# MODELING CONTENT LIFESPAN IN ONLINE SOCIAL NETWORKS USING DATA MINING

BY

JOHN GIBBONS

Submitted to the graduate degree program in Electrical Engineering and Computer Science and the Graduate Faculty of the University of Kansas in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Chairperson Dr. Arvin Agah

Dr. Perry Alexander

Dr. Jerzy Grzymala-Busse

Dr. James Miller

*Dr. Prajna Dhar

Date Defended: 2014-02-03

The Dissertation Committee for JOHN GIBBONS certifies that this is the approved version of the following dissertation:

# MODELING CONTENT LIFESPAN IN ONLINE SOCIAL NETWORKS USING DATA MINING

Chairperson Dr. Arvin Agah

Date approved: 2014-02-03

## Acknowledgements

I would like to thank Dr. Arvin Agah. This dissertation would not have been possible without his guidance, advice, lead, and seemingly endless patience during this process. Because of Dr. Agah I can now look back on the last seven years and know that I have greatly improved as a student, teacher, and person. I also want to thank Dr. Alexander, Dr. Miller, and Dr. Jerzy for being outstanding role models in teaching. Thanks to Dr. Dhar for her help and outside expertise. Last, but certainly not least, thanks to Julia my lovely, Canadian counter-part for years of guidance, proof reading, and support and to Louie the Dog for standing, sitting, and sleeping by my side.

## Abstract

Online Social Networks (OSNs) are integrated into business, entertainment, politics, and education; they are integrated into nearly every facet of our everyday lives. They have played essential roles in milestones for humanity, such as the social revolutions in certain countries, to more day-to-day activities, such as streaming entertaining or educational materials. Not surprisingly, social networks are the subject of study, not only for computer scientists, but also for economists, sociologists, political scientists, and psychologists, among others. In this dissertation, we build a model that is used to classify content on the OSNs of Reddit, 4chan, Flickr, and YouTube according the types of lifespan their content have and the popularity tiers that the content reaches. The proposed model is evaluated using 10-fold cross-validation, using data mining techniques of Sequential Minimal Optimization (SMO), which is a support vector machine algorithm, Decision Table, Naïve Bayes, and Random Forest. The run times and accuracies are compared across OSNs, models, and data mining algorithms.

The peak/death category of Reddit content can be classified with 64% accuracy. The peak/death category of 4Chan content can be classified with 76% accuracy. The peak/death category of Flickr content can classified with 65% accuracy. We also used 10-fold cross-validation to measure the accuracy in which the popularity tier of content can be classified. The popularity tier of content on Reddit can be classified with 84% accuracy. The popularity tier of content on 4chan can be classified with 70% accuracy. The popularity tier of content on Flickr can be classified with 66% accuracy. The popularity tier of content on YouTube can be classified with only 48% accuracy.

Our experiments compared the runtimes and accuracy of SMO, Naïve Bayes, Decision Table, and Random Forest to classify the lifespan of content on Reddit, 4chan, and Flickr as well as classify the popularity tier of content on Reddit, 4chan, Flickr, and YouTube. The experimental results indicate that SMO is capable of outperforming the other algorithms in runtime across all OSNs. Decision Table has the longest observed runtimes, failing to complete analysis before system crashes in some cases. The

statistical analysis indicates, with 95% confidence, there is no statistically significant difference in accuracy between the algorithms across all OSNs. Reddit content was shown, with 95% confidence, to be the OSN least likely to be misclassified. All other OSNs, were shown to have no statistically significant difference in terms of their content being more or less likely to be misclassified when compared pairwise with each other.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1 Introduction

Online Social Networks (OSNs) are integrated into business, entertainment, politics, and education; they are integrated into nearly every facet of our everyday lives. They have played essential roles in milestones for humanity, such as the social revolutions in certain countries, to more day-to-day activities, such as streaming entertaining or educational materials. Not surprisingly, social networks are the subject of study, not only for computer scientists, but also for economists, sociologists, political scientists, and psychologists. The number of people on online social networks has reached a staggering level, servicing billions of people. Table 1.1 lists the population of some of the most popular social networks.

| Social Network | Population | Content Type |
|---|---|---|
| **YouTube (Elliot, 2011)** | 300 million registered | Video |
| **Flickr (Jefferies, 2013)** | 87 million registered | Images and Video |
| **Reddit (Reddit.com, 2013)** | 73 million visitors monthly<br>2.5 registered users | Links to external domains (image, videos, blogs) and text posts |
| **4chan (4chan.org, 2013)** | 25 million visitors monthly | Images |

**Table 1.1 Populations of OSNs**

Despite the growing size and influence of social networks, they all have a common feature: Content sharing. Each OSN allows for the posting, sharing, and interaction with content. Social Networks no longer involve only the interaction with another person, but peoples' interaction with content in the form of liking, favoriting, +1-ing, upvoting or sharing. There are wide ranges of measureable data that revolves around content activities.

It is observed that content does not stay active forever. The content about one's daily activities, country-wide election, or even events that are influential on a world-wide scale, eventually fade out of the public eye. The study of this topic is known as Aging Theory (C. Chen 2012). These works include determining the lifespan of topics in a social network using a custom aging theory algorithm and analyzing the lifespan of content to detect real world events (Cataldi 2010, Sakaki *et al.* 2010). Content on OSNs demonstrates having a lifespan that can be measured and this dissertation experiments with models that attempt to accurate capture the lifespan of content on OSNs.

## 1.1 Research Hypothesis

Applying data mining techniques to data collected from online social networks, a model can be produced that can categorize the type of lifespan and popularity range of content on a social network, which can then be used predictively.

## 1.2 Dissertation Organization

In this dissertation we propose a system capable of extracting data about content from OSNs, analyzing the data with data mining techniques, using a model for content lifespan and popularity tiers, and then comparing the classifiers based on runtime and accuracy. The dissertation is organized into six chapters: Chapter 2 discusses the background and related work. Chapter 3 presents the research methodology. Chapter 4 defines the implementation and the evaluation of the proposed efforts. Chapter 5 includes the experimental results and discussion. Chapter 6 present the contributions, limitations, and future work.

# Chapter 2 Background and Related Work

This chapter provides a description of background and relevant research, and covers the OSNs chosen for research.

Social network research spans a wide variety of areas, some of which having roots that are decades old. Current online social network research topics include:

- Personal Privacy and Information Security (Adams 2011, Krishnamurthy and Wills 2008)
- Community Discovery (Adams, 2010, Allen, 2005, Backstrom *et. al.*, 2006, Cheng *et al.*, 2013, Matsuo and Yamamoto, 2009)
- Alternative Methods for Content Sharing (Gibbons and Agah 2012)
- Identity Discovery Across Multiple Networks (Fard and Ester, 2009, Henr, 2008, Sousa 2009, Stewart 2009,Vosecky *et al.*, 2009)
- Influence Discovery (Wilson 2009, Xu and Lu, 2010)

A topic that has undergone little research, despite being one of the older social networking topics, is the area of the aging theory. The motivation for aging theory research is the observation that publicly shared content goes through a life cycle of activity. For example, a news story about a recent natural disaster initially has a large amount of relevance, being discussed and shared, but after enough time the relevance fades and the story, metaphorically, dies. It has been used in a variety of ways from detecting when new topics are "emerging" (Cataldi *et al.*, 2010) to detecting real world events in real time like detecting earthquakes using only Twitter feeds (Sakaki *et al.*, 2010).

## 2.1 Aging Theory

Aging theory research can be traced back to the field of event detection in online news sources. Allan *et al*. (1998) began developing methods to classify news stories apart as of a current event or a brand new

event. The research was expanded on by Cataldi *et al*. (2010) while studying Twitter.com. Cataldi *et al.* (2010) applied the methods proposed by Allan *et al.* (1998) in order to detect topics in a stream of "tweets" (individual pieces of content from Twitter). Cataldi *et al.* (2010) also applied the life cycle model introduced by Chen *et al*. (2003), which drew analogies between the life cycle of an event and the life cycle of a living thing. Events were detected using clustering algorithms on news articles, and each event had measurable traits that directly influenced its lifespan. These traits are detailed in Table 2.1.

| Trait | Definition |
|---|---|
| **Nutrition** | Contribution of a piece of content to the overall energy of the event |
| **Energy** | The liveliness of an event in its lifespan |
| **Growth** | The increase in energy through nutrition |
| **Decay** | The natural decrease in energy over time |

**Table 2.1 Chen *et al.* (2003) Aging Theory Model Definitions.**

The lifespan was determined by the amount of news articles pertaining to a specific topic (energy), the relevance of each article to the topic (nutrition), and amount of decay set by the user.

One issue with the research of Chen *et al.* (2003) is that many of the traits depend upon user-defined parameters. For example, the decay rate of events must be decided upon either arbitrarily or through testing before experimentation can begin. If a decay rate is too high, then topics will be considered "dead" prematurely, or if the decay rate is too low all topics will be considered to be active for too long.

Cataldi *et al.* (2010) namely, instead of focusing the research on any detectable news event, focused on a specific platform, Twitter.com. It was possible to use the traits of Twitter's infrastructure to more

accurately predict lifespan. For example, they introduced the notion of authority, which can be summarized as a measure of the influence of a content that author has based upon the sum of the authorities of his or her followers (i.e., subscribers). Authority is mathematically defined in Equation 2.1. In the mathematical definition following($u_j$) is the number of followers user $u_i$ has, and d is a dumping factor. All users start with a default authority of (1/U), where U is the total population. The dumping factor represents the probability that a random surfer moves from one user to another, and is typically set to 0.85 (Cataldi *et al.*, 2010). A user gains authority by gaining followers and by his or her followers gaining followers.

$$auth(u_i) = d \times \sum_{u_j \in follower(u_i)} \frac{auth(u_j)}{|following(u_j)|} + (1 - d)$$

**Equation 2.1 Cataldi et al. (2010) Authority Definition.**

If a user with high authority has a follower with high authority, then the number of users a piece of content can potentially reach is significantly increased, thereby increasing the lifespan of a given topic discussed by these users.

Cataldi *et al.* (2010) research retained the notions of energy and nutrition from Chen *et al.* (2003), and added the notions of "hot" and "emergent" which allow users to set thresholds to detect when topics are very popular or up-and-coming, respectively. Again, the issue of user-set parameters is present. The user either arbitrarily sets a value or has to perform rigorous testing before knowing what values are practical for detecting hot or emergent topics.

In this work, we focused on the lifespan of a single piece of content as oppose to a topic. This is motivated by the fact that all prior research observed that individual pieces of content contributed to the lifespan of a topic, but did not investigate the traits that influence the lifespan of that content. We also eliminated the need for the rigorous parameter tweaking required by previous methods by using data mining techniques that only require data input files.

## 2.2 Trend Analysis

Trend analysis monitors the rise and fall of activity of a piece of content or topic. Websites have trend information available, but some do not share the algorithms used to obtain that information. The measuring of trends is a broad field. It can involve measuring a reoccurring topic in Twitter posts (Trendistic, 2007), Google Trends, which measures the reoccurrence of Google searches (Google.com, 2010), or the amount of views, favorites, and comments on a YouTube video (YouTube.com, 2010), as illustrated in Figure 2.1. The trend data on YouTube, though publicly available in 2010, has since been from public access and restricted to the owner of the video.

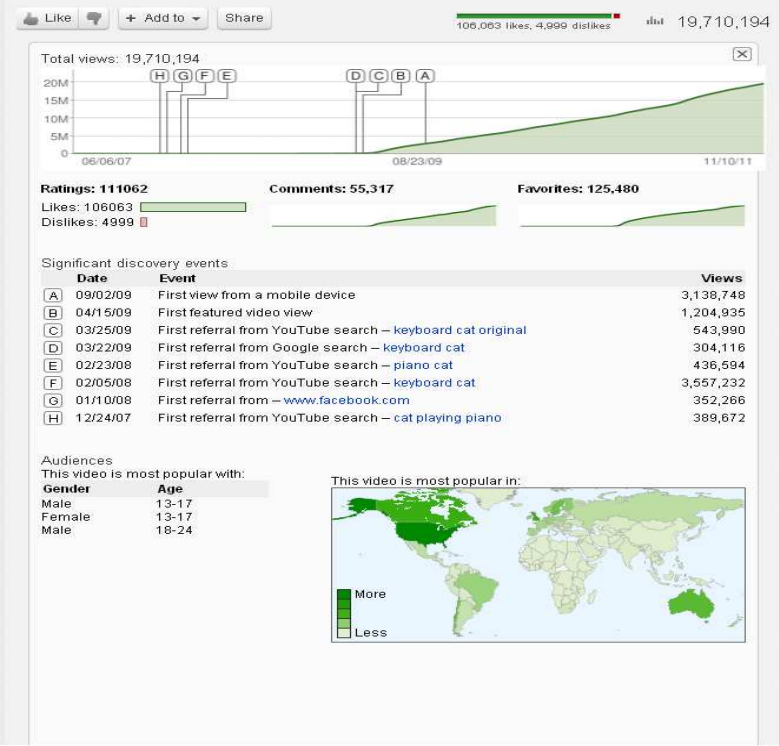


**Figure 2.1 YouTube Trend Analysis (YouTube.com, 2010).**

Trend analysis, in effect, is a post mortem analysis of content activity. It is possible to observe a rising or falling trend and predict the rest of the lifespan; however this does not capture the cause of the trend. YouTube trends and Google trends (Google is the owner of YouTube) attempt to analyze what occurred,

external to the content, to explain any significant changes in a trend. Again, this fails to capture a key piece of information that our work will investigate, i.e., what attributes of the content allowed for the trend to change. For example, in Figure 2.1, YouTube observes that views increased when the video was being linked to from Facebook and searched for on Google. Granted, this altered the trend, thereby extending the lifespan of this video, but this information does not say anything about the attributes of the video that made it worth sharing.

## 2.3 Online Social Networks

This section will discuss the online social networks chosen for this research and experimentation.

### 2.3.1 Reddit

Reddit is a social network that allows for the posting of links to content on other Websites or to posts that exist within Reddit. Links can be to any online content including images, videos, or other Websites. What makes Reddit unique among other social networks is its organization and ability to sort the "hottest" (combination of most popular and most recent) content.

Reddit is divided into subReddits. A subReddit is tailored to a specific topic, for example there is a subReddit devoted entirely to politics, another one to pictures of cute animals, and another one for cooking advice. These subReddits can be for an extremely narrow topic, for example there is a subReddit that only allows for the post of a picture of a dog that have color patterns that give the impression of having eyebrows. A user can post to and subscribe to any subReddit. All content on Reddit exists within in a subReddit. A user's home page, referred to as the front page, shows the top 25 links from all subReddits to which the user is subscribed.

The ranking of all posts is determined by two things: the score and the time since submission. Every piece of content on Reddit, each post, can be voted "up" - increasing the score - or "down" - decreasing

the score - by any user. The total number of upvotes minus the number of downvotes determines the post's overall score. In order to determine the ranking, referred to as the "hot" ranking by the Reddit source code, of a post the algorithm (Github.com/reddit, 2011) in Figure 2.2 is applied. To describe it simply, the longer a post has been alive, the more difficult it is for it to have a high "hot" ranking; a post that is 12 hours old will need to have a score 10 times higher than a post that is just made; a post that is 24 hours old will need a score that is 100 times higher. This algorithm is what lead us to decide our timeframe of 24 hours for watching a post.

```
from datetime import datetime, timedelta
from math import log

epoch = datetime(1970, 1, 1)

def epoch_seconds(date):
    """Returns the number of seconds from the epoch to date."""
    td = date - epoch
    return td.days * 86400 + td.seconds + (float(td.microseconds) / 1000000)

def score(ups, downs):
    return ups - downs

def hot(ups, downs, date):
    """The hot formula. Should match the equivalent function in postgres."""
    s = score(ups, downs)
    order = log(max(abs(s), 1), 10)
    sign = 1 if s > 0 else -1 if s < 0 else 0
    seconds = epoch_seconds(date) - 1134028003
    return round(order + sign * seconds / 45000, 7)
```

**Figure 2.2 Reddit Ranking Code (Github.com/reddit, 2011, Salihefendic, 2010)**

A user is allowed to subscribe to an unlimited number of subReddits, but at the time of experimentation all users were subscribed to the 20 subReddits by default (Martin, 2011), as listed and described in Table 2.2. Our experiments behaved as a new user, subscribed to the default subReddits.

| Default SubReddits | Description |
|---|---|
| **Pics** | Links to pictures of anything. |
| **Gaming** | Video games. |
| **World News** | Links to and discussion about world news - anything outside the USA. |
| **Videos** | Links to videos from other sites. |
| **Today I Learned** | Facts people just became aware of (links are to sources). |
| **IAmA** | Also known as "I am a ___. Ask me anything!" A person announces their job title or position in life and discussion follows. Example "I am The President of the United States. Ask me anything". |
| **Funny** | Links to and discussion about anything funny. |
| **Atheism** | Links to and discussion about atheist related topics. |
| **Politics** | Anything political. |
| **Science** | Links to and discussion about science related topics. |
| **AskReddit** | A text-post only subReddit, where questions can be posted that anyone else on Reddit can reply to. |
| **Technology** | Links to and discussion about technology related topics. |
| **WTF** | An abbreviation for the perplexed; links to anything confusing, shocking, or difficult to explain. |
| **Blog and Announcements** | Official posts from the Reddit employees about the changes, updates, or events. |
| **Bestof** | Links to other posts or comments on Reddit that are considered the best of Reddit. Only links to the Reddit.com domain are allowed. |

| | |
|---|---|
| **Advice Animals** | An Internet meme of animals with text. Typically humorous in nature. |
| **Music** | Links to and discussion about music-related topics. |
| **Aww** | Aww is meant to represent the sound one would make when viewing something adorable. This subReddit consists entirely of links to cute things. |
| **Askscience** | A text-post only subReddit where all posts are questions related to science. |
| **Movies** | Links to and discussion about movie related topics. |

**Table 2.2 Default SubReddits.**

Research on Reddit has included evaluation effectiveness of the ranking system that requires user participation (Gilbert, 2013, Mills, 2011). Gilbert (2013) found that the ranking system used by Reddit has the consequence of missing popular content on early submission attempts. A notable 52% of popular posts, from his sample, were actually resubmission of less successful posts. Mills (2011) found that the majority of posts are seen by very few people, and the most popular post - the post that make the front page viewed by nearly every user - influences the type of posts made. For example, a post that makes it to the front page discussing the presidential election would start a rise in the number of post submissions related to the election. This seemed to be an effort to gain a high scoring post. There is an incentive on Reddit to seek out and post high quality content. As a post with a high score ads to the user's "karma" which is the sum of all positive scoring posts made by the user. Karma is non-transferable and only acts as a metric to gauge how good a user is at providing well-received content consistently.

### 2.3.2 4chan

The site, 4chan.org, is not only one of the most controversial Websites on the Internet, it is also the most frequented English speaking message boards, with over 25 million monthly visitors (4chan.org, 2013). The site is extremely simple. It is split into message boards, each with an abbreviation as its title, ranging "/gif/"

for the board devoted to ".gif" images, to /b/ which is the "random" board, where anything can be posted - no topics are restricted. Anyone can navigate to 4chan and post any image he/she wants, generate a thread, and all other users are allowed to reply to that image with another image, text, or a combination of the two, creating a post. Figure 2.3 shows how boards are organized. Each board has the most recently updated thread at the top, meaning the most recently created thread or the thread that most recently received a reply. Each thread can be navigated to. Figure 2.4 illustrates the organization of a single thread. The replies to a thread are organized newest to oldest.

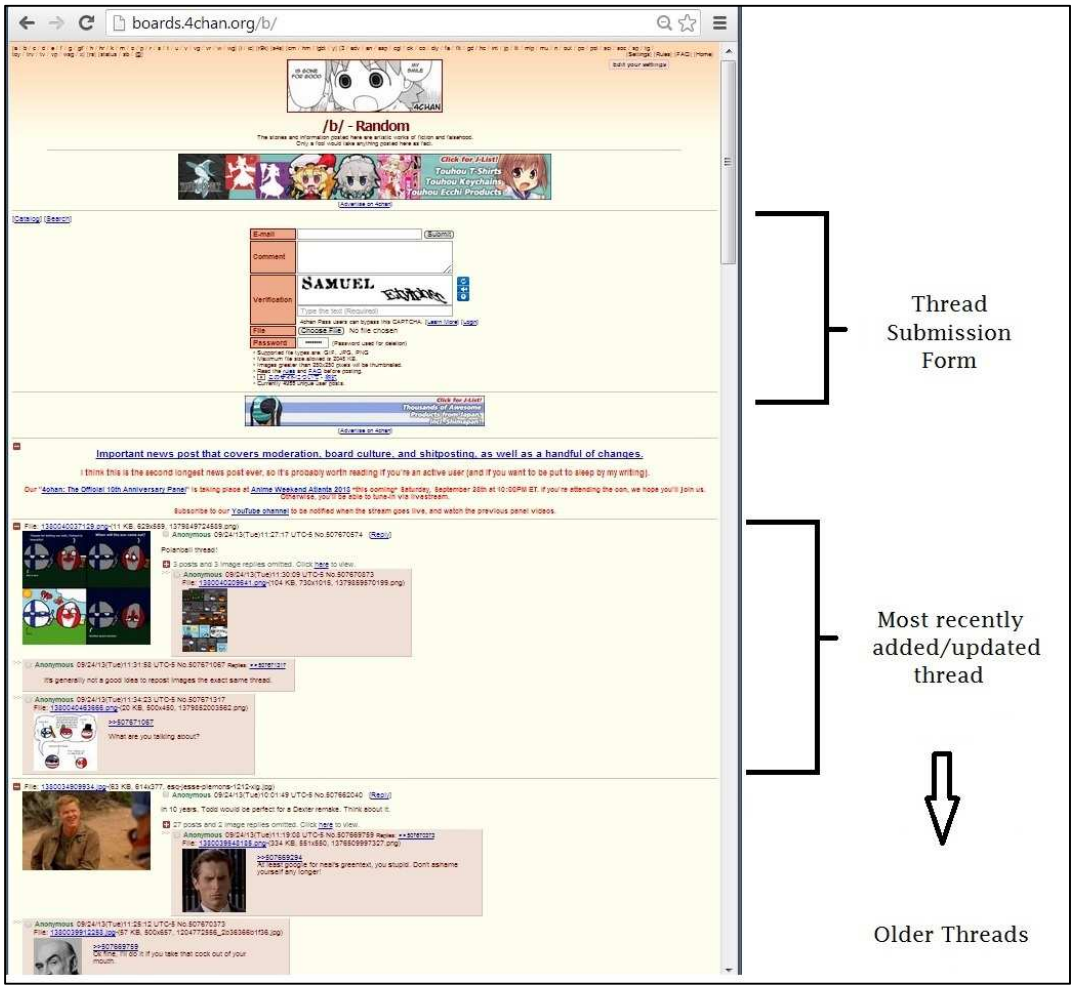


**Figure 2.3 4chan /b/ Board Organization (4chan.org, 2013).**


**Figure 2.4 4chan thread organization (4chan.org, 2013).**

Each board has a limited number of threads at any given time. The board /b/, for example allows for 16 pages with 15 threads each. This means that when a thread is not interacted with, it is pushed further down the list to later pages, eventually being removed from the site. There is no notion of archiving on 4chan. When a thread is pushed to the end of the list it is gone forever. Depending on the number of new threads being posted, a thread that does not receive replies steadily can be removed from the site in as little as 28 seconds (Berstein *et al.*, 2011).

Berstein *et al.* (2011) examined the anonymous and ephemeral environment of 4chan, particularly of 4chan's most active board, "/b/." They found that the majority of posts, over 90%, are made anonymously. This anonymity may cause one to think that 4chan could not act as a social network; however, an incredible amount of identity and organization emerges from 4chan, particularly on its most popular board /b/. The users of this board have invented many Internet memes that eventually reached the main stream, including the "LoLcats", pictures of cats with funny captions (Poole, 2010). The board has organized public rallies, protests, and performed Distributed Denial of Service (DDoS) attack on organizations that do not agree with their way of thinking, such as Scientology (C. Poole Presentation, 2010) sites or the Westboro Baptist Church's funeral protesters (Schwartz, 2012). The ephemeral nature of 4chan seems to be one of its strengths when it comes to building an online culture; since there is no archive, the users serve as the memory of the site, and despite not having any archived threads to reference, the culture at 4chan is constantly rekindled by returning users creating new threads.

### 2.3.3 Flickr

Flickr is an online social network that allows for the sharing of images and videos. Created by Ludicorp in 2004 and purchased by Yahoo! Inc. in 2005, Flickr is home to 87 million registered users, and each day, over 3.5 million images are uploaded to the site (Jefferies, 2013). Flickr does not require registration to view content, but it is required before uploading or interacting with the content. Every registered user is allowed a collection of content. Tags can be applied to content to serve as a piece of metadata that aid

other users when searching Flickr. Geotagging, allows for content to be assigned a place of origin, for example, the location where a photo was taken. Users can comment on images or videos or favorite them. Content on Flickr is persistent, though long-term account inactivity may lead to the account being removed. To find content, users can use the built-in search engine to search by topic or location. Users can add each other as contacts, which acts as a means of subscribing to a user's photo feed. The homepage also consists of recently uploaded photos from all users.

Research on Flickr's content and the activity that surrounds it has included analysis of how content is shared throughout the network and at what speeds (Cha *et al.,* 2009) and, what fraction of Flickr users are active users, meaning they comment or favorite regularly (Valafar *et al.*, 2009). Research suggests that only a small fraction of Flickr's users actively participate in the commenting and favoriting of images. Also, when content is receiving comments and favorites, that activity comes slowly from users that are within a few hops of the uploaders social graph (i.e., friends of friends). The research of Cha *el al.* (2009) also suggests there is no correlation between the age of a photo and its potential to gain in activity.

### 2.3.4 YouTube

YouTube is an online social network that allows users to share videos with one another and the public (YouTube Data API, 2013). YouTube only restricts videos based off of several criteria such as copyright, violence, or being sexually explicit. Each user is given a "channel" in which all of his/her videos are kept. These channels can be subscribed to by other users. Users have a subscription feed that gives notification of when a new video from a publisher has been added. By default, a user does not have subscription and is shown promoted and recommended videos. Promoted videos show up as links in the recommendations list, but are labeled as "promoted." Channel owners have the options to pay to have their videos advertised as a promoted video to increase viewership. Recommended videos can be seen next to any video viewed on the YouTube.com domain — instead of viewing an embedded video on an external site. The recommendation system was researched by Zhou *et al.* (2010) who found it to be

extremely effective at obtaining a user click-through, meaning a user has clicked on a recommendation. A correlation in view counts was also found among videos that could be found in each other's recommendation list.

More research on YouTube was recently performed that explored data sets collected across two years in order to investigate several correlations (Cheng *et al.*, 2013). The research studied data collected from millions of videos collected from 2007 to 2008. Among finding many opportunities for optimizations, Cheng *et al.* (2013) also found the following.

- There is no strong correlation between video popularity and video length.
- Predictive models struggle to accurately predict the popularity videos, especially those with little activity.
- Despite videos being permanent (uploaders can remove videos voluntarily and YouTube can remove if a video violates the terms of service), videos can be demonstrated as having an "active lifespan" where activity trails off or stops completely.

## 2.4 Data Mining Techniques

Data mining is the process of using an algorithm that analyzes a data set in order to learn about the data set and, perhaps, discover patterns or traits within the data that may be useful on other, future, data sets. Data mining algorithms vary in complexity, efficiency, and ability to process certain kinds of data. The range of output of data mining algorithms varies as well. Some, like a decision table, output human readable rules while others, like Support Vector Machines, create hyper-planes in order to classify data. There are many data mining algorithms, and the Waikato Environment for Knowledge Analysis (Hall *et al.*, 2009) has collected many of them into a single software suite that can be used for experiments.

### 2.4.1 WEKA

The Waikato Environment for Knowledge Analysis (WEKA) (Hall *et al.*, 2009) is a machine learning suite that contains several data mining tools for data preprocessing, filtering, clustering, classification, association, and attribute selection. Users can select from dozens of algorithms for experimentation. WEKA also has tools to extract data from a "My Standard Query Language" (MySQL) database (MySQL.com, 2011), or a comma-separated values (CSV) file (Shafranovich, 2005) and to convert data into the standard format used by WEKA, the Attribution Relation File Format (ARFF). WEKA is for public use and falls under the GNU General Public License (GPL) (GNU.org, 2011). For our experiments we chose commonly used data mining techniques, namely, SMO, Decision Table, Naïve Bayes, and Random Forest. WEKA contains many more algorithms such as, KStar, ZeroR, and HyperPipes.

### 2.4.2 SMO

The Sequential Minimal Optimization Algorithm (SMO) is an implementation of support vector machines (SVM) created by John Platt (1998). An SVM uses hyper-planes in order to establish boundaries between data points, separating them into different classes. Platt's (1998) implementation "globally replaces all missing values and transforms nominal attributes to binary ones" (Hall *et al.* 2011). SMO, like all support vector machines perform very well on large sparse data sets where the number attributes and the number of instances are large (Joachims, 2006). Though a large number of attributes can be handled quite efficiently, once the number of instances becomes large (million) they "demonstrate super-linear behavior" (Joachims, 2006).

### 2.4.3 Decision Table

The Decision Table Algorithm, when applied to a data set with instances that belong to different classes, produces table that acts as a means to make decision as to how to classify future instances. The Decision Table algorithm serves two purposes: 1) to correctly classify instances/samples to the correct class, and 2)

to calculate an optimal subset of features that are used in classification. In other words, a Decision Table not only seeks to minimize error when classifying a given instance, it also seeks to obtain a set of feature that appear to dictate the classification of all instances. The Decision Table used in WEKA uses a Decision Table Majority (DTM) in order to classify instances and the "wrapper algorithm" as an induction algorithm in order to select an optimal set of features (Kohavi, 1995). Kohavi (1995), performed a very in-depth survey of the Decision Table, Decision Tree, and rough set theory work along with comparing his implementation of an inducer of DTMs referred to as (IDTM). Kohavi's (1995) work showed that Decision Table's accuracy can compete with that of C4.5's, another decision tree algorithm developed by Quinlan (1993). Though time comparisons were not performed, a long discussion was included about time complexity. Decision Table can suffer from a long running time due to the length of time it take the inducer algorithm to calculate the optimal feature subset. The time complexity of the inducer algorithm, which selects the optimal feature subset, was calculated as

$$O(T + m(t_d + t_c + t_i))$$

**Equation 2.2 Decision Table Time Complexity**

where $T$ is the running time of induction algorithm on the full data set, $m$ is the number of instances and $t_d$, $t_c$, and $t_i$ are the time required for a single instance to be deleted, classified, and inserted during the cross-validation respectively (Kohavi, 1995).

### 2.4.4 Naïve Bayesian

The naïve Bayesian classifier is a probabilistic classifier that assumes the presence or absence of any attribute is independent from all other attributes (John and Langley, 1995). Although naïve Bayesian classifier "models have no explicit mechanism" to handle sparse data sets (Banerjee and Shan, 2007), experiments have shown acceptable performance on data sets similar in size and sparseness similar to our work, namely, Wang's (2007). Improvements to the representation of sparse, taking the "dense" format

(all attributes and values explicitly labeled for each sample) to a "sparse" format (only present attributes listed for each sample) resulted in addition speed ups in Naïve Bayesian performance (Chickering and Heckerman, 1999). Chickering and Hekcerman (1999) gives "sparse" an alternate meaning than the one used in this work. Our definition of a "sparse" data set is a dataset where very few attributes will have values differing from the default for a given sample. For example a large matrix that is mainly zeros with a few nonzero values sprinkled throughout. Although the exact implementation of the file format differs, the WEKA ARFF files used store spare data in a similar way.

### 2.4.5 Random Forest

Random Forest uses a collection of Decision Trees, each tree casting a vote for classification, to pick a majority vote across all Decision Trees (Breiman, 2001). A Decision Tree is a decision making tool, but a Decision Tree has a tree-like graph structure which employs sequential decision making component (i.e., a decision take you down a branch of the tree, removing the possibility of reaching some decisions). A Decision Tree is similar to a Decision Table in that both provide a means to classify an instance based on the values it contains, the the Decision Table algorithm in WEKA is performs an exhaustive search across all attributes to find an ideal subset of features whereas a single Decision Tree is assigned a random set of attributes. The random dimension of the attribute set is chosen at each node where a split is made based on a random threshold for that attribute's value. Splits are continually made until a stopping criteria is reached or a set depth threshold is met. In WEKA the default setting is unlimited depth (Hall *et al.*, 2009). Random Forest has been shown to avoid over-fitting "due to the law of large numbers" and perform on large sparse data sets, such as medical data, with accuracy comparable to that of Bayes (Breiman, 2001). Random forest can be used in a multi-classifier system that utilize several classifiers at different phases of the overall classification process. Recently, Yang (2013) surveyed modifications of Random Forest and applied a novel approach to audio tagging software. The result showed that modifications of Random Forest can produce excellent feature selection, and those features can then be

used by another algorithm that has better time performance on sparse data sets such as a support vector machine that performs classifications in near real-time.

# Chapter 3 Research Approach

This chapter presents the research approach and provides a description of the system architecture.

## 3.1 Methodology

We have built several applications in a three tier approach, using existing applications and APIs. The three tiers are presentation, logic, and data. The presentation tier handles the user interface, passes the user input to the logic tier, and displays the results received from the logic tier to the user. The logic tier handles extracting data from the data tier, processing the data, and passes the results to the presentation tier. The data tier handles data storage. Figure 3.1 illustrates the interaction between applications across tiers.

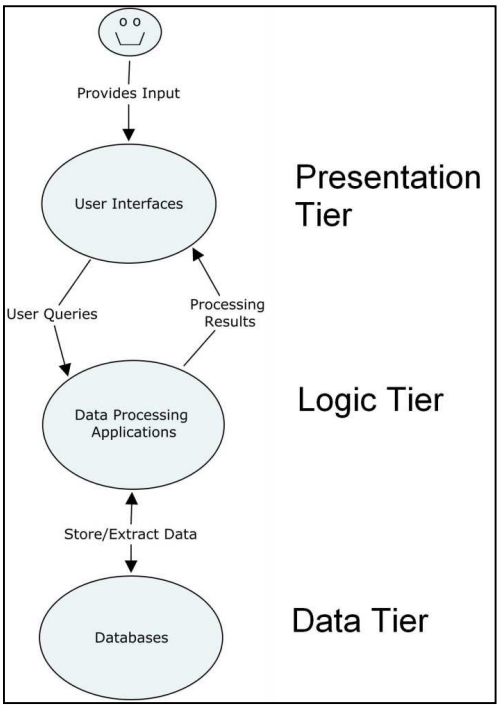


**Figure 3.1 Three Tiers.**

## 3.2 Lifespan Models

The lifespan models we generated were based on a preliminary data set of 10,000 Reddit posts. Through several iterations of experimentation, we decided that in order to model lifespan, we use two main categories, namely, the lifespan's Peak/Death Timings and Popularity Tiers.

### 3.2.1 Peak/Death Timings

We measure the liveliness of content based on the content's activity. Activity varies for each OSN and is detailed in a later section. We use two critical points in time during a content's lifespan. These points of interest are:

(1) The time of the peak of activity.

(2) The time where the post is considered dead.

In order to discover when a piece of content has peaked and died we use t_peak, t_rise, and t_death, as detailed in Table 3.1.

| Name | Definition |
|---|---|
| **t_rise** | The time at which content's activity strictly continues to rise until hitting t_peak. This point in the lifespan is only used to help detect to the point of death, t_death. |
| **t_peak** | The time at which the content's activity is highest. |
| **t_death** | The time at which the activity has hit a level equal to or below the level of activity of either the first snapshot (collection of public data) or t_rise. |

**Table 3.1 t_rise, t_peak, and t_death definitions.**

Using these critical points, the content's peak and death can be categorized into the four categories defined in Table 3.2. Note that t_rise is not listed, but aids in determining when a piece of content can be considered dead.

| Category | Definition |
|---|---|
| **Early peak + early death** | t_peak and t_death occur in the first half of the lifespan |
| **Early peak + late death** | t_peak occurs in the first half the lifespan and t_death occurs in the second half of the lifespan |
| **Late peak + late death** | t_peak and t_death occur in the second half of the lifespan |
| **Dead on arrival** | Zero activity for the entire lifespan |

**Table 3.2 peak/death categories.**

### 3.2.2 Popularity Tiers

From simple observation of a wide variety of social content, it is observed that content lifespan can take on several different forms. For example, if a celebrity on YouTube channel with a large subscription base puts out a new piece of content, typically this content will have a different lifespan, with higher levels of activity than that of a typical user. We analyzed the preliminary data collected from Reddit to define the tiers listed in Table 3.3. What counts as activity will vary with each OSN, but all activity is weighted the same. For example, of a post on Reddit receives an upvote that adds one to the activity. If that same post receives a downvote, that also adds one to the activity. Any user interaction adds one to a post's activity score. The tiers being separated by powers of 10 attempts to model the large differences in activity between the average piece of content and a super popular or viral ones.

| Tier | Activity level at t_peak |
|---|---|
| **Dead on arrival** | Zero activity for the entire lifespan |
| **Below Average** | $0 \leq \text{activity}_{tpeak} < 10$ |
| **Average** | $10 \leq \text{activity}_{tpeak} < 100$ |
| **Popular** | $100 \leq \text{activity}_{tpeak} < 1{,}000$ |
| **Super popular** | $1{,}000 \leq \text{activity}_{tpeak} < 10{,}000$ |
| **Viral** | $\text{activity}_{tpeak} \geq 10{,}000$ |

**Table 3.3 Popularity Tiers.**

## 3.3 Combining Peak/Death Timings with Popularity Tiers

Once the peak/death timing and the popularity tiers are determined, they can be combined to give a general shape to the lifespan, with, $t_0$ and $t_n$ denoting the first time and last time a piece of contents activity is observed, respectively.

### 3.3.1 Dead on Arrival

Dead on arrival implies the content experience zero activity. The content never peaks, instead is considered dead for the entire lifespan as illustrated in Figure 3.2.

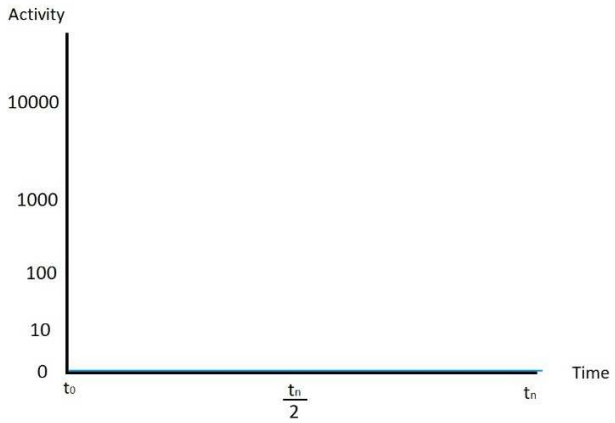


**Figure 3.2 Dead on Arrival.**

### 3.3.2 Below Average Combination

The below average popularity tier combination never surpasses an activity of 10 during any point in the content's life, as illustrated in Figure 3.3, Figure **3.4**, and Figure **3.5**.

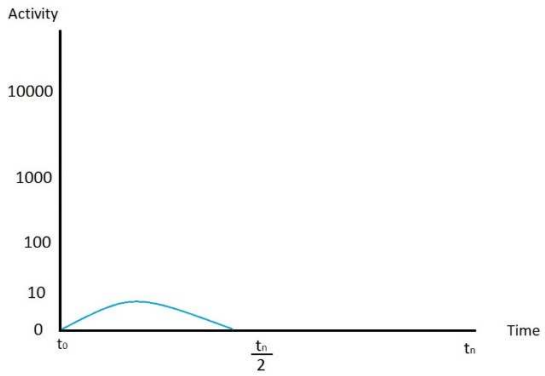


**Figure 3.3 Below Average Early Peak Early Death.**

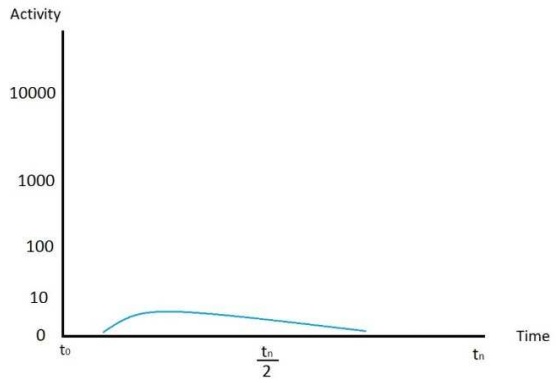


**Figure 3.4 Below Average Early Peak Late Death.**

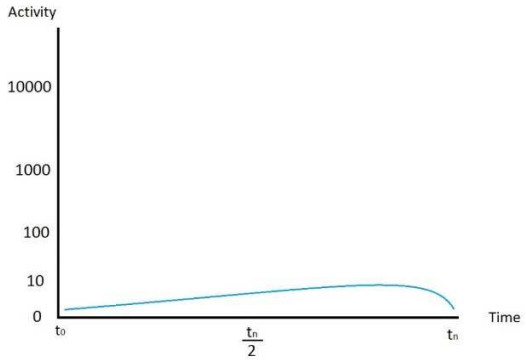


**Figure 3.5 Below Average Late Peak Late Death.**

### 3.3.3 Average Combination

The average popularity tiers combination always peaks above 10 but never surpass an activity of 100 during any point in the content's life, as illustrated in Figure 3.6,

**Figure 3.7**, and Figure **3.8**.

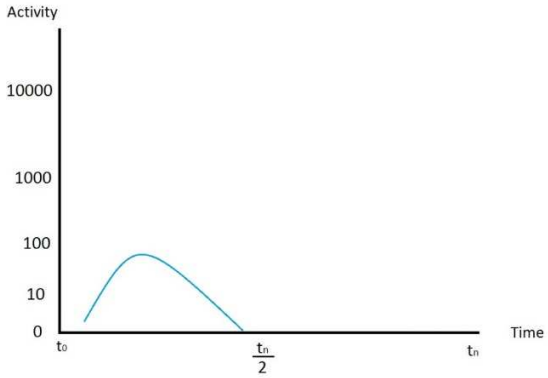


**Figure 3.6 Average Early Peak Early Death.**

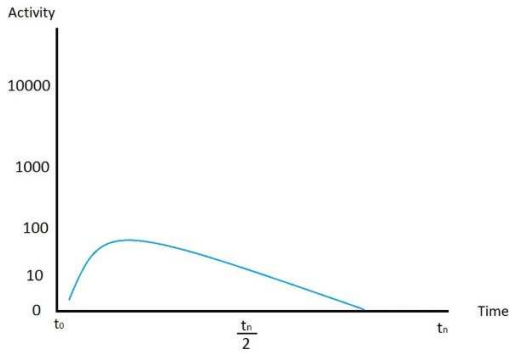


**Figure 3.7 Average Early Peak Late Death.**

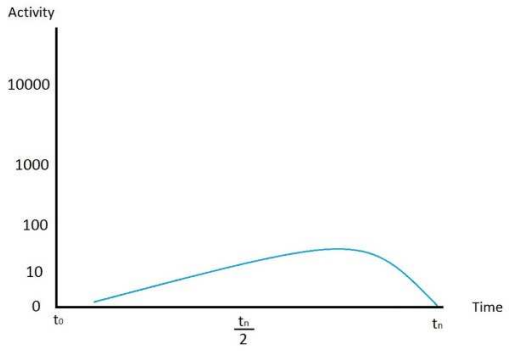


**Figure 3.8 Average Late Peak Late Death.**

### 3.3.4 Popular Combination

The popular popularity tiers combination always peaks above 100 but never surpasses an activity of 1,000 during any point in the content's life as illustrated in Figure 3.9, Figure **3.10**, and Figure **3.11**.

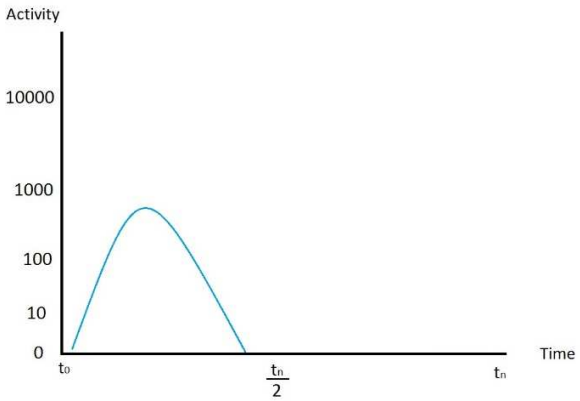


**Figure 3.9 Popular Early Peak Early Death.**

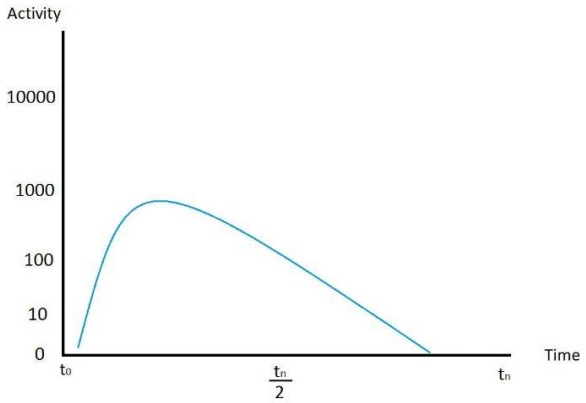


**Figure 3.10 Popular Early Peak Late Death.**

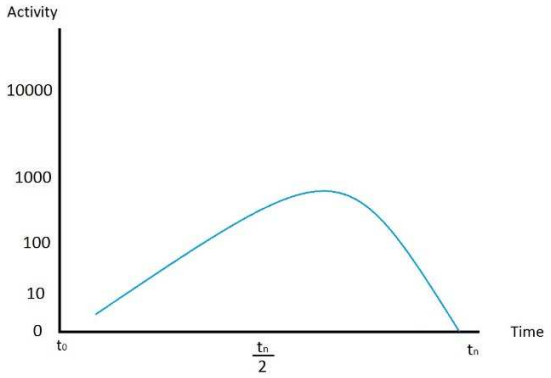


**Figure 3.11 Popular Late Peak Late Death.**

### 3.3.5 Super Popular

The super popular popularity tiers combination always peaks above 1,000 but never surpasses an activity of 10,000 during any point in the content's life as illustrated in Figure 3.12, Figure **3.13**, and Figure **3.14**.

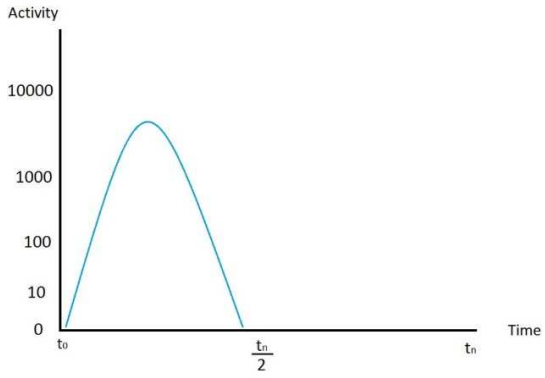


**Figure 3.12 Super Popular Early Peak Early Death.**

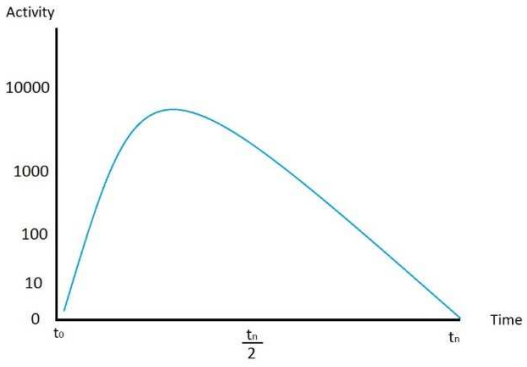


**Figure 3.13 Super Popular Early Peak Late Death.**

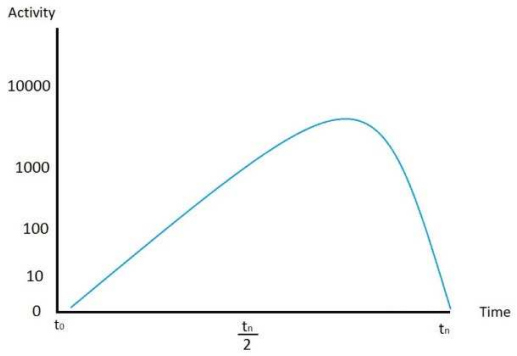


**Figure 3.14 Super Popular Late Peak Late Death.**

### 3.3.6 Viral

The viral popularity tiers combination always peaks above 10,000 as illustrated in Figure 3.15, Figure **3.16**, and Figure **3.17**.

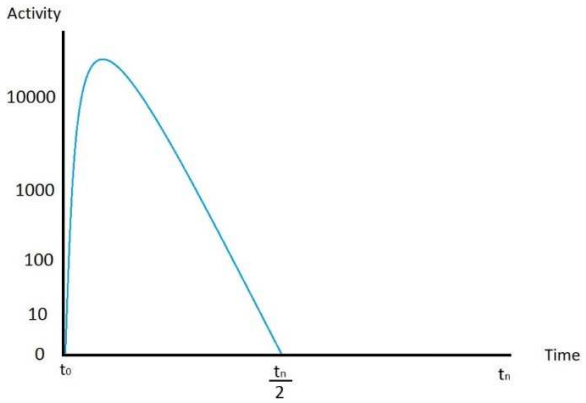


**Figure 3.15 Viral Early Peak Early Death.**

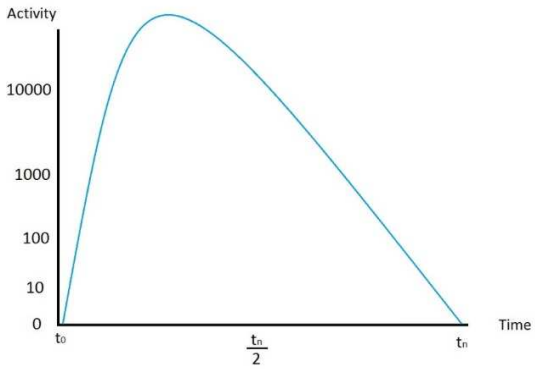


**Figure 3.16 Viral Early Peak Late Death.**

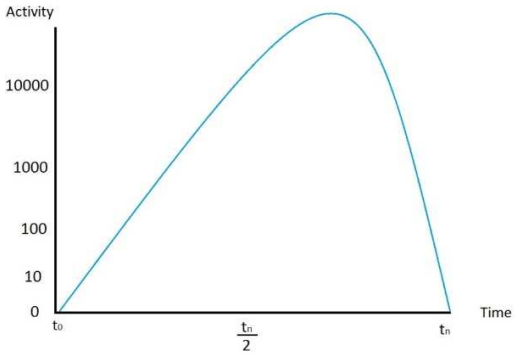


**Figure 3.17 Viral Late Peak Late Death.**

## 3.4 System Architecture

We developed applications for data acquisition, activity analysis, and model generation. The models are then transferred to the data mining applications in order to test the accuracy of a given data mining algorithm using the models.

### 3.4.1 Data Acquisition

Each piece of content has temporal and non-temporal data. The non-temporal data does not change throughout the content's lifespan, such as "date posted" or "author". The temporal could change during the lifespan and were interpreted as activity, such as "number of likes" or "number of comments".

For each OSN, its respective APIs were used to gather all publicly available data from the earliest possible point in each content's lifespan. For Reddit, 4chan, and Flickr, there are location devoted to displaying the most recently added content. The lifespan of YouTube content was determined to be far too long to monitor in the time available, so the peak/death model portion was omitted, but the popularity tier analysis was performed. Videos were chosen at random and all data were collected using the YouTube API.

Each OSN's API was used to gather snapshots of a piece of content over its lifespan. We define a snapshot as:

> ***A collection of all publicly available data for a piece of content accessible through an API at a given time, t.***

Snapshots of a piece content are taken at a regular interval. For example, given a piece of Reddit content denoted by *post_n,* with a unique identifier of *id,* the Reddit API is used to pull all available information using *id.* The extracted information is time-stamped and stored in the local database. Figure 3.18 illustrates the process of acquiring snapshots. All snapshots for a piece of content are analyzed to calculate the change in activity over time. Our data acquisition program will serve as a wrapper to an OSN's API to acquire all publicly available data about a single piece of content starting at an initial time, t0 and reacquiring the same data on a predetermined interval until the maximum time limit is reach (e.g., 24 hours for Reddit). The data is reacquired in order to monitor any changes in activity metrics over time (e.g., number of comments or number of upvotes).

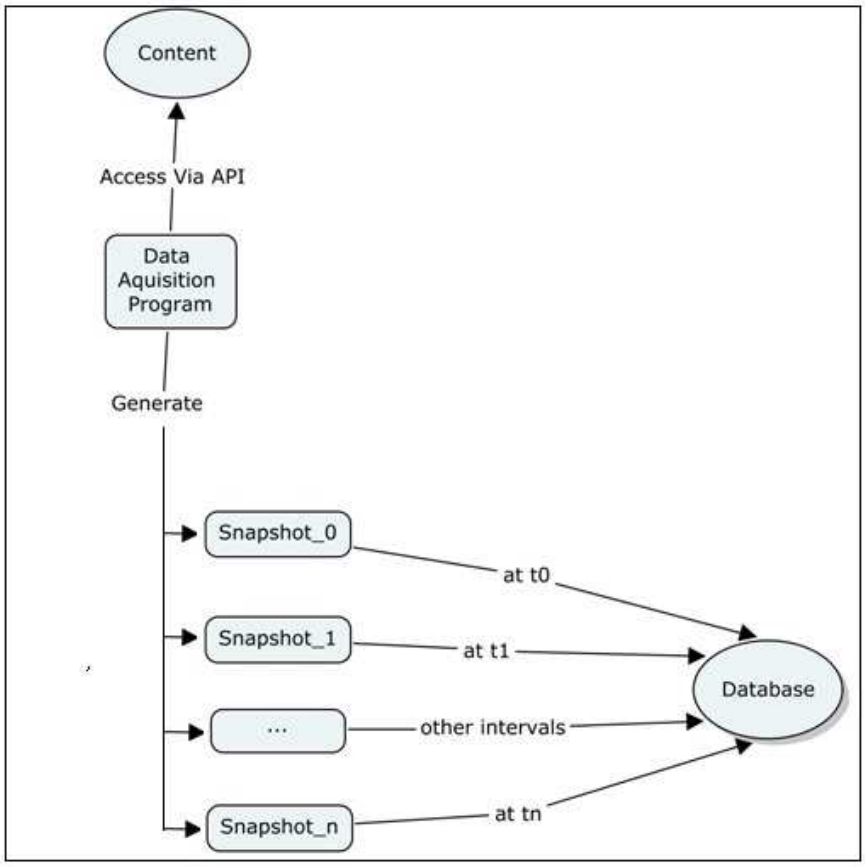


**Figure 3.18: The Process of acquiring Snapshots.**

Table 3.4 shows an example of a collection of snapshot for a single tweet. Each row contains a unique identifier for the content along with the values of all activity metrics. In this example, the activity metrics of upvotes, downvotes, and number of comments on the original post change during the period in which snapshots are taken. The data collected include a wide variety of components, such as author information, content information, and activity information. In Reddit's case, there are several ways to observe activity, including tracking the number of upvotes, downvotes, and comments.

| PostID | TimeOfSnapShot | Upvotes | Downvotes | Comments |
|---|---|---|---|---|
| **123456** | 2011.10.03 09:54:00 | 0 | 5 | 0 |
| **123456** | 2011.10.03 11:54:00 | 2 | 5 | 1 |
| **123456** | 2011.10.03 01:54:00 | 17 | 6 | 3 |
| **123456** | 2011.10.03 03:54:00 | 89 | 8 | 15 |
| **123456** | 2011.10.03 05:54:00 | 350 | 55 | 22 |

**Table 3.4 Sample Post Snapshots.**

### 3.4.2 Activity Analysis

After collecting a snapshots for all observed posts, the lifespan of each post is analyzed. This analysis assigns a peak/death timing and a popularity tier to each post. Then the data are translated into a format that is readable by a data mining program. WEKA connects to the database containing all of the analyzed data and generates a file readable by the data mining applications. This file does not contain temporal data. At this point, the lifespan of the content has been analyzed and classified and is ready to be used for training and testing. Figure 3.19 illustrates the use of snapshots by the activity analyzer.

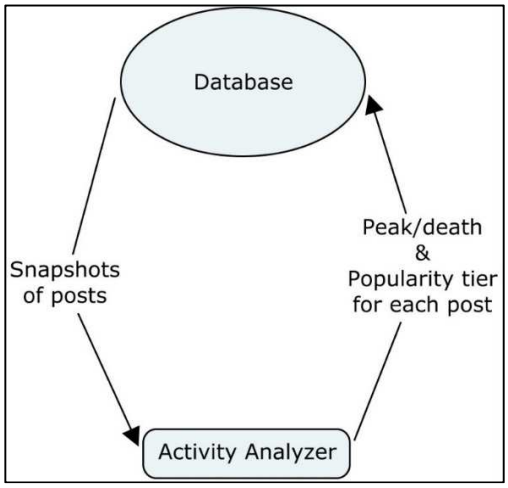


**Figure 3.19 Activity Analysis.**

Table 3.5 lists the activity metrics for each network studied in this work. It should be noted that metrics may be added or removed from this list based on any feature additions or removals from a given OSN. Every activity metric has equal value.

| OSN | Current Activity Metric Candidates |
|---|---|
| **Reddit** | Upvotes, Downvotes, Comments |
| **YouTube** | View Count, Favorites, Likes/Dislikes, Comments, Video Replies |
| **4chan** | Text Replies, Image Replies |
| **Flickr** | Views, Comments |

**Table 3.5 OSN Activity Metrics.**

### 3.4.3 Model Testing

In order to analyze the models generated for each piece of content, WEKA extracts the analyzed data and uses the data as the testing and training data in a K-fold cross-validation, where K=10 (i.e., ten fold). WEKA produces many results. We focused on the accuracy of the models generated and any information that indicated the influence of particular attributes or values of attributes (e.g., observing the word "kitten" in a title influences the activity). It should be noted that when the data are converted into the standard format for WEKA, the ARFF format, the text-based attributes, such as title, are converted into word vectors. Each word is considered to be an individual attribute. Figure 3.20 illustrates the process of using the analyzed data to test the model accuracies.

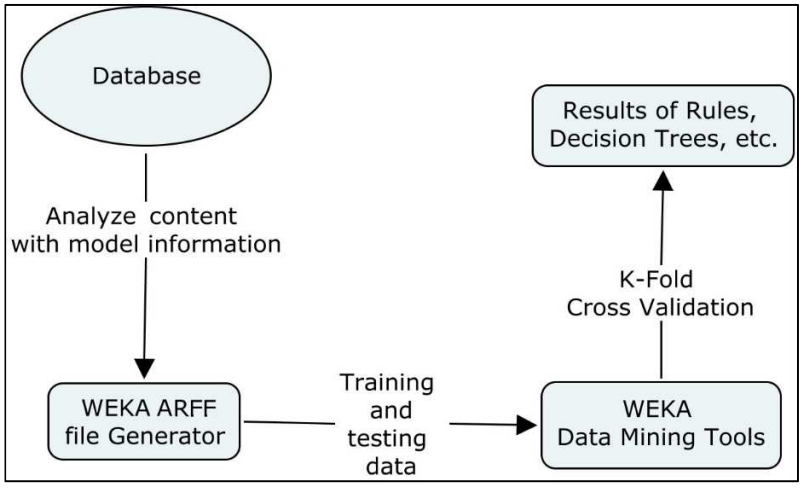


**Figure 3.20 Model Analysis.**

# Chapter 4 Implementation

This chapter details the implementation, application purposes, and relationships between applications. For each API a separate API wrapper was developed to transfer data from a given OSN to a local MySQL database. Figure 4.1 displays the interactions between the various components in our system. The first set is for our API to pull all publicly available data from an OSN via its API and store it locally. Once we have a post in our system, we can then take snapshots of that post and store those locally as well. After we have a desired amount of posts with accompanying snapshots, the activity analyzer tracks how the activity level changed for all the posts and assigns each post a peak/death category (i.e., early peak or late peak and early death or late death) and a popularity tier (i.e., average, popular, viral, etc.) Once all the posts are classified, WEKA extracts the post data from the local database and generates an ARFF file that can be reused for multiple data mining algorithms. WEKA is then used to pick a data mining algorithm to act as a classifier and run the 10-fold cross validation producing an output file with error percentage, runtime, and confusion matrices which we use in our analysis.

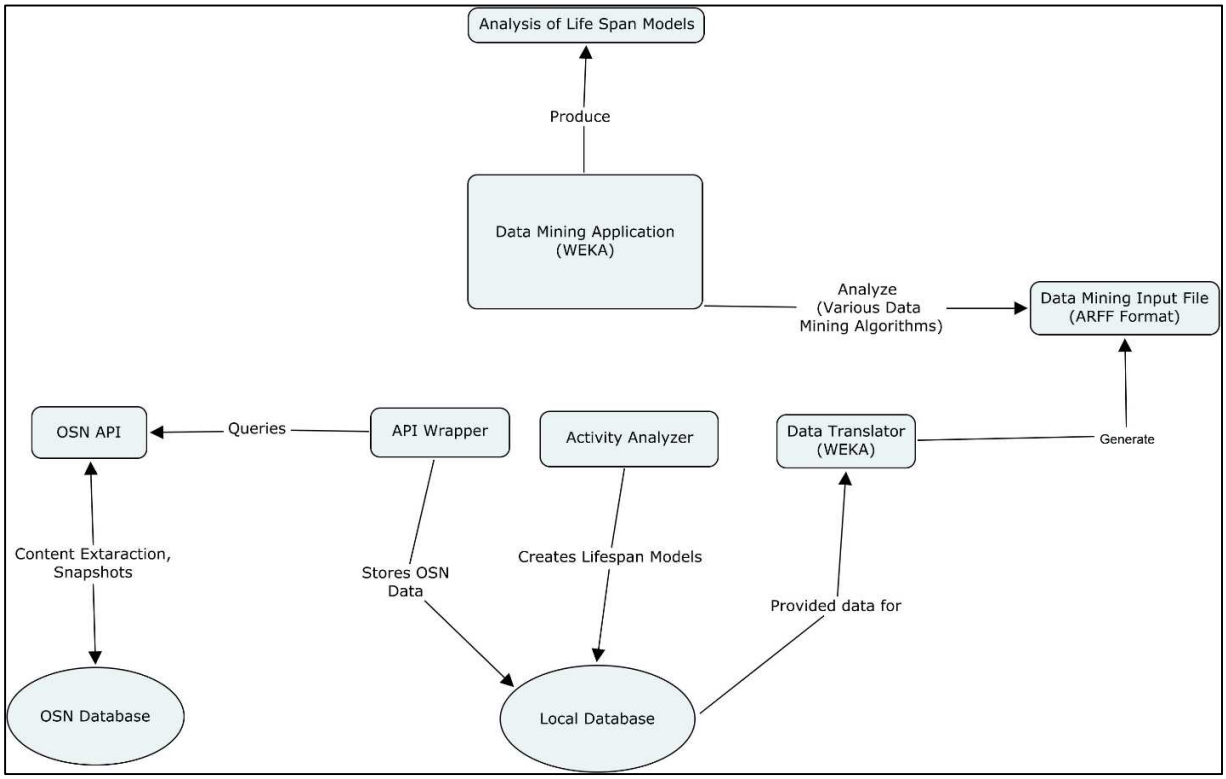


**Figure 4.1 Application Interactions**

Table 4.1 details the roles of the OSN database and the Local database.

| Application | Description |
|---|---|
| **OSN Database** | The database controlled and populated by each OSN. |
| **Local Database** | Stores content extracted from each OSN. |

**Table 4.1 Data Tier Applications.**

Table 4.2 details the roles the OSN API, API wrapper, data translator, activity analyzer, and data mining applications.

| Application | Description |
|---|---|
| **OSN API** | Interfaces to the OSN data. Maintained by each OSN. |
| **API Wrapper** | Handles OSN API querying, interaction, data extraction and storage. There is a wrapper for each OSN API. |
| **Data Translator** | Extracts data from the Local Database and create an input file that is readable by a desired data mining algorithm. The format is the ARFF format and is produced by the WEKA explorer. |
| **Activity Analyzer** | Analyzes the Local Database to measure content activity for the entire interval snapshots that were taken and generates the lifespan model (Popularity tier, Peak/Death) for the data mining input file |
| **Data Mining Application** | Applications analyze the data mining input file and produce rule sets, decision tress, or other forms of analytical output. |

**Table 4.2 Logic Tier Applications.**

Table 4.3 describes the Web interface, console interface, and lifespan model.

| Application | Description |
|---|---|
| **Analysis of Lifespan Model** | The analysis of lifespan models produced by a specific algorithm (Decision Table, Random Forest, etc.) |

**Table 4.3 Presentation Tier Applications.**

## 4.1 Data Acquisition

The following guidelines were used for data collection:

- Data must be collected via an OSN's API.
- All API rules (e.g., request limits) must be honored.
- Data must be publicly available, i.e., no data that are subject to privacy restrictions or relationship dependent access.

Some OSN APIs require registration before any data can be accessed. Table 4.4 details the request limits and registration requirements of the OSNs that were used. During the development, the YouTube API transitioned from version 2.0 to version 3.0. YouTube API v3.0 does not have a strict requests per second limitation, but rather assigns a unit value to each request – some requests being more expensive than others. For example, a video upload costs 1,600 units whereas a write request costs only 50 units (YouTube Data API, 2013).

| OSN | API Request Limits | Registration Required |
|---|---|---|
| **Reddit API** | 1 request per 2 seconds | Yes |
| **4Chan API** | 1 request per second | No |
| **Flickr API** | 1 request per second | Yes |
| **YouTube 3.0 API** | 30,000 units/user/second | Yes |

**Table 4.4 API limit and requirements.**

## 4.2 Discovering and Monitoring New Content

For Reddit, 4chan, and Flickr, the lifespan types of content were analyzed and categorized. To accomplish this, new content needed to be publicly discoverable and to be monitored on a regular interval. Different OSNs have different locations for new content. Figure 4.2 shows Reddit's page with its newest content.

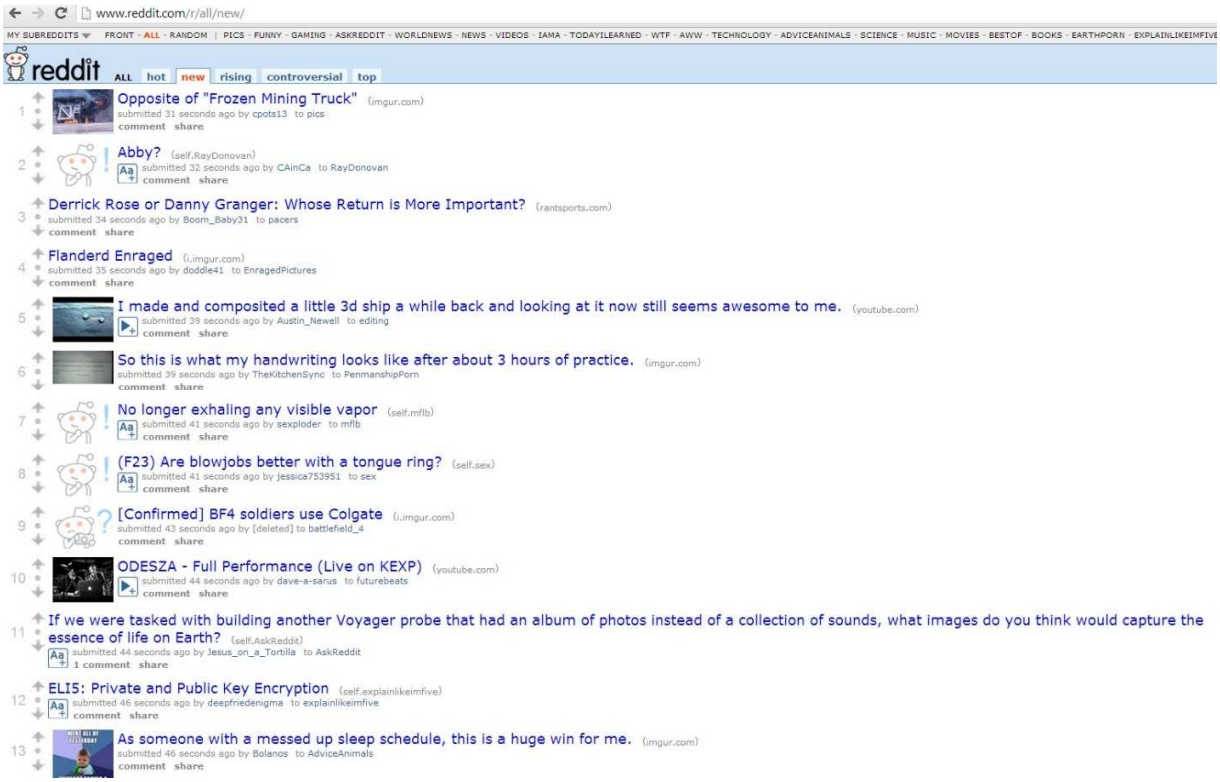


**Figure 4.2 Reddit /all/new: New content location (Reddit.com, 2013)**

Reddit funnels all new posts to the same location. It does not matter if a user is subscribed to a given subReddit or not, new content from all subReddits is posted to a subreddit named "/all/new" which can be accessed by the Reddit API (Github.com/reddit, 2012).

4chan puts all new posts on the front page of a given board, i.e., the content that is either the newest or most recently received a reply is always pushed to the front page. The 4chan API allows access to all posts currently active for a given board, and this collection of posts in known as the "catalog" (Github.com/4chan, 2013).

Flickr has a public centralized location for all new content, but it is only accessible from the API. The API can request a number (limit 1,000) of the most recently added pictures (Flickr API, 2009).

Figure 4.3 illustrates how a piece of data was monitored for a given OSN. When a new piece of content is added our API Wrapper retrieves the information about the newly added post from the OSN's API, and

stores to a “posts” table – this data in stored only once, at the beginning of the content’s lifespan. All information stored in the posts table is non-temporal. Once a post is added to the “posts” table it is then monitored on a regular interval for 24 hours, or until it is removed from the network. On regular intervals, a “snapshot” is taken, which involves our API Wrapper retrieving data from the OSN’s API. The snapshots of activity related data contain all the temporal data and are stored to the “Post Snapshots” table.

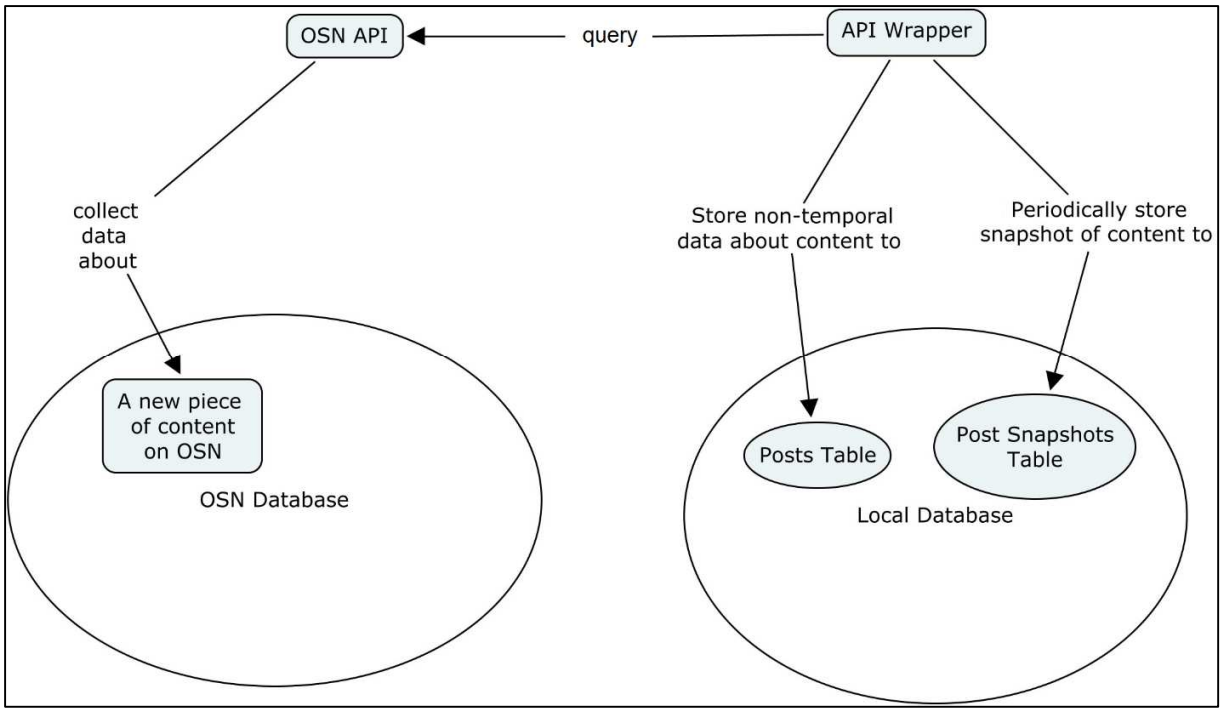


**Figure 4.3 Watching New Content**

The snapshot intervals are listed in Table 4.5. Reddit was the first OSN from which data were extracted, and it requires a delay of 2 seconds between requests. Using this delay, Equation 4.1 dictates the maximum number of requests per hour:

*(3600 sec/hour) / (2 sec/request)* ➔ *1800 requests/hour*

**Equation 4.1 Maximum Requests per Hour in Reddit.**

Each snapshot of a post requires a single request, i.e., a maximum of 1800 snapshots per hour. Keeping in mind Reddit's time-sensitive scoring algorithm that decays a score over time, making most posts effectively dead after 24 hours if not sooner (Github.com/reddit, 2013), we chose to take a snapshot every two hours. Flickr's time was set to follow suite with Reddit, though it does not have a time system decaying algorithm. Threads on 4chan can be removed in seconds do to inactivity (Berstein *et al.* 2011), so we took snapshots on a much shorter interval.

| OSN | Interval between snapshots |
|---|---|
| **Reddit** | 2 hours |
| **4chan** | 1 second |
| **Flickr** | 2 hours |

**Table 4.5 Minimum Snapshot Intervals Allowed By Each OSN.**

## 4.3 Software and Hardware Specifications

This section details the software and hardware specifications of the developed system.

The API wrapper and activity analyzer were implemented in PHP (PHP.net, 2009). The data received from the APIs were in the JSON format (JSON.org, 2011), which were then translated in the MySQL (MySQL.com, 2009) data base via a PHP API wrapper. WEKA (Hall *et al.*, 2009) was used to create the ARFF files that were used in experiments. Table 4.6 lists the different components and the language or application used to implement each components. The hardware specifications are detailed in Table 4.7.

| Component | Implementation |
|---|---|
| **API wrapper** | PHP |
| **Storages of snapshots and posts to database** | PHP/MySQL |
| **Local database** | MySQL |
| **Activity analysis (Popularity Tier and Life Span classification)** | PHP |
| **Translation of local data into ARFF format** | WEKA 3.6 |
| **Experimentation** | WEKA 3.6 |

**Table 4.6 Software Specifications.**

| CPU | **RAM** | **Operating System** |
|---|---|---|
| AMD FX-8120 Eight Core Processor (3.10GHz) | 16 GB | Windows 7 (64 bit Professional Edition) |

**Table 4.7 Hardware Specifications.**

# Chapter 5 Experimental Results

This chapter details the experiments and experimental results.

For Reddit, Flickr, and 4chan, the experimental processes can be summarized by the following sequence of steps:

1. Track the lifespans of a sample of content (a Reddit post, 4chan thread, or Flickr image upload).
2. Store the snapshots and content data locally.
3. For each piece of content.
    3.1. Analyze the activity.
    3.2. Classify the popularity tier and peak/death category.
4. Use WEKA to translate the processed content into the ARFF format.
    4.1. Edit the ARFF file to convert textual attributes to string (WEKA assigns these the nominal type by default, which treated each title as a unique value instead of a collection of words).
    4.2. Expand all words in the title or inner post in order to create a word vector. Convert all words to lowercase. Apply a list of stop words that disallows for any words in the words vector to be used as a column name if that column name was an attribute from the API (e.g., post_id would not be allowed because it was an attribute obtained via the Reddit API) along with a list of short function words (e.g., “the”, “as”, ect.) provided by WEKA.
    4.3. Create two ARFF files, one for classifying the popularity tier and one for classifying peak/death category.
5. Process the ARFF file using SMO, Decision Table, Random Forest, and Naïve Bayes through 10-Fold cross-validation.
6. Compare the accuracy and time required to build the model across all algorithms.
    6.1. Compare the results from 10-fold cross-validation
7. Use Student’s T-Testing to verify statistically significant differences.

In order build the word vector in WEKA, all of the strings were expanded to create new attribute columns. Table 5.1 shows example data from the local database after the activity analysis phase. In the first row, for example, a post that linked to imgur was titled "I love cats petting cats". Table 5.2 shows an example of what happens to the experimentation file when the string vector is generate. Our example post has the words from the title converted to lowercase and separated into individual attributes. Since the second post's title did not have all of the words that the first post's title did, it receives different values in each column, which represent how many times a given word appeared in the title.

| Domain | Title | Popularity Tier |
|---|---|---|
| **imgur.com** | I love cats petting cats | super_popular |
| **cnn.com** | Chocolate lovers will love this | average |

**Table 5.1 Example Reddit Data from Local Database.**

| Domain | i | love | cats | petting | chocolate | lovers | will | this | Popularity Tier |
|---|---|---|---|---|---|---|---|---|---|
| **imgur.com** | 1 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | super_popular |
| **cnn.com** | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | average |

**Table 5.2 Data in ARFF File After Converted to Word Vector.**

The experiments compared several different classification algorithms implemented in WEKA. The selected algorithms were SMO (Platt, 1998), a modification of Support Vector Machines, Decision Table (Kohavi, 1995), Naïve Bayes (John and Langley, 2009), and Random Forest (Breiman, 2001). Some

experiments using Decision Table did not complete. All completed experiments are included in the analysis.

## 5.1 Experimental Data

Table 5.3 lists the number of instances and attributes for each of the OSN's data sets. The attributes consist of the public attributes obtained from each OSN, along with the word vector generated from the collection of all words in any title or description of the content. An instance is single piece of content from an OSN (e.g., a post on Reddit or a video on YouTube). The number and attributes vary based on the number of words that happened be found in any piece of content for a given OSN (i.e., more unique words were used in the titles and descriptions of YouTube videos than Flickr images.

| OSN | Instances | Attributes |
|---|---|---|
| **Reddit** | 19,261 | 35,458 |
| **Flickr** | 28,800 | 74,214 |
| **4Chan** | 23,752 | 51,453 |
| **YouTube** | 29,999 | 147,246 |
| **YouTube (large batch)** | 299,999 | 889,176 |

**Table 5.3 OSN Experimental Data Sizes**

Table 5.4, Table 5.5, Table 5.6, and Table 5.7 list the attributes from each OSN that were used for experimentation. These attributes do no encompass every attribute from the OSN's API, but any attributes that were automatically generated and unique for each post (e.g., post ids) were removed before experimentation since those values were not chosen by the content's author and cannot be used again by a different post.

| Attribute | Description |
|---|---|
| **domain** | Domain of the content being linked to. |
| **subReddit** | Which subReddit the content was posted in. |
| **link_flair_text** | Acts as a label for a post, though it is not in the title or self-text. |
| **over_18** | Flagged as being for over 18 only. |
| **thumbnail** | The automatically generated thumbnail visible next to the post title. |
| **link_flair_css_class** | CSS class of the link flair. CSS is a language used to style HTML. |
| **author_flair_css_class** | CSS class of the author flair, which acts as a label for the author and is visible on all their posts within certain subReddits. |
| **is_self** | Self-posts are text posts. They do not link to an external domain. |
| **url** | The entire url of a link. |
| **author** | Name of the author of a post. |
| **word vector** | Each word from a title or self-text post converted into a unique attribute |

**Table 5.4 Reddit Experiment Attribute list.**

| Attribute | Description |
|---|---|
| **server** | Server number. |
| **farm** | Farm (server farm) number. |
| **license** | License category (8 categories total). |
| **safety_level** | Safety level number. |
| **rotation** | Degree of rotation of picture. |
| **originalformat** | Original format (file extention: jpg, png, etc.). |
| **username** | User name of the uploader. |
| **realname** | Real name of uploader (not required to upload). |
| **location** | Geographical location. |
| **iconserver** | Server id that stores the buddy icon, which is the thumbnail representing the uploader. |
| **iconfarm** | Server farm id of that stores the icon server. |
| **path_alias** | Optional alias of the username that appears in URLs to the user's uploads. |
| **haspeople** | Flag to indicate whether or not the picture contains people. |
| **word vector** | Each word from title or description converted into a unique attribute. |

**Table 5.5 Flickr Experiment Attribute List.**

| Attribute | Description |
|---|---|
| **filename** | File name of the original upload. |
| **ext** | File extension of the uploaded image. |
| **fsize** | File size of the image. |
| **w** | Width of the image. |
| **h** | Height of the image. |
| **tn_w** | Width of the thumbnail visible from the board page. |
| **tn_h** | Height of the thumbnail visible from the board page. |
| **word vector** | The words from the subject or original post converted into unique attributes. |

**Table 5.6 4Chan Experiment Attribute List.**

| Attribute | Description |
|---|---|
| **observation_delay** | The time between the video being posted and the time of it being observed. |
| **category** | Category of the video (e.g., Music, Games, Travel, etc.). |
| **content_type** | Type of video (flash or 3gpp). |
| **author** | Username of the author. |
| **duration** | Duration of the video in seconds. |
| **word vector** | The words from the video title and description converted into unique attributes. |

**Table 5.7 YouTube Experiment Attribute List.**

The data collected from each OSN were processed using the discussed methods. The experiments were ran for two different types of classes: peak/death category and popularity tier.

The content observed from each OSN contained different size samples of each category of lifespan. Table 5.8 lists the number of instances from each category and Figure 5.1 illustrates the differences in sizes. Youtube data were only used in the popularity tier experiments, which is why Youtube is not

present. One interesting point of discussion is the incredible lack of dead on arrival (DOA) posts for Reddit. The DOA category was defined as a piece of content experiencing no activity at all. Every subReddit has a "new" section, this sample suggests the ability to get users interaction with over 99% of content posted to the default subreddits. But, Figure 5.2 demonstrates that the amount of interaction with the majority of data is low. The majority of samples from 4chan and Flickr were DOA posts. In 4chan's case, it is not surprising since a post that is not interacted with very soon after its posting is moved further into the pages listing, decreasing the chance it will be interacted with until it is removed from the site, making it impossible to interact with. In Flickr's case this may be suggestive of the findings in other research, that Flickr posts can take days, weeks, or months to experience activity. We had anticipated this possibility, since our window of observation was only 24 hours, but we had not anticipated the large amount, over 10,000 posts both peaking and dying within in the first 12 hours of being posted. This suggests the possibility that a local maxima early in the content's life and further observation may indicate a true maximum later in life, or that the lifespan of images on Flickr may be briefer than previously observed.

| OSN | DOA | earlypeak_earlydeath | earlypeak_latedeath | latepeak_latedeath |
|---|---|---|---|---|
| **Reddit** | 8 | 12,385 | 5,593 | 1,275 |
| **4chan** | 16,813 | 6,066 | 275 | 598 |
| **Flickr** | 14,918 | 10,195 | 745 | 2,942 |

**Table 5.8 Life Span Categories Sample Sizes.**

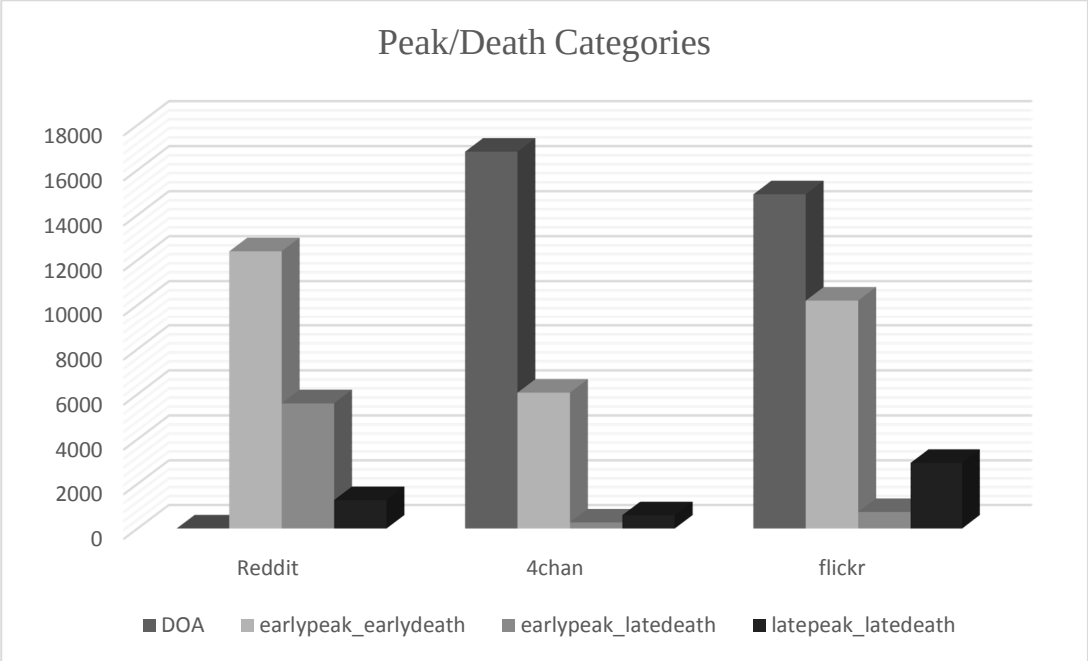


**Figure 5.1 Lifespan Categories Sample Sizes.**

The content observed from each OSN contained different size samples of each popularity tier. Table 5.9 lists the number of instances from each category, and Figure 5.2 illustrates the differences in sizes. Youtube is absent because its data was only used for Popularity Tier analysis. As anticipated, it was very rare to observe a super popular or viral tier piece of content. This causes a problem when trying to train a classifiers because these categories of interest are underrepresented. We considered using the few observed samples and resampling them in order to balance the number of samples across each category, but that plan was decided against since it would over represent the words used in these few posts. To accurately test for the attributes that are important to super popular or viral content, independently generated pieces of content would need to be observed.

| OSN | DOA | below average | average | popular | super popular | Viral |
|---|---|---|---|---|---|---|
| **Reddit** | 8 | 16,098 | 2,492 | 551 | 101 | 11 |
| **4chan** | 16,813 | 6,515 | 313 | 110 | 1 | 0 |
| **Flickr** | 14,918 | 10,401 | 3,363 | 116 | 2 | 0 |

**Table 5.9 Popularity Tier Sample Sizes.**

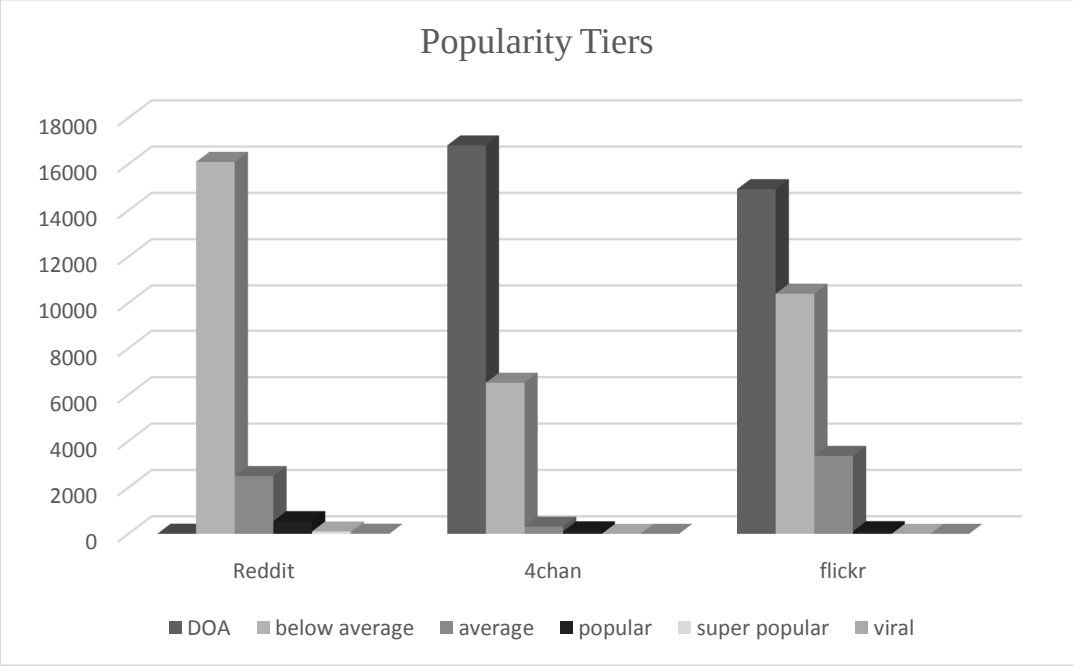


**Figure 5.2 Popularity Tier Sample Sizes.**

## 5.2 Peak/Death Category Experiments

The results from the experiments classifying lifespan categories are evaluated based on two criteria: accuracy and run time. All accuracies and runtimes in this section are the result of 10-fold cross-validation. Table 5.10 lists the runtimes, in seconds, and Figure 5.3 illustrates the differences across OSNs and classifiers. The SMO classifier consistently outperformed all other classifiers in speed, ranging from ~1.5 times faster that Naïve Bayes up to ~36 times faster than the Decision Table. Decision Table's

exhaustive components appear to be a large factor in its runtime, always requiring more time than any other algorithm. The Decision Table experiments for the Flickr data did not finish before a system crash. The longest attempt at finishing experiments was over two months. A more stable or robust system is needed to complete those experiments.

| OSN | SMO | Naïve Bayes | Random Forest | Decision Table |
|---|---|---|---|---|
| **Reddit** | 3,092.53 | 4,763.46 | 9,063.01 | 70,851.44 |
| **Flickr** | 10,506.5 | 25,002.67 | 15,936.73 | unfinished |
| **4chan** | 4,067.54 | 11,350.82 | 11,888.89 | 147,386.04 |

**Table 5.10 Peak/Death Results Time Comparisons (seconds).**

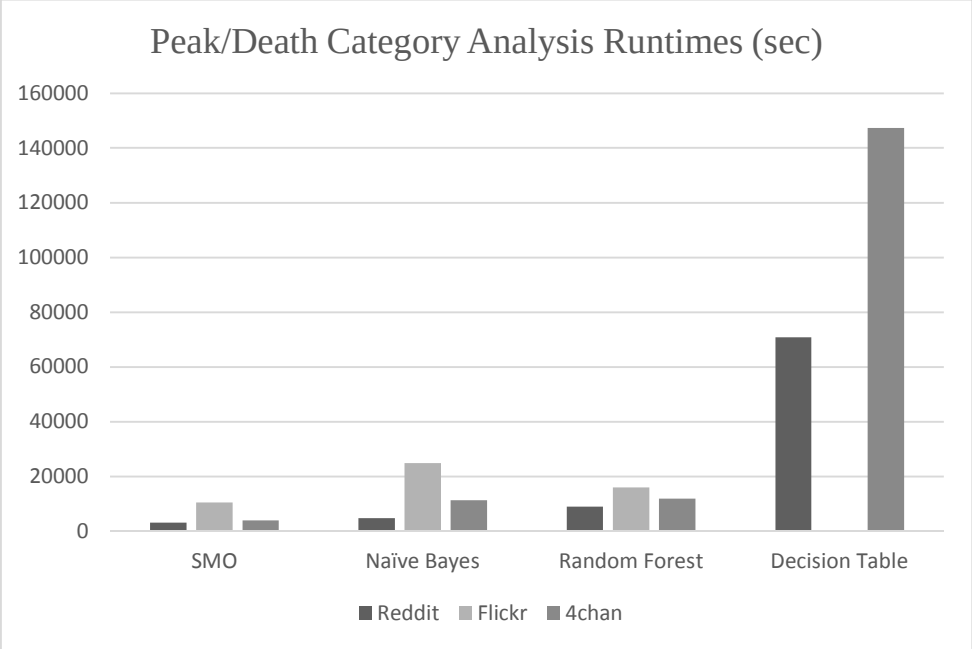


**Figure 5.3 Lifespan Analysis Runtimes (seconds).**

Table 5.11 lists the accuracies from the 10-fold cross-validation and Figure 5.4 shows the differences across OSNs and classifiers. The accuracy of SMO and Random Forest stay above 60% across OSNs, but never break 70%. Naïve Bayes has the largest range of accuracies, including the lowest and highest

accuracies, overall. Naïve Bayes also appears to be best suited for 4chan's image boards, but approaches coin-flip accuracy for Flickr content. Decision Table, while having some of the highest accuracies, also takes up to 35 times more runtime for only a four percent accuracy improvement from SMO and a six percent decrease from Naïve Bayes.

| OSN | SMO | Naïve Bayes | Random Forest | Decision Table |
|---|---|---|---|---|
| **Reddit** | 61.83% | 58.19% | 64.21% | 64.35% |
| **Flickr** | 65.46% | 52.21% | 62.69% | unfinished |
| **4chan** | 66.57% | 76.79% | 69.09% | 70.40% |

**Table 5.11 Peak/Death Testing Accuracy Percentages.**

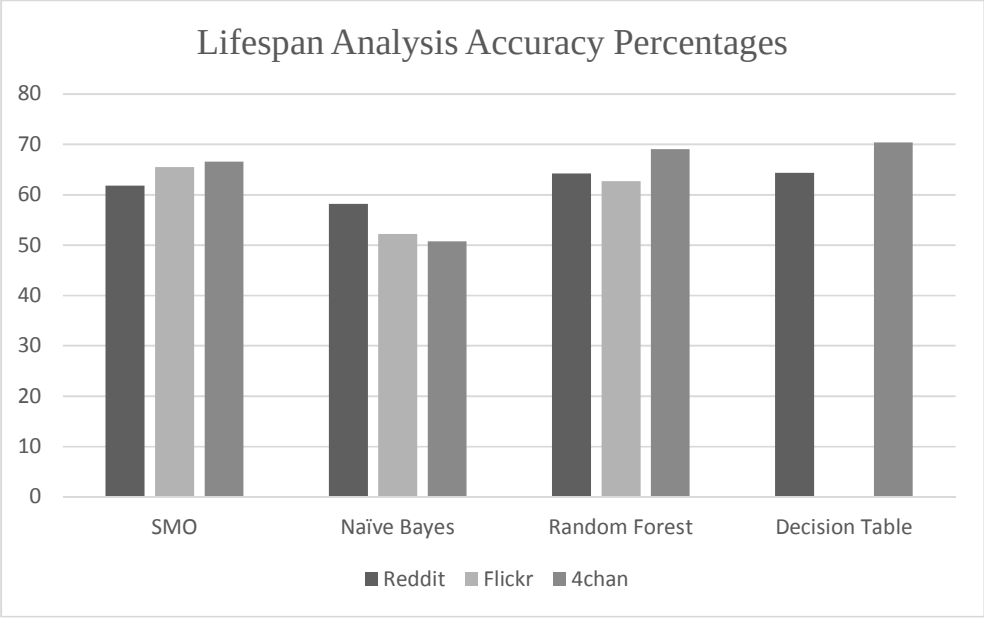


**Figure 5.4 Peak/Death Analysis Accuracy Percentages.**

Along with accuracy, percentage of correctly classified instances out of all instances, for the entire data sets, we also evaluated each classifier's accuracy for individual classes. This was done to investigate the difficulty of correctly classifying some of the underrepresented classes. The complete confusion matrices

from the 10-fold cross-validations can be found in the Appendix. Table 5.12 listed the accuracy each classifier had for all lifespan categories from the Reddit dataset. As expected, the category with the majority of samples, earlypeak_earlydeath, had the highest accuracy for all classifiers. The dead on arrival category, DOA, had the fewest samples and consequentially none of the classifiers were able to correctly classify a single instance. Naïve Bayes demonstrated a very interesting behavior, having the highest accuracy - though only 50% - in the earlypeak_latedeath category. All other classifiers struggled with this category, demonstrated an increase in accuracy when classifying earlypeak_earlydeath, again with the majority of samples. This increase in accuracy was not reflected by Naïve Bayes only increasing to 67.5% while all other the classifiers were 80% or higher. This behavior is also demonstrated in Table 5.13, listing the accuracy by category for 4chan. However, this behavior is not repeated in the Flickr data, found in Table 5.14. In the Flickr dataset Naïve Bayes, Random Forest, and SMO demonstrate the ability to classify each category with similar capability; granted, there are gaps as large as 20% in accuracy. However, when looking at the distribution of accuracies for Reddit and 4chan compared to Flickr, Flickr has non-zero accuracies in all categories. Across all OSNs and classifiers only the category with the majority of samples scored about 56% accuracy.

| | Percent of Data Set | Naïve Bayes | Random Forest | Decision Table | SMO |
|---|---|---|---|---|---|
| **earlypeak_latedeath** | 29.04% | 50.12% | 8.62% | 2.63% | 32.74% |
| **earlypeak_earlydeath** | 63.30% | 67.50% | 95.88% | 98.89% | 81.04% |
| **latepeak_latedeath** | 6.62% | 3.45% | 0.86% | 0.00% | 3.22% |
| **DOA** | .04% | 0.00% | 0.00% | 0.00% | 0.00% |

**Table 5.12 Accuracy by Class - Reddit.**

| | Percent of Data Set | Naïve Bayes | Random Forest | Decision Table | SMO |
|---|---|---|---|---|---|
| **earlypeak_earlydeath** | 1.16% | 44.58% | 7.47% | 1.04% | 15.33% |
| **earlypeak_latedeath** | 25.54% | 3.64% | 0.00% | 0.00% | 0.36% |
| **latepeak_latedeath** | 2.52% | 7.53% | 2.51% | 0.84% | 3.01% |
| **DOA** | 70.79% | 55.29% | 94.81% | 99.05% | 88.40% |

**Table 5.13 Accuracy by Class - 4chan.**

| | Percent of Data Set | Naïve Bayes | Random Forest | Decision Table | SMO |
|---|---|---|---|---|---|
| **earlypeak_earlydeath** | 2.59% | 38.31% | 38.94% | NA | 57.84% |
| **earlypeak_latedeath** | 35.40% | 5.23% | 4.70% | NA | 6.17% |
| **latepeak_latedeath** | 10.22% | 23.73% | 15.77% | NA | 21.82% |
| **DOA** | 51.80% | 83.21% | 91.07% | NA | 82.23% |

**Table 5.14 Accuracy by Class - Flickr.**

## 5.3 Popularity Tier Experiments

The results from the experiments classifying lifespan were evaluated on two criteria: accuracy and run time. All accuracies and runtimes in this section are the result of 10-fold cross-validation. Table 5.15 lists the runtimes, in seconds, each classifier required for a given OSN. The experiments in this section do not include the data from YouTube. Those results are discussed later. Figure 5.5 illustrates the differences in runtimes between classifiers. Similar to the lifespan analysis, SMO had the best observed results in speed. Decision Table once again took longer than any other classifier by several factors ranging from a factor of ~3.8 slower than Naïve Bayes on Reddit data to ~19 slower than SMO on 4chan data.

| OSN | SMO | Naïve Bayes | Random Forest | Decision Table |
|---|---|---|---|---|
| **Reddit** | 1,414.03 | 6,246.82 | 4,145.33 | 24,184.84 |
| **Flickr** | 8,668.68 | 20,502.76 | 12,607.17 | unfinished |
| **4chan** | 3,275.51 | 11,350.82 | 14,411.95 | 62,954.84 |

**Table 5.15 Popularity Tier Results Time Comparisons (seconds).**

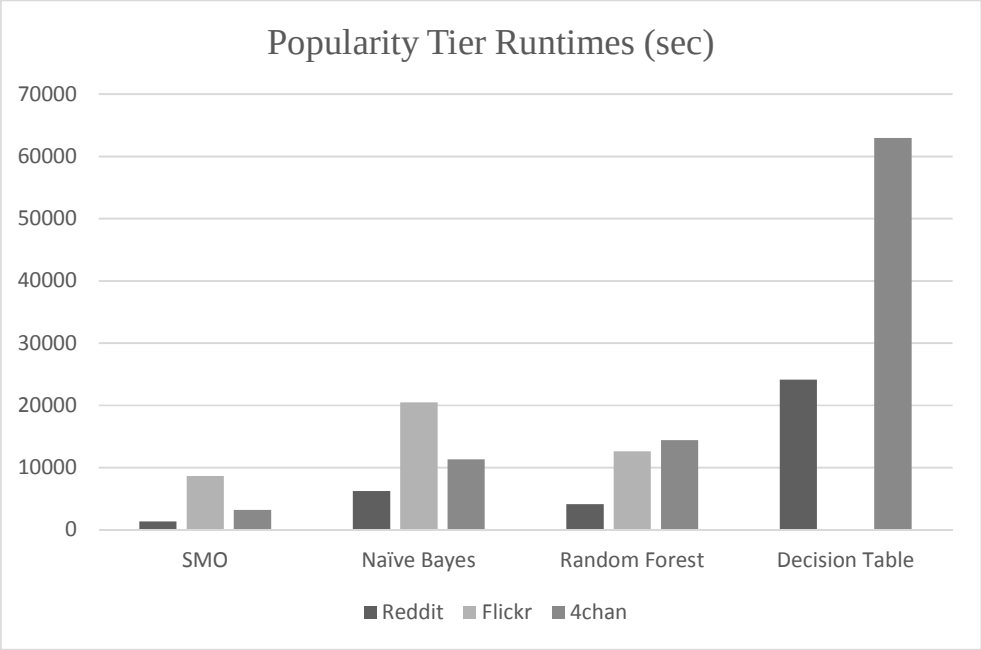


**Figure 5.5 Popularity Tier Runtimes (seconds).**

Table 5.16 lists the accuracies for each classifier across the different OSNs. For all classifiers, Reddit yielded the highest accuracies, the statistical significance of this will be discussed in later section. One might assume that the built-in decay over time, which puts the majority of post into the same tier, influenced the results. However, it is critical to note that the accuracies for 4chan, which has the most drastic and fastest acting decay over time mechanic, went down across the board by at least 14%. Despite the majority of content for both Reddit and 4chan belonging to a single tier and both having a time-sensitive criteria for their content, all classifiers were observed to more accurately classify Reddit content. This discrepancy may be caused by the larger number of attributes in the 4chan dataset, or the slightly more even distribution of popularity tiers. This theory will be further discussed in a later section. Whatever caused the drop in accuracy, it caused Naïve Bayes to drop nearly to 50% accuracy.

| OSN | SMO | Naïve Bayes | Random Forest | Decision Table |
|---|---|---|---|---|
| **Reddit** | 83.06% | 71.25% | 83.53% | 84.01% |
| **Flickr** | 66.49% | 59.37% | 61.80% | NA |
| **4chan** | 66.94% | 52.21% | 68.93% | 70.54% |

**Table 5.16 Popularity Tier Accuracy Percentages.**

Just as we did with lifespan categories, we broke down the accuracies of each classifier by category. The complete confusion matrices for all classifiers and OSNs are included in the Appendix. The accuracies for each popularity tier category of Reddit are listed in Table 5.17. Behaviors similar to those previously discussed are present. The category with the majority of the samples has the highest accuracy across all classifiers. Also, Naïve Bayes again has lower but more evenly distributed accuracies; where the other classifiers have a 90%+ accuracy in a single category and 14% or less in all other. The viral, super popular, and dead on arrival (DOA) categories were too underrepresented to be accurately classified.

| | Percent of Data Set | Naïve Bayes | Random Forest | Decision Table | SMO |
|---|---|---|---|---|---|
| **DOA** | 0.04% | 0.00% | 0.00% | 0.00% | 0.00% |
| **BELOWAVG** | 83.58% | 79.99% | 99.61% | 99.80% | 97.15% |
| **AVG** | 12.94% | 33.67% | 2.05% | 4.53% | 13.56% |
| **POPULAR** | 2.86% | 1.27% | 0.36% | 0.54% | 3.63% |
| **SUPERPOPULAR** | 0.52% | 0.00% | 0.00% | 0.00% | 0.00% |
| **VIRAL** | 0.06% | 0.00% | 0.00% | 0.00% | 0.00% |

**Table 5.17 Popularity Tier Accuracy by Category - Reddit.**

The accuracy result for each popularity tier of the 4chan dataset are listed in Table 5.18. Naïve Bayes DOA accuracy took a dramatic hit, dropping nearly 25%. Naïve Bayes performed the poorest in 4chan overall in both lifespan and popularity tier experiments. Decision Table obtained the highest accuracy on the 4chan data, but it failed to correctly categorize the underrepresented categories of super popular, viral, and DOA.

| | Percent of Data Set | Naïve Bayes | Random Forest | Decision Table | SMO |
|---|---|---|---|---|---|
| **DOA** | 70.79% | 55.52% | 93.87% | 99.82% | 87.66% |
| **BELOWAVG** | 27.43% | 46.62% | 8.92% | 1.69% | 17.57% |
| **AVG** | 1.32% | 9.27% | 2.56% | 2.24% | 5.11% |
| **POPULAR** | 0.46% | 1.82% | 0.00% | 0.00% | 0.00% |
| **SUPERPOPULAR** | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| **VIRAL** | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |

**Table 5.18 Popularity Tier Accuracy by Category - 4chan.**

| | Percent of Data Set | Naïve Bayes | Random Forest | Decision Table | SMO |
|---|---|---|---|---|---|
| **DOA** | 51.80% | 91.65% | 90.47% | NA | 83.56% |
| **BELOWAVG** | 36.11% | 63.01% | 34.57% | NA | 51.80% |
| **AVG** | 11.68% | 61.31% | 20.64% | NA | 37.85% |
| **POPULAR** | 0.40% | 39.66% | 10.34% | NA | 19.83% |
| **SUPERPOPULAR** | 0.01% | 100.00% | 0.00% | NA | 0.00% |
| **VIRAL** | 0.00% | 0.00% | 0.00% | NA | 0.00% |

**Table 5.19 Popularity Tier Accuracy by Category - Flickr.**

The accuracy of each popularity tier by category for Flickr are listed in Table 5.19. Just as before, the Flickr data have a more even distribution of accuracies when compared with Reddit or 4chan. Interestingly, Naïve Bayes, with the worst accuracy overall, was the only classifier to correctly classify the super popular category. Not only that, it had 100% accuracy. This is interesting, but it is important to note that there were only two instances of the super popular category in Flickr, so Naïve Bayes was able to classify two out of two. More analysis would be needed to draw a conclusion.

## 5.4 YouTube Experiments

The data collected from YouTube was not taken from newly posted videos, but instead videos that were a minimum of a year old in order to allow the video's lifespan to complete. Since data were not collected in real time, only the popularity tier analysis was performed. Table 5.20 and Table 5.21 shows the time required to build the model and accuracy of each classifier, respectively. Although the YouTube

experiments did not include an analysis of lifespan categorization, the YouTube data proved valuable in learning the effects of increasing attributes sizes on the accuracies and runtimes of the classifiers. One additional note is that the depth of the Random Forest classifier was changed from the default of "unlimited" and limited to 1000 due to memory limitations - unlimited depth caused "heap out of memory" errors.

| OSN | SMO | Naïve Bayes | Random Forest | Decision Table |
|---|---|---|---|---|
| **YouTube** | 7,132.46 | 49,178.81 | 30,832.67 | NA |

**Table 5.20 YouTube Popularity Tier Analysis Runtimes (seconds).**

Table 5.20 shows that SMO still has the lowest runtime, demonstrating excellent scaling with an increase in attribute and sample sizes. Figure 5.6 further illustrates SMO's excellent runtime performances, maintaining the behavior of a logarithmic runtime, while Naïve Bayes and Random Forest were increasing in linear fashion.

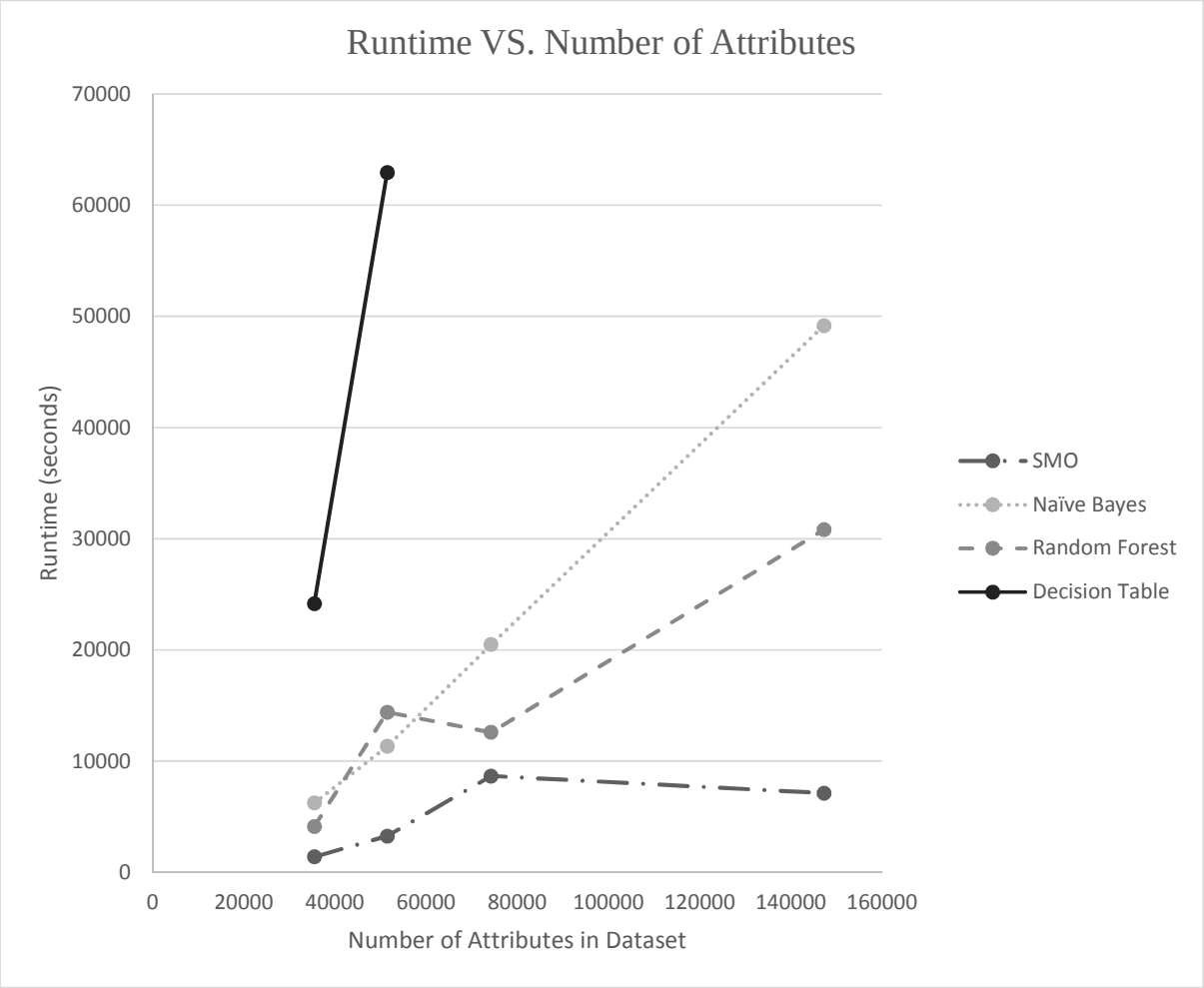


**Figure 5.6 Runtime VS. Number of Attributes.**

| OSN | SMO | Naïve Bayes | Random Forest | Decision Table |
|---|---|---|---|---|
| **YouTube** | 48.19% | 29.72% | 41.49% | unfinished |

**Table 5.21 YouTube Popularity Tier Accuracy Percentages.**

Table 5.21 shows a large decrease in classification accuracy across for all classifiers. All classifiers are below 50% accurate, making them worse than a coin flip. Figure 5.7 illustrates the effect of accuracy as the number attributes increased. All classifiers demonstrate a drop in accuracy when the number of

attributes increase. The increase in attributes is caused by the increase in words taken from the video titles and descriptions, each word being a single attribute. This increase in attributes increased how sparse the dataset was since a given video would contain a very small percentage of all words collected from YouTube videos.

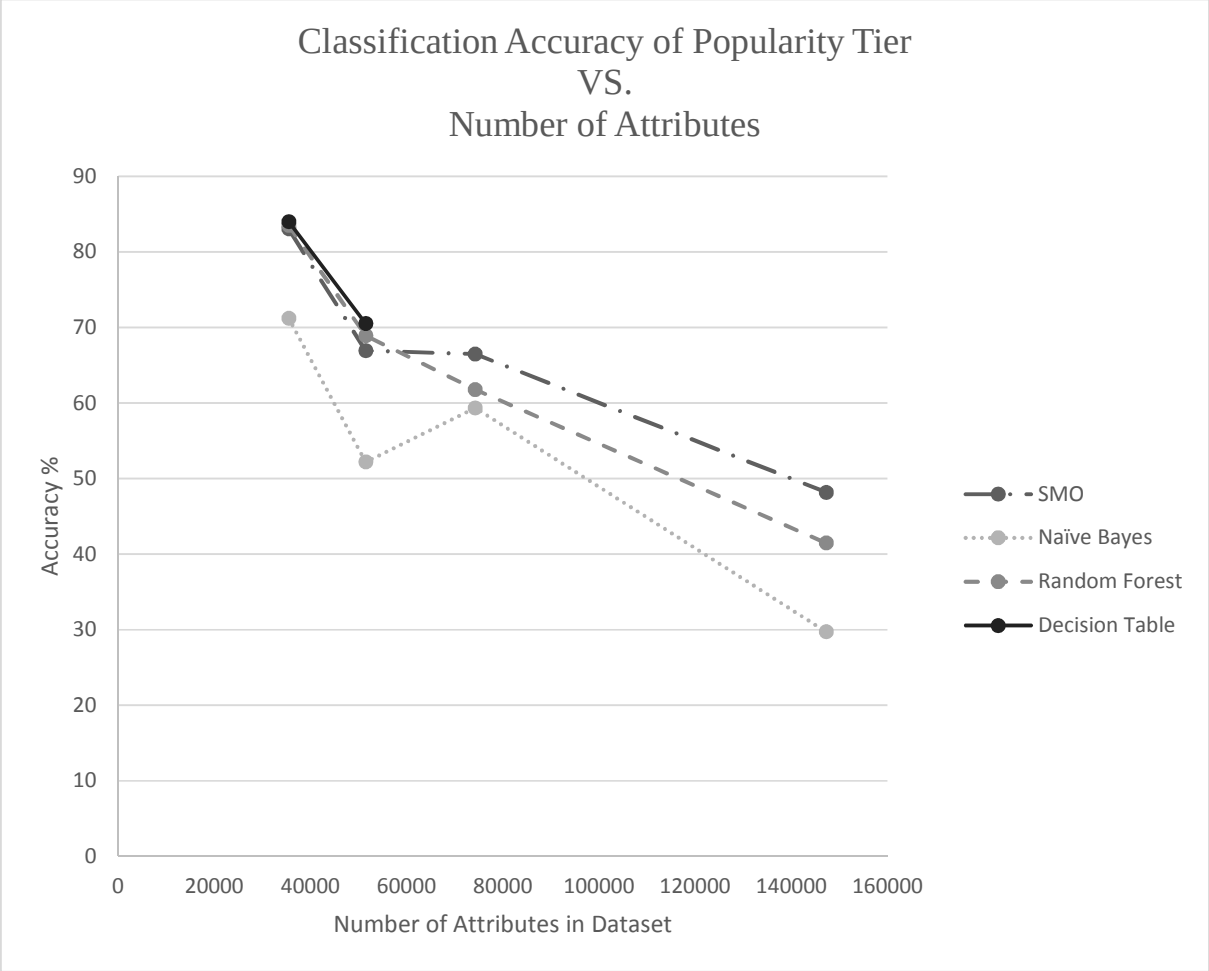


**Figure 5.7 Classification Accuracy of Popularity Tiers VS Number of Attributes.**

Table 5.22 lists the accuracies of each classifier for each category of popularity tier. The distribution of accuracy percentages is more even than the other OSNs, but that is most likely because the sample sizes of each category are more even. This suggests that if enough samples can be collected of super popular or viral data, they can be classified just as well as any other category. But, despite the accuracies being more

evenly distributed, none of the classifiers performed higher than 66.5% in any tier. The highest scored by SMO in the average tier.

| | Naïve Bayes | Random Forest | Decision Table | SMO |
|---|---|---|---|---|
| **DOA** | 13.95% | 8.43% | NA | 16.28% |
| **BELOWAVG** | 44.23% | 6.36% | NA | 13.49% |
| **AVG** | 59.40% | 59.03% | NA | 66.50% |
| **POPULAR** | 14.01% | 53.52% | NA | 51.09% |
| **SUPERPOPULAR** | 11.21% | 18.54% | NA | 30.80% |
| **VIRAL** | 31.57% | 29.41% | NA | 45.88% |

**Table 5.22 Popularity Tier Accuracy by Category - YouTube.**

YouTube's dataset contained the largest number of attributes, but the size of the data set, ~30k instances, was still comparable in size to the other datasets. Though the trend indicates that the number of attributes affect the accuracy of classifiers, we believe that the severe drop in accuracy is also a result of the type of content on YouTube, namely, videos. The video itself was not analyzed, only the publicly available textual information. Also, YouTube videos can be posted to blogs, news sites, and other Websites without the viewer being exposed to the majority of the textual content surrounding the video, such as the description. Portions or the entire video's title may not be viewable when embedded due to the small size of the video's iframe (an element of a Webpage that external resources can be displayed in). After the video is watched, links to other videos are displayed. So, it is possible for a user to watch a YouTube video without seeing a single piece of the text that was used to classify it.

The experiment on the YouTube set with 300k instances ran from August 7th 2013 to October 30th 2013, when the system crashed, without completing. The time to generate the model or what fold of the k-fold validation was reached is unknown.

## 5.5 Statistical Analysis

Student T-Tests were performed to analyze three categories:

(1) Compare each data mining algorithm's accuracy across all OSNs and all classification types.

(2) Evaluate each OSN based on the number of misclassified instances for each OSN.

(3) Compare the two model types, Peak/Death category and Popularity Tier, based on misclassified instances across all OSNs and all algorithms.

Every T-Test assumed no difference in the data sets' variances as the null hypothesis. A P-Value of 0.05 would allow the rejection of the null hypothesis with 95% confidence.

### 5.5.1 Algorithm Misclassifications Comparisons

To perform the first test, every misclassification from the confusion matrices, included in Appendix A, was placed into a vector of misclassifications for a given algorithm. For example, all of the misclassifications for Naïve Bayes across all OSNs and categories (lifespan classes and popularity tier classes) were taken from the confusion matrices and put into a single vector. This process was repeated for SMO, Decision Table, and Random Forest. Student's T-Tests were performed in a pairwise manner, testing two algorithms at a time. Table 5.23 lists the P-Values from the T-Tests comparing the means of misclassified instances of two algorithms. Table 5.24 lists the means of misclassified instances for each algorithm across all OSNs and categories (i.e., both lifespan and popularity tier misclassifications are represented in the means), which are being evaluated as being different in statistically significant way. For example, the first row lists the P-Values, the probabilities that the differences in two sets are due to chance alone, for Naïve Bayes compared to SMO, Decision Table, and Random Forest. The P-Value for Naïve Bayes tested with SMO is 0.42, indicating that there is no statistically significant difference and that the variations in accuracies are probably due to chance. None of the P-Values being lower than 0.05

indicates that there is no statistically significant difference in any algorithm's mean of misclassified instances across all OSNs.

| | SMO | Decision Table | Random Forest |
|---|---|---|---|
| **Naïve Bayes** | 0.42 | 0.17 | 0.55 |
| **SMO** | | 0.43 | 0.89 |
| **Decision Table** | | | 0.40 |
| **Random Forest** | | | |

**Table 5.23 P-Values from Pairwise T-Test (alpha = 0.05) of Misclassifications Across all OSNs and Categories.**

| SMO | Naïve Bayes | Random Forest | Decision Table |
|---|---|---|---|
| 452.57 | 558.29 | 471.23 | 321.99 |

**Table 5.24 Means of Misclassified Instances Across all OSNs and Categories.**

### 5.5.2 OSN Misclassification Comparisons

In order to compare each OSN to another OSN in terms of misclassified instances, all of the misclassifications from the confusion matrices, included in Appendix A, were placed into a single vector for a given OSN. For example, every value from the confusion matrices that represented a misclassification of Reddit instances was placed into a vector that would contain all misclassifications for Reddit. This process was repeated for 4chan, Flickr, and YouTube. These vectors were then compared in a pairwise manner using the Student's T-Test. We assumed that there were no difference in the variances for each pairwise comparison as our null hypothesis. Table 5.25 lists the P-Values for all pairwise comparisons. All P-Values above 0.05 indicate no statistically significant difference between the misclassification means for the OSNs, which are listed in Table 5.26. These P-Values indicate no statistically significant difference between 4chan, Flickr, or YouTube in any combination. Overall, Reddit had the lowest mean of misclassified instances, namely, 263.20. When compared to Flickr and YouTube the P-Values indicate, with 95% confidence, that there is statistically significant difference

between Reddit and these OSNs, indicating that a post on Reddit is in fact less likely to be misclassified by any of the algorithms that were selected. The P-Value for the pairwise comparison of Reddit and 4chan is 0.07, which is approximately 0.02 above the threshold. Though the maximum P-Value of 0.05 needed for 95% confidence was exceeded, we can say that there is a minimum of a 90% probability that the Reddit mean misclassification score was not lower by chance, but Reddit posts are less likely to be misclassified than 4chan posts. This suggests that the model we created requires adjustments before being deployed on a different OSN in future experiments.

| | 4Chan | Flickr | YouTube |
|---|---|---|---|
| **Reddit** | 0.07 | 0.01 | 0.004 |
| **4chan** | | 0.61 | 0.65 |
| **Flickr** | | | 0.91 |
| **YouTube** | | | |

**Table 5.25 P-Values from Pairwise T-Test (alpha = 0.05) of Misclassifications Across all Algorithms and Categories.**

| Reddit | 4chan | Flickr | YouTube |
|---|---|---|---|
| 263.20 | 526.91 | 620.33 | 601.97 |

**Table 5.26 Means of Misclassified Instances Across all Algorithms and Categories.**

### 5.5.3 Life Span and Popularity Tier Misclassification Comparisons

The final statistical analysis was to compare the two models that were developed, namely, peak/death category and popularity tier, and to determine which model is more likely to be misclassified. To compare the two models, all of the misclassification for each were taken from the confusion matrices, included in Appendix A, and put into a separate vectors. This generated two vectors, one containing all the misclassification values for all Peak/Death models and one vector containing all the misclassifications for the Popularity Tier models. The complete table of T-Test data can be found in Appendix B. The

mean value for misclassification for all lifespan models was 314.52. The mean value for misclassifications for all popularity models was 674.72. The P-Value generated by the T-Test was 0.001, indicating with 99.9% confidence that the null hypothesis is rejected and these two sets are, statistically, significantly different. This implies that the lifespan of content from Reddit, 4chan, and Flickr (YouTube was not used in lifespan analysis) is less likely to be misclassified than the popularity tier. This result may be the result of two of our OSNs, namely, Reddit and 4chan, have built in mechanism that limit how long content stays in a place where it can be interacted with easily unless it is gaining in popularity. The popularity tier may benefit from analysis of more than the publicly available text-based information (i.e., analyzing the content of a YouTube video rather than just the text-based data surrounding it).

## 5.6 Results

We generated a model for content lifespan broken into two categories—peak/death category and popularity tier—for the modeling of content on Reddit, 4chan, Flickr, and YouTube. For each category of model there were difficulties in dealing with underrepresentation of certain classes, but our statistical analysis showed, with 95%, confidence that the peak/death category of content, for the selected OSNs, is less likely to be misclassified. Using 10-fold cross-validation, we evaluated the accuracy in which the peak/death category of content can be classified. The peak/death category of Reddit content can be classified with 64% accuracy. The peak/death category of 4Chan content can be classified with 76% accuracy. The peak/death category of Flickr content can be classified with 65% accuracy. We also used 10-fold cross-validation to measure the accuracy in which the popularity tier of content can be classified. The popularity tier of content on Reddit can be classified with 84% accuracy. The popularity tier of content on 4chan can be classified with 70% accuracy. The popularity tier of content on Flickr can be classified with 66% accuracy. The popularity tier of content on YouTube can be classified with only 48% accuracy.

Our experiments compared the runtimes and accuracy of SMO, Naïve Bayes, Decision Table, and Random Forest to classify the lifespan of content on Reddit, 4chan, and Flickr as well as to classify the popularity tier of content on Reddit, 4chan, Flickr, and YouTube. The experimental results indicate that SMO is capable of outperforming the other algorithms in runtime across all OSNs. Decision Table had the longest observed runtimes, failing to complete analysis before system crashes in some cases. The statistical analysis indicates, with 95% confidence, that there is no statistically significant difference in accuracy between the data mining algorithms across all OSNs. Reddit content was shown, with 95% confidence, to be the OSN least likely to be misclassified. All other OSNs, were shown to have no statistically significant difference in terms of their content being more or less likely to be misclassified when compared pairwise with one another.

## 5.7 Hypothesis Evaluation

Our hypothesis of applying data mining techniques to data collected from online social networks, and producing a model that can categorize the peak/death category and popularity tier of content on a social network, which can then be used predictively, is partially confirmed. The accuracy of the models across the different classifiers is not high enough to warrant a sweeping confirmation, but resulted in individual categories of certain OSNs, such as Reddit and 4chan, showing that certain categories of data can be modeled and predicted with satisfactory accuracy. The need for more samples of the underrepresented categories in combination with improved experimentation techniques could lead to better results. In other words, we did not discover a magic bullet for prediction, but we did find a valuable place to begin the search.

# Chapter 6 Conclusion

## 6.1 Contributions

In this work, we obtained publicly available data from Reddit, 4chan, Flickr, and YouTube via their APIs. For Reddit, 4chan, and Flickr we observed the data in real time immediately after publication for 24 hours or until the content was removed. Snapshots of the content were taken on regular intervals in order to monitor changes in activity with a given piece of content. We generated models that were used to classify the lifespan types and popularity tiers across multiple OSNs. After monitoring the content we then analyzed the data in order to classify the type of lifespan for Reddit, 4chan, and Flickr data. For Reddit, 4chan, Flickr, and YouTube we analyzed the different tiers of popularity that the samples reached.

We generated a model for content lifespan broken into two categories, peak/death category and popularity tier for the modeling of content on Reddit, 4chan, Flickr, and YouTube. For each category of model there were difficulties in dealing with underrepresentation of certain classes, but our statistical analysis showed, with 95%, confidence that the lifespan of content, for the selected OSNs, is less likely to be misclassified. Using 10-fold cross-validation, we evaluated the accuracy in which the peak/death category of content be classified. The peak/death category of Reddit content can be classified with 64% accuracy. The peak/death category of 4Chan content can be classified with 76% accuracy. The peak/death category of Flickr content can classified with 65% accuracy. We also used 10-fold cross-validation to measure the accuracy in which the popularity tier of content can be classified. The popularity tier of content on Reddit can be classified with 84% accuracy. The popularity tier of content on 4chan can be classified with 70% accuracy. The popularity tier of content on Flickr can be classified with 66% accuracy. The popularity tier of content on YouTube can be classified with only 48% accuracy.

Our experiments compared the runtimes and accuracy of SMO, Naïve Bayes, Decision Table, and Random Forest to classify the lifespan of content on Reddit, 4chan, and Flickr as well as classify the

popularity tier of content on Reddit, 4chan, Flickr, and YouTube. The experimental results indicate that SMO is capable of outperforming the other algorithms in runtime across all OSNs. Decision Table had the longest observed runtimes, failing to complete analysis before a system crash in some cases. The statistical analysis indicates, with 95% confidence, there is no statistically significant difference in accuracy between the algorithms across all OSNs. Reddit content was shown, with 95% confidence, to be the OSN least likely to be misclassified. All other OSNs, were shown to have no statistically significant difference in terms on their content being more or less likely to be misclassified when compared pairwise with each other.

Another noteworthy find was that on Reddit nearly every single piece of content is interacted with by at least one other user than its author, causing less that 1% of its data to be dead on arrival. For a summary of this work please see Gibbons and Agah (2014).

## 6.2 Limitations and Issues

This section will discuss the limitations and issues encountered during the course of the research. The biggest issues encountered were API limitations and data acquisition and experimentation time.

### 6.2.1 OSN API Limitations

One goal of our research was to ensure that the data sampled were not biased in anyway. This proved to be difficult with some of the OSN's APIs originally selected for research. Twitter, Facebook, and Google Plus were all potential candidates for research because of their vast user base and worldwide influence. They all share the similar problem of a forced perspective, meaning every user's experience is influenced by their social graph. For example if user-A is friends with the following people X, Y, and Z, he/she will encounter a different news/twitter/post feed than a user who does not share a link with those people. In other words, these social networks lack a single source of content that is uninfluenced by social links. Twitter does have an access point like this, called the "fire hose", but access to it must be approved by

twitter (Singletary, 2012). All inquiries that were made into accessing the Twitter fire hose received no replies. This issue led us to pick OSNs with a public, unbiased access point to content as soon as it is posted to the site. Anyone who goes to Reddit's feed of new posts, a board on 4chan, or uses Flickr's API to access the most recently uploaded photos, will have the same level of access to the same content regardless of who is or who is not in his/her social graph.

### 6.2.2 Data Acquisition and Experimentation Time

The most significant time bottleneck during the research was the time needed to obtain the data via the APIs and the time needed for WEKA to run the experiments Since our research required the observing of content from the beginning of its lifespan until its death, this required an API request for every snapshot of every piece of content. Because we needed to obey the rules regarding time between call set by each API, the time needed to observe large sets of data was very large. For example, to watch a single post on Reddit every 2 hours for 24 hours, would require 12 API calls. The Reddit API has a limit of no more than 30 requests per minutes. This means that in one hour, the maximum number of posts that can be observed is 1,800. We had set out to monitor millions of posts, but the time needed, approximately 555 days per million posts, to monitor millions of posts from each site, was not feasible.

The time requirement was compounded by the time WEKA needed to run experiments on very large data sets and slow running algorithms, with Decision Table being the slowest, on smaller data sets. The largest data set was the YouTube data set with 300,000 videos and over 100,000 attributes. The WEKA experiments ran for two months straight without completing. Solutions would be to reduce the number of attributes used in the experimentation phase either by selecting the most valuable attributes from prior experiments, or to apply multiple classifiers to the problem, using one to pick the attributes, then a fast running algorithm, such as SMO for training and testing.

## 6.3 Future Work

In future experiments, a multiple classifier approach may bring benefits in terms of runtime, using faster algorithms in order to reduce the feature space. The time needed for experimentation is currently impractical. Also in future experiments, customizing the model to each OSN may yield more accurate results. Our experiments began with Reddit, which is how our models for lifespan and popularity tier were created. A careful balance between practicality and accuracy is difficult to obtain, i.e., models should not be too broad (e.g. viral and not_viral), nor too specific because too many categories may lead to inaccurate classifications.

Our results from the YouTube data were poor, which suggests that future analysis will need to incorporate analysis of the video's content, not just the title. Along this same line, image analysis of Reddit, Flickr, and 4chan posts could add an important dimension of analysis beyond text alone. We also did not analyze any repeated content. Many images were observed to be repeated, and it would be fair to assume that images that are moderately liked, if not well-liked, are the ones being repeated. In a way, an OSN's population serves as the judge for the content they allow, reject, and want to see more of. Starting with content that is frequently reposted would serve as a great starting point for identifying the traits of what makes content popular. The opposite approach, looking at all the content that never gets repeated may help in identifying the traits of unpopular content, but there is so much unpopular content that there may be too much to digest.

During our research Reddit proved to be a very interesting cite for study. Reddit manages to create unique subcultures within subreddits while also fostering a vibrant site-wide culture. Using Reddit alone as a place for study would certainly produce interesting models. Comparing models across subreddits or even creating new subreddits and monitoring the lifespan of an entire subreddit are definitely ripe with intrigue.

# References


1. “100 Million Voices.” (2011) http://blog.Twitter.com/2011/09/one-hundred-million-voices.html (Accessed: October 2011).

2. “4chan/4chan Wiki – GitHub”. (2011). https://github.com/4chan/4chan-API (Accessed: April 2013).

3. “4chan - Advertise”. (2013). http://www.4chan.org/advertise (Accessed: September, 2013).

4. "About Reddit". (2013). http://www.Reddit.com/about/ (Accessed: September, 2013).

5. N. Ancona, R. Maglietta, and E. Stella. ‘Sparse representations and performances in support vector machines.’ In Proceedings of Machine Learning and Applications, Louisville, Kentucky, USA, December 2004, 129-136.

6. P. Adams, “Communication Mapping: Understanding Anyone's Social Network in 60 Minutes.” In Proceedings of the 2007 Conference on Designing for User eXperiences, Chicago, Illinois, USA, September 2007, 1-8.

7. P. Adams. “New Approach to Social Networks. The Beginning of googleMe.” (2010). http://www.slideshare.net/padday/the-real-life-social-network-v2 (Accessed: July, 2011).

8. J. Allan, J. G. Carbonell, G. Doddington, J. Yamron, and Y. Yang. “Topic Detection and Tracking Pilot Study.” (1998). http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.45.9763 (Accessed: August, 2010).

9. C. Allen. “Dunbar, Altruistic Punishment, and Meta-Moderation.” (2005). http://www.lifewithalacrity.com/2005/03/dunbar_altruist.html (Accessed: October, 2009).

10. P. Allen. “Google+ is Really Taking Off! Millions Joining Daily. 30% Increase in Users in Last 2 Days.” (2011). https://plus.google.com/117388252776312694644/posts/K9Qf1UVNyGy (Accessed: October, 2011).

11. “API – Reddit/Reddit Wiki – GitHub.” (2011). https://github.com/Reddit/Reddit/wiki/API (Accessed: September, 2012).

12. L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. “Group Formation in Large Social Networks: Membership, Growth, and Evolution.” In Proceedings of the 12th International Conference on Knowledge Discovery in Data Mining, New York, New York, USA, August 2006, 44-54.

13. A. Banerjee, H. Shan. “Latent Dirichlet Conditional Naive-Bayes Models.” In Proceedings of the Seventh IEEE International Conference on Data Mining, Piscataway, New Jersey, October 2007, 421-426.

14. M. Berkovich. “Perspective Probe: Many Parts Add Up to a Whole Perspective.” In Proceedings of the 27th International Conference extended abstracts on Human Factors in Computing Systems, Boston, Massachusetts, USA, April 2009, 2945-2954.

15. M. Bernstein, A. Monroy-Hernandez, D. Harry, P. Andre, K. Panovich, and G. Vargas. “4chan and /B/: An Analysis of Anonymity and Ephemerality in a Large Online Community.” In Proceedings of Fifth International AAAI Conference on Weblogs and Social Media, Menlo Park, California, USA, July 2011, 50-57.

16. L. Breiman "Random Forests." Machine Learning, 2001, Vol. 45, No. 1, 5-32.

17. “C++ Resources Network.” (2011). http://www.cplusplus.com (Accessed: November, 2011).

18. C. Canali, M. Colajanni, and R. Lancellotti. “Characteristics and Evolution of Content Popularity and User Relations in Social Networks.” In Proceedings of IEEE Symposium on Computers and Communications, Riccione, Italy, June 2010, 750-756.

19. M. Cataldi, L. Di Caro, and C. Schifanella. “Emerging Topic Detection on Twitter Based on Temporal and Social Terms Evaluation.” In Proceedings of the Tenth International Workshop on Multimedia Data Mining, New York, New York, USA, July 2010, 1-10.

20. M. Cha, A. Mislove, and K. Gummadi. “A Measurement-Driven Analysis of Information Propagation in the Flickr Social Network.” In Proceedings of the 18th International Conference on World Wide Web, Madrid, Spain, April 2009, 721-730.

21. “Charlie Schmidt’s Keyboard Cat! – THE ORIGINAL.” (2007). http://www.youtube.com/watch?v=J---aiyznGQ, (Accessed: January, 2010).

22. C. Chen. “The Creation and Meaning of Internet Memes in 4chan: Popular Internet Culture in the Age of Online Digital Reproduction.” Institutions Habitus Spring 2012, Yale University, 2012, 6-19.

23. C.C. Chen, Y. Chen, Y. Sun, and M.C. Chen. “Life Cycle Modeling of News Events Using Aging Theory.” In Proceedings of European Conference on Machine Learning, Cavtat-Dubrovnik, Croatia, September 2003, 47–59.

24. C.C. Chen, Y. Chen, and C. Chen. “An Aging Theory for Event Life-Cycle Modeling.” Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE, 2007, Vol. 37, No. 2, 237-348.

25. X. Cheng, J. Liu, and C. Dale. “Understanding the Characteristics of Internet Short Video Sharing: A YouTube-Based Measurement Study.” IEEE Transactions on Multimedia, 2013, Vol. 15, No.5, 1184-1194.

26. D. Chickering and D. Heckerman. “Fast Learning from Sparse Data.” In Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence, Stockholm, Sweden, July 1999, 109-115.

27. M. Crawford, J. Ham, Y. Chen, and J. Gosh. “Random Forests of Binary Hierarchical Classifiers for Analysis of Hyperspectral Data.” In Proceedings of IEEE Workshop Advances in Techniques for Analysis of Remotely Sensed Data, Greenbelt, Maryland, USA, 2003, 337-345.

28. R. Dubar. “Neocortex Size as a Constraint on Group Size in Primates,” Journal of Human Evolution, June 1992, Vol. 22, No.6, 469-493.

29. A. Elliot. “10 Fascination YouTube Facts that May Surprise You.” (2011). http://mashable.com/2011/02/19/youtube-facts/ (Accessed: February, 2011).

30. K. El-Arini, M. Xu, E.B. Fox, and C. Guestrin. “Representing Documents through Their Readers.” In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, Illinois, USA, August 2013, 14-22.

31. “Facebook Open Graph API.” (2011). http://developers.Facebook.com/docs/api (Accessed: September, 2011).

32. “Facebook Statistics.” (2011). https://www.Facebook.com/press/info.PHP?statistics (Accessed: November, 2011).

33. “Facebook takes #1 Social Networking Site in India.” (2010). http://www.comscore.com/Press_Events/Press_Releases/2010/8/Facebook_Captures_Top_Spot_among_Social_Networking_Sites_in_India?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed:+comscore+(comScore+Networks) (Accessed: August, 2010).

34. A. Fard and M. Ester, “Collaborative Mining in Multiple Social Networks Data for Criminal Group Discovery,” in Computational Science and Engineering, 2009, Vol. 4, 582-587.

35. “Flickr: The Flickr Development Guide – API”. (2009). http://www.flickr.com/services/developer/api/ (Accessed: May, 2013).

36. J. Gibbons and A. Agah. “Friend lens: Novel Web Content Sharing through Strategic Manipulation of Cached HTML.” International Journal of Web Based Communities, 2012, Vol. 8, No. 2, 242-265.

37. J. Gibbons and A. Agah. “Modeling Content Lifespan in Online Social Networks Using Data Mining.” International Journal of Web Based Communities, Under Review.

38. E. Gilbert. “Widespread Underprovision on Reddit.” Proceedings of the ACM Conference on Computer Supported Cooperative Work, San Antonio, Texas, 2013, 803-808.

39. “GNU Project.” (2011). http://www.gnu.org/licenses/ (Accessed: November, 2011).

40. “Google Trends.” (2006). http://www.google.com/trends (Accessed: September, 2011).

41. “Google+ API – Google+ Platform.” (2011). https://developers.google.com/+/api/ (Accessed: October, 2011).

42. J. Grzymala-Busse. “MLEM2—Discretization During Rule Induction.” In Proceedings of International Conference on Intelligent Information Processing and WEB Mining Systems, Zakopane, Poland, June 2003, 499-508.

43. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten. “WEKA Data Mining Software: An Update.” Special Interest Group on Knowledge Discovery and Data Mining Explorations, 2009, Vol. 11, No. 1, 10-18.

44. M. Harris. “tmhOAuth: PHP wrapper for Twitter API” (2011). https://github.com/themattharris/tmhOAuth (Accessed: October, 2011).

45. N. Henr, A. Bezerianos, J.D. Fekete. “Improving the Readability of Clustered Social Networks using Node Duplication.” In IEEE Transactions on Visualization and Computer Graphics, 2008, Vol. 14, No. 6, 1317-1324.

46. “Hypertext Preprocessor.” (1998). http://PHP.net/ (Accessed: September, 2011).

47. M. Iliofotou and M. Faloutsos, “Exploiting Dynamicity in Graph-based Traffic Analysis: Techniques and Applications.” In Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies, Rome, Italy, 2009, 241-256.

48. “Internet Population Statistics” (2011, March, 31). http://www.internetworldstats.com/stats.htm (Accessed: 2011, April, 13).

49. “Introducing JSON.” (2011). http://www.json.org/ (Accessed: November, 2011).

50. A. Jeffries. “The Man Behind Flickr on Making the Service 'Awesome Again.” (2013). http://www.theverge.com/2013/3/20/4121574/Flickr-chief-markus-spiering-talks-photos-and-marissa-mayer (Accessed: September, 2013).

51. T. Joachims. “Training linear SVMs in Linear Time.” In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, Philadelphia, Pennsylvania, USA, 2006, 217-226.

52. G.H. John and P. Langley. “Estimating Continuous Distributions in Bayesian Classifiers.” In Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence, San Mateo, California, USA, 1995, 338-345.

53. B. Joonhyun and K. Sangwook, “A Global Social Graph as a Hybrid Hypergraph,” In Proceedings of the International Conference on Networked Computing and Advanced Information Management, Seoul, South Korea, 2009, 1025-1031,.

54. A. Josey. “POSIX – Austin Joint Working Group.” (2011). http://standards.ieee.org/develop/wg/POSIX.html (Accessed: November, 2011).

55. R. Kohavi. “The Power of Decision Tables.” In Proceedings of the 8th European Conference on Machine Learning, Crete, Greece, April 1995, 174-189.

56. F. Krienen, P. Tu, and R. Buckner. “Clan Mentality: Evidence That the Medial Prefrontal Cortex Responds to Close Others.” Journal of Neuroscience, 2010, Vol. 30, No. 41, 13906-13915.

57. B. Krishnamurthy and C. Wills, “Characterizing Privacy in Online Social Networks.” In Proceedings of the First Workshop on Online Social Networks, Seattle, WA, USA, August 2008, 37-42.

58. V. Lehtinen, J. Nasanen, and R Sarvas. “A Little Silly and Empty-headed: Older Adults' Understandings of Social Networking Sites.” In Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology, Cambridge, United Kingdom, September 2009, 45-54.

59. M. Lehtonen and A. Doucet. “Phrase detection in the Wikipedia.” In Proceedings of the 6th Annual Initiative for the Evaluation of XML Retrieval, Dagstuhl Castle, Germany, 2007, 115-121.

60. J. Leskovec and E. Horvitz. “Planetary-scale views on a large instant-messaging network.” In Proceedings of the 17th International Conference on World Wide Web, Beijing, China, April 2008, 915-924.

61. W.J. Long and W.X.Zhang. “A Novel Measure of Compatibility and Methods of Missing Attribute Values Treatment in Decision Tables.” In Proceedings of International Conference of Machine Learning and Cybernetics, Shanghai, China, April 2004, Vol. 4, 2356-2360.

62. L. Ma, C.S. Lee, and DH-L. Goh. “Sharing in Social News Websites: Examining the Influence of News Attributes and News Sharers.” In Proceedings of IEEE Ninth International Conference of Information Technology New Generations, Las Vegas, Nevada, April 2012, 726-731.

63. B. Maddock. “Socially Awkward: A History of Google's Social Media Failures.” (2010). http://www.huffingtonpost.com/bryce-maddock/socially-awkward-a-histor_b_685533.html (Accessed: August, 2010).

64. E. Martin. “Saying Goodbye to an Old Friend and Revising the Default SubReddits” (2011). http://blog.Reddit.com/2011/10/saying-goodbye-to-old-friend-and.html (Accessed: July, 2013).

65. Y. Matsuo and H. Yamamoto. “Community Gravity: Measuring Bidirectional Effects by Trust and Rating on Online Social Networks.” In Proceedings of the 18th international Conference on World Wide Web, Madrid, Spain, April 2009, 751-760.

66. C. McCarthy. “MySpace Plugs Into Facebook.” (2010). http://news.cnet.com/8301-13577_3-20015098-36.html?part=rss&tag=feed&subj=TheSocial (Accessed: September, 2010).

67. C. McCarthy. “Digg’s Matt Horn Leaves for Start Up ‘Path’” (2010). http://news.cnet.com/8301-13577_3-20014852-36.html?part=rss&tag=feed&subj=TheSocial (Accessed: August, 2010).

68. “Media Wiki API.” (2011). http://www.mediawi ki.org/wiki/API (Accessed: August, 2011).

69. R. Miller. “Outside of a Small Circle of Friends.” (2010). http://www.socmedia101.com/2009/04/outside-of-a-small-circle-of-friends/ (Accessed: March, 2010).

70. R. Mills. “Researching Social News – Is Reddit.com a mouthpiece for the ‘Hive Mind’, or a Collective Intelligence approach to Information Overload?” (2011). http://eprints.lancs.ac.uk/61646/ (Accessed: June, 2012).

71. F. Moosmann, E. Nowak, and F. Jurie. “Randomized Clustering Forests for Image Classification.” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009, Vol. 30, No. 9, 1632-1646.

72. “MySQL :: The World’s Most Popular Open Source Database.” (2009). http://www.mysql.com (Accessed: September, 2011).

73. M. Naaman, J. Boase, and C.H. Lai. “Is it Really About Me?: Message Content in Social Awareness Streams.” In Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work, Savannah, Georgia, USA, February 2010, 06-10.

74. “OAuth Community Site.” (2007). http://oauth.net (Accessed: August, 2011).

75. “Open Social API Documentation.” (2007). http://code.google.com/apis/opensocial/ (Accessed: August, 2010).

76. J. O’Dell “History of Social Media.” (2011). http://mashable.com/2011/01/24/the-history-of-social-media-infographic (Accessed: January, 2011).

77. L. Page, S. Brin, R. Motwani, and T. Winograd. “The Pagerank Citation ranking: Bringing Order to the Web.” In Proceedings of the 7th International World Wide Web Conference, Brisbane, Australia, 1998, 161-172.

78. Z. Pawlak, “Rough Sets. Theoretical Aspects of Reasoning about Data.” Kluwer Academic Publishers, Dordrecht, Boston, London, 1991.

79. “PHP: Hypertext Preprocessor” (2009). http://php.net (Accessed: October 2009).

80. J. Platt. “Fast Training of Support Vector Machines Using Sequential Minimal Optimization”. In Advances in Kernel Methods—Support Vector Learning, B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, Massachusetts, MIT Press, 1998, 41-64.

81. C. Poole. “Christopher "moot" Poole: The Case for Anonymity Online.” (2010). http://www.youtube.com/watch?v=a_1UEAGCo30 (Accessed: June, 2010).

82. “Reddit: The Front Page of the Internet.” (2010). http://reddit.com (Accessed: June 2013).

83. J.R. Quinlan. “C4. 5: Programs for Machine Learning.” Morgan kaufmann, 1993, Vol. 1.

84. T. Rattenbury and M. Naaman. “Methods for extracting place semantics from Flickr tags.” In ACM Transactions on the Web, 2009, Vol. 3, No. 1, 1-30.

85. “Roost Offers Facebook Engagement Algorithm.” (2011). http://www.allFacebook.com/roost-offers-Facebook-engagement-algorithm-2011-08?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+allFacebook+%28Facebook+Blog%29 (Accessed: September, 2011).

86. T. Sakaki, M. Okazaki, and Y. Matsuo. “Earthquake Detection Using Twitter: Real-time Event Detection by Social Sensors.” In Proceeding of the 19th International Conference on World Wide Web, Raleigh, North Carolina, USA, April 2010, 851-860.

87. A. Salihefendic. (2010). “How Reddit Ranking Algorithms Work” http://amix.dk/blog/post/19588 (Accessed: June, 2012).

88. B. Schneier. (2010). “A Revised Taxonomy of Social Networking Data.” http://www.schneier.com/blog/archives/2010/08/a_taxonomy_of_s_1.html (Accessed: August, 2010).

89. M.J. Schwartz. “Anonymous Posts Westboro Church Members' Personal Information” (2012). http://www.informationweek.com/security/privacy/anonymous-posts-westboro-church-members/240144592 (Accessed: July, 2013).

90. Y. Shafranovich. (2005). “Common Format and MIME Type for Comma-Separated Values (CSV) Files.” http://tools.ietf.org/html/rfc4180 (Accessed: September, 2011).

91. “Shopping by Mobile Will Grow to $119 Billion in 2015.” (2010). http://www.abiresearch.com/press/1605-Shopping+by+Mobile+Will+Grow+to+%24119+Billion+in+2015 (Accessed: February, 2010).

92. T. Singletary. “How Do I Get Firehose Access?” (2011). https://dev.twitter.com/discussions/2752 (Accessed: March, 2012).

93. V. Singh and R. Jain. “Structural Analysis of the Emerging Event-Web.” In Proceedings of the 19th International Conference on World Wide Web, Raleigh, North Carolina, USA, April 2010, 1183-1184.

94. “Social Network Data Collection.” (2010). http://gnip.com/ (Accessed: September, 2011).

95. S. de Sousa. “Multiple Social Networks Analysis of FLOSS Projects Using Sargas.” In Proceedings of 42nd Hawaii International Conference on System Sciences, Hawaii, USA, 2009, 1-10.

96. A. Stewart. “Cross-tagging for Personalized Open Social Networking.” In Proceedings of the 20th ACM Conference on Hypertext and Hypermedia, Torino, Italy, June 2009, 271-278.

97. N. Tanner. “Stacking Up Facebook Games.” (2011). http://au.pc.ign.com/articles/114/1147014p1.html (Accessed: February, 2011).

98. “Trendistic – See Trends in Twitter.” (2007). http://trendistic.indextank.com/ (Accessed: February, 2011).

99. Y. Tzeng. “Event Duration Detection on Microblogging.” In Proceedings of IEEE/WIC/ACM International Conference Web Intelligence and Intelligent Agent Technology, Macau, China, December 2012, Vol. 1, 16-23.

100. P. Van Mieghem. “Human Psychology of Common Appraisal: The Reddit Score.” In IEEE Transactions on Multimedia, December 2011, Vol. 13, No. 6, 1404-1406.

101. M. Valafar, R. Rejaie, and W. Willinger. “Beyond Friendship Graphs: A Study of User Interactions in Flickr.” In Proceedings of the Second ACM Workshop on Online Social Networks, New York, New York, USA, 2009, 25-30.

102. J. Vosecky, D. Hong, and V.Y. Shen. “User Identification Across Multiple Social Networks.” In Proceedings of First International Conference on Networked Digital Technologies, Ostrava, Czech Republic, 2009, 360-365.

103. Q. Wang. “Naive Bayesian Classifier for Rapid Assignment of rRNA Sequences into the New Bacterial Taxonomy.” Applied and Environmental Microbiology, 2007, Vol. 73, No. 16, 5261-5267.

104. “Weka 3 – Data Mining and Open Source Machine Learning Software in Java.” (2009). http://www.cs.waikato.ac.nz/~ml/weka (Accessed: June, 2011).

105. W. Willinger, R. Rejaie, M. Torkjazi, M Valafar, and M. Maggioni. “Research on Online Social Networks: Time to Face the Real Challenges.” ACM SIGMETRICS Performance Evaluation Review, December 2009, Vol. 37, No. 3, 49-54.

106. C. Wilson, B. Boe, A. Sala, K. Puttaswamy, and B. Zhao. “User Interactions in Social Networks and Their Implications.” In Proceedings of the 4th ACM European Conference on Computer Systems, Nuremberg, Germany, 2009, 205-218.

107. B. Wu, F. Zhao, S. Yang, L Suo, and H. Tian. “Characterizing the Evolution of Collaboration Network.” In Proceedings of the 2nd ACM Workshop on Social Web Search and Mining, Hong Kong, China, November 2009, 33-40.

108. L. Xing and R. Ruguo. “The Evolutionary Analysis of Trust Mechanism in the Industrial Cluster Based on Social Network.” In Proceedings of the Information Management, Innovation Management and Industrial Engineering International Conference, Xi’an, China, 2009, Vol. 2, 196-199.

109. A. Xu and X. Zheng. “Dynamic Social Network Analysis Using Latent Space Model and an Integrated Clustering Algorithm.” In Proceedings of the Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, Chengdu, China, December 2009, 620-625.

110. B. Xu and L. Lu. “Information diffusion through online social network.” In Proceedings of the IEEE International Conference on Emergency Management and Management Sciences, Beijing, China, August 2010, 53-56.

111. Y. Yang. “Towards Real-Time Music Auto-Tagging Using Sparse Features.” In Proceedings of the IEEE International Conference on Multimedia and Expo, San Jose, California, USA, July 2013, 1-6.

112. “YouTube APIs and Tools.” (2007). http://code.google.com/apis/YouTube/overview.html (Accessed: June, 2010).

113. “YouTube Data API (v3) – YouTube API – Google Developers.” (2013). https://developers.google.com/YouTube/v3/ (Accessed: June, 2013).

114. D. Zarrella. (2011). “Infographic: 5 Questions and Answers about Facebook Marketing”. http://danzarrella.com/infographic-5-questions-and-answers-about-Facebook-marketing.html (Accessed January, 2011).

115. R. Zhou, S. Khemmarat, and L. Gao. “The Impact of YouTube Recommendation System on Video Views.” In Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement, Melbourne, Australia, November 2010, 404-410.

116. Y. Zhou, K. Fleishmann, and W.A. Wallace. “Automatic Text Analysis of Values in the Enron Email Dataset: Clustering a Social Network Using the Value Patterns of Actors.” In Proceedings of the 43rd Hawaii International Conference on System Sciences, Hawaii, USA, January 2009, pp. 1 – 10.

# Appendix A: Confusion Matrices

This Appendix contains the confusion matrices produced from the 10-fold cross-validation experiments ran on Reddit, 4chan, Flickr, and Youtube. Some tables have empty fields due to experiments not finishing.

## Reddit Confusion Matrices: Popularity Tier

Naïve Bayes

| a | b | c | d | e | f | <-- classified as | | |
|---|---|---|---|---|---|---|---|---|
| 12877 | 2729 | 108 | 109 | 120 | 155 | \| | a | BELOWAVG |
| 1591 | 839 | 20 | 12 | 16 | 14 | \| | b | AVG |
| 308 | 229 | 7 | 4 | 2 | 1 | \| | c | POPULAR |
| 86 | 15 | 0 | 0 | 0 | 0 | \| | d | SUPERPOPULAR |
| 8 | 3 | 0 | 0 | 0 | 0 | \| | e | VIRAL |
| 6 | 2 | 0 | 0 | 0 | 0 | \| | f | DOA |

Random Forest

| a | b | c | d | e | f | <-- classified as | | |
|---|---|---|---|---|---|---|---|---|
| 16036 | 54 | 8 | 0 | 0 | 0 | \| | a | BELOWAVG |
| 2433 | 51 | 8 | 0 | 0 | 0 | \| | b | AVG |
| 533 | 16 | 2 | 0 | 0 | 0 | \| | c | POPULAR |
| 100 | 0 | 1 | 0 | 0 | 0 | \| | d | SUPERPOPULAR |
| 11 | 0 | 0 | 0 | 0 | 0 | \| | e | VIRAL |
| 8 | 0 | 0 | 0 | 0 | 0 | \| | f | DOA |

Decision Table

| a | b | c | d | e | f | <-- classified as | | |
|---|---|---|---|---|---|---|---|---|
| 16066 | 32 | 0 | 0 | 0 | 0 | \| | a | BELOWAVG |
| 2376 | 113 | 3 | 0 | 0 | 0 | \| | b | AVG |
| 538 | 10 | 3 | 0 | 0 | 0 | \| | c | POPULAR |
| 100 | 1 | 0 | 0 | 0 | 0 | \| | d | SUPERPOPULAR |
| 11 | 0 | 0 | 0 | 0 | 0 | \| | e | VIRAL |
| 8 | 0 | 0 | 0 | 0 | 0 | \| | f | DOA |

SMO

| a | b | c | d | e | f | <-- classified as | | |
|---|---|---|---|---|---|---|---|---|
| 15640 | 434 | 19 | 5 | 0 | 0 | \| | a | BELOWAVG |
| 2130 | 338 | 24 | 0 | 0 | 0 | \| | b | AVG |
| 443 | 87 | 20 | 1 | 0 | 0 | \| | c | POPULAR |
| 93 | 5 | 3 | 0 | 0 | 0 | \| | d | SUPERPOPULAR |
| 11 | 0 | 0 | 0 | 0 | 0 | \| | e | VIRAL |
| 8 | 0 | 0 | 0 | 0 | 0 | \| | f | DOA |

## Reddit Confusion Matrices: Life Span (peak/death) Categories

Naïve Bayes

| a | b | c | d | | <-- classified as | | |
|---|---|---|---|---|---|---|---|
| 2803 | 2618 | 125 | 47 | \| | a | = | earlypeak_latedeath |
| 3738 | 8360 | 180 | 107 | \| | b | = | earlypeak_earlydeath |
| 744 | 479 | 44 | 8 | \| | c | = | latepeak_latedeath |
| 3 | 4 | 1 | 0 | \| | d | = | DOA |

Random Forest

| a | b | c | d | | <-- classified as | | |
|---|---|---|---|---|---|---|---|
| 482 | 5091 | 20 | 0 | \| | a | = | earlypeak_latedeath |
| 490 | 11875 | 20 | 0 | \| | b | = | earlypeak_earlydeath |
| 93 | 1171 | 11 | 0 | \| | c | = | latepeak_latedeath |
| 0 | 8 | 0 | 0 | \| | d | = | DOA |

Decision Table

| a | b | c | d | | <-- classified as | | |
|---|---|---|---|---|---|---|---|
| 147 | 5444 | 2 | 0 | \| | a | = | earlypeak_latedeath |
| 130 | 12248 | 7 | 0 | \| | b | = | earlypeak_earlydeath |
| 12 | 1263 | 0 | 0 | \| | c | = | latepeak_latedeath |
| 0 | 8 | 0 | 0 | \| | d | = | DOA |

SMO

| a | b | c | d | | <-- classified as | | |
|---|---|---|---|---|---|---|---|
| 1831 | 3665 | 97 | 0 | \| | a | = | earlypeak_latedeath |
| 2220 | 10037 | 128 | 0 | \| | b | = | earlypeak_earlydeath |
| 454 | 780 | 41 | 0 | \| | c | = | latepeak_latedeath |
| 3 | 5 | 0 | 0 | \| | d | = | DOA |

## 4Chan Confusion Matrices: Popularity Tier

Naïve Bayes

| a | b | c | d | e | | <-- classified as | | |
|---|---|---|---|---|---|---|---|---|
| 2 | 3 | 59 | 46 | 0 | \| | a | = | POPULAR |
| 8 | 29 | 127 | 149 | 0 | \| | b | = | AVG |
| 181 | 134 | 3037 | 3163 | 0 | \| | c | = | BELOWAVG |
| 425 | 318 | 6736 | 9334 | 0 | \| | d | = | DOA |
| 0 | 0 | 0 | 1 | 0 | \| | e | = | SUPERPOPULAR |

Random Forest

| a | b | c | d | e | | <-- classified as | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 9 | 101 | 0 | \| | a | = | POPULAR |
| 0 | 8 | 29 | 276 | 0 | \| | b | = | AVG |
| 1 | 8 | 581 | 5925 | 0 | \| | c | = | BELOWAVG |
| 3 | 9 | 1019 | 15782 | 0 | \| | d | = | DOA |
| 0 | 0 | 0 | 1 | 0 | \| | e | = | SUPERPOPULAR |

Decision Table

| a | b | c | d | e | | <-- classified as | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 110 | 0 | \| | a | = | POPULAR |
| 0 | 7 | 1 | 305 | 0 | \| | b | = | AVG |
| 2 | 0 | 110 | 6403 | 0 | \| | c | = | BELOWAVG |
| 1 | 1 | 28 | 16783 | 0 | \| | d | = | DOA |
| 0 | 0 | 1 | 0 | 0 | \| | e | = | SUPERPOPULAR |

SMO

| a | b | c | d | e | | <-- classified as | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 15 | 95 | 0 | \| | a | = | POPULAR |
| 0 | 16 | 51 | 246 | 0 | \| | b | = | AVG |
| 7 | 25 | 1145 | 5338 | 0 | \| | c | = | BELOWAVG |
| 10 | 39 | 2025 | 14739 | 0 | \| | d | = | DOA |
| 0 | 0 | 1 | 0 | 0 | \| | e | = | SUPERPOPULAR |

# 4chan Confusion Matrices: Life Span (peak/death) Categories

Naïve Bayes

| a | b | c | d | | <-- classified as | | |
|---|---|---|---|---|---|---|---|
| ### | 150 | 2979 | 233 | \| | a | = | earlypeak_earlydeath |
| 109 | 10 | 133 | 23 | \| | b | = | earlypeak_latedeath |
| ### | 410 | 9296 | 643 | \| | c | = | DOA |
| 265 | 14 | 274 | 45 | \| | d | = | latepeak_latedeath |

Random Forest

| a | b | c | d | | <-- classified as | | |
|---|---|---|---|---|---|---|---|
| 453 | 8 | 5594 | 11 | \| | a | = | earlypeak_earlydeath |
| 25 | 0 | 249 | 1 | \| | b | = | earlypeak_latedeath |
| 842 | 7 | 15941 | 23 | \| | c | = | DOA |
| 38 | 2 | 543 | 15 | \| | d | = | latepeak_latedeath |

Decision Table

| a | b | c | d | | <-- classified as | | |
|---|---|---|---|---|---|---|---|
| 63 | 0 | 5997 | 6 | \| | a | = | earlypeak_earlydeath |
| 5 | 0 | 269 | 1 | \| | b | = | earlypeak_latedeath |
| 151 | 0 | 16654 | 8 | \| | c | = | DOA |
| 11 | 0 | 582 | 5 | \| | d | = | latepeak_latedeath |

SMO

| a | b | c | d | | <-- classified as | | |
|---|---|---|---|---|---|---|---|
| 930 | 18 | 5088 | 30 | \| | a | = | earlypeak_earlydeath |
| 47 | 1 | 223 | 4 | \| | b | = | earlypeak_latedeath |
| 1866 | 21 | 14863 | 63 | \| | c | = | DOA |
| 101 | 3 | 476 | 18 | \| | d | = | latepeak_latedeath |

## Flickr Confusion Matrices: Popularity Tier

Naïve Bayes

| a | b | c | d | e | | | | <-- classified as |
|---|---|---|---|---|---|---|---|---|
| 2062 | 786 | 334 | 177 | 4 | \| | a | = | AVG |
| 378 | 13672 | 710 | 155 | 3 | \| | b | = | DOA |
| 551 | 2972 | 6554 | 315 | 9 | \| | c | = | BELOWAVG |
| 30 | 28 | 12 | 46 | 0 | \| | d | = | POPULAR |
| 0 | 0 | 0 | 0 | 2 | \| | e | = | SUPERPOPULAR |

Random Forest

| a | b | c | d | e | | | | <-- classified as |
|---|---|---|---|---|---|---|---|---|
| 694 | 1751 | 911 | 7 | 0 | \| | a | = | AVG |
| 38 | 13497 | 1383 | 0 | 0 | \| | b | = | DOA |
| 230 | 6575 | 3596 | 0 | 0 | \| | c | = | BELOWAVG |
| 18 | 55 | 31 | 12 | 0 | \| | d | = | POPULAR |
| 0 | 2 | 0 | 0 | 0 | \| | e | = | SUPERPOPULAR |

Decision Table

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

SMO

| a | b | c | d | e | | | | <-- classified as |
|---|---|---|---|---|---|---|---|---|
| 1273 | 817 | 1265 | 8 | 0 | \| | a | = | AVG |
| 141 | 12465 | 2312 | 0 | 0 | \| | b | = | DOA |
| 547 | 4465 | 5388 | 1 | 0 | \| | c | = | BELOWAVG |
| 44 | 24 | 25 | 23 | 0 | \| | d | = | POPULAR |
| 0 | 1 | 1 | 0 | 0 | \| | e | = | SUPERPOPULAR |

# Flickr Confusion Matrices: Life Span (peak/death) Categories

Naïve Bayes

| a | b | c | d | | <-- classified as | | |
|---|---|---|---|---|---|---|---|
| 3906 | 4762 | 337 | 1190 | \| | a | = | earlypeak_earlydeath |
| 1610 | 12414 | 216 | 678 | \| | b | = | DOA |
| 224 | 360 | 39 | 122 | \| | c | = | earlypeak_latedeath |
| 678 | 1482 | 84 | 698 | \| | d | = | latepeak_latedeath |

Random Forest

| a | b | c | d | | <-- classified as | | |
|---|---|---|---|---|---|---|---|
| 3970 | 5935 | 43 | 247 | \| | a | = | earlypeak_earlydeath |
| 1198 | 13586 | 15 | 119 | \| | b | = | DOA |
| 226 | 455 | 35 | 29 | \| | c | = | earlypeak_latedeath |
| 649 | 1806 | 23 | 464 | \| | d | = | latepeak_latedeath |

Decision Table

| | | | | | <-- classified as | | |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

SMO

| a | b | c | d | | | | |
|---|---|---|---|---|---|---|---|
| 5897 | 3724 | 96 | 478 | \| | a | = | earlypeak_earlydeath |
| 2371 | 12267 | 25 | 255 | \| | b | = | DOA |
| 359 | 282 | 46 | 58 | \| | c | = | earlypeak_latedeath |

## YouTube Confusion Matrices: Popularity Tier

Naïve Bayes

| a | b | c | d | e | f | <-- classified as | | | |
|---|---|---|---|---|---|---|---|---|---|
| 48 | 6 | 87 | 5 | 190 | 8 | \| | a | = | DOA |
| 165 | 638 | 2532 | 782 | 586 | 988 | \| | b | = | SUPERPOPULAR |
| 148 | 284 | 4649 | 461 | 1878 | 406 | \| | c | = | AVG |
| 221 | 706 | 4953 | 1352 | 1354 | 1062 | \| | d | = | POPULAR |
| 58 | 29 | 627 | 41 | 633 | 43 | \| | e | = | BELOWAVG |
| 122 | 510 | 1873 | 606 | 351 | 1597 | \| | f | = | VIRAL |

Random Forest

| a | b | c | d | e | f | <-- classified as | | | |
|---|---|---|---|---|---|---|---|---|---|
| 29 | 6 | 225 | 62 | 22 | 0 | \| | a | = | DOA |
| 1 | 1055 | 1411 | 2730 | 3 | 491 | \| | b | = | SUPERPOPULAR |
| 10 | 207 | 4620 | 2816 | 70 | 103 | \| | c | = | AVG |
| 0 | 673 | 3520 | 5164 | 12 | 279 | \| | d | = | POPULAR |
| 19 | 17 | 988 | 308 | 91 | 8 | \| | e | = | BELOWAVG |
| 0 | 652 | 979 | 1936 | 4 | 1488 | \| | f | = | VIRAL |

Decision Table

| | | | | | | <-- classified as | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | DOA |
| | | | | | | | | | SUPERPOPULAR |
| | | | | | | | | | AVG |
| | | | | | | | | | POPULAR |
| | | | | | | | | | BELOWAVG |
| | | | | | | | | | VIRAL |

SMO

| a | b | c | d | e | f | <-- classified as | | | |
|---|---|---|---|---|---|---|---|---|---|
| 56 | 6 | 204 | 21 | 54 | 3 | \| | a | = | DOA |
| 3 | 1753 | 1077 | 2096 | 20 | 742 | \| | b | = | SUPERPOPULAR |
| 18 | 256 | 5204 | 2075 | 168 | 105 | \| | c | = | AVG |
| 2 | 1054 | 3229 | 4929 | 30 | 404 | \| | d | = | POPULAR |
| 40 | 21 | 1012 | 150 | 193 | 15 | \| | e | = | BELOWAVG |
| 3 | 919 | 701 | 1105 | 10 | 2321 | \| | f | = | VIRAL |

# Appendix B: Student's T-Tests

This Appendix contains the Student's T-Test information for all the pairwise comparisons made. The tests were used to compare two model categories, namely, peak/death and popularity tier, each of the data mining algorithms, and each OSN with one another. The All Student's t-tests were performed with an Alpha value of 0.05.

| **Peak/Death & Popularity Tier t-Test: Two-Sample Assuming Unequal Variances** | | |
|---|---|---|
| | | |
| | ***Life Span Misclassification*** | ***Popularity Tier Misclassifications*** |
| **Mean** | 314.5217391 | 674.7207207 |
| **Variance** | 1104830.713 | 1582752.591 |
| **Observations** | 230 | 222 |
| **Hypothesized Mean Difference** | 0 | |
| **df** | 431 | |
| **t Stat** | -3.297352151 | |
| **P(T<=t) one-tail** | 0.000528421 | |
| **t Critical one-tail** | 1.648396712 | |
| **P(T<=t) two-tail** | 0.001056843 | |
| **t Critical two-tail** | 1.96548332 | |

| **Naïve Bayes & SMO t-Test: Two-Sample Assuming Unequal Variances** | | |
|---|---|---|
| | | |
| | ***Naïve Bayes*** | ***SMO*** |
| **Mean** | 558.2941176 | 452.5661765 |
| **Variance** | 1353266.609 | 999539.6697 |
| **Observations** | 136 | 136 |
| **Hypothesized Mean Difference** | 0 | |
| **df** | 264 | |
| **t Stat** | 0.803833579 | |
| **P(T<=t) one-tail** | 0.211108143 | |
| **t Critical one-tail** | 1.65064591 | |
| **P(T<=t) two-tail** | 0.422216287 | |
| **t Critical two-tail** | 1.968990497 | |

| Naïve Bayes & Decision Table t-Test: Two-Sample Assuming Unequal Variances | | |
|---|---|---|
| | | |
| | ***Naïve Bayes*** | ***Decision Table*** |
| **Mean** | 558.2941176 | 321.9864865 |
| **Variance** | 1353266.609 | 1466110.972 |
| **Observations** | 136 | 74 |
| **Hypothesized Mean Difference** | 0 | |
| **df** | 145 | |
| **t Stat** | 1.369748559 | |
| **P(T<=t) one-tail** | 0.086441422 | |
| **t Critical one-tail** | 1.655430251 | |
| **P(T<=t) two-tail** | 0.172882843 | |
| **t Critical two-tail** | 1.976459563 | |

| Naïve Bayes & Random Forest t-Test: Two-Sample Assuming Unequal Variances | | |
|---|---|---|
| | | |
| | ***Naïve Bayes*** | ***Random Forest*** |
| **Mean** | 558.2941176 | 471.2279412 |
| **Variance** | 1353266.609 | 1472439.807 |
| **Observations** | 136 | 136 |
| **Hypothesized Mean Difference** | 0 | |
| **df** | 270 | |
| **t Stat** | 0.604025687 | |
| **P(T<=t) one-tail** | 0.273166867 | |
| **t Critical one-tail** | 1.650516748 | |
| **P(T<=t) two-tail** | 0.546333735 | |
| **t Critical two-tail** | 1.968789022 | |

| SMO & Decision Table t-Test: Two-Sample Assuming Unequal Variances | | |
|---|---|---|
| | | |
| | ***SMO*** | ***Decision Table*** |
| **Mean** | 452.5661765 | 321.9864865 |
| **Variance** | 999539.6697 | 1466110.972 |
| **Observations** | 136 | 74 |
| **Hypothesized Mean Difference** | 0 | |
| **df** | 128 | |
| **t Stat** | 0.792311255 | |
| **P(T<=t) one-tail** | 0.214822597 | |
| **t Critical one-tail** | 1.656845226 | |
| **P(T<=t) two-tail** | 0.429645193 | |
| **t Critical two-tail** | 1.97867085 | |

| SMO & Random Forest t-Test: Two-Sample Assuming Unequal Variances | | |
|---|---|---|
| | | |
| | ***SMO*** | ***Random Forest*** |
| **Mean** | 452.5661765 | 471.2279412 |
| **Variance** | 999539.6697 | 1472439.807 |
| **Observations** | 136 | 136 |
| **Hypothesized Mean Difference** | 0 | |
| **df** | 260 | |
| **t Stat** | -0.138420283 | |
| **P(T<=t) one-tail** | 0.445007707 | |
| **t Critical one-tail** | 1.650735342 | |
| **P(T<=t) two-tail** | 0.890015414 | |
| **t Critical two-tail** | 1.969130003 | |

| Decision Table and Random Forest t-Test: Two-Sample Assuming Unequal Variances | | |
|---|---|---|
| | | |
| | ***Decision Table*** | ***Random Forest*** |
| **Mean** | 321.9864865 | 471.2279412 |
| **Variance** | 1466110.972 | 1472439.807 |
| **Observations** | 74 | 136 |
| **Hypothesized Mean Difference** | 0 | |
| **df** | 150 | |
| **t Stat** | -0.852612426 | |
| **P(T<=t) one-tail** | 0.197616795 | |
| **t Critical one-tail** | 1.6550755 | |
| **P(T<=t) two-tail** | 0.39523359 | |
| **t Critical two-tail** | 1.975905331 | |

| Reddit & 4chan t-Test: Two-Sample Assuming Unequal Variances | | |
|---|---|---|
| | | |
| | ***Reddit*** | ***4Chan*** |
| **Mean** | 263.1964 | 526.9141 |
| **Variance** | 690550.8 | 2205809 |
| **Observations** | 168 | 128 |
| **Hypothesized Mean Difference** | 0 | |
| **df** | 187 | |
| **t Stat** | -1.80513 | |
| **P(T<=t) one-tail** | 0.036332 | |
| **t Critical one-tail** | 1.653043 | |
| **P(T<=t) two-tail** | 0.072663 | |
| **t Critical two-tail** | 1.972731 | |

| Reddit & Flickr t-Test: Two-Sample Assuming Unequal Variances | | |
|---|---|---|
| | | |
| | ***Reddit*** | ***Flickr*** |
| **Mean** | 263.1964 | 620.3333 |
| **Variance** | 690550.8 | 1517038 |
| **Observations** | 168 | 96 |
| **Hypothesized Mean Difference** | 0 | |
| **df** | 145 | |
| **t Stat** | -2.53086 | |
| **P(T<=t) one-tail** | 0.006223 | |
| **t Critical one-tail** | 1.65543 | |
| **P(T<=t) two-tail** | 0.012445 | |
| **t Critical two-tail** | 1.97646 | |

| Reddit and YouTube t-Test: Two-Sample Assuming Unequal Variances | | |
|---|---|---|
| | | |
| | ***Reddit*** | ***YouTube*** |
| **Mean** | 263.1964 | 601.9667 |
| **Variance** | 690550.8 | 845421.1 |
| **Observations** | 168 | 90 |
| **Hypothesized Mean Difference** | 0 | |
| **df** | 167 | |
| **t Stat** | -2.91524 | |
| **P(T<=t) one-tail** | 0.002021 | |
| **t Critical one-tail** | 1.654029 | |
| **P(T<=t) two-tail** | 0.004042 | |
| **t Critical two-tail** | 1.974271 | |

| 4chan & Flickr t-Test: Two-Sample Assuming Unequal Variances | | |
|---|---|---|
| | | |
| | ***4Chan*** | ***Flickr*** |
| **Mean** | 526.9141 | 620.3333 |
| **Variance** | 2205809 | 1517038 |
| **Observations** | 128 | 96 |
| **Hypothesized Mean Difference** | 0 | |
| **df** | 220 | |
| **t Stat** | -0.51398 | |
| **P(T<=t) one-tail** | 0.303891 | |
| **t Critical one-tail** | 1.651809 | |
| **P(T<=t) two-tail** | 0.607781 | |
| **t Critical two-tail** | 1.970806 | |

| 4chan & YouTube t-Test: Two-Sample Assuming Unequal Variances | | |
|---|---|---|
| | | |
| | ***4Chan*** | ***YouTube*** |
| **Mean** | 526.9141 | 601.9667 |
| **Variance** | 2205809 | 845421.1 |
| **Observations** | 128 | 90 |
| **Hypothesized Mean Difference** | 0 | |
| **df** | 213 | |
| **t Stat** | -0.45995 | |
| **P(T<=t) one-tail** | 0.323011 | |
| **t Critical one-tail** | 1.652039 | |
| **P(T<=t) two-tail** | 0.646022 | |
| **t Critical two-tail** | 1.971164 | |

| Flickr & YouTube t-Test: Two-Sample Assuming Unequal Variances | | |
|---|---|---|
| | | |
| | ***Variable 1*** | ***YouTube*** |
| **Mean** | 620.3333 | 601.9667 |
| **Variance** | 1517038 | 845421.1 |
| **Observations** | 96 | 90 |
| **Hypothesized Mean Difference** | 0 | |
| **df** | 175 | |
| **t Stat** | 0.115708 | |
| **P(T<=t) one-tail** | 0.454008 | |
| **t Critical one-tail** | 1.653607 | |
| **P(T<=t) two-tail** | 0.908016 | |
| **t Critical two-tail** | 1.973612 | |