A Data Mining based Spam Detection System for YouTube

Rashid Chowdhury, Md. Nuruddin Monsur Adnan, G. A. N Mahmud, Rashedur M Rahman
Electrical Engineering and Computer Science Department
North South University
Dhaka, Bangladesh

ABSTRACT: *We have entered the era of social media networks represented by Facebook, Twitter, YouTube and Flickr. Internet users now spend more time on social networks than search engines. Business entities or public figures set up social networking pages to enhance direct interactions with on- line users. Social media systems heavily depend on users for content contribution and sharing. Information is spread across social networks quickly and effectively. However, at the same time social media networks become susceptible to different types of unwanted and malicious spammer. In order to increase the popularity of a video, marketing advertisements or simply pollute the system malicious users may post video response spam. A video spam is a video response whose content is not related to the topic being discussed in that particular video. In this project we consider finding out relation among different attribute that could lead to video spammers. We first construct moderate test collection of YouTube users, and manually classify them as either legitimate users or spammers. We then devise a number of attributes of video users and their social behavior which could potentially be used to detect spammers. Employing these attributes, we apply MICROSOFTs SQL server data mining tools (SSDT) to provide a heuristic for classifying an arbitrary video as either legitimate or spam. We then show that our approach succeeds at detecting much of the spam while only falsely classifying a small percentage of the legitimate videos as spam. Our results highlight the most important attributes for video response spam detection.*

Keywords: SQL Server data mining tools, Spam detection, You Tube, Social Networks, Data Mining, Spammers, Video Content

Received: 18 February 2014, Revised 21 March 2014, Accepted 28 March 2014

1. Introduction

Recently, online social networking services such as Facebook, Wikipedia and YouTube are experiencing a dramatic growth in terms of popularity. In particular, video content is becoming a predominant part of users' daily lives on the Web. By allowing users to generate and distribute their own multimedia content to large audiences, the Web is being transformed into a major channel for the delivery of multimedia. Video pervades the Internet and supports new types of interaction among users, including political debates, video chats, video mail, and video blogs. A number of Web services are offering video-based functions as alternative to text-based ones, such as video reviews for products, video ads and video responses [18]. In particular, the video response feature allows users to converse through video, by creating a video sequence that begins with an opening video and then followed with video responses from fans and detractors.

By allowing users to publicize and share their independently generated content, social video sharing systems may become susceptible to different types of malicious and opportunistic user actions, such as self-promotion, video aliasing and video spamming [6]. We define a video response spam as a video posted as a response to an opening video, but whose content is completely unrelated to the opening video. Video spammers are motivated to spam in order to promote specific content, advertise to generate sales, disseminate pornography (often as an advertisement) or compromise the system reputation. Spamming has been observed in several different contexts, including email [11], Web search engines [9] and blogs [19]. A number of spam detection techniques exploit characteristics present in the text (e.g., email body, commentaries in a blog) [14]. Moreover, users of such systems can quickly learn to identify some text spams (e.g., URLs to suspect Web sites), skipping or ignoring them. On the other hand, video spamming, particularly in social video sharing systems, can be much more challenging to detect and combat. Content-based detection techniques are not easily applied to non-textual video objects. On the other hand, exploiting characteristics of the traffic to specific videos, such as number of views and number of comments received can be useful to distinguish spams. We propose and evaluate a video spammer detection mechanism that classifies a user as a spammer based on the user's profile, the user's social behavior in the system, and the videos the user has posted. These attributes capture characteristics that are inherent to the user behavior and thus may better distinguish legitimate users from malicious video spammers.

In order to design and evaluate our proposed mechanism, we start by crawling a large user data set from YouTube, a pioneer social media sharing system which generates high volumes of Internet traffic and includes many social networking characteristics. A test collection is then built by carefully selecting users from the crawled data and manually classifying each user as either legitimate or spammer. Our test collection consists of 1800 users, 685 of which are classified as spammers. We then characterize several users and video attributes from our test collection, selecting those that may better distinguish spammers from legitimate users. The selected attributes are grouped into three subsets: user attributes, social network attributes, and video attributes. The user attributes, extracted from the user profile, expresses how the user typically uses the system (e.g., number of videos uploaded, number of friends, and so on). The social network attributes express how the user interacts with other users through video responses, whereas the video attributes capture the interests of other users in the content posted by user. Finally, using our test collection, we evaluate the effectiveness of our detection mechanism using the selected attributes. We also evaluate the relevance to the classification of each subset of attributes.

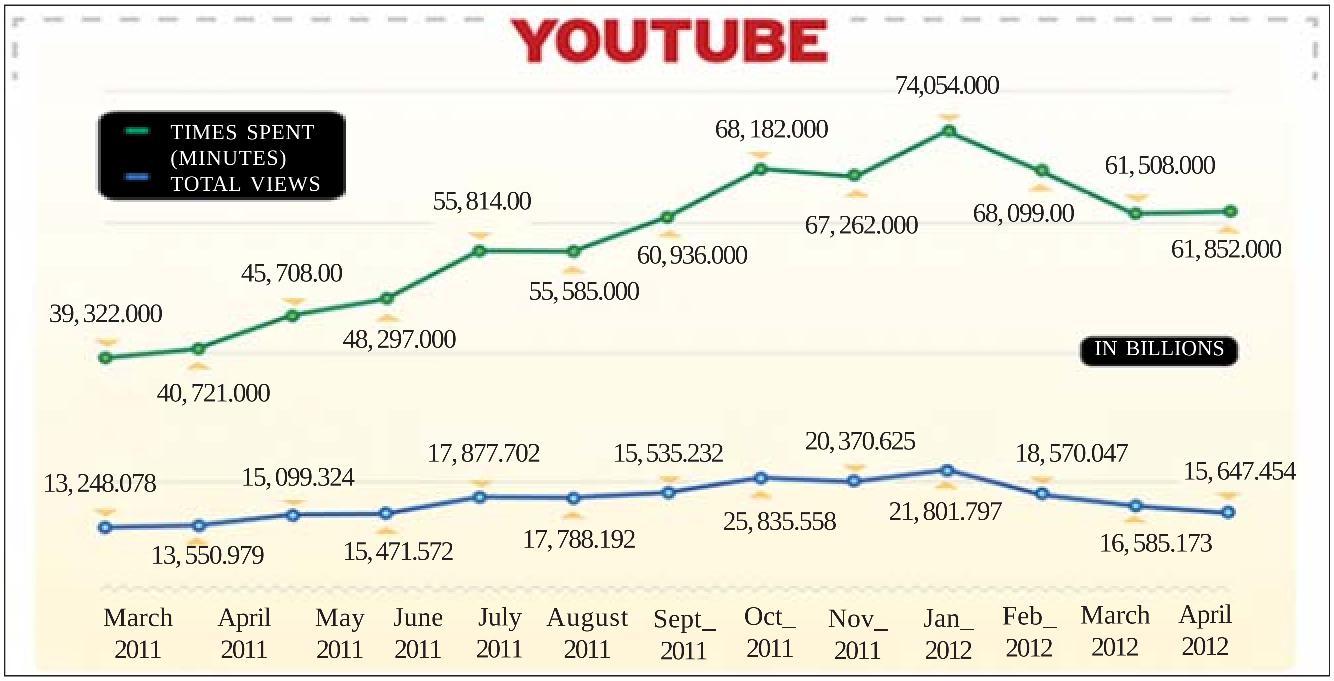


Figure 1. The amount of time spent on YouTube and the monthly view count Source: ComScore

In summary, the main contributions of this project are:

• Quantitative evidence of video spamming activity (as defined above) in social online video sharing systems, particularly YouTube

• The identification and characterization of a set of user and video attributes that can be used to distinguish video spammers from legitimate users.

• A test collection of users from YouTube, classified as spammers or legitimate users.

• A video spammer detection mechanism based on a classification algorithm, decision tree making algorithm and naïve Bayes algorithm generated by Microsoft's built in algorithm generation for clustering, decision tree making and Naïve Bayes which showed to produce reasonably good results, detecting a significant fraction of video spammers with 2% of misclassification of legitimate users.

• Predict the spam and eventually spammers from the data mining model.

• The rest of the paper is organized as follows. The next section briefly discusses related work.

2. Related Work

Mechanisms to detect and identify spam and spammers have been largely studied in the context of Web [5,13] and email spamming [12]. In particular, Castillo et al [5] proposed a framework to detect Web spamming which uses social network metrics. A framework to detect spamming in tagging systems, which is a type of attack that aims at raising the visibility of specific objects, was proposed in [17]. Although applicable to social media sharing systems that allow object tagging by users, such as YouTube, the proposed technique exploits a specific object attribute, i.e., its tags. Our approach is complementary to these efforts as it aims at detecting video spammers, using a combination of different categories of attributes of both objects and users. A survey of approaches to combat spamming in Social Web sites is presented in [14]. Many existing approaches are based on extracting evidence from the content of a text, treating the text corpus as a set of objects with associated attributes and using these attributes to detect spam. These techniques, based on content classification, can be directly applied to textual information, and thus can be used to detect spam in email, text commentaries in blogs, forums, and online social networking sites. Additionally, detection of email spam based on image content was also studied previously [2, 22]. However, content classification is much harder to do for video objects. Our approach to detect video spammers consists on classifying users, as well as their videos, and relies on a set of attributes associated to the user actions and social behavior in the system as well as attributes of their videos. Towards this end, this paper presents a characterization of user and video attributes that can be used to distinguish spammers from legitimate users in YouTube.

Our project was particularly based on a paper published by several research fellows from Federal University of Minas Gerais, Brazil and Polytechnic University, Brooklyn, NY, USA.

3. Youtube Measurements

Our ultimate goal is to design a mechanism to classify users of social video sharing systems into legitimate and video spammers, using a set of their attributes and of their contributed videos. Towards this goal, we crawled data from YouTube, one of the most popular social media networking sites today [1]. A test collection, including a sample of the crawled data, was then built and used to evaluate the effectiveness of our classification approach. Section 3.1 describes our crawling strategy, whereas Section 3.2 presents the criteria used to select users for the test collection.

3.1 Youtube Data Collection

Our crawling strategy is driven by using TubeKit. It is a toolkit for creating customized YouTube crawlers. It allows one to build one's own crawler that can crawl through YouTube based on a set of seed queries and collect up to 16 different attributes.

TubeKit assists in all the phases of this process starting database creation to finally giving access to the collected data with browsing and searching interfaces. In addition to creating crawlers, TubeKit also provides several tools to collect a variety of data from YouTube, including video details and user profiles.

Following is a brief description of its workflow:

• We provide a set of seed queries to monitor.

• The system uses these queries to go out and search on YouTube.

• A set of metadata is extracted from a subset of the results returned from YouTube. We define metadata to be the information about the given video which are provided by the author of that video, and are usually static in nature. For instance, the genre of the video.

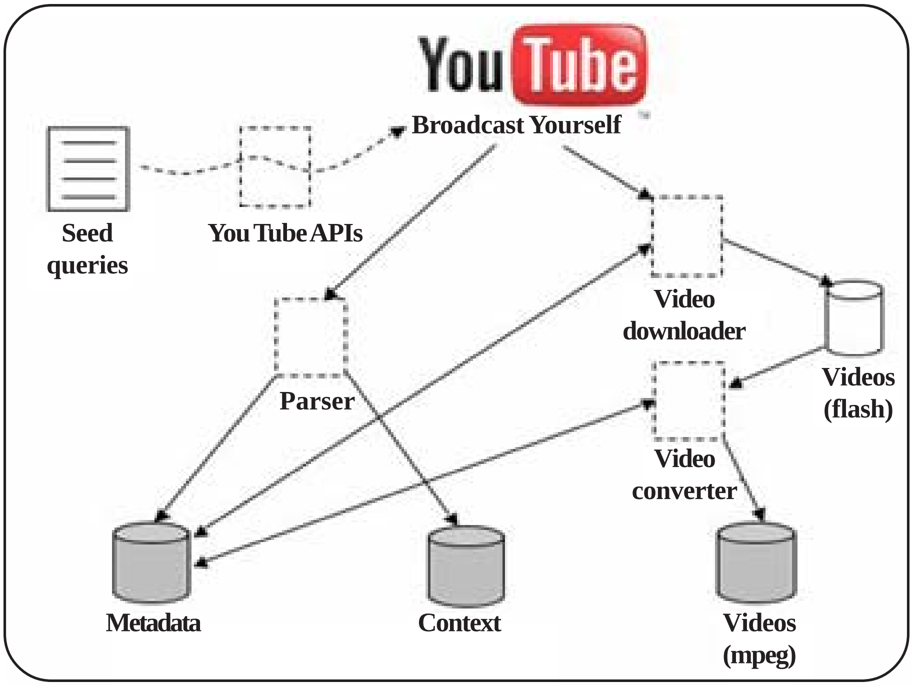


Figure 3.1.1. Our scheme for query-based YouTube crawling

- The video downloader component checks the metadata table to see which videos have not been previously downloaded and collects those videos in ash format from YouTube.

- The video converter component checks which videos are downloaded and not converted, and converts them into mpeg format.

- The context capturing component goes out to YouTube and captures various contextual information about the video items for which the metadata is already collected. Each time such social context is captured, a time-stamp is recorded. We define social context as the data contributed by the visitors to a video page. This would include fields such as ratings and comments. Note that other types of social context in blogs and other sources could also be harvested with different components (discussed later). The context capturing component runs periodically and updates time-sensitive data such as new comments or video postings, thus capturing temporal context.

The data is then saved in mysql database. We then imported the data to Microsoft Excel 2010 to run the main data mining operations.

From the crawler we found some attribute that we could use in our experiment. The name of the attribute that we found is given below.

YouTube ID, Username, upload time, duration, category, video url, video count, view count, rating average, rating count, comment count, spam.

We also verified each video separately to understand whether the video is spam or not. Which has never been done before in similar kind of research or projects.

The design of our crawler is shown in Figure 3.1.1 and was first presented in (Chirag Shah & Marchionini, 2007). It's a php based program runs from a webserver.

3.2 Test Collection Definition

A test collection, containing a set of YouTube users each pre-classified as legitimate or video spammer is required to evaluate the effectiveness of our classification approach.

However, as far as we know, no such collection is publicly available (neither for YouTube nor for any other video sharing system). But how do we create a large and representative test collection? Relying on random sampling to select a reasonable number of users from the crawled data would not be advisable as it could yield a very small fraction of spammers, preventing a sound analysis of the results.

Characteristics	Video Response Dataset
Sample Period	05-04-2013/01-05-2013
# of Videos	1719
# of distinct users	1428
# of comments	10102865
#of ratings count	23013568
#of different categories	21

Table 1. Summary of Video Response Data Set

4. Spammer Detection Mechanism

From the excel file that we created we connected that file with Microsoft SQL server 2012. Then we ran data mining operations with the help of SSDT and excel data mining add-in for SQL server 2012. For deploying the data mining techniques we first needed to create a new Analysis Services database, add a data source and data source view, and prepare the new database to be used with data mining.

After creating the data source from the excel file that we extracted we also needed to create a data source viewer to see the tables and views of the database.

We applied several data-mining methods to understand and predict the behavior of a YouTube video. Among these methods are,

- Decision Tree
- Naïve Bayes
- Clustering
- Neural Networking

We start with Decision Tree.

4.1 Decision Tree

Decision tree learning is a method commonly used in data mining. The goal is to create a model that predicts the value of a target variable based on several input variables. An example is shown on the right. Each interior node corresponds to one of the input variables; there are edges to children for each of the possible values of that input variable. Each leaf represents a value of the target variable given the values of the input variables represented by the path from the root to the leaf.

The Microsoft Decision Trees algorithm is a classification and regression algorithm provided by Microsoft SQL Server Analysis Services for use in predictive modeling of both discrete and continuous attributes.

For discrete attributes, the algorithm makes predictions based on the relationships between input columns in a dataset. It uses the values, known as states, of those columns to predict the states of a column that you designate as predictable. Specifically, the algorithm identifies the input columns that are correlated with the predictable column.

For continuous attributes, the algorithm uses linear regression to determine where a decision tree splits.

If more than one column is set to predictable, or if the input data contains a nested table that is set to predictable, the algorithm builds a separate decision tree for each predictable column.

In our dataset, the classification is done mainly based on average rating, category and comment count. It is a 5 level decision tree. And has 0 missing value on our training set.

4.2 Clustering

The Microsoft Clustering algorithm is a segmentation algorithm provided by Analysis Services. The algorithm uses iterative techniques to group cases in a dataset into clusters that contain similar characteristics. These groupings are useful for exploring data, identifying anomalies in the data, and creating predictions.

The Microsoft Clustering algorithm first identifies relationships in a dataset and generates a series of clusters based on those relationships. A scatter plot is a useful way to visually represent how the algorithm groups data, as shown in the following diagram. The scatter plot represents all the cases in the dataset, and each case is a point on the graph. The clusters group points on the graph and illustrate the relationships that the algorithm identifies.

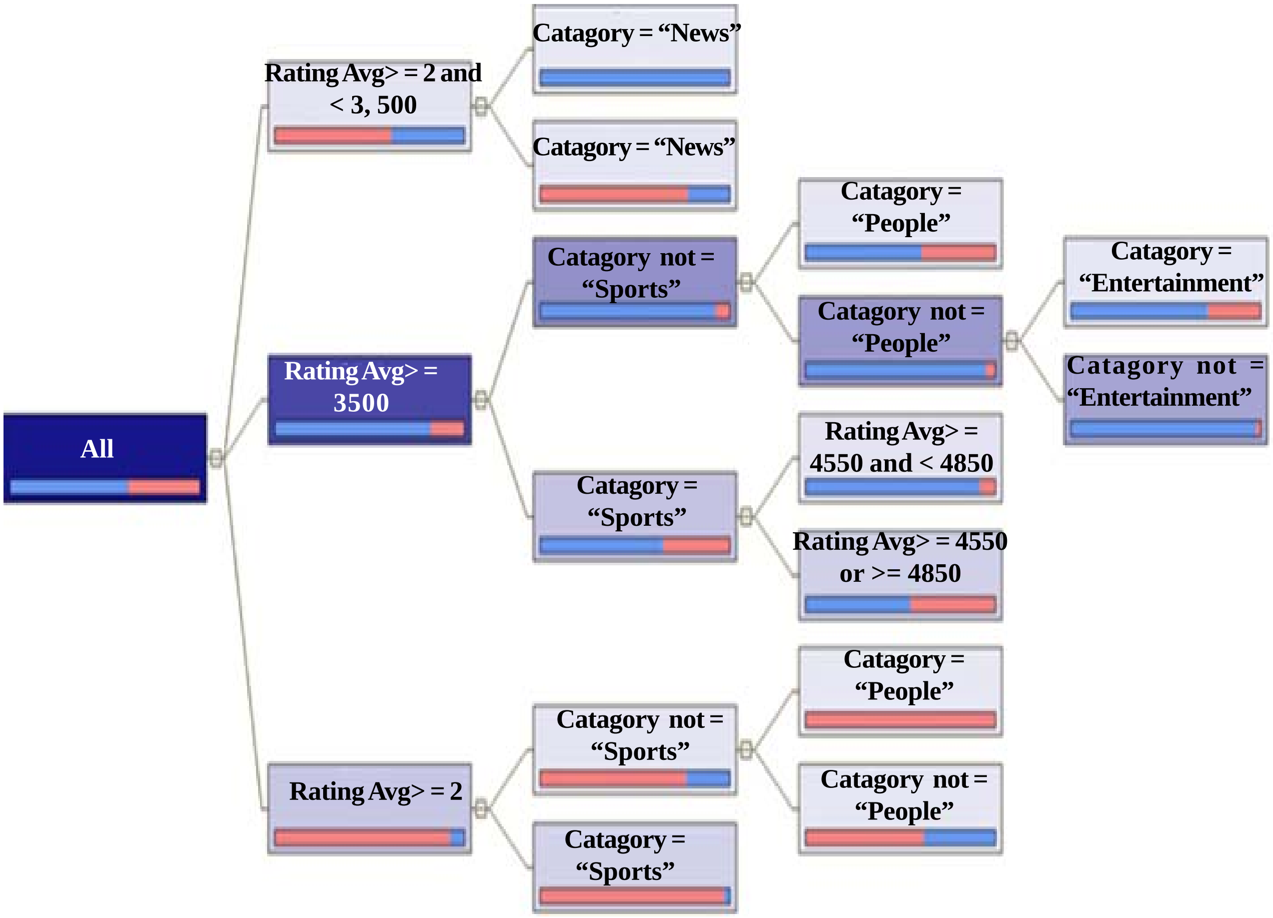
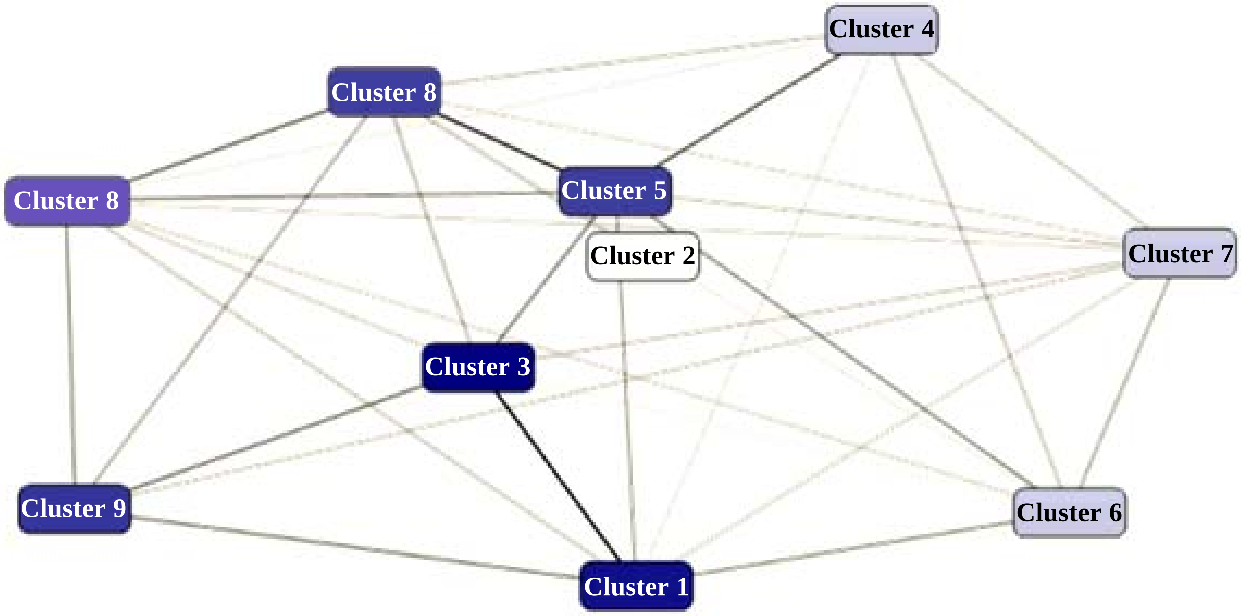


Figure 4.1.1. Decision Tree on YouTube spam detection

After first defining the clusters, the algorithm calculates how well the clusters represent groupings of the points, and then tries to redefine the groupings to create clusters that better represent the data. The algorithm iterates through this process until it cannot improve the results more by redefining the clusters.

When we view a clustering model, Analysis Services shows the clusters in a diagram that depicts the relationships among



Cluster 1		
Variables	Values	Probability
Category	Sports	99.793 %
Username	missing	93.907 %
Yt Id	missing	89.890 %
Spam	YES	61.309 %
Spam	NO	38.691 %
Username	BfASport	0.580 %
Yt Id	#NAME?	0.504 %

Cluster 3		
Variables	Values	Probability
Yt Id	missing	98.032 %
Category	Sports	98.013 %
Spam	YES	65.471 %
Spam	NO	34.529 %
Username	BfASport	9.095 %
Username	missing	5.880 %
Username	HeilRJ03	4.747 %

clusters, and also provides a detailed profile of each cluster, a list of the attributes that distinguish each cluster from the others, and the characteristics of the entire training data set.

Our whole training set was divided into 10 clusters. Among these clusters cluster 1, cluster 3, cluster 5 and cluster 9 have the highest numbers of spam attribute in them. If we analysis the clusters separately we will be able to understand the relation among the attributes better.

Cluster 5		
Variables	Values	Probability
Yt Id	missing	94.051 %
Username	missing	79.429 %
Spam	NO	50.332 %
Spam	YES	49.668 %
Category	Entertainment	48.692 %
Category	People	40.094 %
Username	RayWilliamJohnson	6.384 %

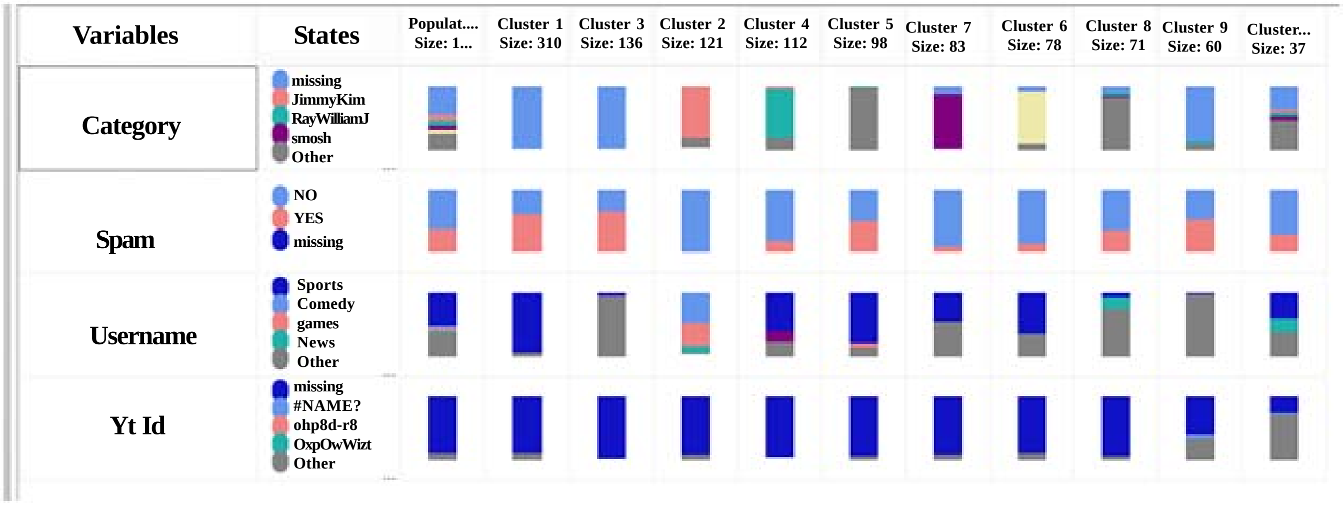
Cluster 9		
Variables	Values	Probability
Category	Sports	85.322%
Yt Id	missing	60.998%
Spam	YES	52.316%
Spam	NO	47.684%
Username	WorldNews365	9.625%
Username	Futbolpasionmundial3	9.282%
Yt Id	#NAME?	7.352%
Username	YeeaaahitIsntme	7.286%

In cluster 1 (Table 4.2.1) we see a strong relationship between Sports category with probable spam count. There are also minor connection with other attributes.

In cluster 3 (Table 4.2.2) we again see a strong relationship between Sports category with probable spam count. There are also other minor connections with other attributes. We also see that spam and non-spam videos have similar attributes. As a result they tend to cluster together in some cases.

Similarly in cluster 5 different YouTube id and usernames show closeness to spamming probabilities. Since, YouTube id or username doesn't help us understand whether a video is spam or not this cluster is less of importance.

Similarly in cluster 9 different YouTube id and a particular username, "*WorldNews365*" show closeness to spamming probabilities. Since, YouTube id or username doesn't help us understand whether a video is spam or not this cluster is less of importance.



4.3 Naïve Bayes Model

The Microsoft Naive Bayes algorithm is a classification algorithm based on Bayes' theorems, and provided by Microsoft SQL Server Analysis Services for use in predictive modeling. The word naïve in the name Naïve Bayes derives from the fact that the algorithm uses Bayesian techniques but does not take into account dependencies that may exist. For more information about Bayesian methods, see Microsoft Research Community.

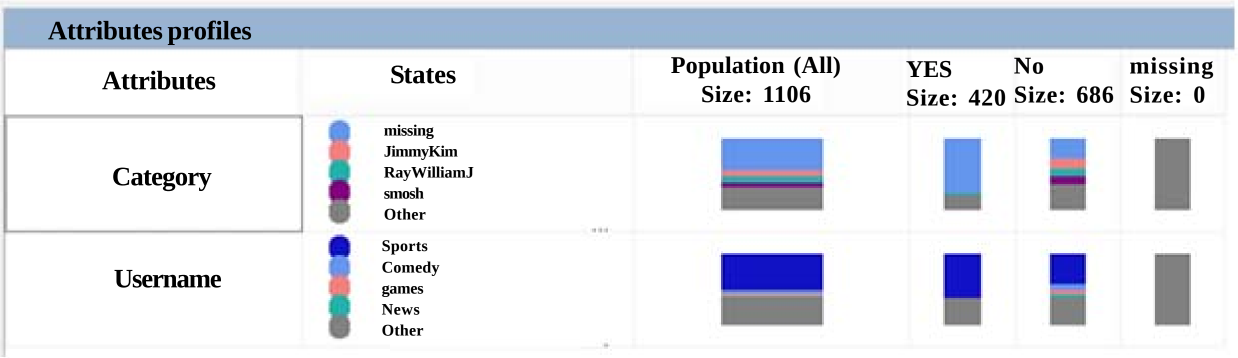
This algorithm is less computationally intense than other Microsoft algorithms, and therefore is useful for quickly generating mining models to discover relationships between input columns and predictable columns. You can use this algorithm to do initial exploration of data, and then later you can apply the results to create additional mining models with other algorithms that are more computationally intense and more accurate.

The Microsoft Naive Bayes algorithm calculates the probability of every state of each input column, given each possible state of the predictable column.

To understand how this works, use the Microsoft Naive Bayes Viewer in SQL Server Data Tools (SSDT) (as shown in the following graphic) to visually explore how the algorithm distributes states.

5. Final Analysis

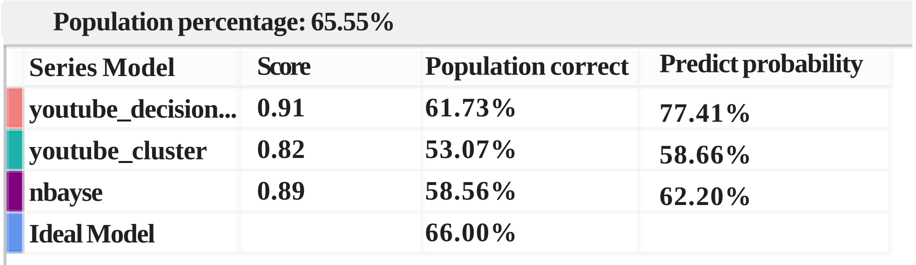
Among the three analysis methods we see a similar result. We can generate a Lift Chart. A Lift Chart graphically represents the improvement that a mining model provides when compared against a random guess, and measures the change in terms of a lift score. By comparing the lift scores for various portions of your data set and for different models, you can determine which model is best, and which percentage of the cases in the data set would benefit from applying the model's predictions.



Attributes profiles

Attributes	States	Population (All) Size: 1106	YES Size: 420	No Size: 686	missing Size: 0
Category	missing JimmyKim RayWilliamJ smosh Other				
Username	Sports Comedy games News Other				

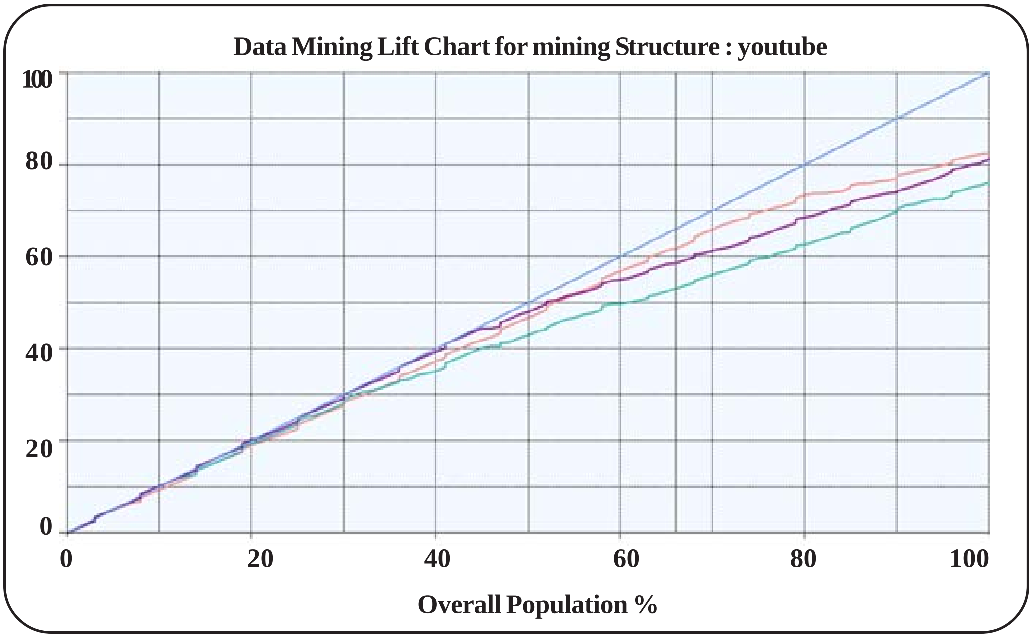
With a lift chart, you can compare the accuracy of predictions for multiple models that have the same predictable attribute. You can also assess the accuracy of prediction either for a single outcome (a single value of the predictable attribute), or for all outcomes (all values of the specified attribute).



Population percentage: 65.55%

	Series Model	Score	Population correct	Predict probability
🟥	youtube_decision...	0.91	61.73%	77.41%
🟩	youtube_cluster	0.82	53.07%	58.66%
🟪	nbayse	0.89	58.56%	62.20%
🟦	Ideal Model		66.00%	

This table (Table 5.1) tells us that, at 66.0 percent of the population, the model that we created correctly predicts 53.07% for clustering, 61.37% for decision tree and 58.56% of the cases for naïve Bayes. We might consider this a reasonably accurate model. However, we have to remember that this particular model predicts all values of the predictable attribute. Therefore, the models might be accurate in predicting accordingly.



References

[1] Alexa. http://www.alexa.com.

[2] Aradhye, H., Myers, G., Herson, J. (2005). Image analysis for efficient categorization of image-based spam e-mail. *In*: Proc. of the International Conf. on Document Analysis and Recognition (ICDAR).

[3] Baeza-Yates, R., Ribeiro-Neto, B. (1999). Modern Information Retrieval. ACM Press / Addison-Wesley.

[4] Brin, S., Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *In*: Proc. of World Wide Web Conf. (WWW).

[5] Castillo, C., Donato, D., Gionis, A., Murdock, V., Silvestri, F. (2007). Know your neighbors: Web spam detection using the web topology. In International ACM SIGIR, p. 423–430.

[6] Cha, M., Kwak, H., Rodriguez, P., Ahn, Y., Moon, S. (2007). I tube, you tube, everybody tubes: Analyzing the world's largest user generated content video system. *In*: Proc. of IMC.

[7] Dorogovtsev, S., Mendes, J. (2003). Evolution of Networks: from Biological Nets to the Internet and WWW. Oxford University Press.

[8] Fan, R., Chen, P., Lin, C. (2005). Working set selection using the second order information for training svm. *Journal of Machine Learning Research* (JMLR), 6, 1889–1918.

[9] Fetterly, D., Manasse, M., Najork, M.(2004). Spam, damn spam, and statistics: Using statistical analysis to locate spam web pages. *In*: Proc. of WebDB, 2004.

[10] Gill, P., Arlitt, M., Li, Z., Mahanti, A. (2007). Youtube traffic characterization: A view from the edge. *In*: Proc. of IMC.

[11] Gomes, L., Almeida, J., Almeida, V., Meira, W. (2007). Workload models of spam and legitimate e-mails. Performance Evaluation, 64, 690–714.

[12] Gomes, L., Castro, F., Almeida, V., Almeida, J., Almeida, R., Bettencourt. (2005). Improving spam detection based on structural similarity. *In*: Proc. of SRUTI.

[13] Gy¨ongyi, Z., Garcia-Molina, H., Pedersen. J. (2004). Combating web spam with trustrank. In International. Conf. on Very Large Data Bases, p. 576–587.

[14] Heymann, P., Koutrika, G., Garcia-Molina, H. (2007). Fighting spam on social web sites: A survey of approaches and future challenges. *IEEE Internet Computing*, 11 (6) 36–45.

[15] Jain, R. (1991). The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling. John Wiley and Sons, INC.

[16] Kleinberg, J., Hubs. (1999). authorities, and communities. ACM Computing Surveys, 31.

[17] Koutrika, G., Effendi, F., Gyongyi, Z., Heymann, P., Garcia-Molina, H. (2007). Combating spam in tagging systems. *In*: Proc. of AIRWeb.

[18] Shannon, M. (2007). Shaking hands, kissing babies, and...blogging? *Communications of the ACM*, 50.

[19] Thomason, A. (2007). Blog spam: A review. *In*: Proc. Of Conf. on Email and Anti-Spam (CEAS).

[20] Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y. (2005). Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research* (JMLR), 6, 1453–1484.

[21] Witten, I., Frank, E. (2005). Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann.

[22] Wu, C., Cheng, K., Zhu, Q., Wu, Y. (2005). Using visual features for anti-spam filtering. *In*: Proc. of IEEE International Conf. on Image Processing (ICIP).

[23] Yang, Y., Pedersen, J. (1997). A comparative study on feature selection in text categorization. *In*: Proc. of the International Conf. on Machine Learning (ICML).

[24] Yu, L., Liu, H. (2004). Redundancy based feature selection for microarray data. *In*: Proc. of ACM SIGKDD International Conf. on Knowledge Discovery and Data Mining (KDD).