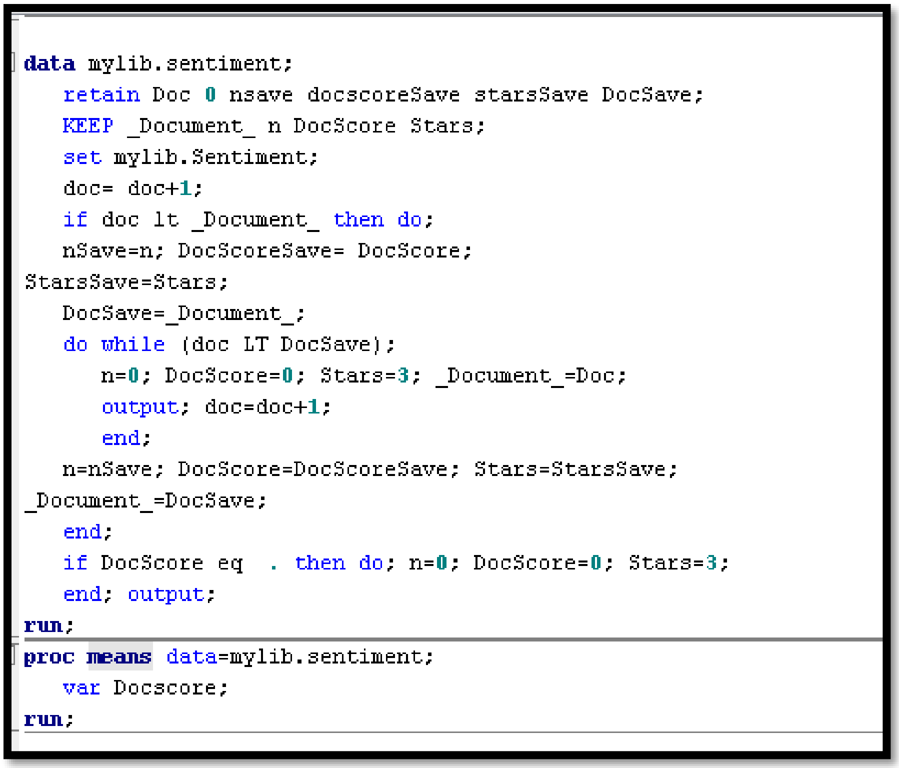


```
                       Analysis Variable : Docscore

   N             Mean         Std Dev          Minimum          Maximum
--------------------------------------------------------------------------
5330       -1.1311518       0.9831462       -4.0000000        3.0000000
--------------------------------------------------------------------------
```

Code for scoring sentiment

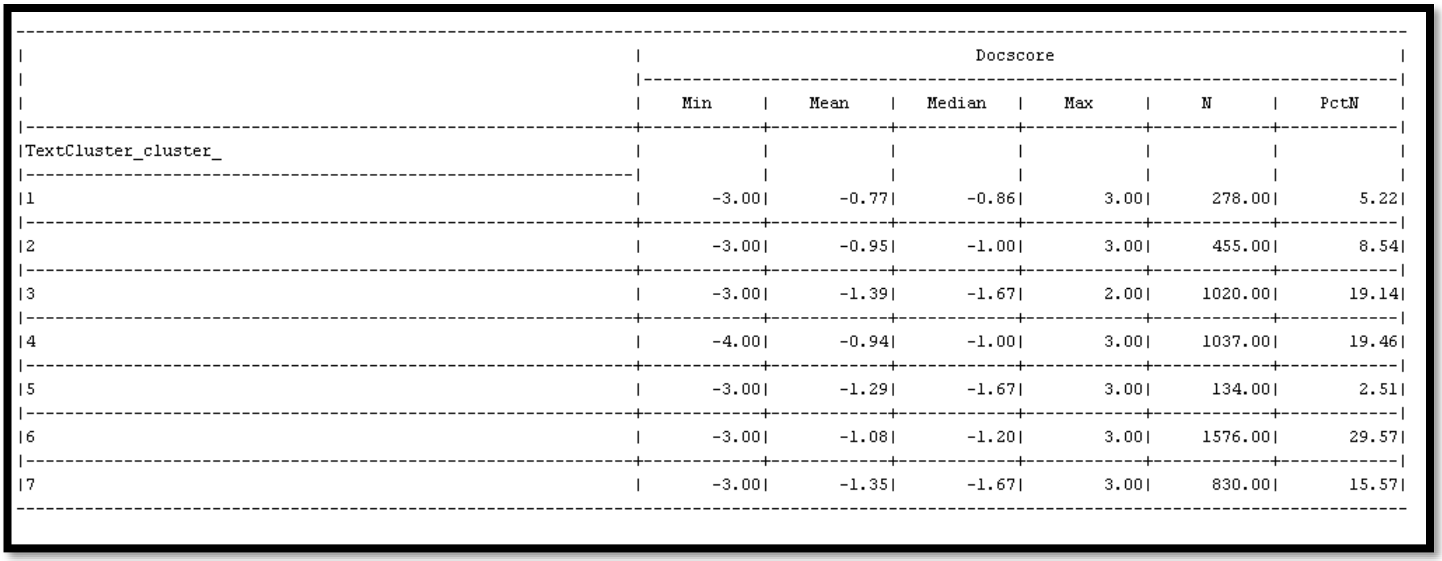


```
proc sort data=mylib.combine_train;
   by term;
run;
data mylib.TermDocMatrix;
   merge mylib.combine_train mylib.afinn_sentimentscore;
   by Term;
   keep Term _Document_ Termnumber Count Score;
   if Score ne . and Term ne MISSING and _Document_ ne . then output;
run;
proc sort data=mylib.TermDocMatrix; by _Document_ Term; run;
Data mylib.sentiment;
   retain Docscore n; keep _Document_ n Docscore stars;
   set mylib.TermDocMatrix;
by _Document_ Term;
   if first._Document_ then do;
      Docscore = count*score;
      n = count;
   end;
   else do;
      docscore= docscore + count*score;
      n= n + count;
   end;
   if last._Document_ then do;
      if n>0 then docscore=docscore/n;
      else docscore=0;
      if Docscore ne MISSING then stars = 3+ (4/6)*Docscore;
output;
```



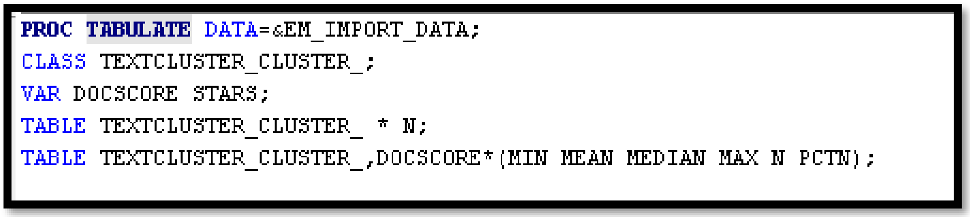
```
data mylib.sentiment;
   retain Doc 0 nsave docscoreSave starsSave DocSave;
   KEEP _Document_ n DocScore Stars;
   set mylib.Sentiment;
   doc= doc+1;
   if doc lt _Document_ then do;
   nSave=n; DocScoreSave= DocScore;
StarsSave=Stars;
   DocSave=_Document_;
   do while (doc LT DocSave);
      n=0; DocScore=0; Stars=3; _Document_=Doc;
      output; doc=doc+1;
      end;
   n=nSave; DocScore=DocScoreSave; Stars=StarsSave;
_Document_=DocSave;
   end;
   if DocScore eq  . then do; n=0; DocScore=0; Stars=3;
   end; output;
run;
proc means data=mylib.sentiment;
   var Docscore;
run;
```

The topic cluster file and sentiment file which contains the sentiment for each complaint was merged by document to score the sentiment of each cluster.



```
---------------------------------------------------------------------------------------------------------
|                                 |                              Docscore                               |
|                                 |---------------------------------------------------------------------|
|                                 |   Min    |   Mean   |  Median  |   Max    |    N     |   PctN    |
|---------------------------------+----------+----------+----------+----------+----------+-----------|
|TextCluster_cluster_             |          |          |          |          |          |           |
|---------------------------------|          |          |          |          |          |           |
|1                                |    -3.00|     -0.77|     -0.86|      3.00|    278.00|       5.22|
|---------------------------------+----------+----------+----------+----------+----------+-----------|
|2                                |    -3.00|     -0.95|     -1.00|      3.00|    455.00|       8.54|
|---------------------------------+----------+----------+----------+----------+----------+-----------|
|3                                |    -3.00|     -1.39|     -1.67|      2.00|   1020.00|      19.14|
|---------------------------------+----------+----------+----------+----------+----------+-----------|
|4                                |    -4.00|     -0.94|     -1.00|      3.00|   1037.00|      19.46|
|---------------------------------+----------+----------+----------+----------+----------+-----------|
|5                                |    -3.00|     -1.29|     -1.67|      3.00|    134.00|       2.51|
|---------------------------------+----------+----------+----------+----------+----------+-----------|
|6                                |    -3.00|     -1.08|     -1.20|      3.00|   1576.00|      29.57|
|---------------------------------+----------+----------+----------+----------+----------+-----------|
|7                                |    -3.00|     -1.35|     -1.67|      3.00|    830.00|      15.57|
---------------------------------------------------------------------------------------------------------
```

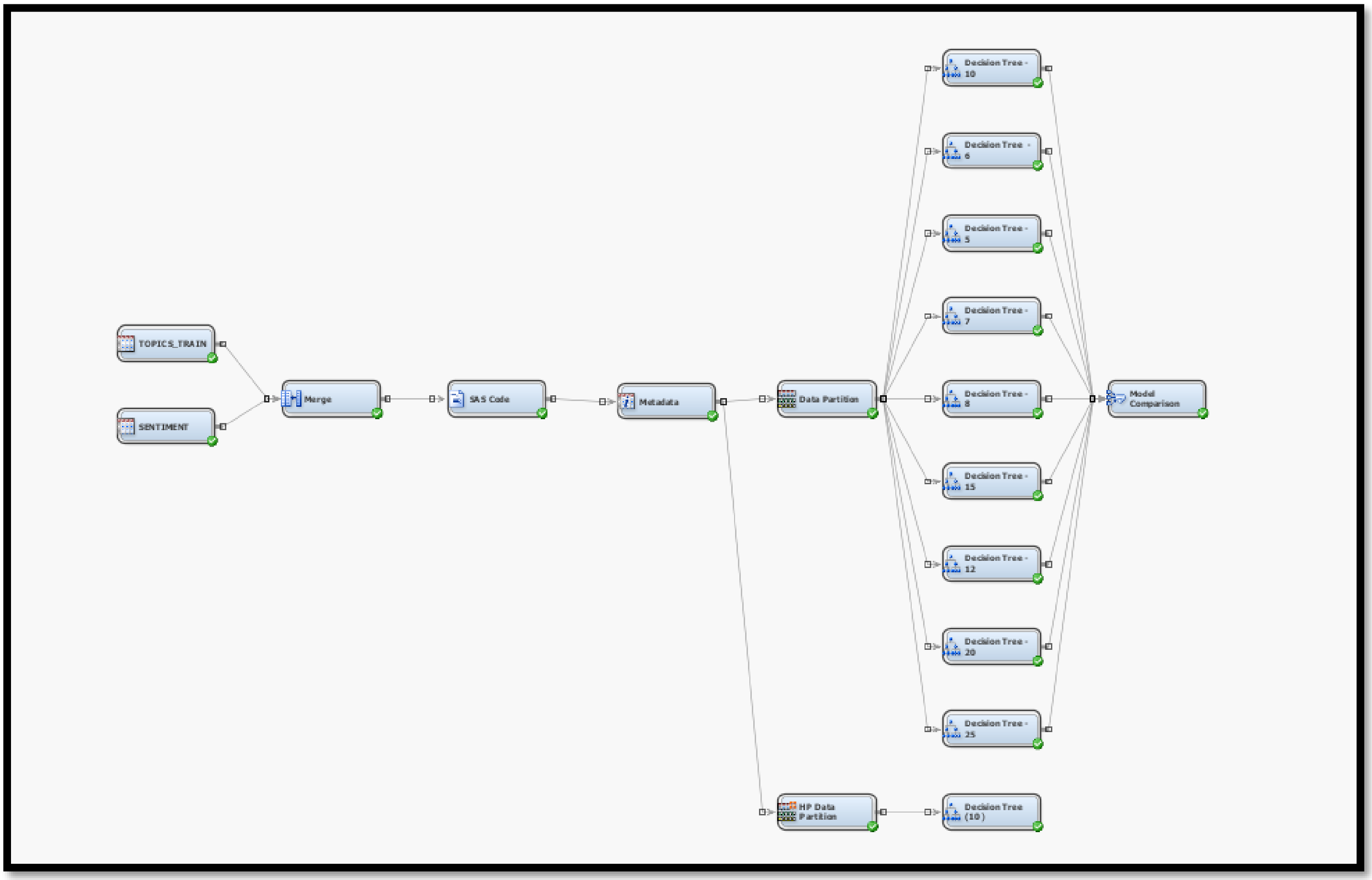
Code for evaluating sentiment score for each cluster node:



```
PROC TABULATE DATA=&EM_IMPORT_DATA;
CLASS TEXTCLUSTER_CLUSTER_;
VAR DOCSCORE STARS;
TABLE TEXTCLUSTER_CLUSTER_ * N;
TABLE TEXTCLUSTER_CLUSTER_,DOCSCORE*(MIN MEAN MEDIAN MAX N PCTN);
```

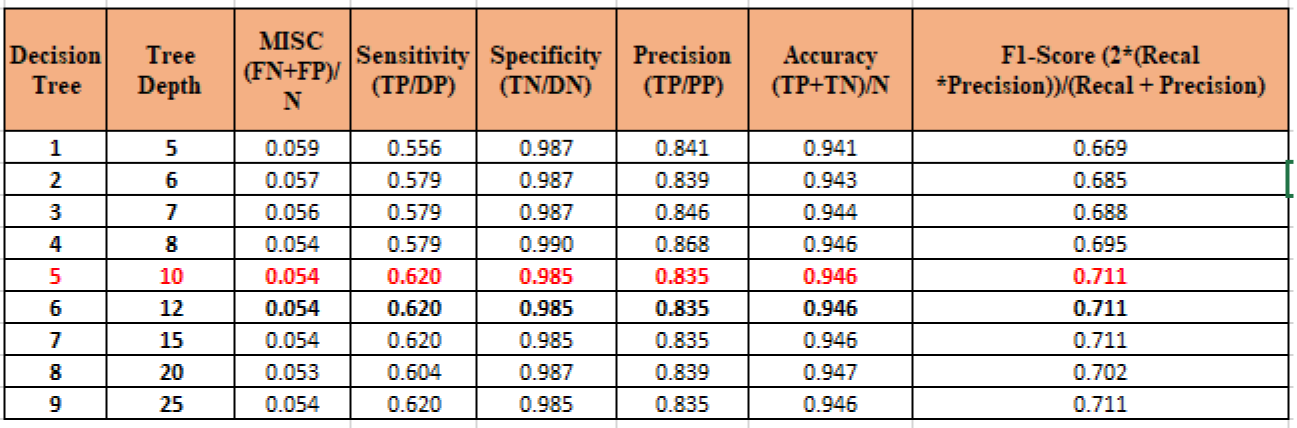
Results

Model for predicting the probability of crash



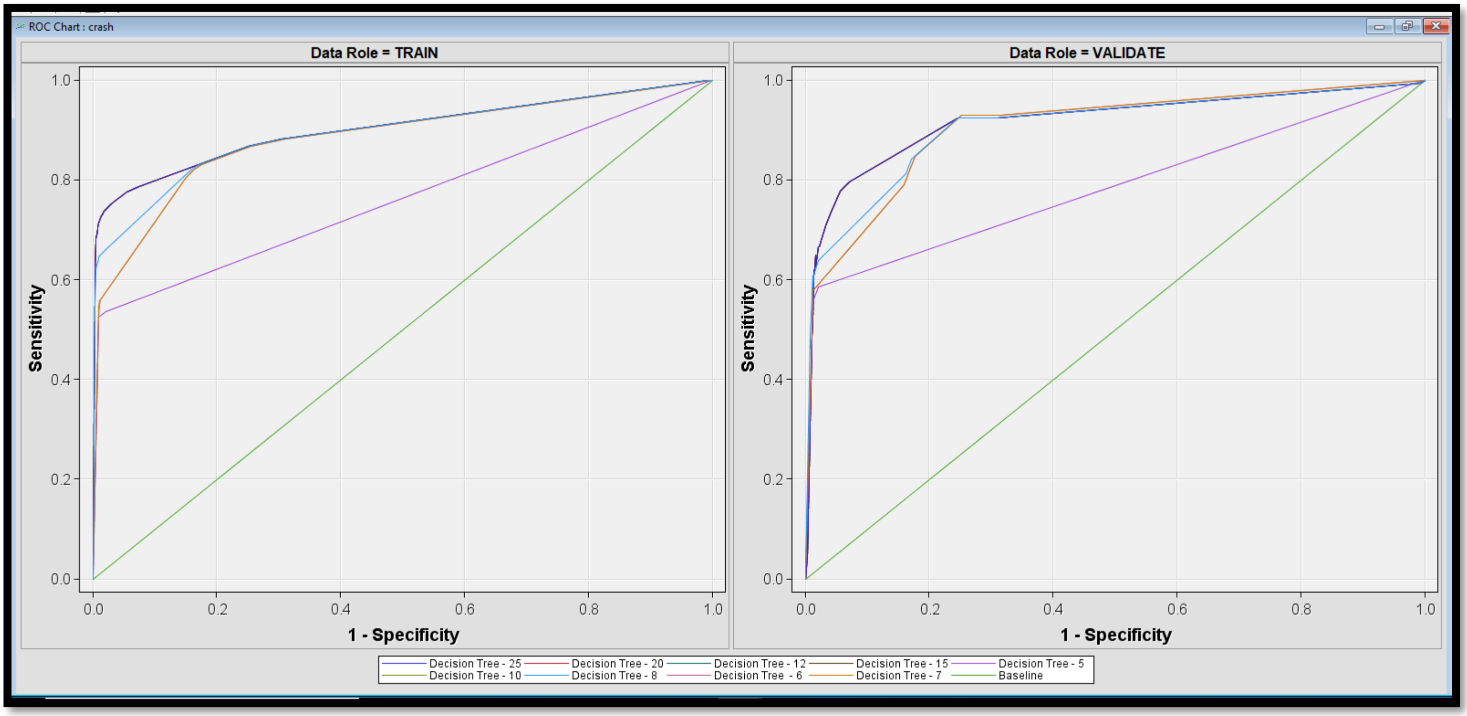
The data was partitioned with data partition node with 70/30 split. As required in the problem statement, decision trees as a classification model was used with different depths, with a default leaf size of 8 to predict the probability of the crash.

Tree depths of 5,6,7,8,10,12,15,20,25 were used to find the best model in predicting the probability of crash. Confusion matrix at different tree depth was generated. Since in the prediction modelling of the probability of crash, the value of True Positive rate is of utmost importance, therefore Sensitivity is chosen as the deciding criteria to select the best model. The highest sensitivity value on the validation data set comes out to be 0.620 which appears first in the Decision tree with depth =10 and further increasing the depth of the tree would only lead to overfitting the data with little improvement in accuracy.

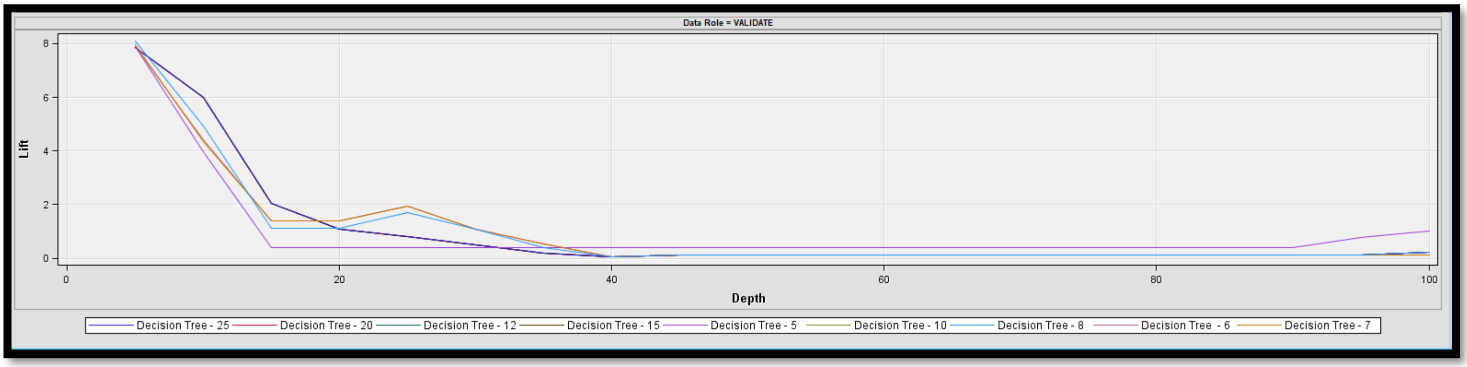


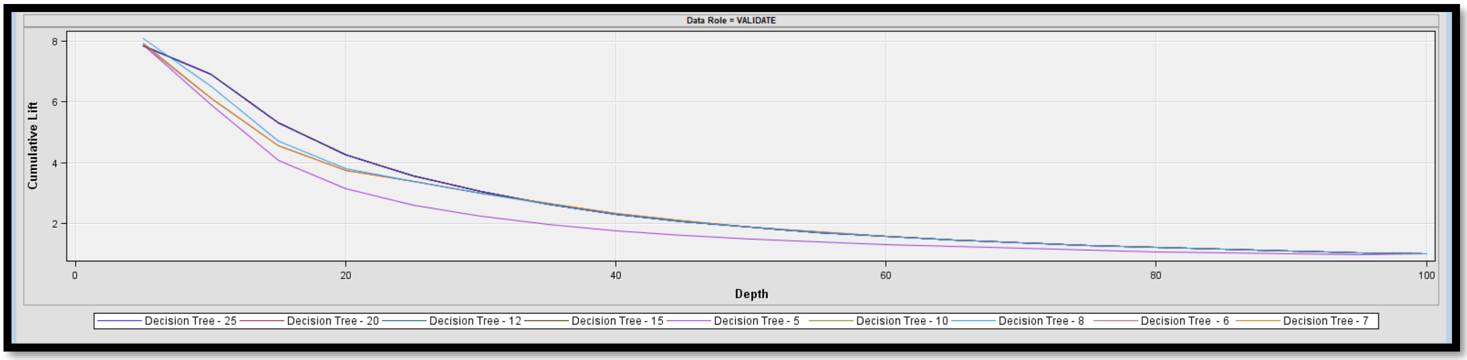
Decision Tree	Tree Depth	MISC (FN+FP)/N	Sensitivity (TP/DP)	Specificity (TN/DN)	Precision (TP/PP)	Accuracy (TP+TN)/N	F1-Score (2*(Recal *Precision))/(Recal + Precision)
1	5	0.059	0.556	0.987	0.841	0.941	0.669
2	6	0.057	0.579	0.987	0.839	0.943	0.685
3	7	0.056	0.579	0.987	0.846	0.944	0.688
4	8	0.054	0.579	0.990	0.868	0.946	0.695
5	10	0.054	0.620	0.985	0.835	0.946	0.711
6	12	0.054	0.620	0.985	0.835	0.946	0.711
7	15	0.054	0.620	0.985	0.835	0.946	0.711
8	20	0.053	0.604	0.987	0.839	0.947	0.702
9	25	0.054	0.620	0.985	0.835	0.946	0.711

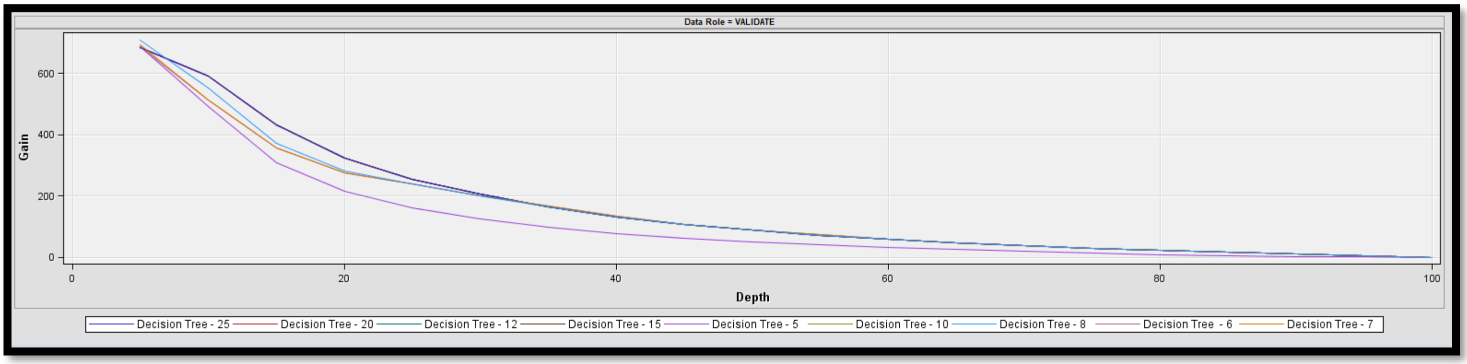
Decision tree model with the depth of 10 was found to be the best model for predicting the probability of crash. The rationale for this conclusion is elaborated below:



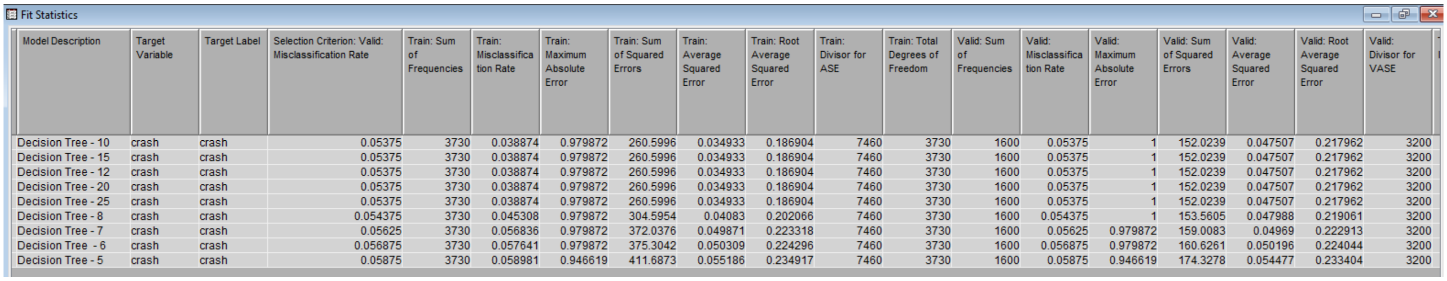
From the above ROC chart, we can conclude that the area under the curve(AUC) is maximum for the tree depth of 10 (purple line).







Lift is a measure of the effectiveness of a predictive model calculated as the ratio between the results obtained with and without the predictive model. For visually predicting the best model performance, cumulative gains and lift charts are the some of the effective parameters. The greater the area between the lift curve and the baseline, the better the model. From the above figure, the decision tree with depth 10 has the highest cumulative gain.


Fit Statistics

Model Description	Target Variable	Target Label	Selection Criterion: Valid: Misclassification Rate	Train: Sum of Frequencies	Train: Misclassification Rate	Train: Maximum Absolute Error	Train: Sum of Squared Errors	Train: Average Squared Error	Train: Root Average Squared Error	Train: Divisor for ASE	Train: Total Degrees of Freedom	Valid: Sum of Frequencies	Valid: Misclassification Rate	Valid: Maximum Absolute Error	Valid: Sum of Squared Errors	Valid: Average Squared Error	Valid: Root Average Squared Error	Valid: Divisor for VASE
Decision Tree - 10	crash	crash	0.05375	3730	0.038874	0.979872	260.5996	0.034933	0.186904	7460	3730	1600	0.05375	1	152.0239	0.047507	0.217962	3200
Decision Tree - 15	crash	crash	0.05375	3730	0.038874	0.979872	260.5996	0.034933	0.186904	7460	3730	1600	0.05375	1	152.0239	0.047507	0.217962	3200
Decision Tree - 12	crash	crash	0.05375	3730	0.038874	0.979872	260.5996	0.034933	0.186904	7460	3730	1600	0.05375	1	152.0239	0.047507	0.217962	3200
Decision Tree - 20	crash	crash	0.05375	3730	0.038874	0.979872	260.5996	0.034933	0.186904	7460	3730	1600	0.05375	1	152.0239	0.047507	0.217962	3200
Decision Tree - 25	crash	crash	0.05375	3730	0.038874	0.979872	260.5996	0.034933	0.186904	7460	3730	1600	0.05375	1	152.0239	0.047507	0.217962	3200
Decision Tree - 8	crash	crash	0.054375	3730	0.045308	0.979872	304.5954	0.04083	0.202066	7460	3730	1600	0.054375	1	153.5605	0.047988	0.219061	3200
Decision Tree - 7	crash	crash	0.05625	3730	0.056836	0.979872	372.0376	0.049871	0.223318	7460	3730	1600	0.05625	0.979872	159.0083	0.04969	0.222913	3200
Decision Tree - 6	crash	crash	0.056875	3730	0.057641	0.979872	375.3042	0.050309	0.224296	7460	3730	1600	0.056875	0.979872	160.6261	0.050196	0.224044	3200
Decision Tree - 5	crash	crash	0.05875	3730	0.058981	0.946619	411.6873	0.055186	0.234917	7460	3730	1600	0.05875	0.946619	174.3278	0.054477	0.233404	3200

PYTHON

The data set contains text in the column “descriptions” for which bag-of-words model would be used that allows us to represent text as numerical feature vectors described in the following steps :-

i) A vocabulary of unique tokens is created from the entire set of documents

ii) A feature vector from each from each document that contains the counts of how often each word occurs in the particular document is created.

The unique word in each document represent only a small subset of all the words in the bag-of-words vocabulary, the feature vectors will consist of mostly zeros and hence the Term-Document matrix will be a sparse matrix.

For the bag-of-words analysis, the followings steps are carried out systematically:-

1. Tokenizing:
   This is the process of splitting the entire document into individual words. These tokens will be used in the further steps.

2. POS Tagging:
   Words in the text(Corpus) are tagged as corresponding to a particular parts of speech, based on both its definition and its context.

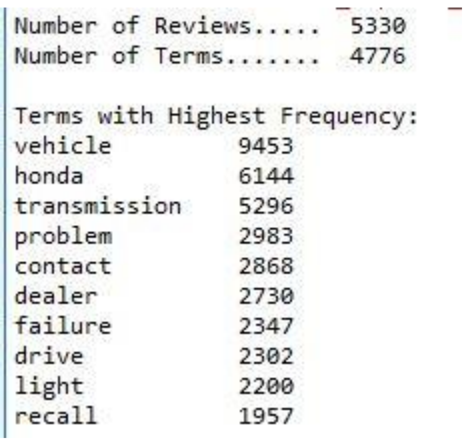
3. STOP Word Filter:
   The words that are extremely common in all sorts of documents and bear no to only little information needs to filtered. This is done by using the NLTK package which contains a list of Stop Words.

4. Stemming:
   The words are transformed into its root form that allows us to map related words to the same stem. Lemmatization is also used to obtain the grammatically correct form of individual words.

   These above steps are carried out by a user defined function which is customized here to give the optimum result based on the fact that the text is related to crash of an automobile so the knowledge of English language and human interaction with machine plays a vital role in customizing the user defined function “my analyzer”.

The next step is the Topic Analysis which identifies the number of important topics in the document. For Topic analysis, term document is created which can be done in two different ways:-

i) Raw term frequency-
   The raw term frequency matrix (n,d) represents the number of times a term n occurs in the document d. Raw term frequencies are larger for larger documents. “CountVectorizer” is imported form Sklearn package. Using this raw term frequency approach, it was found that the document contains 5330 reviews with 4776 terms and the terms occurring most frequently are shown below in the output from the print command when the ngram parameter was set to (1,2) which means extraction of pair of words as well as individual word:

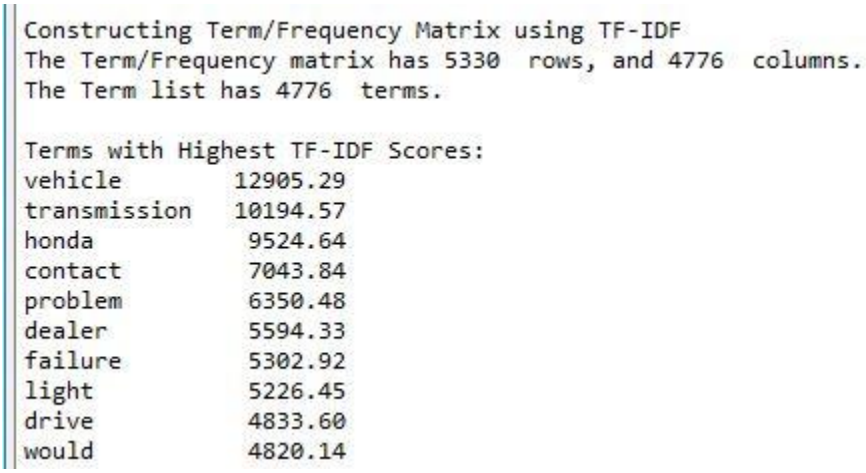


```
Number of Reviews.....  5330
Number of Terms.......  4776

Terms with Highest Frequency:
vehicle         9453
honda           6144
transmission    5296
problem         2983
contact         2868
dealer          2730
failure         2347
drive           2302
light           2200
recall          1957
```

But it observed that some of the words such as recall and dealer might not contain useful information but it is still included in the Term/Frequency matrix.

ii) Tf-Idf-

To counter the problem mentioned above, a useful technique called term frequency-inverse document frequency (tf-idf) was used which downweight the frequently occurring words which doesn't contain useful or discriminatory information. Essentially TF-IDF reduces the frequency of terms appearing in all or most of the documents while increasing the term frequency when the term appears in small number of documents. This is done by using the "TfidfTransformer" that takes the raw term frequencies from CountVectorizer as input and transforms them into tf-idfs. The output of the tf-idf matrix creation is shown below:



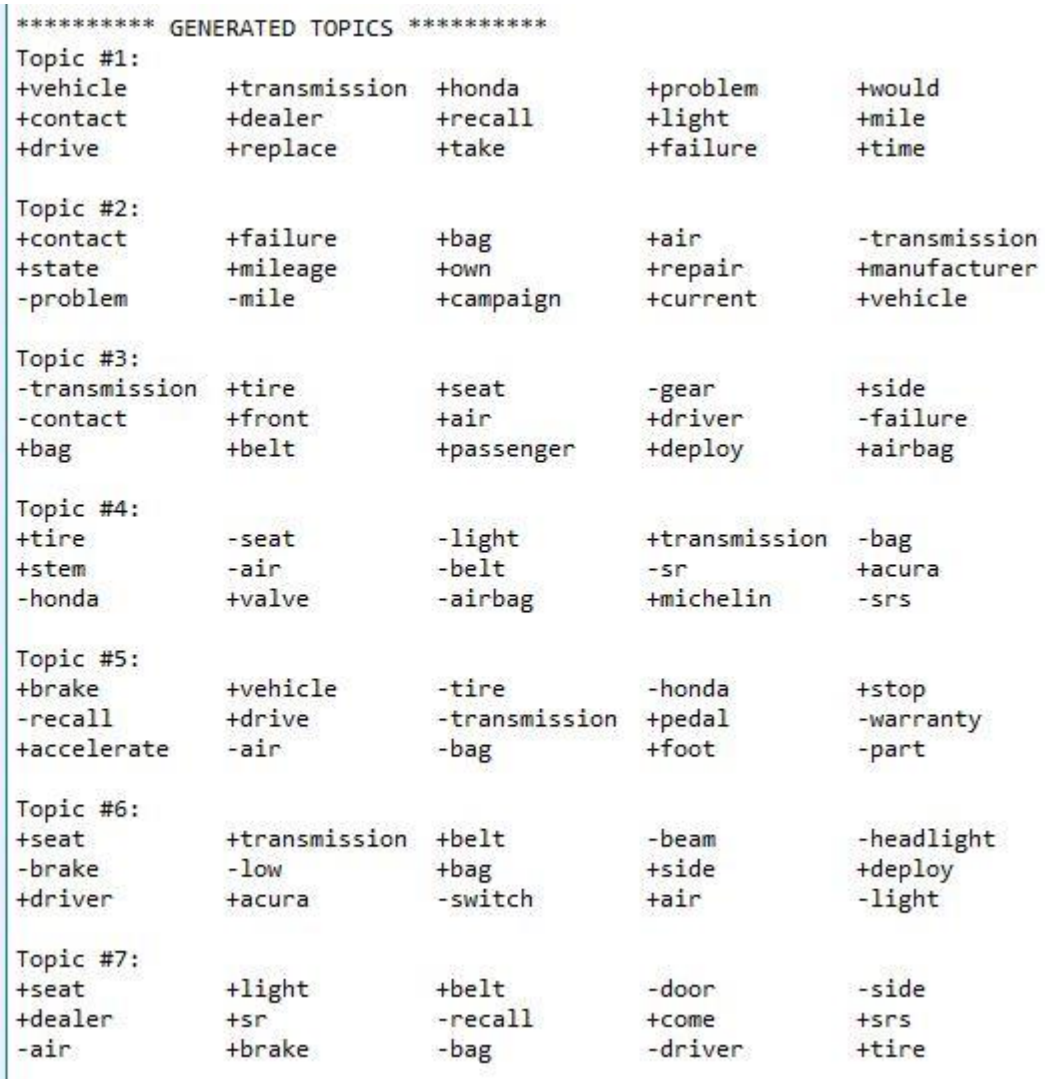
```
Constructing Term/Frequency Matrix using TF-IDF
The Term/Frequency matrix has 5330  rows, and 4776  columns.
The Term list has 4776  terms.

Terms with Highest TF-IDF Scores:
vehicle        12905.29
transmission   10194.57
honda           9524.64
contact         7043.84
problem         6350.48
dealer          5594.33
failure         5302.92
light           5226.45
drive           4833.60
would           4820.14
```

It can be observed from the TF-IDF matrix that the TF-IDF score of the same term is different as compared to the raw term frequency matrix created by using CountVectorizer because of the different weights placed on the terms based on their appearance in the number of documents.

Topic Analysis:-

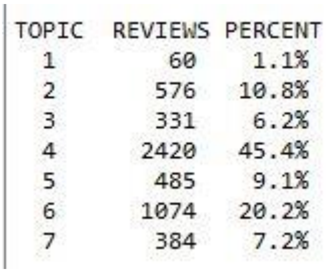
The next most important step is identifying the number of important topics found in the body of the document or grouping together important words to make cluster which can be used for analysis. This is done by decomposing the matrix created above by the method of Singular Value Decomposition. SVD on the TFIDF matrix is referred to as LSA, *Latent Semantic Analysis.*

The complete document is decomposed into 7 different topics with 15 terms in each topic which is shown below:



```
********** GENERATED TOPICS **********
Topic #1:
+vehicle       +transmission  +honda         +problem       +would
+contact       +dealer        +recall        +light         +mile
+drive         +replace       +take          +failure       +time

Topic #2:
+contact       +failure       +bag           +air           -transmission
+state         +mileage       +own           +repair        +manufacturer
-problem       -mile          +campaign      +current       +vehicle

Topic #3:
-transmission  +tire          +seat          -gear          +side
-contact       +front         +air           +driver        -failure
+bag           +belt          +passenger     +deploy        +airbag

Topic #4:
+tire          -seat          -light         +transmission  -bag
+stem          -air           -belt          -sr            +acura
-honda         +valve         -airbag        +michelin      -srs

Topic #5:
+brake         +vehicle       -tire          -honda         +stop
-recall        +drive         -transmission  +pedal         -warranty
+accelerate    -air           -bag           +foot          -part

Topic #6:
+seat          +transmission  +belt          -beam          -headlight
-brake         -low           +bag           +side          +deploy
+driver        +acura         -switch        +air           -light

Topic #7:
+seat          +light         +belt          -door          -side
+dealer        +sr            -recall        +come          +srs
-air           +brake         -bag           -driver        +tire
```

The scores of each topic in the number of reviews/documents can be shown below:



```
TOPIC  REVIEWS PERCENT
  1        60    1.1%
  2       576   10.8%
  3       331    6.2%
  4      2420   45.4%
  5       485    9.1%
  6      1074   20.2%
  7       384    7.2%
```

It can be concluded that the 4th topic appears in most of the reviews/documents which contains words such as tire, seat, airbag, light etc which are definitely related to crash/accident as there might be light failure at night or a flat tire. Therefore it can be concluded that the important topics have been obtained from the crash review data set by preprocessing data using tokenizing, stemming, lemmatization, filtering stop words then creating a term/document frequency matrix and finally decomposing the matrix.

These topic scores were stored in the topic attribute T1-T7 which is then added to the data frame. Finally topic score attributes would be used further in building the predictive model for crash report.

Sentiment Analysis:-

Sentiment analysis is measuring the emotional sentiment in documents or reviews. Sentiment analysis is only concerned with terms that are known to carry emotional content, either positive or negative.

Sentiment analysis identifies and counts the sentiment words found in a document and then scores them to develop a score that on average reflects the emotional content of a document. High sentiment scores represent positive emotional content and low score represent negative emotional content.

For doing such analysis, a list of words prepared by A. Finn is imported in the Python to compare. This list contains scores ranging from -5 to +5 for all the sentiment words.

For capturing the sentiment words and their points, Dictionary was created in Python which has sentiment words as their keys and points as their values. The term/frequency matrix was already created during topic analysis.

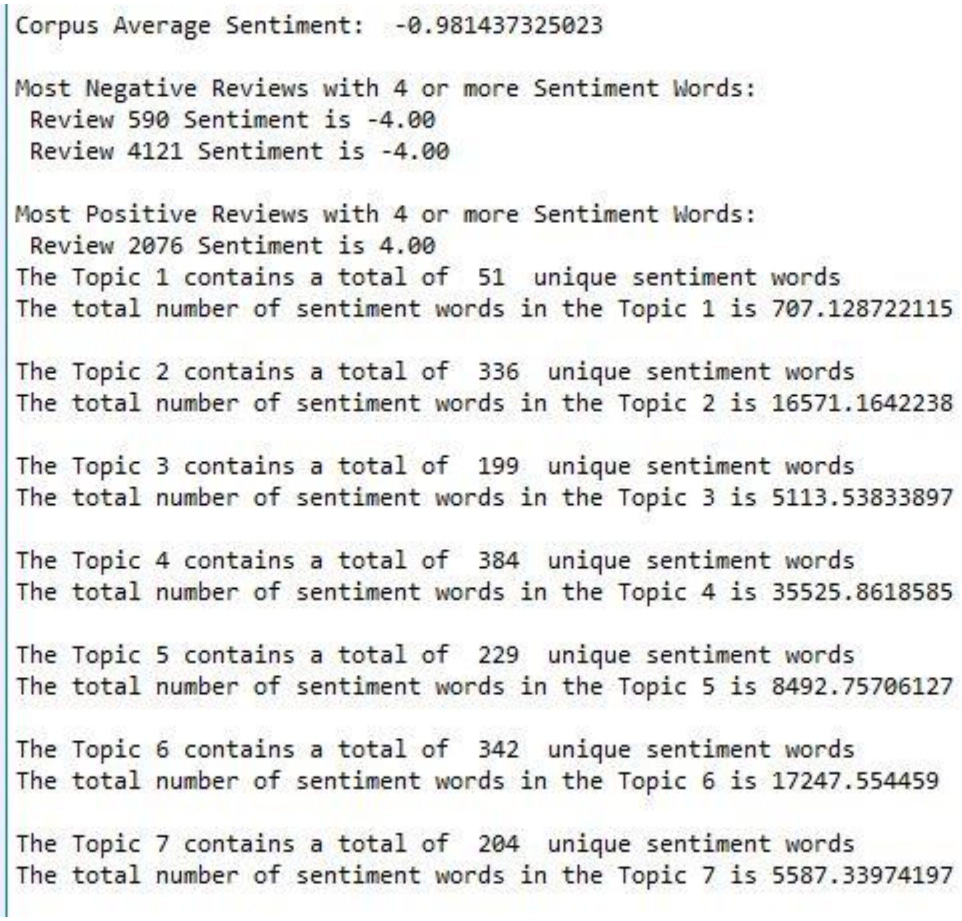
After creating term/frequency matrix along with the dictionary of sentiment words, each document is scored.

Firstly, the average sentiment of the corpus is calculated i.e the overall sentiment of the complete document which comes out to be -0.981 which was quite obvious since it is related to automobile crash, so there would be dissatisfaction/unhappiness among the consumers which is perfectly reflected by the average negative sentiment score.

Also, the review with the most negative and most positive sentiment scores with at least 4 sentiment words were observed. It was found that review 590 and 4121 were the most negative review with a sentiment score of -4.0 and the review number 2076 was the most positive review with a sentiment score of +4.0.

Moreover, the number of sentiment words in each topics along with the number of unique sentiment words contained in each topic was also observed to find out the most important topic. From the sentiment analysis of each topic, it was observed that topic 4 has the highest number of sentiment words along with 384 unique sentiment words making it the most important topic which can be used for model building. This aligns with the result obtained from the topic analysis scores which shows that topic 4 is in 2420.

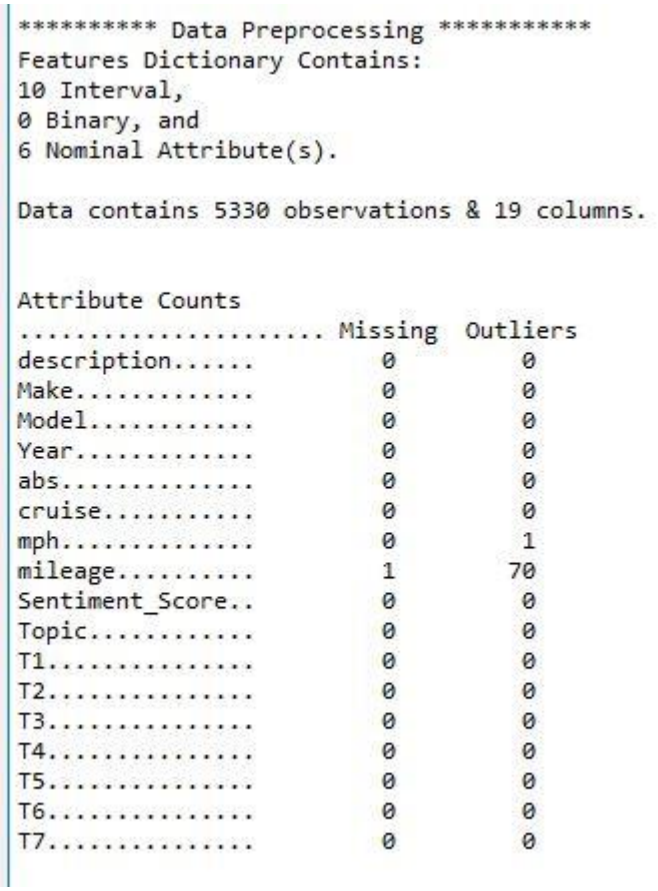
The output of the sentiment analysis is shown below:



```
Corpus Average Sentiment:  -0.981437325023

Most Negative Reviews with 4 or more Sentiment Words:
 Review 590 Sentiment is -4.00
 Review 4121 Sentiment is -4.00

Most Positive Reviews with 4 or more Sentiment Words:
 Review 2076 Sentiment is 4.00
The Topic 1 contains a total of  51  unique sentiment words
The total number of sentiment words in the Topic 1 is 707.128722115

The Topic 2 contains a total of  336  unique sentiment words
The total number of sentiment words in the Topic 2 is 16571.1642238

The Topic 3 contains a total of  199  unique sentiment words
The total number of sentiment words in the Topic 3 is 5113.53833897

The Topic 4 contains a total of  384  unique sentiment words
The total number of sentiment words in the Topic 4 is 35525.8618585

The Topic 5 contains a total of  229  unique sentiment words
The total number of sentiment words in the Topic 5 is 8492.75706127

The Topic 6 contains a total of  342  unique sentiment words
The total number of sentiment words in the Topic 6 is 17247.554459

The Topic 7 contains a total of  204  unique sentiment words
The total number of sentiment words in the Topic 7 is 5587.33974197
```

Decision Tree Model:-

The Decision Tree Classifier was finally used to predict the probability of crash using these Topic Analysis and Sentiment Analysis results along with other predictors such as mph, mileage, cruise etc. The data set was split into 70/30 ratio and the model was trained on 70% of the observations while the accuracy and other metric values were calculate on the remaining test data to obtain the results.



```
********** Data Preprocessing ***********
Features Dictionary Contains:
10 Interval,
0 Binary, and
6 Nominal Attribute(s).

Data contains 5330 observations & 19 columns.


Attribute Counts
...................... Missing  Outliers
description......        0        0
Make.............        0        0
Model............        0        0
Year.............        0        0
abs..............        0        0
cruise...........        0        0
mph..............        0        1
mileage..........        1       70
Sentiment_Score..        0        0
Topic............        0        0
T1...............        0        0
T2...............        0        0
T3...............        0        0
T4...............        0        0
T5...............        0        0
T6...............        0        0
T7...............        0        0
```

For obtaining the best classification model, tuning parameter, maximum depth of the tree was varied from 5-25 and cross validation applied to obtain the best results.

Web Scraping:-

Web Scraping is technique employed to extract large amount of data from websites and saving it to a local file in the computer or to a database in a table. Here news articles from different sources such as Huffington Post, Reuters, CBS News etc are downloaded and 'TAKATA' search word is used to extract useful information. Following are the steps used for Web Scraping:-

i) Newspaper, newsapi and requests packages are downloaded using the pip commands.

ii) Then a dictionary is created containing the URLs for different agencies such as Huffington Post, Reuters, CBS News etc used by the API news feed package.

iii) After downloading the raw HTML files, it is cleaned and the HTML markups removed.

iv) After preprocessing the data by removing the HTML markups, comment tags, regular tags and dealing with white spaces, it is translated to ordinary text files for topic and sentiment analysis.

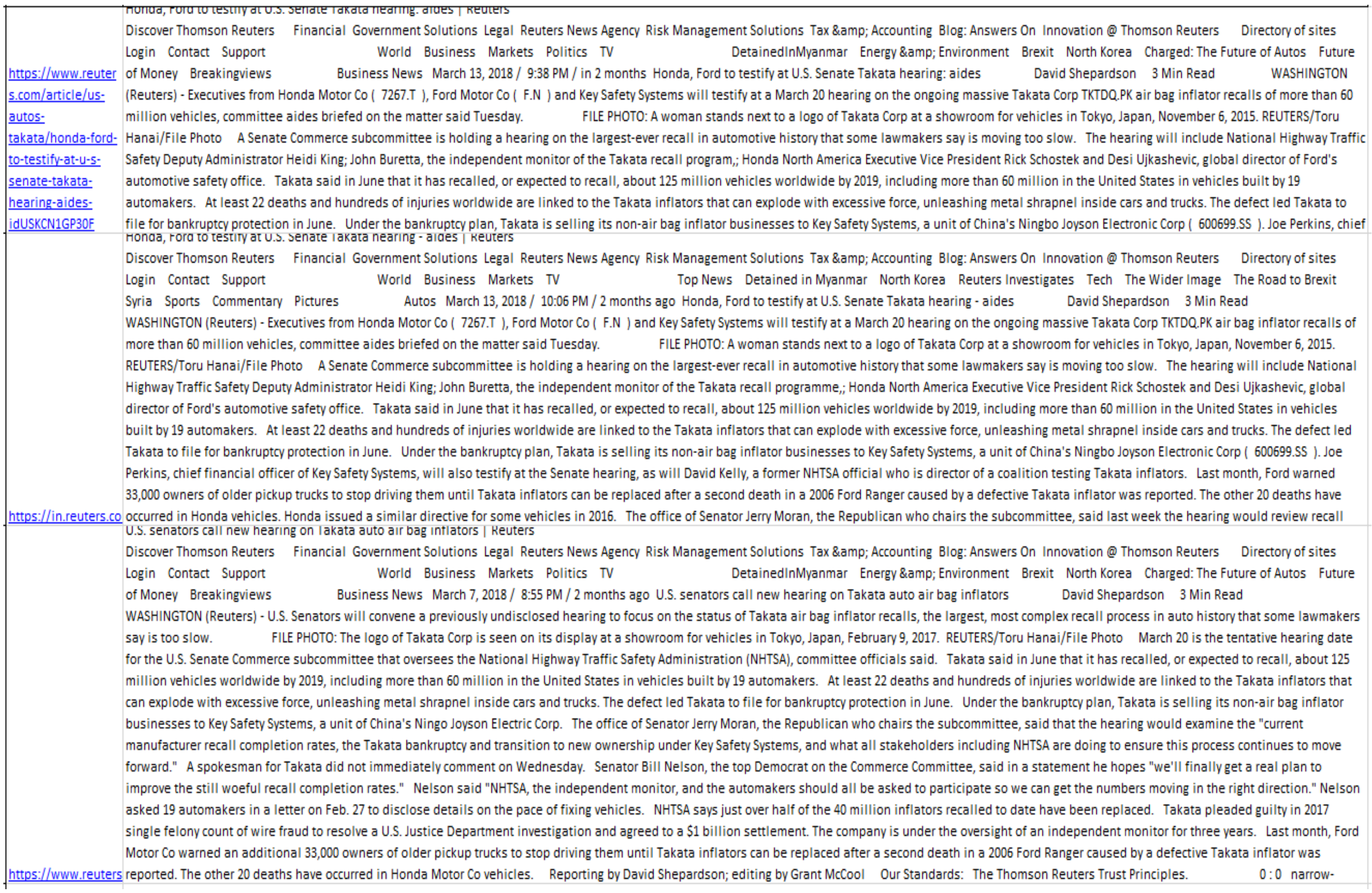
v) Using the search word 'TAKATA', all the files downloaded from the News Agency were read and the output returns the document containing the word 'TAKATA'.

vi) It is found that 65 out of total URLS contains the word 'TAKATA' out of which 62 were unique.

vii) However, this number is too low to justify a topic/sentiment analysis but these articles surely do relate to the topic groups.

viii) After reading the article, it was found that due to the failure of airbags manufactured by 'TAKATA', an automotive company based in Japan, there was a lot of automobile accidents leading to death/ severe injury of passengers which led to largest recall in Automotive history.

ix) Therefore, we can conclude that the sentiment and topic analysis of this web scraping can provide useful information provided it contains significant number of documents.

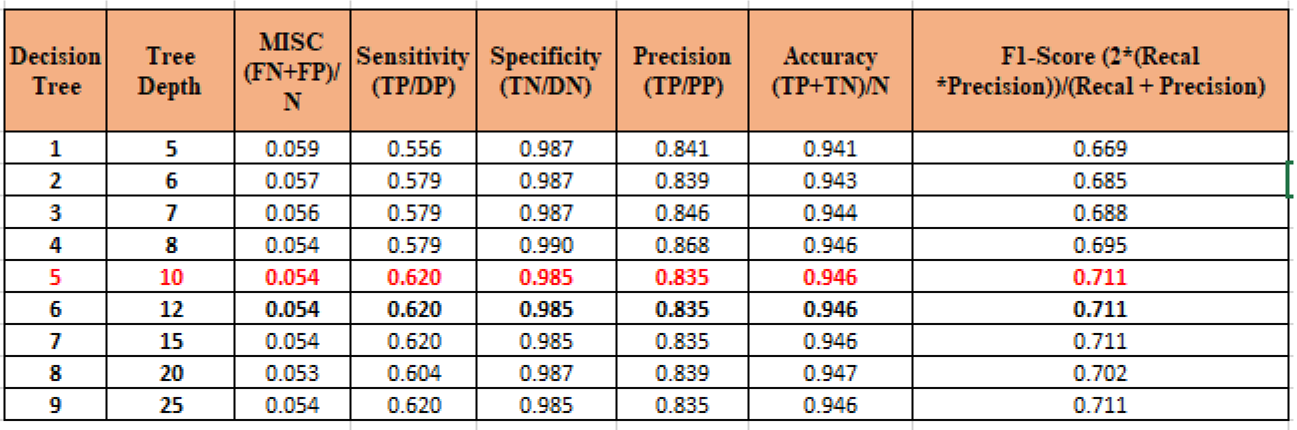


https://www.reuters.com/article/us-autos-takata/honda-ford-to-testify-at-u-s-senate-takata-hearing-aides-idUSKCN1GP30F	Honda, Ford to testify at U.S. Senate Takata hearing: aides | Reuters Discover Thomson Reuters Financial Government Solutions Legal Reuters News Agency Risk Management Solutions Tax &amp; Accounting Blog: Answers On Innovation @ Thomson Reuters Directory of sites Login Contact Support World Business Markets Politics TV DetainedInMyanmar Energy &amp; Environment Brexit North Korea Charged: The Future of Autos Future of Money Breakingviews Business News March 13, 2018 / 9:38 PM / in 2 months Honda, Ford to testify at U.S. Senate Takata hearing: aides David Shepardson 3 Min Read WASHINGTON (Reuters) - Executives from Honda Motor Co (7267.T), Ford Motor Co (F.N) and Key Safety Systems will testify at a March 20 hearing on the ongoing massive Takata Corp TKTDQ.PK air bag inflator recalls of more than 60 million vehicles, committee aides briefed on the matter said Tuesday. FILE PHOTO: A woman stands next to a logo of Takata Corp at a showroom for vehicles in Tokyo, Japan, November 6, 2015. REUTERS/Toru Hanai/File Photo A Senate Commerce subcommittee is holding a hearing on the largest-ever recall in automotive history that some lawmakers say is moving too slow. The hearing will include National Highway Traffic Safety Deputy Administrator Heidi King; John Buretta, the independent monitor of the Takata recall program,; Honda North America Executive Vice President Rick Schostek and Desi Ujkashevic, global director of Ford's automotive safety office. Takata said in June that it has recalled, or expected to recall, about 125 million vehicles worldwide by 2019, including more than 60 million in the United States in vehicles built by 19 automakers. At least 22 deaths and hundreds of injuries worldwide are linked to the Takata inflators that can explode with excessive force, unleashing metal shrapnel inside cars and trucks. The defect led Takata to file for bankruptcy protection in June. Under the bankruptcy plan, Takata is selling its non-air bag inflator businesses to Key Safety Systems, a unit of China's Ningbo Joyson Electronic Corp (600699.SS). Joe Perkins, chief
https://in.reuters.co	Honda, Ford to testify at U.S. Senate Takata hearing - aides | Reuters Discover Thomson Reuters Financial Government Solutions Legal Reuters News Agency Risk Management Solutions Tax &amp; Accounting Blog: Answers On Innovation @ Thomson Reuters Directory of sites Login Contact Support World Business Markets TV Top News Detained in Myanmar North Korea Reuters Investigates Tech The Wider Image The Road to Brexit Syria Sports Commentary Pictures Autos March 13, 2018 / 10:06 PM / 2 months ago Honda, Ford to testify at U.S. Senate Takata hearing - aides David Shepardson 3 Min Read WASHINGTON (Reuters) - Executives from Honda Motor Co (7267.T), Ford Motor Co (F.N) and Key Safety Systems will testify at a March 20 hearing on the ongoing massive Takata Corp TKTDQ.PK air bag inflator recalls of more than 60 million vehicles, committee aides briefed on the matter said Tuesday. FILE PHOTO: A woman stands next to a logo of Takata Corp at a showroom for vehicles in Tokyo, Japan, November 6, 2015. REUTERS/Toru Hanai/File Photo A Senate Commerce subcommittee is holding a hearing on the largest-ever recall in automotive history that some lawmakers say is moving too slow. The hearing will include National Highway Traffic Safety Deputy Administrator Heidi King; John Buretta, the independent monitor of the Takata recall programme,; Honda North America Executive Vice President Rick Schostek and Desi Ujkashevic, global director of Ford's automotive safety office. Takata said in June that it has recalled, or expected to recall, about 125 million vehicles worldwide by 2019, including more than 60 million in the United States in vehicles built by 19 automakers. At least 22 deaths and hundreds of injuries worldwide are linked to the Takata inflators that can explode with excessive force, unleashing metal shrapnel inside cars and trucks. The defect led Takata to file for bankruptcy protection in June. Under the bankruptcy plan, Takata is selling its non-air bag inflator businesses to Key Safety Systems, a unit of China's Ningbo Joyson Electronic Corp (600699.SS). Joe Perkins, chief financial officer of Key Safety Systems, will also testify at the Senate hearing, as will David Kelly, a former NHTSA official who is director of a coalition testing Takata inflators. Last month, Ford warned 33,000 owners of older pickup trucks to stop driving them until Takata inflators can be replaced after a second death in a 2006 Ford Ranger caused by a defective Takata inflator was reported. The other 20 deaths have occurred in Honda vehicles. Honda issued a similar directive for some vehicles in 2016. The office of Senator Jerry Moran, the Republican who chairs the subcommittee, said last week the hearing would review recall
https://www.reuters	U.S. senators call new hearing on Takata auto air bag inflators | Reuters Discover Thomson Reuters Financial Government Solutions Legal Reuters News Agency Risk Management Solutions Tax &amp; Accounting Blog: Answers On Innovation @ Thomson Reuters Directory of sites Login Contact Support World Business Markets Politics TV DetainedInMyanmar Energy &amp; Environment Brexit North Korea Charged: The Future of Autos Future of Money Breakingviews Business News March 7, 2018 / 8:55 PM / 2 months ago U.S. senators call new hearing on Takata auto air bag inflators David Shepardson 3 Min Read WASHINGTON (Reuters) - U.S. Senators will convene a previously undisclosed hearing to focus on the status of Takata air bag inflator recalls, the largest, most complex recall process in auto history that some lawmakers say is too slow. FILE PHOTO: The logo of Takata Corp is seen on its display at a showroom for vehicles in Tokyo, Japan, February 9, 2017. REUTERS/Toru Hanai/File Photo March 20 is the tentative hearing date for the U.S. Senate Commerce subcommittee that oversees the National Highway Traffic Safety Administration (NHTSA), committee officials said. Takata said in June that it has recalled, or expected to recall, about 125 million vehicles worldwide by 2019, including more than 60 million in the United States in vehicles built by 19 automakers. At least 22 deaths and hundreds of injuries worldwide are linked to the Takata inflators that can explode with excessive force, unleashing metal shrapnel inside cars and trucks. The defect led Takata to file for bankruptcy protection in June. Under the bankruptcy plan, Takata is selling its non-air bag inflator businesses to Key Safety Systems, a unit of China's Ningo Joyson Electric Corp. The office of Senator Jerry Moran, the Republican who chairs the subcommittee, said that the hearing would examine the "current manufacturer recall completion rates, the Takata bankruptcy and transition to new ownership under Key Safety Systems, and what all stakeholders including NHTSA are doing to ensure this process continues to move forward." A spokesman for Takata did not immediately comment on Wednesday. Senator Bill Nelson, the top Democrat on the Commerce Committee, said in a statement he hopes "we'll finally get a real plan to improve the still woeful recall completion rates." Nelson said "NHTSA, the independent monitor, and the automakers should all be asked to participate so we can get the numbers moving in the right direction." Nelson asked 19 automakers in a letter on Feb. 27 to disclose details on the pace of fixing vehicles. NHTSA says just over half of the 40 million inflators recalled to date have been replaced. Takata pleaded guilty in 2017 single felony count of wire fraud to resolve a U.S. Justice Department investigation and agreed to a $1 billion settlement. The company is under the oversight of an independent monitor for three years. Last month, Ford Motor Co warned an additional 33,000 owners of older pickup trucks to stop driving them until Takata inflators can be replaced after a second death in a 2006 Ford Ranger caused by a defective Takata inflator was reported. The other 20 deaths have occurred in Honda Motor Co vehicles. Reporting by David Shepardson; editing by Grant McCool Our Standards: The Thomson Reuters Trust Principles. 0 : 0 narrow-

Section 4: Results:

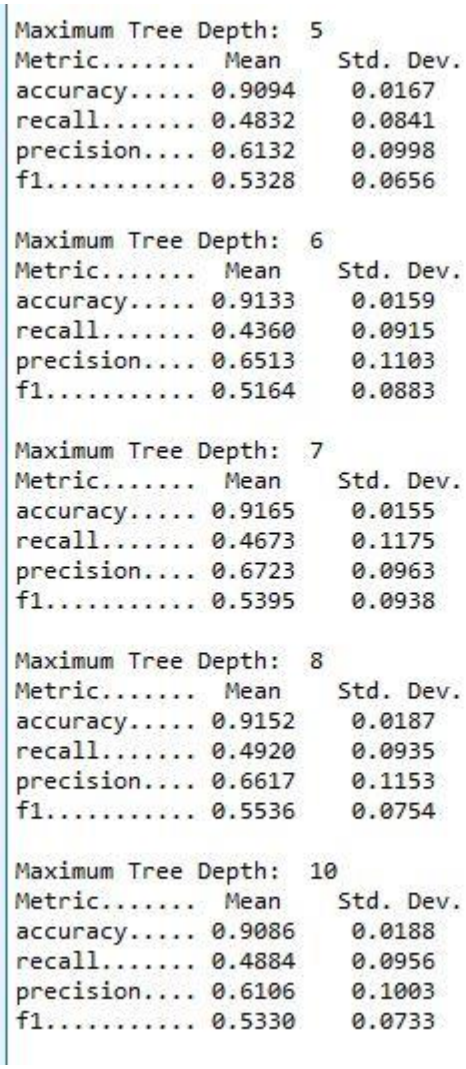
The accuracy, recall, precision and F1 values of various decision tree models are shown below:-

SAS EM

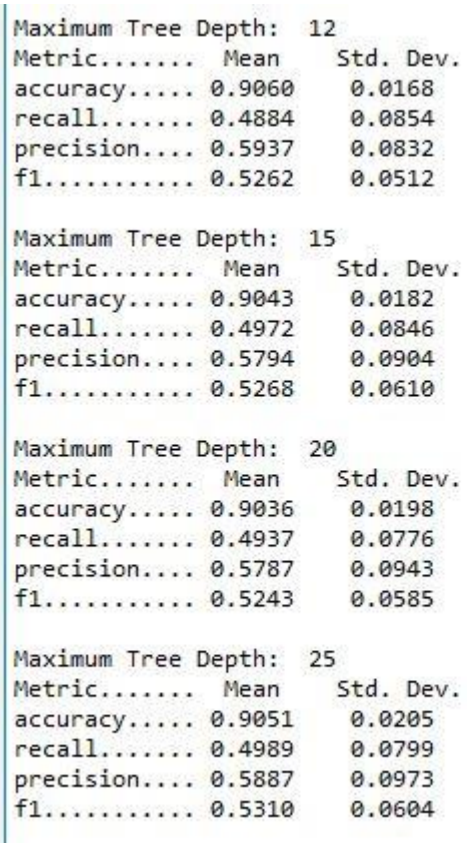


Decision Tree	Tree Depth	MISC (FN+FP)/ N	Sensitivity (TP/DP)	Specificity (TN/DN)	Precision (TP/PP)	Accuracy (TP+TN)/N	F1-Score (2*(Recal *Precision))/(Recal + Precision)
1	5	0.059	0.556	0.987	0.841	0.941	0.669
2	6	0.057	0.579	0.987	0.839	0.943	0.685
3	7	0.056	0.579	0.987	0.846	0.944	0.688
4	8	0.054	0.579	0.990	0.868	0.946	0.695
5	10	0.054	0.620	0.985	0.835	0.946	0.711
6	12	**0.054**	**0.620**	**0.985**	**0.835**	**0.946**	**0.711**
7	15	0.054	0.620	0.985	0.835	0.946	0.711
8	20	0.053	0.604	0.987	0.839	0.947	0.702
9	25	0.054	0.620	0.985	0.835	0.946	0.711

PYTHON



```
Maximum Tree Depth:  5
Metric.......  Mean    Std. Dev.
accuracy..... 0.9094    0.0167
recall....... 0.4832    0.0841
precision.... 0.6132    0.0998
f1........... 0.5328    0.0656

Maximum Tree Depth:  6
Metric.......  Mean    Std. Dev.
accuracy..... 0.9133    0.0159
recall....... 0.4360    0.0915
precision.... 0.6513    0.1103
f1........... 0.5164    0.0883

Maximum Tree Depth:  7
Metric.......  Mean    Std. Dev.
accuracy..... 0.9165    0.0155
recall....... 0.4673    0.1175
precision.... 0.6723    0.0963
f1........... 0.5395    0.0938

Maximum Tree Depth:  8
Metric.......  Mean    Std. Dev.
accuracy..... 0.9152    0.0187
recall....... 0.4920    0.0935
precision.... 0.6617    0.1153
f1........... 0.5536    0.0754

Maximum Tree Depth:  10
Metric.......  Mean    Std. Dev.
accuracy..... 0.9086    0.0188
recall....... 0.4884    0.0956
precision.... 0.6106    0.1003
f1........... 0.5330    0.0733
```



```
Maximum Tree Depth:  12
Metric.......  Mean    Std. Dev.
accuracy..... 0.9060    0.0168
recall....... 0.4884    0.0854
precision.... 0.5937    0.0832
f1........... 0.5262    0.0512

Maximum Tree Depth:  15
Metric.......  Mean    Std. Dev.
accuracy..... 0.9043    0.0182
recall....... 0.4972    0.0846
precision.... 0.5794    0.0904
f1........... 0.5268    0.0610

Maximum Tree Depth:  20
Metric.......  Mean    Std. Dev.
accuracy..... 0.9036    0.0198
recall....... 0.4937    0.0776
precision.... 0.5787    0.0943
f1........... 0.5243    0.0585

Maximum Tree Depth:  25
Metric.......  Mean    Std. Dev.
accuracy..... 0.9051    0.0205
recall....... 0.4989    0.0799
precision.... 0.5887    0.0973
f1........... 0.5310    0.0604
```

Section 5: Observations & Conclusion:

SAS EM

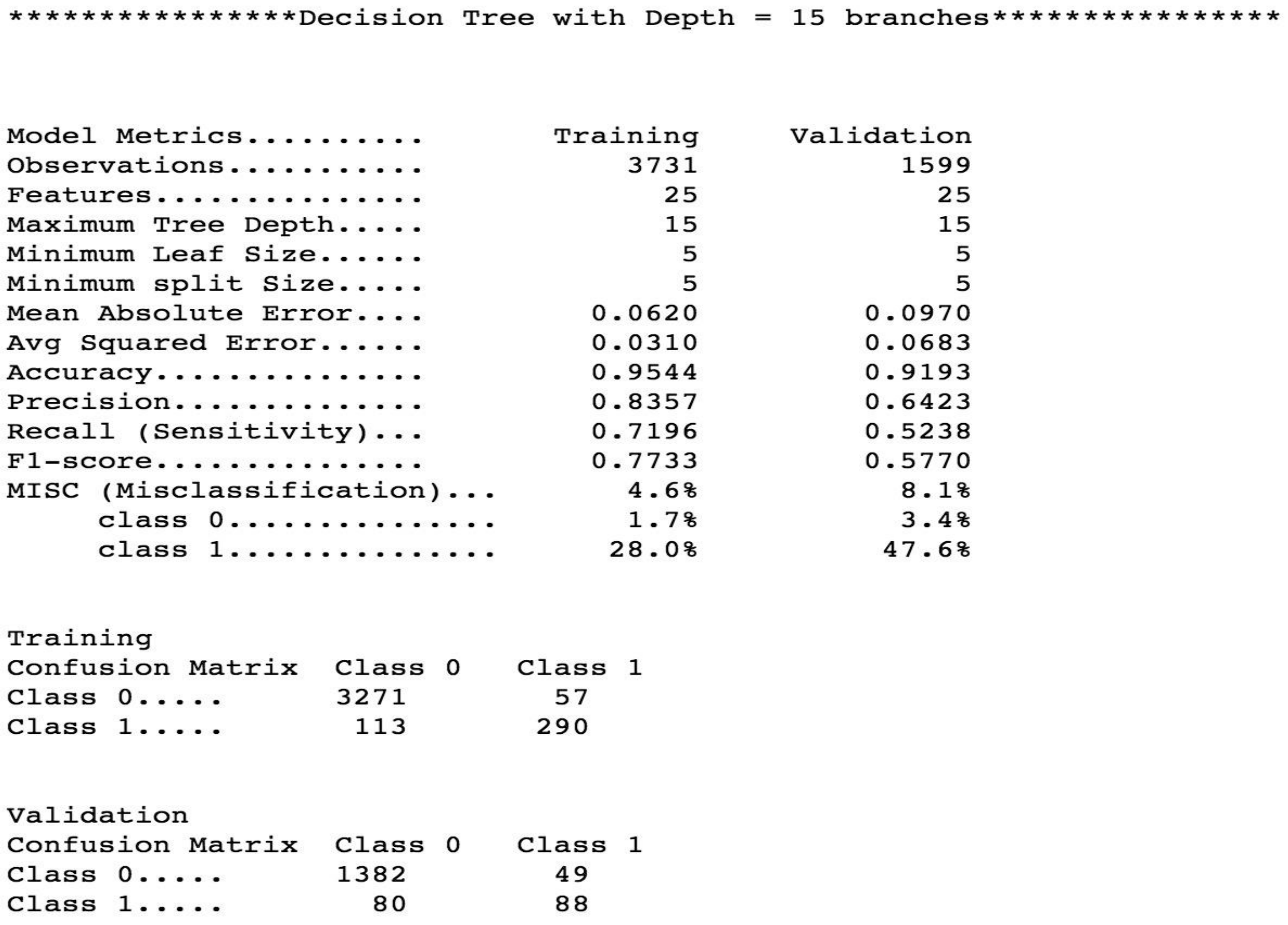
The values for confusion matrix of the best Decision Tree model in SAS EM is shown below:

	Prediction		
Data	Negative	Positive	Total
Negative	1408	21	1429
Positive	65	106	171
Total	1473	127	1600

From the table, metric values were calculated and the Decision tree with Depth =10 has a Misclassification error rate of 0.054, Sensitivity = 0.620, Specificity = 0.985, Accuracy = 0.946 and F1 score = 0.711.

PYTHON

It can be observed that the decision tree with depth=8 has Accuracy= 0.9193, Recall= 0.5238, Precision= 0.6423 and F1= 0.5770 gives the best result accurately predicting a crash 91.99% of the time.



```
****************Decision Tree with Depth = 15 branches****************



Model Metrics..........          Training      Validation
Observations...........              3731            1599
Features...............                25              25
Maximum Tree Depth.....                15              15
Minimum Leaf Size......                 5               5
Minimum split Size.....                 5               5
Mean Absolute Error....            0.0620          0.0970
Avg Squared Error......            0.0310          0.0683
Accuracy...............            0.9544          0.9193
Precision..............            0.8357          0.6423
Recall (Sensitivity)...            0.7196          0.5238
F1-score...............            0.7733          0.5770
MISC (Misclassification)...          4.6%            8.1%
     class 0...............          1.7%            3.4%
     class 1...............         28.0%           47.6%


Training
Confusion Matrix  Class 0   Class 1
Class 0.....       3271        57
Class 1.....        113       290


Validation
Confusion Matrix  Class 0   Class 1
Class 0.....       1382        49
Class 1.....         80        88
```

Comparison between SAS EM & Python Results

	SAS EM	Python
MISC	0.054	0.081
Sensitivity	0.62	0.5238
Precision	0.835	0.6423
Accuracy	0.946	0.9193
F1	0.711	0.577

However, both the SAS EM & Python provides comparable results, SAS EM is the winner in this case giving better accuracy in predicting the probability of crash.

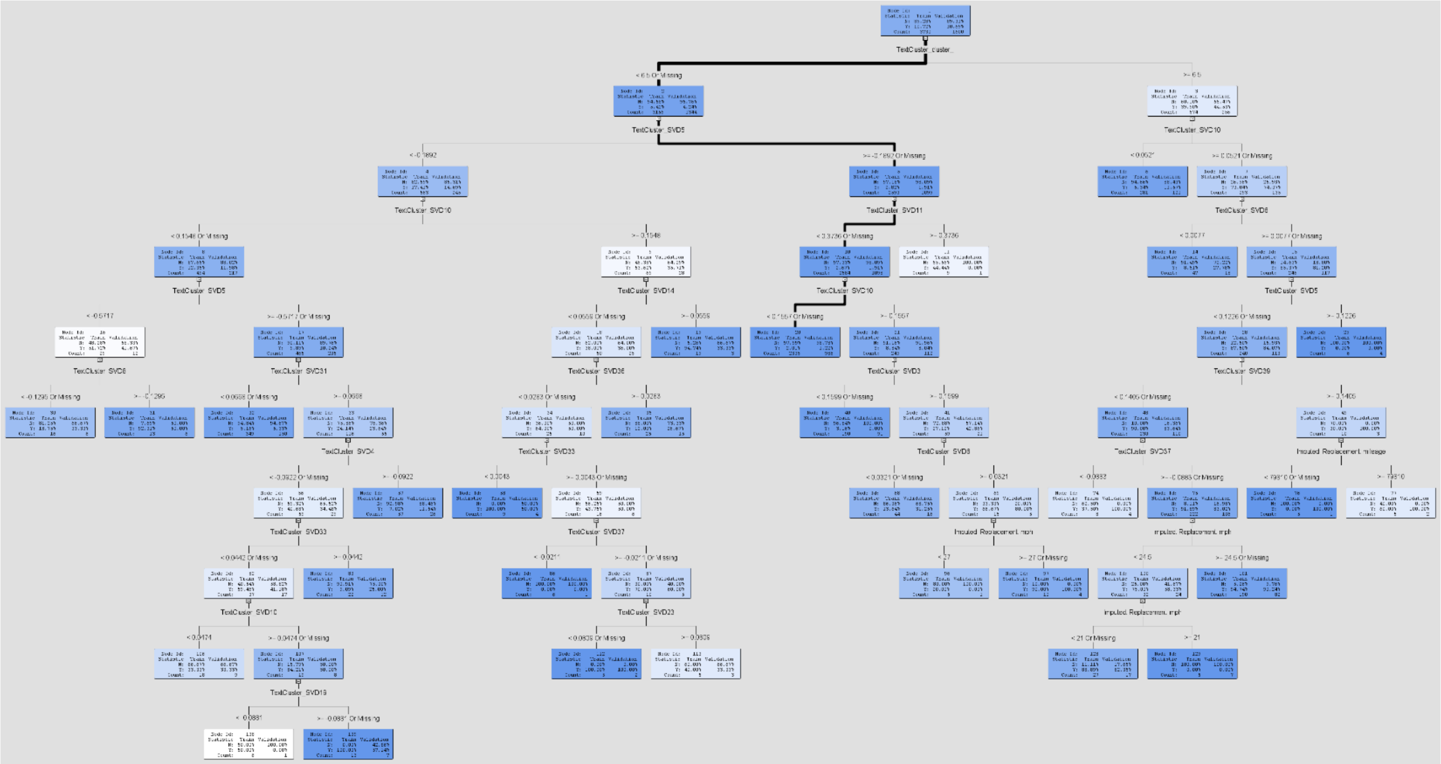
Summary:-

The project presents to us the data collected from consumer reviews for Honda by NHTSA and a Decision Tree Classifier was required to be built both in SAS EM & Python that can predict the probability of crash. Along with the various concepts used in numerical data analysis, bag-of-words machine learning algorithms are also utilized to build and validate the classification model. The text document was preprocessed in which concepts of Tokenizing, Parts of Speech Tagging, Stemming, Lemmatization were used. Term/document frequency matrix was created after preprocessing and finally it was decomposed to obtain the topic clusters which would be used as an attribute in modelling. Also the concepts of sentiment analysis and Web Scraping were utilized. Although sentiment analysis was used directly in model fitting as the negative sentiment review from the consumer can help us in predicting a crash and the same can be utilized by extracting data from various News Agency containing the word 'TAKATA' in their documents. Although, only 62 documents contain the useful information and this number is too low to justify a topic or sentiment analysis but these articles are related to the topic groups as it is seen that the failure of airbags manufactured by 'TAKATA' led to accidents and deaths of consumers.

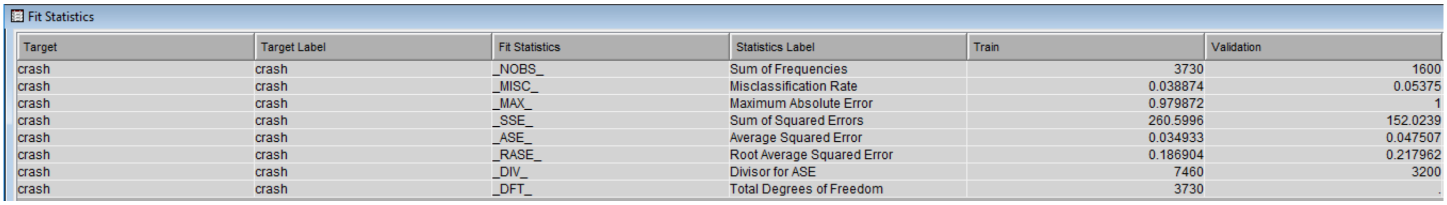
Finally, the Decision tree Classifier was built using the Topic clusters, Sentiment analysis and other attributes such as mileage, cruise, abs etc. Maximum depth parameter of the Decision tree was tuned to optimize the result and using cross validation approach it was found that the best Decision Tree was the one with maximum depth=15, minimum samples split= 5 and minimum samples split = 5 with and accuracy of 91.93% in Python and the Decision tree with maximum depth = 10 provides the accuracy of 94.6% in predicting the probability of Crash in SAS EM.

Appendix

Tree Diagram of tree depth of 10



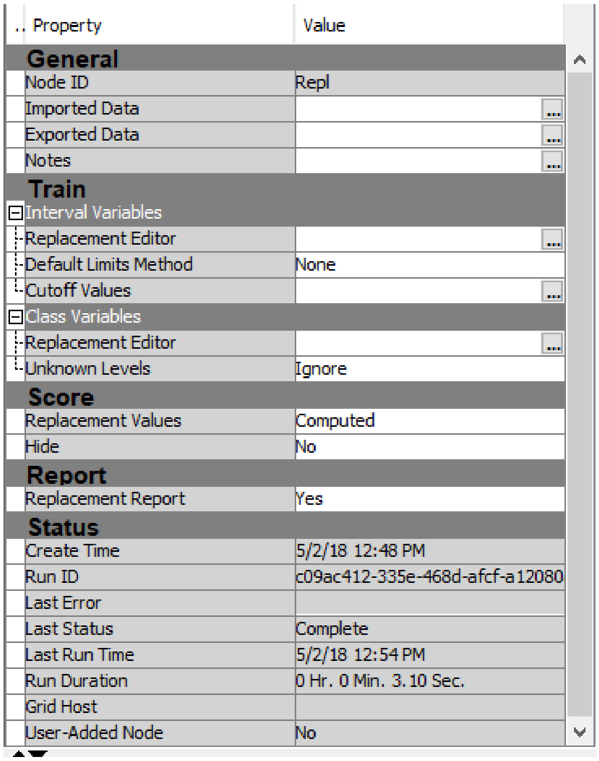
Fit statistics of decision tree of depth 10 (Best Model)


Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation
crash	crash	_NOBS_	Sum of Frequencies	3730	1600
crash	crash	_MISC_	Misclassification Rate	0.038874	0.05375
crash	crash	_MAX_	Maximum Absolute Error	0.979872	1
crash	crash	_SSE_	Sum of Squared Errors	260.5996	152.0239
crash	crash	_ASE_	Average Squared Error	0.034933	0.047507
crash	crash	_RASE_	Root Average Squared Error	0.186904	0.217962
crash	crash	_DIV_	Divisor for ASE	7460	3200
crash	crash	_DFT_	Total Degrees of Freedom	3730	.

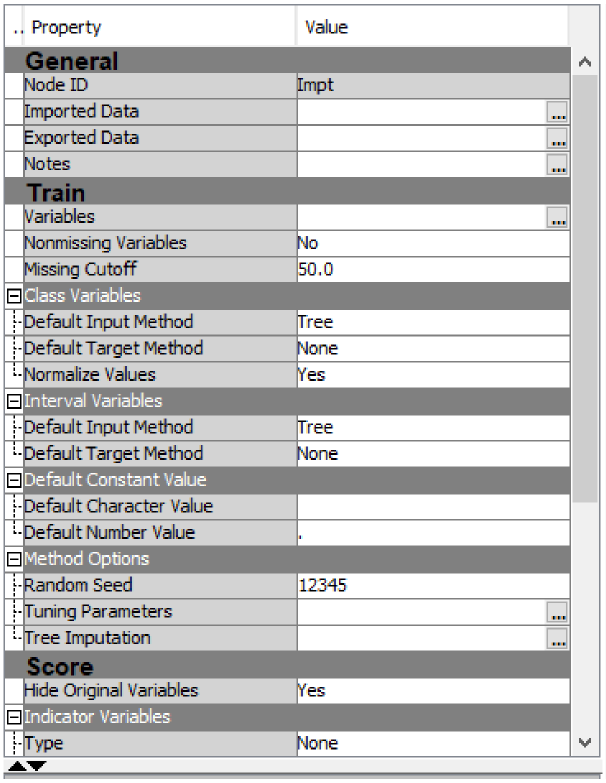
Properties window of SAS EM nodes

Replacement node



Property	Value
General	
Node ID	Repl
Imported Data	...
Exported Data	...
Notes	...
Train	
⊟ Interval Variables	
Replacement Editor	...
Default Limits Method	None
Cutoff Values	...
⊟ Class Variables	
Replacement Editor	...
Unknown Levels	Ignore
Score	
Replacement Values	Computed
Hide	No
Report	
Replacement Report	Yes
Status	
Create Time	5/2/18 12:48 PM
Run ID	c09ac412-335e-468d-afcf-a12080
Last Error	
Last Status	Complete
Last Run Time	5/2/18 12:54 PM
Run Duration	0 Hr. 0 Min. 3.10 Sec.
Grid Host	
User-Added Node	No

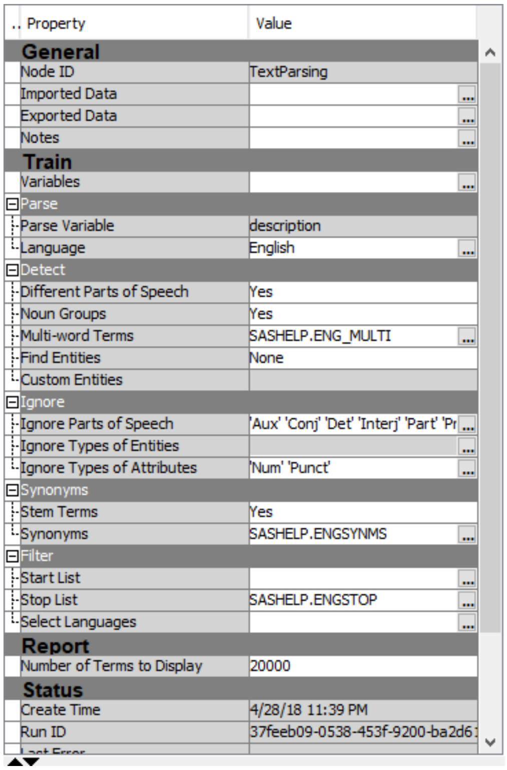
Impute node



Property	Value
General	
Node ID	Impt
Imported Data	...
Exported Data	...
Notes	...
Train	
Variables	...
Nonmissing Variables	No
Missing Cutoff	50.0
⊟ Class Variables	
Default Input Method	Tree
Default Target Method	None
Normalize Values	Yes
⊟ Interval Variables	
Default Input Method	Tree
Default Target Method	None
⊟ Default Constant Value	
Default Character Value	
Default Number Value	.
⊟ Method Options	
Random Seed	12345
Tuning Parameters	...
Tree Imputation	...
Score	
Hide Original Variables	Yes
⊟ Indicator Variables	
Type	None

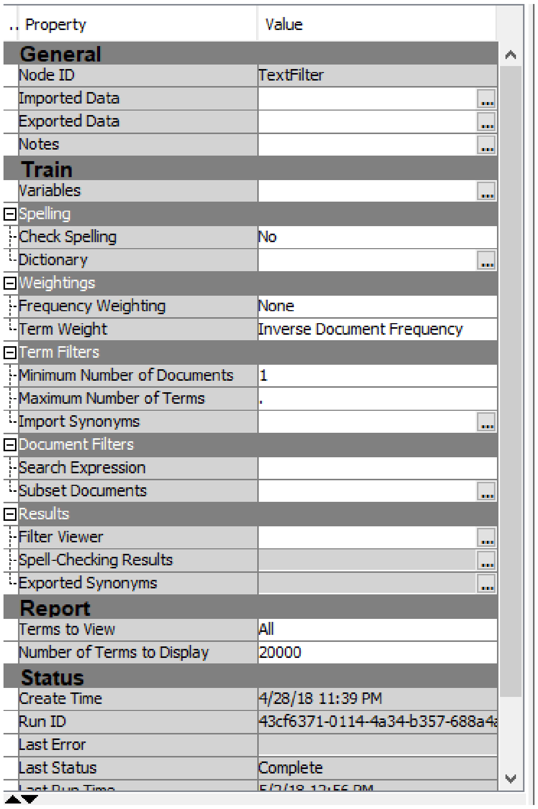
Topic Modelling

Parsing Node



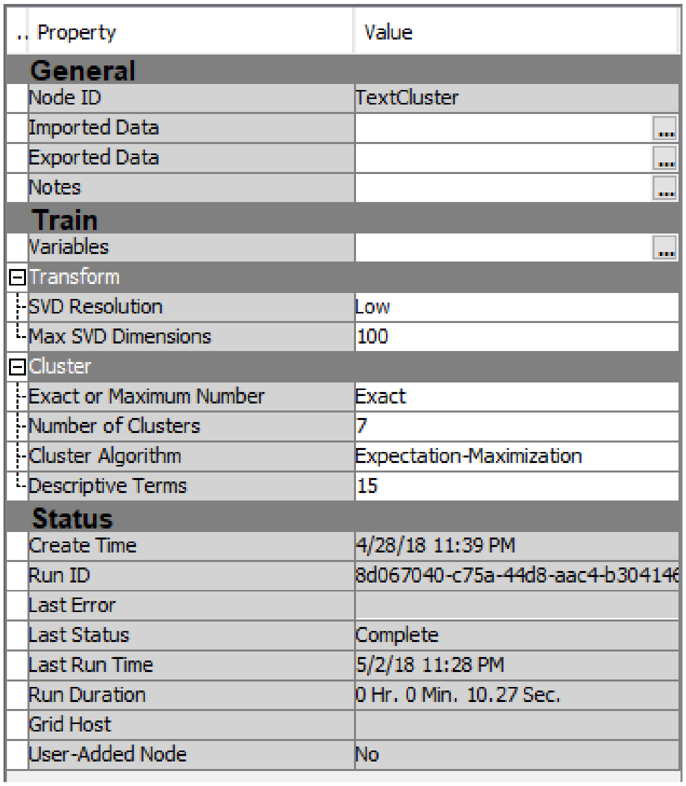
Property	Value
General	
Node ID	TextParsing
Imported Data	...
Exported Data	...
Notes	...
Train	
Variables	...
Parse	
Parse Variable	description
Language	English
Detect	
Different Parts of Speech	Yes
Noun Groups	Yes
Multi-word Terms	SASHELP.ENG_MULTI
Find Entities	None
Custom Entities	
Ignore	
Ignore Parts of Speech	'Aux' 'Conj' 'Det' 'Interj' 'Part' 'Pr
Ignore Types of Entities	...
Ignore Types of Attributes	'Num' 'Punct'
Synonyms	
Stem Terms	Yes
Synonyms	SASHELP.ENGSYNMS
Filter	
Start List	...
Stop List	SASHELP.ENGSTOP
Select Languages	...
Report	
Number of Terms to Display	20000
Status	
Create Time	4/28/18 11:39 PM
Run ID	37feeb09-0538-453f-9200-ba2d61
Last Error	

Filter Node



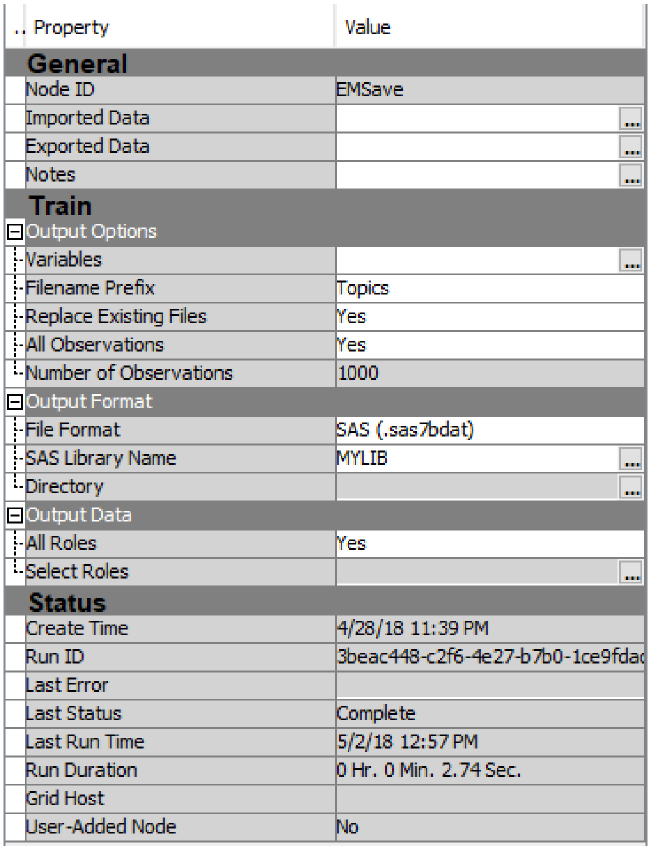
Property	Value
General	
Node ID	TextFilter
Imported Data	...
Exported Data	...
Notes	...
Train	
Variables	...
Spelling	
Check Spelling	No
Dictionary	...
Weightings	
Frequency Weighting	None
Term Weight	Inverse Document Frequency
Term Filters	
Minimum Number of Documents	1
Maximum Number of Terms	.
Import Synonyms	...
Document Filters	
Search Expression	
Subset Documents	...
Results	
Filter Viewer	...
Spell-Checking Results	...
Exported Synonyms	...
Report	
Terms to View	All
Number of Terms to Display	20000
Status	
Create Time	4/28/18 11:39 PM
Run ID	43cf6371-0114-4a34-b357-688a4a
Last Error	
Last Status	Complete
Last Run Time	5/2/18 12:56 PM

Text cluster



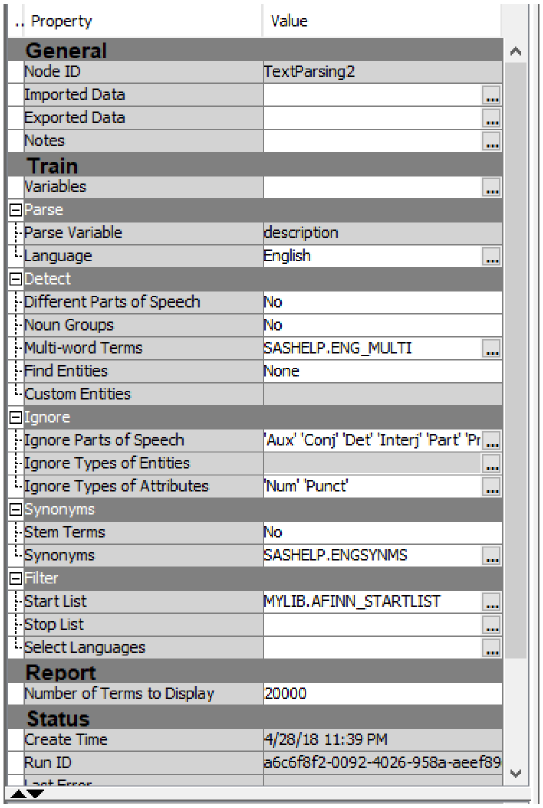
Property	Value
General	
Node ID	TextCluster
Imported Data	...
Exported Data	...
Notes	...
Train	
Variables	...
Transform	
SVD Resolution	Low
Max SVD Dimensions	100
Cluster	
Exact or Maximum Number	Exact
Number of Clusters	7
Cluster Algorithm	Expectation-Maximization
Descriptive Terms	15
Status	
Create Time	4/28/18 11:39 PM
Run ID	8d067040-c75a-44d8-aac4-b304146
Last Error	
Last Status	Complete
Last Run Time	5/2/18 11:28 PM
Run Duration	0 Hr. 0 Min. 10.27 Sec.
Grid Host	
User-Added Node	No

Save Data

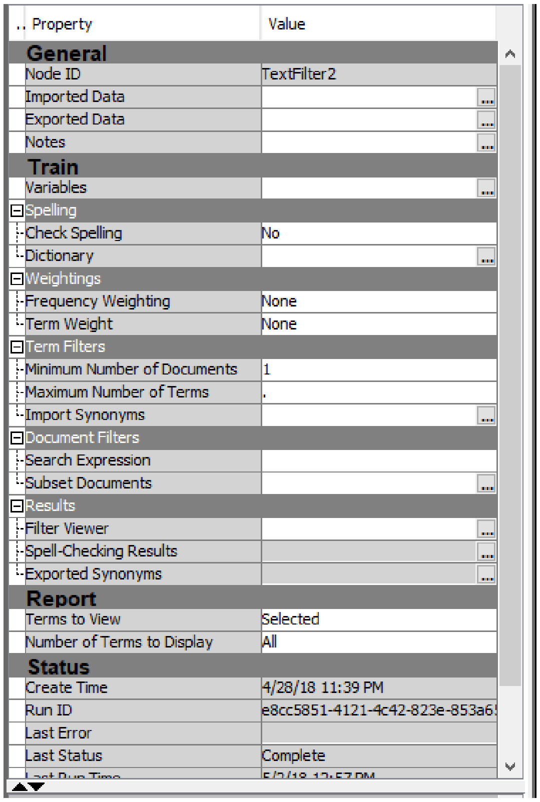


Property	Value
General	
Node ID	EMSave
Imported Data	...
Exported Data	...
Notes	...
Train	
Output Options	
Variables	...
Filename Prefix	Topics
Replace Existing Files	Yes
All Observations	Yes
Number of Observations	1000
Output Format	
File Format	SAS (.sas7bdat)
SAS Library Name	MYLIB ...
Directory	...
Output Data	
All Roles	Yes
Select Roles	...
Status	
Create Time	4/28/18 11:39 PM
Run ID	3beac448-c2f6-4e27-b7b0-1ce9fdac
Last Error	
Last Status	Complete
Last Run Time	5/2/18 12:57 PM
Run Duration	0 Hr. 0 Min. 2.74 Sec.
Grid Host	
User-Added Node	No

Text parsing and filter

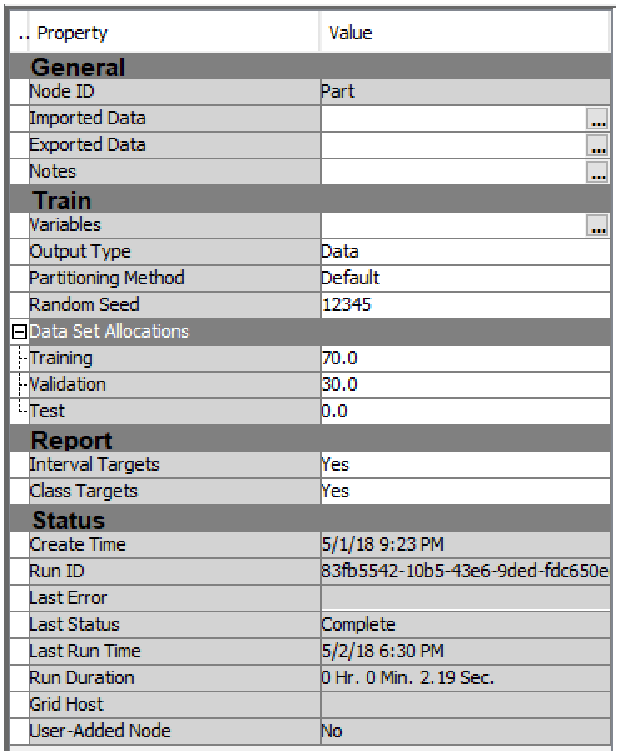


Property	Value
General	
Node ID	TextParsing2
Imported Data	
Exported Data	
Notes	
Train	
Variables	
⊟ Parse	
Parse Variable	description
Language	English
⊟ Detect	
Different Parts of Speech	No
Noun Groups	No
Multi-word Terms	SASHELP.ENG_MULTI
Find Entities	None
Custom Entities	
⊟ Ignore	
Ignore Parts of Speech	'Aux' 'Conj' 'Det' 'Interj' 'Part' 'Pr
Ignore Types of Entities	
Ignore Types of Attributes	'Num' 'Punct'
⊟ Synonyms	
Stem Terms	No
Synonyms	SASHELP.ENGSYNMS
⊟ Filter	
Start List	MYLIB.AFINN_STARTLIST
Stop List	
Select Languages	
Report	
Number of Terms to Display	20000
Status	
Create Time	4/28/18 11:39 PM
Run ID	a6c6f8f2-0092-4026-958a-aeef89
Last Error	



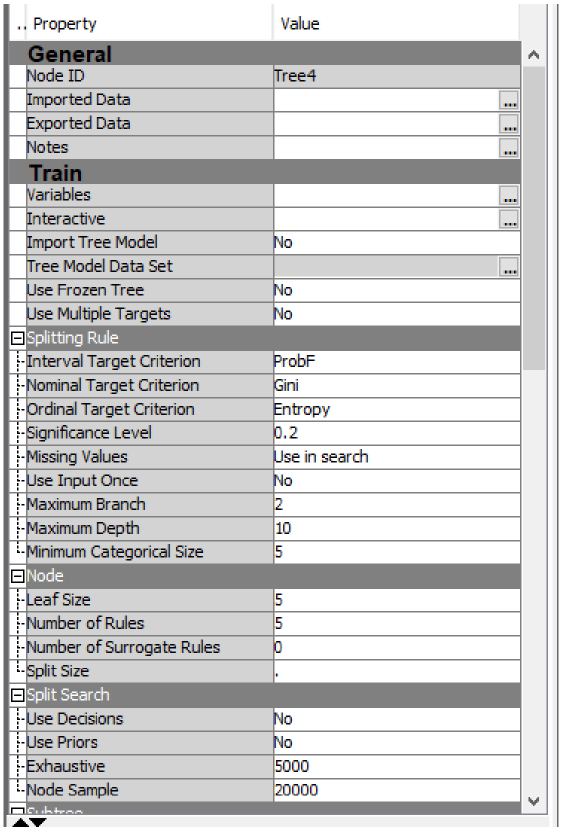
Property	Value
General	
Node ID	TextFilter2
Imported Data	
Exported Data	
Notes	
Train	
Variables	
⊟ Spelling	
Check Spelling	No
Dictionary	
⊟ Weightings	
Frequency Weighting	None
Term Weight	None
⊟ Term Filters	
Minimum Number of Documents	1
Maximum Number of Terms	.
Import Synonyms	
⊟ Document Filters	
Search Expression	
Subset Documents	
⊟ Results	
Filter Viewer	
Spell-Checking Results	
Exported Synonyms	
Report	
Terms to View	Selected
Number of Terms to Display	All
Status	
Create Time	4/28/18 11:39 PM
Run ID	e8cc5851-4121-4c42-823e-853a6
Last Error	
Last Status	Complete
Last Run Time	5/2/18 12:57 PM

Data partition



Property	Value
General	
Node ID	Part
Imported Data	...
Exported Data	...
Notes	...
Train	
Variables	...
Output Type	Data
Partitioning Method	Default
Random Seed	12345
⊟Data Set Allocations	
Training	70.0
Validation	30.0
Test	0.0
Report	
Interval Targets	Yes
Class Targets	Yes
Status	
Create Time	5/1/18 9:23 PM
Run ID	83fb5542-10b5-43e6-9ded-fdc650e
Last Error	
Last Status	Complete
Last Run Time	5/2/18 6:30 PM
Run Duration	0 Hr. 0 Min. 2.19 Sec.
Grid Host	
User-Added Node	No

Decision Tree



Property	Value
General	
Node ID	Tree4
Imported Data	...
Exported Data	...
Notes	...
Train	
Variables	...
Interactive	...
Import Tree Model	No
Tree Model Data Set	...
Use Frozen Tree	No
Use Multiple Targets	No
⊟Splitting Rule	
Interval Target Criterion	ProbF
Nominal Target Criterion	Gini
Ordinal Target Criterion	Entropy
Significance Level	0.2
Missing Values	Use in search
Use Input Once	No
Maximum Branch	2
Maximum Depth	10
Minimum Categorical Size	5
⊟Node	
Leaf Size	5
Number of Rules	5
Number of Surrogate Rules	0
Split Size	.
⊟Split Search	
Use Decisions	No
Use Priors	No
Exhaustive	5000
Node Sample	20000