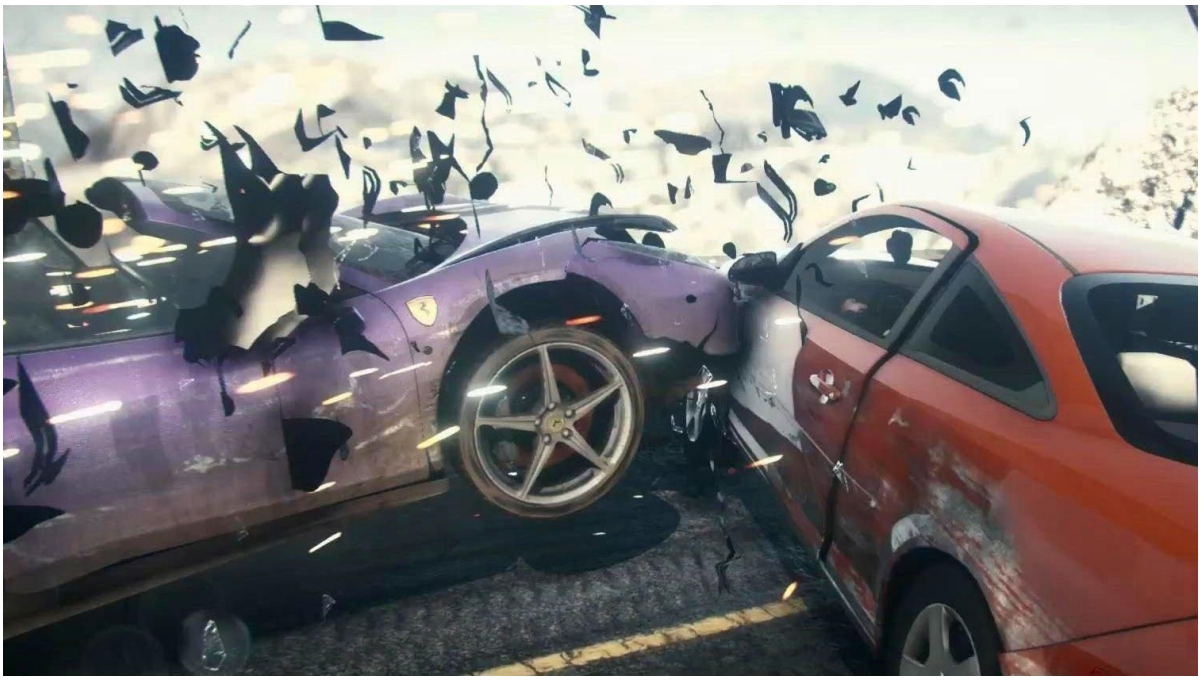


Final Project Report

PREDICTION OF A CAR CRASH

Applied Analytics | STAT 656 | 05/03/2018

(Prof. Edward Jones)

Team members: -

1) Krit Gupta
2) Neehar Yalamarti
3) Kshitij Sasavade
4) Aparajit Koshal
5) Ashish Jatav

Table of Contents

Problem Statement

The given data set consists of 5331 complaints from customers having a Honda car and 10 features that describe the complaints and the state of the car. The target attribute is 'Crash' with Crash = 1 signifying that the customer has experienced a crash while Crash = 0 is when the customer has not experienced a crash. The Problem can be broken down into three stages:

· Conduct Topic Cluster Analysis on the complaints to identify which issues are most related to a crash occurring.

· Conduct Sentiment Analysis to obtain a sentiment score for each complaint to understand the severity of the complaint.

· Build a predictive model (Decision Tree) which predicts the probability of the target – Crash using the topic and sentiment attribute previously created and the best model (best depth) is to be obtained by performing cross validation.

Metrics from the predictive model are then to be found which predict the Crash to the best degree.

The problem is to be solved using Python and SAS EM.

Introduction

This report entails the approach to build a predictive model after having done topic and sentiment analysis on the complaints from customers. This is done using all the features with the additional features of the topic and the sentiment score.

The Data Set consist of 10 features,

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 addition to this, is the actual Complaint where the customer has given his reviews with respect to the Crash/Accident/Issues.

The approach is to first replace and impute outliers according to the limits defined in the data set dictionary. Topic Analysis is then done to obtain topic clusters within the complaints to identify associated complaints. This is done with the help of Text Parsing and Text Filtering. Sentiment Analysis is also done on the original data set to obtain a sentiment score for each complaint. This is done by first performing Text Parsing and Text Filtering. The output files from this are then used to calculate average sentiments.

Having done the above two steps, the newly obtained data set now contains all the original features along with two new attributes: Topic Cluster Number and Sentiment Score.

A decision tree is then built and a best decision tree is selected by varying the parameters of the decision tree (depth) and obtaining the best results using 10-Fold Cross Validation. With the help of this best model, the original data is split to train and test to obtain the prediction metrics on the test data set. The metrics to be obtained are as follows:

Accuracy: Crash correctly predicted amongst all the complaints from users.

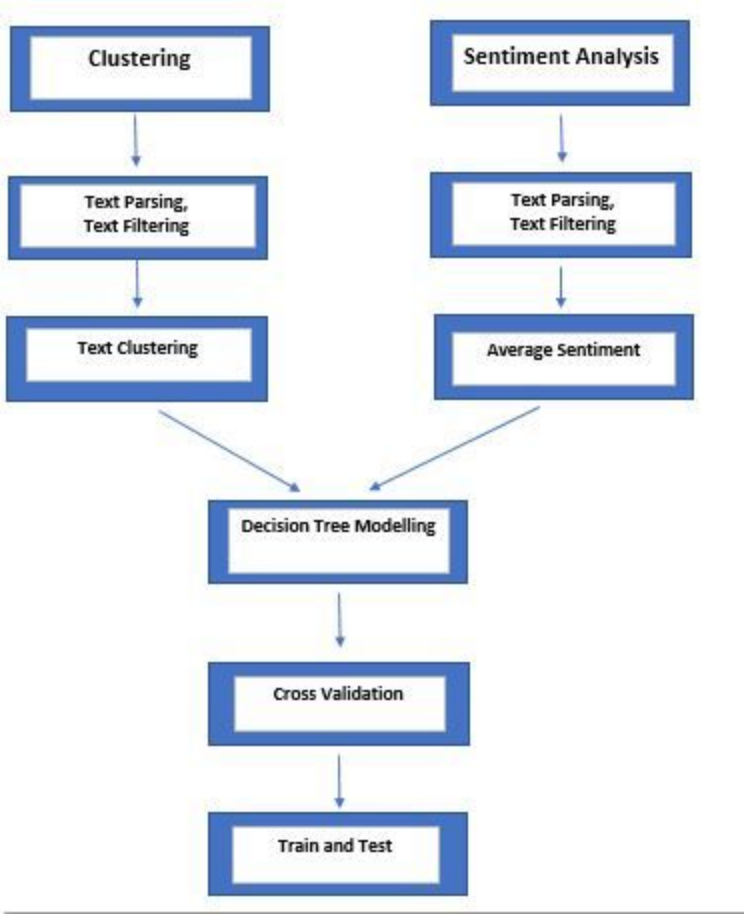
Recall: Also known as sensitivity which tells us how well the model predicts a crash with respect to the actual number of crashes that have occurred.

Specificity: Defines how well the model predicts no crash with respect to the actual no crash incidents.

F1 Score: A weighted score of recall and specificity.

Project Methodology

Clustering and sentiment analysis can be done simultaneously to produce the required attributes of text cluster number and average sentiment per complaint. The decision tree model is then built on this data set upon which cross validation and testing is performed to obtain the required metrics.



Data (Descriptive Statistics)

To obtain the summary of the dataset i.e. to understand what is the distribution of predictor space, to look for outliers within the data and to get the overall feel of the dataset, data was uploaded in python and descriptive statistics were obtained.

For text mining and predictive modeling it was essential for us to analyse the data before hand. We looked for synonyms in the data for text analysis and outliers within the data set for predictive modeling. Some of the common synonyms that were present in the data.

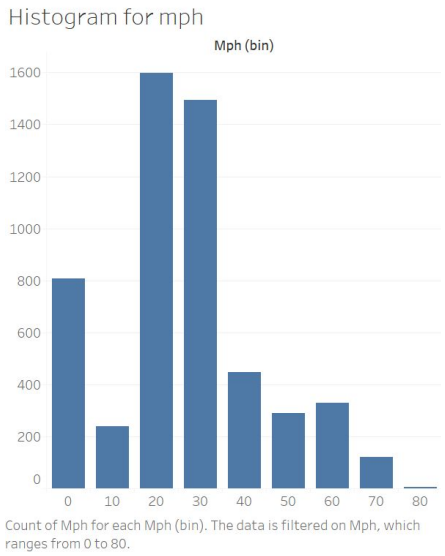
Parent word	Synonym
air bag	airbag
air bags	airbag

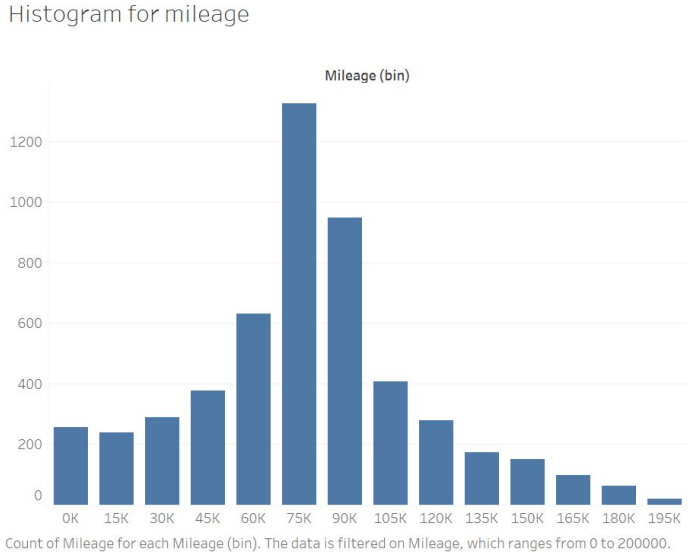
For outlier detection: -

Interval Attributes

Aforementioned parameters do not match the parameters the dataset description specified earlier in the report. There are some outliers in the dataset which should be removed.

To comprehend the distribution of interval variables, histograms were plotted.





As can be seen mileage follows approximately the normal distribution, and mph follows a right skewed distribution.

Nominal Attributes

All the nominal attributes had specified categories.

Comparison in terms of percentage of crashes for each category of the nominal variable has been tabulated.

1. Year

Year	Total Values	Crashed Incidents	% of crashes
2001	1783	252	14.13348
2002	3154	293	9.289791
2003	393	26	6.615776

This clearly shows that in the year 2001, there was a high percentage of car crashes for every complaint, but it has been on a decreasing trend, a reason of happiness for the Honda company.

2. Anti Braking System: -

Anti-Braking System	Total Values	Crashed Incidents	% of crashes
Yes	1426	192	13.46424
No	3894	279	7.164869

This is contrary to one's believe at the first sight. But there have been studies that once ABS is installed in the sub-system, it increases the chance of crash by 9%. This idea is coherent with the fact that when driver has a sense of increased safety, they might speed up and increase the probability of crashing.

3. Cruise Control: -

Cruise Control	Total Values	Crashed Incidents	% of crashes
Yes	1712	217	12.67523
No	3618	354	9.784411

This ties to the fact that while the vehicle is in cruise control, complacency is generally the case. Hence it might be the reason why there is a higher percentage of crashes in cruise control.
For attributes like model and make, there can be differences in manufacturing methods which might be the reason of crashes. After some basic exploratory data analysis, we moved on to the statistical learning methods.

Although we were supposed to ignore the nominal state variable, we wanted to make sure that the relative importance of this importance is fairly low hence it is included in the analysis.

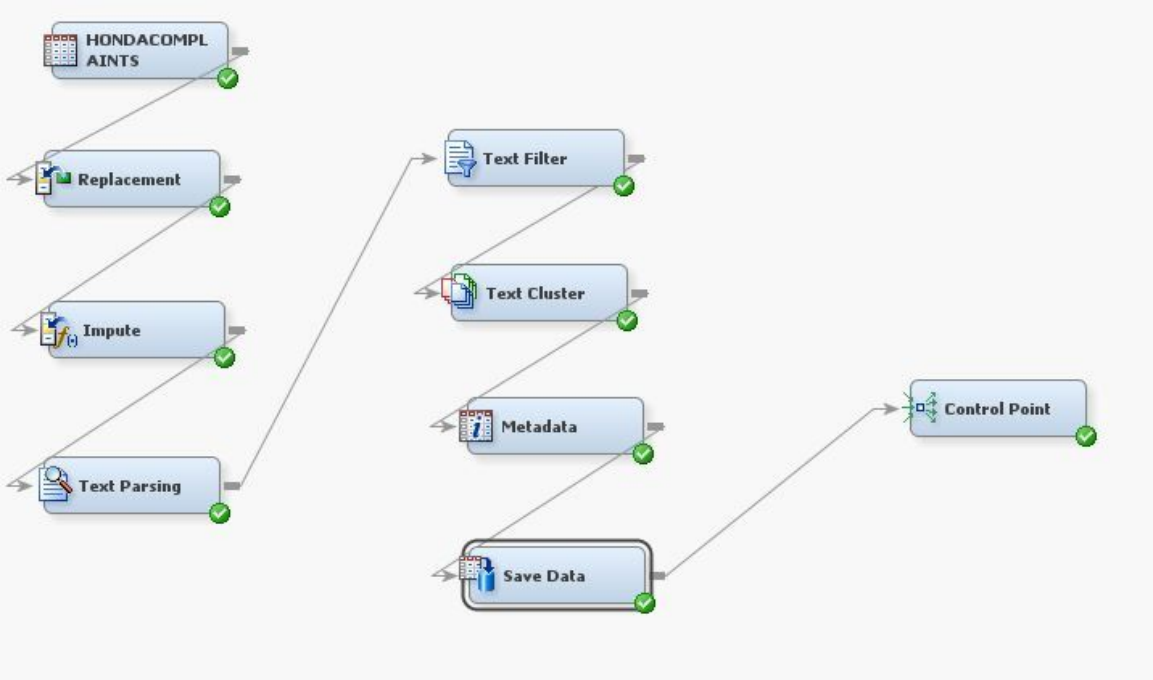
Statistical Learning Methods

SAS Approach

There were three separate diagrams that were used to develop the SAS model. In the first diagram with the help of a text cluster node, 7 different clusters were developed. After developing clusters, sentiment analysis was performed in the second diagram.

In the third diagram, results from the previous two diagrams were used to develop a predictive model keeping crash as the response variable.

First diagram (Topic Cluster Analysis): -



While performing text clustering, it was essential to achieve dimensionality reduction and hence classic text mining approaches were used: -

1) POS Tagging: - This helped in defining what was the part of speech for every single term.
2) With POS, some of the unnecessary POS like Auxiliaries, Conjunction, Determiner, which are some of the noun modifiers were ignored.
3) Synonyms and stop words certainly helped in trimming down the predictor space.

All aforementioned steps were present in the text parsing node in SAS. Since removing stop words cannot completely ensure that weak predictors are removed from the model, TF-IDF was used to develop the document term matrix.

In SAS, text filter node was run with the following properties: -

Frequency Weighting Method was set to none which implies that only frequency count will be taken into consideration. Term weight in the text was set to Inverse Document Frequency which is computed by the following formula: -

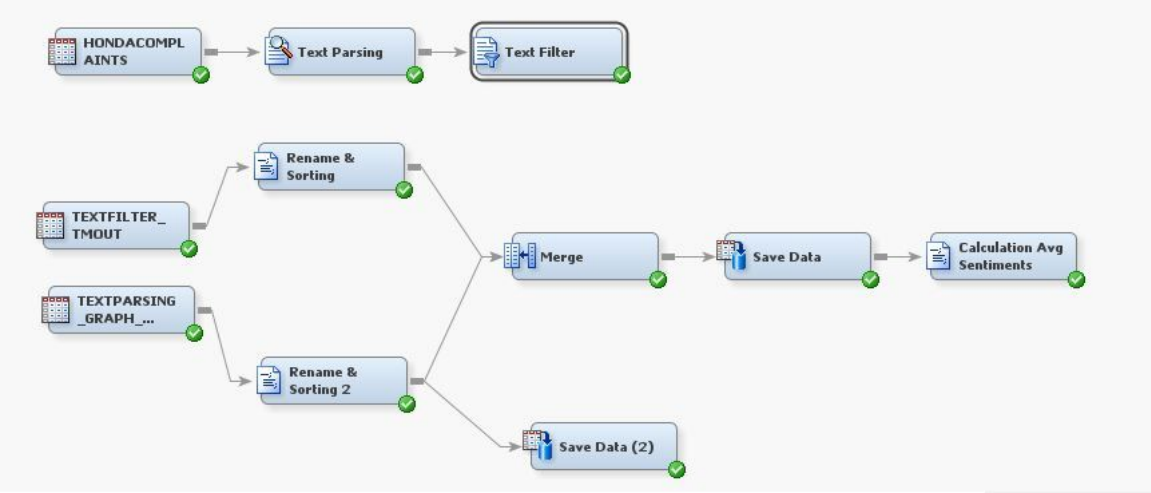
$$w_i = log_2\left(\frac{1}{P_{(t_i)}}\right) + 1$$

Here, P is the proportion of documents that contains the term t. This method gives greater weight to terms that occur infrequently in the document collection by placing the number of documents that contain the term in the numerator of the formula.

After text filter node, text cluster node was used to develop 7 clusters and 15 words represented these clusters. In order to get the greater accuracy, high resolution SVD option was selected within the text cluster node. After developing clusters dataset was saved for predictive modeling.

Second diagram (Sentiment Analysis)

To understand the relationship between a complaint being a negative complaint (Document score being less than zero) and happening of a crash, sentiment analysis was carried out.



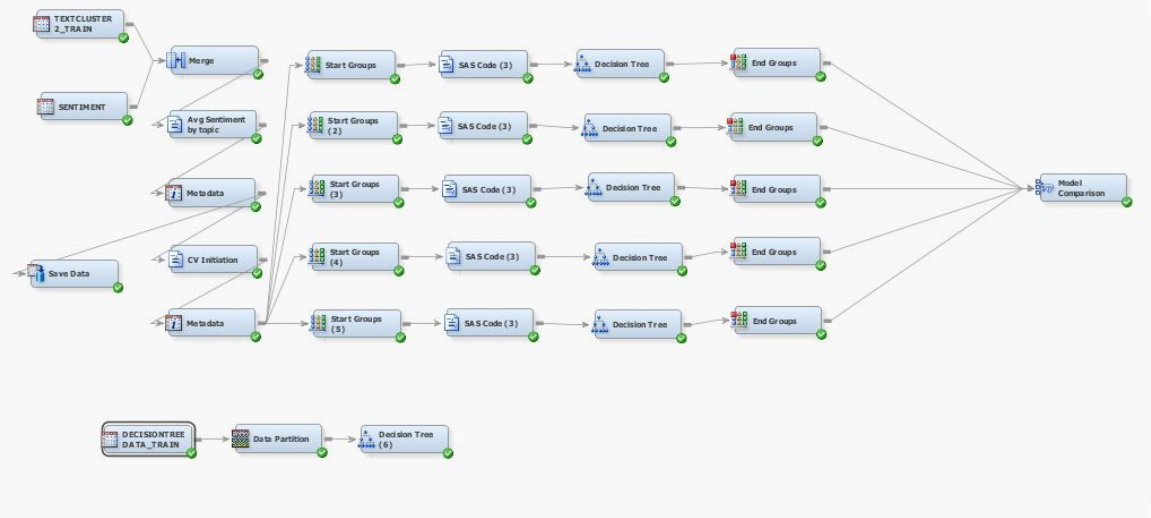
Since it is important to include every word in sentiment analysis that is also present in the AFINN list. POS tagging and stemming were ignored and synonyms were allowed in the analysis. Only change in the text filter node was that AFINN list was added as the start list.

The helped in identifying what words were similar in both the datasets. After filtering, two different datasets were extracted from the workspace library. The text filter tmout consisted of

termID, count and document number and the text parsing graph table contained the termID and actual terms plus some other things which were not useful. By applying the concept of relational databases both of these datasets were inner joined keeping termID as the primary key.

The reason for merging these two dataset was to match the count, document number by the term and not by the termID. These terms and their count were later used to calculate the document score (whether the document has positive sentiment or a negative sentiment).

Third diagram (Predictive Model)



With the results from the previous two diagrams: the text cluster number and the average sentiment per complaint, A decision tree model is then built. The data set is obtained by merging the hotel_cluster data set and the sentiment data set by _document_ (complaint number). The average sentiment per topic is also found out on this data to obtain insights on the topics.

The Metadata node is put to reject irrelevant attributes like the description of the complaint, n and stars (which were used to obtain the doc score and are not required now). Following that is the code to initiate cross validation i.e creating segments (Folds) according to a random seed.

Cross Validation is then performed on the different decision tree models to obtain the best depth using the start and end group nodes with the rerun property set to Yes. The model comparison gives a superficial results (i.e no confusion table / metrics ; only the misclassification rate is given)

Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Target Label	Selection Criterion: Train: Misclassification Rate
Y	EndGrp2	EndGrp2	End Groups	crash	crash	0.056098
	EndGrp5	EndGrp5	End Groups	crash	crash	0.056098
	EndGrp4	EndGrp4	End Groups	crash	crash	0.059287
	EndGrp3	EndGrp3	End Groups	crash	crash	0.061726
	EndGrp	EndGrp	End Groups	crash	crash	0.066417

The best depth of 10 is taken even though depth of 12 gave similar results since depth 10 will have a lower complexity.

The data set is finally split to train and test (70-30 %) and the best decision tree depth is modelled to obtain the required results.

Python Approach

Data Preprocessing

The consumer complaints (the data) is preprocessed to make it ready for analysis. The steps taken for data preprocessing are as follows:

1) Duplicate reviews are dropped in the initial go itself, to make sure that the words contained in them are not considered multiple times and their weights are not artificially inflated. This ensures that the sentiments obtained later on are not affected by artificially inflated duplicated words.
2) A cursory glance of the data showed us that 'Air Bag' is one of the largest used words. We replaced 'Air Bag' with 'AirBag' to ensure consistency in the word usage.
3) Lower cased every review. This ensures that the uppercase and lowercase words are treated the same in the analysis.
4) Removed the Pronouns, words like 'the', 'him', 'it', to ensure that the analysis contains the important words. The major reason of doing this is that the stop words list in Python is extremely narrow and needs to be expanded for a good data analysis.
5) Words like 'nt, n't is replaced with 'not', to ensure consistency.
6) We ran the preliminary topic modeling multiple times to get meaningful clusters and the words that we found to be not representative of important information to classify the cluster were added in either the Stop Words list or the Synonyms list, as the case be. This gave us words like 'car' (since all the reviews are about cars only, hence, using the word 'car' will inflate the term frequency of the word 'car' without giving a lot of useful information. Unexpectedly though, the word 'car' still appears in the analysis!
7) We expanded the Stop Words list after multiple preliminary revisions.

Stemmer

We considered Porter Stemmer and Snowball Stemmer to stem the incoming words from the reviews.

However, a reading of the official Porter page at : http://snowball.tartarus.org/algorithms/porter/stemmer.html made us choose Snowball Stemmer over Porter Stemmer for our analysis.

Topic Modeling

For creating concrete topics from the reviews, we first create a Term Frequency Dictionary of unique words, after removing the Stop Words and Stemming them using SnowBall Stemmer.

We obtained:

Number of Reviews..... 5310
Number of Terms..... 2842

We then set out to calculate the Term Frequency for every unique word that we could find. The top 10 words that we could find are:

Terms with Highest Frequency:

honda	6149.000000
transmission	5288.000000
problem	2986.000000
contact	2858.000000
dealer	2728.000000
failure	2346.000000
drive	2312.000000
light	2205.000000
recall	1962.000000
replace	1828.000000

We then set out to create Term Frequency Matrix, with the number of reviews as the rows and the each unique word as the column. This gives us a matrix that essentially gives the count of each word used in every review.

We then give an Inverse Document Frequency weight to give more weight to the words which are ‘rare’ and present in less number of documents. The intuition is that the ‘rare’ words are specific and convey more meaning about the topic being discussed about in the review.

This gives us a matrix with TF-IDF (Term Frequency - Inverse Document Frequency) scores. TF-IDF matrix was calculated using TfidfVectorizer of scikit-learn Python library.
The top 10 words that we found are:

WORD	TF-IDF SCORE
transmission	10184.37
honda	9511.28
contact	7013.14
problem	6343.92
dealer	5582.90
failure	5302.72
light	5231.68
drive	4821.29
recall	4766.07
airbags	4439.84

We then segregated the reviews into 7 clusters, using the LDA (Latent Dirichlet Allocation) method. Once again, we utilize the LatentDirichletAllocation( ) method of Scikit-learn Python Library.

Online Bayes method was used to obtain the to learn the latent factors. Using the 'online' setting in the method makes the use of mini-batches of training data in each update, thus, making the analysis quicker and effective.

This step was repeated multiple terms in order to effectively increase the Stop Words list and the Synonyms list for a better analysis.

We finally calculated a topic-score for each review, after getting the Probabilities or the likeliness of how much each review 'belonged' to each cluster.
We gave the (cluster number - 1) of the largest Probability as the topic-score for each review.
The reason why we subtracted 1 from the topic cluster number, was to ease the further processing in Python, as the index starts from 0 in Python and it becomes a lot easier to do predictive analysis further down the road.

We then added the calculated topic-score as a feature in the main data set, to be used in the Predictive Analysis.

3 clusters dominate our analysis - Clusters 1, 2 and 6. The topic distribution that we obtained is as follows:

TOPIC DISTRIBUTION

TOPIC	N	PERCENT
1	364	6.9%
2	487	9.2%
3	1357	25.6%
4	580	10.9%
5	1003	18.9%
6	973	18.3%
7	526	9.9%

Sentiment Analysis

Sentiment Analysis is the indication/strength of the opinion and the feeling present in a piece of text. It is essentially calculated by the number of emotive words in the text, with a score calculated from a list of all the emotive words, as defined. We use the AFINN List for the calculating the sentiment score for each review. AFINN list has a score range from [-5, 5] to represent the emotion of a word.

We calculate the sentiment score for each review, and then average them for each topic and finally average to get the Total Average Sentiment for all the reviews.

The value obtained is: -1.0836

Predictive Analysis (Decision Tree)

With the topic-score and sentiment calculated from the previous sections, we append both the scores as features to the original dataset. These features, in addition to the original dataset are used to predict whether a Crash has occurred or not, using the Decision Tree as a classifier.

The steps are as follows:

1) We drop the State, Nhstat_ID and description columns from the dataset, as they dont have useful information for the predictive analysis.
2) We convert the String fields to numeric for the scikit-learn's methods to able to use them.
3) Find all the outliers and set them as 'Missing'. The results we obtained are:

Number of missing values and outliers by attribute:

Year:	0 missing 0 outlier(s)
Make:	0 missing 0 outlier(s)
Model:	0 missing 0 outlier(s)
crash:	0 missing 0 outlier(s)
cruise:	0 missing 0 outlier(s)
abs:	0 missing 0 outlier(s)
mileage:	1 missing 87 outlier(s)
mph:	0 missing 1 outlier(s)
topic:	20 missing 20 outlier(s)
T1:	20 missing 0 outlier(s)
T2:	20 missing 0 outlier(s)
T3:	20 missing 0 outlier(s)
T4:	20 missing 0 outlier(s)
T5:	20 missing 0 outlier(s)
T6:	20 missing 0 outlier(s)

T7: 20 missing 0 outlier(s)

4) We then impute the missing values by the Mean for Interval data and by the Most Frequent for the Categorical Data.
5) One hot Encoding was used next in order to make the data in the form that can be used for predictive analysis, further down.
6) Cross Validation was done for getting the best fit depth of the tree. We used 5,6,7,8,10,12,15,20,25 as the depths.
7) On getting the best depth for the decision tree, we split the test and train data in the 30:70 ratio, and perform predictive analysis on it.

Results and Conclusion

SAS Results

Results from Topic Analysis are as follows:

	Descriptive Terms	Frequency
1	air front rear +notice safety +side +cause +find +system +happen +driver +recall back +time +passenger ...	783
2	+contact +vehicle +repair +mileage +failure +state +dealer +manufacturer current nhtsa +number +'current mileage' ca...	876
3	+car +mile +engine +stop +start +happen +transmission +drive +road +accelerate +problem +year +service +fix +issue...	770
4	+transmission +gear +replace +failure +mile +shift +slip +odyssey +fail +'transmission failure' +cost automatic +start +...	1125
5	+airbag +deploy +driver +seat +passenger +belt front mph side +side +hit +impact +injury +collision +'seat belt' ...	443
6	+'srs light' +airbag +light +tire safety +brake +fix +system +seat front +accident +problem +warranty +deploy +time ...	791
7	+vehicle +consumer +brake +control +dealer +state +cause +dealership +problem driving +engine +notify +stop +accel...	542

The cluster which has the most frequency of words is due to Transmission Issues followed by Issues related to failure of deployment of the Airbag.

Results from Sentiment Analysis show that there is an overall negative sentiment from all the complaints, with the following statistics:

The MEANS Procedure

Analysis Variable : docScore

N	Mean	Std Dev	Minimum	Maximum
5330	-1.2373865	1.0827836	-5.6666667	3.5000000

Sentiment Score was also calculated topic wise which have the following statistics:

	docScore					
	Min	Mean	Median	Max	N	PctN
TextCluster_cluster_						
1	-4.00	-0.91	-1.00	3.50	783.00	14.69
2	-5.00	-1.73	-2.00	2.00	876.00	16.44
3	-5.67	-1.08	-1.11	3.00	770.00	14.45
4	-4.25	-1.37	-1.50	3.00	1125.00	21.11
5	-4.50	-1.39	-1.67	2.00	443.00	8.31
6	-5.50	-1.09	-1.00	3.00	791.00	14.84
7	-5.50	-0.97	-1.00	3.00	542.00	10.17

It is observed that Topic 2 has the most negative sentiment which includes issues due to mileage meaning the vehicle is too old and has crossed several thousand miles of distance. Words like dealer and manufacturer are also present indicating issues that involve rework of the vehicle.

Having performed the Decision Tree Modelling, Cross Validation over the different depths yielded the best depth to be 10. Hence Training and Testing was done with that depth. The following results were obtained from cross validation:

	Accuracy	**Recall**	**Specificity**	**F1 Score**	**Precision**
Depth = 6	0.933583	0.508499	0.983792	0.614578	0.801179
Depth = 7	0.938274	0.526894	0.986933	0.642551	0.841318
Depth = 8	0.940713	0.548548	0.987146	0.661217	0.847274
Depth = 10	0.943902	0.571627	0.987987	0.682656	0.858371
Depth = 12	0.943902	0.571627	0.987987	0.682656	0.858371

As is observed, Depth = 10 gives the best results with respect to all the metrics. Even though depth = 12 gives similar metrics, Depth of 10 is taken as there is lesser complexity in the tree.

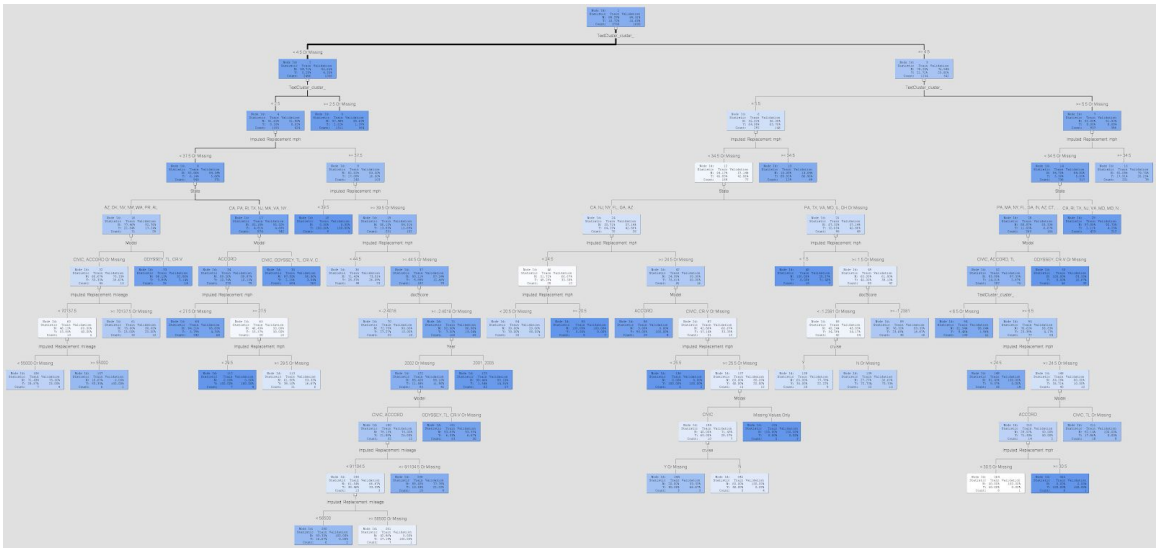
Training and Testing is then done with this depth to obtain the following metrics for the train and validate data sets:

Train	**FN**	**TN**	**FP**	**TP**	**Accuracy**	**Recall**	**Specificity**	**F1 Score**	**Precision**
	194	3299	31	206	0.93967	0.515	0.9906907	0.646782	0.869198

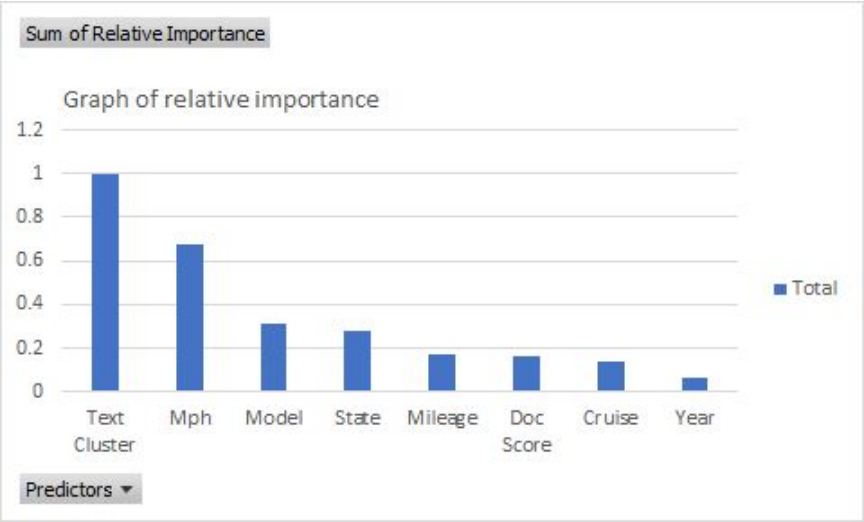
Validate	FN	TN	FP	TP	Accuracy	Recall	Specificity	F1 Score	Precision
	80	1414	15	91	0.940625	0.532164	0.9895031	0.65704	0.858491

As it is observed that the metrics from the test data set are similar to those obtained from cross validation for that depth = 10. Hence this is evidence that the model does not overfit the data.

The final decision tree is shown below:



The Relative importance of the attributes used in the decision trees is also displayed. It is observed that the Topic Cluster plays an important role in the prediction of a crash, followed by Mph and the model of the vehicle.


Sum of Relative Importance
Graph of relative importance
1.2
1
0.8
0.6
0.4
0.2
0
Text Cluster
Mph
Model
State
Mileage
Doc Score
Cruise
Year
Total
Predictors

Python Results

Predictive Analysis

We tried different depths of the Decision Tree to find the best depth of the Decision Tree, in order to perform Predictive Analysis on it.

The different results that we obtained are summarized in a table :

	Accuracy	Recall	F1 Score	Precision
Depth = 5	0.9051	0.2214	0.3215	0.7106
Depth = 6	0.9090	0.2548	0.3665	0.7466
Depth = 7	0.9066	0.3041	0.4024	0.6434
Depth = 8	0.8983	0.3357	0.4072	0.5411
Depth = 10	0.8893	0.3690	0.4089	0.4912
Depth = 12	0.8893	0.3919	0.4305	0.4865
Depth = 15	0.8842	0.4024	0.4235	0.4602
Depth = 20	0.8847	0.4077	0.4290	0.4667
Depth = 25	0.8849	0.4059	0.4219	0.4633

As can be seen, the best accuracy was obtained for a depth = 6.
Hence, we perform the Predictive Analysis for the depth of 6.

We perform the 70:30 split for the train and test data, the metrics are as follows:

Table of the metrics for 70/30 split

Model Metrics.....	Training	Validation
Observations.....	3717	1593
Features.....	22	22
Maximum Tree Depth.....	6	6
Minimum Leaf Size.....	5	5
Minimum split Size.....	5	5
Mean Absolute Error....	0.7349	0.7123
Avg Squared Error.....	0.6191	0.5948
Accuracy.....	0.8009	0.7646
Precision.....	0.3292	0.2665
Recall (Sensitivity)...	0.7931	0.7423
F1-score.....	0.4653	0.3922
MISC (Misclassification)...	19.9%	23.5%
class 1.....	20.7%	25.8%
class 2.....	19.8%	23.3%

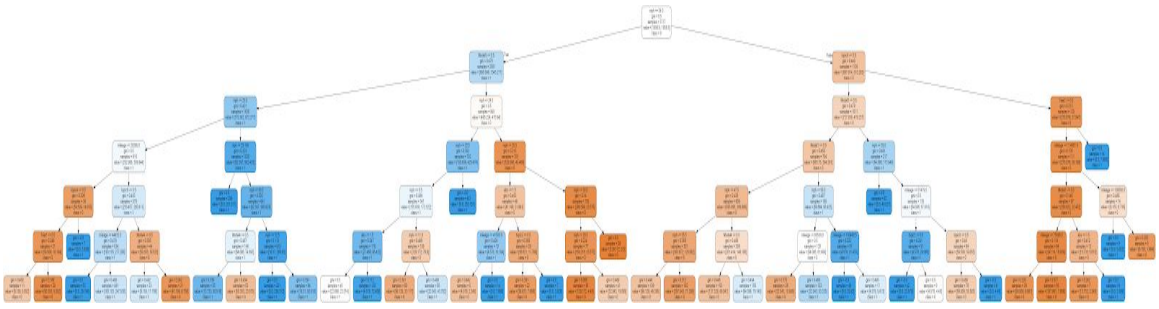
Training

Confusion Matrix	Class 1	Class 2
Class 1.....	322	84
Class 2.....	656	2655

Validation

Confusion Matrix	Class 1	Class 2
Class 1.....	121	42
Class 2.....	333	1097

The Decision tree is as follows:



Comparison With News API

1) Common words found amongst all the topics from the News API and the topics found from the reviews.

   {'mile', 'drive', 'honda', 'state', 'new', 'turn', 'crash', 'time', 'nhtsa'}
   It only has 9 common words. This is not very encouraging.

2) Average Sentiment

   As can be seen below, the average sentiment for both the classes - Reviews and the News API are not similar

   Reviews = (-1.0836184919952587)
   News API = (0.07861443970773985)

   Reviews have an average negative sentiment, a little above -1, whereas, the News API has a sentiment close to 0, which is towards neutral.

   Hence, it shows that the topics generated from the reviews do not relate much to the documents from News API.

The topics generated by both the classes can be seen below:

REVIEWS DATA TOPICS

********** GENERATED TOPICS **********
Topic #1:
+door +headlight +beam +low +information
+please +side +open +break +frame
+honda +driver +work +front +window
Topic #2:
+brake +stop +pedal +happen +gas
+back +accelerator +time +van +drive
+road +noise +home +middle +someone
Topic #3:
+honda +light +problem +issue +transmission
+recall +tell +warranty +fix +come
+dealership +service +safety +sr +replace
Topic #4:
+drive +engine +accelerate +speed +stop

+turn +mph +air +ignition +tow
+pull +road +without +suddenly +park
Topic #5:
+transmission +gear +shift +mile +acura
+tire +slip +honda +problem +replace
+2nd +new +jerk +start +odyssey
Topic #6:
+contact +failure +state +mileage +repair
+own +manufacturer +dealer +recall +nhtsa
+number +current +honda +airbags +campaign
Topic #7:
+airbags +deploy +seat +side +driver
+passenger +front +belt +injury +crash
+head +airbag +hit +accident +damage

NEWS API TOPICS

********** GENERATED TOPICS **********
Topic #1:
+wsj +video +journal +podcast +real
+art +section +estate +jones +dow
+street +popular +deal +commercial +subscribe
Topic #2:
+time +york +new +art +opinion
+page +today +american +navigation +state
+subscription +subscribe +video +event +image
Topic #3:
+ford +mile +inflator +mazda +pickup
+drive +setting +president +tablet +usa
+china +sale +linkedin +reuters +know
Topic #4:
+honda +yen +sale +cost +grow
+cut +percent +nearly +trillion +january
+year +annual +march +sell +make
Topic #5:
+apr +trump +white +policy +house
+name +man +feature +online +week
+turn +president +ad +morning +image
Topic #6:
+tablet +reuters +browser +landscape +inflator
+portrait +wide +honda +automaker +bankruptcy
+thomson +trust +motor +unit +injure
Topic #7:
+hyundai +kia +bank +crash +public

+brand +nhtsa +new +make +control
+index +dow +percent +car +story