Paper 072-31

# Text Mining with “Holographic” Decision Tree Ensembles

Barry de Ville, SAS Institute Inc., Cary, NC


## ABSTRACT

The data-mining classification and predictive modeling algorithms that are based on bootstrapping techniques re-use a source data set, repeatedly, to create a family of predictive and classification models that can be said to render a "holographic" view of the modeled data. The results offer a classification and prediction performance that is superior to single-model approaches. These holographic approaches are applied in an industrial setting that involves text mining warranty claims at a major international car, truck, and heavy equipment manufacturer.

This paper explains the methods used, how they work, and how they perform in the text-mining area as applied to warranty claims. Combined text and quantitative data models are developed, tested, and validated in order to address the goal of achieving "better-than-human" classification performance on warranty claims.


## INTRODUCTION

Text and data mining are increasingly common approaches used to examine text and data in order to draw conclusions about the structure and relationships between sets of information contained in data. The text- and data-mined results are often used to build scoring facilities that can be used to predict (or classify) values in new, previously unseen, data records. This provides the opportunity to screen incoming records for quality and also to project values on these records based on historical analytics.

We explored the use of this text and data mining capability by using SAS® Enterprise Miner™ and SAS® Text Miner to build scoring facilities. These scoring facilities could improve the speed and accuracy of assigning warranty repair designators to industrial data used during the processing of warranty claims at Volvo Truck sites. The data were taken from the stream of warranty data produced by Volvo engineers who were carrying out warranty actions in various North American facilities.

This was a preliminary exploration that produced encouraging results: automated algorithms that were constructed using text mining approaches were able to produce warranty classification accuracy scores close to 90%. This compared favorably to earlier audited results that showed accuracy in the range of 50% in manually-coded data sets. This suggests that text mining can be used to increase the accuracy of truck warranty data and that automated systems can be built to produce more timely and economical results.

## THE CURRENT ENVIRONMENT

Currently the warranty claim process at Volvo Truck assigns an eight-digit type/cause code to warranty claims. This system, carried out by human operators, has been in production for 15 years and when audited was shown to produce an approximately 50% accuracy rate in assigning claim codes based on the conditions of the warranty repair. Claim type and cause assignment serves as the basis for a variety of post-assignment engineering and accounting tasks such as the quality targets that are set for the vehicle manufacturing division.

The main goal of the analysis was to conduct a specific proof-of-concept to determine whether there is enough information in the current warranty claim records to automate the warranty code assignment process. This would save time and costs and would also contribute to the establishment of a more reproducible code-assignment process.

## APPROACH

There were over 1900 eight-digit warranty codes contained in the data set that was used in this exercise. There were over 600,000 records available for the analysis (an average of about 300 records per code). In order to make the proof-of-concept task more manageable, we picked 12 warranty codes for our initial set of analyses. These were the codes that are most prevalent and which, when automated, would be likely to yield the greatest benefits and potential cost savings.

These codes, and original distributions, are shown in the “Before Sampling” columns in Table 1.

| | Before Sampling | | After Sampling | |
|---|---|---|---|---|
| **Code** | **Frequency** | **Percent** | **Frequency** | **Percent** |
| Air Compressor | 376 | 0.05% | 376 | 0.32% |
| Alternator | 4755 | 0.68% | 4755 | 4.06% |
| Axle Cross Steering | 76 | 0.01% | 76 | 0.06% |
| Battery | 3221 | 0.46% | 3221 | 2.75% |
| Brake Lines/Clamps/Fittings | 7123 | 1.02% | 7123 | 6.09% |
| Fan Drive | 9234 | 1.32% | 9234 | 7.89% |
| Fuel Tank | 4432 | 0.64% | 4432 | 3.79% |
| Heater A/C Panel | 6467 | 0.93% | 6467 | 5.53% |
| Piping Connect/Fit | 5405 | 0.78% | 5405 | 4.62% |
| Starter | 2997 | 0.43% | 2997 | 2.56% |
| Wire Cab To Chassis | 7778 | 1.12% | 7778 | 6.65% |
| Other | 645441 | 92.56% | 65146 | 55.68% |
| **Total** | 697305 | | 117010 | |

**Table 1. Claim Code Distributions in Host and Sample Data Sets**

As shown in the “Before Sampling” column in the table, the “Other” category constituted over 90% of the data records. To simplify the analysis and to eliminate the possibly redundant information that these records contained, we took a 10% sample of the “Other” category. As shown in the “After Sampling” column this produced a final distribution of codes for analysis where “Other” represented just over 50% of the records. After sampling all the percentages of the non-“Other” codes increased although their raw frequencies remained the same. Reducing the dominance of the “Other” category will enable the non-“Other” data to be more strongly represented in subsequent analysis.

## SEQUENCE OF ANALYSIS

Once the test data set was assembled we moved onto the next step of processing text for analysis using SAS Text Miner to analyze the unstructured text. SAS Text Miner is an add-on to SAS Enterprise Miner that provides a wide range of data mining capabilities that are specifically directed towards analyzing unstructured text. One of the first tasks that is typically undertaken in text mining is to set up filters to remove words that would convey little or no meaning or predictive information in the analysis. The default Text Miner stop list was used for this task.

The next step was to generate a synonym list so that words in different forms but which contained essentially the same meaning or predictive capacity were mapped to a common value. The text mining text synonym macro (%textsyn) that is included in the 4.3/5.1 release of SAS Text Miner was used to accomplish this task. This produced a reduced text representation in the data set that consisted of approximately 110,000 unique terms.

Once this step is complete it is normal to run the Text Mining node in SAS Enterprise Miner using the synonym list. A number of options are possible, but the main raw input is a term x document matrix that contains entries that show which terms (and how many) are contained in which documents. It is normal to adjust the frequencies in the term-by-document matrix to prevent high-frequency, commonly-occurring terms from dominating the analysis. Because unique, often rare terms can play a significant role in distinguishing between different types of documents, it is normal to try to adjust rare term frequencies with a weight factor to give them an opportunity to contribute more to the analysis. For this purpose we used log-scaled entropy weights. These weights are calculated so as to emphasize terms that occur more frequently in fewer documents.

From a data analysis point of view this results in the creation of a data set that contains a large number of fields that may be associated with the warranty target field. In this case, the fields are the scale-weighted frequencies of terms that are associated with the target warranty code. Because it is normal--and indeed expected--to have a particular subset of terms that are associated with a given target, most of the entries that come from this term-by-document frequency matrix are empty. The Text Mining node provides a unique analytic technique (based on a form of factor analysis) that will group similar codes together. This technique, called “Singular Value Decomposition,” compresses the number of potential terms that are used to characterize and predict a warranty code. Compression is accomplished by summing the weights of terms that are associated with the various dimensions that are produced by the method so that the ultimate analysis document consists of summary scores (rather than individual term

frequencies). Once the compressed representation is calculated the original terms can be dropped from the analysis. In our case, we produced 42 such dimensional summaries. This reduction represents a considerable condensation when compared to the initial 110,000 term entries on the analysis data set.

In addition to the text fields contained in the data a number of structured, quantitative data fields were available. These fields are shown in Table 2.

The fixed field data that were available for use in generating the predictive models were as follows:

| Field | Description |
|---|---|
| PLANT | Plant Code |
| PLACEDTE | Build date |
| DELIVERY | Delivery date |
| REPAIRDT | Repair Date |
| CLAIMDTE | Claim Date |
| CLAIMTYP | Type of claim (fir example, warranty, prep for Delivery) |
| SERVMOS | Months in Service |
| MILES | Mileage ( X 1000) |
| TOTCREDT | Total Credit |
| FAILPART | Failed Part |

**Table 2. Structured (Quantitative) Fields in the Analysis**

The last field contains the text descriptions that described the warranty action. This was the source of the text data used in the text mining.

One goal of the analysis exercise was to use these quantitative measures in addition to the text data to determine whether a combination of quantitative and text data could produce better predictions.

## THE MODELING APPROACH

Four distinct sets of models were developed to determine which approach yielded the best results. All results were validated for accuracy through the use of a split-sample train/validation approach whereby 30% of the original data records were set aside to compare predicted results from the model with actual results based on this holdout set of test data. It is normal to use a 70:30 ratio of training records to test records. This ratio works well in preserving as much information as possible to train the predictive models while at the same time setting aside enough “hold-back” information (a 30% sample) to effectively rate the validity of the trained models after the fact.

- The first analysis used only the quantitative data fields to produce a predictive classification model for the warranty code.
- The second approach supplemented the quantitative fields with the introduction of the text mining summary predictors (42 dimensional summaries).
- The third approach was based on bagging the combined quantitative and text mining predictors.
- The fourth approach was based on boosting the combined quantitative and text mining predictors.

The SAS Enterprise Miner (release 4.3) diagram produced for this analysis is shown in Figure 1.

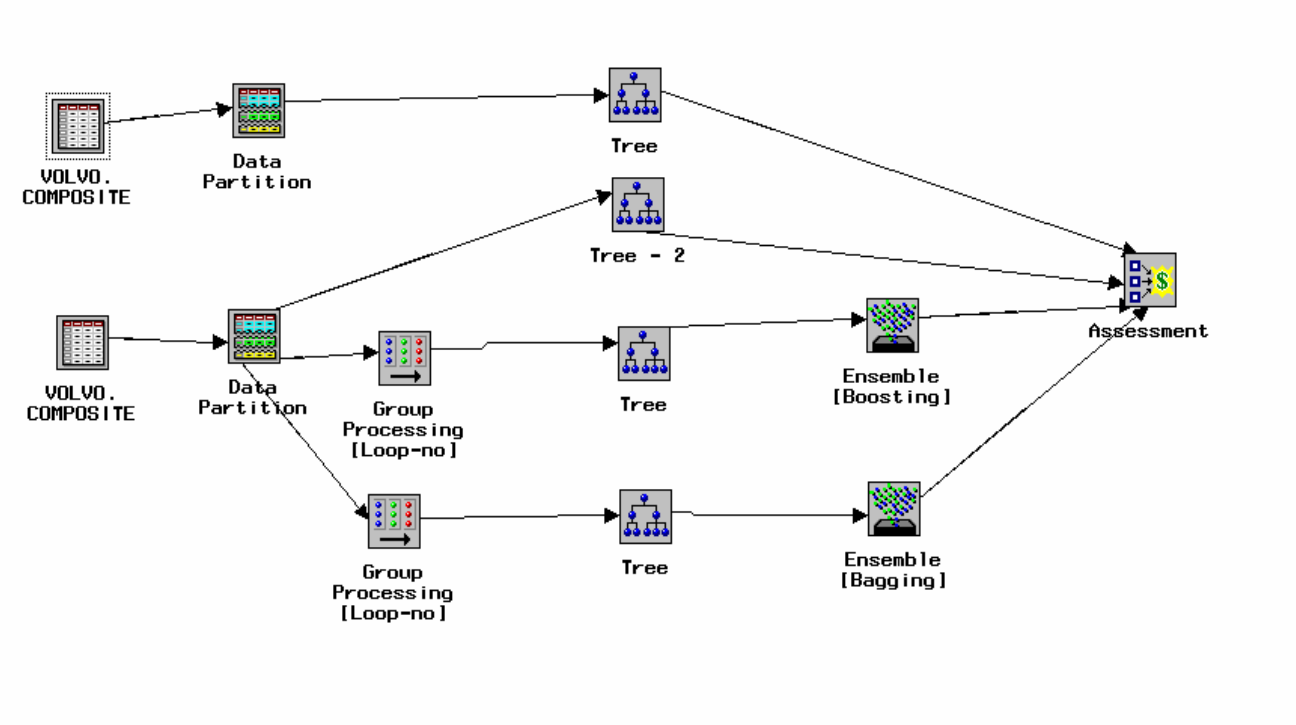


**Figure 1. Data Mining Process Flow Diagram for the Warranty Analysis**

Bagging and boosting are multiple tree methods and are described more fully in the next section.

## MULTIPLE TREES

Since the mid-1990s a number of multiple random decision tree approaches have been developed that take advantage of advances in computing power and user-computer interfaces to increase classification accuracy, including situations with relatively rare codes [1, 2, 3, 4]. Multiple decision trees are frequently referred to as "forests," “bagged,” "boosted," “committee,” or "ensemble" classifiers. One of the original developers of the approach--the team of Amit and Geman (1996, 1997)--describe these as “holographic” methods because each data point in the analysis can be resampled many times--under different circumstances--and therefore have the ability to contribute to the shaping of a multi-dimensional view of the data.

A number of studies and practical applications indicate that multiple trees provide dramatically increased classification and prediction performance over single-methods, classifiers, and predictors. An added benefit, in text classification applications, is that multiple trees can offer a superior approach to the prediction and classification of multi-class textual outcomes. In our case, as indicated, we had a multi-class textual outcome with as many as 1900 distinct codes (however, as indicated above, this range of values was reduced down to 12 distinct outcomes for purposes of this exercise).

A compelling metaphor that helps explain the efficacy of these approaches lies in a comparison of the accuracy of single witness reports of events versus reports produced by a committee of experts. Most of us would tend to value the committee results when compared to a single observer. The classification results produced by multiple trees appear to produce a similar kind of accuracy in numerical prediction and classification tasks such as the tasks we undertook in this exercise.

### MULTIPLE TREES EXPLAINED

The basic idea of bootstrapping is to repeatedly analyze subsamples of data. There are many forms of bootstrapping methods that have been adapted for data mining, particularly in the area of decision trees. These include “bagging” (which stands for bootstrap aggregation) introduced by Leo Breiman in 1996 and “boosting” (or “Arcing”) introduced respectively by Freund and Schapire in 1995 and by Breiman in 1998. More recently, “random forests” (Breiman 2001) have also been introduced. *Random forests* randomize not only the observations that are selected on each of the multiple passes against the data but also the variables that are used for prediction. All these techniques attempt

to assemble more stable results through averaging. This can help offset the risk of widely-varying results that could be produced by single-model estimates. Boosting, in particular, addresses the problems that are produced in trying to work with rare instances in data. In a sense, boosting does what was done above when the “other” category was sampled; in effect, this “boosted” the contribution of the main 12 codes that were in the analysis by changing their relative frequencies upward even though their absolute frequencies remained the same.

The bagging and boosting approaches that were used in this exercise are diagrammed in Figure 2. In bagging, the approach is typically to form different random subsamples of the source data. In boosting, the approach is normally to reweight the contribution of subsampled observations. Observations are reweighted based on how frequently they have been misclassified in previous runs in the multi-tree sequence. Observations that have been poorly classified are given more weight than those that have been well classified. Intuitively we can see that boosting methods would tend to classify rare events better because, as the algorithm unfolds, rare and poorly classified observations are given more and more weight with respect to the other records in the data set and therefore have a greater chance of contributing to the predictive rules that are formed.

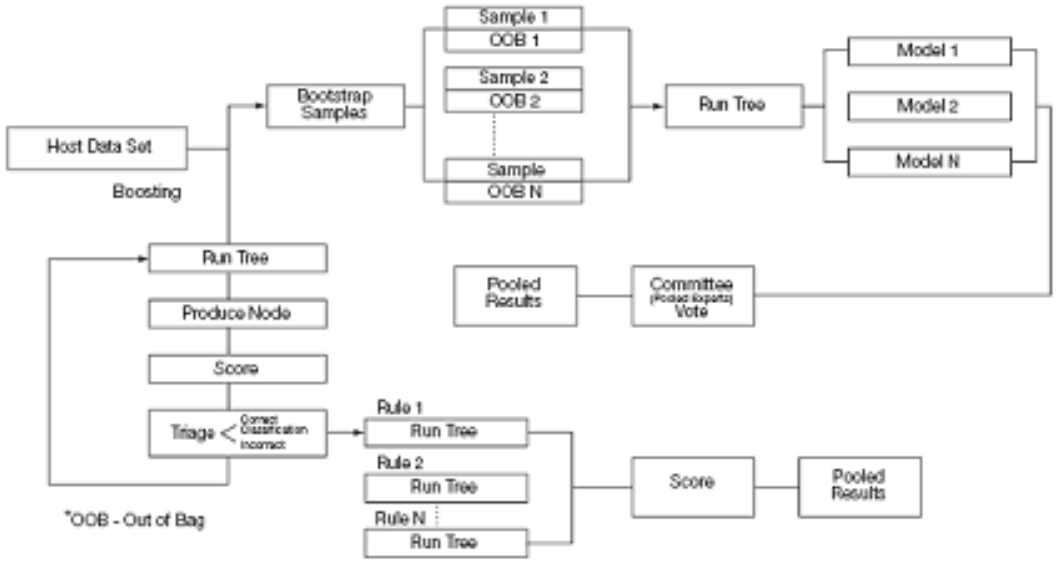


**Figure 2. Illustrative Multiple Tree Process Flows (Boosting and Bagging)**

Although multiple tree approaches have convincingly demonstrated their predictive efficacy when compared to single-tree approaches, they have not been adopted enthusiastically by practitioners. It is more difficult to display the pooled results of multiple trees than it is to present the relatively simple graph of a single tree. Multiple trees are more computationally complex and many of the perturbing and combing algorithms that are used in multiple trees are experimental and even idiosyncratic. The development of graphical methods to present pooled results promise to make them more presentable and interpretable and, as we gain experience in using them, we can expect that the experimental idiosyncrasies will gravitate towards known and reliable methods.

## RESULTS

The results of the four sets of analyses can be summarized by reference to the captured response chart shown in Figure 3. *Captured response* shows how many of the warranty codes on the test data were actually correctly predicted for the percentage of the data records that were processed by the respective scoring models (up to 100% shown on the right-hand side of the chart).

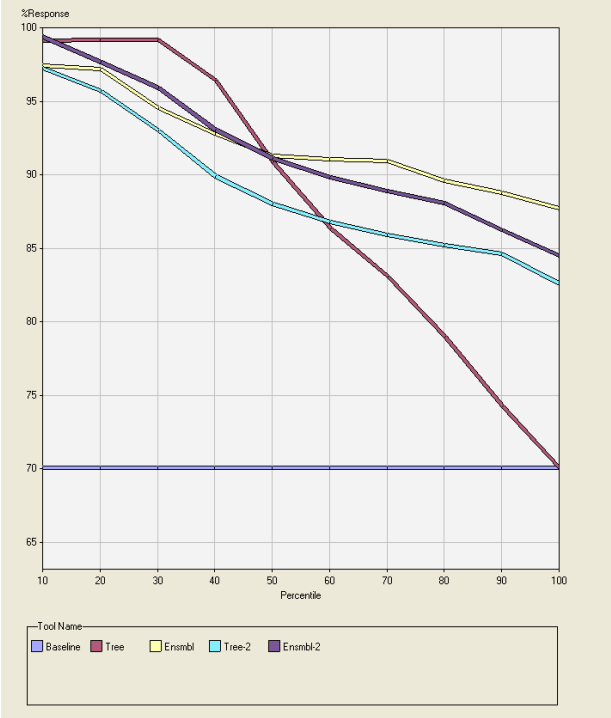


**Figure 3. Captured Response Comparison of Four Analytical Approaches**

The final results were as follows:

| | |
|---|---|
| Model with Quantitative Only Fields (shown as a red line in the figure) | 70% |
| Model with a Single tree and text and quantitative fields contained in the data (shown as a blue line in the figure) | 82% |
| Model with Bagged trees ( this includes text and quantitative fields and is shown as a purple line in the figure) | 84% |
| Model with Boosted trees (includes text and quantitative fields and is shown as a yellow line in the figure) | 87% |

As can be seen in the captured response chart, the single quantitative decision tree starts out well enough: it leads all other methods until the 50th percentile. As decisions become increasing difficult and as outcomes become increasingly rare, the quantitative-only approach begins to fade. At this point decisions benefit from the introduction of the kind of nuanced, qualitative specification that the text mining of results provides. Finally, multiple tree methods that represent a combination of all quantitative and qualitative prediction information pooled together produce the best results. Because boosting, in particular, is “tuned” to detect rare, difficult-to-classify cases, it is no surprise that the boosting results would produce the best overall results.

These results enable us to move from an overall accuracy rate of about 70% with the single tree (quantitative fields only) model to an accuracy rate that is close to 90% with the multi-tree boosted method. Of course, all our models substantially outperformed the observed reproducibility rate of only 50% that characterizes the current (manual) method of assigning warranty codes.

## NEXT STEPS

Based on the results to date we have been authorized to move into the classification of the full 1900 codes available on the warranty data set. Among other tasks, we will be exploring a number of refinements such as pre-clustering similar codes together. This approach is motivated by the success we observed in using quantitative information only as gauged by the captured response rate (up to the 50$^{th}$ percentile), as shown in Figure 3. It is also an approach that

was used successfully by Hewlett Packard in the classification and combining of product identification codes (required when the separate product lines of Compaq and Hewlett Packard were combined in the company merger of 2002).

## REFERENCES

Amit, Y. and D. Geman. 1996,1997. “Shape Quantization and Recognition with Randomized Trees.” *Neural Computation 9:*1545–1588.

Breiman, L. 1996. “Bagging Predictors.” *Machine Learning 24(2)*:123–140.

Breiman, L. 1998. “Arcing Classifiers.” *The Annals of Statistics 26(3)*:801–849.

Breiman, L. 2001. “Random Forests.” *Machine Learning 45(1)*:5–32.

Freund, Y. and R. E. Schapire. 1995. “A Decision-theoretic Generalization of On-line Learning and an Application to Boosting.” *Proceedings of the 2nd European Conference on Computational Learning Theory* (Eurocolt95). Barcelona, Spain. pp. 23–37.

## RECOMMENDED READING

Quinlan, J. R. 1996. “Bagging, Boosting, and C4.5.” *Proceedings of the Thirteenth International Conference on Artificial Intelligence*. pp. 725–730.

## ACKNOWLEDGEMENTS

I would like to warmly thank Manya Mayes, Ross Bettinger, and Russ Albright for their considerate review and thoughtful suggestions in the production of the final draft.

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author:

Barry de Ville
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513
Work Phone: 919-677-8000
Fax: 919-677-4444
Email: barry.deville@sas.com