Implementation Of Prototype To Detect Spam In YouTube Using The Application TubeKit And Naïve Bayes Algorithm

GRADUATE PROJECT REPORT

Submitted to the Faculty of
The School of Engineering & Computing Sciences
Texas A&M University-Corpus Christi
Corpus Christi, TX

in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Science

by

Srilaxmi Gandra
Fall 2014

Committee Members

Dr. Mario Garcia
Committee Chairperson

Dr. David Thomas
Committee Member
ABSTRACT

A lot of time of the internet user is invested on social network rather than search engines. Many of the business organizations and celebrities are taking advantage of this service by setting up social networking pages to improve direct communication with online users. Social media heavily relies on user-generated content which makes them incredibly powerful. In a very quick and effective way the information is passed across YouTube. However, YouTube network is prone to various types of unnecessary and malicious spammer activities. To maximize the popularity of a video, spammers post comments and video links, where the comment has irrelevant content of the subject being discussed. If spams are left unchecked, it attenuates resources sharing, intercommunication and openness. Hence there is a crucial need for security solution and a technique to combat video spamming in YouTube.

In this proposed system, the Naive Baye’s algorithm which is based on Baye’s Theorem is used. In this system, using Tube kit, data of YouTube videos are collected and manually categorizes it as legitimate or a spam video. Various attributes that could lead to spammers are explored using TubeKit. Later Microsoft SQL Server Data Mining Tools is used to determine if a video is legitimate or spam.
TABLE OF CONTENTS

Abstract ............................................................................................................................... 2
Table of Contents .................................................................................................................3
List of Figures .....................................................................................................................5
List of Tables ......................................................................................................................6
1. Background and Rationale ..........................................................................................7
   1.1 Statistics of YouTube ...........................................................................................7
   1.2 Impact of video spam ...........................................................................................9
   1.3 Related work ......................................................................................................10
      1.3.1 Video sharing Spammers and content Promoter detection .....................10
      1.3.2 Multi-View approach for detecting non-cooperative users ....................11
      1.3.3 Context-aware description for content filtering ......................................12
      1.3.4 Mining User Comment Activity .............................................................12
      1.3.5 One-class classifier approach ..................................................................13
2. Narrative......................................................................................................................14
   2.1 Proposed system ...............................................................................................14
      2.1.1 TubeKit Procedure ..................................................................................15
      2.1.2 Spammer Detection Mechanism .............................................................17
      2.1.3 Naive Bayes Model .................................................................................17
   2.2 Flow chart ........................................................................................................19
3. Implementation .........................................................................................................20
4. Testing and Evaluation .................................................................46
   4.1 Test Cases ........................................................................46
5. Conclusion and Future Work ....................................................54
Bibliography and References ..........................................................56
LIST OF FIGURES

Figure 2.1. Working Flow of Proposed .................................................................14
Figure 2.2. Scheme for Query Based YouTube Crawling .................................15
Figure 2.3. Flow Chart .......................................................................................19
Figure 3.1. TubeKit Setup ..................................................................................20
Figure 3.2. TubeKit Video Data .........................................................................21
Figure 3.3. Extracted Video Data ......................................................................22
Figure 3.4. TubeKit User Data ..........................................................................23
Figure 3.5. Extracted User Data ........................................................................24
Figure 3.6. Import and Export Wizard Setup Page ...........................................25
Figure 3.7. Import and Export Wizard Source Selection .................................26
Figure 3.8. Import and Export Wizard Destination ..........................................27
Figure 3.9. Import and Export Wizard Source Tables and Views .....................28
Figure 3.10. User Data load in Database .............................................................29
Figure 3.11. Video Data load in Database ............................................................30
Figure 3.12. User and Video Data View in Database ..........................................31
Figure 3.13. Visual Studio New Project ...............................................................32
Figure 3.14. Visual Studio New Project Details ..................................................33
Figure 3.15. Visual Studio Data Source Creation Complete .............................34
Figure 3.16. Visual Studio Data Source View Creation Complete .....................35
Figure 3.17. Visual Studio Mining Structure Algorithm selection .....................37
Figure 3.18. Visual Studio Mining Structure Training Data .............................38
Figure 3.20. Visual Studio Mining Structure Suggested columns.................................39
Figure 3.21. Visual Studio Mining Structure Test Set.........................................................40
Figure 3.22. Visual Studio Mining Structure Completion....................................................41
Figure 3.23. Visual Studio Mining Model Viewer process .................................................41
Figure 3.24. Visual Studio Mining Model Viewer process ....................................................42
Figure 3.25. Visual Studio Mining Model Attribute Profile.................................................43
Figure 3.26. Visual Studio Mining Model Attribute Characteristics.................................44
Figure 3.27. Visual Studio Mining Model Lift Chart............................................................45
Figure 4.1. Mining Model Viewer ....................................................................................47
Figure 4.2. User Data with HasSpamHint as True..............................................................48
Figure 4.3. User Data with HasSpamHint as False.............................................................49
Figure 4.4. Mining Model Viewer Avg ............................................................................50
Figure 4.5. User Data with Avg comment Less ...............................................................51
Figure 4.6. User Data with Avg comment Greater ............................................................51
Figure 4.7. User and Video Data with Category.................................................................52
Figure 4.8. Category Mining Details ................................................................................53
1. BACKGROUND AND RATIONALE

Social networking media such as Facebook, Flickr, twitter, and YouTube have gained huge following among internet users in the most recent decade. YouTube is one of the most popular and widely used video sharing websites on the Internet. Video especially is turning into the most important aspect of web user's daily routine. The reason being, video is the most usable platform to express our thoughts with others and is a platform to communicate with wide range of users, about the subject, related to educational tips, global issues, political disputes, business discussion etc. Such type of interaction allows the initiator and viewers to input additional context to a video to engage more viewers.

YouTube has a feature that allows its users to post comments in response to a particular video. The comment can include links to a particular video in response to a video. This type of response through a video is one of the most extensively used features of YouTube. Reports point out that the volume of content posted in YouTube in 60 days is equal to the data that would have been telecasted for 60 years by ABC, CBS and NBC conjointly without any hindrance [2]. On an observation, the most viewed and most discussed videos on an average have more than 1000 video responses which highlights the use and significance of the feature [3].

1.1 Statistics of YouTube

Below are the statistics about YouTube as of 1st OCT 2014 that shows the enormous popularity of YouTube [1].

- Over 1 Billion people use YouTube.
• Videos viewed on YouTube per day are over 4 Billion.

• Over 6 Billion hours of videos are watched per month on YouTube.

• 100 hours of videos are uploaded to YouTube per minute.

• On an average, the YouTube mobile video views per day are 1 Billion.

• 14.4% of Americans use YouTube during office hours.

• 50% of teenagers consider YouTube as their favorite website.

• Amount that is spent by advertisers on YouTube in 2013 is $5.6 billion.

• There is a growth of 13% among YouTube's mobile app user from Dec'13-May'14, thus shows the immense fame of YouTube on mobile devices.

• 50% of 2013 Inc 500 Companies uses YouTube.

The enormous popularity of YouTube and the less publication cost has led to misuse of the YouTube video feature since the spammers upload videos which have violated the copyrights, upload malicious and spam videos. Generally a spam is an irrelevant message passed via Internet, mostly to huge numbers of users, with the intention to advertise, phishing, spread malware, etc. The spammer’s activity mostly involves posting links of video to sites, on the comment area of videos or people’s profiles. In an observation it is pointed that among various social networking sites, Facebook and YouTube contain the most spam data. The spam ratio on Facebook or YouTube to the other media is 100 to 1 [9].
Below are the links for spam videos in YouTube.

i)  
https://www.youtube.com/all_comments?v=lWA2pjMjpBsii (last viewed on 09/27/2013)
This video is about a popular Hollywood song. Viewers posted their opinion about the song. Among them one of the viewer Sunil Lokumal posted a video link which is about his business which is irrelevant to the topic, hence is a spam.

ii)  
https://www.youtube.com/all_comments?v=52jQMfVAEEU (Last viewed on 09/27/2014)
This link is about a Bollywood movie song and text comments are given by the viewer’s regarding the song and there is a comment where the viewer advertises his business regarding astrology, which is irrelevant to the topic being posted, hence is a spam.

1.2 Impact of video spam:
Below mentioned are the effects of spam on YouTube.

- Since large volume of data flows on YouTube every second, existence of spam leads to bandwidth wastage (on user side) [5].
- As the data fetched is mostly repetitive or doesn’t suite with the choice of the user, the user may show disinterest to use YouTube. [4].
- Some of the data extracted is perhaps disturbing or posted to benefit from the user. Such type of data forces the system to comprise its main purpose of
improving social interaction, as the user losses trust and value of the legitimate application [4].

- Pornographic videos are posted as video responses to some of the videos which are meant for children. Such types of video responses are considered as spam and its presence has a negative effect on children [3].

- **Phishing:** Malicious users create phishing site misleading as a trusted site and share it among the users on internet. Upon clicking the link, sensitive data is asked to share and once the user inputs the data, the malicious users uses the shared data to access account and personal data.

- **Malware:** The malicious user creates malware and share among internet users. This link to the website will download malware without user’s interest. Once the user click the link, malware gets downloaded and malign user have the control of the device of the user.

1.3 Related Work:

The below are few of the methods or approaches considered to detect spam in YouTube.

1.3.1. Video sharing Spammers and content Promoter detection [2]

In this paper the solution for detecting online video sharing spammers and content promoters is discussed. Spammers are the one who post irrelevant videos as response to the popular videos while promoters post large number of videos to a particular video to gain popularity of that video and be considered in top rated category. In this approach a test set is created manually where the actual users of YouTube are considered as input and segregate the users into spammers, legitimate users and promoters. Characterization
of content, social attributes and user profile are provided so as to classify each user. For automatic identification of promoters and spammers a supervised classification algorithm is used. This approach was able to correctly identify maximum number of promoters and many spammers, miscalculating a very few number of legitimate users.

**Disadvantage:** Research found that the cost involved to manually label the data is huge.

### 1.3.2. Multi-View approach for detecting non-cooperative users [6]

The proposed method uses a multi-view semi-supervised classification algorithm, which considers small amount of training data than previous work of supervised methods. In this research, two approaches are evaluated which are designed considering two different strategies for blending results from various view, considering sample of preclassified users and user behavior attributes. When compared with the supervised approach, the paper combines the Borda Count view and View Agreement plan and yields more positive results in identifying non-cooperative users and minimizes the training data, at the cost of small fraction of increase in miscalculating legitimate users. Since the paper focuses on non-collaborative users, promoters have more predictable behavior and obtain high detection rates than spammers with true/false rate as 0.56/0.08.

**Disadvantage:** The work mentioned in section 1.3.1 and 1.3.2 focuses on the classification of users, while noticing that there are varied set of behaviors, from heavy spammers to one time spammers. Such variation affects the classification accuracy.
1.3.3. **Context-aware description for content filtering** [8]

The work mentioned in section 1.3.1 and 1.3.2 focuses on the classification of users, while noticing that there are varied set of behaviors, from heavy spammers to on time spammers. Such variation affects the classification accuracy. This approach emphasizes videos instead of the users which will be considered as fundamental source of evidence, their visual content. Using context-aware, the detection is improved when compared to the bag-of-visual model thus allowing one to consider context of the video in the representation. This model is examined using two video dataset which yield better results.

1.3.4. **Mining User Comment Activity** [7]

This approach is used to automatically detect YouTube comment spammers by mining the user’s comment activity log and fetch pattern which help in identifying spam behavior. Various comment attributes like comment text, VideoID of the video for which comment is passed, timestamp and binary variable value HasSpamHint are considered and empirical analysis is carried on. Attributes such as existence of vast number of exactly like comment in one or across various videos, short time intervals between consecutive comments and huge percentage of comments with spam hint tag are definite indicators for classifying YouTube spammers. This approach is robust in detecting spammers.
1.3.5. One-class classifier approach [3]

In this approach, the problem of video response spam is divided into three subproblems: pornographic video response detection, promotional or commercial video response detection, and botnet video response detection. It collects positive class training dataset to train the classifier then work on characterization and identification of discriminator feature. Weighted similarity function is considered to calculate resemblance between objects in training dataset and target class object. Precision and recall, which are the standard information retrieval metric, are used to evaluate the performance of these three independent classifiers and outputs if the object is in the target class or is unknown.

Disadvantage: This approach yields 80% accuracy and not 100% due to the below mentioned reasons:

1) Since the size of the training dataset is less and is a time consuming and challenging process as manual work involved is more to gather spam video and to label.

2) Description not available on few spam video thus text classification methods can’t be used on it.

3) Some spam videos are tagged as educational, health etc. which are totally irrelevant to the video type.
2. NARRATIVE

The objective of this research is to design a framework to identify the legitimate and spammers of systems related to video sharing. The approach used in this system is to identify spammers by categorizing users, the user videos, and depend on a group of attributes related with the user activity and social reaction in the system as well as the video attributes. In this system we are using tools to accomplish the task of preventing the video spamming in YouTube such as TubeKit, Microsoft SQL Import and Export, Microsoft SQL Server Data Tools and Microsoft Office Excel.

2.1 Proposed system:

![Image of Working Flow of Proposed System]

In the proposed systems, the first step involved is Crawling strategy. Using TubeKit the Crawling operation is accomplished which is explained below. We collect the data of YouTube video’s and manually categorize as legitimate videos or spam by using TubeKit. This task is accomplished by detecting the spammers by categorizing
users, the user videos, and depend on a group of attributes related with the user activity and social reaction in the system as well as the video attributes. There are many attributes related with the user action such as title, Average Comment Count, View Count, Subscriber Count etc. The user creeps through a large data set related with the user from YouTube which generally causes huge internet congestion.

2.1.1 TubeKit Procedure:

TubeKit is used for designing custom-built YouTube crawlers. The user can design his own crawler by using TubeKit, since TubeKit allows creeping through YouTube based on a group of seed queries and helps the user to gather up to 16 various attributes. The working of crawler is based on PHP program which runs on the web server. It undergoes several phases as shown in figure 2.2.

![Figure 2.2: Scheme for query based YouTube crawling](image)
The below mentioned are the steps involved in the crawling process.

- The user provides a group of seed queries to supervise.
- Searching activity on YouTube takes place using these queries,
- From the group of the results that are resulted from YouTube, a group of metadata is fetched. The metadata would be the information about the video which the owner of the video has provided which remains static, for example, the video genre.
- The video downloader component looks at the metadata table to check which videos have not been downloaded in the previous extraction and collects those videos from the YouTube in ash format.
- The video converter will analyze the downloaded video which are not converted and then converts those videos into mpeg style.
- The context capturing component looks at YouTube and fetches several contextual related data about the video for the metadata which is gathered already. A time-stamp is noted for each time a social context is fetched. A social context is the data that a visitor contributes to a video page; this includes fields such as comments and ratings. This component fetches the temporal context since it runs repeatedly and refreshes time-sensitive information like new comments or video postings.

The data that is extracted is loaded into Microsoft Excel 2010 for both Video and User. Some of the attributes that are identified using Crawlers for User are Username,
duration, category, video url, video count, view count and comment count (refer Figure 3.7) and for Video are Title, Author, Category and Duration etc. (refer Figure 3.5).

2.1.2 Spammer Detection Mechanism:

From the excel file that we created, we connect file to Microsoft SQL Import and Export and load the extracted data in a database of Microsoft SQL Management studio 2014. Then data mining approach are carried out using tools related to SQL Server. For the purpose of setting up the data mining method, a new Analysis Service database has to be created and a data source, data source view has been built. Once the data source from the excel file is extracted, a data source viewer has to be built to look at the tables and views of the database. Naive Bayes data-mining technique is applied to know and predict the behavior of a video.

2.1.3 Naive Bayes Model

The Naive Bayes algorithm is depended on Bayes’ theorems is a classification algorithm and given by services of Microsoft SQL Server Analysis for benefiting in predicting mining models [13]. The word Naive evolved from the reality that the algorithm considers Bayesian approach but doesn’t considers the dependences if any.

When compared with other Microsoft algorithms, this algorithm is less intense in terms of calculating, and hence is favorable for quickly developing mining designs to find out the association between input and predictable columns. Data can be initially analyzed using the algorithm and the results that are obtained can be used to build additional models using various algorithms that provides more precise and more powerful in terms of calculation.
Let ‘X’ be the data sample or tuple and it is represented by attribute vector \( X=(x_1,x_2,\ldots,x_n) \) and let there be ‘m’ classes \( C_1,C_2,\ldots,C_m \). The assumption considered is the attributes are conditionally independent i.e. no dependence relation between attributes. To find if the tuple belongs to a particular class, the naïve bayes algorithm is given by

\[
P(X \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i) = P(x_1 \mid C_i) \times P(x_2 \mid C_i) \times \ldots \times P(x_n \mid C_i)
\]

\[
P(X \mid C_i) = P(C_i) \times \prod_{k=1}^{n} P(x_k \mid C_i)
\]

Here the algorithm first tries to find the probability of the predictable class.

If the predictable class is true then \( P(C_i) = \) (No.of ways the predictable class is ‘true’)/ Total no. of outcomes.  

If the predictable class is false then \( P(C_i) = \) (No.of ways the predictable class is ‘False’)/ Total no. of outcomes.

Once the probability of the predictable class is found, the algorithm tries to find the probability of each attribute for various states against the predictable class (‘true/false').

\( P(x_1 \mid C_i) = \) (No.of ways \( x_1 \) happens)/Total outcome for the predictable class as ‘true’.

\( P(x_1 \mid C_i) = \) (No.of ways \( x_1 \) happens)/Total outcome for the predictable class as ‘false’.

This computation is done for all the attributes in the vector.

Finally to check if the tuple is spam or not, the calculated values for the attributes are substituted in the given formula ‘A’ for predictable class as true and false. For which ever class the probability is maximum, the algorithm picks and suggests that the tuple is spam or not.
2.2 Flow chart:

Figure 2.3 describes the flow of execution of the proposed system.

Figure 2.3: Flow Chart
3. IMPLEMENTATION

The User data and the Video data of YouTube are extracted using TubeKit first. Figure 3.1 gives an idea of TubeKit setup where the user define the project name and the destination folder in which all the files created dynamically and stored into it. In this page, the user specifies the database name which should be existing already and its username and password. After defining all the details of the project and database, the tables were generated in the database in which all the attributes information stored and go through the crawling phase.

![Figure 3.1: TubeKit Setup](image)
Once a user selects the video and clicks on the ‘Video Data’ tab as highlighted in the figure 3.2 he gets the metadata like Author, Ratings and comments etc. of that particular video.

Figure 3.2: TubeKit Video Data
To load the extracted video data, TubeKit provides facility to load it as Text, CSV or XML, the user has to select the option according to his convenience. In this project the data is loaded as CSV file and figure 3.3 gives the information about the loaded data.

Figure 3.3: Extracted Video Data
Once the video is opened, out of the multiple users who commented, select a particular user id and then click on the ‘User Data’ tab to get the metadata like Username, Firstname and Last name etc. of that particular user as mentioned in figure 3.4.

Figure 3.4: TubeKit User Data
To load the extracted user data, TubeKit provides facilities to load it as Text, CSV or XML, the user has to select the option according to his convenience. In this project the data is loaded as CSV file and figure 3.5 gives the information about the loaded data.

Figure 3.5: Extracted User Data
Once the user data and video data are extracted in the Microsoft Excel sheet, the user needs to load the data into the database using Microsoft SQL Import and Export Wizard. Figure 3.6 is the set up page of the wizard.

Figure 3.6: Import and Export Wizard setup page
Figure 3.7 provides details to select the source that has to be loaded. Here the source is Microsoft Excel and by using the drop down box in the Data source the user selects the source type and browses the location of the file.

Figure 3.7: Import and Export Wizard Source Selection
Figure 3.8 gives details about the destination selection. It provides various options like Flat File Destination, Microsoft Excel, .Net Framework Data Provider for Odbc etc. In this project the Destination considered is ‘SQL Server Native Client 1.0’, the server name ‘Sneha’ and the database selected is ‘YoutubeDataCollection’.

Figure 3.8: Import and Export Wizard Destination
Figure 3.9 gives details about the Source Tables and Views selection. Here the user needs to check a sheet and select the required table name, here the tables considered are of user Data and Video Data.

Figure 3.9: Import and Export Wizard Source Tables and Views

Once the tables are selected as mentioned in Figure 3.9, click on ‘Finish’ to complete the process to load data into database.
Upon clicking the ‘Finish’, the loading process is complete and the user needs to check the data that is loaded into the database. Open Microsoft SQL 2014 Management Studio and then connect to the server i.e. ‘Sneha’ and select the database which is ‘YoutubeDataCollection’. Expand the Tables in the selected database to see the tables that are considered i.e. user table and video table. Select User table and execute the query to select the rows in the table. Figure 3.10 ensures that the user table is created and data is loaded into that.

Figure 3.10: User Data Load in Database
Select the Video Data table and execute the query to see the rows that are in the table of video. Figure 3.11 ensures that the video table is created and data is loaded.

Figure 3.11: Video Data Load in Database
The data for the User and Video when is available on the database, create a view based on the User and Video data. The view data in the database is shown in figure 3.12.

Figure 3.12: User and Video Data View in Database
Once the user data and video data both are loaded into the Database, mining operation is performed on that. To do this use Microsoft Visual Studio. First create a project in the visual studio. Figure 3.13 gives an idea to create a new project by selecting file from the menu bar and then selecting ‘New’ and then ‘Project’.

Figure 3.13: Visual Studio New Project
Details for the new project are given in Figure 3.14. The user has to select ‘Business Intelligence’ from the Templates, and then select ‘Analysis Services multidimensional and Data Mining’ and name the project, here the project is named as ‘YouTubeSpamming’.

Figure 3.14: Visual Studio New Project Details
Once the new project is created, the user needs to create a new Data source. Right click on a data source in the right pane, and a data source creation wizard popup and the user needs to mention the source of database connection and then the data source name as mentioned in figure 3.15 to complete the process of Data source creation.

Figure 3.15: Visual Studio Data Source Creation Complete
The user has to create a new data source view by right click on Data source view option on the right side pane. After selecting new data source view, a Data Source view wizard popup and then select the tables and include those tables as views by clicking ‘>’ as mentioned in Figure 3.16

Figure 3.16: Visual Studio Data source View Creation Details
Figure 3.17 represents the completion wizard for Data source view and when click ‘Finish’ the data source view creation is complete.

Figure 3.17: Visual Studio Data Source View Creation Complete
The user needs to right click on the Data mining Structure and then select ‘New Data Mining structure. In the Data mining wizard as mentioned in Figure 3.18 select the type of data mining algorithm to be considered. In this project ‘Naïve Bayes’ algorithm is selected.

Figure 3.18: Visual Studio Mining Structure Algorithm Selection
Figure 3.19 below specifies the training data to consider. The user needs to select the predictable column and key column and click ‘suggest’. Predictable column can be the column which helps us to identify spam more effectively. Here the predictable column considered is ‘HasSpamHint’ and ‘Username’ as key column.

Figure 3.19: Visual Studio Mining Structure Training Data
Once the ‘suggest’ tab is selected, for the predictable column the suggested input columns are given out. Here for the predictable column i.e. ‘HasSpamHint’ the suggested columns are listed out in figure 3.20.

Figure 3.20: Visual Studio Mining Structure Suggested Columns
Figure 3.21 helps to select the percentage of the testing data to be considered. The default is ‘30%’ and in this project default percentage is considered.

Figure 3.21: Visual Studio Mining Structure Test Set

After test set is mentioned as in figure 3.21 , name the mining structure to close the completion process of mining structure creation and figure 3.22 gives the view of the create mining structure.
Figure 3.22: Visual Studio Mining Structure Completion

Figure 3.23: Visual Studio Mining Model Viewer process

Figure 3.23 gives the information about completion of processing.
The figure 3.24 is about the mining model viewer. This viewer gives data about how the input and predictable columns are associated. In figure 3.24 the predictable column i.e. HasSpamHint is at the center and surrounded by the suggested input columns for the predictable column.

Figure 3.24: Visual Studio Mining Model Viewer process

Figure 3.25 gives information of the attribute profile. Attribute profile visually shows how the algorithm distributes data. The algorithm distributes the data of each state of input column, for a given state of each predictable column. Here for the suggested columns the data information is given out.
Figure 3.25: Visual Studio Mining Model Attribute Profile

Figure 3.26 gives information about the probability of each state of the input column for a given state of the predictable column. The algorithm picks the maximum probability of each attribute for the given state of predictable column. Thus help in selecting patterns through which the user can predict the status about the pattern if it occurs in future.
Figure 3.27 gives information about Lift chart. Lift chart graphically represents the improvement that a mining model provides when compared against a random guess and measures in terms of a lift score. Figure 3.27 shows that at 51.61% of the total population Naïve Bayes has a predict probability of 97.40%.
Figure 3.27: Visual Studio Mining Model Lift Chart
4. Testing and Evaluation:

Testing is an examination performed on a product or service and provides the information regarding its quality to the user. In general it is used to execute a program or application with the aim of discovering software bugs (errors or defects). The main intention of testing is to identify failures in software so that defects may be identified and redressed. Testing can be carried when executable software (regardless of the fact that part of it is finish) exists.

For testing the system some users and videos from YouTube were considered, these examples are both legitimate and spam. Using TubeKit the metadata is extracted and contextual features for this data from YouTube. This extracted data is loaded into the database on which Naïve Bayes algorithm is implemented. Considering the results of predicting and input columns, various patterns are found out, which will help in identifying the spam.

4.1 Test Cases:

Below are few of the test cases for various predictable columns of user data, video data and combination of user and video data. The patterns which are resulted will help us to predict spam.
Test case 1:

The user data is considered in this scenario and the predictable column that is selected is ‘HasSpamHint’, Figure 4.1 gives the Mining Model view which represents the relationship between input column and predictable column.

Since the SQL Server Data Tool is used and has the naïve bayes logic implemented in it and the tool is in production, the resulted probability values will be correct. The user has to pick the pattern which has the highest probability for different attributes. Figure 4.2 shows the probability that is given by naïve bayes algorithm for a video being spam with combination of various attributes. The results are for the User Data with predictable column ’HasSpamHint’ as ‘True’. Here if Video watch count >= 34, Average Comment Count between 4.84-7.68 and View Count < 6.25(as they hold
high probability values) then this pattern is referred to as spam, so if in future such pattern occurs then the user can predict it as spam.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Values</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video Watch Count</td>
<td>&gt;= 34</td>
<td>81.250%</td>
</tr>
<tr>
<td>Subscriber Count</td>
<td>5.3933949064 - 8.2328814064</td>
<td>65.625%</td>
</tr>
<tr>
<td>Average Comment Count</td>
<td>4.8433949064 - 7.6828814072</td>
<td>65.625%</td>
</tr>
<tr>
<td>View Count</td>
<td>&lt; 6.255490064</td>
<td>65.625%</td>
</tr>
<tr>
<td>Age</td>
<td>22 – 23</td>
<td>40.625%</td>
</tr>
<tr>
<td>Subscriber Count</td>
<td>&lt; 5.3933949064</td>
<td>34.375%</td>
</tr>
<tr>
<td>Average Comment Count</td>
<td>&lt; 4.8433949064</td>
<td>34.375%</td>
</tr>
<tr>
<td>Age</td>
<td>&lt; 15</td>
<td>31.250%</td>
</tr>
<tr>
<td>Age</td>
<td>23 – 34</td>
<td>28.125%</td>
</tr>
<tr>
<td>Video Watch Count</td>
<td>23 – 34</td>
<td>18.750%</td>
</tr>
<tr>
<td>View Count</td>
<td>9.3083874992 - 24.5714285696</td>
<td>18.750%</td>
</tr>
<tr>
<td>View Count</td>
<td>24.5714285696 – 39</td>
<td>9.375%</td>
</tr>
<tr>
<td>View Count</td>
<td>&gt;= 39</td>
<td>6.250%</td>
</tr>
</tbody>
</table>

Figure 4.2: User Data with HasSpamHint as True

**Test Case 2:**

Figure 4.3 shows the probability of a video being spam with combination of various attributes. The results are for the User Data with predictable column 'HasSpamHint' as ‘False’. Here if Video watch count between 6-23, Average Comment Count between 9.155-11.684 and View Count between 66.25 -9.30 (as they hold high probability values) then this pattern is referred to as spam, so if in future such pattern occurs then the user can predict it as spam.
<table>
<thead>
<tr>
<th>Attributes</th>
<th>Values</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video Watch Count</td>
<td>6 – 23</td>
<td>61.905%</td>
</tr>
<tr>
<td>View Count</td>
<td>6.255490064 - 9.3083874992</td>
<td>50.000%</td>
</tr>
<tr>
<td>Average Comment Count</td>
<td>9.155117056 - 11.6845896784</td>
<td>50.000%</td>
</tr>
<tr>
<td>Subscriber Count</td>
<td>9.705117056 - 12.2345896784</td>
<td>50.000%</td>
</tr>
<tr>
<td>Age</td>
<td>&gt;= 34</td>
<td>40.476%</td>
</tr>
<tr>
<td>Age</td>
<td>15 – 22</td>
<td>30.952%</td>
</tr>
<tr>
<td>Age</td>
<td>23 – 34</td>
<td>28.571%</td>
</tr>
<tr>
<td>Subscriber Count</td>
<td>8.2328814064 - 9.705117056</td>
<td>26.190%</td>
</tr>
<tr>
<td>Average Comment Count</td>
<td>7.6828814072 - 9.155117056</td>
<td>26.190%</td>
</tr>
<tr>
<td>View Count</td>
<td>&lt; 6.255490064</td>
<td>26.190%</td>
</tr>
<tr>
<td>Subscriber Count</td>
<td>&gt;= 12.2345896784</td>
<td>23.810%</td>
</tr>
<tr>
<td>Average Comment Count</td>
<td>&gt;= 11.6845896784</td>
<td>23.810%</td>
</tr>
<tr>
<td>View Count</td>
<td>9.3083874992 - 24.5714285696</td>
<td>23.810%</td>
</tr>
<tr>
<td>Video Watch Count</td>
<td>&lt; 4</td>
<td>19.048%</td>
</tr>
<tr>
<td>Video Watch Count</td>
<td>4 – 6</td>
<td>19.048%</td>
</tr>
</tbody>
</table>

Figure 4.3: User Data with HasSpamHint as False

**Test Case 3:**

The user data is considered in this scenario and the predictable column that is selected is ‘Average Comment Count’, Figure 4.4 gives the Mining Model view.
Figure 4.4: Mining Model Viewer Avg

The Figure 4.5, 4.6 shows the probability of a video being spam with combination of various attributes. The results are for the User Data with predictable column = 'AverageCommentCount' with lesser and greater proportion.

Lesser proportion:

Here if Video watch count between 23-34, HasSpamHint as True, and View Count between 9.308-24.57 (as they hold high probability values) then this pattern is referred to as spam, so if in future such pattern occurs then the user can predict it as spam.
<table>
<thead>
<tr>
<th>Attributes</th>
<th>Values</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subscriber Count</td>
<td>&lt; 5.3933949064</td>
<td>100.000%</td>
</tr>
<tr>
<td>Has Spam Hint</td>
<td>True</td>
<td>100.000%</td>
</tr>
<tr>
<td>Age</td>
<td>22 – 23</td>
<td>64.286%</td>
</tr>
<tr>
<td>View Count</td>
<td>9.3083874992 - 24.5714285696</td>
<td>64.286%</td>
</tr>
<tr>
<td>Video Watch Count</td>
<td>23 – 34</td>
<td>64.286%</td>
</tr>
<tr>
<td>Video Watch Count</td>
<td>&gt;= 34</td>
<td>35.714%</td>
</tr>
<tr>
<td>Age</td>
<td>23 – 34</td>
<td>21.429%</td>
</tr>
<tr>
<td>View Count</td>
<td>&gt;= 39</td>
<td>21.429%</td>
</tr>
<tr>
<td>Age</td>
<td>&lt; 15</td>
<td>14.286%</td>
</tr>
<tr>
<td>View Count</td>
<td>24.5714285696 – 39</td>
<td>14.286%</td>
</tr>
</tbody>
</table>

**Figure 4.5: User Data with Avg comment Less**

**Greater Proportion:**

Here if Video watch count between 6-23, HasSpamHint as True, and View Count between 6.255-9.30 (as they hold high probability values) then this pattern is referred to as spam, so if in future such pattern occurs then the user can predict it as spam.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Values</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subscriber Count</td>
<td>9.705117056 - 12.2345896784</td>
<td>100.000%</td>
</tr>
<tr>
<td>Has Spam Hint</td>
<td>False</td>
<td>100.000%</td>
</tr>
<tr>
<td>View Count</td>
<td>6.255490064 - 9.3083874992</td>
<td>100.000%</td>
</tr>
<tr>
<td>Age</td>
<td>15 – 22</td>
<td>60.000%</td>
</tr>
<tr>
<td>Video Watch Count</td>
<td>6 – 23</td>
<td>60.000%</td>
</tr>
<tr>
<td>Age</td>
<td>&gt;= 34</td>
<td>40.000%</td>
</tr>
<tr>
<td>Video Watch Count</td>
<td>&lt; 4</td>
<td>40.000%</td>
</tr>
</tbody>
</table>

**Figure 4.6: User Data with Avg comment Greater**
Test Case 4:

The Figure 4.7 shows the probability of a video being spam with combination of various attributes. The results are for the Video Data combine with User Data considering predictable column ‘Category’ which predict category contains large number of spam data.

Figure 4.7: User And Video Data with Category

For Category = ‘Entertainment’

Here if Average Comment Count between 4.19-8.75, HasSpamHint = False and View count between 7-9 (as they hold high probability values) then this pattern is referred to as spam, so if in future such pattern occurs then the user can predict it as spam.
<table>
<thead>
<tr>
<th>Attributes</th>
<th>Values</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Views</td>
<td>&gt;= 971127</td>
<td>100.000%</td>
</tr>
<tr>
<td>Age</td>
<td>&lt; 20.9501617824</td>
<td>100.000%</td>
</tr>
<tr>
<td>Has Spam Hint</td>
<td>False</td>
<td>100.000%</td>
</tr>
<tr>
<td>Video Watch Count</td>
<td>&lt; 7.2924889616</td>
<td>100.000%</td>
</tr>
<tr>
<td>Average Comment Count</td>
<td>4.1927603688 - 8.7590978672</td>
<td>50.000%</td>
</tr>
<tr>
<td>Average Comment Count</td>
<td>8.7590978672 - 10.2557597184</td>
<td>50.000%</td>
</tr>
<tr>
<td>Avg# Rating</td>
<td>4.6572146664 - 4.760315896</td>
<td>50.000%</td>
</tr>
<tr>
<td>Avg# Rating</td>
<td>4.5575325232 - 4.6572146664</td>
<td>50.000%</td>
</tr>
<tr>
<td>Comments</td>
<td>58 – 303</td>
<td>50.000%</td>
</tr>
<tr>
<td>Comments</td>
<td>&gt;= 303</td>
<td>50.000%</td>
</tr>
<tr>
<td>Subscriber Count</td>
<td>8.6492854864 - 11.7417266976</td>
<td>50.000%</td>
</tr>
<tr>
<td>Subscriber Count</td>
<td>11.7417266976 - 20.5785600832</td>
<td>50.000%</td>
</tr>
<tr>
<td>Ratings</td>
<td>208 – 2093</td>
<td>50.000%</td>
</tr>
<tr>
<td>Ratings</td>
<td>&gt;= 2093</td>
<td>50.000%</td>
</tr>
<tr>
<td>View Count</td>
<td>7 – 9</td>
<td>50.000%</td>
</tr>
<tr>
<td>View Count</td>
<td>&lt; 6</td>
<td>50.000%</td>
</tr>
</tbody>
</table>

Figure 4.8: Category Mining Details
5. CONCLUSIONS AND FUTURE WORK

YouTube being one of the most popular media in the social networking site is attracted by many of the malicious users. Spam if unchecked has undesirable effects such as consumption of computing resources, degrading the reputation of the targeted legitimate web application, affects the search engine ranking and misguides genuine usage of legitimate users. Thus the proposed approach is used to detect the spam in YouTube using a set of their attributes and of their contributed videos. For the selected attributes using Naïve Bayes algorithm we will find out various patterns that will help in predicting the spam.

In this research, test case 1 is to find the pattern of spam for user data where predictable column selected is ‘HasSpamHint’ as ‘True’ and conclusion was that with various states of the input columns for the predictable column as ‘true’, the pattern with various range and its probability is given out.

In Test case 2 the predictable column selected is ‘HasSpamHint’ as ‘False’ and conclusion was that with various states of the input columns for the given predictable column the pattern with various range and its probability is given out.

In Test case 3 user data is considered and the predictable column selected is ‘Average Comment Count’, this is tested considering less and greater proportion of Average Comment Count and conclusion was that with various state of the input columns for the given predictable column the pattern with various ranges and its probability is given out.
In Test case 4 user as well as video data is considered and the predictable column selected is ‘Category’ and the conclusion is that the given category as Entertainment the spam pattern and its probabilities are given out.

After testing various test cases I conclude that the proposed approach will be able to detect YouTube Spam by using the patterns that are generated.

In future we would consider content based features of video like pixel, HD quality etc. to detect spam in YouTube.


   Media-Spam-Research-Report.pdf

    Proceedings of YouTube and 2008 Election Cycle in the United States, Amherst,


[12] Microsoft SQL Server 12 Analytics Service,