# *Using Data Mining to Improve the Readership Experience At TechJournal*

**TERM PROJECT**
**CPSC 445b**
**Yale University**

***Al Bashawaty***
***May 6, 2007***

# *EXECUTIVE SUMMARY*

This project focuses on a data mining challenge facing a certain U.S. company. We will give this company the fictitious name - “TechPub”. TechPub, which is roughly fifteen years old, is principally in the business of providing sales leads to technology manufacturers. TechPub distinguishes itself from competitors by providing high quality leads as measured by their ability to generate above average response rates.

In order to produce these high quality leads, TechPub focuses the majority of its corporate effort on finding out who are the current IT purchasing decision makers in corporate America. Having identified these decision makers, TechPub goes a step further. It attempts to discover the issues of concern that are currently on minds of these decision makers.

TechPub produces an on-line technology journal which covers topics across the entire IT spectrum. This journal, which we will call “TechJournal”, is e-mailed to known active IT decision makers and prospective ones, alike. TechJournal is also is available as a stand-alone website. The content on this site is free to both registered readers and anonymous visitors. TechJournal is the primary lead discovery/data enrichment tool for TechPub.

By virtue of the nature of the lead-generation business, TechPub knows some basic attributes for every potential/actual reader it comes in contact with. The challenge, therefore, is to data mine the information TechPub has collected on solicitation and reader behavior, given these known attributes, to stimulate new readership and increased activity among existing readers.

This project focused on three key questions which lie at the heart of getting new readers for TechJournal and which result in the most active and satisfied readership base:

1. **“Read or Not”** - Given email recipient attributes, what is the likelihood of a visit to website?
2. **“Read More”** - Given registered readers’ attributes, which will be most active?
3. **“Read What**” - Given registered readers’ attributes, which stories will they be interested in?

These are the three questions that this project investigated.

This project was granted access by TechPub to over 10 Gigabytes of its data, subject to our promise to keep this data confidential. The data sample gave us the following three attributes which we can use as the classification classes for the three questions we are looking at. The classes are:

***Read or Not*** – A boolean value which is zero if prospective reader never visits the TechJournal website and one if the reader does visit after receiving the prospecting email;

***Read More*** – A boolean value which is zero if reader of TechJournal reads only one story or one if the reader reads more than one story; and

***Read What*** – An integer (46 values) corresponding to the content taxonomy class matching the subject matter of the story read.

As for features, five attribute groups were used in the data mining:

**Location**
**Industry**
**Size of Company**
**Recipient Title**
**Source of Recipient Name**

The features turned out to be largely uncorrelated with one another in our data sample.

Numerous issues were encountered during data preparation. The data was noisy, dirty and unevenly populated. Consistency of the data was marginal in some spots, so I had to run double checks to make sure that the values seemed plausible. The data was incomplete, eliminating the majority of records from consideration. Furthermore, a great deal of work had to be done to come up with useful discrete values for all of the feature groups.

As for data mining methodology, all work was done in R using the Rattle interface. For the three questions, I employed four techniques: decision tree induction (rpart), boosting (gbm) with 100 iterations, random forests (rf) of 500 trees and support vector machines (ksvm) with the Gaussian kernel.

The data samples were consistently split 70% into a training set and 30% into the test set for validation purposes.

At the end of each set of runs, I compared the results of the different methods in terms of error rates on the test set, their resulting confusion matrices, and in the case of the first two questions which have binary classes, ROC curves.

Here are the results our analysis produced across the three questions:

***Error rates on test data set***

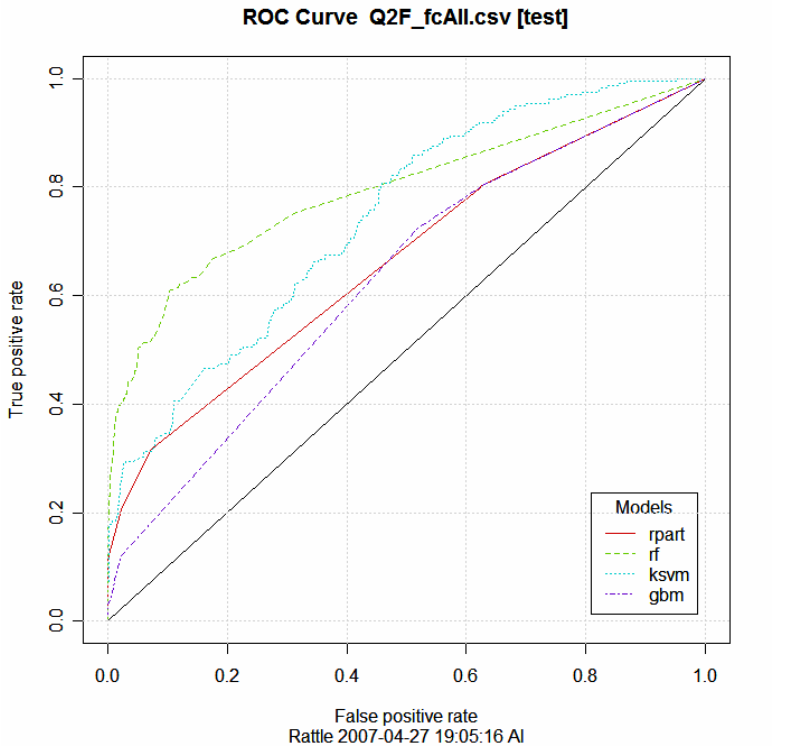
| | *Decision Tree* | *Random Forest* | *Boosting* | *SVM* |
|---|---|---|---|---|
| **Read or Not** | **0.0437** | **0.0408** | **0.0427** | **0.0460** |
| **Read More** | **0.2246** | **0.1737** | **0.2492** | **0.2330** |
| **Read What** | **0.6691** | **0.5164** | **0.6121** | **0.5700** |

As for confusion matrices and ROC curves, let’s look at each question one by one.

**Q1. “Read or Not”:**

CONFUSION MATRICES

| | Decision Tree | | Random Forest | | SVM | |
|---|---|---|---|---|---|---|
| | *Actual* | | *Actual* | | *Actual* | |
| *Predicted* | *0* | *1* | *0* | *1* | *0* | *1* |
| *0* | 4633 | 207 | 4632 | 192 | 4639 | 224 |
| *1* | 6 | 27 | 7 | 42 | 0 | 10 |

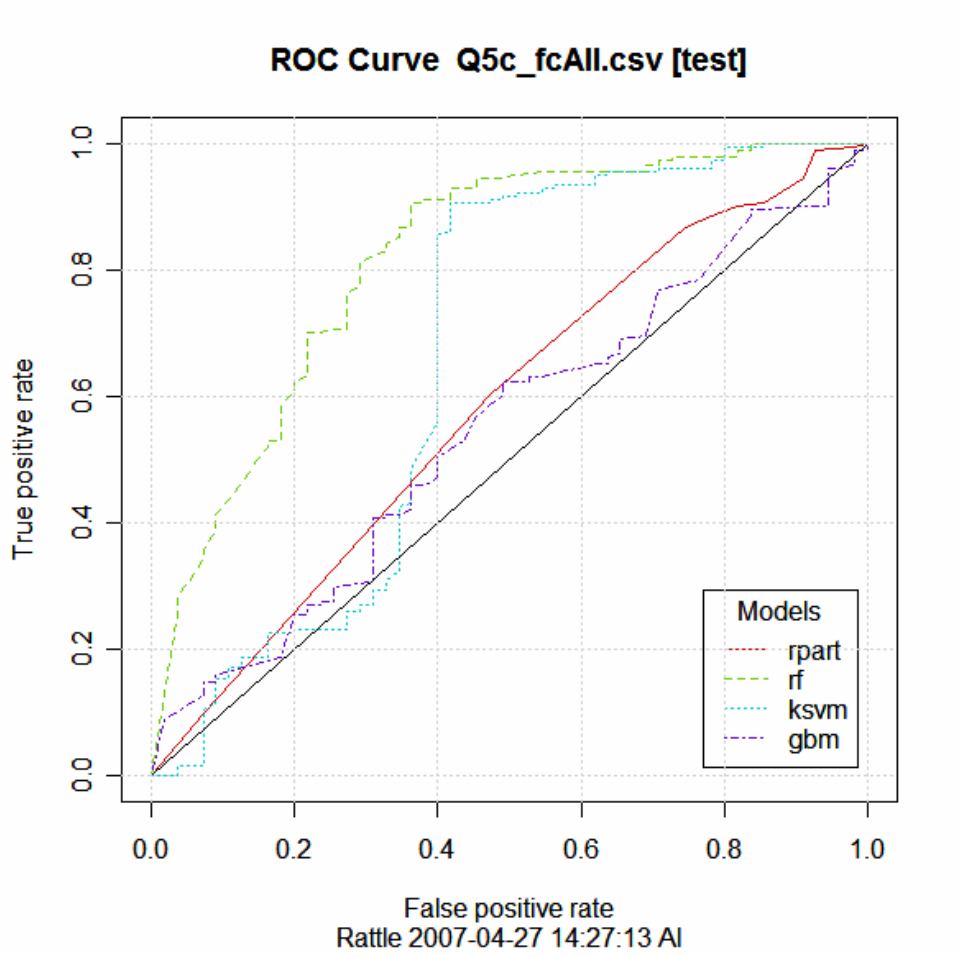



The random forest method produced the strongest model as can be clearly seen by in the ROC curve.

**Q2. “Read More”:**

CONFUSION MATRICES

| | Decision Tree | | Random Forest | | SVM | |
|---|---|---|---|---|---|---|
| | *Actual* | | *Actual* | | *Actual* | |
| *Predicted* | *0* | *1* | *0* | *1* | *0* | *1* |
| *0* | 4 | 2 | 16 | 2 | 0 | 0 |
| *1* | 51 | 179 | 39 | 179 | 55 | 181 |




Once again, the random forest model was superior returning a model with decent classification accuracy.

## Q3. “Read What”:

***Random Forest Confusion Matrix***

| | 0 | 1 | 2 | 5 | 6 | 9 | 10 | 12 | 13 | 16 | 17 | 18 | 20 | 24 | 25 | 27 | 30 | 42 | 44 | 45 | 46 | class.error | num |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ***0%*** | ***2*** |
| **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | ***100%*** | ***2*** |
| **2** | 1 | 0 | 10 | 0 | 2 | 0 | 0 | 1 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 1 | 0 | 5 | 0 | 8 | 0 | ***68%*** | ***31*** |
| **5** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 4 | 0 | ***100%*** | ***8*** |
| **6** | 0 | 0 | 2 | 0 | 12 | 2 | 0 | 4 | 0 | 1 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 13 | 0 | 15 | 0 | ***76%*** | ***51*** |
| **9** | 0 | 0 | 0 | 0 | 1 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ***20%*** | ***5*** |
| **10** | 0 | 0 | 0 | 0 | 1 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | ***60%*** | ***5*** |
| **12** | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 40 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 12 | 0 | 11 | 1 | ***42%*** | ***69*** |
| **13** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 4 | 0 | ***100%*** | ***11*** |
| **16** | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 2 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 5 | 2 | ***79%*** | ***14*** |
| **17** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | ***25%*** | ***4*** |
| **18** | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 2 | 0 | 0 | 1 | 1 | 1 | 10 | 0 | 5 | 2 | ***93%*** | ***27*** |
| **20** | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | ***100%*** | ***3*** |
| **24** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ***0%*** | ***0*** |
| **25** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ***0%*** | ***0*** |
| **27** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 3 | 0 | 1 | 1 | 1 | 0 | ***63%*** | ***8*** |
| **30** | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 1 | 0 | ***100%*** | ***6*** |
| **42** | 0 | 0 | 2 | 0 | 8 | 1 | 0 | 9 | 0 | 1 | 0 | 3 | 0 | 0 | 0 | 1 | 0 | 92 | 0 | 25 | 3 | ***37%*** | ***145*** |
| **44** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 2 | ***100%*** | ***4*** |
| **45** | 0 | 0 | 2 | 1 | 3 | 0 | 0 | 6 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 24 | 0 | 79 | 3 | ***34%*** | ***120*** |
| **46** | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 2 | 12 | 14 | ***60%*** | ***35*** |

***Support Vector Machine Confusion Matrix***

| *% Predictions* | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ***Were Accurate*** | | True | | | | | | | | | | | | | | | | | | | | | | | | | |
| | **Pred** | **0** | **1** | **2** | **5** | **6** | **7** | **9** | **10** | **12** | **13** | **16** | **17** | **18** | **20** | **24** | **25** | **27** | **30** | **33** | **42** | **43** | **44** | **45** | **46** |
| ***67%*** | **2** | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| ***60%*** | **6** | 0 | 0 | 0 | 0 | 15 | 0 | 0 | 1 | 2 | 1 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| ***40%*** | **12** | 0 | 0 | 3 | 0 | 9 | 0 | 0 | 1 | 33 | 1 | 0 | 1 | 5 | 0 | 0 | 0 | 1 | 2 | 0 | 12 | 1 | 3 | 7 | 4 |
| ***83%*** | **16** | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ***45%*** | **42** | 3 | 0 | 21 | 5 | 29 | 0 | 2 | 1 | 34 | 3 | 5 | 1 | 17 | 1 | 0 | 0 | 5 | 4 | 1 | 151 | 0 | 1 | 44 | 9 |
| ***39%*** | **45** | 0 | 2 | 19 | 6 | 20 | 3 | 3 | 4 | 18 | 10 | 10 | 0 | 16 | 2 | 2 | 1 | 2 | 5 | 0 | 42 | 1 | 3 | 126 | 28 |
| ***67%*** | **46** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 6 |
| ***% In Class Pred ----------:*** | | **0%** | **0%** | **4%** | **0%** | **20%** | **0%** | **0%** | **0%** | **37%** | **0%** | **25%** | **0%** | **0%** | **0%** | **0%** | **0%** | **0%** | **0%** | **0%** | **73%** | **0%** | **0%** | **71%** | **12%** |

Unlike in the previous two questions where we had binary classes, it was not easy to generate a ROC curve for these results. So, the best way to compare these methods is through their respective error rates on the test set and through the confusion matrices they generate. The Random Forest method generates the lowest area and also has a better distribution of results across the confusion matrix, in that it classifies into all 31 active categories in the taxonomy.

In conclusion, the data sample was sufficient to allow us to generate working models to predict the classifications of interest in each of the three areas we focused upon.

During the data cleaning stage, we found that a large amount of our data records were disqualified by virtue of having one or more missing feature values. Being able to enrich these records in the future would potentially provide a great deal more data to rerun this analysis with. Alternatively, since we found that some variables were less valuable than others, we could go back and relax this complete feature set restriction and rerun the analysis once again. The downside of this latter approach however, is that we will have precluded ourselves somewhat from getting an accurate sense of how valuable the excluded feature data could be on a broader sample.

In the analysis on all three questions, the random forest proved to be the most effective technique for this data set. This approach resulted in the lowest error rates when applied to the test data sample, as well as the best ratio of true positives to false positives in the first two cases.

The **"Read or Not"** question presents a challenge because of the 19:1 dominance of email recipients who do not become readers over those who do. The random forest method did the best job of dealing with this low class representation problem. The confusion matrix showed that this ensemble method was able to make a significant number of classifications in all four states. The simple decision tree induction method gave us a set of rules which were not bad, as simple heuristics go.

We were able to come up with a reasonable ability to classify instances of the test set in the **"Read More"** question (the random forest method produced a 17% error rate). However, when one steps back and thinks beyond the numbers, it becomes clear that we are missing one critical aspect necessary to properly attack this question: the dimension of time. As we noted above, by adding the notion of how long it takes readers to go from reading one story to another, we can get a much better definition of an "active" reader. Armed with this new classifier definition, this work should be repeated when a more complete dataset of timestamp information is available.

The analysis for all three questions would greatly benefit from a greater number of records with complete feature data as well as more feature groups, in general, to work with. This problem was felt most acutely in the **"Read What"** analysis. With forty seven content classes, we simply need more features and more records to be able to really be able to predict which readers will read what. The random forest method was able to get us to the point of being able to make an even money prediction as to which content group a reader would select. While a promising result, there is much room for further improvement on this question.

Moreover, the results of our analysis in this area lead to the recommendation that TechPub continue to actively refine its content taxonomy. I have two specific suggestions on this matter. First, the taxonomy has been built from the perspective of how people who write about technology think about the grouping of subject matter. However, increased domain insight into how people who read this content think about the grouping of subject matter would be very valuable. Secondly, the employment of text mining techniques to build a taxonomy of content similarity could also prove to be quite enlightening.

Finally, the **"Read What"** question should be tackled with a more sophisticated approach overall. Specifically, we need to move beyond simple semantic grouping of stories and into text mining techniques which model the exact word groupings that prospective readers see in the headline or abstract of the story that they peruse. It is this text on which they base the decision of whether to read or not. This is where they are giving TechPub the first clues as to their real interests.

By again pairing readership info with timestamp data, we could text mine the actual stories and pair that information with data on how long the reader spends in that story. This would make for a much better gauge of true interest. These techniques should provide meaningful insight.

Unfortunately, they were simply beyond the scope of this project.

*Full Discussion Of Project and Results*

## Background

This project focuses on a data mining challenge facing a certain U.S. company. We will give this company the fictitious name - “TechPub”. TechPub, which is roughly fifteen years old, is principally in the business of providing lists of sales leads to technology manufacturers. TechPub distinguishes itself from competitors by providing high quality leads as measured by above average response rates that the manufacturers experience having rented the leads.

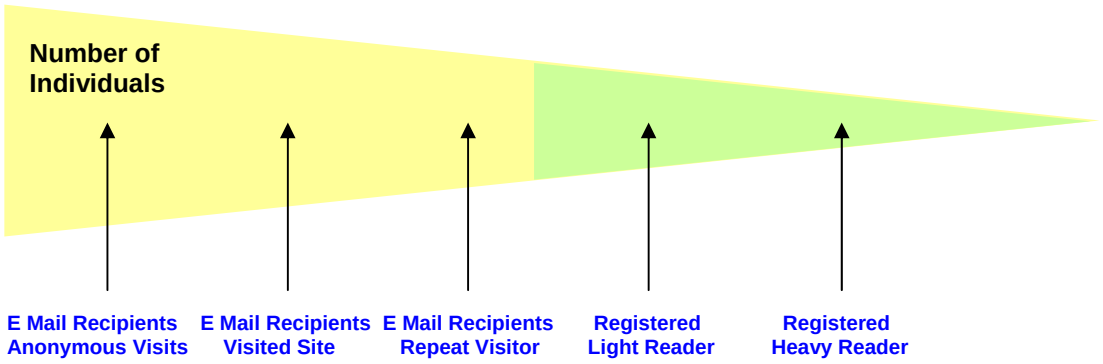
In order to produce these high quality leads, TechPub focuses the majority of its corporate effort on finding out who in corporate America are the current IT purchasing decision makers. Having identified these decision makers, TechPub goes a step further. It attempts to discover the issues of concern that are currently on minds of these decision makers.

TechPub produces an on-line technology journal which covers topical issues across the entire IT spectrum. This journal, which we will call “TechJournal”, is e-mailed to known active IT decision makers and prospective ones, alike. TechJournal is also available as a stand-alone website. The content on this site is free to both registered readers as well as anonymous visitors. TechJournal is the primary lead discovery/data enrichment tool for TechPub.

Over the past year, TechPub has upgraded the technology and publishing infrastructure supporting TechJournal. As a result, the company now is in a position where it is able to deliver each issue of TechJournal with its content personalized to the known interests of a given reader or prospective reader.

In order to stimulate a constantly growing readership base, TechPub continuously emails free copies of TechJournal out to suspected IT decision makers with whom it has no current relationship. As a result, TechJounal is constantly in the hands of people with a broad spectrum of interest and awareness of the product. Over time, email recipients migrate down the following funnel of interest until they either fall out of the process or end up as readers:

**Figure 1 – TechJournal’s Readership Cultivation Model**

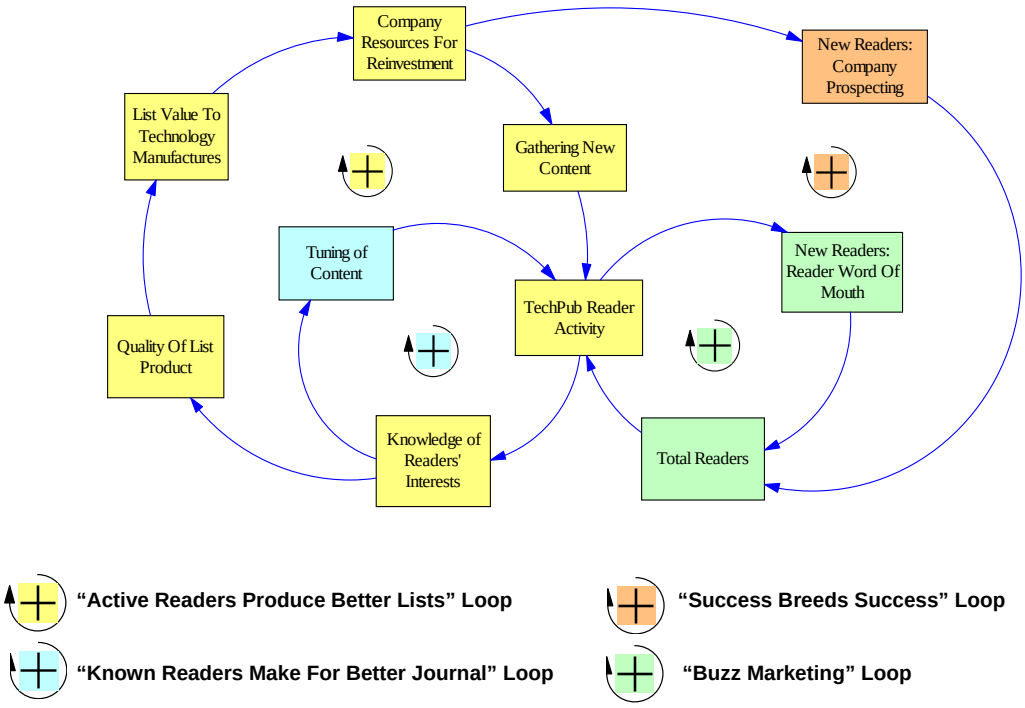



With this funnel in mind, it is clear that one of TechPub’s critical goals for success is to stimulate as many prospective readers into become first readers and then active readers of TechJournal.

## Business Model

By having TechJournal act as a free lead-generation and data-enrichment tool, TechPub has ended up with a business model which has at least four very interesting positive feedback loops. A gross oversimplification of these feedback loops is shown in Figure 2.

**Figure 2 – Positive Feedback Loops in TechJournal's Business Model**



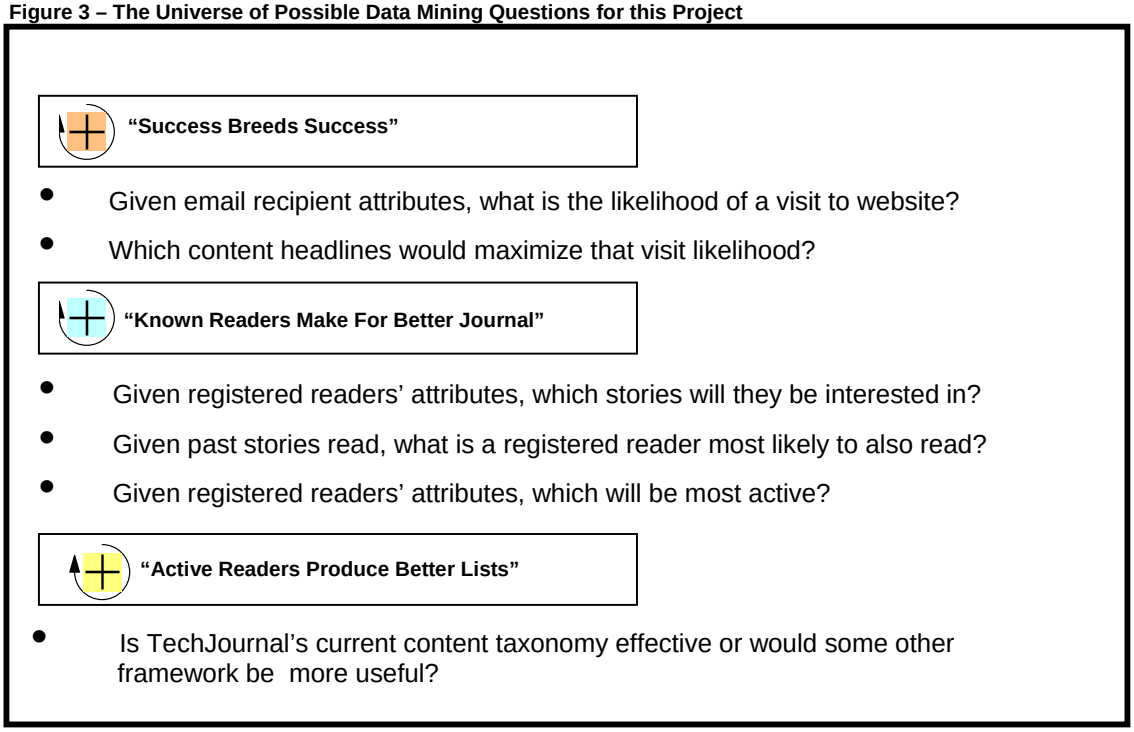

These four positive feedback loops are important. If one can stimulate one of these loops to grow, the net result is that revenues will grow faster than costs at TechPub. Note that each of these four feedback loops pass through one key variable: TechPub Reader Activity. Therefore, a key goal for TechPub is to stimulate as much reader activity (defined as number of stories read per visit) as possible.

## Questions of Interest

By virtue of the nature of the lead-generation business, TechPub knows some basic attributes for every potential/actual reader it comes in contact with. The data mining challenge, therefore, is to data mine the information TechPub has collected on solicitation and reader behavior, given these known attributes, to stimulate new readership and increased activity among existing readers.

Specifically, there was an identified universe of six possible questions that this project could shed light on. Each aimed at stimulating one of these key areas in the readership "funnel" or in the business model "feedback" processes. These six questions are shown in Figure 3.

**Figure 3 – The Universe of Possible Data Mining Questions for this Project**




At the outset of this project, it was clear that we would not have the time or data to sufficiently attack all six of these questions. Therefore, the approach followed was to make a decision on which subset of questions to work on once the data was cleaned and prepared for mining.

Having reached that stage, it became clear that there were three questions we could work on:

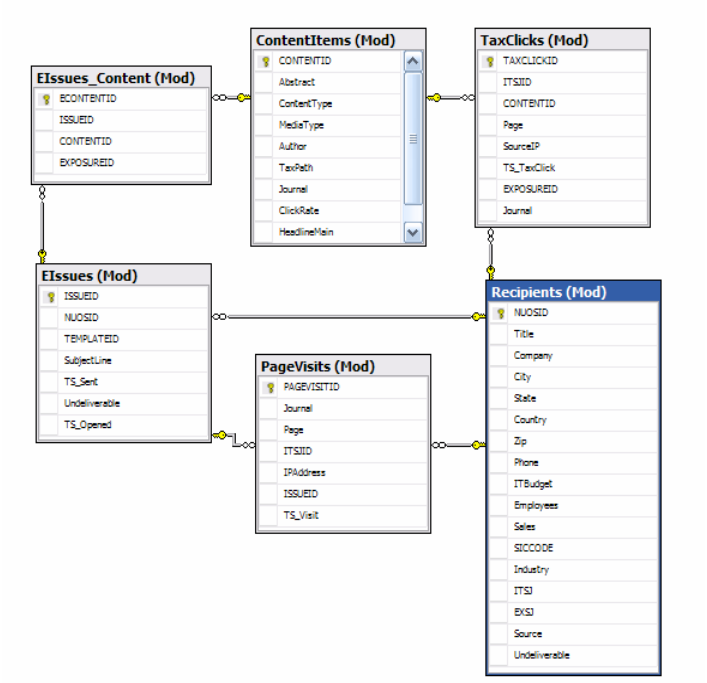
1. **"Read or Not"** - Given email recipient attributes, what is the likelihood of a visit to website?
2. **"Read More"** - Given registered readers' attributes, which will be most active?
3. **"Read What"** - Given registered readers' attributes, which stories will they be interested in?

These are the three questions that this project investigated.

## Data

This project was granted access by TechPub to over 10 Gigabytes of its data, subject to our promise to keep this data confidential. The data is grouped into the six data tables shown in Figure 4.

**Figure 2 – Schema of Our Data to Mine**




Here is a brief description of each table:

**EIssues** ***(713,110 records)*** – Contains a record for each email sent to a prospective or current reader containing an issue of TechJournal

**ContentItems** ***(69 records)*** – Contains a record for each story item which we have full information for in any given issue of TechJournal

**EIssues_Content** ***(2,185,664 records)*** – A linker table facilitating a many to many relationship between the first two tables. of no interest from a data mining vantage pt.

**TaxClicks*****(9,385 records)*** – Contains a record for each instance of a readers selecting a content item to read. The item is semantically classified in this table into a standard taxonomy.

**PageVisits*****(43,580 records)*** – Contains a record for each reader page visit.

**Recipients*****(195,455 records)*** – Contains a record for each email recipient or reader with all known attribute values for each reader

*Classes for Categorization*

This data sample gives us the following three attributes which we can use as the classification classes for the three questions we are looking at. The classes are:

***Read or Not*** – A boolean value which is zero if prospective reader never visits the TechJournal website and one if the reader does visit after receiving the prospecting email

***Read More*** – A boolean value which is zero if reader of TechJournal reads only one story and one if the reader reads more than one story

***Read What*** – An integer (46 values) corresponding to the content taxonomy class matching the subject matter of the story read.

As noted earlier, TechPub has created a hierarchical tree of technology subject matter. That tree ranges from three to twenty one levels deep. It is the content taxonomy which this project used as the basis for putting content into semantic classes. Given the size of this tree, however, I had to

simplify the taxonomy to a common depth in the tree. I choose to pick three levels of depth as the cutoff point, resulting in 46 possible content classes. I picked this level simply because it gave me the right order of magnitude of classes and yet still kept the conceptual specificity in any single area of knowledge consistent. My possible choices of levels and the resulting number of classes are shown in Figure 5a. The resulting content classes are shown in figure 5b.

**Figure 5a – Possible Taxonomy Simplification Levels**

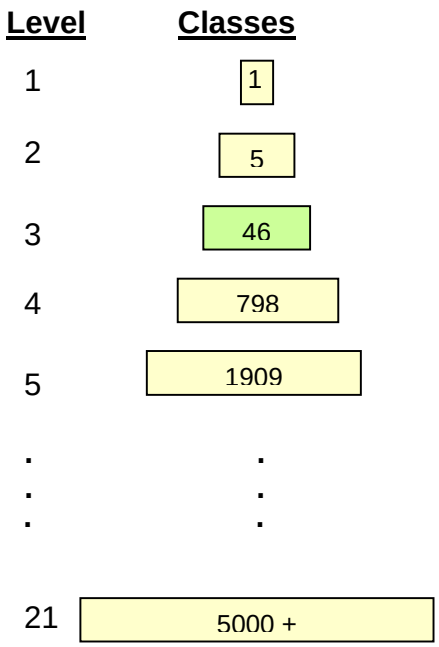
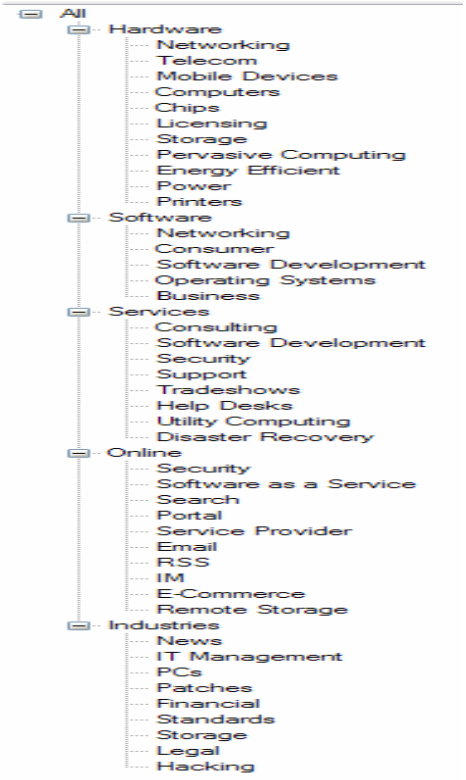



**Figure 5b - Resulting Content Classes**



*Available Attributes*

This data sample provides us the following attributes to work with as well:

| | Reader Attributes | Content Attributes | Format Attributes |
|---|---|---|---|
| **Primary Key** | Recipient ID<br>IP Address | Content ID<br>Issue ID | |
| **Data Mining Attributes** | Title<br>City<br>State<br>Country<br>Zip<br>Phone<br><br>IT Budget<br>Employees<br>Sales<br>SIC Code<br>Industry<br><br>Time Sent<br>Time Opened<br>Time of Visit<br>Time Content Click | Abstract<br>Headline Main<br>Content Type<br>Media Type<br>Author<br>Content Taxonomy<br>Click Rate | Template Type<br>Media Type (HTML, Or Video) |

*Feature Selection*

Unlike many data mining projects, this study was not burdened with too many features to use. Therefore, techniques like PCA were not required to figure out which features were the best to incorporate into the analysis. Instead, this project, if anything, struggled with too few features at times. The only two feature groups not used above were the Time features and the Format features.

I did not use the Time features because I did not have a complete enough set of time stamps to do anything meaningful with this feature group. As for the Format features, closer inspection of the data revealed that this information was too sparsely and inconsistently populated to be of much use. Moreover, I felt that is was more sensible to start simple. I did so by trying to understand the nature of who was reading what rather than to complicate this analysis with the added dimension of what format that content was presented with.

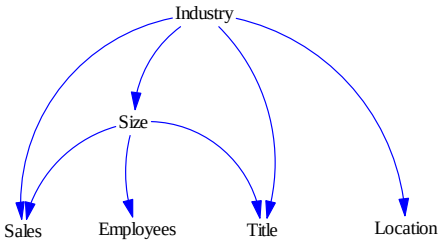
With these exclusions in mind, all of the remaining attributes were used. They can be conceptually thought of as falling into the following feature groups:

**Location**
**Industry**
**Size of Company**
**Recipient Title**
**Source of Recipient Name**

These five feature groups were employed in the data mining exercises.

As a final measure, I checked for feature correlation amongst these groups. While I planned to use all of the features, I decided that it would still be helpful to know how correlated they were

with one another for data mining methodology selection. Prior to doing my calculation, I had the following intuition as to their correlation:




Within this data sample, however, it turned out that this was not the case. The features were largely uncorrelated with one another. The following table shows the cross feature group correlation:

```
Correlation Summary.

Note that only correlations between numeric variables are reported.

                LocGrpID Size_Employees         RIC  Read_Class  Title_Code
LocGrpID       1.0000000     0.26310210 -0.13262760 -0.12134580 -0.08417289
Size_Employees 0.2631021     1.00000000 -0.13448772 -0.10155000 -0.02738014
RIC           -0.1326276    -0.13448772  1.00000000  0.06366969  0.04575218
Read_Class    -0.1213458    -0.10155000  0.06366969  1.00000000  0.04625975
Title_Code    -0.0841729    -0.02738014  0.04575218  0.04625975  1.00000000

Generated by Rattle 2007-04-23 16:50:11 Al
=======================================================================
```

**Data Preparation Issues**

I encountered numerous issues during data preparation. The data was noisy, dirty and unevenly populated. Consistency of the data was marginal in some spots, so I had to run double checks to make sure that the values seemed plausible.

The bigger issues were two-fold: data completeness and bucketing.

*Completeness*

First, since we started with very few features to work with, it was important that every record in the final test and training sets have fully populated features. This meant that while I started with a relatively large number of incomplete records, by the time that I weaned the sample down to just those records which had all features, the size of my data set had dropped dramatically.

After raw data cleaning, I started with 60,751 distinct recipients. Missing values knocked out records in the following fashion:

**Records knocked out by empty:**

| | |
|---|---|
| **titles -** | **27,742** |
| **sales -** | **9,516** |
| **industry/SIC -** | **7,860** |
| **employees -** | **409** |
| **location -** | **0** |

The final number of complete records available for each analysis was as follows:

**Read or Not** – 16,241 records
**Read More** - 789 records
**Read What** - 789 records

*Bucketing*

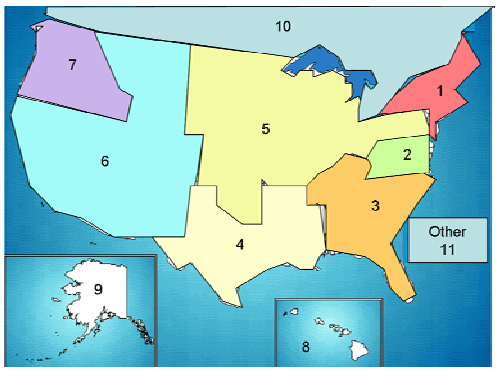
Secondly, we need to come up with standardized discrete values for the titles, locations, industry, and size feature groups.

**Size:** There were two size measures we could rely on: number of employees and annual sales. Employees turned out to be the more complete and useful measure. I set up seven categories for size by number of employees:

**Size Categories**

| Num Employees | Size |
|---|---|
| 0-10 | Very_Small |
| 10-99 | Small |
| 100-999 | Small_Mid |
| 1000-9999 | Mid |
| 10000-49999 | Large |
| 50000-99999 | Very_Large |
| >100000 | Largest |

**Location:** I grouped all locations into 11 categories. The map below shows the grouping scheme used. This scheme was based on my intuition as to where there was similarity amongst technology firms in the US.

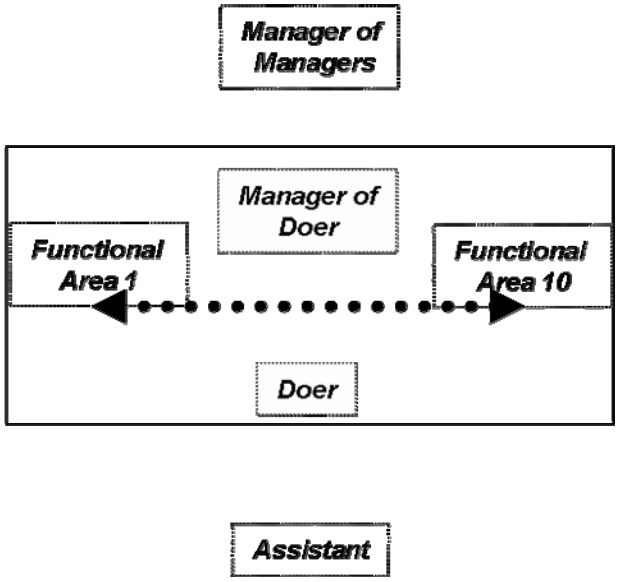



Title: The database had self-reported titles. There were 91 distinct titles. Some were garbage (i.e. “Smartest man in the world”). I standardized the titles by breaking them into two dimensions. The first dimension captures the seniority implied by the title. I had four seniority categories: (1)

Assistant, (2) Worker, (3) Manager of Worker, (4) Manager of Managers. The second dimension captures the functionality noted in the title. I used ten functional groups:

| Group | Function |
|---|---|
| | Business |
| 0 | Information |
| 1 | Data |
| 2 | Desktop |
| 3 | Communications |
| 4 | Network |
| 5 | Sys Admin |
| 6 | Sys Arch |
| 7 | Development |
| 8 | Security |
| 9 | Web |
| 10 | Generalist |

If no function was listed, the functional code was set to Generalist. Additionally, if the seniority of the title was either Assistant or Manager of Managers, the Generalist function was also selected, so as not to dilute the meaning of the categories. Figure 6 gives a visual representation of this title breakdown process.



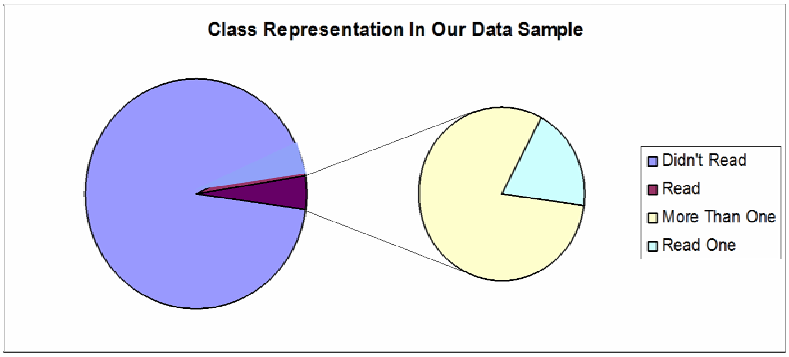

For example, if the reported title is “Manager of Network Operations”, the title code is set to 2 (manager of worker) followed by 4 (network function) for a full title code of 24.

**SIC/Industry**: The standard SIC Code for industry classification is six numbers or more which produces too many discrete values to be usable for data mining. I took the first three digits of the SIC code and grouped them into what I called Reduced Industry Codes (36 discrete values).

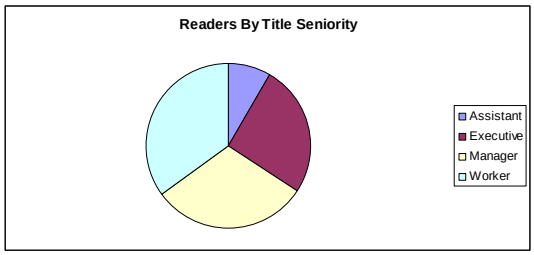
| RIC | SIC | Reduced_Industry_Name |
|---|---|---|
| 11 | 310 - 350 | Basic industries \| Basic Materials |
| 12 | 280-309 | Basic industries \| Chemical |
| 13 | 150-199 | Basic industries \| Construction |
| 14 | 351-356 | Basic industries \| Equipment/Manufacturing |
| 15 | 240-259 | Basic industries \| Lumber/Furniture |
| 16 | 260 - 266 | Basic industries \| Paper/Forest products |
| 17 | 267-269 | Basic industries \| Plastics |
| 18 | 220-239 | Basic industries \| Textiles/Apparel |
| 19 | 400-479 | Basic industries \| Transportation |
| 21 | 200-219 | Consumer \| Food/Beverage |
| 22 | 519-590 | Consumer \| Retail |
| 23 | 592-599 | Consumer \| Retail |
| 24 | 500-518 | Consumer \| Wholesalers |
| 31 | 600-616 | Financial institutions \| Banks |
| 32 | 617-699 | Financial institutions \| Insurance/Non-bank fin. |
| 41 | 591 | Healthcare \| Drugs and Supplies |
| 42 | 800-809 | Healthcare \| Health Services |
| 43 | 383-399 | Healthcare \| Medical Equipment |
| 51 | 371-376 | Media & technology \| Aerospace |
| 52 | 357-370 | Media & technology \| High tech |
| 53 | 270-279 | Media & technology \| Media related |
| 54 | 377-382 | Media & technology \| Other Tech |
| 55 | 480-489 | Media & technology \| Telco/Cellular |
| 61 | 100-130 | Natural resources \| Metal/Mining |
| 62 | 131-149 | Natural resources \| Oil/Gas |
| 63 | 490-499 | Natural resources \| Utilities |
| 70 | 730-736 | Services \| Advertising |
| 71 | 737-749 | Services \| Computer Software |
| 72 | 870-871 | Services \| Engineering |
| 73 | 781-799 | Services \| Entertainment |
| 74 | 889-998 | Services \| Governmental |
| 75 | 750-780 | Services \| Mechanical |
| 76 | 999 | Services \| Other Services |
| 77 | 810-869 | Services \| Other Services – Education |
| 78 | 872-888 | Services \| Other Services |
| 79 | 700-729 | Services \| Personal |

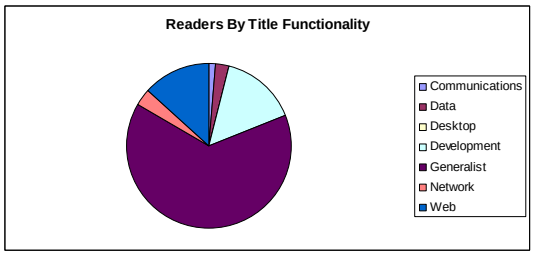
*Email Recipient/Reader Breakdown*

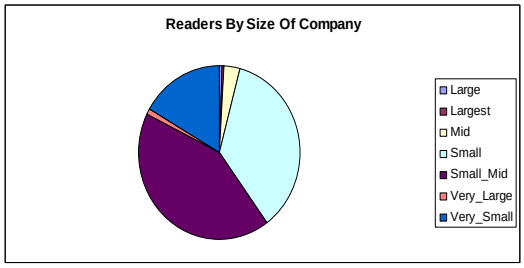
Only five percent of all email recipients have moved down the funnel into readership. Of those 786 readers, 81% have read more than one story. Of those that have read more than one story, the average number of stories read is close to 5 stories. These numbers suggest that we will have a representation issue in data mining the Read or Not question, since the Read class is so small relative to the Did Not Read class (19:1). This will have to be addressed with our data mining methodology. The figure below summarizes these relationships:

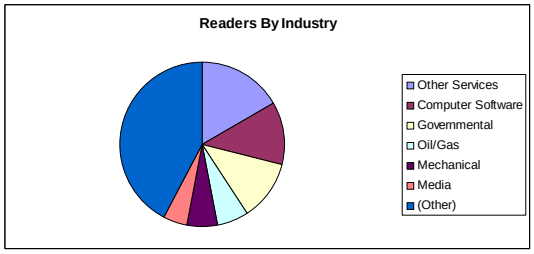


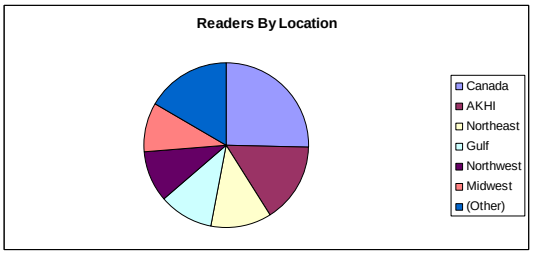

The breakdown of our Readers by feature group is summarized by the following charts:




This breakdown shows a good percentage of workers and direct managers, so we will see a strong functional dimension to the Title features. Additionally, one notes a good industry and geographic mix, as well as a heavy concentration of readers from small and small to mid size companies.

## Data Mining Methodology

As noted earlier, since we do not have a large number of possible features to choose from, we will are not concerned with dimensionality reduction. We are faced with the opposite problem of not having potentially enough features.

All data mining work was done in R using the Rattle interface. For the three questions, I employed multiple techniques.

I started simply with Decision Tree induction (using rpart). The trees were induced with the minimum complexity penalty value to encourage the most detailed tree to be produced. Then, the trees were induced with the default complexity value (cp = .01) to pare the trees back to levels where over-fitting is less likely. The best thing about this technique was that it gave rules that were understandable for predicting classes.

In an effort to deal with cases of low class representation, boosting (via gbm) was employed to see if that would improve the results. One hundred iterations were used during the boosting.

Finally, in attempt to try to find more subtle patterns, I used two other techniques: a random forest with 500 trees where two variables at a time where considered at each point and a support vector machine with a Gaussian kernel (ksvm). While the svm is best designed to work with binary classes, I did try it on the Read What question (with 46 classes) as well. I just wanted to see how it would fare.

The data samples were consistently split 70% into a training set and 30% into the test set for validation purposes.

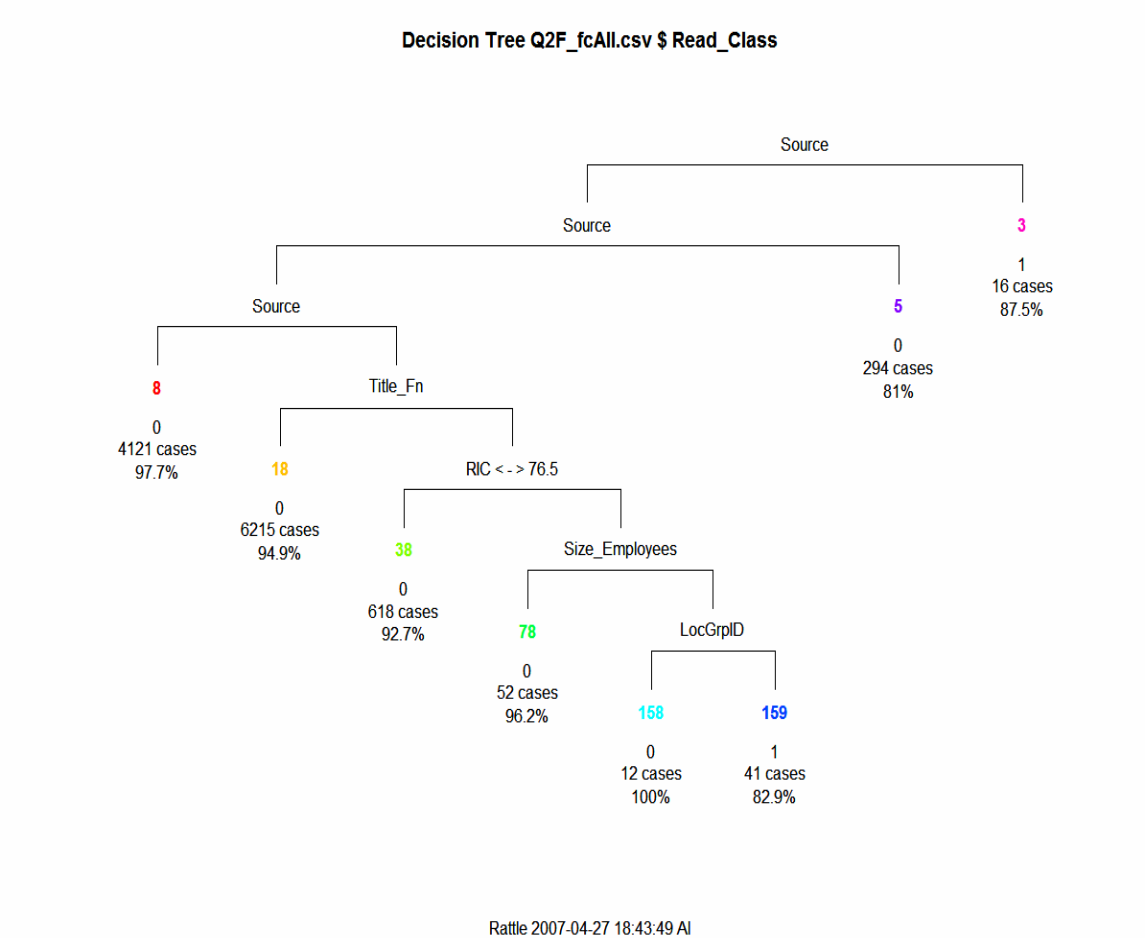
At the end of each set of runs, I compared the results of the different methods in terms of error rates on the test set, their resulting confusion matrices, and in the case of the first two questions which have binary classes, ROC curves.

## Results and Analysis

**Q1. "Read or Not"** - Given email recipient attributes, what is the likelihood of a visit to website?

---

***Method 1a: Decision Tree Induction – no pruning (cp = .0001)*** ***Error rate = .0429***




This decision tree generates two rules (#3, #159) which predict email recipients who will become readers. These rules are:

| ***Rule Number*** | ***3*** | |
|---|---|---|
| **Recipient Attribute** | | **Attribute Values that pass** |
| | | |
| Data Source | | The company's original dataset or website data |

OR

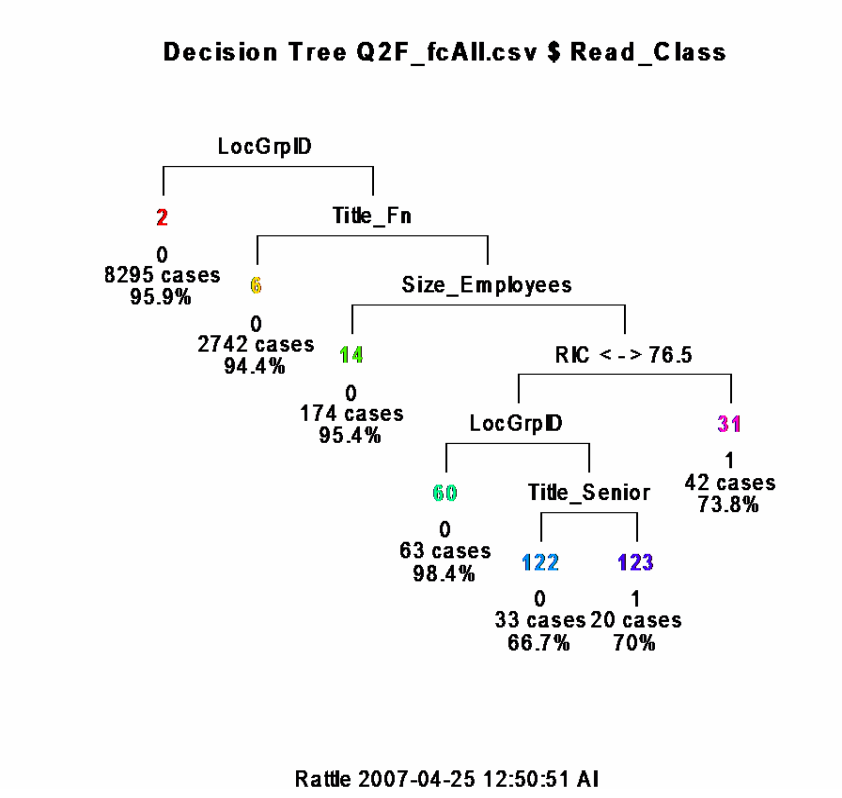
| ***Rule Number*** | ***159*** | |
|---|---|---|
| **Recipient Attribute** | | **Attribute Values that pass** |
| | | |
| Data Source | | Other than the company's original dataset or website data |
| Title (function) | | Software Development or Communications |
| Industry | | Other Services |
| Company Size | | Small or Very Large Employee Base |
| Location | | Gulf, or Mid Atlantic States |

If an email recipient passes either of these two rules, then they are likely to become a reader.

Discussion: In addition to thinking that this tree is overfit due to the low complexity penalty, this result caused me to focus on the need to drop source as a feature in the analysis of this question. The data sources that were highly predictive have been exhausted, so there is little predictive value going forward in continuing to include this feature in the analysis. TechPub will not be getting any new leads from these highly predictive sources. A second observation is that the error rate is misleadingly low. There are only 5% of the sample recipients who become readers. Therefore, even an extremely simple rule like predict that everyone is a non-reader would only have a 5% error rate.

---

***Method 1b: Decision Tree Induction – pruning (cp = .005), no Source  Error rate = .0437***



This decision tree finds two rules (#31, #123) which predict email recipients who will become readers. These rules are:

| ***Rule Number*** | ***31*** | |
|---|---|---|
| **Recipient Attribute** | | **Attribute Values that pass** |
| Location | | Gulf, Midwest, or South region |
| Data Source | | Other than the company's original dataset or website data |
| Title (function) | | Software Development, Networking or Communications |
| Industry | | Other Services |
| Company Size | | Small |

OR

| ***Rule Number*** | ***123*** | |
|---|---|---|
| **Recipient Attribute** | | **Attribute Values that pass** |
| Location | | Midwest, or South region |
| Title (function) | | Software Development, Networking or Communications |
| Company Size | | Small |
| Industry | | Any Industry other than Other Services |
| Title (seniority) | | Manager |

If an email recipient has these attributes, then the prospect is likely to become a reader.

Discussion: The tree pruning resulted in a very small increase in error, but resulted in a set of rules which are more useful going forward.

---

***Method 2: Random Forest – 500 Trees, no Source Error rate = .0408***

Discussion: The table below shows, on average, how much each variable was able to assist in identifying unique classifiers.

| | **MeanDecreaseGini** |
|---|---|
| Title_Senior | 29.12 |
| Title_Fn | 35.37 |
| LocGrpID | 60.49 |
| RIC | 110.36 |
| Size_Employees | 48.16 |
| Source | 51.53 |

---

***Method 3: Boosting – 100 Iterations, no Source*** ***Error rate = .0427***

***Summary of relative influence of each variable:***

| | | |
|---|---|---|
| *1* | *LocGrpID* | *63.044419* |
| *2* | *Title_Fn* | *25.396156* |
| *3* | *RIC* | *8.851960* |
| *4* | *Title_Senior* | *1.982461* |
| *5* | *Size_Employees* | *0.725003* |

---

***Method 4: SVM – Gaussian kernel, no Source*** ***Error rate = .0460***
***# Support Vectors = 1960***

**CONFUSION MATRIX**

**SVM**

| | ***Actual*** | |
|---|---|---|
| ***Predicted*** | ***0*** | ***1*** |
| ***0*** | 4639 | 224 |
| ***1*** | 0 | 10 |

---

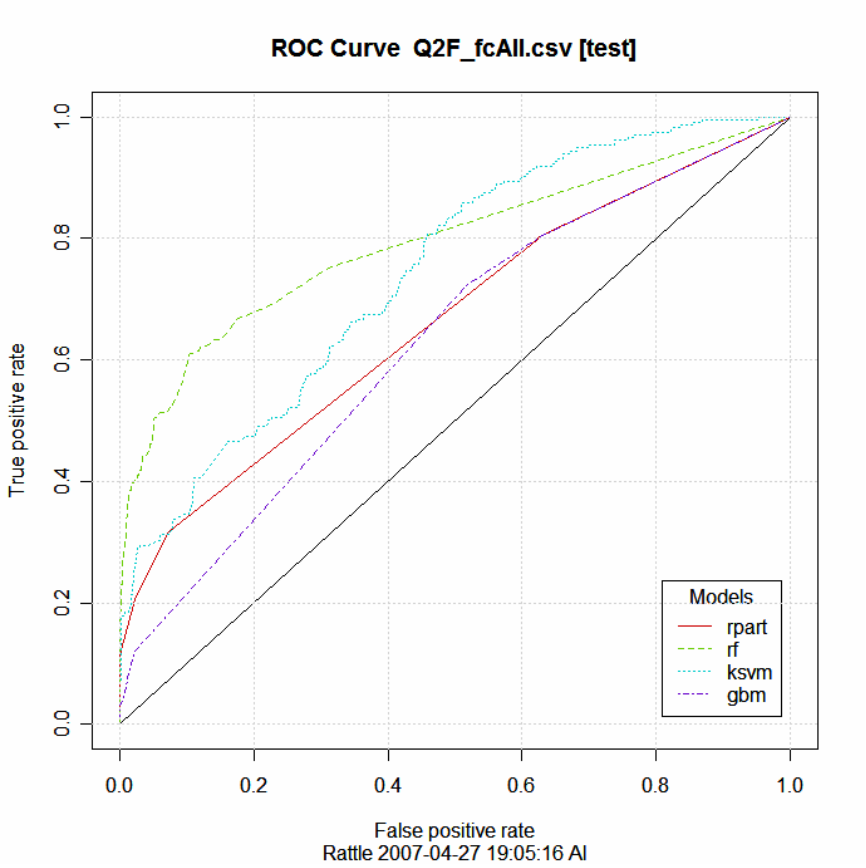
*METHODS COMPARISION*

**READ OR NOT**

| | ***Decision Tree*** | ***Random Forest*** | ***Boosting*** | ***SVM*** |
|---|---|---|---|---|
| **Error rate on test set** | **0.0437** | **0.0408** | **0.0427** | **0.0460** |

**CONFUSION MATRICES**

| | **Decision Tree** | | **Random Forest** | | **SVM** | |
|---|---|---|---|---|---|---|
| | ***Actual*** | | ***Actual*** | | ***Actual*** | |
| ***Predicted*** | ***0*** | ***1*** | ***0*** | ***1*** | ***0*** | ***1*** |
| ***0*** | 4633 | 207 | 4632 | 192 | 4639 | 224 |
| ***1*** | 6 | 27 | 7 | 42 | 0 | 10 |




Discussion:

The ROC curve really tells the story. Boosting does not make a dramatic impact. The SVM method is a somewhat of an improvement over the decision tree, but it is the Random Forest which is by far the best method employed. Its ability to generate a much higher true positive rate without significantly increasing its false positive rate stands out vis-à-vis all other methods. In the confusion matrix comparison, we see this difference in the number of Reads correctly predicted (42 for the random forest, 27 for the SVM, and 10 for the decision tree).

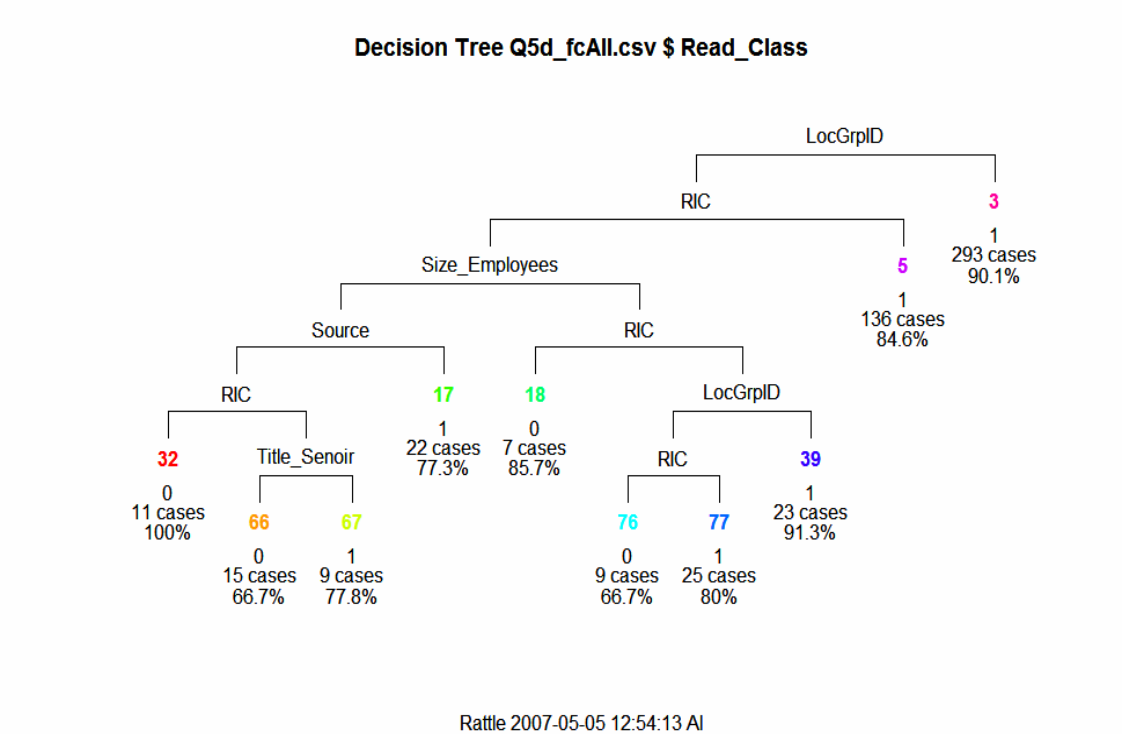
The only drawback to the random forest is that one cannot get a simple set of rules, as we did with the decision tree, for predicting which recipients will read. We simply have to run the random forest and let each of the 500 tree models vote.

---
---

## **Q2. "Read More"** - Given registered readers' attributes, which will be most active?

---

***Method 1a: Decision Tree Induction – no pruning (cp = .0001)*** ***Error rate = .2254***




This decision tree generates six rules (3, 5, 17, 39, 67, and 77) which predict readers who will become active readers, reading more than one story. These rules are:

| ***Rule Number*** | ***3*** | |
|---|---|---|
| **Reader Attribute** | | **Attribute Values that pass** |
| | | |
| Location | | Gulf, Midwest, Northeast, and South regions |

OR

| ***Rule Number*** | ***5*** | |
|---|---|---|
| **Reader Attribute** | | **Attribute Values that pass** |
| Location | | Alaska/Hawaii, Canada, MidAtlantic, West regions |
| Industry | | Aerospace,Banks,Basic Materials,Computer Software,Drugs and Supplies,Engineering,Governmental,Health Services,Mechanical,Medical Equipment,Metal/Mining,Retail,Telco/Cellular |

OR

| ***Rule Number*** | ***17*** | |
|---|---|---|
| **Reader Attribute** | | **Attribute Values that pass** |
| Location | | Alaska/Hawaii, Canada, MidAtlantic, West regions |
| Data Source | | Dunhill, New company website |
| Industry | | Aerospace,Banks,Basic Materials,Computer Software,Drugs and Supplies,Engineering,Governmental,Health Services,Mechanical,Medical Equipment,Metal/Mining,Retail,Telco/Cellular |
| Company Size | | Largest, Small, Very Small |

OR

| ***Rule Number*** | ***39*** | |
|---|---|---|
| **Reader Attribute** | | **Attribute Values that pass** |
| Location | | Canada |
| Industry | | Equipment,High tech,Media related,Oil/Gas,Other Services,Transportation,Wholesalers |
| Company Size | | Small, Small Mid, Mid, Large, Very Large |

OR

| ***Rule Number*** | ***67*** | |
|---|---|---|
| **Reader Attribute** | | **Attribute Values that pass** |
| Location | | Alaska/Hawaii, Canada, MidAtlantic, West regions |
| Data Source | | Default source, ACR-East 2007, or company website data |
| Title (functional) | | Web Development |
| Industry | | High tech,Media related,Other Services,Utilities |
| Company Size | | Very Small, Small, Largest |

OR

| ***Rule Number*** | ***77*** | |
|---|---|---|
| **Reader Attribute** | | **Attribute Values that pass** |
| Location | | MidAtlantic, West regions |
| Data Source | | Dunhill, new company website data |
| Title (function) | | Software Development, Networking or Communications |
| Industry | | Equipment,High tech,Oil/Gas,Other Services,Wholesalers |
| Company Size | | Small_Mid, Mid, Large, Very_Large |

If an reader passes any of these six rules, then they are likely to become an active reader.

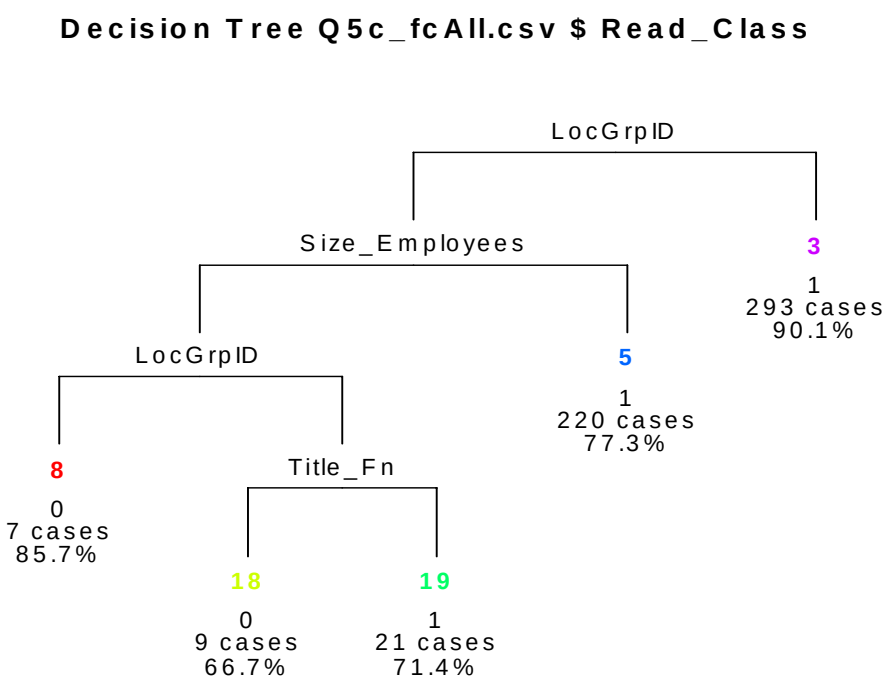
Discussion: Again, this tree is probably overfit to the data due to the low complexity penalty. The small size of the clusters of multi-story readers combined with the relatively large number of levels to the tree support this conclusion.

Moreover, this analysis may be completely missing one very important aspect of the puzzle. People are probably likely to become readers as a function of the passage of time. What would be much more informative, if we had complete time stamp data to work with, would be to change the classification variable such that an active reader would be defined to be one who reads more than one story in a given period of time.

---

***Method 1b: Decision Tree Induction – pruning (cp = .01)*** ***Error rate = .2246***

Discussion: The pruning process via a higher complexity price setting produced a simpler tree which roughly the same error rate.




This decision tree generates three rules (3, 5, and 19) which predict readers who will become active readers, reading more than one story. These rules are:

| ***Rule Number*** | ***3*** | |
|---|---|---|
| **Reader Attribute** | | **Attribute Values that pass** |
| | | |
| Location | | Gulf, Midwest, Northeast, and South regions |

OR

| ***Rule Number*** | ***5*** | |
|---|---|---|
| **Reader Attribute** | | **Attribute Values that pass** |
| Location | | Alaska/Hawaii, Canada, MidAtlantic, West regions |
| Size | | Mid,Small,Small_Mid,Very_Large |

OR

| ***Rule Number*** | ***17*** | |
|---|---|---|
| **Reader Attribute** | | **Attribute Values that pass** |
| Location | | MidAtlantic, West regions |
| Size | | Very_Small, Large, Largest |
| Title (seniority) | | Executive, Worker |

If a reader passes any of these three rules, then they are likely to become an active reader.

Discussion: This tree is a better result given its relatively similar error rate on the test set and lower likelihood of being overfit. The same need for factoring in the time dimension exists here as in method 1a.

---

***Method 2: Random Forest – 500 Trees*** ***Error rate = .1737***

Discussion: This method produced a higher error rate, than the decision tree. However, it was less likely to be overfit to the data. The table below shows, on average, how much each variable was able to assist in identifying unique classifiers.

| | **MeanDecreaseGini** |
|---|---|
| Title_Senoir | 10.77 |
| Title_Fn | 10.91 |
| LocGrpID | 26.33 |
| RIC | 33.62 |
| Size_Employees | 19.84 |
| Source | 10.77 |

---

***Method 3: Boosting – 100 Iterations*** ***Error rate = .2492***

Discussion: The table below shows the relative importance of each variable in the boosting process. However, the error rate is not better than that produced by the random forest method.

**Summary of relative influence of each variable:**

| | | |
|---|---|---|
| 1 | LocGrpID | 86.221277 |
| 2 | RIC | 8.173867 |
| 3 | Size_Employees | 2.992091 |
| 4 | Title_Senoir | 2.612764 |
| 5 | Title_Fn | 0.000000 |
| 6 | Source | 0.000000 |

---

***Method 4: SVM – Gaussian kernel***

***Error rate = .2330***
***# Support Vectors = 259***

**CONFUSION MATRIX**

**SVM**

| | ***Actual*** | |
|---|---|---|
| ***Predicted*** | ***0*** | ***1*** |
| ***0*** | 0 | 0 |
| ***1*** | 55 | 181 |

Discussion: The SVM seemed to fail demonstrably by simply classifying all readers as active readers. Given that the original data set is roughly 80/20 active reader to one time reader, the error rate shows up as close to 20% from this approach.

---

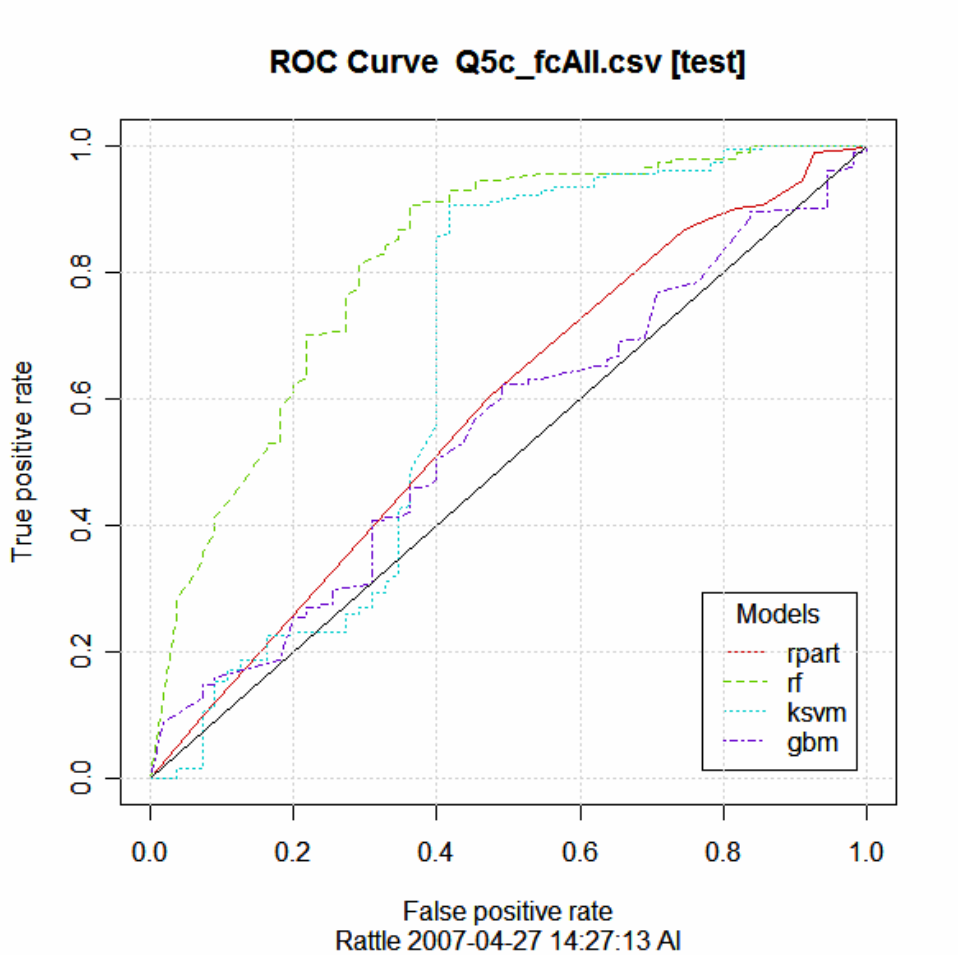
*METHODS COMPARISION*

**READ MORE**

| | ***Decision Tree*** | ***Random Forest*** | ***Boosting*** | ***SVM*** |
|---|---|---|---|---|
| **Error rate on test set** | **0.2246** | **0.1737** | **0.2492** | **0.2330** |

**CONFUSION MATRICES**

| | Decision Tree | | Random Forest | | SVM | |
|---|---|---|---|---|---|---|
| | *Actual* | | *Actual* | | *Actual* | |
| ***Predicted*** | ***0*** | ***1*** | ***0*** | ***1*** | ***0*** | ***1*** |
| ***0*** | 4 | 2 | 16 | 2 | 0 | 0 |
| ***1*** | 51 | 179 | 39 | 179 | 55 | 181 |




Discussion:

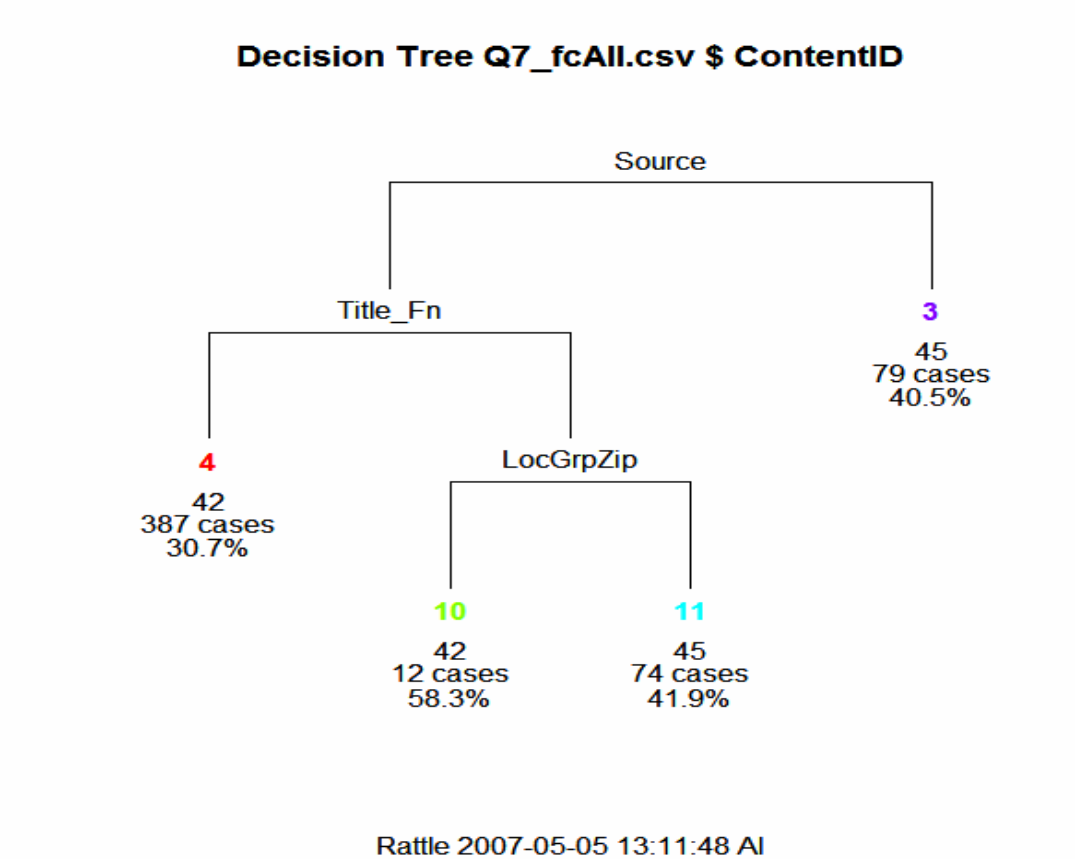
Once again, the ROC curve identifies the Random Forest as the best approach. Boosting does not make a dramatic impact. Once you allow for a higher false positive rate, the SVM improves. But our earlier analysis showed that this was because the SVM has defaulted to a really simple rule of classifying everything as an active reader. Therefore, the SVM is the worse method given the shape of this data sample. It is worth noting that the simpler Decision Tree was not that bad in terms of increased error for the insight it gives us in terms of some simple rules of thumb that we can focus on.

---
---

**Q3. "Read What"** - Given registered readers' attributes, which stories will they be interested in?

---

***Method 1a: Decision Tree Induction – no pruning (cp = .01)*** ***Error rate = .6691%***




Discussion: This decision tree generates four rules which predict what type of content a reader will be interested in. However, this tree is pretty useless.

It has a high error rate and is being dominated by the fact that roughly 60% of all of the content items read are in either category 42 ("Software| Business") or 45 ("Software|Operating Systems).

A reduction of the complexity penalty variable generates a tree that is too complex to display (76 rules). The error rate on this model is still very high at 61%. The results, while still dominated by classes 42 and 45 do expand out to classify the five next most frequently read classes (2,6, 12,16, and 46).The following table gives the name of these classes:

| | | |
|---|---|---|
| **15 \|Industries\|Hacking** | **24 \|Online\|Email** | **37 \|Services\|Security** |
| **16 \|Industries\|IT Management** | **25 \|Online\|IM** | **42 \|Software\|Business** |
| **17 \|Industries\|Legal** | **26 \|Online\|News** | **43 \|Software\|Consumer** |
| **18 \|Industries\|News** | **27 \|Online\|Portal** | **44 \|Software\|Networking** |
| **20 \|Industries\|PCs** | **30 \|Online\|Search** | **45 \|Software\|Operating Systems** |
| **21 \|Industries\|Standards** | **33 \|Online\|Software as a Service** | **46 \|Software\|Software Development** |

After looking at these results, I discussed the content taxonomy further with TechPub. I learned that classes 42 and 45 did tend to become “catch-all” categories to some extent.

Therefore, with these issues in mind combined with the lack of attributes to work with, it is doubtful that we can dramatically reduce the classification error for this question. That having been said, I did try more advanced techniques to improve the result.

---

***Method 2: Random Forest – 500 Trees*** ***Error rate = .5164***

Discussion: This method produced a lower error rate than the decision tree. The tables below show, on average, how much each variable was able to assist in identifying unique classifiers, as well as the how well the model predicted each content category.

**MeanDecreaseGini**

| | |
|---|---|
| Title_Senoir | 104.83 |
| Title_Fn | 98.39 |
| LocGrpID | 250.19 |
| RIC | 302.38 |
| Size_Combined | 147.58 |
| Source | 113.90 |

Confusion Matrix

| | **0** | **1** | **2** | **5** | **6** | **9** | **10** | **12** | **13** | **16** | **17** | **18** | **20** | **24** | **25** | **27** | **30** | **42** | **44** | **45** | **46** | **class.error** | **num** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ***0%*** | ***2*** |
| **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | ***100%*** | ***2*** |
| **2** | 1 | 0 | 10 | 0 | 2 | 0 | 0 | 1 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 1 | 0 | 5 | 0 | 8 | 0 | ***68%*** | ***31*** |
| **5** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 4 | 0 | ***100%*** | ***8*** |
| **6** | 0 | 0 | 2 | 0 | 12 | 2 | 0 | 4 | 0 | 1 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 13 | 0 | 15 | 0 | ***76%*** | ***51*** |
| **9** | 0 | 0 | 0 | 0 | 1 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ***20%*** | ***5*** |
| **10** | 0 | 0 | 0 | 0 | 1 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | ***60%*** | ***5*** |
| **12** | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 40 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 12 | 0 | 11 | 1 | ***42%*** | ***69*** |
| **13** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 4 | 0 | ***100%*** | ***11*** |
| **16** | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 2 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 5 | 2 | ***79%*** | ***14*** |
| **17** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | ***25%*** | ***4*** |
| **18** | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 2 | 0 | 0 | 1 | 1 | 1 | 10 | 0 | 5 | 2 | ***93%*** | ***27*** |
| **20** | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | ***100%*** | ***3*** |
| **24** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ***0%*** | ***0*** |
| **25** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ***0%*** | ***0*** |
| **27** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 3 | 0 | 1 | 1 | 1 | 0 | ***63%*** | ***8*** |
| **30** | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 1 | 0 | ***100%*** | ***6*** |
| **42** | 0 | 0 | 2 | 0 | 8 | 1 | 0 | 9 | 0 | 1 | 0 | 3 | 0 | 0 | 0 | 1 | 0 | 92 | 0 | 25 | 3 | ***37%*** | ***145*** |
| **44** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 2 | ***100%*** | ***4*** |
| **45** | 0 | 0 | 2 | 1 | 3 | 0 | 0 | 6 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 24 | 0 | 79 | 3 | ***34%*** | ***120*** |
| **46** | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 2 | 12 | 14 | ***60%*** | ***35*** |

---

***Method 3: Boosting – 100 Iterations*** ***Error rate = .6121***

Discussion: The table below shows the relative importance of each variable in the boosting process. However, the error rate is not better than that produced by the random forest method.

**Summary of relative influence of each variable:**

| | | |
|---|---|---|
| 1 | LocGrpID | 39.958 |
| 2 | RIC | 16.689 |
| 3 | Size_Combined | 6.288 |
| 4 | Title_Senoir | 18.966 |
| 5 | Title_Fn | 11.628 |
| 6 | Source | 9.468 |

---

***Method 4: SVM – Gaussian kernel*** ***Error rate = .5700***
***# Support Vectors = 468***

| *% Predictions Were Accurate* | True | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Pred** | **0** | **1** | **2** | **5** | **6** | **7** | **9** | **10** | **12** | **13** | **16** | **17** | **18** | **20** | **24** | **25** | **27** | **30** | **33** | **42** | **43** | **44** | **45** | **46** |
| *67%* | **2** | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| *60%* | **6** | 0 | 0 | 0 | 0 | 15 | 0 | 0 | 1 | 2 | 1 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| *40%* | **12** | 0 | 0 | 3 | 0 | 9 | 0 | 0 | 1 | 33 | 1 | 0 | 1 | 5 | 0 | 0 | 0 | 1 | 2 | 0 | 12 | 1 | 3 | 7 | 4 |
| *83%* | **16** | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| *45%* | **42** | 3 | 0 | 21 | 5 | 29 | 0 | 2 | 1 | 34 | 3 | 5 | 1 | 17 | 1 | 0 | 0 | 5 | 4 | 1 | 151 | 0 | 1 | 44 | 9 |
| *39%* | **45** | 0 | 2 | 19 | 6 | 20 | 3 | 3 | 4 | 18 | 10 | 10 | 0 | 16 | 2 | 2 | 1 | 2 | 5 | 0 | 42 | 1 | 3 | 126 | 28 |
| *67%* | **46** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 6 |
| *% In Class Pred* ----------> | | **0%** | **0%** | **4%** | **0%** | **20%** | **0%** | **0%** | **0%** | **37%** | **0%** | **25%** | **0%** | **0%** | **0%** | **0%** | **0%** | **0%** | **0%** | **0%** | **73%** | **0%** | **0%** | **71%** | **12%** |

Discussion: The SVM seemed to capture roughly half of the improvement in the error rate that the random forest model did. Like the decision tree method, it classified into the seven most common content classes.

---

*METHODS COMPARISION*

| READ WHAT | | | | |
|---|---|---|---|---|
| | ***Decision Tree*** | ***Random Forest*** | ***Boosting*** | ***SVM*** |
| **Error rate on test set** | **0.6691** | **0.5164** | **0.6121** | **0.5700** |

Discussion:

Unlike in the previous two questions where we had binary classes, it was not easy to generate a ROC curve for these results. So, the best way to compare these methods is through their respective error rates on the test set and through the confusion matrices they generate. The Random Forest method generates the lowest area and also has a better distribution of results across the confusion matrix, in that it classifies into all 31 active categories in the taxonomy.

## Conclusions and Suggested Next Steps

The data sample that we were provided by TechPub was sufficient to allow us to get an initial insight into the answers to the three questions of interest.

During the data cleaning stage, we found that a large amount of our data records were disqualified by virtue of having one or more missing feature values. Being able to enrich these records in the future would potentially provide a great deal more data to rerun this analysis with.

Alternatively, since we found that some variables were less valuable than others, we could go back and relax this complete feature set restriction and rerun the analysis once again. The downside of this latter approach however, is that we will have precluded ourselves somewhat from getting an accurate sense of how valuable the excluded feature data could be on a broader sample.

At the outset of the project, we decided to employ four different data mining techniques: decision tree induction, random forests, a boosting algorithm, and a support vector machine.

In all three cases, the random forest proved to be the most effective technique for this data set. This approach resulted in the lowest error rates when applied to the test data sample, as well as the best ratio of true positives to false positives in the first two cases.

The **“Read or Not”** question presents a challenge because of the 19:1 dominance of email recipients who do not become readers over those who do. The random forest method did the best job of dealing with this low class representation problem. The confusion matrix showed that this ensemble method was able to make a significant number of classifications in all four states. The simple decision tree induction method gave us an set of rules which were not bad as simple heuristics go either.

We were able to come up with a reasonable ability to classify instances of the test set in the **“Read More”** question (the random forest method produced a 17% error rate). However, when

one steps back and thinks beyond the numbers, it becomes clear that we are missing one critical aspect necessary to properly attack this question: the dimension of time. As we noted above, by adding the notion of how long it takes readers to go from reading one story to another, we can get a much better definition of an "active" reader. Armed with this new classifier definition, this work should be repeated when a more complete dataset of timestamp information is available.

The analysis for all three questions would greatly benefit from a greater number of records with complete feature data as well as more feature groups, in general, to work with. This problem was felt most acutely in the **"Read What"** analysis. With forty seven content classes, we simply need more features and more records to be able to really be able to predict which readers will read what. The random forest method was able to get us to the point of being able to make an even money prediction as to which content group a reader would select. While a promising result, there is much room for further improvement on this question.

Moreover, the results of our analysis in this area lead to the recommendation that TechPub continue to actively refine its content taxonomy. I have to specific suggestions on this matter. First, the taxonomy has been built from the perspective of how people who write about technology think about the grouping of subject matter. However, increased domain insight into how people who read this content think about the grouping of subject matter would be very valuable. Secondly, the employment of text mining techniques to build a taxonomy of how stories are actually similar to one another could also prove to be quite enlightening.

Finally, I think this "Read What" question should be tackled with a more sophisticated approach overall. Specifically, we need to move beyond simple semantic grouping of stories and into text mining techniques which model the exact word groupings that would be readers see in the headline or abstract of the story that they peruse in deciding whether to read or not. This is where they are giving us the first clue as to their real interest. Secondly, by again pairing readership information with timestamp data, we could text mine the actual stories and pair that info with how long the reader spends in that story as a much better gauge of interest. These techniques should provide meaningful insight. Unfortunately, they were simply beyond the scope of this project.

**References**

1. Margaret Dunham, Data Mining Introductory and Advanced Topics, ISBN: 0130888923, Prentice Hall, 2003.

2. Jiawei Han and Micheline Kamber (2005), Data Mining Concepts and Techniques. Morgan Kaufmann. 2nd Ed. (URL: http://www-faculty.cs.uiuc.edu/~hanj/bk2/) Chapters 4-5.

3. Pang-Ning Tan, Michael Steinbach, and Vipin Kumar (2005). Introduction to Data Mining. Addison Wesley. Chapters 4-6.