Distribution good and spam blogs in the dataset

|  |  |
| --- | --- |
| **Blog Type** | **No. of entries in DB** |
| Spam blog | 3683 |
| Good blog | 21424 |

First experiment that was run is listed below,

1. Training set created with TFIDF vectors and testing set created with Binary Term occurrences with random sampling from the dataset (spam and good blogs)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Training (No. of entries)** | | **Test (No. of entries)** | | **Accuracy (Average %)** |
| **Spam Blog** | **Good Blog** | **Spam Blog** | **Good Blog** |
| 50 | 200 | 3000 | 10000 | 88.85 |
| 50 | 200 | 1000 | 5000 | 90.03 |
| 100 | 500 | 3000 | 10000 | 91.34 |
| 100 | 500 | 1000 | 5000 | 93.26 |
| 100 | 500 | 100 | 500 | 93.56 |

1. Training and testing both created using Binary Term occurrences vector with random sampling from the dataset

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Training (No. of entries)** | | **Test (No. of entries)** | | **Accuracy (Average %)** |
| **Spam Blog** | **Good Blog** | **Spam Blog** | **Good Blog** |
| 50 | 200 | 3000 | 10000 | 84.85 |
| 50 | 200 | 1000 | 5000 | 90.03 |
| 100 | 500 | 3000 | 10000 | 86.13 |
| 100 | 500 | 1000 | 5000 | 90.83 |
| 100 | 500 | 100 | 500 | 90.28 |

1. Training and testing set both created using TFIDF vector along with random sampling from the dataset (TF-IDF vectors were created separately for training and test dataset)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Training (No. of entries)** | | **Test (No. of entries)** | | **Accuracy (Average %)** |
| **Spam Blog** | **Good Blog** | **Spam Blog** | **Good Blog** |
| 50 | 200 | 3000 | 10000 | 87.1 |
| 50 | 200 | 1000 | 5000 | 91.55 |
| 100 | 500 | 3000 | 10000 | 91.4 |
| 100 | 500 | 1000 | 5000 | 94.63 |
| 100 | 500 | 100 | 500 | 95.03 |

PS:

* The above results were obtained by running each experiment 3 to 5 times and each of those times, a new random test set was generated from the corpus but the same training set was used.
* Each of these experiments were run by sampling data randomly from the whole of the corpus.

The rapidminer process used to generate and save the training model was “svd\_blogs\_different\_train\_and\_test\_save\_model” and the rapidminer process used to generate the results using this saved model was “svd\_blogs\_different\_train\_and\_test\_use\_model”.

The highlighted result shows that TF-IDF based feature vector creation for both train and test set has better performance when compared to the other two approaches. I took this combination for comparison because the test set has enough representation of the dataset.

Second experiment includes picking training and test set as a random sampling from different parts of the dataset. Training set made up of 500 bad blogs and 500 good blogs was created through random sampling from the first 1000 bad blogs and first 10000 good blogs, whereas test set which was madeup of 500 bad blogs and 500 good blogs was generated by randomly sampling from the next 1000 bag blog entries and the next 10000 good blog entries correspondingly.

|  |  |  |
| --- | --- | --- |
| **Experiment** | | **Accuracy (Average %)** |
| **Training Set** | **Test Set** |
| Binary Vector | Binary Vector | 89.43 |
| Binary Vector | TF-IDF | 50 (all predicted 0) |
| TF-IDF | Binary Vector | 88.43 |
| TF-IDF | TF-IDF | 91.12 |

PS:

* Each of the above combination was run for 5 to 8 times.

From the above two experiments it is clear that TF-IDF feature vector creation for training and test set separately is the best combination.

Below is the important set of features when the training set was created using different approaches,

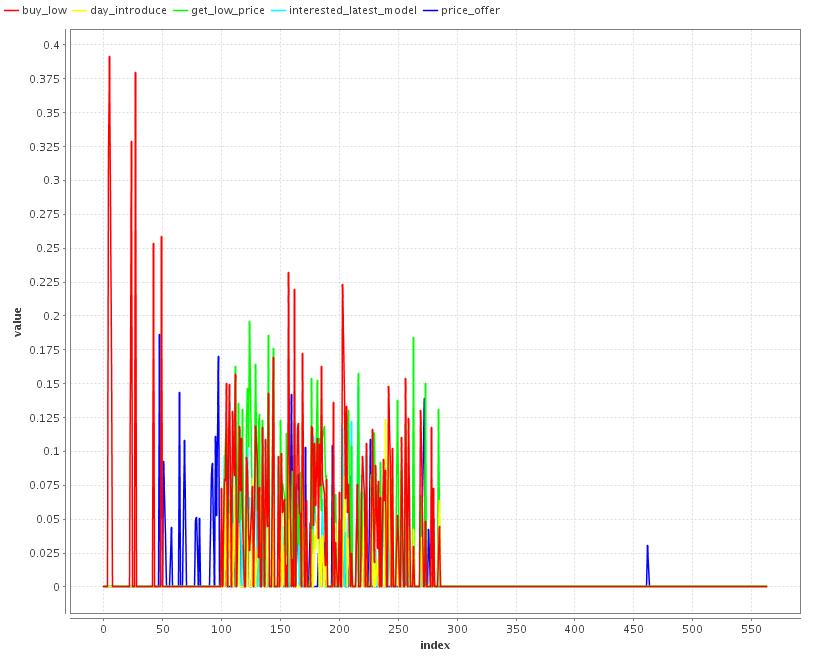
1. Significant features (in SVM weights) obtained through TF-IDF feature vector creation

|  |  |  |  |
| --- | --- | --- | --- |
| exist | travel | wide | ideal |
| source | type | compare | shipping |
| twitter | era | http | buy |
| marketing | status | net | discount |
| blog | come | savings | retail |
| park | links | offer | find |
| dan | filed | click | item |
| club | words | lowest | cheaper |
| care | view | selection | low |
| moon | miss | price | recommend |
| visit | comments | class | button |

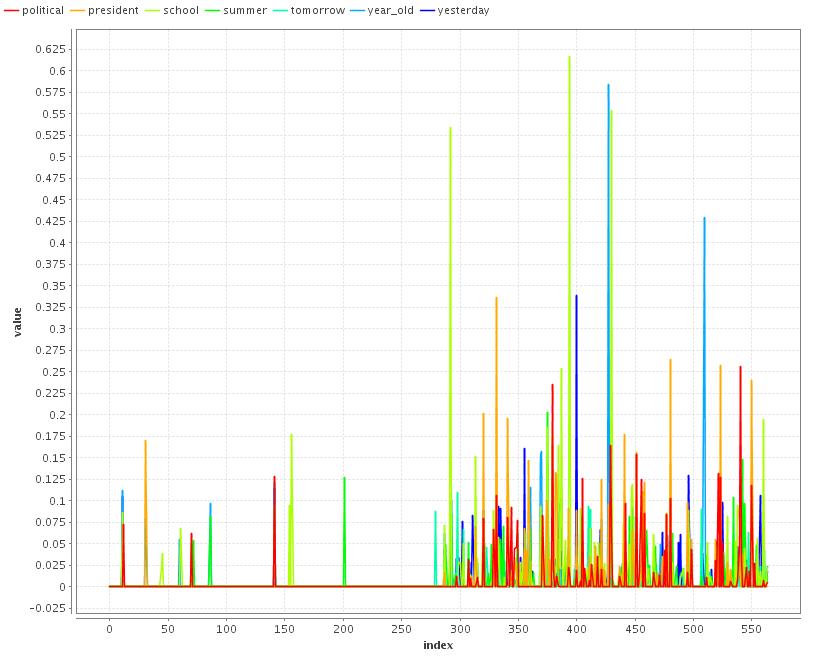
1. Significant features (in SVM weights) obtained through binary term occurrence - feature vector creation

|  |  |  |  |
| --- | --- | --- | --- |
| com | director | lowest | inch |
| obama | hosting | retail | value |
| los | email | cheaper | pleasure |
| april | visit | selection | prices |
| earth | city | item | low |
| spring | net | savings | stores |
| son | html | price | review |
| festival | please | deals | offer |
| capital | launch | compare | customer |
| tom | summer | supply | buy |
| monday | maker | wide | shop |

Here are two plots showing the distribution of terms (unigrams, bigrams and trigrams) across both the good and spam blogs. For this experiment, 300 spam and 300 good blogs were chosen. In both the plots the first 300 divisions on the X-axis represent the spam blogs and the 300 divisions on the X-axis represent the good blogs.



The above plot shows that the terms like “buy low”, “interested latest model”, “price offer” occur more often in spam blogs than in good blogs.



Similarly the above plot shows that the generic set of terms like political, president, summer, tomorrow etc. occur more often in good blogs than in spam blogs.