SPAM Detection – Data Exploration

Unstructured Data

Prepared by Ken Venturi 6/3/2023

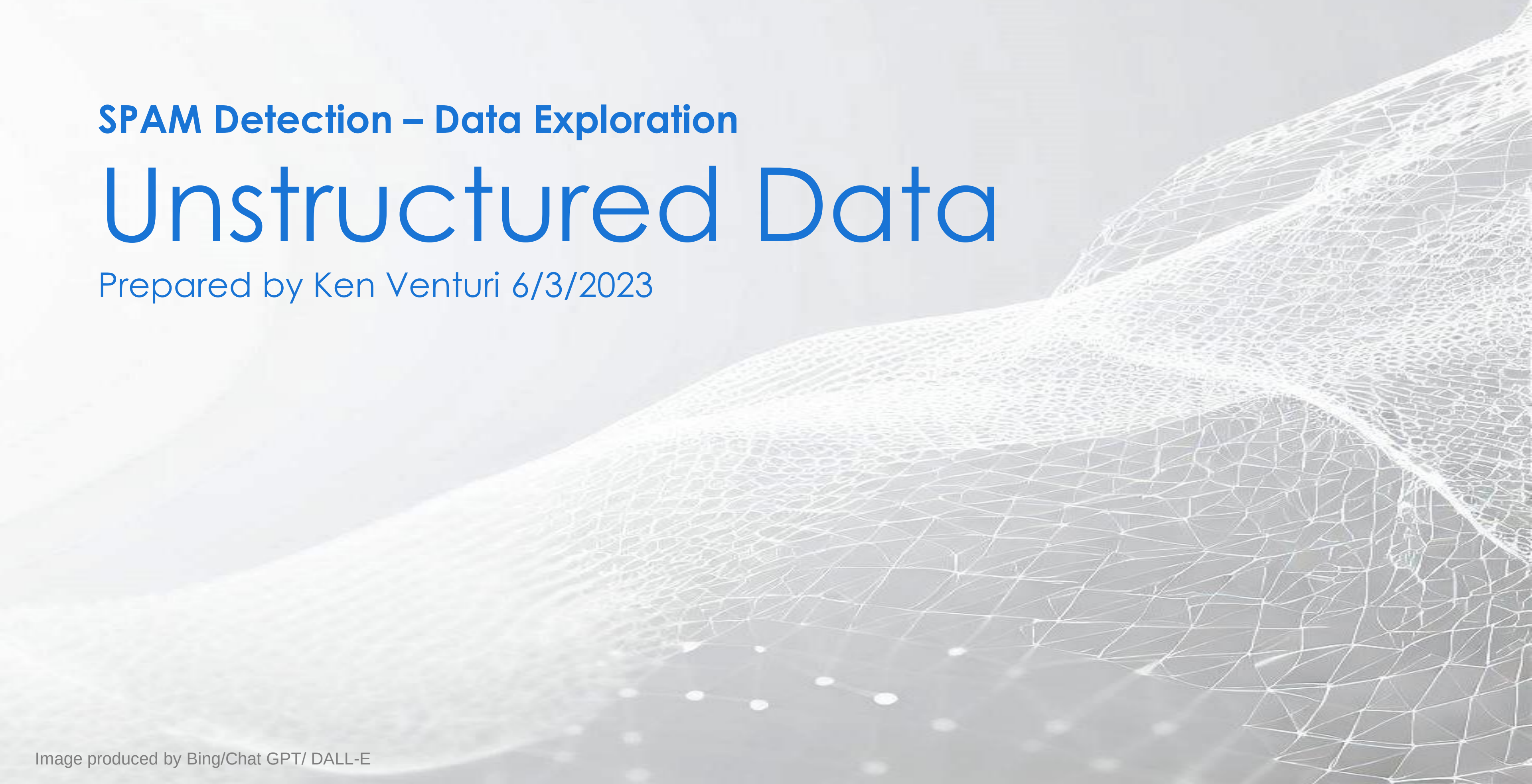


Image produced by Bing/Chat GPT/ DALL-E

Contents / Agenda

- Executive Summary
- Business Problem Overview and Solution Approach
- Exploratory Data Analysis Results
- Data Preprocessing
- Model Performance Summary
- Appendix

Executive Summary

Cyber Solutions is in need of a Machine Learning based classifier that will allow its emails systems to detect and quarantine SPAM emails that might be leverged to initiate a cyber attack. After much modeling and callibration of the Machine Learning models, the chosen model that best perfomed in all categories and removed the least of what could be important actual emails (HAM) was the ***Random Forest without Pruning.***

- The RF Model delivered a **98.15 F1 Score** and an **accuracy of 96.77%**
- Importantly, the **Class Recall and Class Precision** were the best of the group of models
- Employing the RF model does consume significant computational resources. As such, if the appropriate level of computing resources are not available to the company, then another model may be more practical in daily application. In fact, the Decision Tree Model performed almost as well and lost only 1 HAM email errantly classified as SPAM. The Decision Tree model produced results in 14% of the computational time of the Random Forest model. Still, even that one lost email may be important enough to employ the additional resources depending on the nature of the company's business and the roles of its email users.
- After deeper analysis of the email messages, it appears the data itself is a highly random collection of scripts not necessarily representative of actual email message content and so actual email analysis should provide at least as good of results as the model did on the highly irregular text.
- It is also recommended that a model that incorporates some level of insight into the links or external references within the message content itself could be both important and helpful in creating distinctions between true HAM and SPAM examples. This would require additional work on a new set of data that includes links in the text.
- Practically, it would be recommended to install the RF model and trigger its use based on a scripted trigger within the email server systems themselves intercepting and classifying the emails in real time of receipt and prior to distribution to the final enduser. Essentially creating an "inline workflow" email SPAM detection system.

Business Problem Overview and Solution Approach

Context

Short Message Services (SMS) is far more than just a technology for chat. SMS technology evolved out of the global system for internationally accepted mobile communications standards. But with the introduction of every technological advancement, we see the rise of many unnecessary evils that affect our usage of technology and how we interact with it.

Spam is the abuse of electronic messaging systems to send unsolicited messages in bulk indiscriminately. **SMS spam is used for commercial advertising and spreading phishing links**. While the former reason doesn't have any harmful implications per se, the latter can be dangerous if unsuspecting users were to fall for it. There have been many incidents of people having their bank accounts hacked or depleted just because they fell victim to phishing links.

'Spam' texts have been a major contributor in cyber crimes all over the world and with time, phishing techniques have only gotten stronger. This has led to widespread research and applications on spam classifiers using Natural Language Processing.

Objective

Cyber Solutions is a company that provides security measures against cyber attacks on businesses. You are a Data Science Manager at Cyber Solutions, and you have been assigned a new project to prevent cyber attacks on an organization through SMS messages to employees' phones. You have a collection of labeled SMS texts - 'spam' and 'ham' (not Spam) are the two labels. The goal is to extract meaningful insights from the data and build a classification model to predict whether an SMS is 'spam' or not, using Machine Learning algorithms on the preprocessed SMS text data.

Solution Approach and Employed Methods

EDA revealed that significant NLP text processing is required to create a dataset that can be employed to extract meaningful correlations with its categorizing of SPAM or HAM email messages. To do this, we will employ a variety of operators to "skin out" unwanted characters, words, symbols, spaces or other text entries that may create distortion in our predictive results.

Operators include (RapidMiner):

- **Replace** – a method employed to remove special symbols, spaces, and unwanted characters
- **Stop Word Filtering** – to remove words deemed not to have a significant impact on meaning
- **Stemming** and **Tokenizing** – used to establish a word level data realm and identify the core (stem) meaning of the token or word regardless of tense, plurality, or possession.
- Finally we will employ a **Decision Tree** model and its peer **Random Forest** model with limited depths, tree numbers, various pruning methods and the use of Gini_index as its criterion. We will compare these models testing for accuracy and paying special attention to reducing the number of HAM emails errantly predicted as SPAM in an effort not to filter out what could be important HAM messages.

Exploratory Data Analysis Results

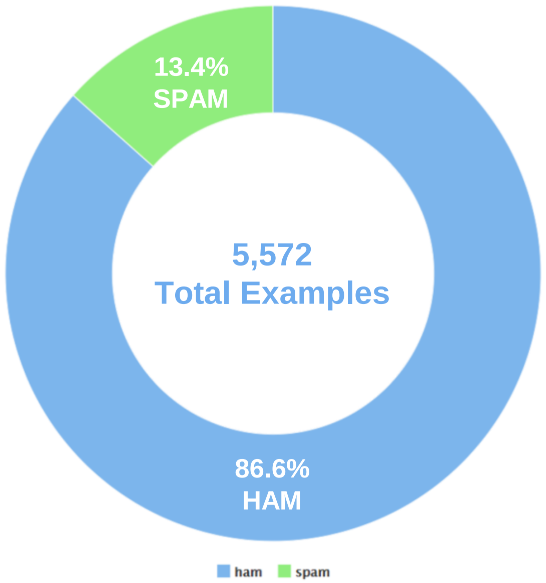
Data Dictionary

Category: Contains the labels 'spam' or 'ham' for the corresponding text data
Message: Contains the SMS text data

3 attributes including:

- Row Number (ID)
- Category (Spam or Ham, Binominal),
- Message (open text, Nominal)

- The Data does not have missing attributes values.
- Within the provided dataset 4,825 (86.6%) examples were categorized as HAM and 747 (13.4%) examples as SPAM
- While the Row No. and Category attributes have clean data, the Message attribute has open text entries from users resulting in highly inconsistent unstructured data with various uses of symbols and acronyms.
- Significant Text Processing will be required on the Message attribute values to develop a predictor.



Link to Appendix slide on data background check

Model Performance Summary

	Highly Important		Highly Important		Most Important				
Model Description	F1 Score	Training Set Accuracy	Test Set Accuracy		Test Set Recall (FP-FN)	Test Set Weighted Recall	Test Set Precision (TP-TN)	Test Set Weighted Precision	Test Set Classification Error
Decision Tree	97.90%	98.98%	96.32%		98.86%	89.36%	96.95%	96.95%	3.68%
Decision Tree Pruned	97.84%	98.92%	96.23		98.76%	89.31%	96.95%	93.89%	3.46%
Random Forest	98.15%	99.19%	96.77%		98.96%	90.76%	97.35%	94.92%	3.23%
Random Forest Pruned	98.05%	96.59%	96.59%		98.96%	90.09%	97.15%	94.76%	3.41%

Random Forest (no pruning) produced the best results in all categories. While the variances between the employed models were quite small, even the smallest variance could result in quality, important emails being classified as SPAM and therefore being hidden from the company. That risk far outweighs the risk of letting too many SPAMs get through. For that reason, we placed additional emphasis on Test Set Recall, F1 Score, and overall accuracy.

The Random Forest Model employed the following key parameters:

- Limited number of trees to 100
- Employed the Gini_Index criterion
- Maximum Depth set to 35
- Subset Ratio of 0.2 with a confidence voting method
- Text Vectorization was based on TF-IDF
- A maximum number of columns set to 1,000

Link to Appendix slide on model assumptions

APPENDIX

Data Preparation (Decision Tree and Random Forest)

As revealed in the EDA phase of the project, significant processing is required on the values in the Message attribute to create a dataset that can be helpful in determining the nature of the SMS examples and whether or not a given SMS example is categorized as HAM or SPAM. In both cases of the Decision Tree and the Random Forest models the same text processing techniques were employed.

1. **Replace** non-text, non-numeric, and non-space values with “nothing” affectively removing these values. We employed the Regular Expression [^A-Za-z0-9\s]+ to identify these values.
2. **Replace** multiple spaces with a single space for later processes and again leveraged Regular Expression [s]+
3. Converting **Nominal to Text** for later text specific operations
4. **Tokenize** was employed to break the text into tokens very near a word equivalent using Regular Exp \s+ as its separator expression
5. **Transform Cases** – remove letter case as a differentiator converting all values to lower case
6. **Filter Stopwords** – remove words that typically carry very little to no real information or meaning in our analysis
7. **Fitter Tokens (by Length)** – used to eliminate tokens that are unusually long or simply too short to matter, we set min and max to 3 and 20 respectively
8. **Stem (porter)** – leveraged stemming to get to a more singular value for each token as a root word meaning no matter the words tense, plurality or possession.
9. Finally, **Text Vectorization** was employed to create vectors and it was set to a limit of 1,000 columns (unique tokens)
10. A **Process Documents from Data** was employed as a container for much of the NLP operators. It was also employed to cate a word vector and like Text Vectorization was set to TF-IDF as its method for vector creation.
11. **TF-IDF** is referencing the Term Frequency – Integrated Differential Frequency creating a score for both the frequency of the term within the document and relative frequency across all documents to offer the best model for relevance of the particular token.

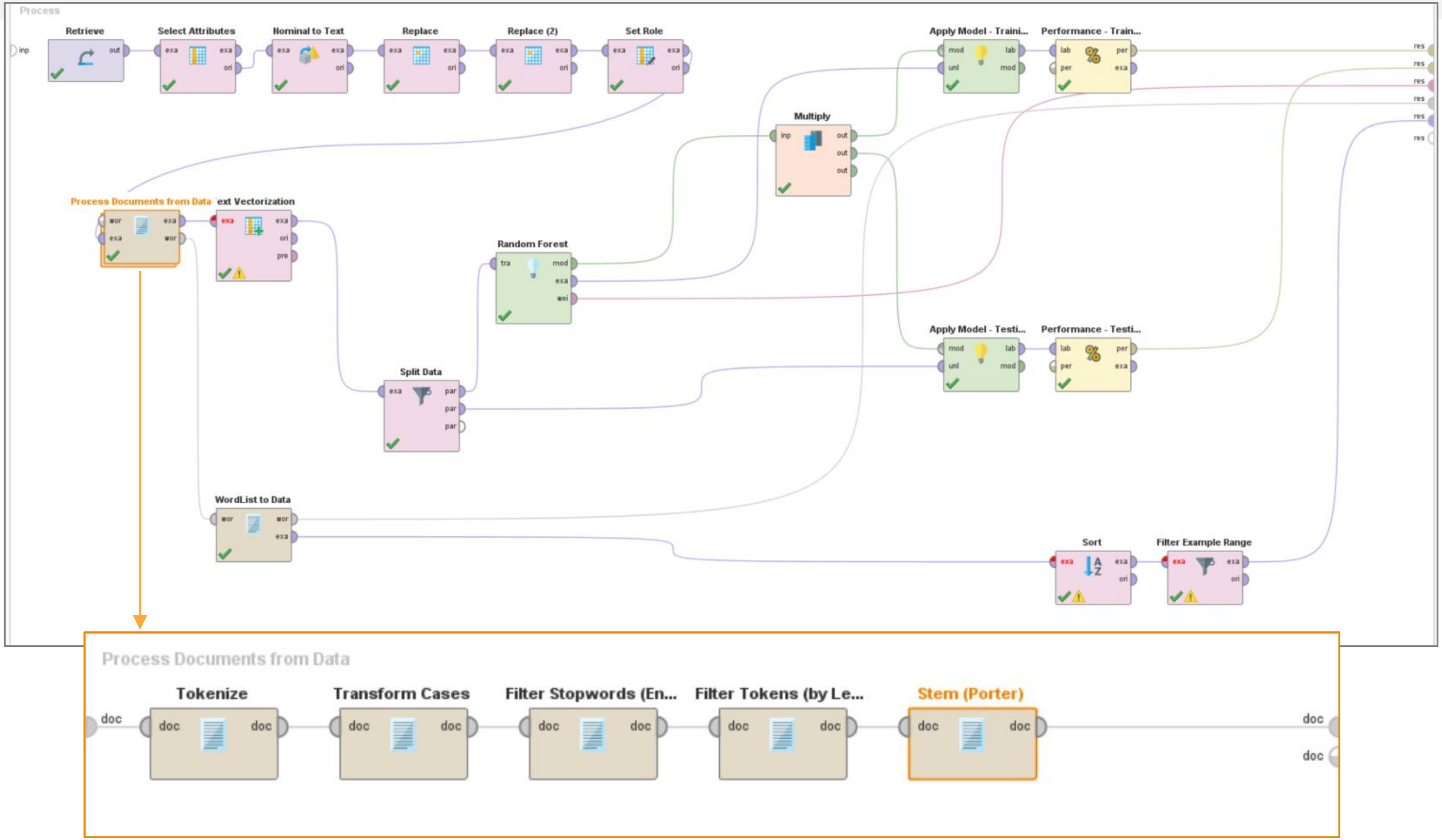
Model Building

Two models were employed to create the final recommendation system.

- **Decision Tree** model of high processed text values with hyperparameters set to gini_index and a maximum tree depth of 25. Our best results did not employ pruning or prepruning.
- **Random Forest** model of highly processed text values with hyperparameters to include gini_index with a maximum depth of 35. Our best results did not employ pruning and had subset ratio of 0.2, based on a confidence voting strategy.

Random Forest model is the model of choosing. While it required almost 600% more resources than the decision tree, it had a modest increase in accuracy, precision and recall and saved the user 1 true HAM email from being classified as SPAM. In this particular use case, even one email can be of great importance to the company if lost to a SPAM categorization.

Random Forest Model



Performance

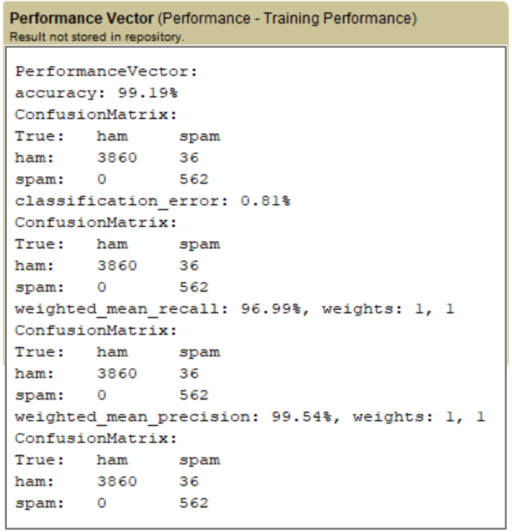
Random Forest Final Result

TRAINING

99.19%

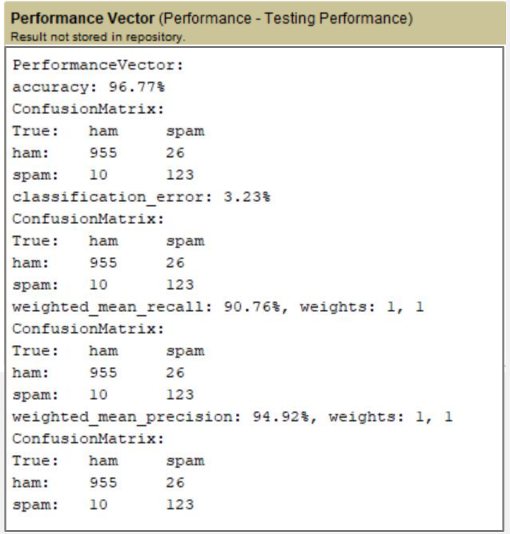
TEST

96.77%



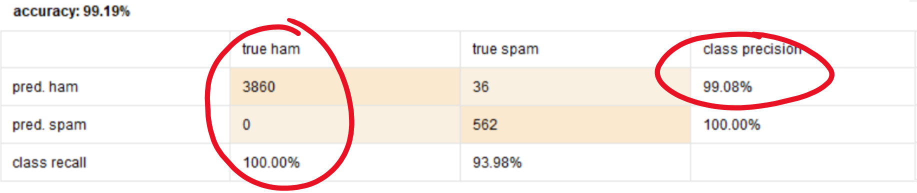
Performance Vector (Performance - Training Performance)
Result not stored in repository.

```
PerformanceVector:
accuracy: 99.19%
ConfusionMatrix:
True:   ham     spam
ham:    3860    36
spam:   0       562
classification_error: 0.81%
ConfusionMatrix:
True:   ham     spam
ham:    3860    36
spam:   0       562
weighted_mean_recall: 96.99%, weights: 1, 1
ConfusionMatrix:
True:   ham     spam
ham:    3860    36
spam:   0       562
weighted_mean_precision: 99.54%, weights: 1, 1
ConfusionMatrix:
True:   ham     spam
ham:    3860    36
spam:   0       562
```



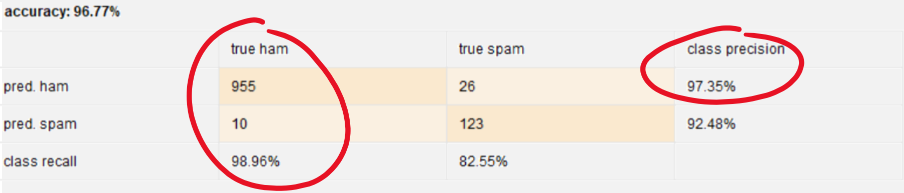
Performance Vector (Performance - Testing Performance)
Result not stored in repository.

```
PerformanceVector:
accuracy: 96.77%
ConfusionMatrix:
True:   ham     spam
ham:    955     26
spam:   10      123
classification_error: 3.23%
ConfusionMatrix:
True:   ham     spam
ham:    955     26
spam:   10      123
weighted_mean_recall: 90.76%, weights: 1, 1
ConfusionMatrix:
True:   ham     spam
ham:    955     26
spam:   10      123
weighted_mean_precision: 94.92%, weights: 1, 1
ConfusionMatrix:
True:   ham     spam
ham:    955     26
spam:   10      123
```



accuracy: 99.19%

	true ham	true spam	class precision
pred. ham	3860	36	99.08%
pred. spam	0	562	100.00%
class recall	100.00%	93.98%	

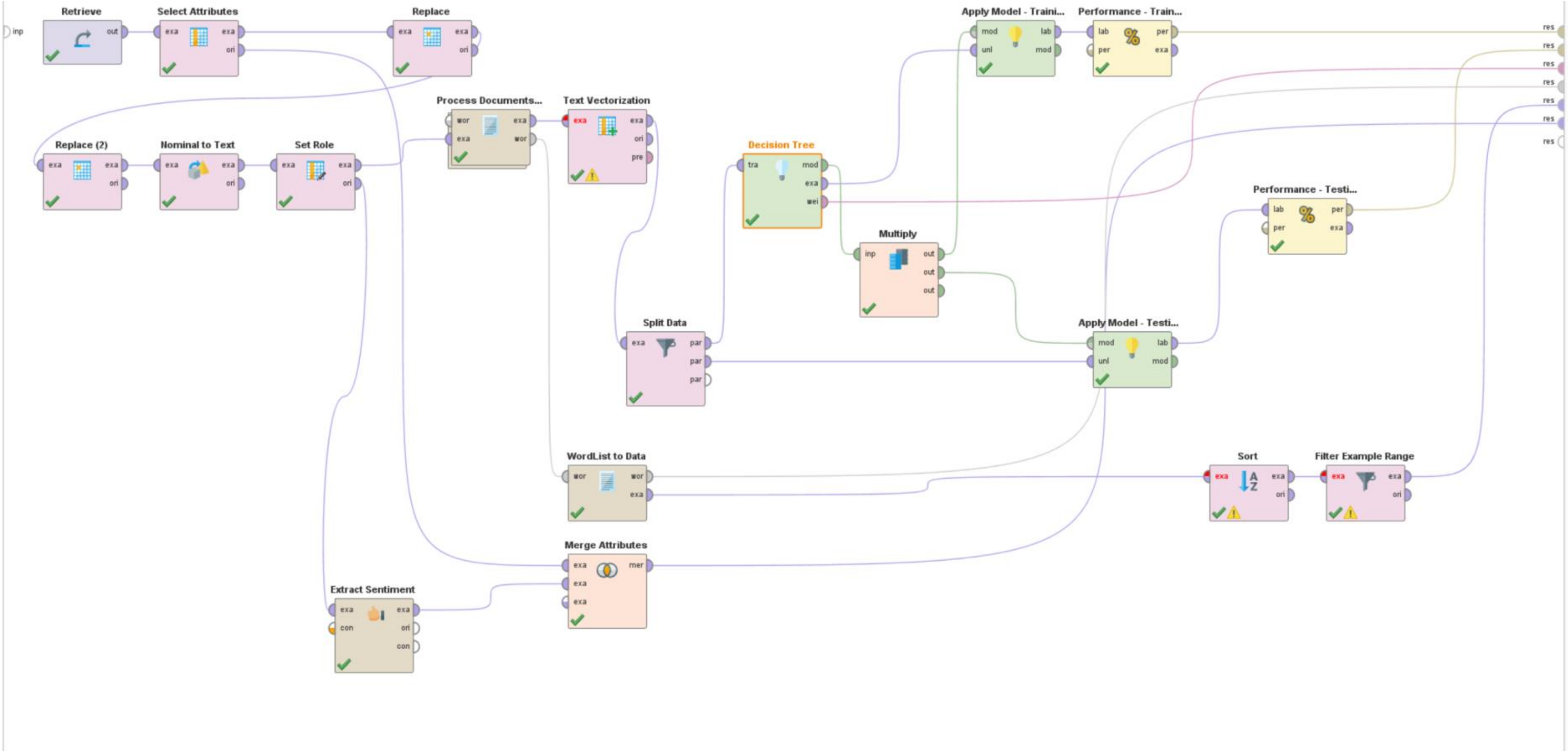


accuracy: 96.77%

	true ham	true spam	class precision
pred. ham	955	26	97.35%
pred. spam	10	123	92.48%
class recall	98.96%	82.55%	

Paying special attention to TRUE HAM predicted as SPAM. The idea here is to choose a model that will provide a high percentage of accurate detection on this class of values. For every message that was True HAM errantly predicted as SPAM a possible important message that should make it through might get filtered. This is a larger risk and cost to the company than a True SPAM errantly being classified as HAM and therefor a message with no importance slips through the filter only to be viewed and discarded later. The resulting TEST PERFORMANCE VECTOR illustrated above had a 99.19% recall on this class and a 96.77% precision on predicted ham. This provides a high level of confidence for this consideration.

Decision Tree Model

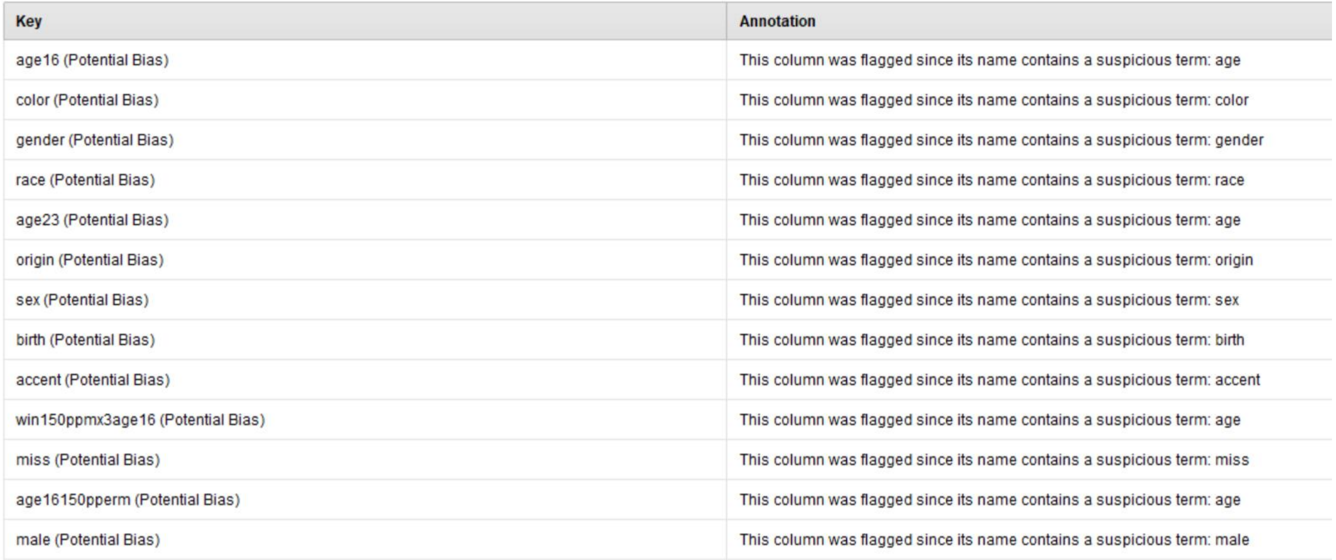


Decision Tree Model

```
Tree
call > 0.066
| call > 0.117
| | call > 0.226
| | | 150ppm > 0.096: spam {ham=0, spam=3}
| | | 150ppm ≤ 0.096
| | | | text_0 > 0.064
| | | | | call > 0.263: ham {ham=1, spam=0}
| | | | | call ≤ 0.263: spam {ham=0, spam=2}
| | | | text_0 ≤ 0.064
| | | | | 02070836089 > 0.445: spam {ham=0, spam=1}
| | | | | 02070836089 ≤ 0.445
| | | | | | 09058094583 > 0.199: spam {ham=0, spam=1}
| | | | | | 09058094583 ≤ 0.199: ham {ham=119, spam=0}
| | call ≤ 0.226
| | | ervic > 0.096: spam {ham=0, spam=9}
| | | ervic ≤ 0.096
| | | | min > 0.072: spam {ham=0, spam=9}
| | | | min ≤ 0.072
| | | | | plea > 0.296
| | | | | | father > 0.301: ham {ham=1, spam=0}
| | | | | | father ≤ 0.301: spam {ham=0, spam=8}
| | | | | plea ≤ 0.296
| | | | | | claim > 0.080: spam {ham=0, spam=6}
| | | | | | claim ≤ 0.080
| | | | | | | 08712460324 > 0.119: spam {ham=0, spam=5}
| | | | | | | 08712460324 ≤ 0.119
| | | | | | | | free > 0.054
| | | | | | | | | call > 0.162: spam {ham=0, spam=5}
| | | | | | | | | call ≤ 0.162: ham {ham=2, spam=0}
| | | | | | | | free ≤ 0.054
| | | | | | | | | collect > 0.127: spam {ham=0, spam=3}
| | | | | | | | | collect ≤ 0.127
| | | | | | | | | | 0844 > 0.144: spam {ham=0, spam=2}
| | | | | | | | | | 0844 ≤ 0.144
| | | | | | | | | | | contact > 0.120: spam {ham=0, spam=2}
| | | | | | | | | | | contact ≤ 0.120
| | | | | | | | | | | | 07090201529 > 0.386: spam {ham=0, spam=1}
| | | | | | | | | | | | 07090201529 ≤ 0.386
| | | | | | | | | | | | | 08704439680t > 0.241: spam {ham=0, spam=1}
| | | | | | | | | | | | | 08704439680t ≤ 0.241
| | | | | | | | | | | | | | 09058097218 > 0.241: spam {ham=0, spam=1}
| | | | | | | | | | | | | | 09058097218 ≤ 0.241
| | | | | | | | | | | | | | | 09096102316 > 0.204: spam {ham=0,
spam=1}
| | | | | | | | | | | | | | | 09096102316 ≤ 0.204: ham {ham=71,
spam=0}
| call ≤ 0.117
| | ltgt > 0.055: ham {ham=6, spam=0}
| | ltgt ≤ 0.055
| | | give > 0.185: ham {ham=4, spam=0}
| | | give ≤ 0.185
| | | | darlin > 0.148: ham {ham=2, spam=0}
| | | | darlin ≤ 0.148
| | | | | dont > 0.148: ham {ham=2, spam=0}
| | | | | dont ≤ 0.148
| | | | | | hei > 0.069: ham {ham=2, spam=0}
| | | | | | hei ≤ 0.069
| | | | | | | lor > 0.087: ham {ham=2, spam=0}
| | | | | | | lor ≤ 0.087
| | | | | | | | tell > 0.069: ham {ham=2, spam=0}
| | | | | | | | tell ≤ 0.069
| | | | | | | | | 2mrw > 0.164: ham {ham=1, spam=0}
| | | | | | | | | 2mrw ≤ 0.164
| | | | | | | | | | abnorm > 0.218: ham {ham=1, spam=0}
| | | | | | | | | | abnorm ≤ 0.218
| | | | | | | | | | | admiti > 0.142: ham {ham=1, spam=0}
| | | | | | | | | | | admiti ≤ 0.142
| | | | | | | | | | | | adr > 0.219: ham {ham=1, spam=0}
| | | | | | | | | | | | adr ≤ 0.219
| | | | | | | | | | | | | afternoon > 0.123: ham {ham=1, spam=0}
| | | | | | | | | | | | | afternoon ≤ 0.123
| | | | | | | | | | | | | | alon > 0.190: ham {ham=1, spam=0}
| | | | | | | | | | | | | | alon ≤ 0.190
| | | | | | | | | | | | | | | becau > 0.238: ham {ham=1, spam=0}
| | | | | | | | | | | | | | | becau ≤ 0.238: spam {ham=0, spam=202}
call ≤ 0.066
| txt > 0.052
| | txt > 0.201
| | | free > 0.059: spam {ham=0, spam=3}
| | | free ≤ 0.059
| | | | get > 0.042
| | | | | get > 0.142: ham {ham=2, spam=0}
| | | | | get ≤ 0.142: spam {ham=0, spam=3}
| | | | get ≤ 0.042: ham {ham=5, spam=0}
| | txt ≤ 0.201
| | | bak > 0.156: ham {ham=1, spam=0}
| | | bak ≤ 0.156
| | | | beverag > 0.180: ham {ham=1, spam=0}
| | | | beverag ≤ 0.180: spam {ham=0, spam=104}
| txt ≤ 0.052
| | stop > 0.129
| | | final > 0.093: ham {ham=2, spam=0}
| | | final ≤ 0.093
| | | | know > 0.289: ham {ham=1, spam=0}
| | | | know ≤ 0.289: spam {ham=0, spam=32}
| | stop ≤ 0.129
| | | text_0 > 0.051
| | | | text_0 > 0.183
| | | | | mobil > 0.077: spam {ham=0, spam=3}
| | | | | mobil ≤ 0.077
| | | | | | 150p > 0.111: spam {ham=0, spam=2}
| | | | | | 150p ≤ 0.111
| | | | | | | 1unbreak > 0.157: spam {ham=0, spam=2}
| | | | | | | 1unbreak ≤ 0.157
| | | | | | | | 500 > 0.152: spam {ham=0, spam=2}
| | | | | | | | 500 ≤ 0.152
| | | | | | | | | 150pm > 0.105: spam {ham=0, spam=1}
| | | | | | | | | 150pm ≤ 0.105
| | | | | | | | | | 1million > 0.172: spam {ham=0, spam=1}
| | | | | | | | | | 1million ≤ 0.172
| | | | | | | | | | | 83110 > 0.202: spam {ham=0, spam=1}
| | | | | | | | | | | 83110 ≤ 0.202
| | | | | | | | | | | | bought > 0.226: spam {ham=0, spam=1}
| | | | | | | | | | | | bought ≤ 0.226
| | | | | | | | | | | | | eg23f > 0.154: spam {ham=0, spam=1}
| | | | | | | | | | | | | eg23f ≤ 0.154: ham {ham=53, spam=0}
| | | | text_0 ≤ 0.183
| | | | | end > 0.127: ham {ham=3, spam=0}
| | | | | end ≤ 0.127
| | | | | | affidavit > 0.197: ham {ham=1, spam=0}
| | | | | | affidavit ≤ 0.197
| | | | | | | chip > 0.173: ham {ham=1, spam=0}
| | | | | | | chip ≤ 0.173
| | | | | | | | fullon > 0.108: ham {ham=1, spam=0}
| | | | | | | | fullon ≤ 0.108: spam {ham=0, spam=39}
| | | text_0 ≤ 0.051
| | | | free > 0.074
| | | | | free > 0.156
| | | | | | end > 0.066
| | | | | | | ad > 0.163: ham {ham=1, spam=0}
| | | | | | | ad ≤ 0.163: spam {ham=0, spam=6}
| | | | | | end ≤ 0.066
| | | | | | | repli > 0.193: spam {ham=0, spam=3}
| | | | | | | repli ≤ 0.193
| | | | | | | | 80488biz > 0.336: spam {ham=0, spam=1}
| | | | | | | | 80488biz ≤ 0.336
| | | | | | | | | activ8 > 0.134: spam {ham=0, spam=1}
| | | | | | | | | activ8 ≤ 0.134
| | | | | | | | | | httptm > 0.235: spam {ham=0, spam=1}
| | | | | | | | | | httptm ≤ 0.235: ham {ham=27, spam=0}
| | | | | free ≤ 0.156
| | | | | | abl > 0.114: ham {ham=1, spam=0}
| | | | | | abl ≤ 0.114
| | | | | | | adi > 0.177: ham {ham=1, spam=0}
| | | | | | | adi ≤ 0.177
| | | | | | | | anythingtomorrow > 0.185: ham {ham=1, spam=0}
| | | | | | | | anythingtomorrow ≤ 0.185: spam {ham=0, spam=15}
| | | | free ≤ 0.074
| | | | | 500 > 0.085: spam {ham=0, spam=9}
| | | | | 500 ≤ 0.085
| | | | | | admir > 0.149: spam {ham=0, spam=7}
| | | | | | admir ≤ 0.149
| | | | | | | mobil > 0.065
| | | | | | | | mobil > 0.189
| | | | | | | | | club > 0.161: spam {ham=0, spam=2}
| | | | | | | | | club ≤ 0.161
| | | | | | | | | | content > 0.296: spam {ham=0, spam=1}
| | | | | | | | | | content ≤ 0.296: ham {ham=6, spam=0}
| | | | | | | | mobil ≤ 0.189
| | | | | | | | | bak > 0.125: ham {ham=1, spam=0}
| | | | | | | | | bak ≤ 0.125
| | | | | | | | | | bakrid > 0.156: ham {ham=1, spam=0}
| | | | | | | | | | bakrid ≤ 0.156: spam {ham=0, spam=8}
| | | | | | | mobil ≤ 0.065
| | | | | | | | 1000 > 0.094: spam {ham=0, spam=5}
| | | | | | | | 1000 ≤ 0.094
| | | | | | | | | fanta > 0.102
| | | | | | | | | | bigger > 0.206: ham {ham=1, spam=0}
| | | | | | | | | | bigger ≤ 0.206: spam {ham=0, spam=5}
| | | | | | | | | fanta ≤ 0.102
| | | | | | | | | | tone > 0.064
| | | | | | | | | | | good > 0.043: ham {ham=2, spam=0}
| | | | | | | | | | | good ≤ 0.043
| | | | | | | | | | | | famili > 0.327: ham {ham=1, spam=0}
| | | | | | | | | | | | famili ≤ 0.327: spam {ham=0, spam=6}
| | | | | | | | | | tone ≤ 0.064
| | | | | | | | | | | freem > 0.101: spam {ham=0, spam=3}
| | | | | | | | | | | freem ≤ 0.101
| | | | | | | | | | | | ltd > 0.092: spam {ham=0, spam=3}
| | | | | | | | | | | | ltd ≤ 0.092
| | | | | | | | | | | | | rington > 0.134: spam {ham=0, spam=3}
| | | | | | | | | | | | | rington ≤ 0.134
| | | | | | | | | | | | | | video > 0.105: spam {ham=0, spam=3}
| | | | | | | | | | | | | | video ≤ 0.105
| | | | | | | | | | | | | | | wap > 0.104: spam {ham=0, spam=3}
| | | | | | | | | | | | | | | wap ≤ 0.104
| | | | | | | | | | | | | | | | voucher > 0.099
| | | | | | | | | | | | | | | | | bdai > 0.158: ham {ham=1, spam=0}
| | | | | | | | | | | | | | | | | bdai ≤ 0.158: spam {ham=0, spam=3}
| | | | | | | | | | | | | | | | voucher ≤ 0.099
| | | | | | | | | | | | | | | | | 02073162414 > 0.172: spam {ham=0, spam=2}
| | | | | | | | | | | | | | | | | 02073162414 ≤ 0.172
| | | | | | | | | | | | | | | | | | 0870241182716 > 0.137: spam {ham=0, spam=2}
| | | | | | | | | | | | | | | | | | 0870241182716 ≤ 0.137
| | | | | | | | | | | | | | | | | | | 150pm > 0.137: spam {ham=0, spam=2}
| | | | | | | | | | | | | | | | | | | 150pm ≤ 0.137
| | | | | | | | | | | | | | | | | | | | 447797706009 > 0.193: spam {ham=0, spam=2}
| | | | | | | | | | | | | | | | | | | | 447797706009 ≤ 0.193
| | | | | | | | | | | | | | | | | | | | | 88066 > 0.327: spam {ham=0, spam=2}
| | | | | | | | | | | | | | | | | | | | | 88066 ≤ 0.327
| | | | | | | | | | | | | | | | | | | | | | couk > 0.152: spam {ham=0, spam=2}
| | | | | | | | | | | | | | | | | | | | | | couk ≤ 0.152
| | | | | | | | | | | | | | | | | | | | | | | fantast > 0.147: spam {ham=0, spam=2}
| | | | | | | | | | | | | | | | | | | | | | | fantast ≤ 0.147: ham {ham=3524,
spam=39}
```

Decision Tree Model

Certain token values (columns) were flagged for creating potential bias in the analysis.



Key	Annotation
age16 (Potential Bias)	This column was flagged since its name contains a suspicious term: age
color (Potential Bias)	This column was flagged since its name contains a suspicious term: color
gender (Potential Bias)	This column was flagged since its name contains a suspicious term: gender
race (Potential Bias)	This column was flagged since its name contains a suspicious term: race
age23 (Potential Bias)	This column was flagged since its name contains a suspicious term: age
origin (Potential Bias)	This column was flagged since its name contains a suspicious term: origin
sex (Potential Bias)	This column was flagged since its name contains a suspicious term: sex
birth (Potential Bias)	This column was flagged since its name contains a suspicious term: birth
accent (Potential Bias)	This column was flagged since its name contains a suspicious term: accent
win150ppmx3age16 (Potential Bias)	This column was flagged since its name contains a suspicious term: age
miss (Potential Bias)	This column was flagged since its name contains a suspicious term: miss
age16150pperm (Potential Bias)	This column was flagged since its name contains a suspicious term: age
male (Potential Bias)	This column was flagged since its name contains a suspicious term: male

TRAINING

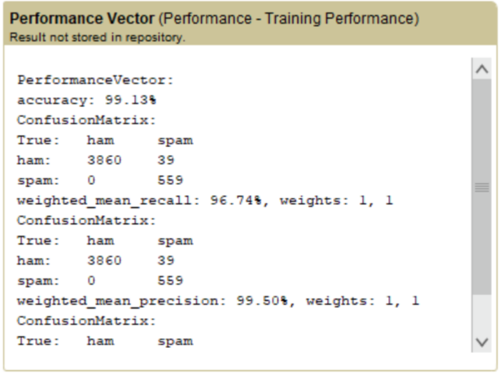
99.13%

Performance

Decision Tree Final Result

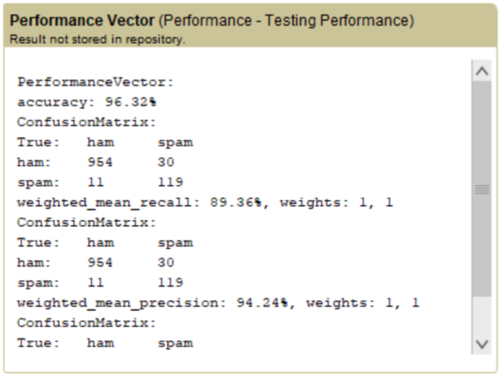
TEST

96.32%



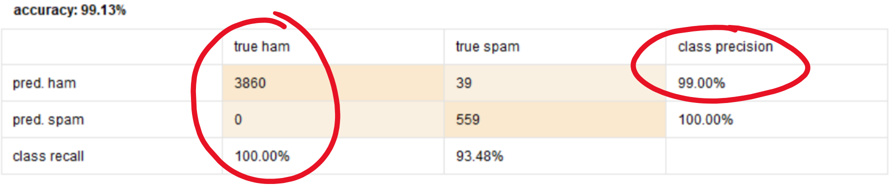
Performance Vector (Performance - Training Performance)
Result not stored in repository.

```
PerformanceVector:
accuracy: 99.13%
ConfusionMatrix:
True:   ham     spam
ham:    3860    39
spam:   0       559
weighted_mean_recall: 96.74%, weights: 1, 1
ConfusionMatrix:
True:   ham     spam
ham:    3860    39
spam:   0       559
weighted_mean_precision: 99.50%, weights: 1, 1
ConfusionMatrix:
True:   ham     spam
```



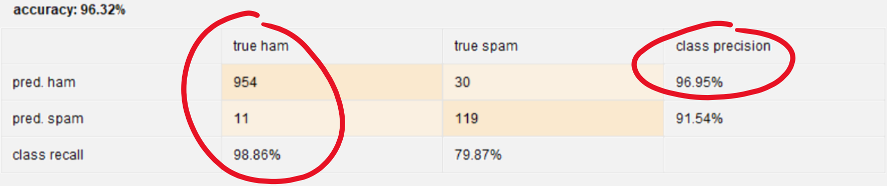
Performance Vector (Performance - Testing Performance)
Result not stored in repository.

```
PerformanceVector:
accuracy: 96.32%
ConfusionMatrix:
True:   ham     spam
ham:    954     30
spam:   11      119
weighted_mean_recall: 89.36%, weights: 1, 1
ConfusionMatrix:
True:   ham     spam
ham:    954     30
spam:   11      119
weighted_mean_precision: 94.24%, weights: 1, 1
ConfusionMatrix:
True:   ham     spam
```



accuracy: 99.13%

	true ham	true spam	class precision
pred. ham	3860	39	99.00%
pred. spam	0	559	100.00%
class recall	100.00%	93.48%	



accuracy: 96.32%

	true ham	true spam	class precision
pred. ham	954	30	96.95%
pred. spam	11	119	91.54%
class recall	98.86%	79.87%	

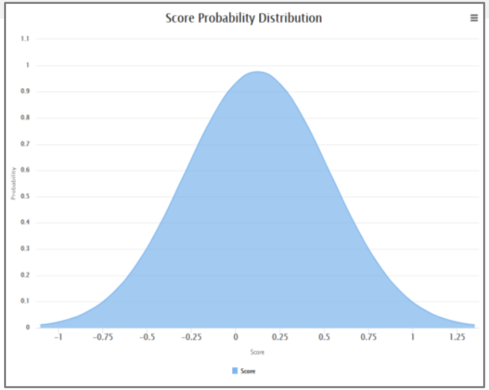
Paying special attention to TRUE HAM predicted as SPAM. The idea here is to choose a model that will provide a high percentage of accurate detection on this class of values. For every message that was True HAM errantly predicted as SPAM a possible important message that should make it through might get filtered. This is a larger risk and cost to the company than a True SPAM errantly being classified as HAM and therefor a message with no importance slips through the filter only to be viewed and discarded later. The resulting TEST PERFORMANCE VECTOR illustrated above had a 98.9% recall on this class and a 96.95% precision on predicted ham. This provides a high level of confidence for this consideration.

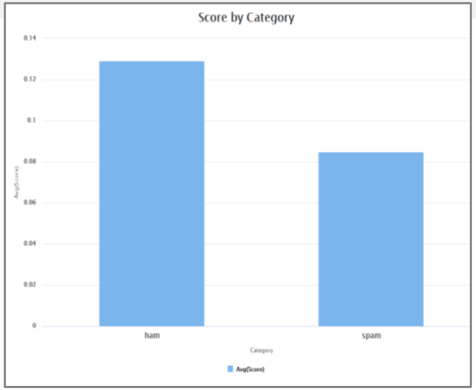
Sentiment Analysis

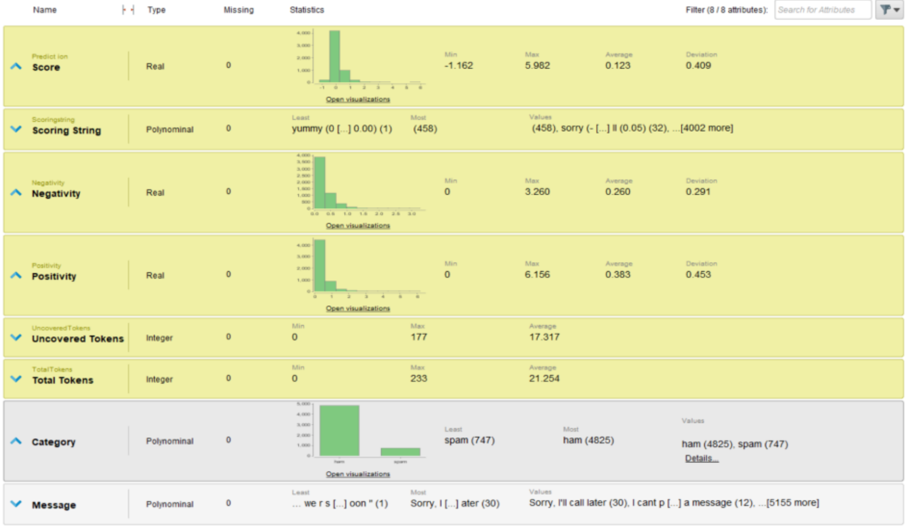
Sentiment analysis revealed a primarily positive sentiment for the data set with a fairly normal distribution of probability and frequency. Correlating to the majority of HAM examples, its is observable that most of the HAM messages were indeed delivered with Positive Sentiment.

The highest ***Positive Value Scoring Strings*** had strong positive words like “happy”, “love”, “great” and “attraction”. The highest ***Negative Value Scoring Strings*** included words like “hurt”, “die”, “grave” and “murderer”. Clearly the sentiment extraction appears to worked with respect to the intent of message.

The project requested the use of a “word cloud of sentences”. No known operators to this student are capable of creating such a word cloud. In the Live Mentoring Session, the mentor also mentioned that he felt request was errant as Sentence Clouds don’t make sense and did no offer a method to produce such an artifact.

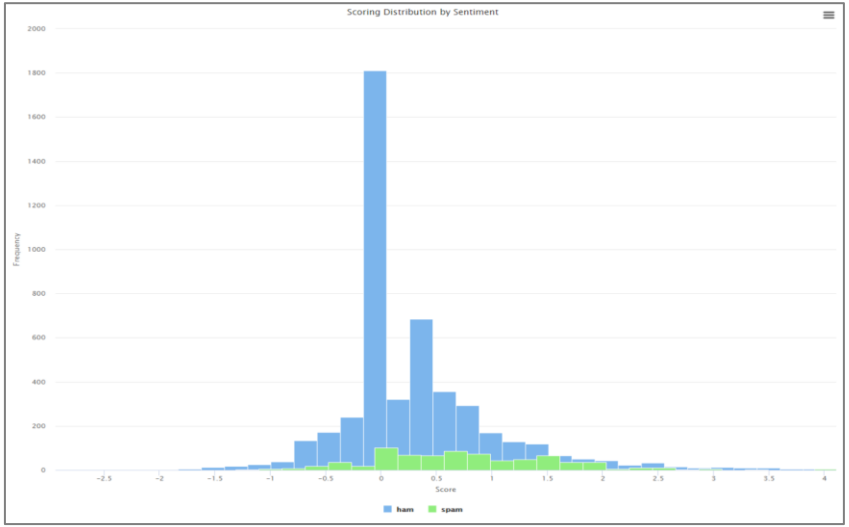






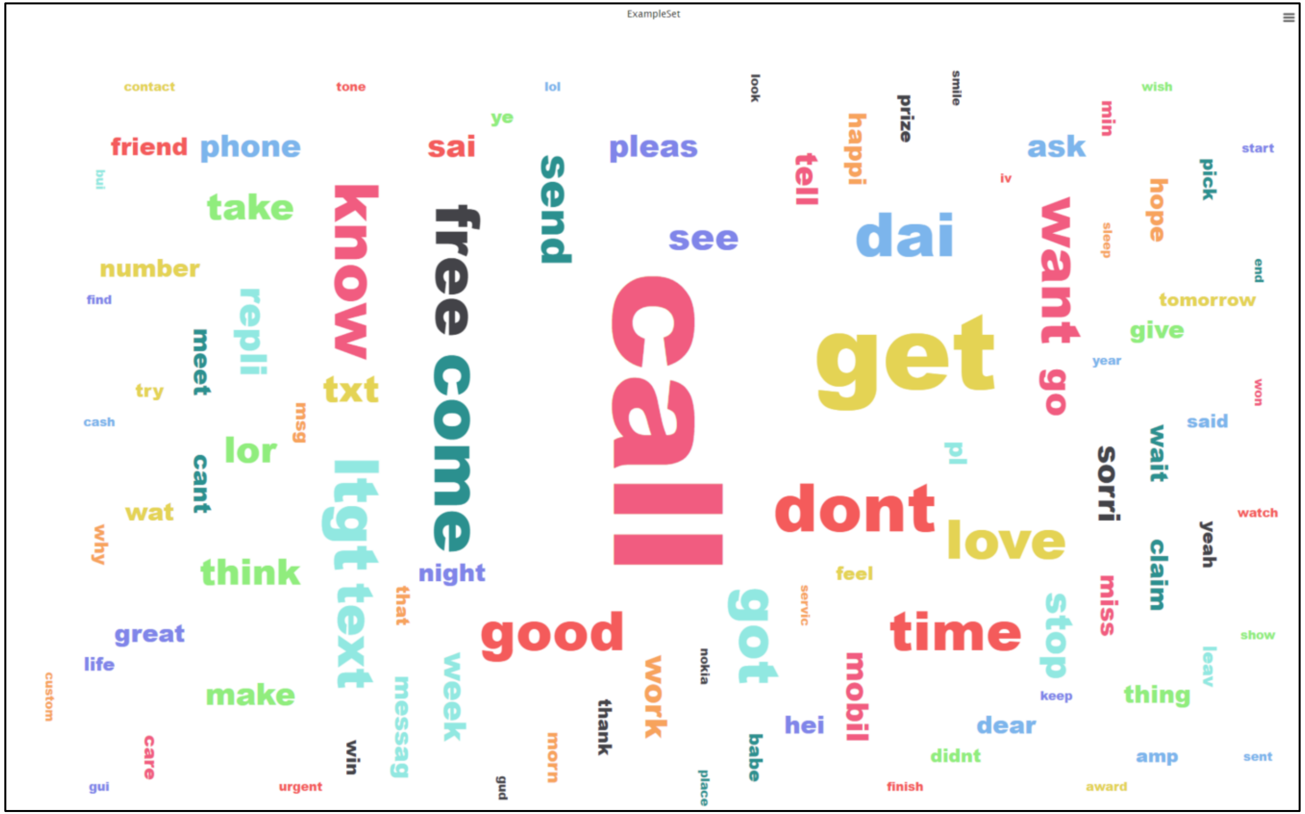
Filter (8 / 8 attributes): Search for Attributes

Name	Type	Missing	Statistics
Predict ion Score	Real	0	Open visualizations Min -1.162 Max 5.982 Average 0.123 Deviation 0.409
Scoringstring Scoring String	Polynominal	0	Least yummy (0 [...] 0.00) (1) Most (458) Values (458), sorry (- [...] ll (0.05) (32), ...[4002 more]
Negativity Negativity	Real	0	Open visualizations Min 0 Max 3.260 Average 0.260 Deviation 0.291
Positivity Positivity	Real	0	Open visualizations Min 0 Max 6.156 Average 0.383 Deviation 0.453
UncoveredTokens Uncovered Tokens	Integer	0	Min 0 Max 177 Average 17.317
TotalTokens Total Tokens	Integer	0	Min 0 Max 233 Average 21.254
Category	Polynominal	0	Open visualizations Least spam (747) Most ham (4825) Values ham (4825), spam (747) Details...
Message	Polynominal	0	Least ... we r s [...] oon " (1) Most Sorry, I [...] ater (30) Values Sorry, I'll call later (30), I cant p [...] a message (12), ...[5155 more]



Weighted Words (EDA)

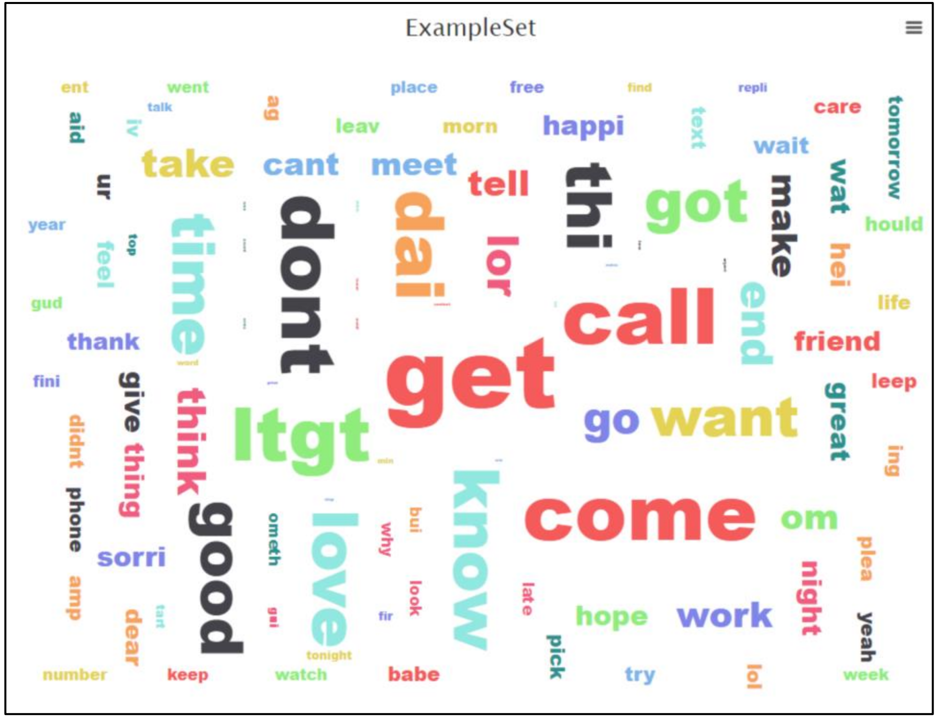
All Examples Word Cloud (top 100)

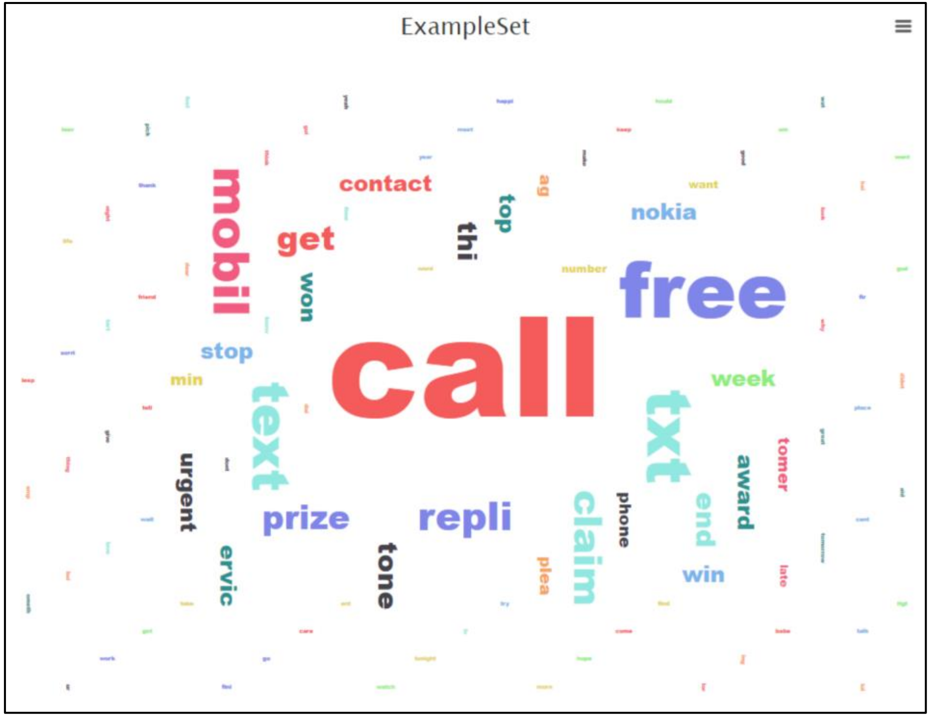


Weighted Words (EDA)

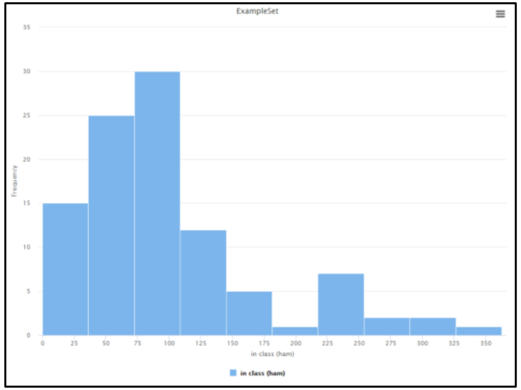
HAM word Cloud

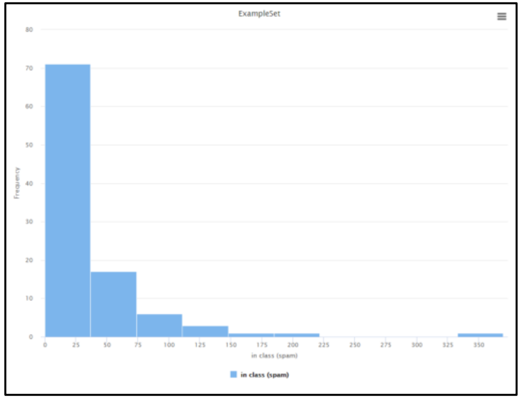
SPAM word Cloud



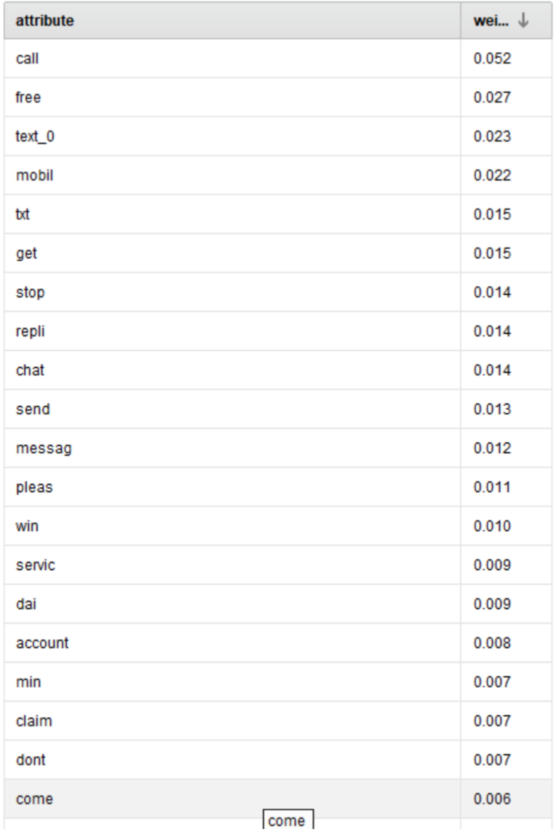


Weighted Words (EDA)





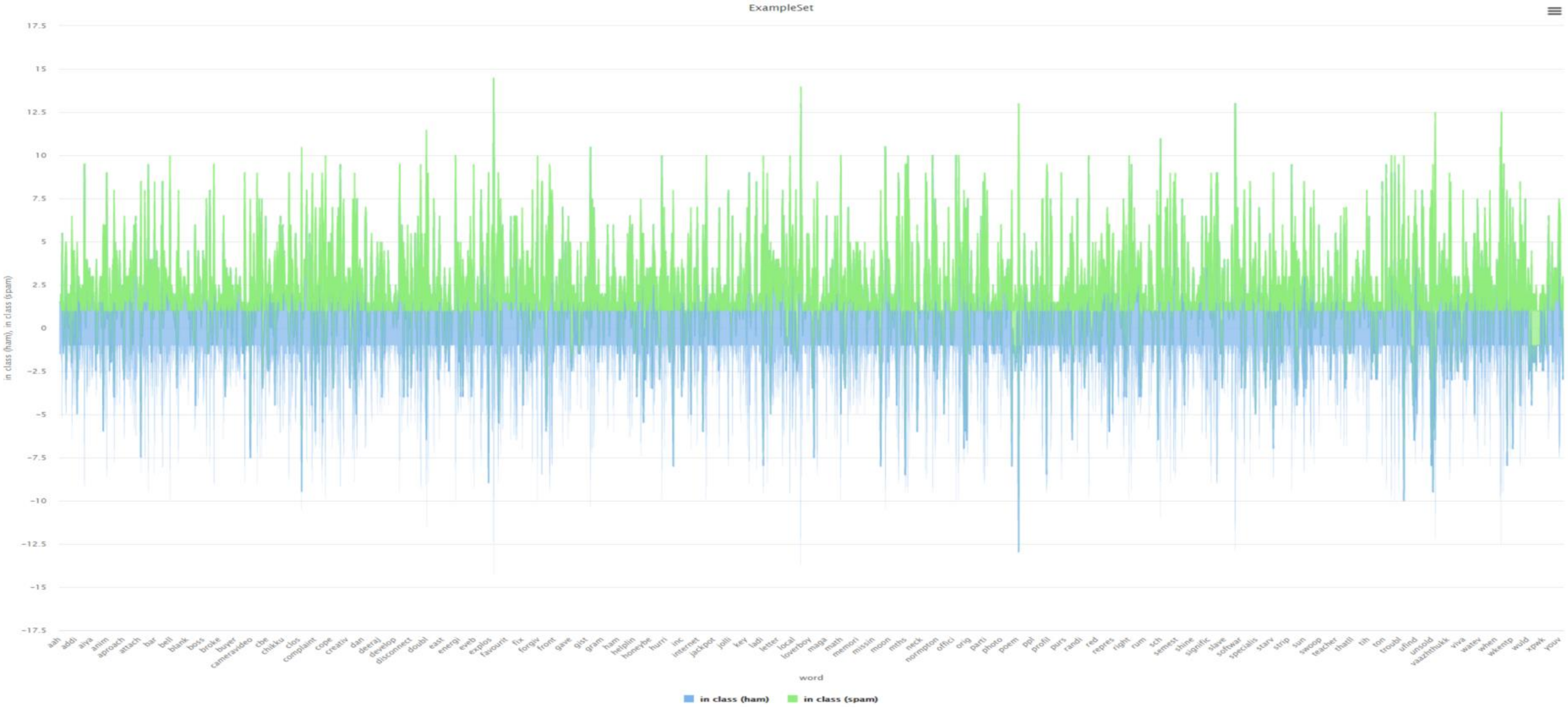
Top 20 Words in the dataset



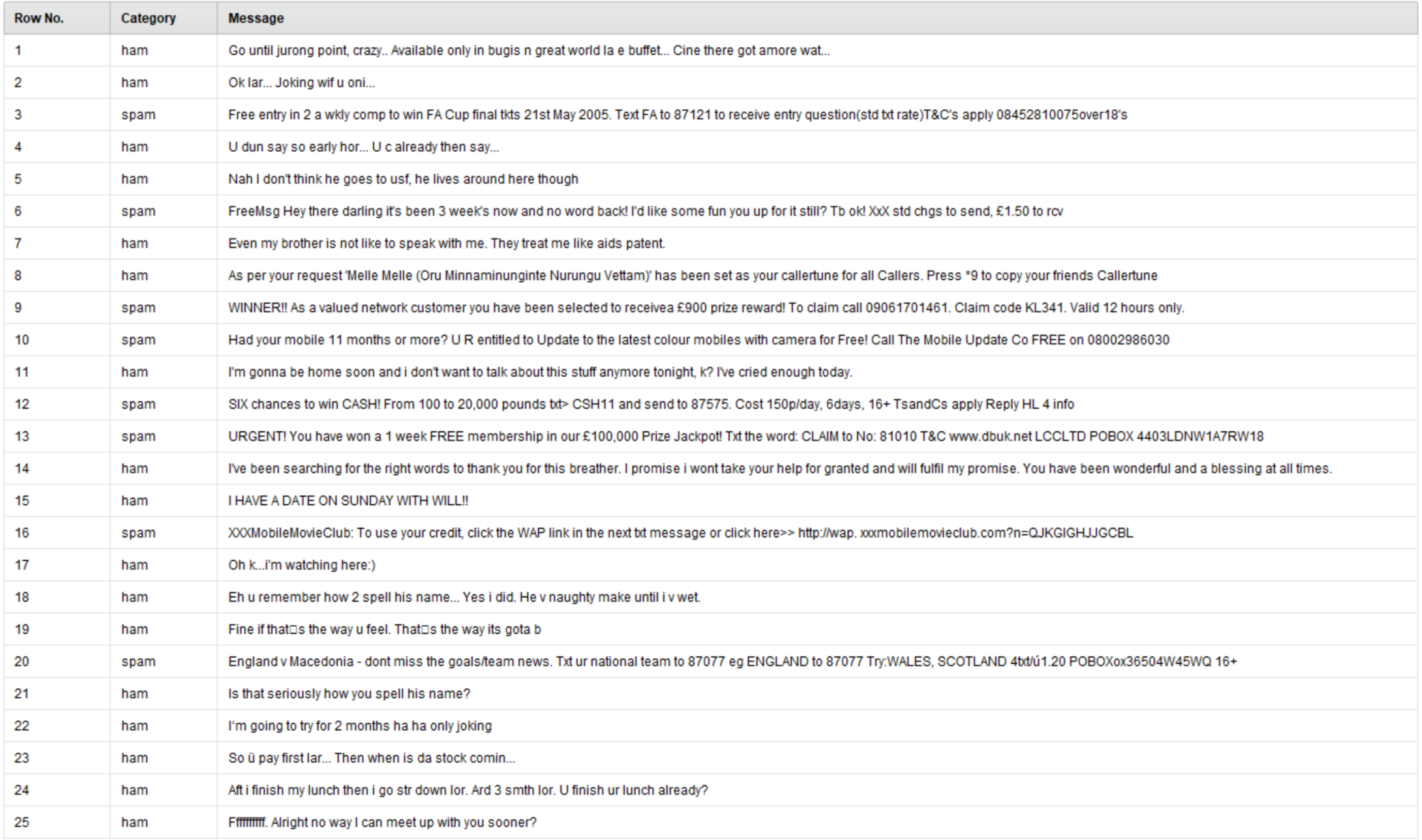
attribute	wei... ↓
call	0.052
free	0.027
text_0	0.023
mobil	0.022
txt	0.015
get	0.015
stop	0.014
repli	0.014
chat	0.014
send	0.013
messag	0.012
pleas	0.011
win	0.010
servic	0.009
dai	0.009
account	0.008
min	0.007
claim	0.007
dont	0.007
come	0.006

Word List Frequency Distribution of top 100 Words

Leveraging a stream graph to visualize the word frequency distribution of the top 200 words (tokens) by count in the data



Sampling of Source Data

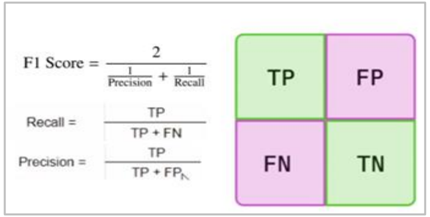


Row No.	Category	Message
1	ham	Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...
2	ham	Ok lar... Joking wif u oni...
3	spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's
4	ham	U dun say so early hor... U c already then say...
5	ham	Nah I don't think he goes to usf, he lives around here though
6	spam	FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun you up for it still? Tb ok! XxX std chgs to send, £1.50 to rcv
7	ham	Even my brother is not like to speak with me. They treat me like aids patent.
8	ham	As per your request 'Melle Melle (Oru Minnaminunginte Nurungu Vettam)' has been set as your callertune for all Callers. Press *9 to copy your friends Callertune
9	spam	WINNER!! As a valued network customer you have been selected to receivea £900 prize reward! To claim call 09061701461. Claim code KL341. Valid 12 hours only.
10	spam	Had your mobile 11 months or more? U R entitled to Update to the latest colour mobiles with camera for Free! Call The Mobile Update Co FREE on 08002986030
11	ham	I'm gonna be home soon and i don't want to talk about this stuff anymore tonight, k? I've cried enough today.
12	spam	SIX chances to win CASH! From 100 to 20,000 pounds txt> CSH11 and send to 87575. Cost 150p/day, 6days, 16+ TsandCs apply Reply HL 4 info
13	spam	URGENT! You have won a 1 week FREE membership in our £100,000 Prize Jackpot! Txt the word: CLAIM to No: 81010 T&C www.dbuk.net LCCLTD POBOX 4403LDNW1A7RW18
14	ham	I've been searching for the right words to thank you for this breather. I promise i wont take your help for granted and will fulfil my promise. You have been wonderful and a blessing at all times.
15	ham	I HAVE A DATE ON SUNDAY WITH WILL!!
16	spam	XXXMobileMovieClub: To use your credit, click the WAP link in the next txt message or click here>> http://wap. xxxmobilemovieclub.com?n=QJKGIGHJJGCBL
17	ham	Oh k...i'm watching here:)
18	ham	Eh u remember how 2 spell his name... Yes i did. He v naughty make until i v wet.
19	ham	Fine if that□s the way u feel. That□s the way its gota b
20	spam	England v Macedonia - dont miss the goals/team news. Txt ur national team to 87077 eg ENGLAND to 87077 Try:WALES, SCOTLAND 4txt/ú1.20 POBOXox36504W45WQ 16+
21	ham	Is that seriously how you spell his name?
22	ham	I‘m going to try for 2 months ha ha only joking
23	ham	So ü pay first lar... Then when is da stock comin...
24	ham	Aft i finish my lunch then i go str down lor. Ard 3 smth lor. U finish ur lunch already?
25	ham	Fffffffff. Alright no way I can meet up with you sooner?

Appendix: The Work of Model Comparison

These excel spreads were developed to drive the comparison of the various attempted models. While the results are relatively close, even the smallest of variances between accuracy, recall, precision and errors could make the difference of an important email getting through to the company or being removed by and overactive filter. As such, special attention is being paid to the Class Recall of True HAM examples.

In all cases the variance between Training and Test performance was <3% and just 2.6% on the chosen RF model. All of these models are well fitted.

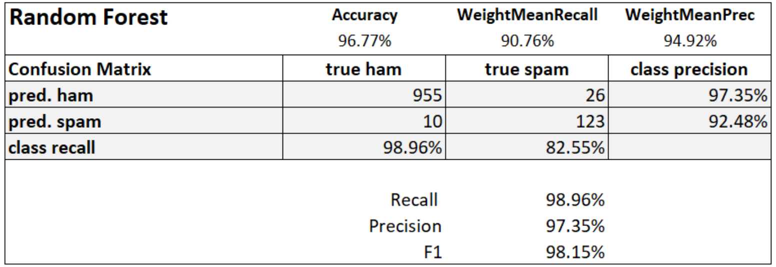


$$\text{F1 Score} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}}$$

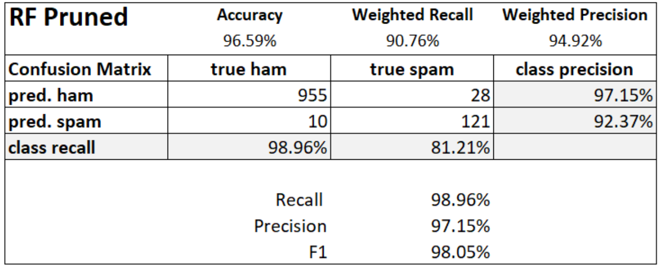
$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

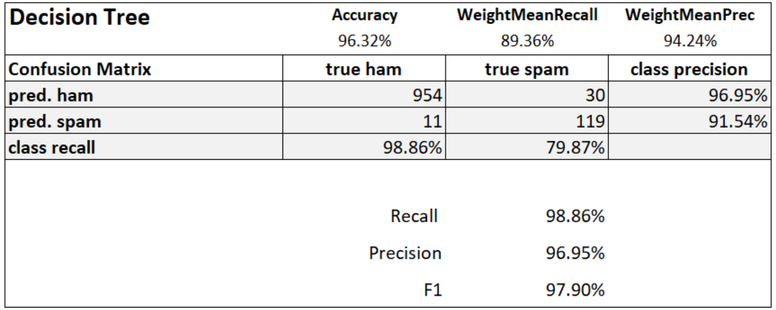
TP	FP
FN	TN



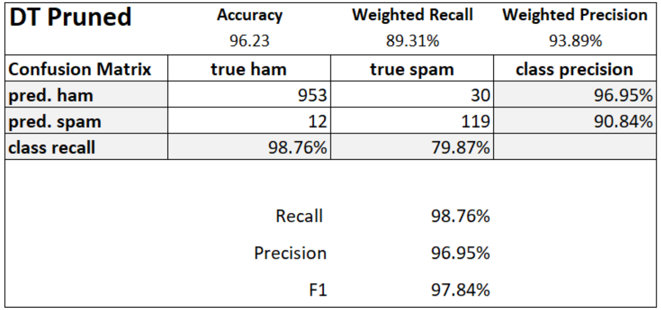
Random Forest	Accuracy	WeightMeanRecall	WeightMeanPrec
	96.77%	90.76%	94.92%
Confusion Matrix	**true ham**	**true spam**	**class precision**
pred. ham	955	26	97.35%
pred. spam	10	123	92.48%
class recall	98.96%	82.55%	
	Recall	98.96%	
	Precision	97.35%	
	F1	98.15%	



RF Pruned	Accuracy	Weighted Recall	Weighted Precision
	96.59%	90.76%	94.92%
Confusion Matrix	**true ham**	**true spam**	**class precision**
pred. ham	955	28	97.15%
pred. spam	10	121	92.37%
class recall	98.96%	81.21%	
	Recall	98.96%	
	Precision	97.15%	
	F1	98.05%	



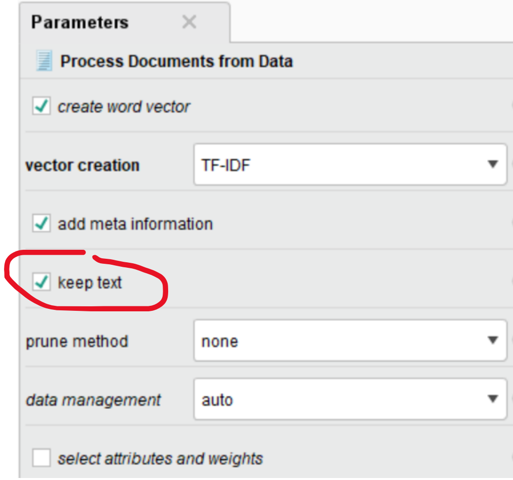
Decision Tree	Accuracy	WeightMeanRecall	WeightMeanPrec
	96.32%	89.36%	94.24%
Confusion Matrix	**true ham**	**true spam**	**class precision**
pred. ham	954	30	96.95%
pred. spam	11	119	91.54%
class recall	98.86%	79.87%	
	Recall	98.86%	
	Precision	96.95%	
	F1	97.90%	



DT Pruned	Accuracy	Weighted Recall	Weighted Precision
	96.23	89.31%	93.89%
Confusion Matrix	**true ham**	**true spam**	**class precision**
pred. ham	953	30	96.95%
pred. spam	12	119	90.84%
class recall	98.76%	79.87%	
	Recall	98.76%	
	Precision	96.95%	
	F1	97.84%	

Appendix: Model Key Parameters

Below are screenshot from the RapidMiner process of the Random Forest (non pruned) model. These images present the key parameters and hyper parameters of the process.



Parameters ✕

Process Documents from Data

☑ *create word vector*

vector creation TF-IDF ▾

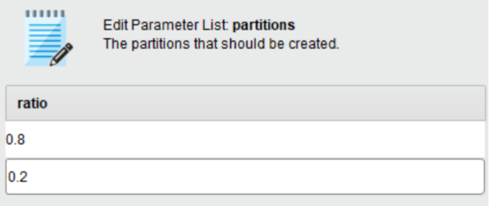
☑ add meta information

☑ keep text

prune method none ▾

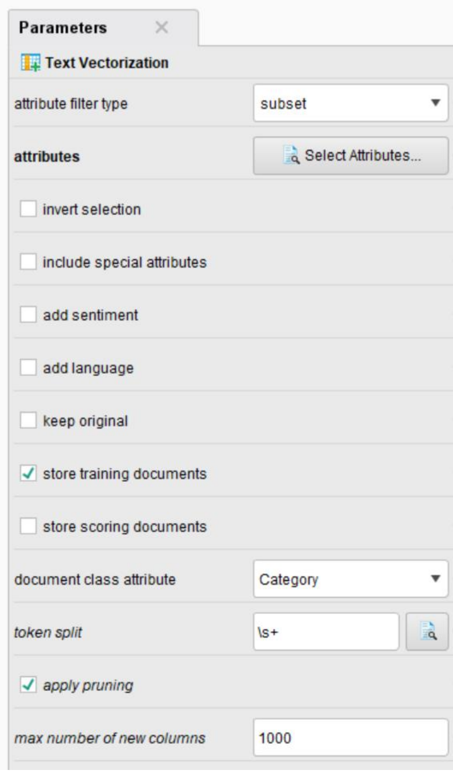
data management auto ▾

☐ *select attributes and weights*



Edit Parameter List: **partitions**
The partitions that should be created.

ratio
0.8
0.2



Parameters ✕

Text Vectorization

attribute filter type subset ▾

attributes Select Attributes...

☐ invert selection

☐ include special attributes

☐ add sentiment

☐ add language

☐ keep original

☑ store training documents

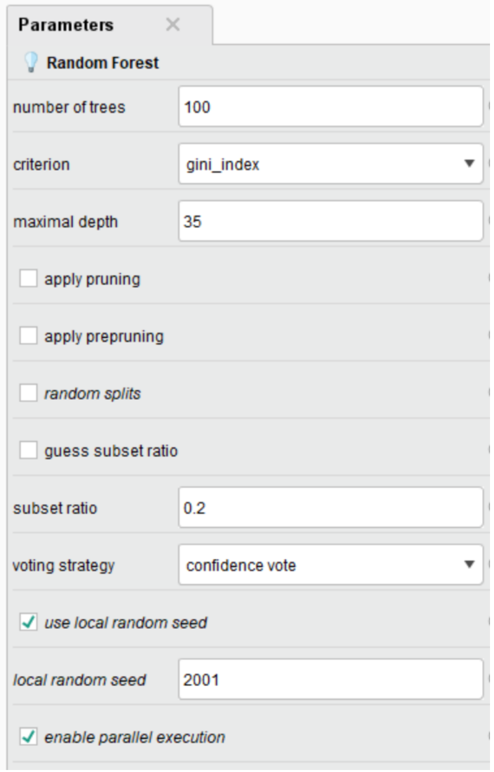
☐ store scoring documents

document class attribute Category ▾

token split \s+

☑ *apply pruning*

max number of new columns 1000



Parameters ✕

Random Forest

number of trees 100

criterion gini_index ▾

maximal depth 35

☐ apply pruning

☐ apply prepruning

☐ *random splits*

☐ guess subset ratio

subset ratio 0.2

voting strategy confidence vote ▾

☑ *use local random seed*

local random seed 2001

☑ *enable parallel execution*

Appendix: Future Considerations

Due to the very small variance and overall strong performance of the models, it is a difficult decision to choose any one model without also considering the level of resource consumption required to process the model. For this reason, future work on the actual systems of the company conducted on the actual inline scripting of the classifier would be very helpful in making a final determination of the best model to employ from a practicality standpoint.

Also, while its possible that links and external references occurred in the message text, it was not appropriately processed using the NLP operators employed here. It would be beneficial to determine if there are other fields or other operators that could aid in the detection of external links as the presence of one or more external link could assist in identifying “phishing” SPAM emails.

Finally, there are specific terms that almost exclusively appear in SPAM emails and almost never appear in HAM emails. Some of those terms include the words “Unsubscribe”, “Callback”, “Discounts”, “Free”, “Claim” and “Act Now”. A custom list of trigger words employed in an operator that would force classification with a higher weight than others could be useful in driving more accurate classification.

Thank You!