2017

Machine Learning for the Automated Identification of Cyberbullying and Cyberharassment

Daniel N. Ducharme
University of Rhode Island, daniel.ducharme@outlook.com

MACHINE LEARNING FOR THE AUTOMATED IDENTIFICATION OF CYBERBULLYING AND CYBERHARASSMENT

BY

DANIEL N. DUCHARME

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN

COMPUTER SCIENCE

UNIVERSITY OF RHODE ISLAND

2017

DOCTOR OF PHILOSOPHY DISSERTATION

OF

DANIEL N. DUCHARME

APPROVED:

Dissertation Committee:

Major Professor Lisa DiPippo

Lutz Hamel

Lubos Thoma

Nasser H. Zawia

DEAN OF THE GRADUATE SCHOOL

UNIVERSITY OF RHODE ISLAND

2017

ABSTRACT

Cyberbullying and cyberharassement are a growing issue that is straining the resources of human moderation teams. This is leading to an increase in suicide among the affected teens who are unable to get away from the harassment. By utilizing n-grams and support vector machines, this research was able to classify YouTube comments with an overall accuracy of 81.8%. This increased to 83.9% when utilizing retraining that added the misclassified comments to the training set. To accomplish this, a 350 comment balanced training set, with 7% of the highest entropy 3 length n-grams, and a polynomial kernel with the C error factor of 1, a degree of 2, and a Coef0 of 1 were used in the LibSVM implementation of the support vector machine algorithm. The 350 comments were also trimmed with a k-nearest neighbor algorithm where k was set to 4% of the training set size. With the algorithm designed to be heavily multi-threaded and capable of being run across multiple servers, the system was able to achieve that accuracy while classifying 3 comments per second, running on consumer grade hardware over Wi-Fi.

ACKNOWLEDGMENTS

First and foremost, I would like to thank Dr. Lisa DiPippo for helping to guide me through the years of research. I would also like to thank Dr. Lutz Hamel for teaching me about machine learning and for writing the excellent source material that sparked this research. I would like to thank Leandro Costa for providing the code from his own research to help with generating a better training set. I would like to thank my friends and family, who have put up with the long hours and aided me in whatever way they were able. I would like to thank the editor, Kristy Poisson, who made this read much more like English than the original draft. Finally, I would like to thank my loving wife Tracey, who has been there through all the years of college and supported me in this life long dream.

TABLE OF CONTENTS

Page

Page

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

Table **Page**

CHAPTER 1

Introduction

The purpose of this research is to develop a technique to automatically identify cyberbullying, cyberharassment and other prohibited speech. This research will implement an algorithm using existing machine learning techniques that will be able to identify cyberbullying in a single sample. With retraining, the algorithm must be able to adapt as laws about cyberbullying are changed. With this research, major social networking sites, such as Facebook, would be able to automatically identify harmful comments, relieving some of the stress on moderators.

1.1 Background

Cyberbullying is a growing phenomenon that is plaguing today's youth and is increasing at an alarming rate. As technology advances and becomes prevalent in more facets of our lives, the potential for bullies to reach into a teens' life increases, causing additional hardship leading to depression and, in some cases, even suicide. The Cyberbullying Research Center's research[1] showed that in 2013 about one in four teens had been the victim of cyberbullying and one in six teens was involved in the bullying. Their research also shows, in every study, that cyberbullying is on the rise. Extrapolating from the studies, they estimate that 2.2 million teens were cyberbullied nationwide in 2011 up from an estimated 1.9 million in 2009. This number is expected to increase as both teens and adults continue to have an increased online presence.

One of the major problems behind cyberbullying is the difficulty for parents to spot and identify the bullying. Research shows that only one in ten teens will ever report it to an adult[2]. This lack of reporting also overlaps with the fact that there are no physical signs that cyberbullying is occurring thus, without manually

monitoring all of the child's on-line interactions, it can easily go unnoticed. Then, even if a parent does recognize that some communication could be construed as cyberbullying, they do not know any relevant rules and regulations in order to stop it effectively.

Cyberbullying is a major issue because many teens have committed suicide due to the pressures of cyberbullying. This can be evidenced by the death of a 14 year old girl, Rebecca Sedwick, which resulted in the arrest of her 12 and 14 year old classmates[3]. In 2011 and the first four months of 2012, there were 18 cases of suicide that were linked to cyberbullying in the US, UK, Australia and Canada. This is up from 23 cases identified between the years of 2003 and 2010[4].

Cyberbullying takes place in numerous different locations and as such cannot be monitored with just one application. For example, one project that will be discussed below is an attempt to make an application that will identify and report cyberbullying taking place in Facebook, and while that is a great idea, it needs to be expanded to include other sites such as Twitter or text messaging. In 2012 and 2013 alone there were 9 suicides that were linked to the social network Ask.fm[5]. As the number of these social media outlets is always increasing, so to will the avenues of cyberbullying.

The current method of dealing with the problem is to use human moderators and administrators to remove offending comments and ban repeat offenders. However, on most sites, the number of comments far outweighs the ability of moderators to read and approve every comment. Thus, in order to combat this, moderators typically rely on users to flag or report offending comments. This means that users have already seen and been affected by the cyberbullying at which point it is too late to remove it. This algorithm will improve this situation by automatically flagging offending comments, at which point a human moderator could approve or

deny them without the intended victims having ever had to read it.

1.2 Goals

These are the goals I will be trying to achieve in order to complete this dissertation.

1. Develop a scientific definition of the legal version of cyberbullying and harassment

2. Develop an algorithm that is able to distinguish cyberbullying from other forms of communication

3. Ensure that the algorithm has a low number of false positives in order to not infringe on free speech

4. Ensure that the algorithm has a low number of false negatives to protect users from harmful conversations

5. The algorithm must be capable of handling multiple comments in parallel

6. The algorithm allows for retraining to handle new laws or regulations

7. The algorithm is capable of scaling up to thousands of comments per second on hardware available to large websites

List of References

[1] J. W. Patchin. Cyberbullying Research Center. "Cyberbullying research: 2013 update." [Online; accessed 2-January-2014]. Nov. 2013. [Online]. Available: http://cyberbullying.us/cyberbullying-research-2013-update/

[2] T. P. Pope. New York Times. "Parents often unaware of cyber bullying." [Online; accessed 25-January-2014]. Oct. 2008. [Online]. Available: http://well.blogs.nytimes.com/2008/10/03/parents-often-unaware-of-cyber-bullying/?_php=true&_type=blogs&_r=0

[3] D. Stanglin and W. M. Welch. USA Today. "Two girls arrested on bullying charges after suicide." [Online; accessed 25-January-2014]. Oct. 2013. [Online]. Available: http://www.usatoday.com/story/news/nation/2013/10/15/florida-bullying-arrest-lakeland-suicide/2986079/

[4] CBC News. "Cyberbullying-linked suicides rising, study says." [Online; accessed 25-January-2014]. Oct. 2012. [Online]. Available: http://www.cbc.ca/news/technology/cyberbullying-linked-suicides-rising-study-says-1.1213435

[5] R. Broderick. BuzzFeed. "9 teenage suicides in the last year were linked to cyber-bullying on social network ask.fm." [Online; accessed 25-January-2014]. Sept. 2013. [Online]. Available: http://www.buzzfeed.com/ryanhatesthis/a-ninth-teenager-since-last-september-has-committed-suicide

CHAPTER 2

Review of Literature

There has been little work done in the automatic identification of Cyberbullying although there has been some activity in using machine learning to handle the parsing of natural language.

2.1 Defining Cyberbullying

Before there can be any discussion about what can be done to stop cyberbullying, the first step needs to be to clearly define what cyberbullying is and what it is not. To that end there have been studies by sociologists around the world to help define both classical bullying and cyberbullying. One such study has been published by José Pinheiro Neves and Luzia de Oliveira Pinheiro from the University of Minho, Portugal, in the International Journal of Technoethics [1].

As cyberbullying is a type of bullying, it makes sense to first define what bullying is. According to a report by the National Center for Education Statistics (NCES) published in 2005, bullying includes three essential elements. "(1) the behavior must be aggressive and negative; (2) the behavior is carried out repeatedly; and (3) the behavior occurs in a relationship where there is an imbalance of power between the parties involved." [2] This definition has been upheld by numerous researchers over the years and although there are some differences the broad idea is the same.

For cyberbullying, the definition used for bullying does not fully work. For the first element there is no change between the two, however the other elements are not as useful in a digital medium where people are anonymous and thousands of people can jump on the bandwagon. One of the broadest definitions used by Neves and Pinheiro is "the use of communication technologies and information to

denigrate, humiliate and / or defame a person or a group of people." This broad definition shows that any use of modern communication, including websites such as Facebook or even Internet of Things devices that are able to transmit messages, can be used for cyberbullying, and it also removes the requirement of repeated behavior.

2.2 Relevant Laws

Now that we have a sociological definition of cyberbullying, a legal method needs to be completed so that rules can be devised that will be applied to the comments for classification. In order to inform the classification in a method that infringes the least amount on the first amendment, the laws at both the state level[3] and federal level[4] were utilized to create a definition that mimics what is considered a federal or state crime.

2.2.1 U.S. Code

The U.S. Federal Code is the laws that govern all American citizens across the country regardless of state. Within the US Code there are two federal titles that concern cyberbullying, Title 18 Crimes and Criminal Procedure[5] and Title 47 Telecommunications[6], and several subsections each of which will be explained.

Title 18 Crimes and Criminal Procedure

Section 1470 Transfer of obscene material to minors This section of the U.S. Code deals with protecting minors from obscene material. The important part is that "Whoever, using... any facility or means of interstate... commerce, knowingly transfers obscene matter to another individual who has not attained the age of 16 years, knowing that such other individual has not attained the age of 16 years, or attempts to do so..."[7] This means that if you use a commercial communications method (such as YouTube) to transfer obscene material to someone you

know to be under the age of 16 then you are in violation of federal law.

Section 1514 Civil action to restrain harassment of a victim or witness The only important piece of this section is the definition of harassment[8] which is an act or course of conduct directed at a person that causes substantial emotional distress and serves no legitimate purpose.

Section 2261A Stalking This section is an important one in this research because many of the pieces of cyberbullying can be found under the stalking laws.[9] Some relevant portions are:

> Whoever-
>
> 1. ...is present within the... territorial jurisdiction of the United States... with the intent to... harass, intimidate... and in the course of, or as a result of, such ... presence engages in conduct that -
>   (a) causes, attempts to cause, or would be reasonably expected to cause substantial emotional distress to a person...
> 2. with the intent to... harass, intimidate... uses... any interactive computer service or electronic communication service or electronic communication system or interstate commerce...
>   (a) causes, attempts to cause, or would be reasonably expected to cause substantial emotional distress to a person

Section 2266 Definitions The important definition is the course of conduct which appears in several of the other laws which requires a pattern of 2 or more acts, evidencing a continuity of purpose.[10]

Title 47 Telecommunications

Title 47 is the laws related to the use of telecommunication equipment such as computers distributing comments over the internet. Of all of the sections of the title only one relates in any way to cyberbullying.

Section 223 Obscene or harassing telephone calls in the District of Columbia or in interstate or foreign communications This section, though badly titled, lays out that whoever in interstate communications knowingly makes any comment with intent to abuse, threaten, or harass another person is in violation of federal law.[11]

2.2.2 Rhode Island General Law

Outside of federal law, each individual state has their own laws that govern citizens present or doing business within that state. While this does make it more difficult to come up with a one-size fits all definition that meets all national laws, the laws from the State of Rhode Island will be used as that is the jurisdiction in which this research was conducted.

Title 11 Criminal Offenses

Within the Rhode Island general law there is only one relevant title which is Title 11 on Criminal Offenses[12].

Section 11-42-2 Extortion and Blackmail Most of this section is outside of the scope of cyberbullying, however, there is a small section that does come up. This law states that anyone who maliciously threatens any injury to the reputation of another is in violation of state law.[13]

Section 11-52-4.2 Cyberstalking and Cyberharassment This section just reinforces that the laws that govern physical conduct such as the laws against stalking, are also present to any communication transmitted over an electronic device.[14]

Section 11-59-1 Definitions The section definitions just confirm the definition of both harasses and course of conduct as found in the federal law.[15] The state

definition of "harass" does include additional things such as the intent to seriously alarm, annoy, or bother the person.

Section 11-59-2 Stalking In the Rhode Island general law, the law against stalking is very straightforward, it is simply any person who harasses another person.[16]

2.3 BullyBlocker

A similar project being worked on, is the BullyBlocker from Arizona State University[17]. Its primary purpose is "to exploit social media data and, based off of a model built on previous research findings in areas of traditional and cyberbullying in adolescents, to then identify an instance of cyberbullying and notify the parents." To do this they have designed a calculation they are calling the Bullying Rank and, using calculated warning signs and vulnerability, they are able to calculate a risk ranking that will place the child in either a low, moderate, or severe risk category. Using these categories parents will be notified via e-mail and will be able to provide feedback in order to improve the identification.

While there has been future work mentioned to integrate machine learning, the problem exists that all of the research is focused solely upon Facebook and requires multiple warning signs in order to identify if a message was bullying or not. The proposed solution should be able to use machine learning to identify bullying in a single message and flag it as such.

2.4 Hate Speech Detection

A subset of the problem has already been solved by Columbia University where they utilized support vector machine learning in order to classify hate speech.[18] They define hate speech as "any communication that disparages a person or a group on the basis of some characteristic such as race, color, ethnicity, gender,

sexual orientation, nationality, religion, or other characteristic." Thus, while not all hate speech is cyberbullying (some hate speech has made it into the common vernacular and at this point is so common as to not be considered bullying), there is an overlap between the two.

When classifying anti-semitic speech they were able to achieve an accuracy of 94% and a precision of 68%. They used a sample set of 1000 paragraphs and determined if they contained hate speech by having 3 different annotators classify if it was or was not anti-semitic. To process this data, they used a template-based strategy which applied various positive and negative templates to the paragraph and kept count of how many occurrences were found, which they called the log-odds. Overall they managed to create an algorithm that equaled the performance of their annotators.

There are several differences between this research project and the research being done by this dissertation. First, the research was only classifying anti-semitic speech which is a small subset of the harassment that occurs. Thus, while they will achieve greater accuracy within the context of a certain type of harassment it does not necessarily generalize to other forms of harassment. Second, they utilized works by the paragraph where many of the cyberbullying comments online are small statements as sites such as Twitter[19] restrict you to 140 characters.

2.5 League of Legends Player Reform System

League of Legends (LoL) is a multiplayer online battle arena (MOBA) released by Riot Games, Inc. in October of 2009 and is one of the most popular MOBAs on the market today.[20] While the majority of players are friendly and just there to enjoy the game, there are around 3% of games where some form of homophobic, racist, or sexist language is used by up to 5% of the player base.[21] In an effort to combat this, Riot games created an automated player reform system. This new

system would allow them to apply bans, such as two-week or permanent bans, due to homophobia, racism, sexism, death threats or abuse. However, they still rely on the offended player to flag the comments, at which point the automated system determines if the flagging is correct and then applies the appropriate ban.[22] After several months of using the system and utilizing several million games worth of data, they expanded the system to handle more complex behaviors such as determining if your character was intentionally feeding the enemy (allowing the enemy to kill your character intentionally thus making them more powerful).[23]

While this system does appear to be robust and was created with a large amount of data and training, the major downside is that this is a commercial enterprise that has designed this system for use solely within their product. None of the research that has gone into this project has been published or peer-reviewed and thus the system acts as a closed box. This prevents others from taking this research and utilizing it to work in their own situation. The lack of peer-review also means that the black box could be malfunctioning and we will never know as they are unlikely to self report the statistics of false positive bans.

2.6 Twitch AutoMod

Starting in December of 2016, Twitch, which is a online video game streaming service owned by Amazon, rolled out a new tool called AutoMod. [24] This is a machine learning based automatic moderation software designed to hold back messages for moderator approval. It looks for and filters based on identity language, sexually explicit language, aggressive language, and profanity. Moderators can also set what level of filtering they wish to use which will ignore some of the filtered types depending on which of the four tiers is chosen. [25]

Unlike the research done in this dissertation, AutoMod seems to utilize a dictionary that must be kept up by the developers at Twitch. Beyond this, none

of the research Amazon put into this has been published and because of that, as of the writing of this dissertation it is unknown what sort of accuracy is possible with this product as well as how much machine learning plays into the tool versus simply using the dictionary and pattern matching to flag comments with certain works or combinations of words.

This research goes beyond the Twitch AutoMod tool in better matching the actions of moderators. Instead of utilizing training based on the moderators, the developers have arbitrarily designed four security levels that the moderators can choose from. What each of these levels has been trained on is unclear and must have been selected by the developers. The goal of this research is instead to utilize the existing moderators and to attempt to simply match their moderation patterns without regard to "types" of speech such as race, religion, gender, orientation or disability as the first tier of the AutoMod handles.

2.7 Machine Learning

Machine learning dates back to the 1950's[26] and a variety of algorithms now exist to do everything from language translating[27] to financial trading[28]. Of the different possible machine learning techniques, the two best possibilities for classifying cyberbullying are neural networks and support vector machines.

Perceptrons and later neural networks are based on the use of neurons; one in the case of a perceptrons and groups of them in a neural network. Each of these neurons receives n input signals along with n associated weights telling the neuron how to evaluate those inputs. After processing those inputs, it passes through a transfer function that will return either a +1, if the solution was positive, or a -1, if it was negative. The training of these perceptrons involves updating the associated weights until the n inputs on each of the training data sets correctly resolves to either a positive or negative result. There is a problem, however; while the

perceptrons and neural networks will find a solution, that solution is not guaranteed to be the optimum decision surface. The algorithm simply stops once it tests a solution that is found to be correct.[29] Another problem is that the created model is extremely complex, having multiple inputs and outputs and various weights on all of them. This makes it difficult to understand how it is arriving at an answer and to influence that process to a better answer.[30]

Decision trees are much easier for humans to understand as they are simple statements that you can easily follow to a conclusion. While they are simple to follow, that does not mean that they cannot become quite complex. At each node on the tree there is a test, and depending on the outcome of the data on that test, you go down to a certain leaf that may contain the solution or another test. These algorithms tend to be fast learning and have good accuracy and as such are used in things such as medical diagnosis. The downside to decision trees is that they have some limitations such as the inability to express all first order logic as well as the fact that duplication of tests can occur on the tree leading to much larger trees then necessary.[31]

Support vector machines are a dual maximum margin classifier which means that the algorithm ensures that the decision surface that it generates is equidistant from both sets of data, which is assumed to be the optimal placement. In order to properly classify as many different types of data sets as possible, support vector machines also contain a kernel which can be changed to several different formulas to better fit. Another main advantage is that, because the support vector machines generate an optimum solution, they will always return a unique solution for the given data unlike neural networks which will just give a solution though a better one may exist.[29] Similar to the neural networks, however, the models from a support vector machine are not recognizable by a human as a solution to the

problem.[30]

In this research, the support vector machines will be utilized as the machine learning algorithm as was used in the Columbia University research. Unlike perceptrons, the support vector machine is a maximum margin classifier, so while a perceptron will arrive at an answer, it is not guaranteed to be the best answer as once a solution is found the algorithm stops. Furthermore although the initial training of a support vector machine can take some time, the speed of classifying subsequent comments onto the decision surface is considerably faster then decision trees as there is only one comparison.

2.8 Natural Language Processing

Computational linguistics is the field of computer science that deals with the processing of language and has been an active field of research since the 1950s[32]. One of the first researchers who looked into the field was Alan Turing who created the Turing test to identify the point at which a computer was considered to be intelligent[33]. Since its inception, it has been used for a variety of purposes from the translation from one language to another in software such as Google Translate[34], to the processing of spoken language into text with software such as Dragon[35].

This research is built utilizing the research done throughout the field of computational linguistics in order to process the meaning from small statements of natural language into a form usable by the machine learning algorithm previously mentioned. The most useful model that will be used in this research is the n-gram models in which the frequency of words is used in order to classify types of speech[32].

2.8.1 N-Grams

N-grams are quite simply a word, called a token, or a group of tokens, that can be used to predict a statistical language model[32]. N-grams are used in many different types of language processing, from speech recognition to ensure that any words not clear are guessed correctly, to machine translation to help select the most accurate translation.

They can be done for multiple lengths which is typically denoted by replacing the N with the length number. Take the following example phrase:

The quick brown fox jumps over the lazy dog.

This phrase contains 9 words, 1 capital letter and one punctuation character. For the purposes of N-grams capitalization is ignored and punctuation and spaces are removed as well. So for the 1-grams there is *the*, *quick*, *brown*, fox, *jumps*, *over*, *the*, *lazy*, *dog*. Now if the 2-grams are calculated they would contain *the*|*quick*, *quick*|*brown*, *brown*|fox, fox|*jumps*, *jumps*|*over*, *over*|*the*, *the*|*lazy*, *lazy*|*dog*. This would continue for the 3-grams all the way up through 9-grams at which point there would be no further difference with this phrase.

Now, once the n-grams are separated, they are used to calculate the frequency of occurrence since machine learning algorithms work on numbers. This is done differently depending on the language processing in question but in general takes the form of:

$$Probability = \frac{CountwithNgram}{TotalCount} \tag{1}$$

So again looking at the phrase we see that in 1-grams the token "the" has a 22.2% occurrence while all the other words have a 11.1%.

2.9 Information Theory

Information theory is the study of how information is encoded into bits either for storage or transmission. It began in 1948 with a paper called "A Mathematical Theory of Communication" that was published in the Bell Systems Technical Journal by Claude Shannon. In this paper, Shannon identified the mathematical limit to how fast information could be transmitted without error. Combined with that, he was able to describe how all information could be encoded in bits that could then be compressed and transmitted, which is considered to be the beginning of the digital age[36]. The most important piece of information theory that is being used in this dissertation is the calculation of entropy and information gain.

2.9.1 Entropy and Information Gain

In the field of information theory, entropy (H) is the measure of how much information is being held towards a given probability, in a symbol such as a bit or in our case an n-gram[37]. Using this information, the n-grams can be pruned down to only those that provide solid information instead of utilizing all of the n-grams, many of which will just increase the complexity of the machine learning without aiding in separating the classes. This will allow us to eliminate low information n-grams such as the token $|a|$ which would not be an indicator of bullying, while prioritizing the features that are the best indicators of the classes.

In order to calculate the entropy, first we calculate the probability of a class with or without a certain feature. So for each n-gram we needed to calculate the number of positive comments with the n-gram, the number of negative comments with the n-gram, the total number of comments with the n-gram, and then the same three statistics but for comments without the n-gram. So first using the probability with a feature, for each class i you would calculate:

$$p_i = i_with/total_with \tag{2}$$

With both p_+ and p_- calculated for the comments with a feature, next the entropy of a comment having a specific feature can be calculated with:

$$H(P) = \sum_i p_i * LOG_2(\frac{1}{p_i}) \tag{3}$$

Fixing up the fraction this simplifies to:

$$H(P) = -\sum_i p_i * LOG_2(p_i) \tag{4}$$

Since we are only using two classes, that finally becomes:

$$H(P) = -p_+ * LOG_2(p_+) - p_- * LOG_2(p_-) \tag{5}$$

This is all repeated by calculating p_+ and p_- on the comments the do not have the feature. Those two H(P) values are then combined into a weighted averaged[38]. This is done using the total number of comments (T), the total number with the feature (TW), and the total number without the feature (TWO):

$$AvgEntropy = \frac{TW}{T} * H(P_{WithAttrb}) + \frac{TWO}{T} * H(P_{WithoutAttrb}) \tag{6}$$

The last piece needed before the information gain can be calculated is to get the amount of information stored in the Parent before splitting on this feature. This is done again using the total number of comments (T), the total number with the feature (TW), and the total number without the feature (TWO):

$$H(Parent) = -\frac{TW}{T} * LOG_2(\frac{TW}{T}) - \frac{TWO}{T} * LOG_2(\frac{TWO}{T}) \tag{7}$$

And finally we can than calculate the information gain from the attribute:

$$InformationGain = H(Parent) - AvgEntropy \tag{8}$$

When this is positive, it indicates that the n-gram in question is a good candidate to separate the classes. N-grams that are negative contain a loss of information. For this research, the n-grams are sorted by their Information Gain in descending order, and we then take a certain percent (being decided in testing) of those top most values.

List of References

[1] L. d. O. P. José Pinheiro Neves, "Cyberbullying: A sociological approach," *International Journal of Technoethics*, vol. 1, pp. 24–34, 2010, [Online; accessed 2-January-2016]. [Online]. Available: http://www.scribd.com/doc/134607212/Cyberbullying-a-Sociological-Approach#

[2] K. C. Jill F. DeVoe, Sarah Kaffenberger, "Student reports of bullying. results from the 2001 school crime supplement to the national crime victimization survey." National Center for Education Statistics, Tech. Rep., 2005, [Online; accessed 7-February-2016]. [Online]. Available: http://nces.ed.gov/pubs2005/2005310.pdf

[3] State of Rhode Island. "State of rhode island general laws." [Online; accessed 20-October-2015]. Dec. 2014. [Online]. Available: http://webserver.rilin.state.ri.us/Statutes/

[4] Legal Information Institute. "U.s. code." [Online; accessed 20-October-2015]. [Online]. Available: https://www.law.cornell.edu/uscode/text

[5] Legal Information Institute. "U.s. code: Title 18 - crimes and criminal procedure." [Online; accessed 20-October-2015]. Oct. 1970. [Online]. Available: https://www.law.cornell.edu/uscode/text/18

[6] Legal Information Institute. "U.s. code: Title 47 - telecommunications." [Online; accessed 20-October-2015]. Mar. 2013. [Online]. Available: https://www.law.cornell.edu/uscode/text/47

[7] Legal Information Institute. "18 u.s. code § 1470 - transfer of obscene material to minors." [Online; accessed 20-October-2015]. Oct. 1998. [Online]. Available: https://www.law.cornell.edu/uscode/text/18/1470

[8] Legal Information Institute. "18 u.s. code § 1514 - civil action to restrain harassment of a victim or witness." [Online; accessed 20-October-2015]. Dec. 2012. [Online]. Available: https://www.law.cornell.edu/uscode/text/18/1514

[9] Legal Information Institute. "18 u.s. code § 2261a - stalking." [Online; accessed 20-October-2015]. Mar. 2013. [Online]. Available: https://www.law.cornell.edu/uscode/text/18/2261A

[10] Legal Information Institute. "18 u.s. code § 2266 - definitions." [Online; accessed 20-October-2015]. Aug. 2006. [Online]. Available: https://www.law.cornell.edu/uscode/text/18/2266

[11] Legal Information Institute. "47 u.s. code § 223 - obscene or harassing telephone calls in the district of columbia or in interstate or foreign communications." [Online; accessed 20-October-2015]. Mar. 2013. [Online]. Available: https://www.law.cornell.edu/uscode/text/47/223

[12] State of Rhode Island. "Title 11 criminal offenses." [Online; accessed 20-October-2015]. 1992. [Online]. Available: http://webserver.rilin.state.ri.us/Statutes/TITLE11/INDEX.HTM

[13] State of Rhode Island. "Title 11 criminal offenses, chapter 11-42 threats and extortion, section 11-42-2 extortion and blackmail." [Online; accessed 20-October-2015]. 1992. [Online]. Available: http://webserver.rilin.state.ri.us/Statutes/TITLE11/11-42/11-42-2.HTM

[14] State of Rhode Island. "Title 11 criminal offenses, chapter 11-52 computer crime, section 11-52-4.2 cyberstalking and cyberharassment prohibited." [Online; accessed 20-October-2015]. 1992. [Online]. Available: http://webserver.rilin.state.ri.us/Statutes/title11/11-52/11-52-4.2.htm

[15] State of Rhode Island. "Title 11 criminal offenses, chapter 11-59 stalking, section 11-59-1 definitions." [Online; accessed 20-October-2015]. 2002. [Online]. Available: http://webserver.rilin.state.ri.us/Statutes/TITLE11/11-59/11-59-1.HTM

[16] State of Rhode Island. "Title 11 criminal offenses, chapter 11-59 stalking, section 11-59-2 stalking prohibited." [Online; accessed 20-October-2015]. 2002. [Online]. Available: http://webserver.rilin.state.ri.us/Statutes/TITLE11/11-59/11-59-2.HTM

[17] Y. N. Silva. Arizona State University. "Bullyblocker: Towards the identification of cyberbullying in facebook." [Online; accessed 2-January-2014]. [Online]. Available: http://www.public.asu.edu/~ynsilva/BullyBlocker/index.html

[18] W. Warner and J. Hirschberg, "Detecting hate speech on the world wide web," pdf, Columbia University, June 2012, [Online; accessed 25-May-2014]. [Online]. Available: http://aclweb.org/anthology//W/W12/W12-2103.pdf

[19] Twitter. "Twitter." [Online; accessed 15-September-2015]. Jan. 2015. [Online]. Available: https://twitter.com/

[20] Wikipedia, The Free Encyclopedia. "League of legends." [Online; accessed 20-October-2015]. Oct. 2015. [Online]. Available: https://en.wikipedia.org/wiki/League_of_Legends

[21] Riot Games, Inc. "Exploring player behavior design values." [Online; accessed 20-October-2015]. Nov. 2014. [Online]. Available: http://na.leagueoflegends.com/en/news/game-updates/player-behavior/exploring-player-behavior-design-values

[22] Riot Games, Inc. "New player reform system heads into testing." [Online; accessed 20-October-2015]. May 2015. [Online]. Available: http://na.leagueoflegends.com/en/news/game-updates/player-behavior/new-player-reform-system-heads-testing

[23] Riot Games, Inc. "Instant feedback powers up." [Online; accessed 20-October-2015]. Sept. 2015. [Online]. Available: http://na.leagueoflegends.com/en/news/game-updates/player-behavior/instant-feedback-powers

[24] Twitch Interactive. "How to use automod." [Online; accessed 26-December-2016]. Dec. 2016. [Online]. Available: https://help.twitch.tv/customer/portal/articles/2662186-how-to-use-automod

[25] M. Kamen. Wired. "Twitch now blocks trolls and hate-speech in real-time." [Online; accessed 3-February-2017]. Dec. 2016. [Online]. Available: http://www.wired.co.uk/article/twitch-introduces-anti-troll-automod-for-game-streams

[26] J. Hu. "History of machine learning." [Online; accessed 6-April-2014]. Apr. 2013. [Online]. Available: http://www.aboutdm.com/2013/04/history-of-machine-learning.html

[27] T. Brants, A. C. Popat, P. Xu, F. J. Och, and J. Dean, "Large language models in machine translation," pdf, Google, 2007. [Online]. Available: http://research.google.com/pubs/MachineTranslation.html

[28] S. Patterson. CNBC. "Man vs. machine: Seven major players in high-frequency trading." [Online; accessed 6-April-2014]. Sept. 2010. [Online]. Available: http://www.cnbc.com/id/39038892

[29] L. Hamel, *Knowledge Discovery with Support Vector Machines*. 111 River St, Hoboken, New Jersey 07030: John Wiley & Sons Inc., 2009.

[30] M. Dredze, "Machine learning finding patterns in the world," pdf, Johns Hopkins University, 2009. [Online]. Available: http://old-site.clsp.jhu.edu/workshops/ws09/documents/machine-learning-overview.pdf

[31] Y. Visell, "Lecture 12: Decision trees," pdf, McGill University, 2006. [Online]. Available: www.cim.mcgill.ca/~yon/ai/lectures/lec12.pdf

[32] D. Jurafsky and J. H. Martin, *Speech and Language Processing*. Upper Saddle River, New Jersey 07458: Pearson Education, Inc., 2009.

[33] A. Hodges. The Alan Turing Internet Scrapbook. "The turing test, 1950." [Online; accessed 20-October-2015]. 1997. [Online]. Available: http://www.turing.org.uk/scrapbook/test.html

[34] Google. "Google translate." [Online; accessed 20-October-2015]. 2015. [Online]. Available: https://translate.google.com/

[35] Nuance. "Dragon speech recognition software." [Online; accessed 20-October-2015]. 2015. [Online]. Available: http://www.nuance.com/dragon/index.htm

[36] K. T. Y. Aftab, Cheung. "Information theory: Information theory and the digital age." [Online; accessed 12-August-2016]. 2001. [Online]. Available: http://web.mit.edu/6.933/www/Fall2001/Shannon2.pdf

[37] T. Carter. "An introduction to information theory and entropy." [Online; accessed 12-August-2016]. 2011. [Online]. Available: http://astarte.csustan.edu/~tom/SFI-CSSS/info-theory/info-lec.pdf

[38] L. Shapiro. "Information gain." [Online; accessed 25-January-2016]. 2010. [Online]. Available: https://courses.cs.washington.edu/courses/cse455/10au/notes/InfoGain.pdf

CHAPTER 3

Methodology

There are several distinct steps to the methodology employed in this research. First, comments need to be gathered for use in the research. Next, the definition of cyberbullying needs to be clearly laid out in order to classify the comments. After that, several different programs need to be utilized in order to process the comments and then to train and test the support vector machine.

3.1 Gathering Data

The first step in the research was to gather both bullying and non-bullying comments to both train and test the machine learning algorithm in its ability to detect cyberbullying. In order to collect enough comments, a web crawler was utilized to harvest comments off of Twitter and YouTube. This web crawler was designed to grab all of the comments from the videos found in the YouTube playlist Popular Right Now by #PopularOnYouTube[1] where the top 200 videos at that time are displayed. The web crawler was designed in such a way that no user information was recorded and any username in the replies were stripped out. These comments were grabbed while the video was still popular which helps to ensure that the comments that were present at the time the site was crawled did not have time to be fully moderated, removing the offending comments that are needed for the research.

For the web crawler comment process, YouTube was chosen for several reasons. First the site has a variety of comments, users, and viewpoints resulting in arguments that can get heated. Next, it allows users to post with usernames that are completely removed from their non-internet alias. This allows users to have no consequences for anything they might say, outside of having their comments

moderated or their account shutdown. Another key factor was the fact that the site has no differentiation between public and private comments. This means that any comment made by a user, that has not been removed by a moderator, is visible and public to every other user. Facebook and Twitter, on the other hand, allow users to specify who can view comments and as such there is a greater expectation of privacy on those sites.

After crawling for only two hours, across two different days, over 118,000 comments were recorded into the database. This amount provides a large selection of comments for the classification and will be pared down as needed in later steps.

In order to ensure that the algorithm would be capable of handling comments regardless of site, a web crawler was used to grab comments from Twitter as well. In order to ensure that this also grabbed data that was as random as possible, data was crawled from the Twitter public sample stream which represents 1% of all Twitter messages posted at the time the web crawler was running. The only filter that was used was to restrict the 1% to English language tweets in order to ensure that it would be possible to determine if they are bullying. After running the web crawler for just a few hours, over 72,000 tweets and re-tweets were collected.

3.2 Classifying Cyberbullying

The next phase of the research was to determine the criteria that would be utilized to mark comments as cyberbullying for the purposes of the research. To that end two different methodologies were chosen to show the ability of the algorithm to conform to the terms of service of various sites allowing it to be used more generically. The first method involved the use of the Rhode Island General Laws and the US Federal Code while the second method involved the use of a theoretical website's policies on moderating comments. The algorithm was first trained on the legal method to assess it's accuracy, and then, with that complete, was retrained

on first the terms-of-service method and then a combination of both methods to determine how well it can be retrained.

While in a court room, the legal method would be decided by a determination of twelve jurors who would need to agree that something is obscene. On the internet, it is typically one to several moderators who are given the full authority to remove comments they find harmful to the ecosystem of the site. Because of this, all of the comments were classified by the researcher by developing a set of standard rules that each comment was weighed against. Every effort was made to keep the rules objective and to remove as much subjectivity from the methodology as possible. Because this tool is designed as a website filter and not a replacement for a legal jury, this methodology simply shows the ability of the algorithm to replicate the moderation techniques done by the small moderation teams.

3.2.1 Legal Method

The less restrictive of the two definitions is the definition following both Rhode Island general law as well as the U.S. Code governing federal law. Unlike the field of science, the field of law is not a strictly defined medium. Most laws are written in such a way as they need to be interpreted by the individual lawyers and judges rather then being strictly defined. Even defined terms such as Obscene are defined using tests such as the three-pronged Miller test[2]:

1. Whether the average person, applying contemporary adult community standards, finds that the matter, taken as a whole, appeals to prurient interests (i.e., an erotic, lascivious, abnormal, unhealthy, degrading, shameful, or morbid interest in nudity, sex, or excretion);
2. Whether the average person, applying contemporary adult community standards, finds that the matter depicts or describes sexual conduct in a patently offensive way (i.e., ultimate sexual acts, normal or perverted, actual or simulated, masturbation, excretory functions, lewd exhibition of the genitals, or sado-masochistic sexual abuse); and

3. Whether a reasonable person finds that the matter, taken as a whole, lacks serious literary, artistic, political, or scientific value.

As the quote shows, much of the law has to do with what an average person would find to be the case. In the court of law, this average is done by forming a 12 member jury of random citizens taken from the local area to help to establish an average. However, there is a problem with this approach. When it comes to the internet, what community standard should be applied? In the supreme court case Miller v. California [3] the opinion of the majority written by Justice Burger is that there should be no national community standard and that obscenity should be decided at the community level. However, as the years have progressed, it has been shown that not having a national standard is beginning to cause issues.

Recently the Santa Clara University School of Law reviewed the Miller Test in light of a new circuit split in attempting to apply the Miller test to the internet [4]. In the review it was pointed out that even after the Miller case was decided, there have been several other supreme court cases dealing with obscenity that have shown the Miller test is not sufficient. In each case, however, the majority opinion has been that a national standard can not be created and a community standard must be used. The major issue is which community do you apply it to? In Sable Communications v. FCC, the courts ruled that Sable Communications, a "dial-a-porn" business, must meet local obscenity laws and "may be forced to incur some cost in developing and implementing a system of screening the local of incoming calls." This will not work in the day and age of the internet however, and currently the Ninth and Eleventh circuits are split on the issue.

Because of the split and the lack of a firm standard, this research simply treated all comments that are sexual in nature as obscene. This is because the sites that are being utilized in the research are designed to be used by both minors and adults and do not contain nor allow adult content. While this is of course stricter

than allowed by the First Amendment, this tool is not designed to replace the legal juror system and is only intended to be run on a private companies website, where the company can decide which communication it deems to be obscene based upon its community.

Each of the comments were analyzed for the following to conform with all of the laws:

1. Is the comment sexual in nature?
2. Is the comment intended to seriously alarm, annoy, or bother the person?
3. And does the comment serve a legitimate purpose?

By using those three tests all of the state and federal laws are satisfied. See table 1 for four examples of comments that were classified and the reasoning as to why.

3.2.2 Terms of Service Method

The second method of determining if a comment is cyberbullying is to use the terms of service that governs a social media site to determine what they do or don't want on their site. These terms of service are enforced by the moderation team and the goal of this research is to be able to implement these terms of service in an automated fashion. For the purposes of determining what rules would be followed, the terms of service were pulled from Reddit.com which is a news aggregation website allowing numerous users to post content and comment on a variety of different subjects. Their terms of service were relatively simple and thus were much clearer as to what they did not want. Specifically they list the following under the Unwelcome content section[5]:

> While Reddit generally provides a lot of leeway in what content is acceptable, here are some guidelines for content that is not. Please

Comment	Class	Reasoning
Jorge Ramos is such a fucking race baiter. "It's because of the color of his skin!"	1	This was not bullying because while it fails step two, it is legitimate and because the subject of the harassment is not present in the conversation
We'll see this lil shit on Ellen	-1	This one is bullying because the subject of the comment is the person who posted the video and is likely to read the comment and be bothered by it
Please show this to the narrow-minded right wing fucks!	-1	While this comes close to serving a purpose, the likely-hood of a right wing reader being upset by this outweighs the need for this comment to exist
Butterface bitch. I bet if she didn't have those big ass tits the comments here would be she's ugly and everything lol.	-1	This one fails every step of the test

Table 1. Example Comment Classifications

keep in mind the spirit in which these were written, and know that looking for loopholes is a waste of time.

Content is prohibited if it

1. Is illegal
2. Is involuntary pornography
3. Encourages or incites violence
4. Threatens, harasses, or bullies or encourages others to do so
5. Is personal and confidential information
6. Impersonates someone in a misleading or deceptive manner
7. Is spam

Because of rule 1, everything that is found to be against the legal method would also fall against the terms of service method. However, the terms of service method adds some extra tests such as encouraging violence or encouraging others to harass or bully. Thus it is possible to fail the terms of service method even if technically the speech is legal and thus protected by the first amendment.

Furthermore, Reddit goes on to define what they classify as harassment:

> We do not tolerate the harassment of people on our site, nor do we tolerate communities dedicated to fostering harassing behavior.
>
> > Harassment on Reddit is defined as systematic and/or continued actions to torment or demean someone in a way that would make a reasonable person conclude that Reddit is not a safe platform to express their ideas or participate in the conversation, or fear for their safety or the safety of those around them.
>
> Being annoying, vote brigading, or participating in a heated argument is not harassment, but following an individual or group of users, online or off, to the point where they no longer feel that it's safe to post online or are in fear of their real life safety is.

A major difference between this and the legal method is that because the terms of service are not designed as a legal document, the spirit of the rules are more important than the strict wording, and moderators are given more leeway in deciding what is unwanted on the site. For each subreddit, the creator can define who they wish to have as moderators, and those people have the ability to suppress or remove any comments that violate not only Reddit's terms of service, but also whatever other rules they create for that sub-forum.

One of the goals is to allow for retraining to handle new situations as the laws and culture change. In order to ensure a sufficient difference between the legal method and the terms of service method, the comments that were used in the legal method were reclassified as if they were posted on a fictional site that simply had the rule: Content is prohibited if it discusses politics and/or religion. This will ensure a substantial difference as most of the political talk in the comments did not rise to the level where it could be deemed illegal.

Finally, after testing just the terms of service method, the combination of both the legal method and the terms of service method will be run, simulating a more

realistic portrayal of the terms of service actually used on websites. For both the terms of service method and this method, the only test run is to confirm that by simply retraining, the algorithm is capable of generating a useful model without the need to re-optimize all of the parameters.

3.3 Support Vector Machine Model

For the model to reach its optimum potential there are a number of different parameters that were optimized. The writers of LIBSVM [6] recommend that the optimization take place with, first knowledge of the data set, and then a grid search over the relevant parameters. This is, however, only done over the parameters directly built into the model. For this research, there are additional parameters which will each be optimized individually.

Before any parameters can be optimized, the range needs to be established on each of the parameters to ensure that the testing covers all necessary values. The first parameter is the number of comments that are used for testing purposes. While a balanced data set will be used to ensure that there is no bias introduced to one class or the other, we still need to determine the optimum number of comments to ensure that we are always getting a reasonable training set while also keeping complexity, and thus training time, low. For this reason, we will test from 50 comments (25 bullying) to 500 comments (250 bullying) in steps of 50.

Next, we need to test which of the four possible SVM kernels will perform best on the data. For the initial tests, we will be using the linear kernel which is the faster kernel and contains the smallest number of additional parameters to optimize. Once we have optimum values for the other parameters, we will also test the Polynomial and the Radial Basis Function (RBF) kernels. We will not be testing the Sigmoid kernel because of research done at the National Taiwan University. They have shown that, not only is the kernel not positive semi-definite

(will not find a solution for all valid values of parameters), but the kernel also does not perform better than the RBF kernel in general since it was designed to mimic the function of neural networks within a SVM [7].

With any of the three kernels, one of two error weighting parameters, C and ν, must be used. C is the first soft margin parameter that sets the cost of an error to allow potentially mislabeled data to exist across the boundary. It can take any positive value and, due to its functionality, an exponential grid search is the best method to find the optimum [8]. To this end, C takes the form of 2^k with k ranging from -5 to 15. ν is a newer soft margin parameter that replaces C in order to reduce the allowable values from all positive numbers to 0 to 1. ν can be shown to have the same optimal solution set as C [9], and as such does not completely replace it, but, in certain circumstances, one may perform better than the other and as such both will be tested. Because ν can be any number between 0 and 1 the test will begin at 0.1 and go to 0.9 in 0.1 increments.

With the polynomial or RBF kernel there is an additional gamma parameter which could be found through grid searching. Alternatively, research has shown that gamma can be estimated mathematically and to do that a C# package called the Accord.NET Framework was used [10].

The final free parameters are the degree and Coef0 which are only used in the polynomial function. The degree will be constrained from 1 to 10 unless 10 is found to be optimal at which point we will expand the range. Coef0 on the other hand is only used in special data sets and in general should be fine as 0. To ensure this isn't one of the special cases, we will test with a value of 0 and 1.

The next step to training the SVM is to determine what attributes are going to be used for the data. With even just 200 comments, the number of unique 6-grams was over 19,000 and due to this, the generation of the training file was taking

in excess of 2 hours, while each comment was taking one minute just to generate the data file to be passed into the SVM. In order to reduce this time without substantially reducing the accuracy of the model, the entropy of the n-grams were calculated.

This introduces two more parameters to be optimized in the testing. First what length of n-gram provides the best accuracy while minimizing the amount of time required. The other is what percentage of the n-grams should be utilized of the ones that have an entropy greater than 0. For the first one, we will test n-gram lengths from 1 to 10, while for the second we will test 1 to 10 in 1% increments and, assuming 10% is the optimal, we will increase by 5% increments until performance degrades. In each case, the n-grams used are marked as being used in the training set and are then calculated for each message. This marking ensures that the same n-grams will be used for the testing comments later, and because the n-grams are then sorted by a unique identifier, ensures that there will be no discrepancies between the data sets.

With all of the data being passed in, there is the potential that the support vector machine may not perform well due to the number of comments that are not on the decision surface that may still be influencing the model. In an effort to reduce that and the time that the cross validation will take, the research is borrowing an experimental function from Leandro Costa at URI who is developing a method to reduce the training set[11] built upon the K-Nearest Neighbors algorithm (knn)[12]. This method calculates the distance of each comment from every other comment and then uses that to try to determine if it is against the decision surface or not. This will add another parameter as the method takes a count of how many closest points to look at to test if it is close to the decision surface. A low number here will greatly trim the data to only those points directly next to

$$SampleSize = (z * \frac{StdDev}{MarginOfError})^2 \quad (9)$$

Figure 1. Sample Size Calculation

an opposing point, while a higher number will result in more time being spent on training. This was tested from 10% up to 100% in 10% intervals.

After all of these individual parameters have been tuned based on the best performing linear kernel, the next step was to perform a grid search on the three possible kernels and all of their parameters. While in the first step only the linear kernel was utilized to kick-start the process, here all of the possibilities were run to ensure the optimum was found.

In an effort to help distinguish good performing parameters, the full testing set of data is run through each created model and the training time and testing time are recorded along with the results of the test. Next, all first run tests were duplicated 10 times and all this data was graphed to help give a visual representation to the performance.

Finally, the average standard deviation was calculated and used to find the required sample size (figure 1) to ensure that number of runs were enough [13]. For this calculation, the 95% confidence level was used which results in a z-score of 1.96, and a margin of error of 5%. If this resulted in more runs being required, they were conducted as appropriate.

Once the full run-through of the initial parameters is completed, a second run-through was done with the optimums found on the first attempt to ensure that the starting parameters did not influence the results. Instead of doing 10 runs on each test, the number that was calculated on the first set of runs was used instead. With those done a new sample size calculation was done to ensure that it is still a proper result.

$$n = \frac{(CV^2 * z^2)}{MarginOfError^2} \tag{10}$$

Figure 2. Coefficient of Variation Sample Size Calculation

The primary motivation for doing this second run through was due to the internal stratification that LibSVM utilizes on their implementation of cross-validation. When an attempt was made to calculate the 95% confidence interval utilizing bootstrapping, the numbers returned were often higher than the original number. For this reason, there was not a high level of confidence that one pass on the variables would be sufficient to get the optimum model parameters.

The final test of the system will be to calculate the throughput of the system. Because the units of the standard deviation in this test will be in milliseconds, instead of an accuracy, the coefficient of variation will be computed instead since it removes the units[14]. The coefficient of variation is then used as shown in figure 2 to calculate the sample size[15]. The same z-score of 1.96 and margin of error of 5% was used as the standard deviation method.

3.4 System Design

The system designed for the dissertation is broken into four separate sections. The first section is the website which facilitates access to the rest of the pieces. The second section is the database where all of the data is stored and worked on by the other sections. Third is the service where the data is actually processed, manipulated and finally passed into the Support Vector Machine. Finally is the Support Vector Machine itself where the model is trained and the data is classified.

As you can see in figure 3, the overall system is complex. This diagram represents the ideal system in an actual website setup. The system used by the research contains identical functionality but at a smaller scale, combining all servers and databases onto one virtual machine. In the following sections each piece will

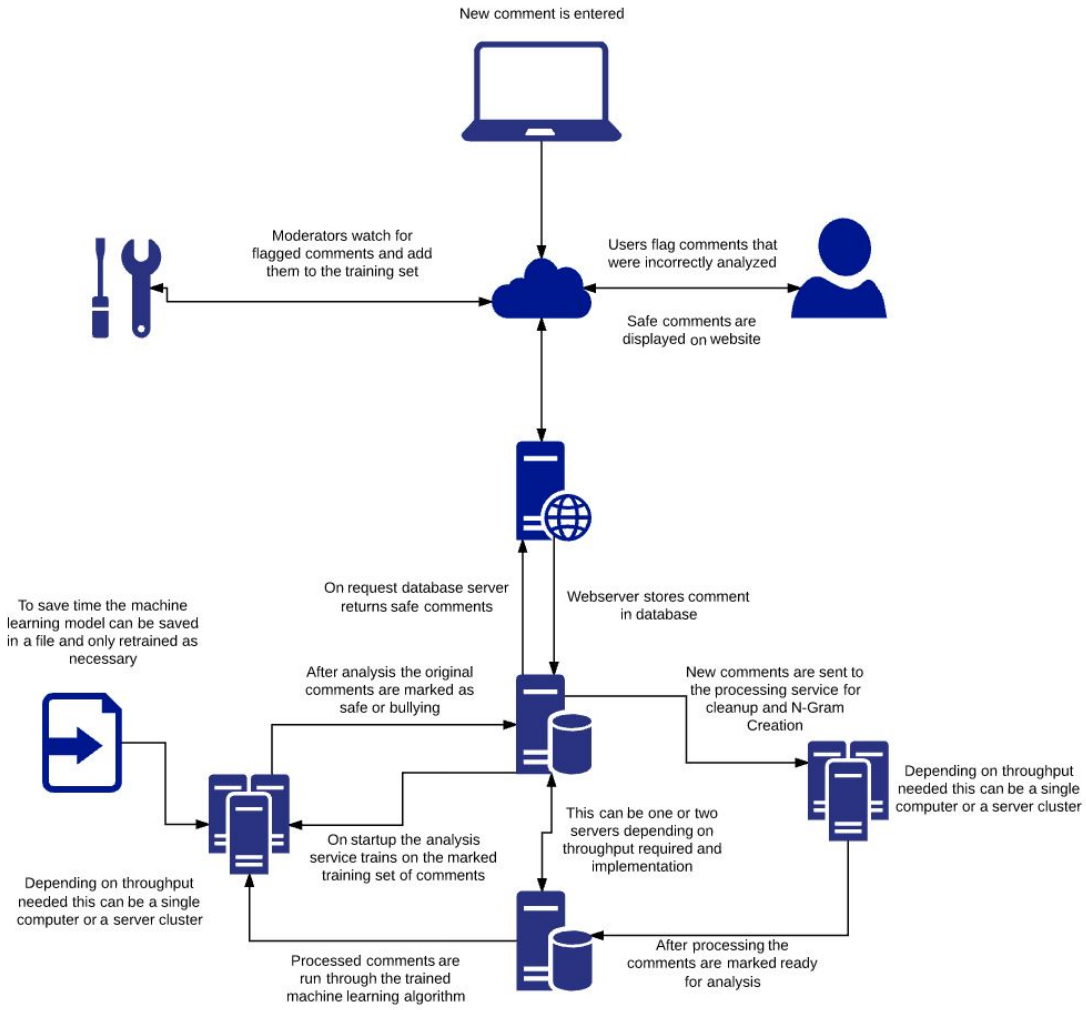

Figure 3. Overall System

be broken down to make it clearer what is happening.

3.4.1 Website

In order to show that this algorithm is fully functional regardless of the website, the website was written separately from the algorithm and simply utilized SQL stored procedures to function. Because of this, it can be shown that any web language capable of interacting with a Microsoft SQL Server database would have the same functionality.

The first site is designed to show some of the potential functionality that can be built into a website if required. This site is written in ASP.NET, utilizing the C# language and uses a very utilitarian visual style. It first takes and allows the user to submit a comment for analysis. After it uploads the comment, it waits for the comment to be processed, and then to be analyzed while letting the user know the steps are happening. Finally, once the analysis has been completed, it lets the user know if the comment that has been posted is considered bullying or not.

Another simple website was created in C# to allow for easy classification of data. This site selects a random comment from the database and then provides two buttons, one for Bullying and one for Non-Bullying. This site is the principle way in which the data was classified. After the initial classification, the site was also used to reclassify the comments for the other tests required.

Finally, a website was setup that captures the current statistics to get an accurate count of the throughput of the system. This website shows how many comments are waiting to be processed and analyzed, and shows the average time required to handle a single comment.

In a fully implemented system, the website component would function as shown in figure 4. In this ideal system, there are three possible classes of user. There are the posters who are adding comments to the site, the readers who are

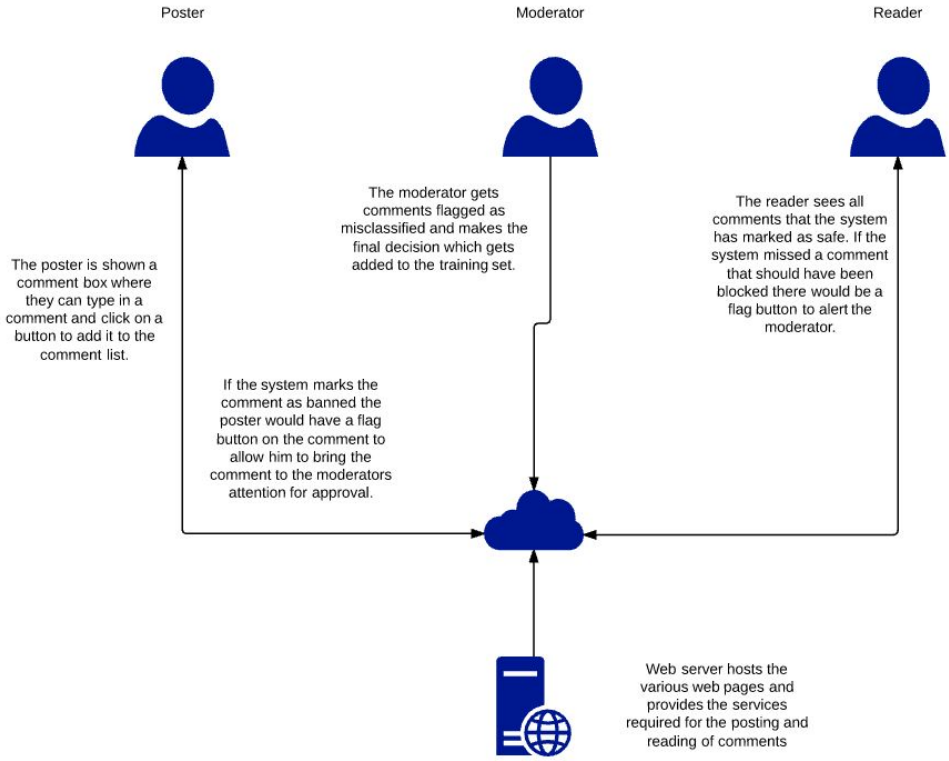

Figure 4. Website System Diagram

the target of the comments, and the human moderators who handle the misclassifications from the system. The posters will see if their comments have been marked as banned and have a method to flag their comment for review by a human moderator. The readers will have a similar flag to mark that a comment that came through should have been banned by the moderator. Moderators will no longer need to review every comment, but instead can focus their effort on the comments that the two other classes have marked as incorrect. While this in theory could rise to the level of every comment getting marked, policies should be setup to prevent this. As an example, there could be a strike system that if you attempt to post inappropriate comments and flag them as being misclassified, and the moderator finds in favor of the system, you are banned for a certain amount of time up to permanent. This will ensure, that while some users will still flag it for human review when they know it is bad, the majority should just move on to either toning down the message of their comment or not commenting altogether.

The system, as shown, is not what was fully implemented because in the case of this research there are no readers. Thus, the first website was setup to mimic the functionality of the posters and the second site is similar to that of the moderators.

3.4.2 Database

For this research two different databases were utilized. Both were run on SQL Server 2012 (11.0.2100). The first database was used to store the data that was collected from both YouTube and Twitter. This data was stored in plain text with no processing done to it, outside of the stripping of user identification information, as well as storing the original source and holding the manual classification for any of the users.

The second database is the more important and is the database where all of the data is held for the processing and analysis sections. This database contains

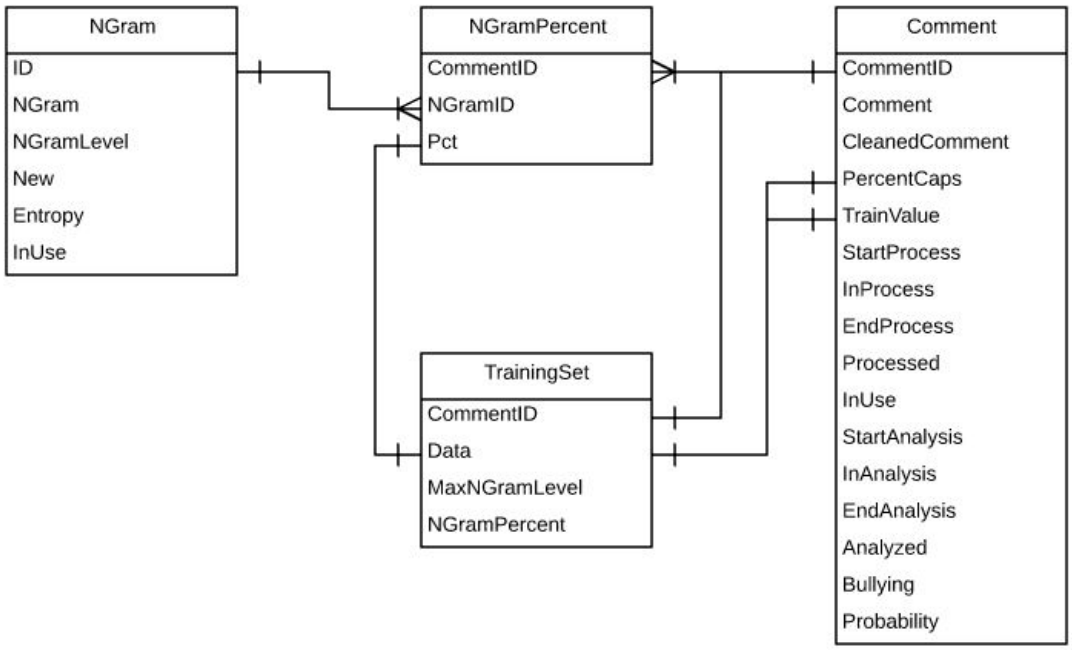

Figure 5. Database System Diagram

four primary tables which are shown in figure 5.

The first is the Comment table which contains both the original comment as well as a cleaned version that has had white space and symbols removed. It also stores comment specific information such as the percent of the comment that is capitalized, the manual classification value, and if it was processed or analyzed yet, when the service occurred, and by which thread.

The second table is the NGram table which contains the list of every n-gram that was found in any processed comments up to the 10-grams. That means that it contains every grouping of single words all the way though ten consecutive words. The other important feature of the NGram table is that it stores if the n-gram is new, as in, it was added from a comment after the SVM was last trained, as well as the entropy of the n-gram which can be used to decide which to use as attributes. This information is important because any n-gram that was added after the training run is a feature that has not been utilized by any of the training

set and as such cannot be a feature of any classification set without retraining.

The third table combines the first two in order to increase the speed of generating the training and testing sets. Each comment has every n-gram it contains in this table along with the percent of the comment that n-gram represents. This allows for table operations to create the data rows instead of utilizing cursors which wastes the databases time and greatly increases I/O.

The final table holds the set of training data that is being used on this run of the service. While at this time the script simply truncates the table and refills it with a random subset of data, it could also be used to hold special comments that a site decides should always be used in the training.

Finally, there are a number of secondary tables that were used to store temporary testing sets as well as results from each of the analysis runs. These are used only for the purposes of optimizing and testing this dissertation and would not be present in a product utilizing this algorithm.

Along with the three tables, there are also some stored procedures that are used to handle tasks such as locking a record to a certain thread or selecting all unprocessed comments etc. This is done to ensure consistency across threads and to ensure data consistency is kept at every stage. These stored procedures are, for the most part, dependent on the individual implementation, but there are two that will need to be in any implementation. The first calculates the entropy of the n-grams in regards to the comments that are being utilized to train the SVM. The second procedure clears out the training set and then randomly selects an equal number of both classes and sets up their features for the machine learning.

3.4.3 Services

One of the goals of the algorithm is to allow for as much multithreading as possible to ensure that the algorithm can scale as needed to handle large websites

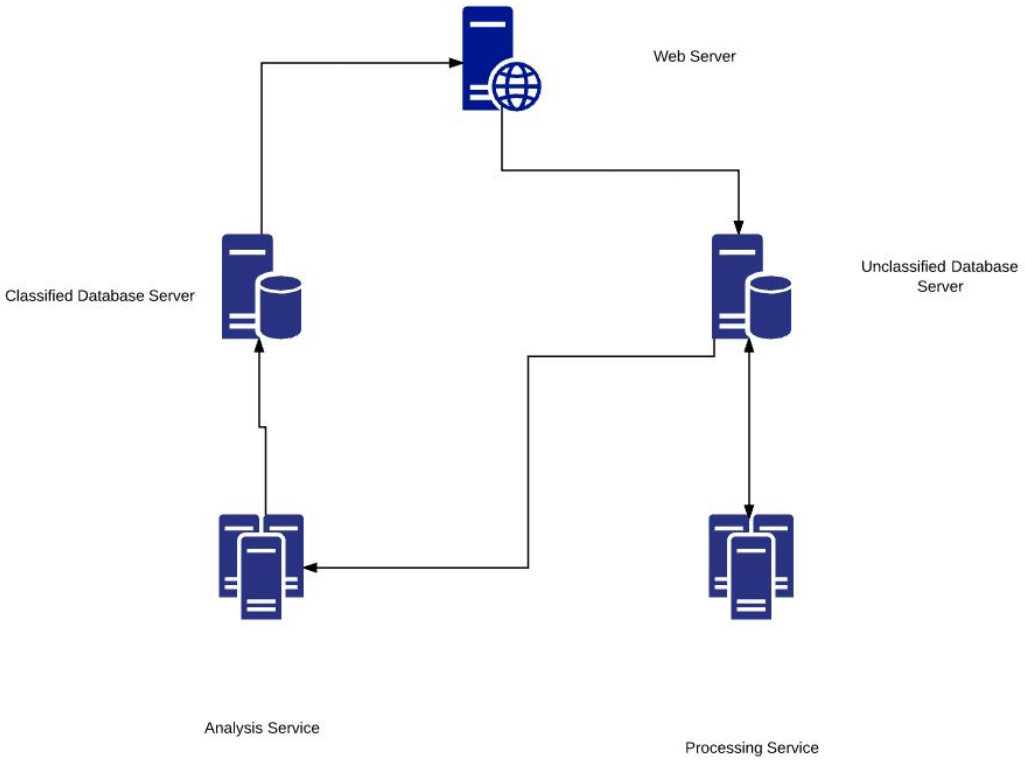

Figure 6. Services System Diagram

such as Facebook. To this end, the programming for the processing and analyzing of the comments were created in services that are run on the server. The main workload in this case is spread across two different services that can run concurrently to spread the work out as much as possible. In fact, as shown in figure 6, there is no reason these services could not be setup to run on multiple server clusters to handle as many comments as needed.

As shown in the diagram, the first step is the web server where the new comments come in prior to processing and the comments marked safe are hosted for the readers. The incoming comments are first placed into a database where they can undergo the initial processing. At the database level, the comments have all of their non-alpha numeric characters stripped out and replaced by | along with all spaces. Multiple | are also then combined so that there there is a single | at the start and end of the message as well as in-between each unigram.

The first service that is utilized by the comments is the Processor. As soon as a new comment is inserted into the database, the processor is run against it in order to calculate the different stats that are needed for the classification, such as the percent capitalization and the creation of the n-grams needed. In order to speed up later steps, the percent of the comment each n-gram represents is also calculated at this step and stored in a separate table. This ensures that everything that is needed for the SVM classification is already handled so that the only thing the analysis service needs to handle is the classification itself. Because the processor only handles a comment at a time per thread, it is easy to increase not only the number of threads available, but to also scale this across multiple servers since the results do not rely on any factors external to the single comment.

The other service is the one that handles the SVM classification. While the training of the SVM is a time consuming task that must be completed at the

startup of the service, once the SVM is trained, the service is setup to allow multiple comments to be classified at the same time with proper distributed multithreading. This SVM model can either be recreated at each service startup on every server, or after the first service is brought online and trained, the SVM model can be stored to a file which is then utilized on startup of the other servers to reduce training time and allow for a more efficient spinup of additional servers during peak times.

Finally, after the analysis is complete, the result is stored in either a new database or, in the case of this small scale research, as a simple flag in the comment table. This flag is what is utilized by the webserver to indicate if the comment should be displayed to readers or hidden as harassing message.

3.4.4 Support Vector Machine

Because the purpose of the research was to use an existing machine learning algorithm in a new and novel way, the project used a wrapper called LibSVMsharp [16] which calls a C++ dll implementation of LIBSVM [6], fully implementing Support Vector Machines within the .Net language. This allowed the research to focus on the training and testing of the model rather then focusing on re-implementing the existing Support Vector Machine algorithm which may have introduced both additional complexity as well as being an added vector for bugs.

There were several changes made to the DLL's utilized by the wrapper, however, to optimize the functionality of the libsvm implementation. All of the changes were generated from the libsvm faq page in order to better parallelize the LIBSVM dll and allow it to better utilize a multicore system for training and predicting. This DLL was based on the 3.21 version of the LIBSVM code and LibSVMSharp was rebuilt using this code as well. Both the DLL and the Wrapper were also compiled in 64-bit instead of their normal 32-bit version in order to accommodate the large data sets required for the research.

List of References

[1] YouTube. "Popular right now." [Online; accessed 20-October-2015]. 2015. [Online]. Available: https://www.youtube.com/playlist?list=PLrEnWoR732-BHrPp_Pm8_VleD68f9s14-

[2] Department of Justice. "Citizen's guide to u.s. federal law on obscenity." [Online; accessed 20-October-2015]. July 2015. [Online]. Available: http://www.justice.gov/criminal-ceos/citizens-guide-us-federal-law-obscenity

[3] "Miller v. california, (1973)," June 1973, [Online; accessed 1-May-2016]. [Online]. Available: http://caselaw.findlaw.com/us-supreme-court/413/15.html

[4] E. M. Laird, "The internet and the fall of the miller obscenity standard: Reexamining the problem of applying local community standards in light of a recent circuit split," *Santa Clara Law Review*, vol. 52, 2012, [Online; accessed 1-May-2016]. [Online]. Available: http://digitalcommons.law.scu.edu/cgi/viewcontent.cgi?article=2735&context=lawreview

[5] Reddit. "Reddit content policy." [Online; accessed 14-December-2015]. 2015. [Online]. Available: https://www.reddit.com/help/contentpolicy

[6] C.-C. Chang and C.-J. Lin. "Libsvm – a library for support vector machines." [Online; accessed 12-September-2012]. Apr. 2012. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[7] C.-J. L. Hsuan-Tien Lin. "A study on sigmoid kernels for svm and the training of non-psd kernels by sm o-type methods." [Online; accessed 14-May-2016]. 2003. [Online]. Available: www.csie.ntu.edu.tw/~htlin/paper/doc/tanh.pdf

[8] C.-C. C. Chih-Wei Hsu and C.-J. Lin. "A practical guide to support vector classification." [Online; accessed 3-January-2016]. Apr. 2010. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

[9] C.-J. L. Chih-Chung Chang, "Training nu-support vector classifiers: Theory and algorithms," *Neural Computation*, vol. 13, pp. 2119–2147, 2001, [Online; accessed 11-June-2016]. [Online]. Available: http://ntur.lib.ntu.edu.tw/bitstream/246246/155217/1/09.pdf

[10] C. Souza. "Accord.net framework." [Online; accessed 14-May-2016]. 2012. [Online]. Available: http://accord-framework.net

[11] L. Costa, "Voting neighbours: Svm border selection algorithm based on knn," [PhD Thesis, to appear].

[12] "k-nearestneighbors." [Online; accessed 1-October-2016]. 2013. [Online]. Available: http://www.statsoft.com/textbook/k-nearest-neighbors

[13] M. L. R. III. “Basic concepts of statistics - class 23.” [Online; accessed 1-October-2016]. 2010. [Online]. Available: http://www.unc.edu/~rls/s151-2010/class23.pdf

[14] L. Green. “Mean, mode, median, and standard deviation.” [Online; accessed 18-February-2017]. 2008. [Online]. Available: https://www.ltcconline.net/greenl/courses/201/descstat/mean.htm

[15] “Confidence levels and sample size.” [Online; accessed 18-February-2017]. 2000. [Online]. Available: http://download.ctpp.transportation.org/training/mod_8_part1.pdf

[16] C. Erhan. “Libsvmsharp.” [Online; accessed 15-September-2015]. Apr. 2015. [Online]. Available: https://github.com/ccerhan/LibSVMsharp

CHAPTER 4

Findings

This section begins with the findings from the first two optimization runs where each of the parameters are set to their optimum value in turn. For this research, the optimum is a balance between the highest accuracy and the best performance. With the multiple parameters optimized, it then moves on to the average accuracy against all of the classified data, first using the same legal method that the parameters were optimized with, and then with the terms of service method and the overall method. After that, it tests the Twitter comments to ensure that it works across sites. Finally, it will test two methods of retraining to see which performs better followed by a performance test.

All graphs in this section will show the average cross validated accuracy (Avg. Accuracy) and the average weighted real accuracy (Avg. Real Accuracy). The trend lines and confidence bands, as calculated by Tableau[1], are also shown for each of the values to help illustrate what is going on with the data.

4.1 Preliminary Findings

These findings show the results of the optimization steps taken on the support vector machine. All tests were run in a virtual machine running Windows Server 2012 R2 and were run on a desktop with an Intel i7 5930k processor overclocked to 4.2 GHz, 32 GB of Crucial DDR4-2400 RAM and a Samsung 850 EVO SSD. 6 Cores and 27.9 GB of RAM were assigned to the virtual machine.

4.1.1 Initial Run

This is the first run-through on each of the parameters to get an initial best case.

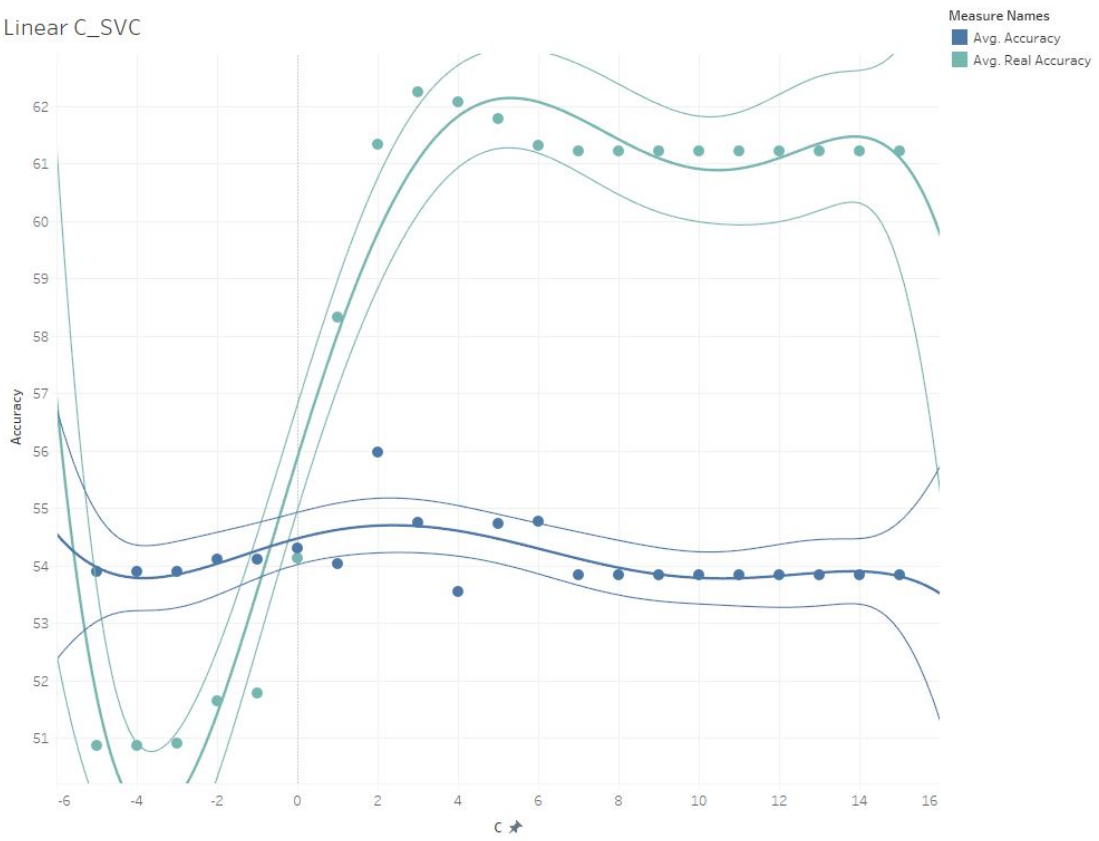

Figure 7. Initial C Accuracy

Initial Grid Search

The first grid search is used to choose the initial C or ν value on the linear kernel so that we could begin with a decent starting point. The linear kernel is used in the initial run because it has the smallest number of input parameters. For the other parameters this run used 100 comments as the training set, 10% of 1-grams, and a knn level of 10%.

On this test, the full 10 runs were completed and resulted in an average standard deviation of 7.37. This means that the minimum number of runs necessary to be sure of the results was 8.35.

In order to properly visualize the data, the C and ν sets were split into two different graphs so that their individual effects could be seen. The first graph

C	Nu	Accuracy	Std Dev	Positive	Negative
-5		53.89	8.42	67.74	34.09
-4		53.89	8.42	67.74	34.09
-3		53.89	8.42	67.74	34.09
-2		54.11	8.13	67.11	35.98
-1		54.11	8.13	67.45	35.98
0		54.30	8.61	66.90	41.29
1		54.02	7.74	70.73	45.83
2		55.96	5.2	68.91	53.79
3		54.76	9.06	64.70	59.85
4		53.55	8.6	61.46	62.88
5		54.74	8.57	60.97	62.50
6		54.78	5.35	59.93	62.88
7		53.84	6.02	59.45	62.88
8		53.84	6.02	59.45	62.88
9		53.84	6.02	59.45	62.88
10		53.84	6.02	59.45	62.88
11		53.84	6.02	59.45	62.88
12		53.84	6.02	59.45	62.88
13		53.84	6.02	59.45	62.88
14		53.84	6.02	59.45	62.88
15		53.84	6.02	59.45	62.88
	0.1	53.87	6.77	59.08	63.26
	0.2	54.75	6.21	59.90	62.88
	0.3	54.79	8.2	61.57	61.89
	0.4	52.58	7.89	62.44	61.36
	0.5	53.64	7.98	62.33	61.74
	0.6	54.60	8.61	64.09	60.61
	0.7	56.48	8.07	65.64	55.68
	0.8	52.60	9.45	74.24	47.39
	0.9	48.72	9.2	81.39	32.58

Table 2. Initial Grid Search

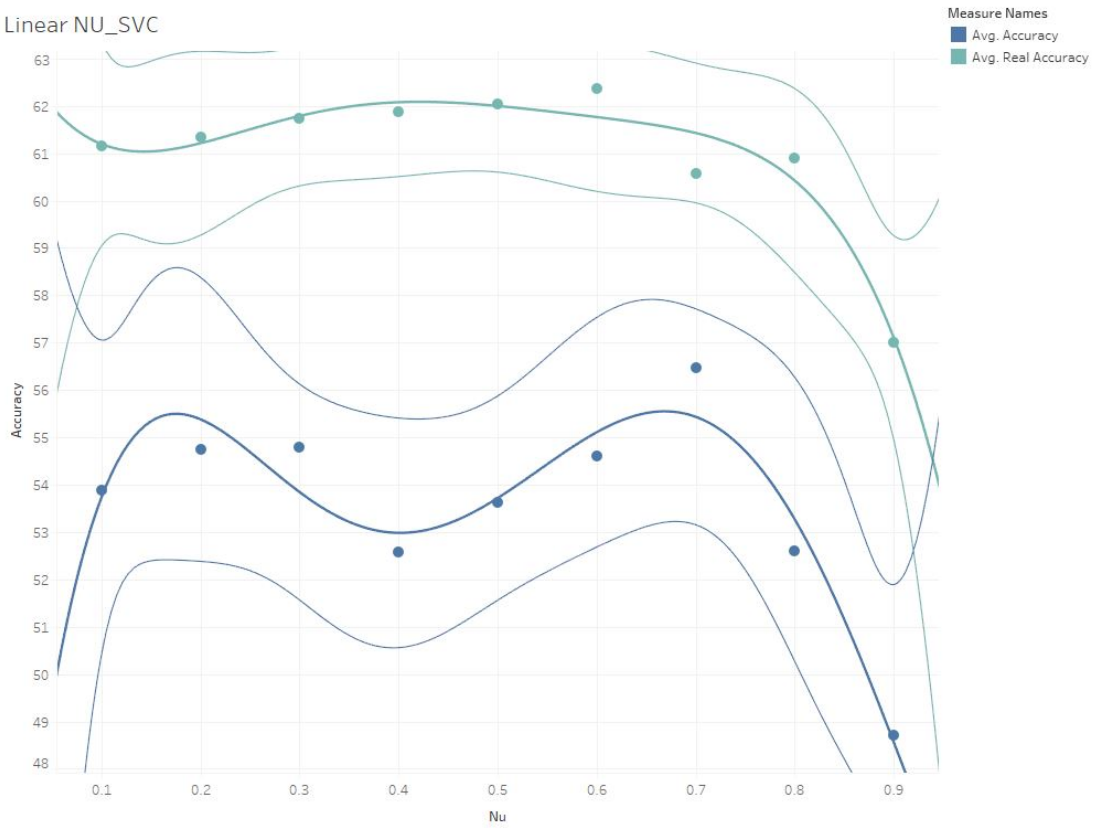

Figure 8. Initial ν Accuracy

shown in figure 7 illustrates the effect of varying the C parameter on the linear kernel. In this graph, you can see that the best the cross validation achieves is 55.96% accuracy with a C value of 2. However, the slope is flat on the cross validated accuracy and close to logarithmic on the other accuracy. This shows that the best possible C value is between 2 and 7 since the weighted real accuracy begins dropping slightly.

The second graph in figure 8 shows the effect of varying the ν parameter on the linear kernel. In this graph, you can see that the best the cross validation achieves is 56.48% accuracy with a ν value of 0.7. In these tests, the real accuracy was a relatively flat slope that turned down at high ν values.

As figures 9 and 10 show, the ν_SVC method with a ν value of 0.7 is the

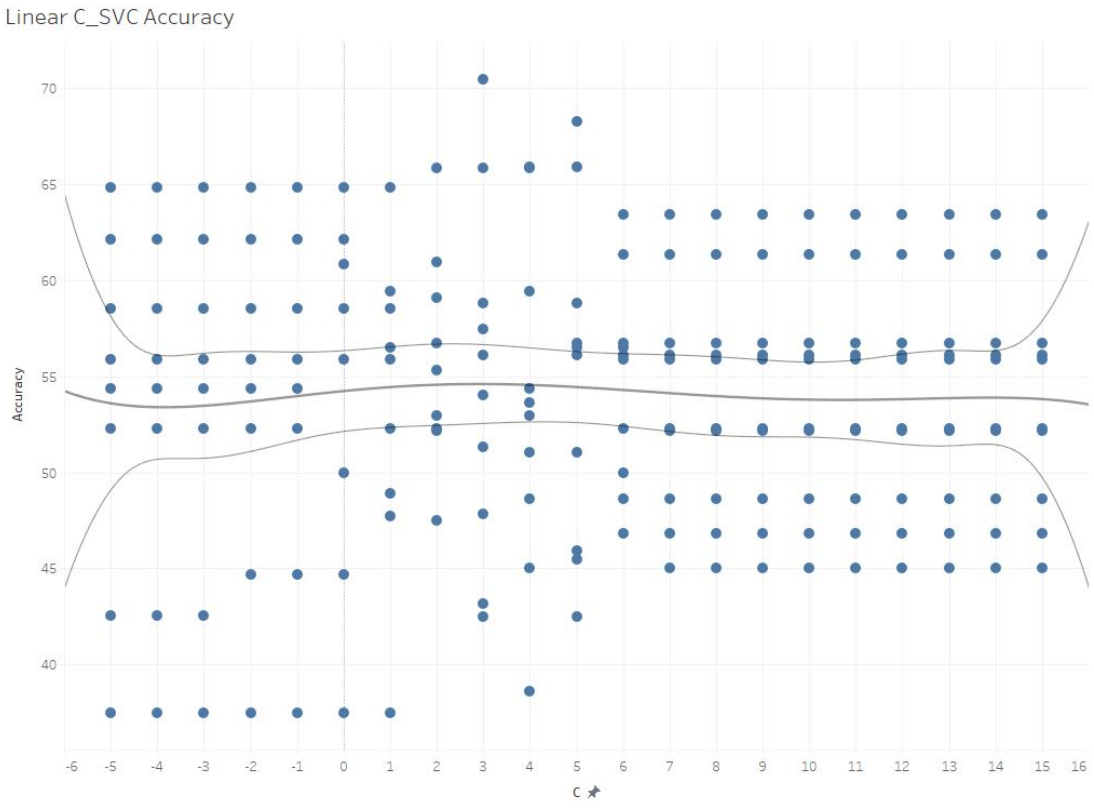

Figure 9. Initial C Cross Validated Accuracy

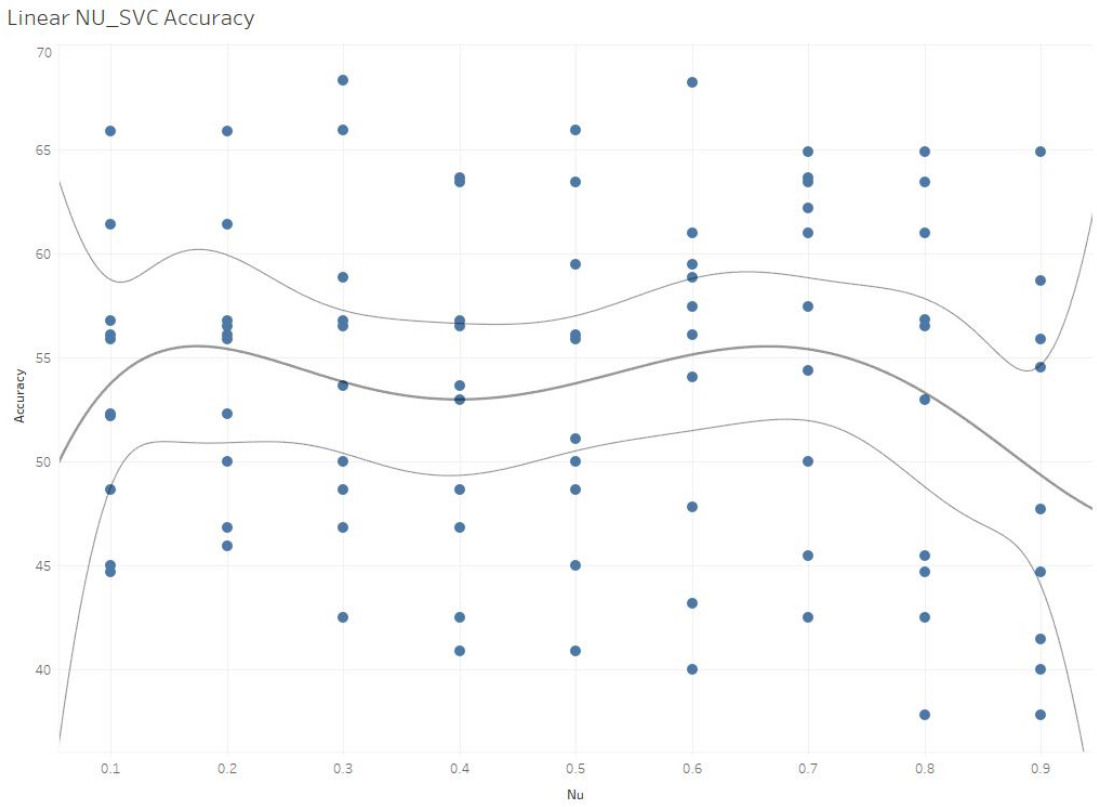

Figure 10. Initial ν Cross Validated Accuracy

Size	Accuracy	Std Dev	Train Time	Test Time	Positive	Negative
50	47.63	8.31	33	1421	63.74	53.59
100	62.40	6.21	21	2657	60.12	67.80
150	63.62	3.14	34	4032	64.46	69.32
200	64.83	2.87	48	5337	69.47	68.94
250	67.00	4.16	71	6561	69.26	72.35
300	68.13	2.52	92	7895	71.47	72.73
350	67.54	1.31	120	8549	72.81	73.11
400	68.25	1.6	131	10496	72.11	77.27
450	68.67	1.69	179	11247	73.45	76.14
500	69.90	1.26	234	12707	74.11	77.27

Table 3. Initial Number of Comments

parameter with the highest cross validated accuracy. This is because even though ν 0.7 has a higher deviation than the next best, which is C of 2, it has its average being pulled down by a few bad runs with 4 runs over 60% versus only 2.

Number of Comments

With the initial linear kernel grid search done, the next parameter to isolate was the number of comments needed for the training set. From the last test, we are using the ν_SVC linear kernel with a ν value of 0.7, 10% of length 1 n-grams, and a knn level of 10%. To ensure that one class does not overwhelm the other, the training set is balanced so in each case half is pulled from each class.

On this test the full 10 runs were completed and resulted in an average standard deviation of 3.31. This means that the minimum number of runs necessary to be sure of the results was 1.68.

As figure 11 shows, increasing the number of comments improves the accuracy of the model with a logarithmic curve. Under 150 comments the accuracy quickly drops off making small training set sizes too inaccurate even though they will perform faster.

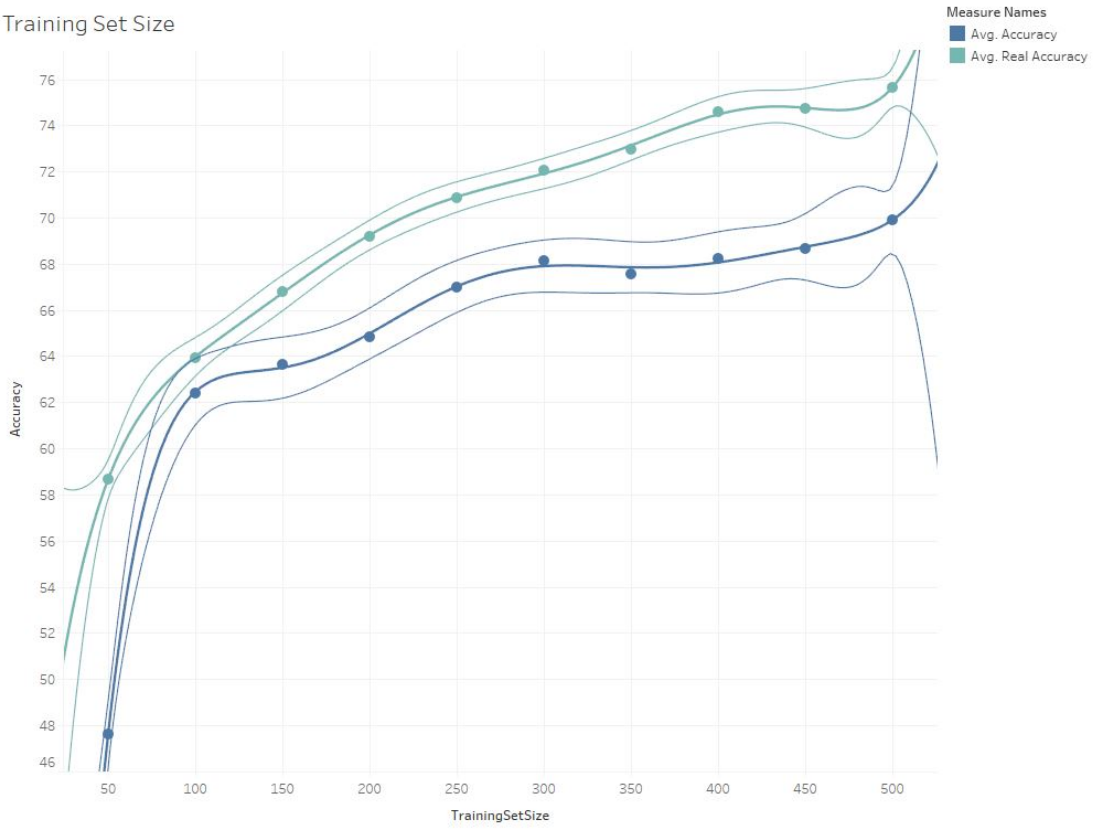

Figure 11. Training Set Accuracy

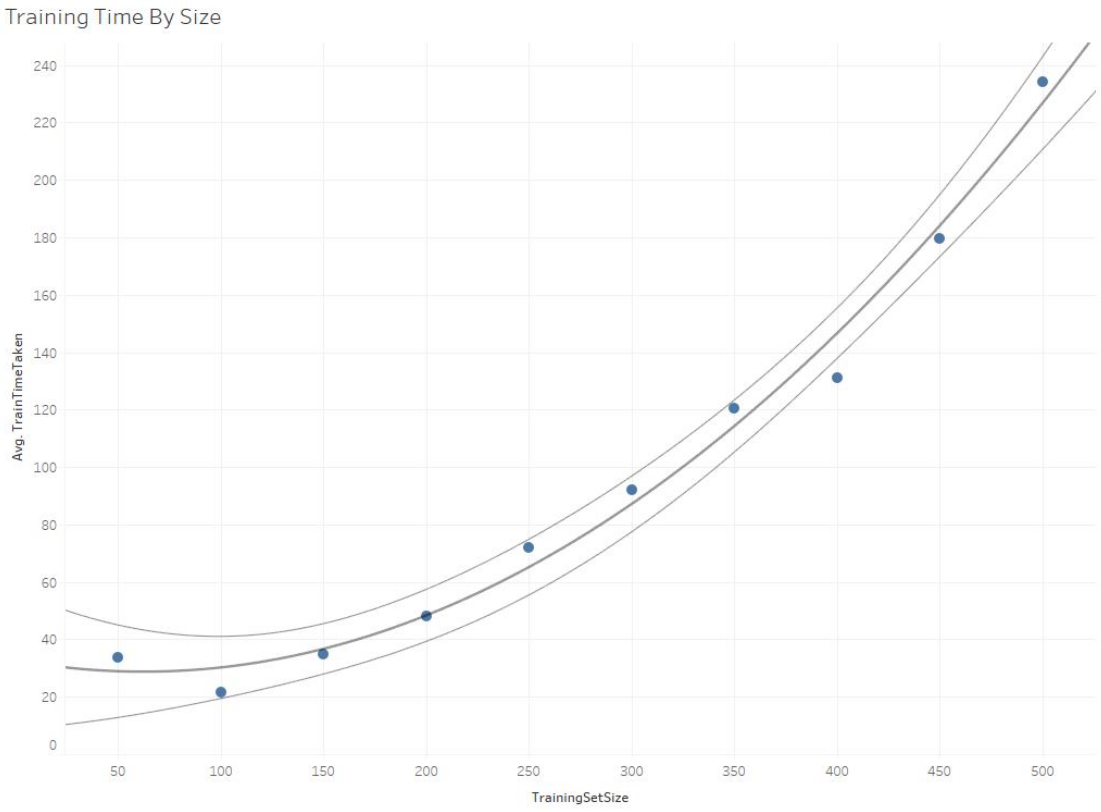

Figure 12. Training Set Training Time

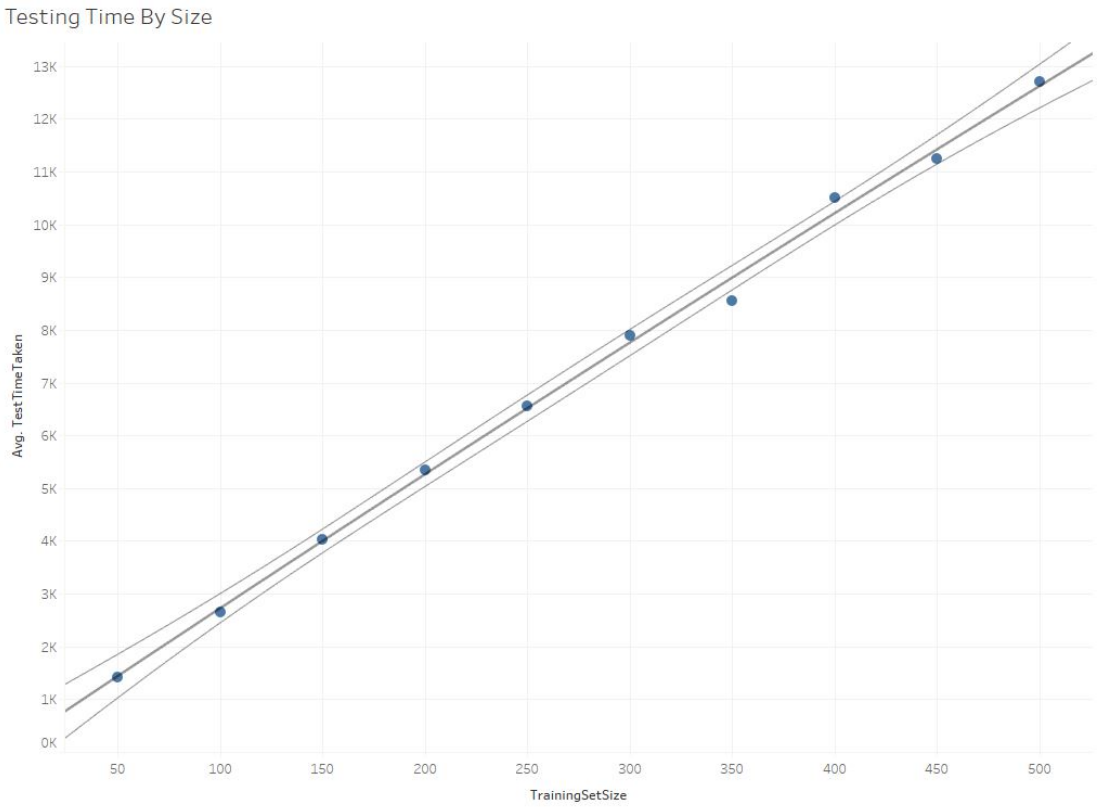

Figure 13. Training Set Testing Time

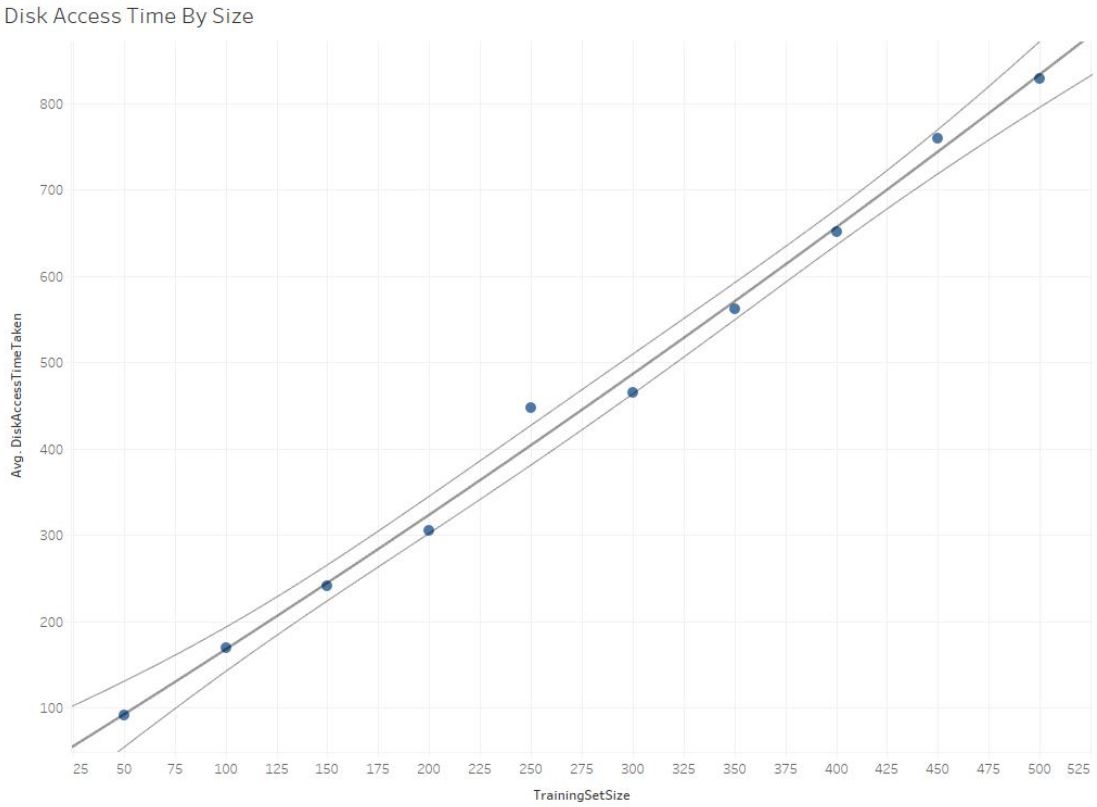

Figure 14. Training Set Disk Access Time

Beyond just the accuracy of the number of comments, there is also a significant difference in time between the different sizes. As shown in figures 12, 13, and 14, the time required is increased as the size of the training set increases. The disk access and testing time both increase linearly as the number of comments increases, but the training time increases exponentially as the comment size increases. This is partially due to the increased time it takes to compute the k-nearest neighbor as that is an $O(n^2 * d)$ algorithm.

With these graphs, the best size was determined to be 300 comments since it maintained the best ratio of accuracy and real accuracy vs the training and testing time. In an ideal world, where time is not a factor, this should be set to the maximum size possible, but since time is always a factor, a size must be chosen that will allow the throughput required while giving acceptable accuracy.

N-gram Length

With the number of comments optimized, the next phase was to find the best the n-gram length. This test used 300 comments as the training set, the ν_SVC linear kernel with a ν value of 0.7, 10% of the n-grams at the various levels, and a knn level of 10%.

On this test the 9 runs were completed and resulted in an average standard deviation of 2.87. This means that the minimum number of runs necessary to be sure of the results was 1.26. There are only 9 runs because during the analysis it was discovered that something happened during run 3 that caused the training time to increase to greater than 2,000 seconds and so it was excluded from the analysis.

Figure 15 shows that the cross validated accuracy does not alter much regardless of the n-gram length, while the upper and lower bounds as well as the real accuracy doesn't stabilize until at least the 3-gram.

Length	Accuracy	Std Dev	Train Time	Test Time	Positive	Negative
1	61.67	1.68	89	6586	71.45	71.73
2	60.90	2.37	111	10493	68.90	78.90
3	60.43	3.6	126	11304	68.93	80.25
4	61.17	2.91	123	12938	66.68	82.28
5	60.87	3.05	132	13412	69.19	80.25
6	62.03	4.07	117	13376	72.48	78.90
7	60.60	3.48	126	13247	68.07	81.43
8	61.47	1.56	114	14434	69.08	82.35
9	60.73	2.85	145	14484	68.60	82.35
10	61.37	3.07	128	15156	70.05	80.67

Table 4. N-gram Length

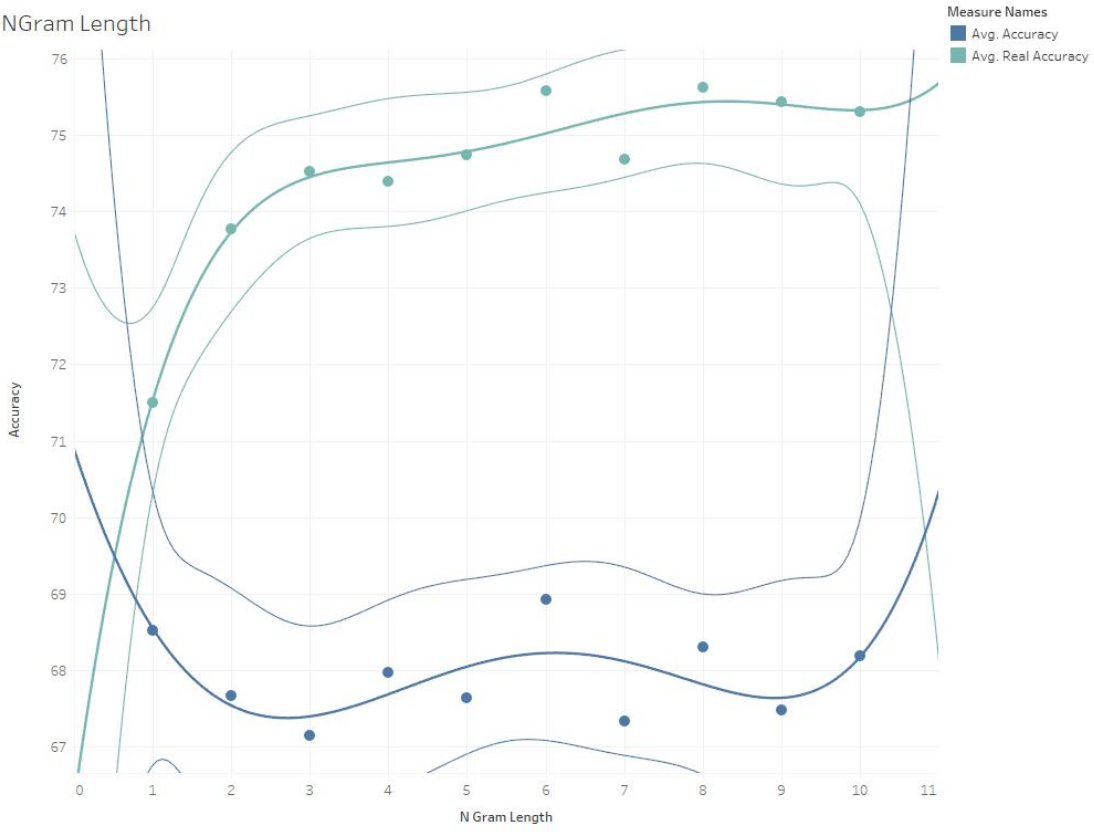

Figure 15. N-gram Length Accuracy

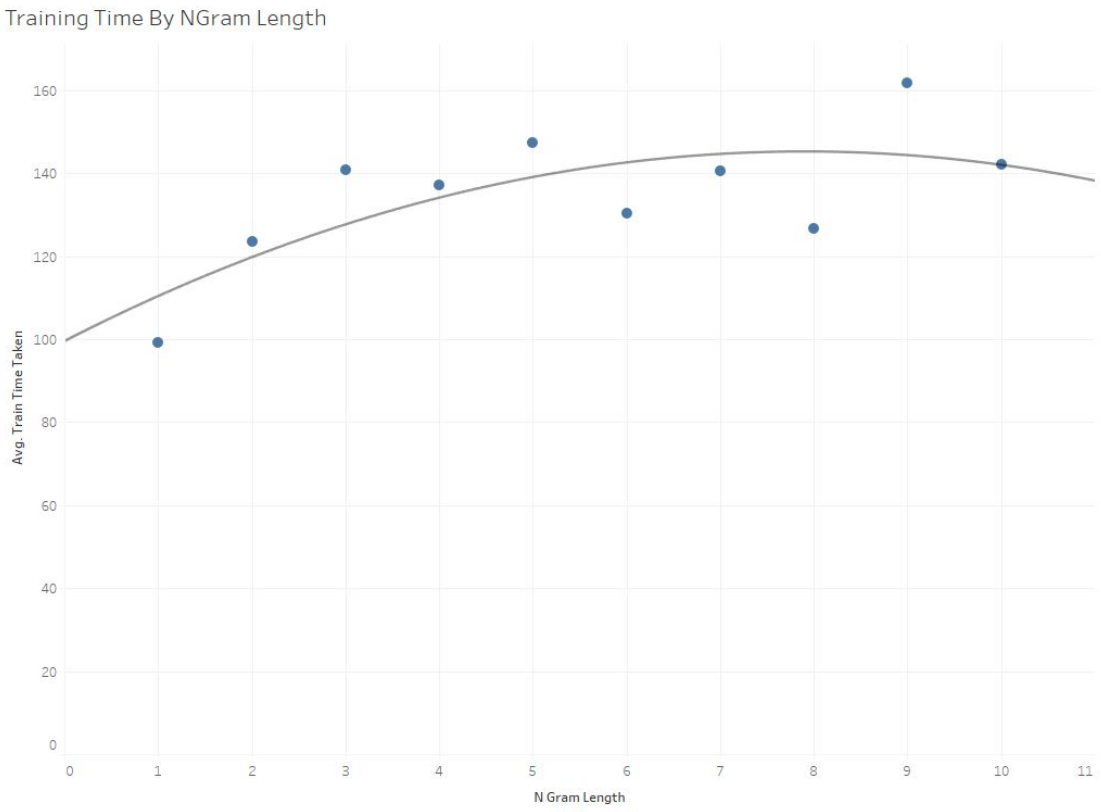

Figure 16. N-gram Length Training Time

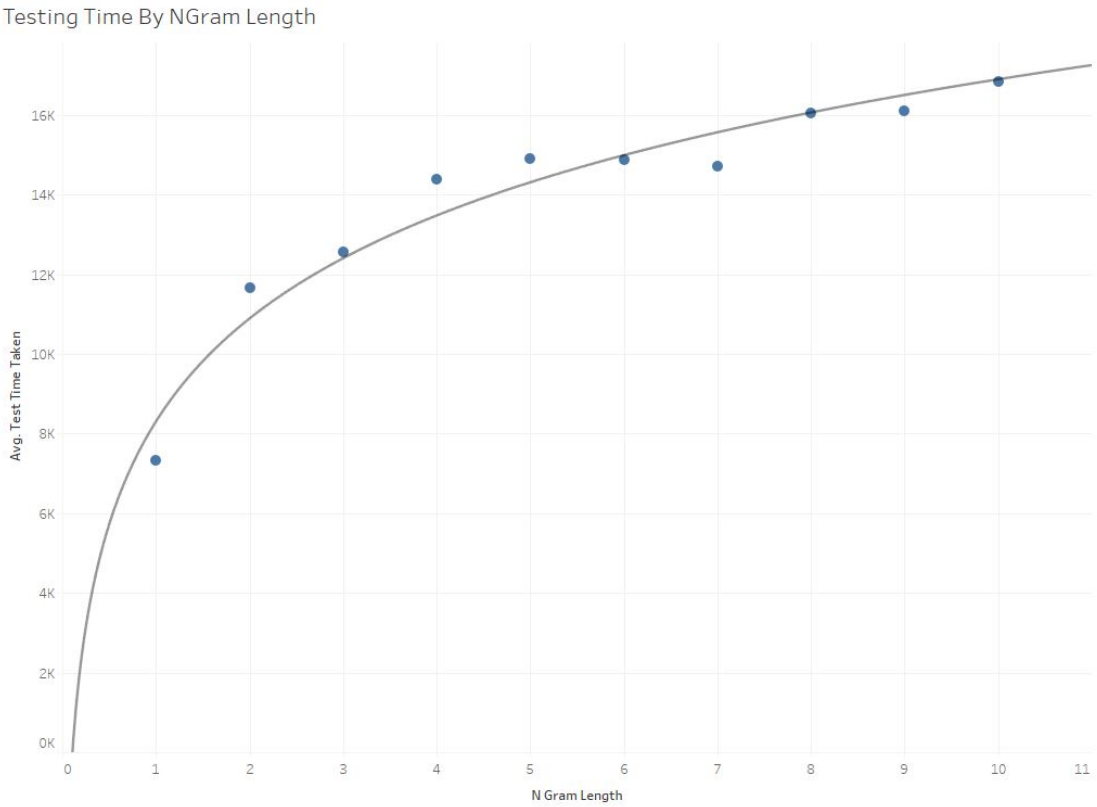

Figure 17. N-gram Length Testing Time

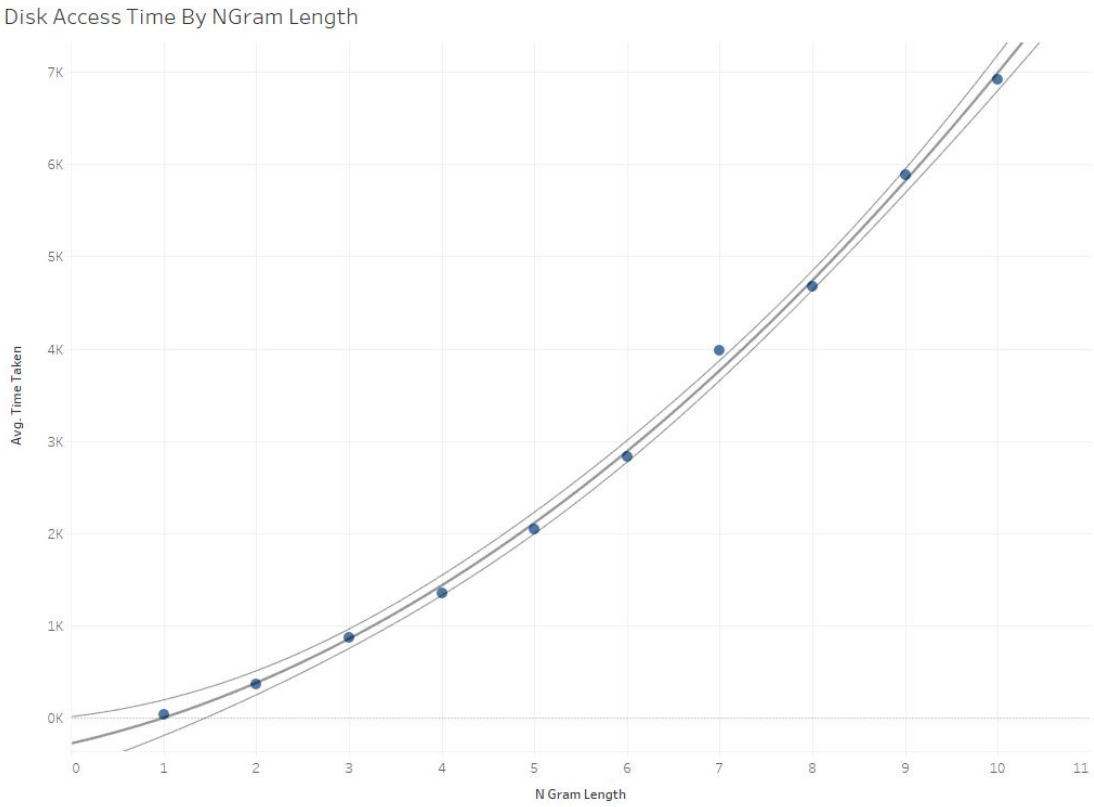

Figure 18. N-gram Length Disk Access Time

Percent	Accuracy	Std Dev	Train Time	Test Time	Positive	Negative
1	69.03	4.61	119	7412	67.85	75.38
2	68.07	1.8	99	8909	67.78	78.79
3	67.37	2.56	106	9345	69.53	78.87
4	70.30	3.26	121	10548	71.93	75.76
5	68.27	4.97	113	10559	71.53	77.65
6	67.40	2.89	118	12320	68.68	80.30
7	67.03	3.6	117	12998	68.38	81.44
8	68.23	2.05	125	12447	67.21	81.06
9	69.23	2.75	118	12863	70.47	79.17
10	70.23	3.53	162	13614	71.15	79.92

Table 5. N-gram Percent

Unlike with the number of comments, figures 16 and 17 show that the time required to train the model and run the accuracy test does not increase drastically with an increase of n-gram length. However, the time required to build the training and testing set does increase exponentially as seen in figure 18. For this reason the 6-gram was chosen as after that point there is some improvement, but not enough to justify the exponential increase in disk I/O time.

N-gram Percent

Now that the 6-gram has been chosen, it is time to figure out the optimum percent of those 6-grams to utilize. Continuing from the last test, this test used 300 comments as the training set, the ν_SVC linear kernel with a ν value of 0.7, a length up to 6-grams, and a knn level of 10%.

On this test the full 10 runs were completed and resulted in an average standard deviation of 3.20. This means that the minimum number of runs necessary to be sure of the results was 1.58.

Figure 19 shows that, like the n-gram length,the cross validated accuracy does not alter much regardless of the n-gram percent, while the upper and lower bounds

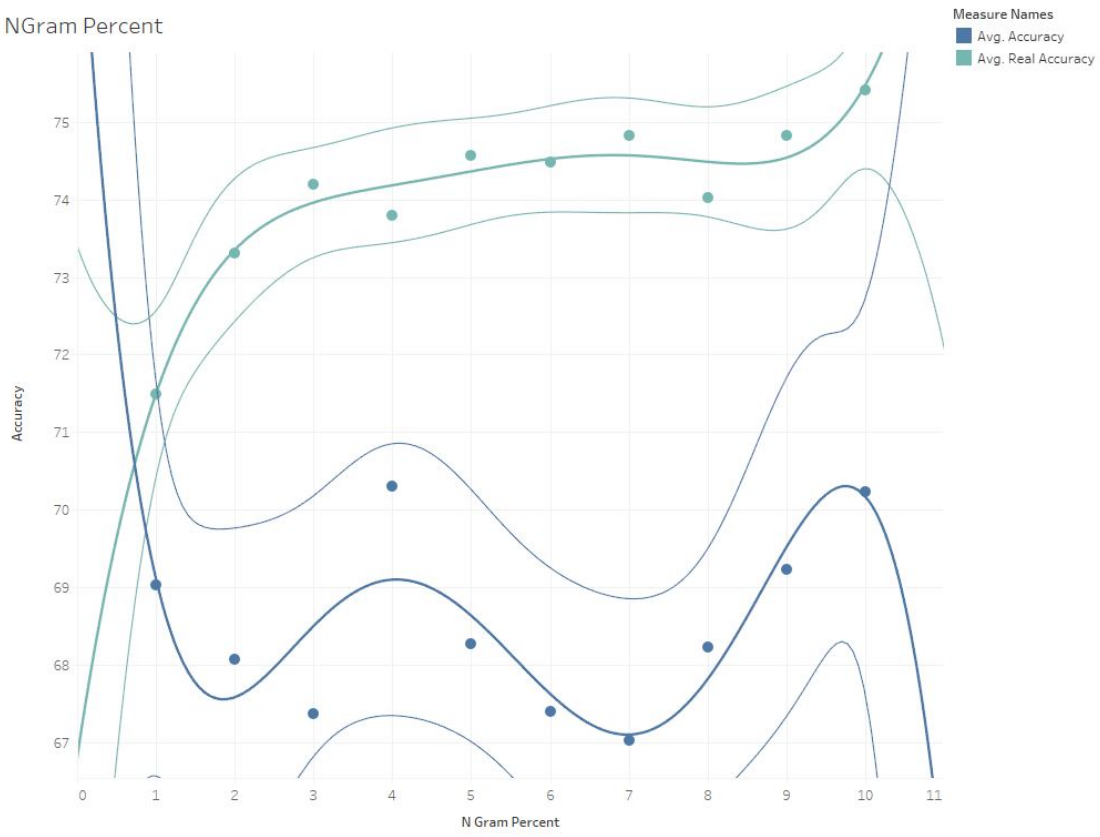

Figure 19. N-gram Percent Accuracy

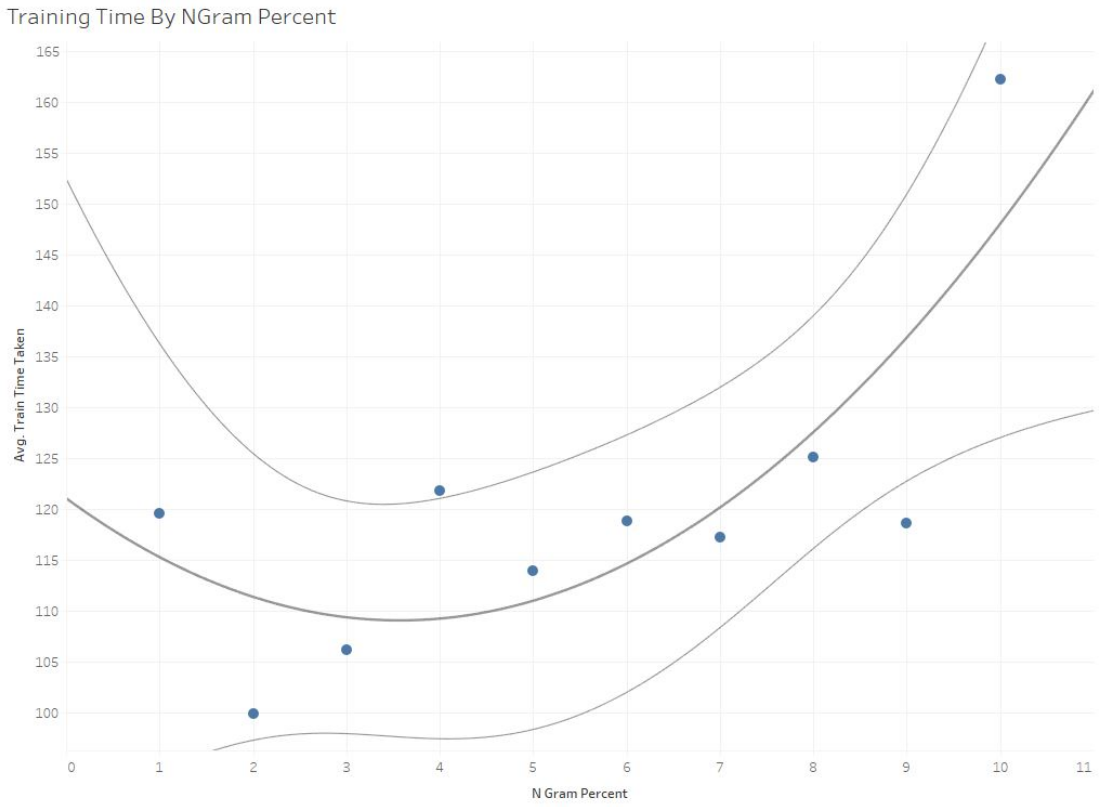

Figure 20. N-gram Percent Training Time

as well as the real accuracy are slightly sloped upwards.

As the N-grams are sorted based on the amount of information they bring to the machine learning, it wasn't surprising that increasing the percent of them that were taken had little effect on the overall result. Increasing the percent taken also increased the disk IO access time exponentially as seen in figure 22, so for that reason, 4% was taken as the optimum. Because of the exponential nature of the disk IO time, only up to 10% was tested as already the diminishing returns were not worth continuing and the trends did not point to it getting any better.

KNN Level

With all of the N-gram parameters locked down, the next step was to identify the best KNN level and if it should be used at all. This test used 300 comments as

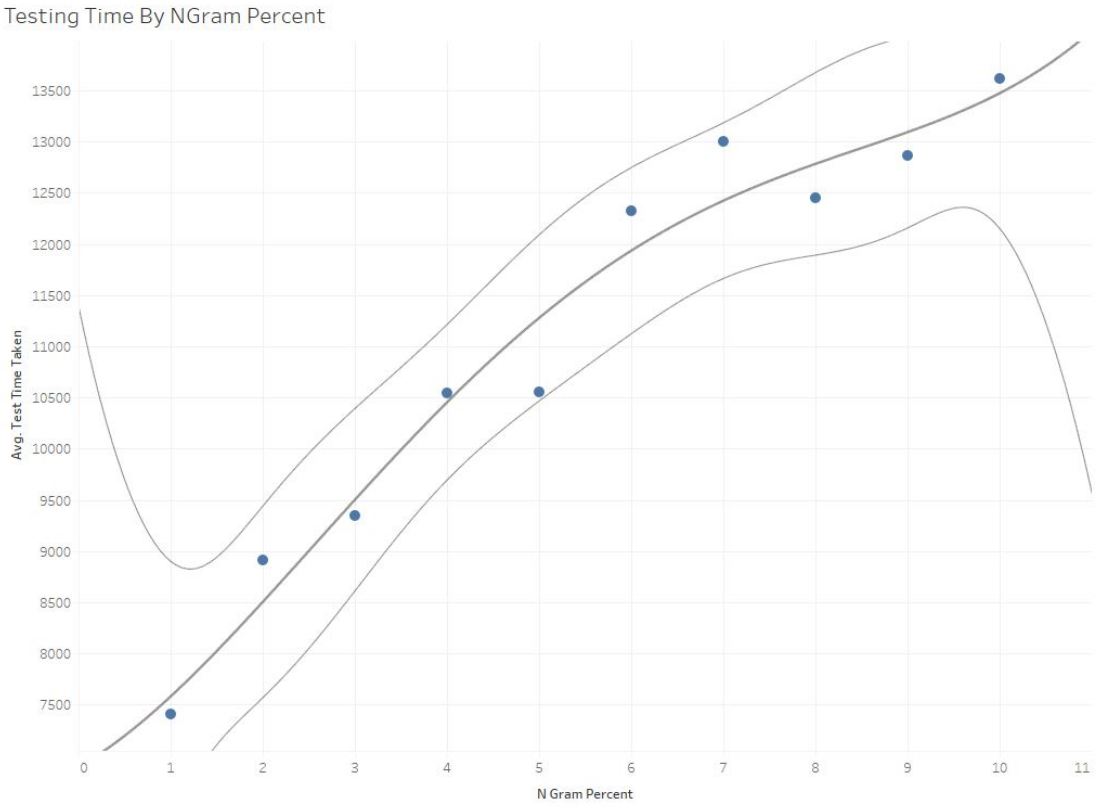

Figure 21. N-gram Percent Testing Time

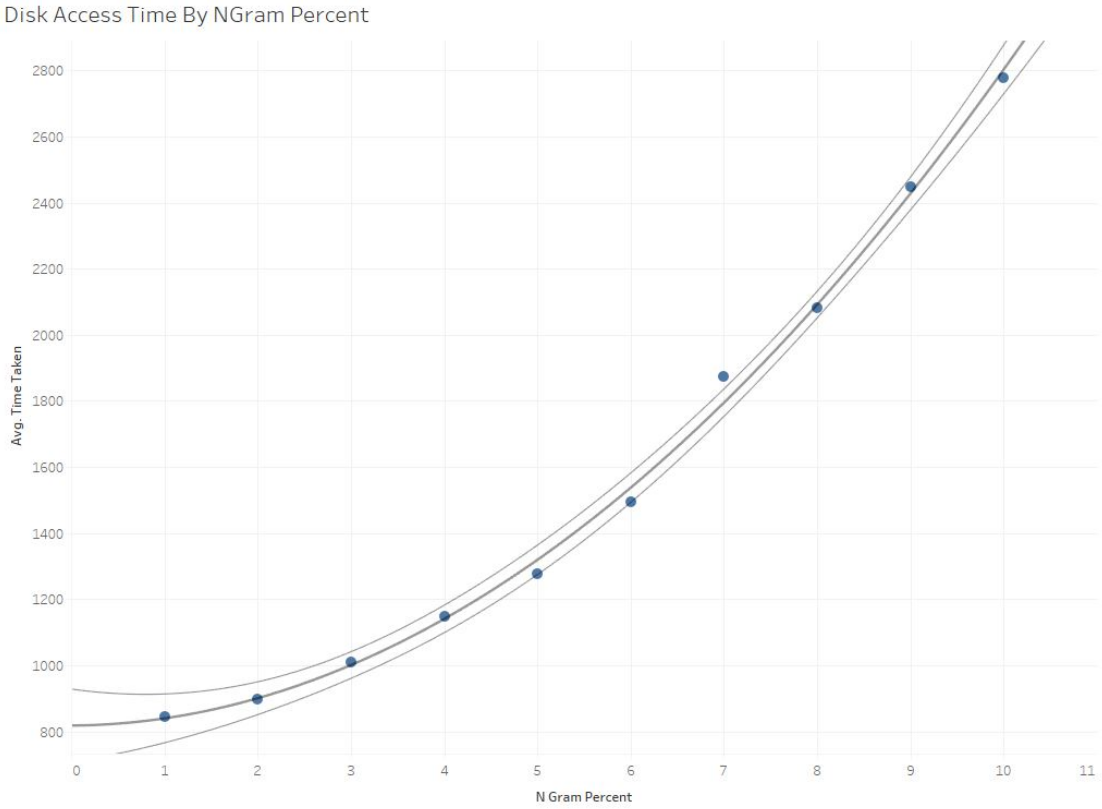

Figure 22. N-gram Percent Disk Access Time

knn Level	Accuracy	Std Dev	Train Time	Test Time	Positive	Negative
10	68.60	1.5	109	11790	71.18	77.27
20	68.60	1.5	119	11418	71.18	77.27
30	68.60	1.5	115	11499	71.18	77.27
40	68.60	1.5	122	11849	71.18	77.27
50	68.60	1.5	121	11587	71.18	77.27
60	68.60	1.5	109	11962	71.18	77.27
70	68.60	1.5	112	12022	71.18	77.27
80	68.60	1.5	131	11678	71.18	77.27
90	68.60	1.5	118	12290	71.18	77.27
100	68.60	1.5	118	12125	71.18	77.27

Table 6. KNN Level First Run

the training set, the ν_SVC linear kernel with a ν value of 0.7, and 4% of n-grams up to length 6. For this test run, in order to ensure that the results were due to the differences in the knn level and not in random training set changes, the training set was held constant.

On this test 5 runs were completed and resulted in an average standard deviation of 1.50. This means that the minimum number of runs necessary to be sure of the results was 0.35. This was stopped at 5 runs because it was noticed that in every one of the 5 runs there was no difference in the cross validated accuracy. Because of this, the test was rerun in 1 percent increments up to 10% instead of 10% increments.

On this test, the full 10 runs were completed and resulted in an average standard deviation of 5.54. This means that the minimum number of runs necessary to be sure of the results was 4.72.

Figure 23 shows that below the KNN Level of 3% there is a significant drop off of the cross validated accuracy, the real accuracy and the minimum accuracy.

With figures 24 and 25, it is clear that there is little effect on the training or testing time so there is no reason not to choose the level at which the accuracies

knn Level	Accuracy	Std Dev	Train Time	Test Time	Positive	Negative
1	53.62	19.31	118	4118	48.50	68.65
2	62.75	3.38	93	10762	69.21	74.62
3	65.91	2.61	117	11165	71.53	77.65
4	66.85	3.65	117	11662	70.68	78.41
5	67.04	3.55	106	11826	70.54	78.41
6	66.95	3.57	124	11801	70.73	78.03
7	66.83	3.38	112	11784	70.75	78.03
8	66.83	3.38	118	11068	70.75	78.03
9	66.83	3.38	112	11825	70.75	78.03
10	66.83	3.38	111	12085	70.75	78.03

Table 7. KNN Level Second Run

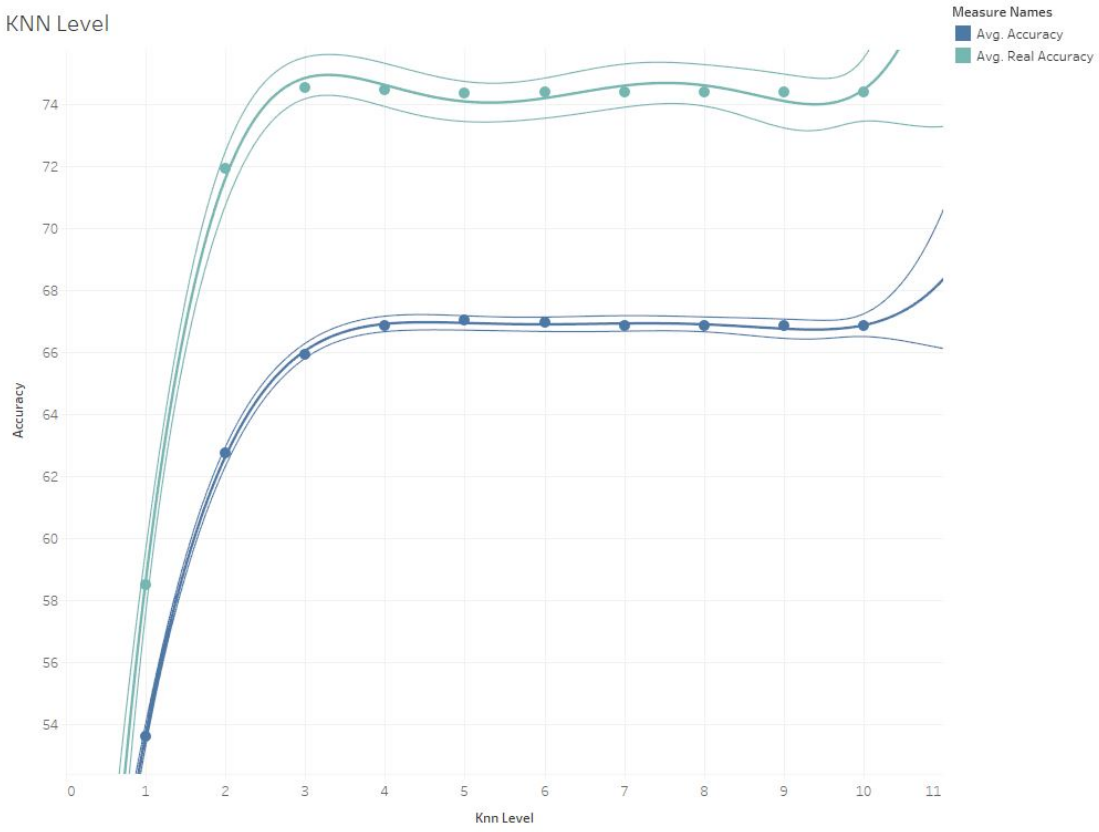

Figure 23. KNN Level Accuracy

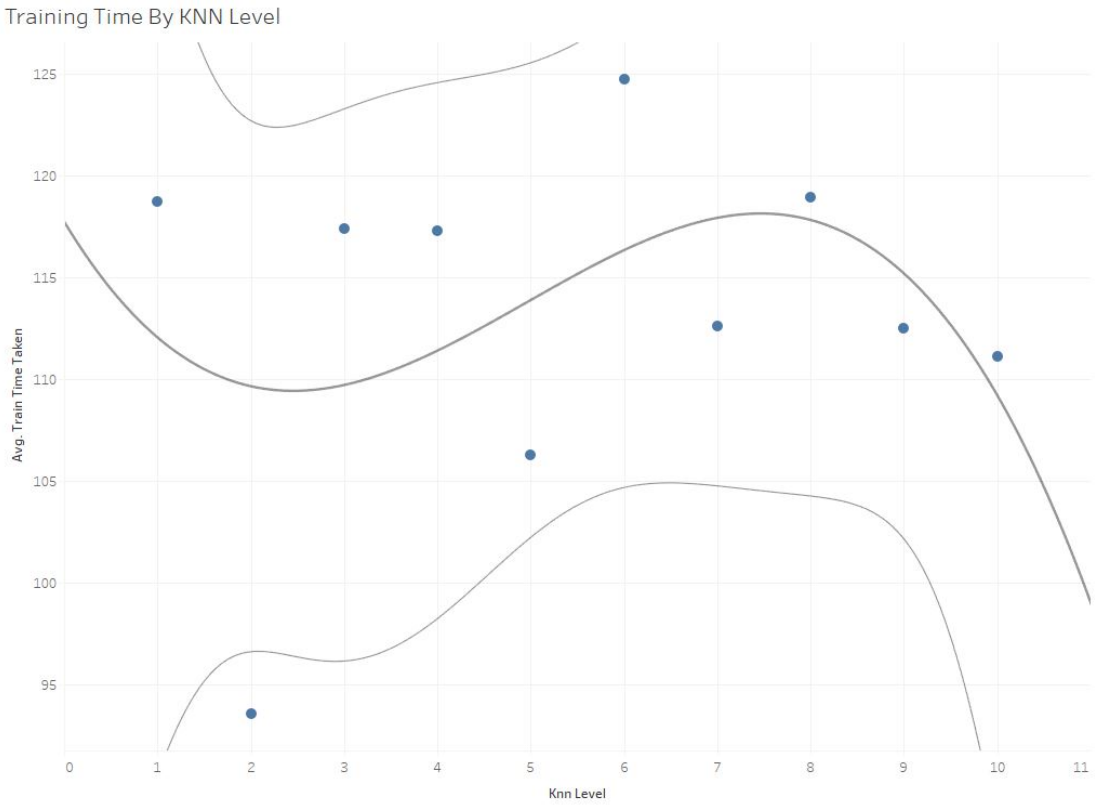

Figure 24. KNN Level Training Time

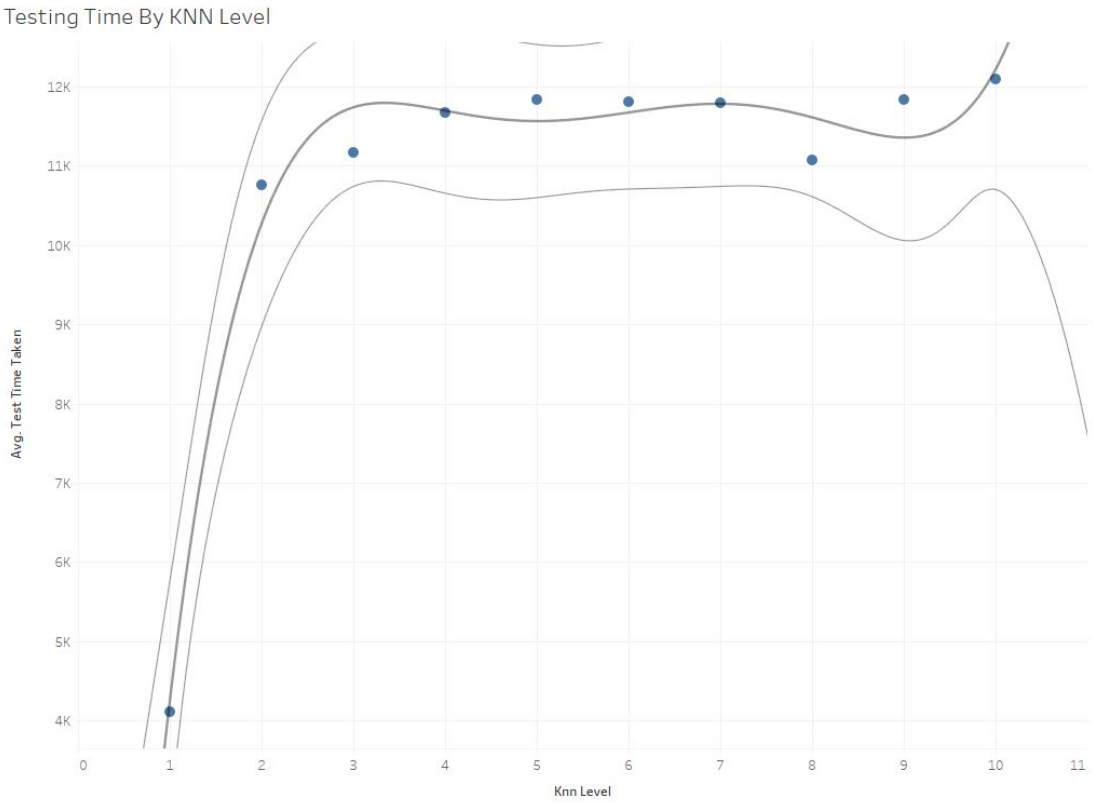

Figure 25. KNN Level Testing Time

C	Accuracy	Std Dev	Train Time	Test Time	Positive	Negative
-5	54.12	2.3	682	124771	73.32	37.12
-4	60.14	3.58	661	161911	81.90	49.81
-3	62.92	4.69	1249	40333	73.51	63.40
-2	64.14	2.42	1625	28453	69.37	72.08
-1	65.48	2.97	958	72818	67.96	78.49
0	68.04	1.79	6172	77599	68.44	81.89
1	67.82	2.96	7807	272860	69.74	82.58
2	67.15	2.21	20426	199458	69.47	83.77
3	66.26	2.61	19286	144323	68.79	86.36
4	65.26	4	9864	100047	68.36	86.36
5	65.70	3.11	14495	64984	68.62	86.74
6	65.37	3.52	8301	95765	68.90	86.74
7	65.37	3.52	15370	67437	68.90	86.74

Table 8. Linear C_SVC kernel Grid Search

level out at the maximum which, in this case, is 5%.

Grid Search

The grid search test is designed to optimize all of the parameters across the three possible kernels that were used and to test all of the different possible parameter combinations used by each of them. For this test a 300 comment training set, 4% of the n-grams up to length 6 and a KNN level of 5% were used.

On this test, 3 runs were completed and resulted in an average standard deviation of 3.78. This means that the minimum number of runs necessary to be sure of the results was 2.20. On this test the minimum cutoff was used because each of these runs took an average of 2 weeks from start to finish. Due to the large amount of data found in the polynomial kernel run, the Coef0 parameter was split into two charts, and the high degree and low C values were excluded from the charts since they were low performing, as is shown in the graphs.

The first set of graphs in figures 26 and 27 shows the performance of the linear

ν	Accuracy	Std Dev	Train Time	Test Time	Positive	Negative
0.1	65.59	3.24	9197	128270	68.68	86.74
0.2	65.15	3.62	9323	105364	68.39	86.74
0.3	66.15	3.25	13140	76615	68.89	85.98
0.4	66.82	1.89	11656	53140	69.42	84.85
0.5	68.15	3.54	8005	126583	69.72	83.02
0.6	67.60	3.15	14682	30062	69.26	83.02
0.7	67.82	2.12	11395	175307	68.02	81.89
0.8	66.26	2.88	9776	119380	67.26	79.17
0.9	65.48	4.12	16671	41781	70.43	69.32

Table 9. Linear ν_SVC kernel Grid Search

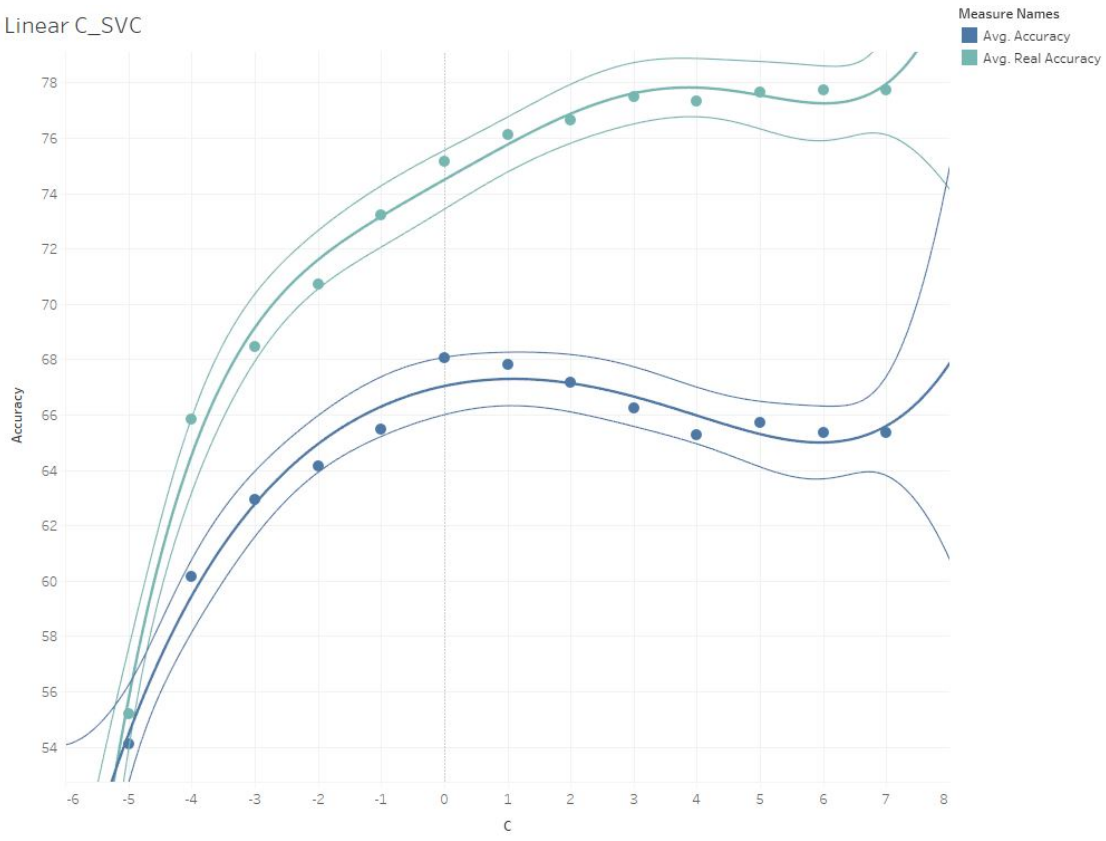

Figure 26. Grid Search Linear C Accuracy

C	Degree	Accuracy	Std Dev	Train Time	Test Time	Positive	Negative
-2	1	65.59	3.18	5130	166140	68.33	75.38
-2	2	66.82	3.85	8625	53536	69.29	84.47
-2	3	63.14	3.3	14750	144188	69.07	84.91
-2	4	62.70	3.34	8374	42114	67.02	84.91
-1	1	66.93	2.3	5645	106484	68.34	79.17
-1	2	66.26	5.38	8286	214552	69.46	84.91
-1	3	63.03	5.14	10015	115427	68.41	85.28
-1	4	63.37	4.21	10276	70442	65.42	85.23
0	1	68.49	3.38	13234	75323	68.82	83.33
0	2	65.15	5.88	9088	80468	69.22	85.98
0	3	63.25	4.06	7948	109318	67.26	85.98
0	4	63.03	4.54	8380	268704	65.21	85.23
1	1	68.49	3.27	10026	119113	69.95	82.95
1	2	64.37	6.55	15520	74979	69.00	86.36
1	3	63.25	4.06	12751	41984	67.00	85.98
1	4	63.03	4.54	10578	86932	65.21	85.23
2	1	67.60	2.4	9874	55937	69.26	84.85
2	2	64.26	6.78	6203	43374	68.68	86.74
2	3	63.25	4.06	8885	121172	67.00	85.98
2	4	63.03	4.54	10453	94021	65.21	85.23
3	1	65.59	2.87	11453	154662	68.68	86.74
3	2	64.15	5.95	11051	63724	68.63	86.36
3	3	63.25	4.06	11885	101174	67.00	85.98
3	4	63.03	4.54	7781	55254	65.21	85.23
4	1	65.59	3.49	10006	114797	68.30	86.74
4	2	64.15	5.95	10552	52116	68.63	86.36
4	3	63.25	4.06	13845	122033	67.00	85.98
4	4	63.03	4.54	15351	96700	65.21	85.23
5	1	65.59	3.24	11398	147241	68.89	86.74
5	2	64.15	5.95	7150	90288	68.63	86.36
5	3	63.25	4.06	11817	114653	67.00	85.98
5	4	63.03	4.54	11016	134195	65.21	85.23
6	1	65.37	3.52	9160	108761	68.89	86.74
6	2	64.15	5.95	11482	46409	68.63	86.36
6	3	63.25	4.06	7711	105790	67.00	85.98
6	4	63.03	4.54	13142	125807	65.21	85.23
7	1	65.37	3.52	13585	83650	68.89	86.74
7	2	64.15	5.95	10616	48533	68.63	86.36
7	3	63.25	4.06	7725	106929	67.00	85.98
7	4	63.03	4.54	6238	122289	65.21	85.23

Table 10. Polynomial C_SVC kernel with Coef0 = 0 Grid Search

C	Degree	Accuracy	Std Dev	Train Time	Test Time	Positive	Negative
-2	1	65.59	3.18	7812	150500	68.33	75.38
-2	2	68.49	4.41	11500	96359	69.43	84.53
-2	3	63.26	6.09	8391	74750	68.34	86.36
-2	4	63.81	4.36	8890	89026	67.00	85.98
-1	1	66.93	2.3	8765	230000	68.34	79.17
-1	2	65.48	5.94	11567	149078	68.75	86.36
-1	3	64.04	4.51	7411	79041	67.80	86.36
-1	4	63.81	4.36	6604	232349	67.00	85.98
0	1	68.49	3.38	17052	60203	68.82	83.33
0	2	64.70	6.24	12130	208531	68.75	85.66
0	3	64.15	4.69	8474	83052	67.69	86.04
0	4	63.81	4.36	8547	296943	67.00	85.98
1	1	68.49	3.27	10838	56354	69.94	82.95
1	2	64.48	5.87	9187	92704	68.71	86.74
1	3	64.15	4.69	11218	81089	67.69	86.04
1	4	63.81	4.36	9557	105396	67.00	85.98
2	1	67.60	2.4	13432	100385	69.26	84.85
2	2	64.70	6.65	6036	118187	68.68	86.74
2	3	64.15	4.69	8317	130771	67.69	86.04
2	4	63.81	4.36	10958	74130	67.00	85.98
3	1	65.59	2.87	12869	132708	68.68	86.74
3	2	64.70	6.65	9104	132714	68.68	86.74
3	3	64.15	4.69	10166	34115	67.69	86.04
3	4	63.81	4.36	12296	142372	67.00	85.98
4	1	65.59	3.49	11792	62099	68.30	86.74
4	2	64.70	6.65	7574	209446	68.68	86.74
4	3	64.15	4.69	10771	110698	67.69	86.04
4	4	63.81	4.36	4571	116820	67.00	85.98
5	1	65.59	3.24	18464	105519	68.89	86.74
5	2	64.70	6.65	11242	77823	68.68	86.74
5	3	64.15	4.69	11715	41791	67.69	86.04
5	4	63.81	4.36	10828	97734	67.00	85.98
6	1	65.37	3.52	10031	58649	68.90	86.74
6	2	64.70	6.65	8627	169548	68.68	86.74
6	3	64.15	4.69	10361	165843	67.69	86.04
6	4	63.81	4.36	9164	76402	67.00	85.98
7	1	65.37	3.52	18108	63711	68.90	86.74
7	2	64.70	6.65	8849	41016	68.68	86.74
7	3	64.15	4.69	7670	113872	67.69	86.04
7	4	63.81	4.36	7284	170595	67.00	85.98

Table 11. Polynomial C_SVC kernel with Coef0 = 1 Grid Search

ν	Degree	Accuracy	Std Dev	Train Time	Test Time	Positive	Negative
0.1	1	65.59	3.24	10736	33531	68.66	86.79
0.1	2	63.81	5.95	6188	83236	68.58	86.74
0.1	3	63.14	3.87	8588	77449	67.18	85.98
0.1	4	62.81	4.53	7983	195969	65.72	85.23
0.2	1	65.15	3.62	3568	89962	68.41	86.74
0.2	2	64.48	6.73	10475	46573	68.73	86.36
0.2	3	63.26	5.48	7350	67612	68.44	85.28
0.2	4	63.14	3.26	7391	85403	68.22	84.09
0.3	1	66.15	3.25	7559	46223	68.86	85.98
0.3	2	64.92	6.11	6941	109917	69.31	85.98
0.3	3	63.37	3.92	12422	44825	69.13	84.85
0.3	4	62.92	4.03	9543	77463	69.29	82.95
0.4	1	66.82	1.89	7896	105816	69.42	84.85
0.4	2	65.93	5.36	5977	56015	69.18	85.23
0.4	3	64.92	4.04	11606	75379	69.82	84.47
0.4	4	63.70	4.23	11234	213335	69.42	83.33
0.5	1	68.15	3.54	13743	106142	69.72	83.02
0.5	2	66.82	4.2	8451	114073	69.31	84.91
0.5	3	65.26	3.74	8489	157674	68.63	84.09
0.5	4	64.81	3.09	9124	70887	67.00	84.09
0.6	1	67.71	3.35	13971	97572	69.26	83.02
0.6	2	66.37	2.78	14040	114336	68.74	84.91
0.6	3	65.70	3.2	11170	173692	67.53	84.09
0.6	4	65.37	3.2	9179	204422	65.90	85.23
0.7	1	67.82	2.12	9786	78855	68.04	81.89
0.7	2	67.37	3.35	8725	147029	67.43	83.71
0.7	3	66.37	1.99	12688	41319	66.20	83.71
0.7	4	64.15	2.53	9038	108947	61.91	84.53
0.8	1	66.26	2.88	7984	137423	67.26	79.17
0.8	2	66.15	2.83	12442	87678	65.85	81.51
0.8	3	66.04	1.62	6791	89167	62.84	83.33
0.8	4	63.25	2.28	17072	140198	56.08	85.61
0.9	1	65.48	4.12	10319	65749	70.43	69.32
0.9	2	65.37	2.87	6082	36650	66.12	76.52
0.9	3	60.91	0.84	16461	84982	53.80	82.64
0.9	4	55.79	3.29	8540	152171	36.03	87.92

Table 12. Polynomial ν_SVC kernel with Coef0 = 0 Grid Search

ν	Degree	Accuracy	Std Dev	Train Time	Test Time	Positive	Negative
0.1	1	65.70	3.11	10213	37147	68.68	86.74
0.1	2	64.04	6.08	7081	57495	68.66	86.74
0.1	3	64.04	4.51	9974	94277	67.74	86.36
0.1	4	63.59	4.45	6696	105028	67.11	85.98
0.2	1	65.26	3.57	7482	75545	68.39	86.74
0.2	2	64.59	6.63	10545	83241	68.73	86.04
0.2	3	63.59	6.36	6359	90083	68.14	85.98
0.2	4	63.70	5.45	8518	152689	68.10	85.23
0.3	1	66.15	3.25	7805	293012	68.86	85.98
0.3	2	64.92	4.91	8961	217802	68.81	86.04
0.3	3	64.92	4.35	13887	212513	68.52	85.98
0.3	4	63.48	4.71	8757	119766	68.89	85.23
0.4	1	66.82	1.89	9202	155417	69.42	84.85
0.4	2	65.93	5.91	13921	56884	68.90	86.36
0.4	3	65.48	5.42	10852	57003	68.95	85.61
0.4	4	65.37	5.69	9837	144426	68.75	84.15
0.5	1	68.15	3.54	6315	59562	69.72	83.02
0.5	2	68.04	4.51	6562	194382	69.26	84.85
0.5	3	66.26	4.81	10441	247020	68.65	85.23
0.5	4	65.04	4.21	15117	89461	68.06	84.15
0.6	1	67.71	3.35	15702	235797	69.26	83.02
0.6	2	67.71	2.77	5119	64056	68.46	85.23
0.6	3	66.37	3.06	8550	28214	67.83	85.23
0.6	4	66.04	3.2	10791	261295	67.21	84.53
0.7	1	67.82	2.12	9410	104394	68.04	81.89
0.7	2	67.71	3.65	8120	136188	67.21	83.71
0.7	3	67.71	3.07	10563	76768	66.74	83.71
0.7	4	65.82	3.06	9893	42997	65.72	83.71
0.8	1	66.26	2.88	11246	218868	67.26	79.17
0.8	2	66.59	3.45	8381	175003	66.18	81.51
0.8	3	66.48	2.78	16231	158954	64.78	81.51
0.8	4	66.26	1.7	6026	268180	62.36	82.95
0.9	1	65.48	4.12	13525	148336	70.43	69.32
0.9	2	65.04	3.12	12338	164143	68.33	74.62
0.9	3	63.92	1.66	15240	48057	61.90	79.92
0.9	4	60.47	1.04	9309	69141	50.61	83.71

Table 13. Polynomial ν_SVC kernel with Coef0 = 1 Grid Search

C	Accuracy	Std Dev	Train Time	Test Time	Positive	Negative
-5	52.78	2.1	5890	173386	90.83	23.48
-4	52.78	2.1	6255	151224	90.83	23.48
-3	52.67	2.13	5791	124198	90.48	25.00
-2	53.90	2.68	12468	108177	83.18	42.80
-1	58.02	1.82	14875	35218	62.01	75.38
0	64.03	1.47	14490	73719	66.60	83.40
1	64.25	1.43	12276	93213	67.93	83.77
2	64.03	2.11	11333	146859	69.03	84.09
3	63.14	1.89	10583	174141	69.08	83.71
4	63.25	2.02	12041	103505	69.08	83.71
5	63.25	2.02	12859	52942	69.08	83.71
6	63.25	2.02	7214	62161	69.08	83.71
7	63.25	2.02	7182	92000	69.08	83.71

Table 14. RBF C_SVC kernel Grid Search

ν	Accuracy	Std Dev	Train Time	Test Time	Positive	Negative
0.1	63.25	2.02	15333	76312	69.10	84.09
0.2	63.14	1.86	7489	106156	69.08	83.71
0.3	64.15	2.3	18406	67854	68.91	84.09
0.4	64.03	1.66	11901	88641	68.38	84.09
0.5	63.92	2	11229	60094	67.82	84.09
0.6	64.03	1.34	11494	120719	67.85	84.09
0.7	63.36	1.62	8953	31630	67.56	83.77
0.8	63.92	2.34	4729	33578	67.05	83.71
0.9	62.92	1.72	13375	11047	66.89	81.89

Table 15. RBF ν_SVC kernel Grid Search

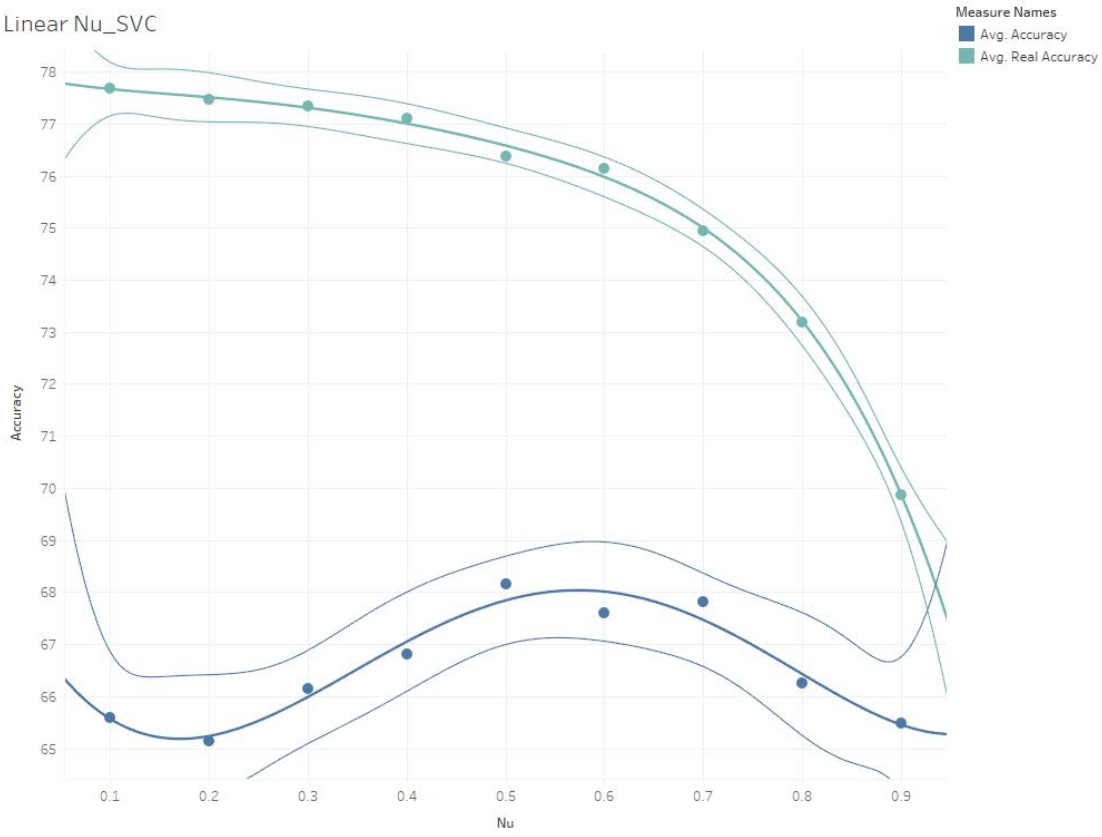

Figure 27. Grid Search Linear ν Accuracy

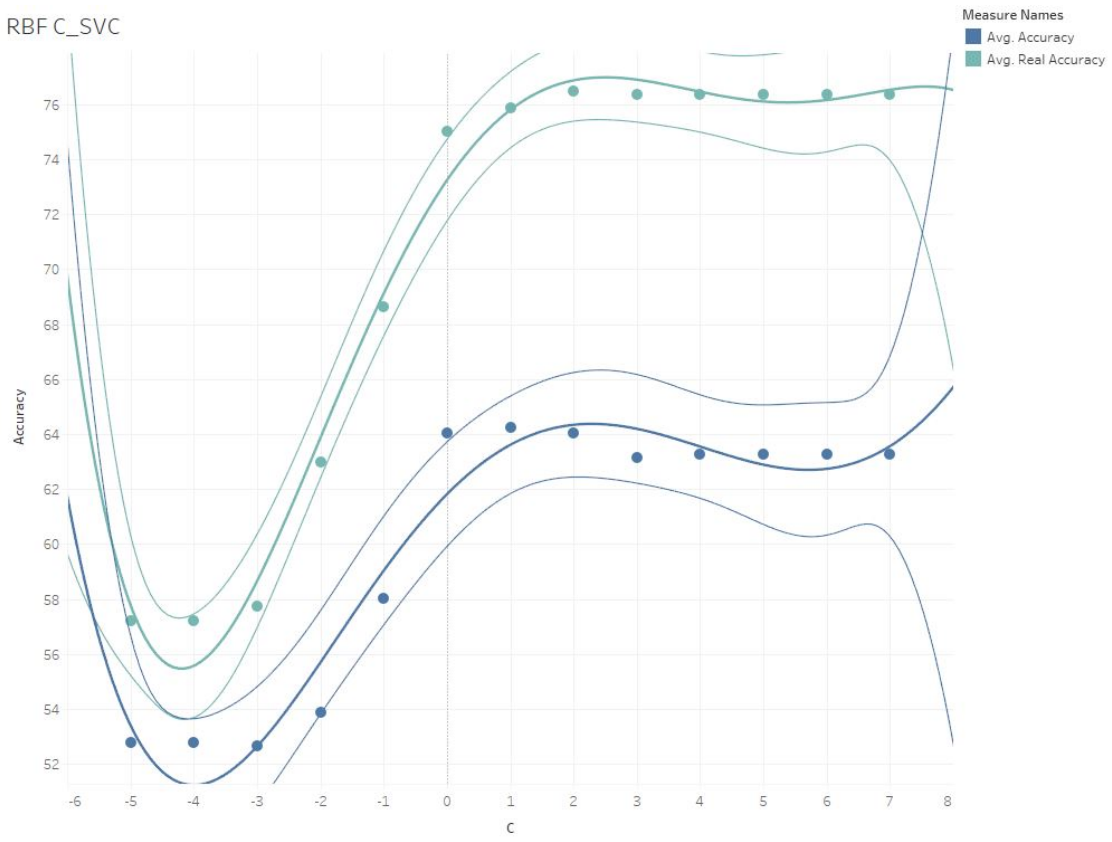

Figure 28. Grid Search Radial Basis Function C Accuracy

kernel and its two parameters. On the C graph, the C values below 0 are very poor performers, while above 0, the slope of each of the accuracies is almost flat. While the cross validated accuracy is highest at a C value of 0, the real accuracy is highest at a C value of 6 and 7. Based on all of this a C of 6 looks to be the best compromise. This has an accuracy of 65.37 and 77.72.

Moving onto ν, it is almost the opposite of the C graph, with the values starting highest at ν of 0.1 and dropping sharply after ν of 0.7. The cross validated accuracy again does not match the pattern of the rest of the data and increases slightly from 0.1 to 0.5. It too begins dropping after 0.7. Overall a ν value of 0.3 appears to be the best. With its accuracy of 66.15 and 77.33, it is almost completely tied with the C value.

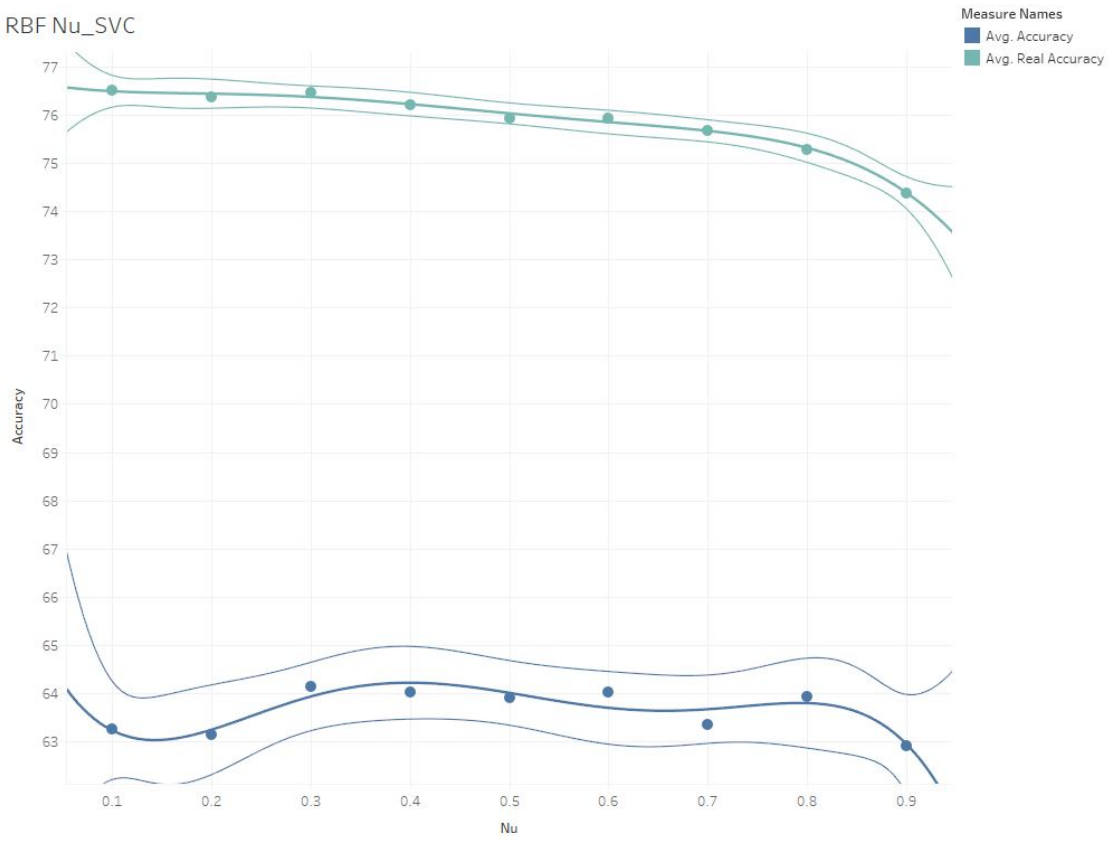

Figure 29. Grid Search Radial Basis Function ν Accuracy

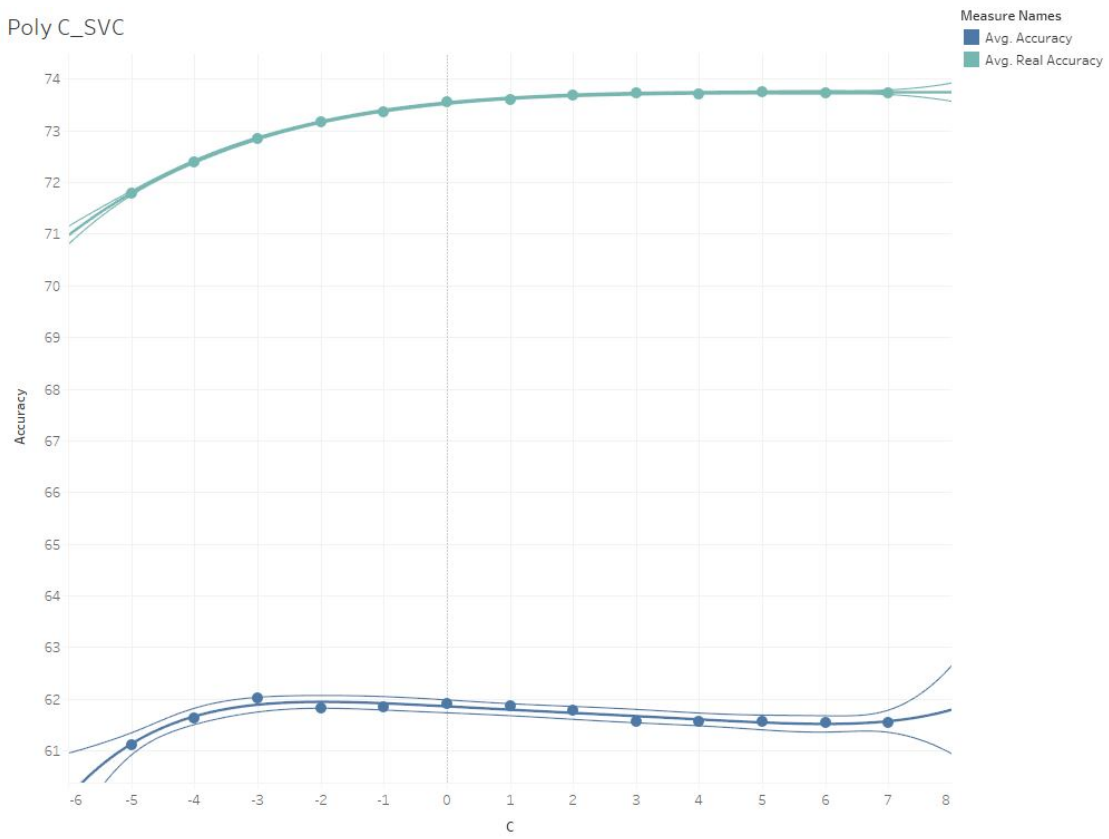

Figure 30. Grid Search Polynomial C Accuracy

The radial basis function kernel ended up performing in an almost identical manner to the linear kernel as seen in figures 28 and 29. In fact, it is so similar that again it appears that a C of 6 is the best with an accuracy of 63.25 and 76.37. The ν value of 0.3 is also again a good performer, although on figure 29 it can be noticed that the slope is essentially flat on all of the accuracies until after 0.6. Using 0.3 again we get an accuracy of 64.15 and 76.47. Thus, in every way the linear kernel outperformed the RBF kernel.

The polynomial kernel is the most difficult to study because unlike the linear and radial basis function kernels, there are more parameters than just the C or ν values. If you just look at the C and ν values, then figures 30 and 31 show that the polynomial kernel still follows the same pattern as the linear and radial basis

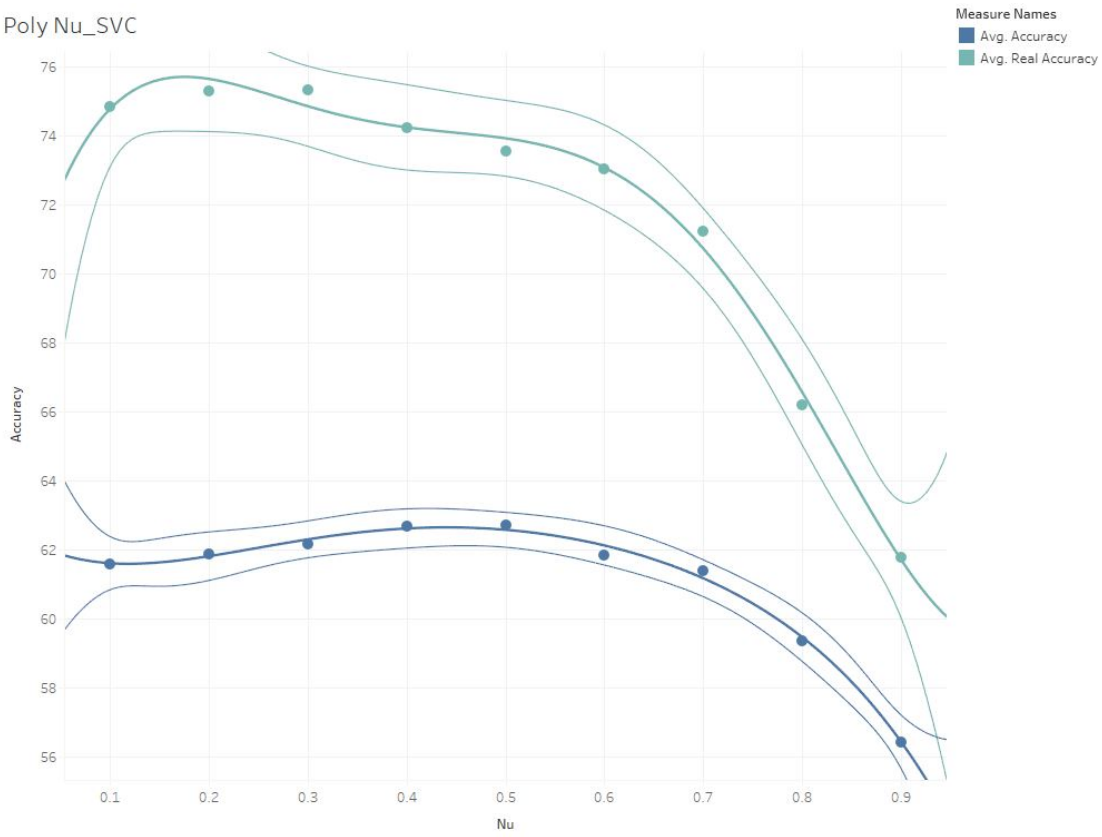

Figure 31. Grid Search Polynomial ν Accuracy

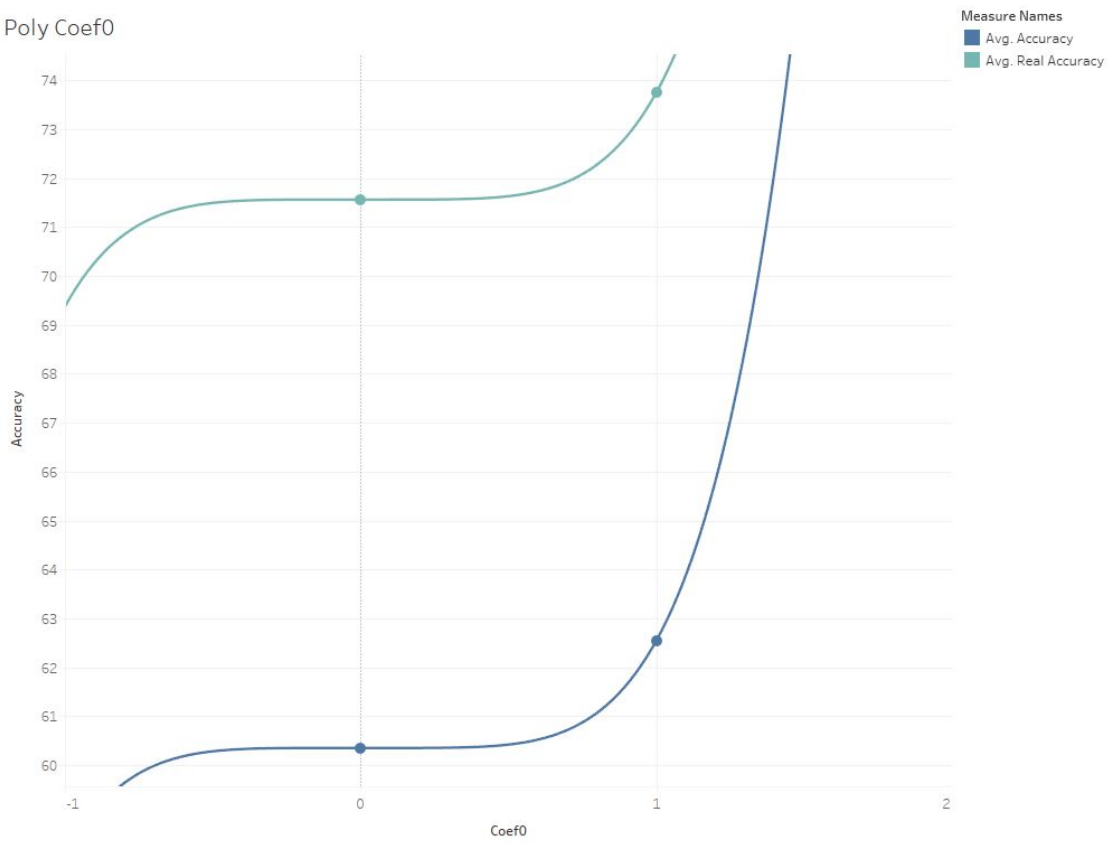

Figure 32. Grid Search Polynomial Coef0 Accuracy

function. In an effort to simplify the data and to give us a better idea of what is going on, first the parameter Coef0 was analyzed to see which of the two states was better.

As figure 32 shows, a Coef0 of 1 outperforms a Coef0 of 0. In fact, it goes even further than this. If you check the charts like tables 10 and 11, comparing any of the rows, it can be seen that the Coef0 = 1 chart always outperforms the Coef0 = 0 chart.

After isolating the Coef0 value, the next parameter to choose is the Degree, although this one is not as clear cut. Figure 33 shows the the cross validated accuracy starts highest at a degree of 1 and slopes downwards after that, while the other three accuracies trend up to a degree of 3 and then slope downwards.

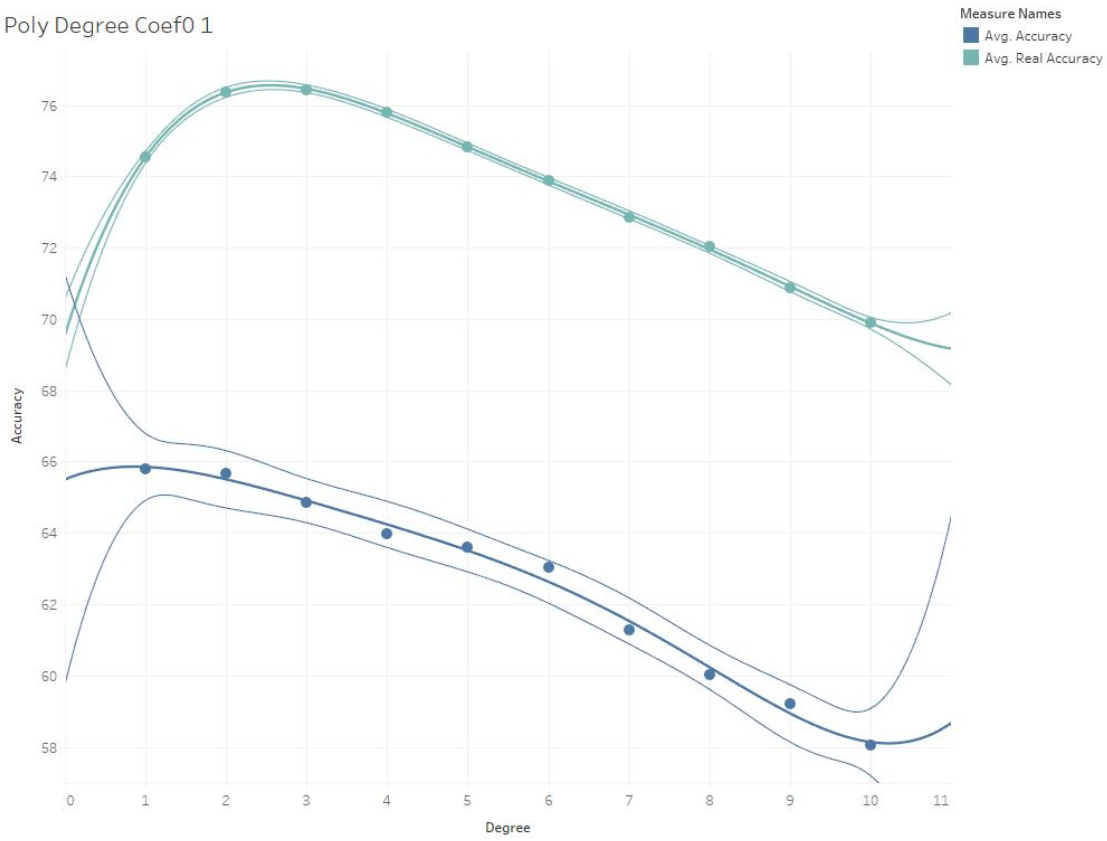

Figure 33. Grid Search Polynomial Degree Accuracy

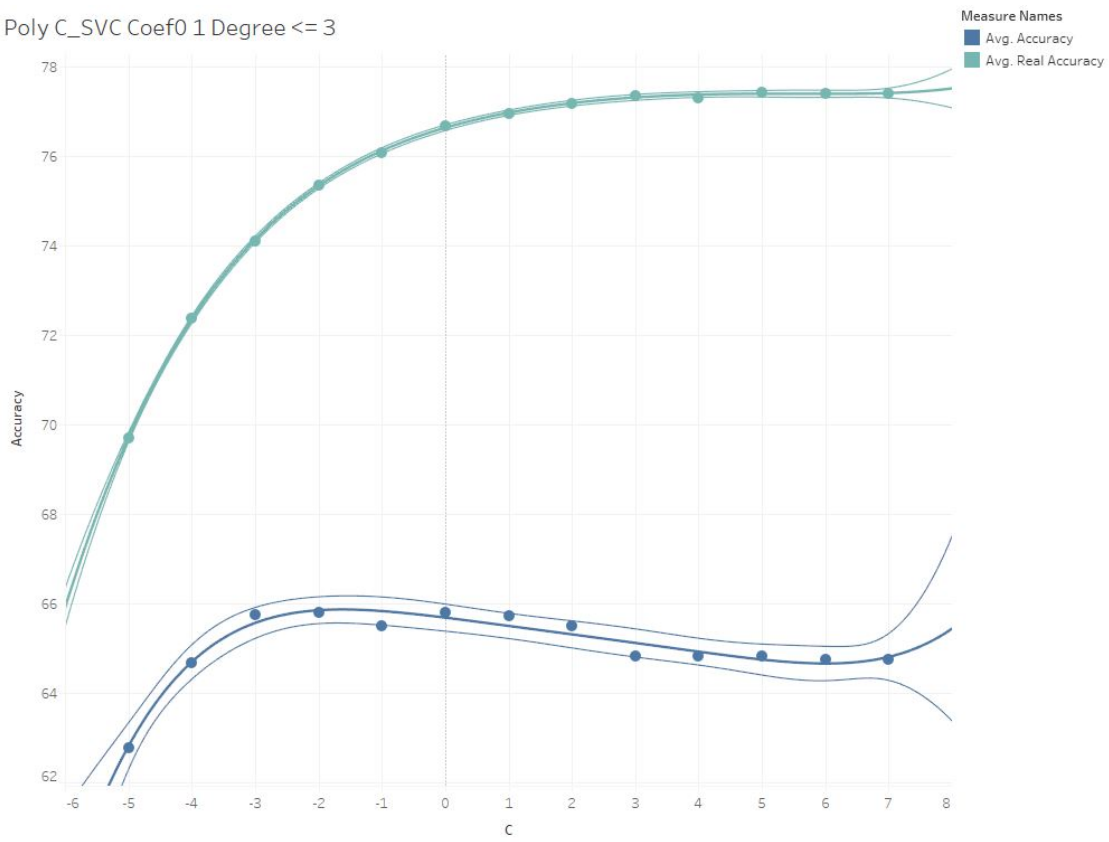

Figure 34. Grid Search Polynomial C Degree 3 Accuracy

Because of this, the next step is to try to narrow down the C and ν values with a degree of less than or equal to 3. While the degree of 1 is poorly performing in this graph, it should be noted that this is due to some poor performance on the C and ν choices that will become apparent later.

If you compare figures 34 and 35 to figures 30 and 31, all of the accuracies have improved so this is on the right track. Now, within the C graph, it again is obvious that a C of less than 0 is poor performing. In the future tests, we will remove $C < 0$ as it was consistently poor across all kernels and all other parameters. In this case a C of 2 was chosen as the best as the cross validated accuracy sloped downwards while the other accuracies were fairly flat after that point.

Looking at the ν graph, again it is obvious that after a ν of 0.7 the accuracies

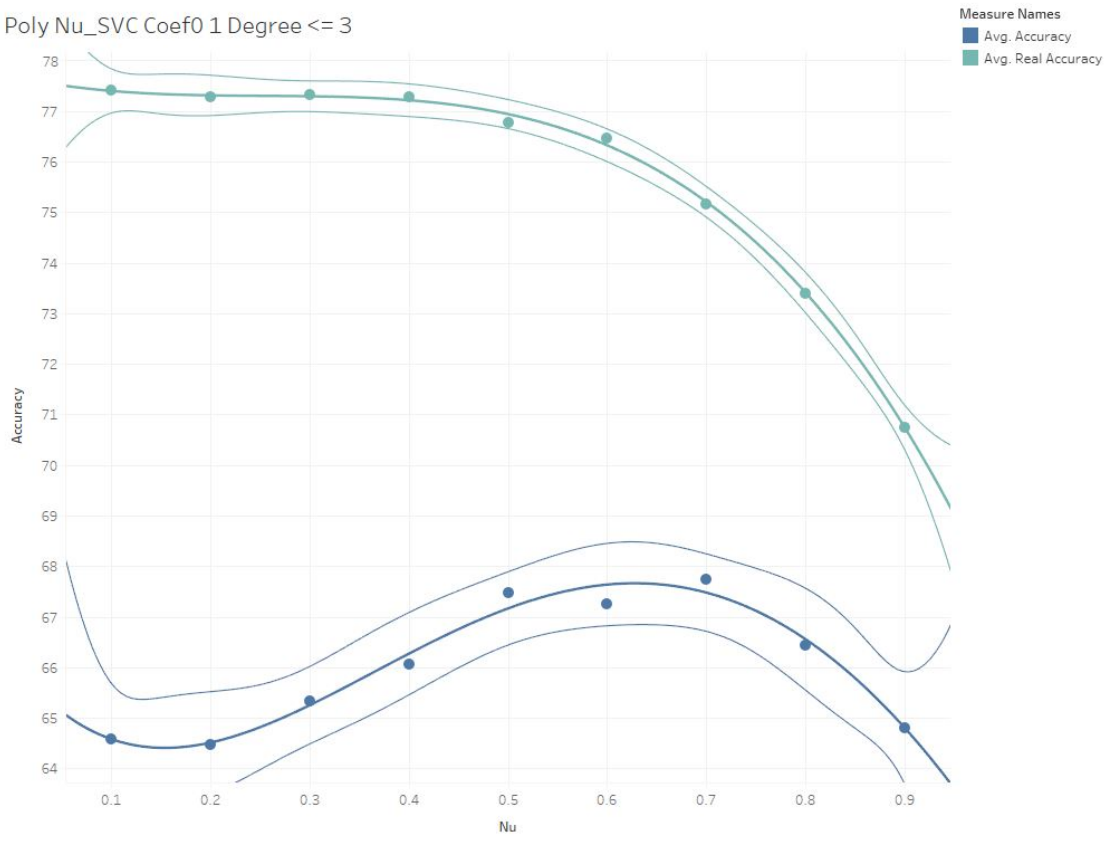

Figure 35. Grid Search Polynomial ν Degree 3 Accuracy

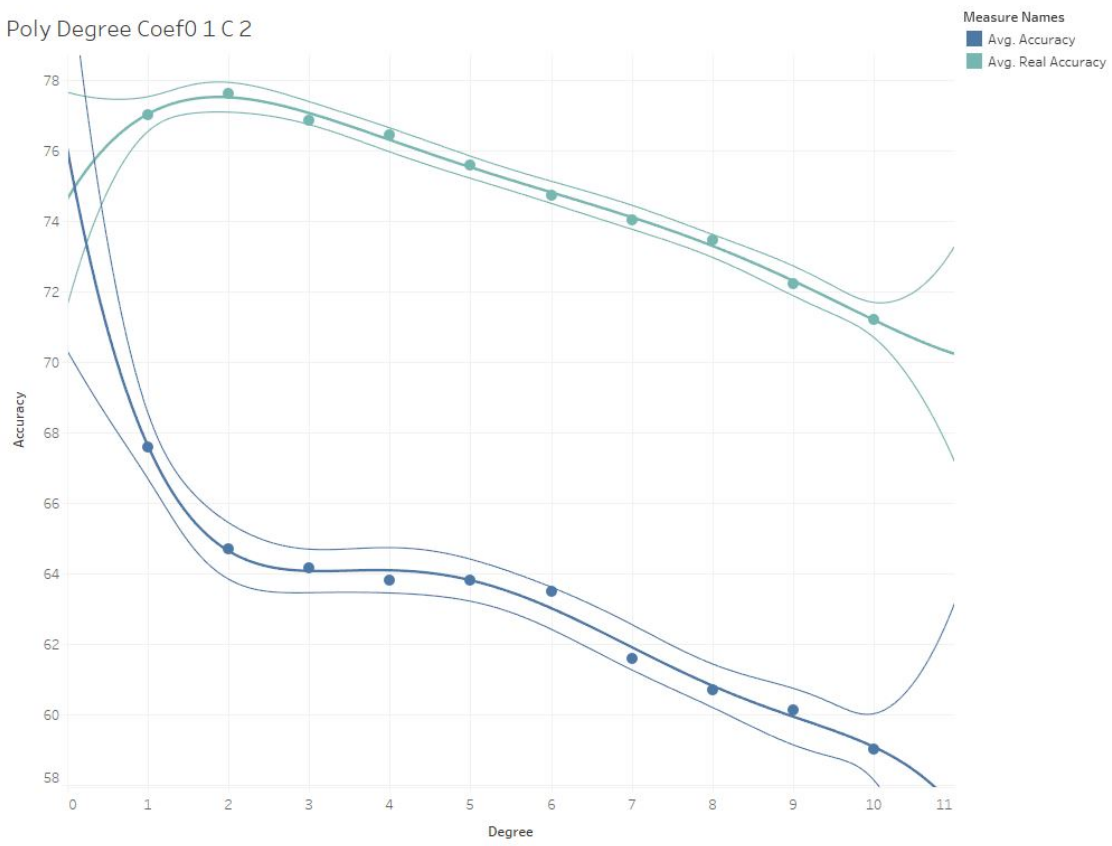

Figure 36. Grid Search Polynomial Degree C 2 Accuracy

dropped quickly. Because that was also consistent across the kernels, the future tests will only go up to a ν value of 0.7. In this case, it appears that again a ν value of 0.4 seems to be the best balance of cross validated accuracy to real accuracy.

Finally, the degree can be reexamined now that the poor performers of C and ν have been removed from the data. As figures 36 and 37 show, while the degree of 3 was the best when bad data is included, the degree of 1 actually outperforms the others on good data. With this, we get a best-case accuracy on C of 67.60 and 77.02, while the best case ν has an accuracy of 66.82 and 77.10.

Because the cross validated accuracy of the best C was almost 1% better while the real accuracy was only .08% worse, the C method was chosen to go forward with. When compared against the linear kernel, which was the best before this, it

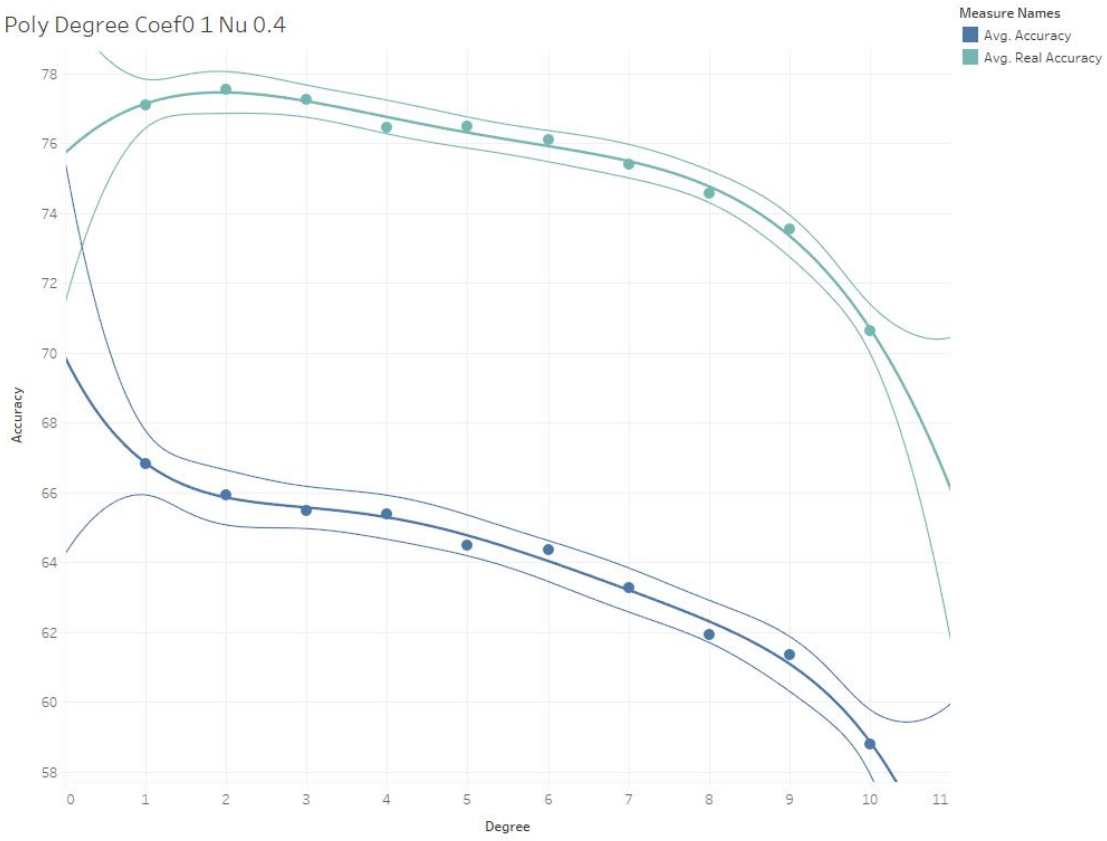

Figure 37. Grid Search Polynomial Degree ν 0.4 Accuracy

N-gram	Entropy
k	0.029
f	0.007
d|	0.006
b	0.004
|i	0.003
in	0.001
he	0.000
y|	0.000
o|	-0.001
an	-0.002

Table 16. N-gram By $H(Parent)$ Entropy

again out-performs the cross validated accuracy by over 1% and again only loses the real accuracy by 0.23%.

Entropy Test

The final test of the first run is to see what effect modifying the entropy formula has. Early on in the research, the formula was being calculated incorrectly by replacing the total entropy of the parent $H(Parent)$ with a hard coded 1. This results in a much different ordering of the N-grams as can be seen in figures 16 and 17. More interestingly, this error was also producing significantly higher accuracies in the preliminary testing. Therefore this test will confirm which method gets the best results. This run was done with a 300 comment training set, a length up to 6-grams, a n-gram percent of 4%, a KNN level of 5% was used and the polynomial kernel using a C of 2, a degree of 1, and a Coef0 of 1.

On this test, 4 runs were completed and resulted in an average standard deviation of 3.70. This means that the minimum number of runs necessary to be sure of the results was 2.10. As can be seen in figure 18, there is no need to graph this as the hard coded 1 value significantly outperforms the $H(Parent)$ method.

N-gram	Entropy
fuck	0.100
uc	0.081
ck	0.081
fu	0.060
fuck|	0.055
fuckin	0.051
u	0.050
fucking	0.047
k	0.031
king	0.030

Table 17. N-gram By Hard Coding Entropy

Entropy	Accuracy	Std Dev	Train Time	Test Time	Positive	Negative
$H(Parent)$	68.31	4	250	11875	66.25	83.77
1	79.30	3.4	82	5043	67.40	86.04

Table 18. Entropy Test

4.1.2 Second Run

With the first full run of parameter optimization completed, it is time to move onto the second run in order to check how much the data changed based on the other parameters.

Number of Comments

The first parameter that had been optimized is the number of comments used in the training set. Again, this will test between 50 and 500 comments and will use a length up to 6-grams, a n-gram percent of 4%, a KNN level of 5%, and the polynomial kernel using a C of 2, a degree of 1, and a Coef0 of 1.

On this test, the 5 runs were completed and resulted in an average standard deviation of 5.27. This means that the minimum number of runs necessary to be sure of the results was 4.27.

Size	Accuracy	Std Dev	Train Time	Test Time	Positive	Negative
50	70.00	19.88	15	672	19.66	79.62
100	69.92	7.19	18	1703	56.50	76.52
150	74.44	6.34	28	2365	62.50	75.38
200	79.17	4.07	47	3180	68.83	77.27
250	78.74	3.78	53	4280	71.98	77.65
300	77.22	4.61	68	5434	70.35	83.33
350	80.93	2.42	84	6087	71.85	87.88
400	79.40	1.11	109	7543	73.96	89.77
450	80.01	2.23	146	8065	74.27	92.42
500	80.19	1.08	159	9465	72.76	94.70

Table 19. Number of Comments Run 2

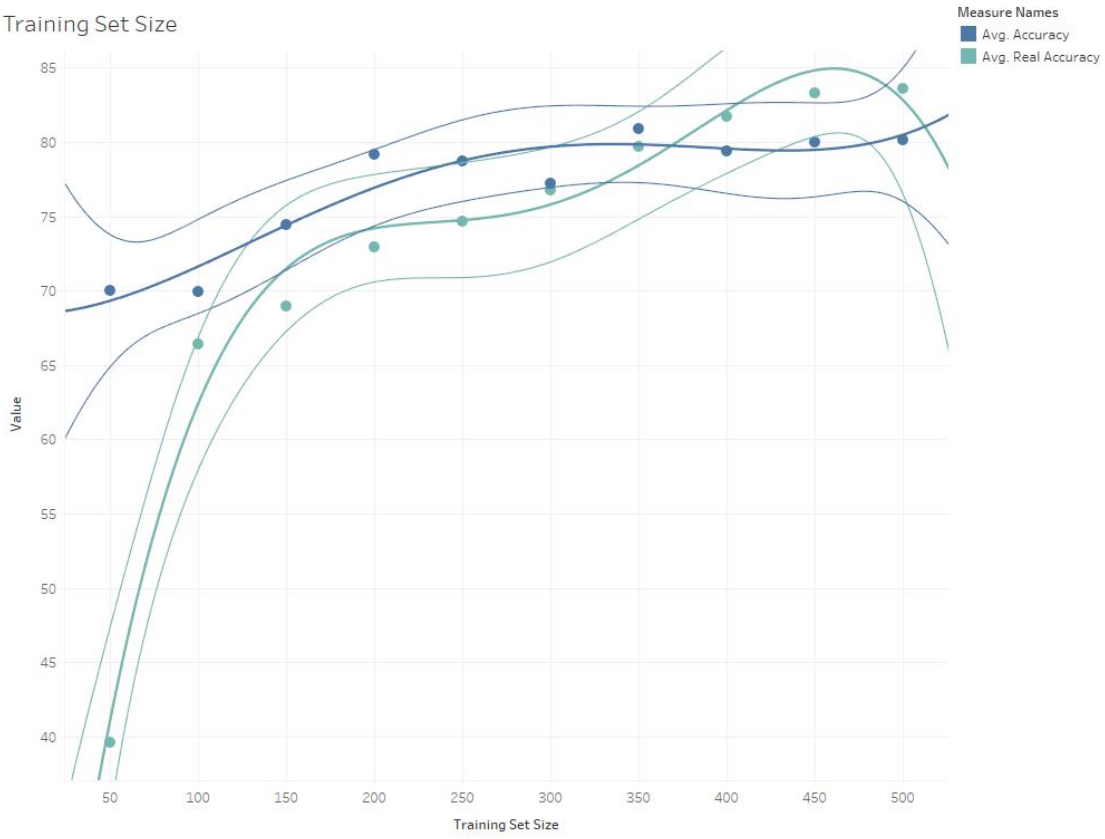

Figure 38. Training Set Accuracy Run 2

Length	Accuracy	Std Dev	Train Time	Test Time	Positive	Negative
1	74.38	2.66	117	2429	74.14	76.52
2	77.70	1	70	3296	73.69	82.64
3	81.48	2.99	63	3524	76.55	85.23
4	79.06	4.73	70	5031	72.89	85.23
5	80.63	0.65	77	4531	76.15	83.02
6	81.00	0.2	93	6117	70.26	89.02
7	78.39	3.08	94	6110	71.54	84.47
8	81.92	1.33	93	7086	70.10	90.19
9	81.50	1.04	86	6617	72.91	89.06
10	80.54	0.37	101	9805	73.13	87.50

Table 20. N-gram Length Run 2

Figure 38 shows that the 300 that was used before is pretty good, but it appears as though 350 represents a substantial bump in this case. Again, were the comments to increase more, the accuracy would as well, but the disk access time would increase exponentially and the accuracy after 350 does not justify that.

N-gram Length

With the training set size re-optimized, the next parameter is the n-gram length. This was retested with 350 comments, a n-gram percent of 4%, a KNN level of 5%, and the polynomial kernel using a C of 2, a degree of 1, and a Coef0 of 1.

On this test, the 2 runs were completed and resulted in an average standard deviation of 1.80. This means that the minimum number of runs necessary to be sure of the results was 0.50.

As can be seen in figure 39, until the length 3 n-gram, the accuracies are low, but after 3 there is no significant increase in the accuracies. This is most notable when comparing length 3 to the length that had been used of 6 where both the cross validated and real accuracies are now higher at 3. Thus, going forward, the

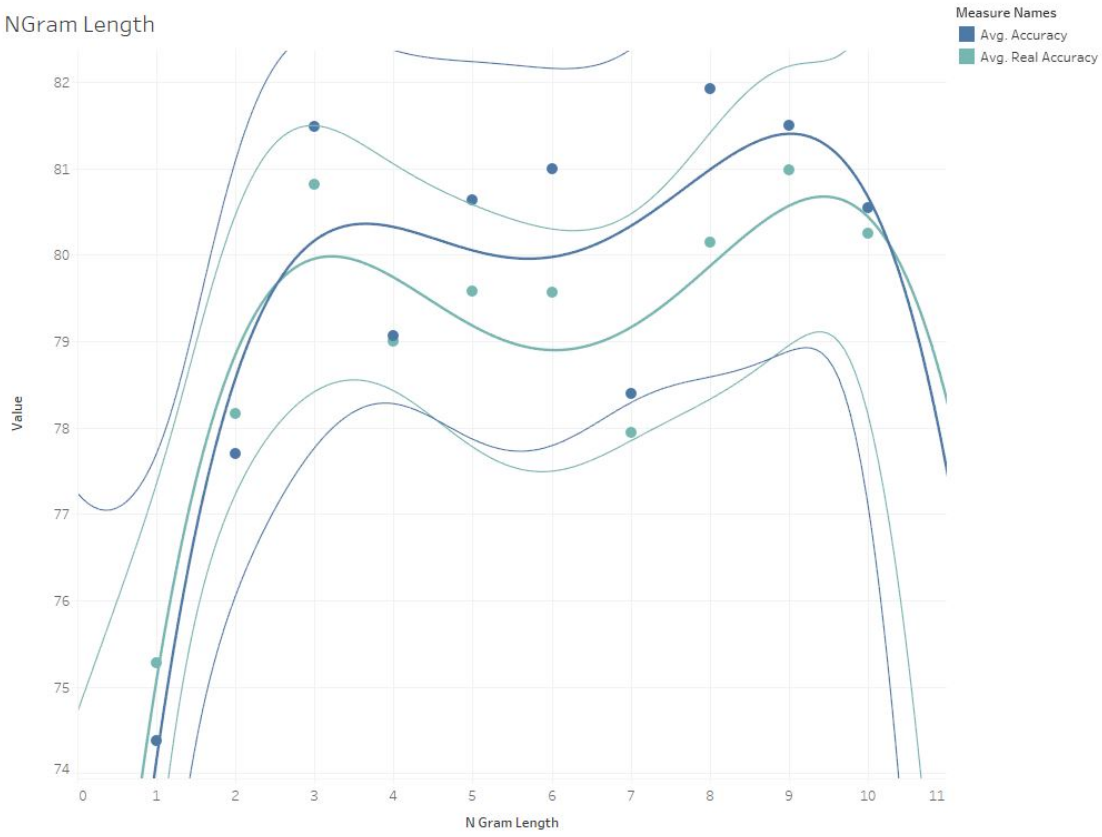

Figure 39. N-gram Length Run 2 Accuracy

Percent	Accuracy	Std Dev	Train Time	Test Time	Positive	Negative
1	78.39	3.59	70	2344	75.31	76.60
2	78.79	0.97	62	2765	77.55	79.17
3	77.88	0.5	63	2976	76.91	83.77
4	78.24	0.33	63	4195	73.53	83.71
5	80.92	1.13	62	3695	73.13	84.47
6	79.22	1.52	78	4875	73.61	84.91
7	83.40	2.46	78	5086	71.66	87.88
8	77.25	1.37	94	7148	71.06	84.85
9	76.62	1.37	101	6461	76.63	84.47
10	78.65	2.32	94	6719	72.57	87.55

Table 21. N-ram Percent Run 2

lower length of 3 will be used.

N-gram Percent

With the new n-gram length chosen, the next step is to reevaluate the percent of n-gram's we are taking. This was retested with 350 comments, a length up to 3-grams, a KNN level of 5% and the polynomial kernel using a C of 2, a degree of 1, and a Coef0 of 1.

On this test, the 2 runs were completed and resulted in an average standard deviation of 1.56. This means that the minimum number of runs necessary to be sure of the results was 0.37.

As figure 40 shows, at a low percent the accuracies are poor, but the gains after are not dramatic. In the case of this data, the 7% was the best performer and was used for the rest of the tests.

KNN Level

After the n-grams are optimized, the next parameter was the KNN Level. This was retested with 350 comments, a length up to 3-grams, a n-gram percent of 7%, and the polynomial kernel using a C of 2, a degree of 1, and a Coef0 of 1.

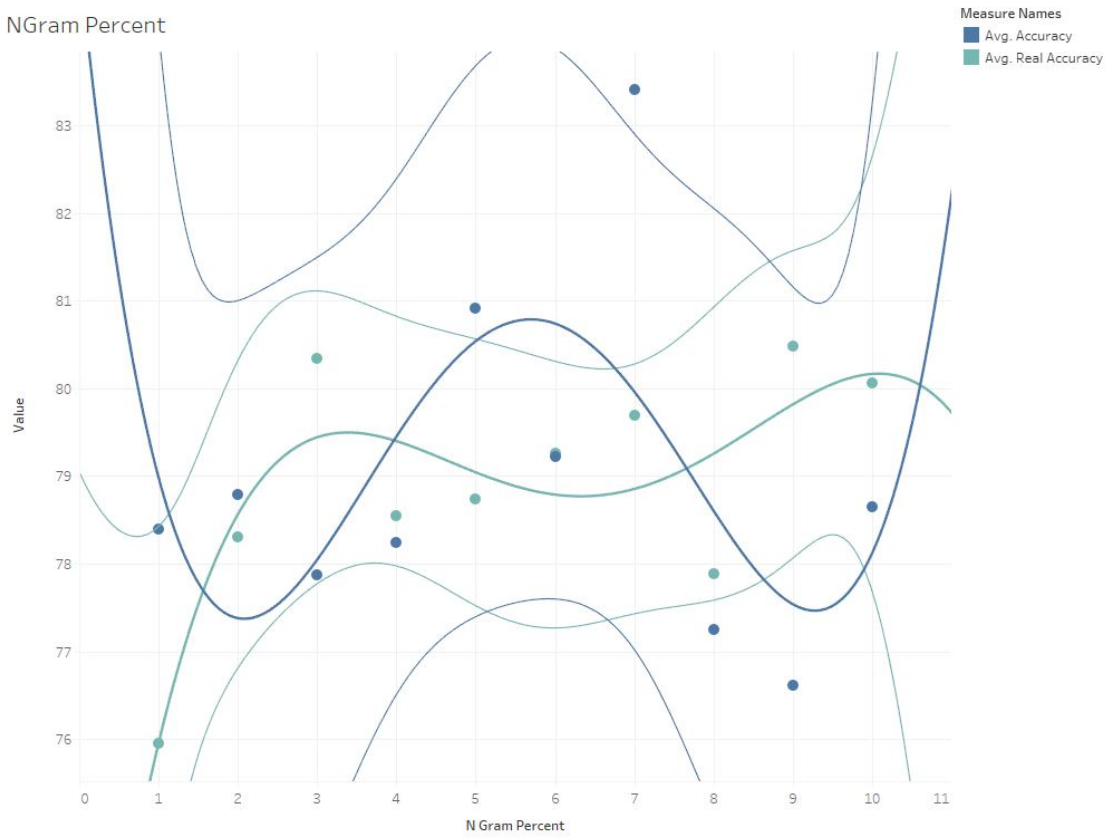

Figure 40. N-gram Percent Run 2 Accuracy

knn Level	Accuracy	Std Dev	Train Time	Test Time	Positive	Negative
1	68.05	0.94	46	3203	46.16	86.04
2	74.90	1.24	54	3867	70.83	85.98
3	78.78	0.98	69	4195	74.38	84.85
4	82.34	0.24	70	4640	76.01	83.71
5	81.18	1.02	62	4437	76.04	83.71
6	81.83	1.05	70	4429	75.99	84.09
7	81.71	1.21	70	4570	76.01	84.09
8	81.71	1.21	78	4202	76.01	84.09
9	81.71	1.21	70	4265	76.01	84.09
10	81.71	1.21	70	4461	76.01	84.09

Table 22. KNN Level Run 2

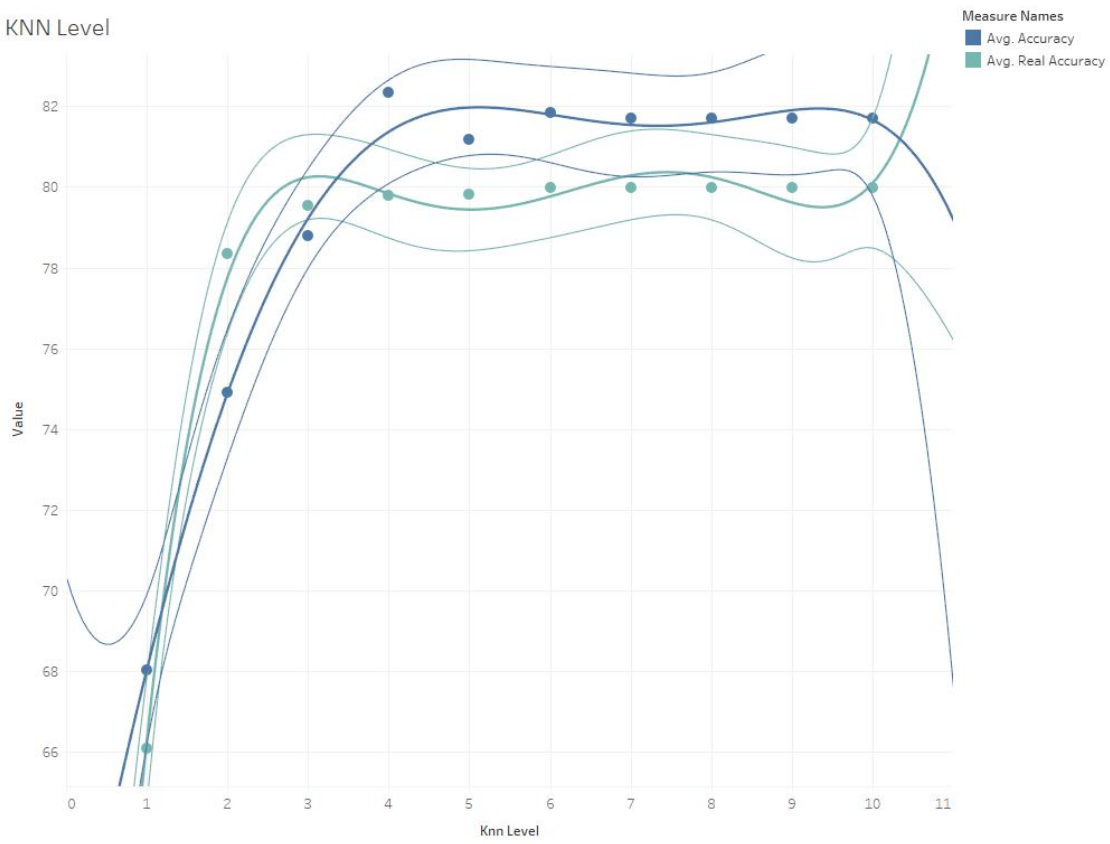

Figure 41. KNN Level Run 2 Accuracy

On this test, the 2 runs were completed and resulted in an average standard deviation of 1.03. This means that the minimum number of runs necessary to be sure of the results was 0.16.

As happened in the first run through, the KNN level eventually levels out which can be seen in figure 41. The best-case prior to the leveling out is 4% which managed to outdo the eventual level rate of 81.71.

Grid Search

Finally, it is time to rerun the grid search to see how much has changed with the new optimizations. For this run we retested with 350 comments, a length up to 3-grams, a n-gram percent of 7%, and a knn level of 4%. Because of the results from the last run, the range on C has been restricted to 0 to 7, the range of ν is

C	Accuracy	Std Dev	Train Time	Test Time	Positive	Negative
0	73.88	1.29	906	122524	77.42	70.08
1	76.76	1.26	414	48828	75.13	76.89
2	77.77	1.86	2054	48867	74.84	81.82
3	79.94	0	2929	54273	73.90	85.98
4	79.80	1.02	839	43913	73.13	86.04
5	80.67	1.03	15195	27219	72.15	86.36
6	80.52	0.82	18484	148641	71.55	85.98
7	79.08	0.01	15812	115961	70.89	85.98

Table 23. Linear C_SVC kernel Grid Search Run 2

ν	Accuracy	Std Dev	Train Time	Test Time	Positive	Negative
0.1	79.66	0.81	16359	51773	70.73	85.98
0.2	81.10	0.01	10539	41093	71.86	85.66
0.3	80.09	1.02	11929	65235	72.57	86.04
0.4	79.94	0.82	14008	120524	72.97	86.36
0.5	78.93	0.21	10421	43968	74.36	84.47
0.6	77.63	1.25	8664	181125	74.70	81.06
0.7	76.47	0.85	10211	87890	74.94	76.23

Table 24. Linear ν_SVC kernel Grid Search Run 2

reduced to 0.1 to 0.7, Coef0 is set to 1 and the range of degrees is reduced to 1 to 5.

On this test, the 2 runs were completed and resulted in an average standard deviation of 1.43. This means that the minimum number of runs necessary to be sure of the results was 0.31.

Beginning again with the linear kernel, figure 42 shows the accuracies increase until a C of 5 before beginning to curve back down. This results in a best case of 80.67 and 79.19. The ν values in figure 43 show that in this case a ν of 0.2 is the best performing, although the real accuracy does continue to increase slightly after that point. This results in the best case of 81.10 and 78.76.

C	Degree	Accuracy	Std Dev	Train Time	Test Time	Positive	Negative
0	1	76.04	1.88	11515	182883	74.25	78.79
0	2	80.65	1.83	12601	39969	76.33	85.28
0	3	78.36	0.19	11672	84532	75.75	87.50
0	4	78.35	0.22	20226	83390	74.35	88.30
0	5	76.91	0.17	15406	30789	73.63	87.88
1	1	78.93	0.21	13578	272938	74.63	83.71
1	2	80.66	0.61	17571	89539	75.51	86.79
1	3	79.08	0.4	15297	55968	73.93	88.30
1	4	77.92	0.84	13859	82890	73.35	88.30
1	5	75.17	2.29	13992	316649	72.95	88.30
2	1	79.80	0.21	22234	179024	73.24	86.36
2	2	80.52	0.01	17297	140172	74.01	87.50
2	3	79.36	0.82	26835	31023	73.53	88.64
2	4	76.90	2.28	22812	48883	72.52	88.64
2	5	74.88	2.71	10430	53789	71.91	88.68
3	1	79.95	1.22	15320	319938	72.94	85.66
3	2	80.38	0.21	21422	175188	73.07	87.88
3	3	78.05	2.67	14656	105891	72.49	89.02
3	4	76.61	2.69	12524	138141	71.70	89.06
3	5	74.59	3.12	12109	91438	71.21	88.68
4	1	80.81	0.82	10203	39820	72.19	86.36
4	2	79.79	1.02	15133	75695	72.73	88.26
4	3	78.05	2.67	15977	21414	71.83	89.06
4	4	75.45	4.33	9148	62883	70.92	89.06
4	5	74.59	3.12	5070	212774	71.21	88.68
5	1	80.09	0.2	9773	77734	71.48	85.98
5	2	79.07	2.05	9234	48320	71.96	88.30
5	3	77.18	3.91	15281	218867	71.26	89.06
5	4	75.45	4.33	15773	116593	70.92	89.06
5	5	74.59	3.12	12031	105727	71.21	88.68
6	1	79.08	0.01	18007	29945	70.71	85.98
6	2	79.21	1.84	16187	69242	71.46	88.68
6	3	76.74	4.52	24382	115328	70.75	89.39
6	4	75.45	4.33	13750	246273	70.92	89.06
6	5	74.59	3.12	13351	24851	71.21	88.68
7	1	78.78	2.46	11515	164946	70.01	86.04
7	2	77.61	4.11	10968	37734	70.47	88.30
7	3	76.74	4.52	11835	84101	70.75	89.39
7	4	75.45	4.33	9749	55328	70.92	89.06
7	5	74.59	3.12	18601	61617	71.21	88.68

Table 25. Polynomial C_SVC kernel Grid Search Run 2

ν	Degree	Accuracy	Std Dev	Train Time	Test Time	Positive	Negative
0.1	1	79.51	1.02	14203	73297	70.90	85.98
0.1	2	80.81	0.82	6632	118976	73.07	87.55
0.1	3	79.22	0.62	9327	52063	73.55	88.30
0.1	4	77.63	0.84	11867	52945	73.98	88.30
0.1	5	76.04	0.65	16242	29218	74.30	88.30
0.2	1	81.24	0.19	14930	224141	71.85	85.61
0.2	2	80.52	0.01	9335	229703	74.17	87.12
0.2	3	79.08	1.62	7368	170391	75.07	87.50
0.2	4	77.92	0.02	19219	80375	75.45	87.92
0.2	5	76.62	1.05	11726	22054	75.80	87.92
0.3	1	80.09	1.02	14851	176289	72.65	86.04
0.3	2	80.52	0.81	5976	80328	75.13	86.79
0.3	3	78.93	1.03	9914	157602	76.22	86.42
0.3	4	77.34	1.25	8812	265626	77.02	86.42
0.3	5	75.90	0.45	10593	110867	77.98	85.28
0.4	1	79.79	0.61	12132	254930	72.95	86.42
0.4	2	81.09	0.81	21187	203781	75.80	85.61
0.4	3	78.20	1.24	17390	294586	77.85	85.28
0.4	4	76.18	1.67	15492	348859	80.04	84.47
0.4	5	74.74	1.28	17117	184461	81.50	80.30
0.5	1	78.93	0.21	15328	171132	74.35	84.47
0.5	2	79.36	0.82	15789	259680	77.05	83.71
0.5	3	77.04	2.48	16851	177828	79.37	82.26
0.5	4	75.32	0.86	17281	263141	81.93	77.74
0.5	5	73.58	1.7	14577	78015	83.82	74.34
0.6	1	77.63	1.25	19492	76469	74.70	81.06
0.6	2	77.78	0.23	11257	26446	78.01	80.38
0.6	3	75.04	0.05	13828	35586	81.18	77.27
0.6	4	73.29	2.52	15413	24547	83.37	72.08
0.6	5	70.84	2.75	13820	227125	85.29	66.42
0.7	1	76.47	0.85	10773	201766	74.92	76.23
0.7	2	74.89	0.26	11234	28930	79.23	74.72
0.7	3	72.86	2.32	19969	128078	83.66	69.43
0.7	4	69.25	2.15	19515	217859	87.29	62.50
0.7	5	65.06	3.21	15867	67696	89.85	56.06

Table 26. Polynomial ν_SVC kernel Grid Search Run 2

C	Accuracy	Std Dev	Train Time	Test Time	Positive	Negative
0	73.88	0.88	7085	221961	74.31	82.26
1	77.49	0.18	9086	315876	73.69	87.55
2	79.09	2.44	25062	60937	72.25	89.02
3	79.52	2.24	15742	394571	70.60	89.06
4	78.07	1	13789	16836	69.75	89.39
5	77.64	0.38	6461	210875	68.95	89.81
6	77.34	0.03	18930	237125	68.28	90.19
7	77.34	0.03	26008	96172	68.20	90.19

Table 27. RBF C_SVC kernel Grid Search Run 2

ν	Accuracy	Std Dev	Train Time	Test Time	Positive	Negative
0.1	77.93	1.2	38422	126664	69.45	89.39
0.2	79.52	2.24	10601	86493	70.49	89.06
0.3	79.67	2.45	10125	17820	71.47	89.06
0.4	78.36	1.41	5414	10945	72.55	88.26
0.5	78.21	0.39	7726	48727	73.16	88.26
0.6	77.20	0.18	9430	48328	73.74	86.79
0.7	75.61	1.27	7422	21906	73.79	84.53

Table 28. RBF ν_SVC kernel Grid Search Run 2

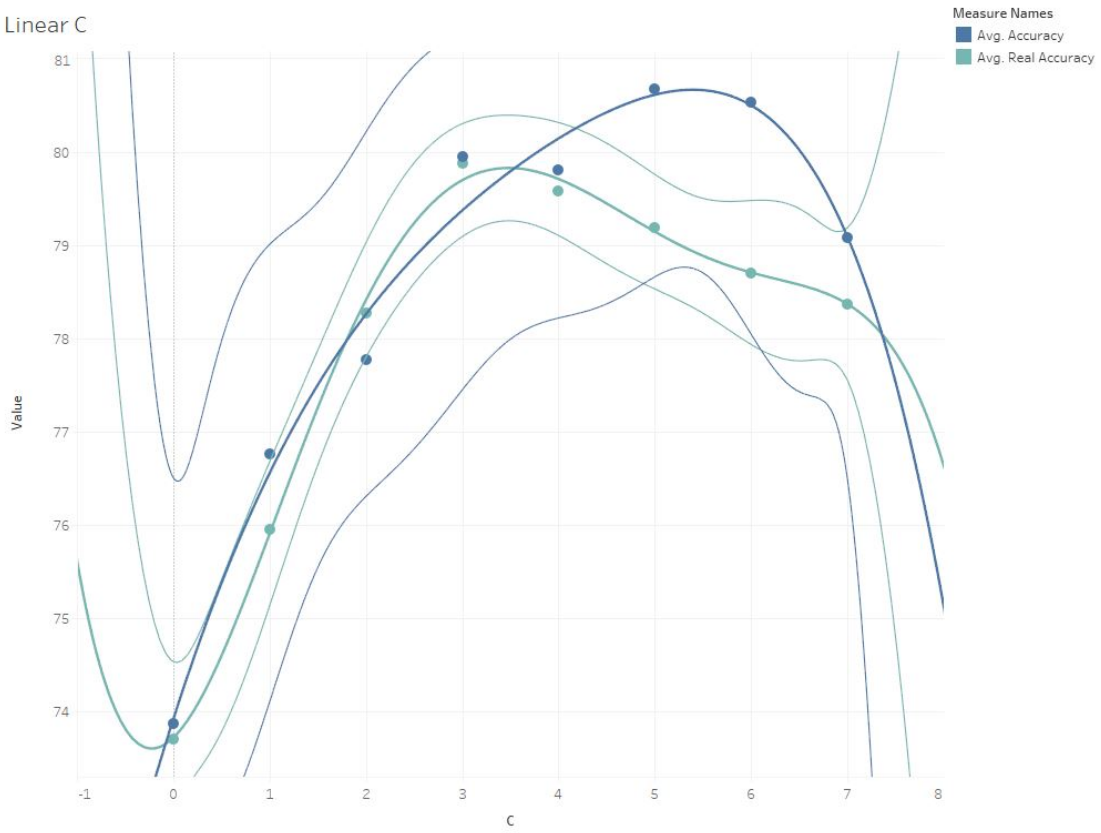

Figure 42. Grid Search Linear C Run 2 Accuracy

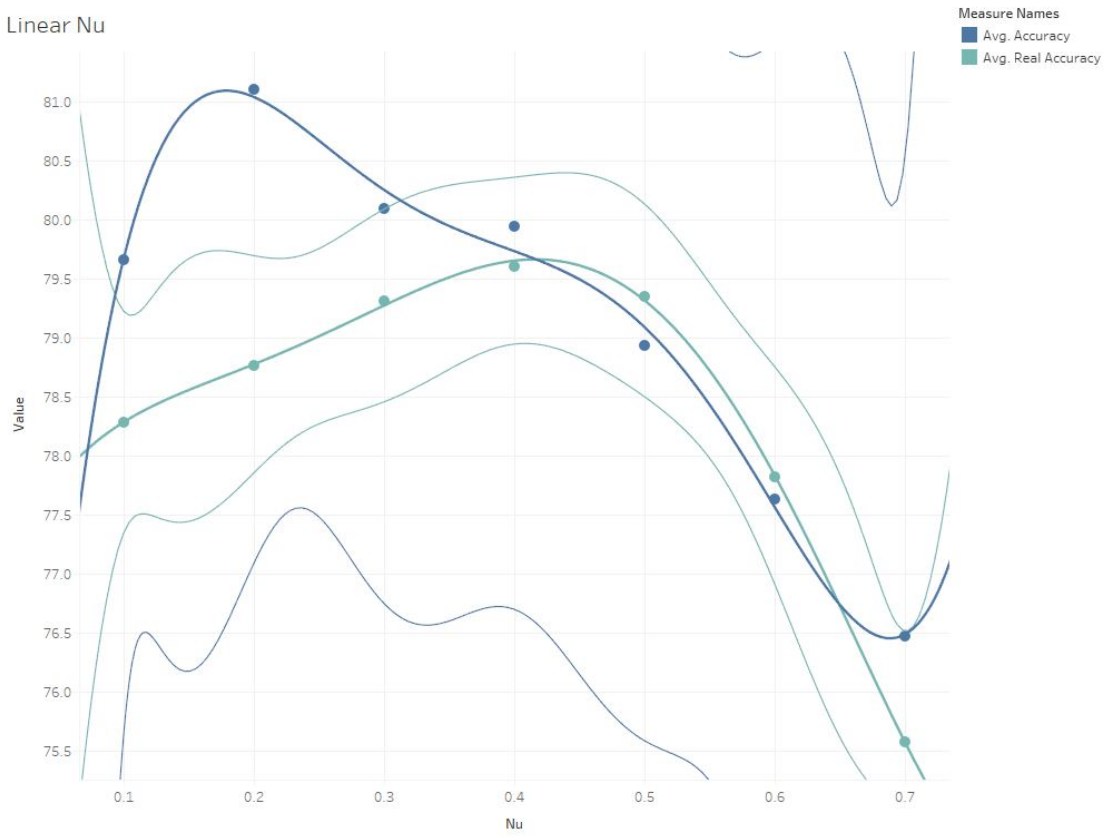

Figure 43. Grid Search Linear ν Run 2 Accuracy

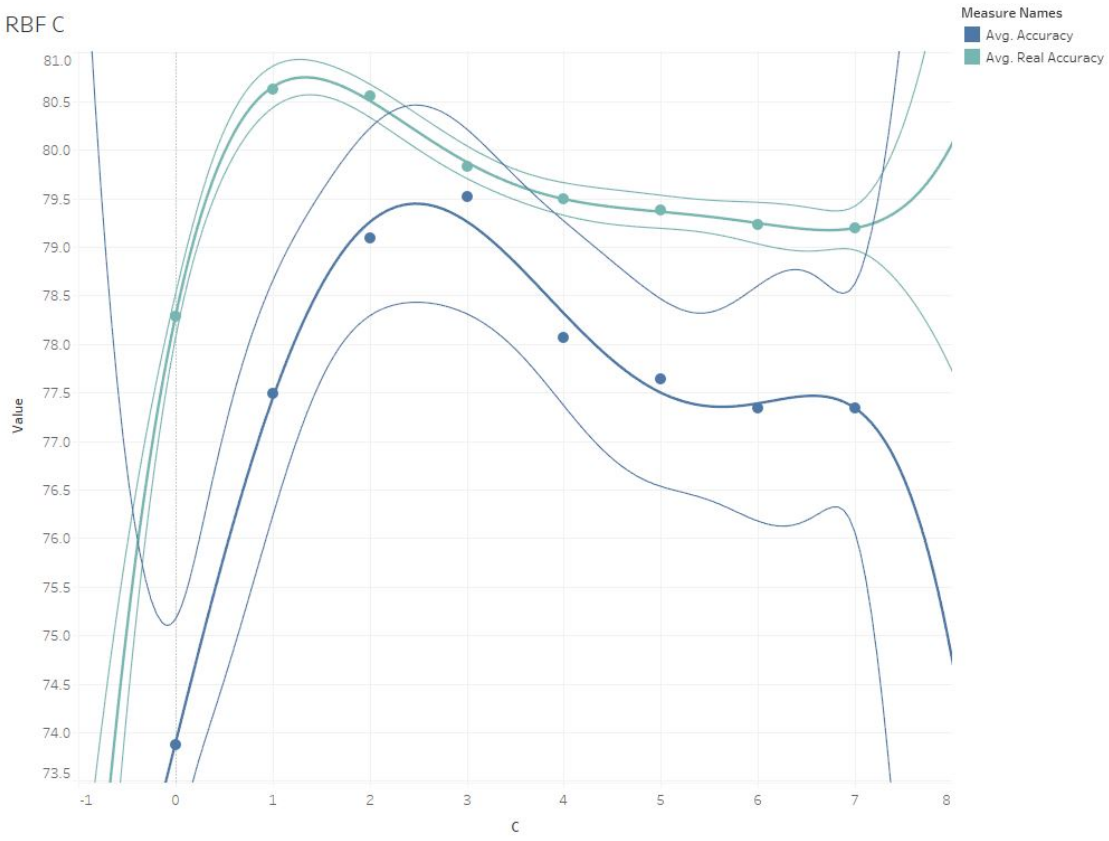

Figure 44. Grid Search RBF C Run 2 Accuracy

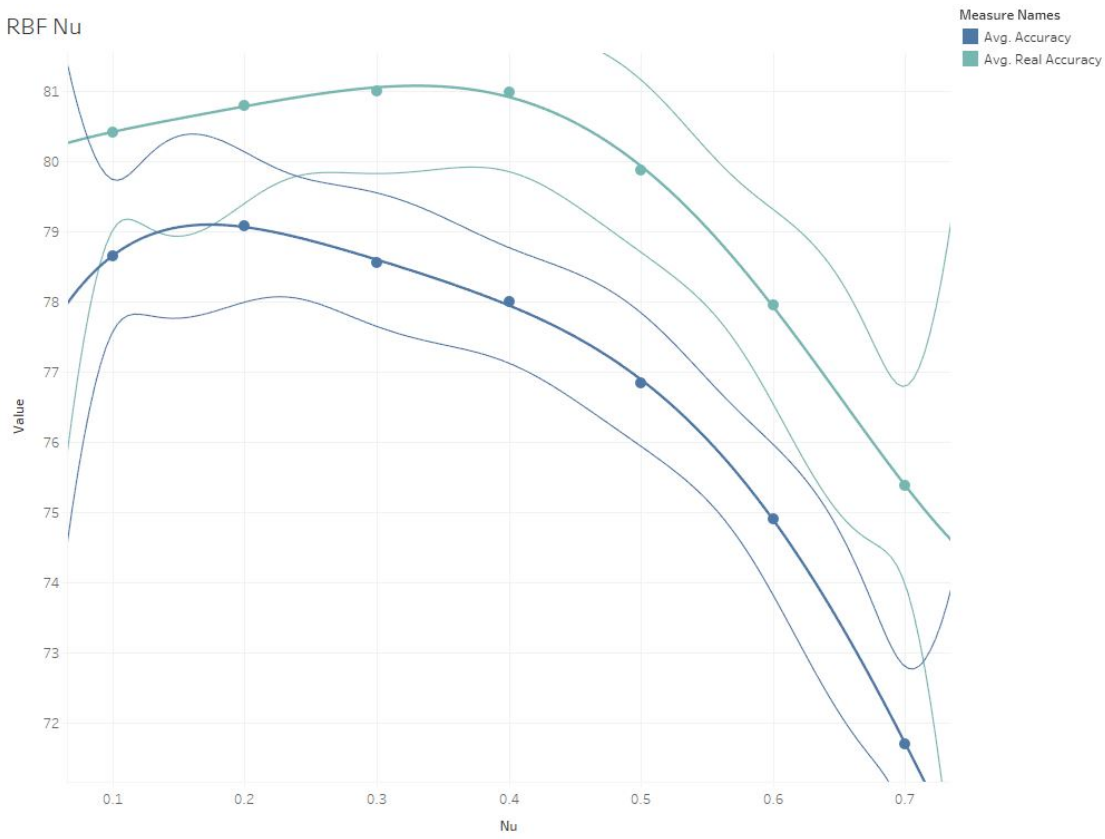

Figure 45. Grid Search RBF ν Run 2 Accuracy

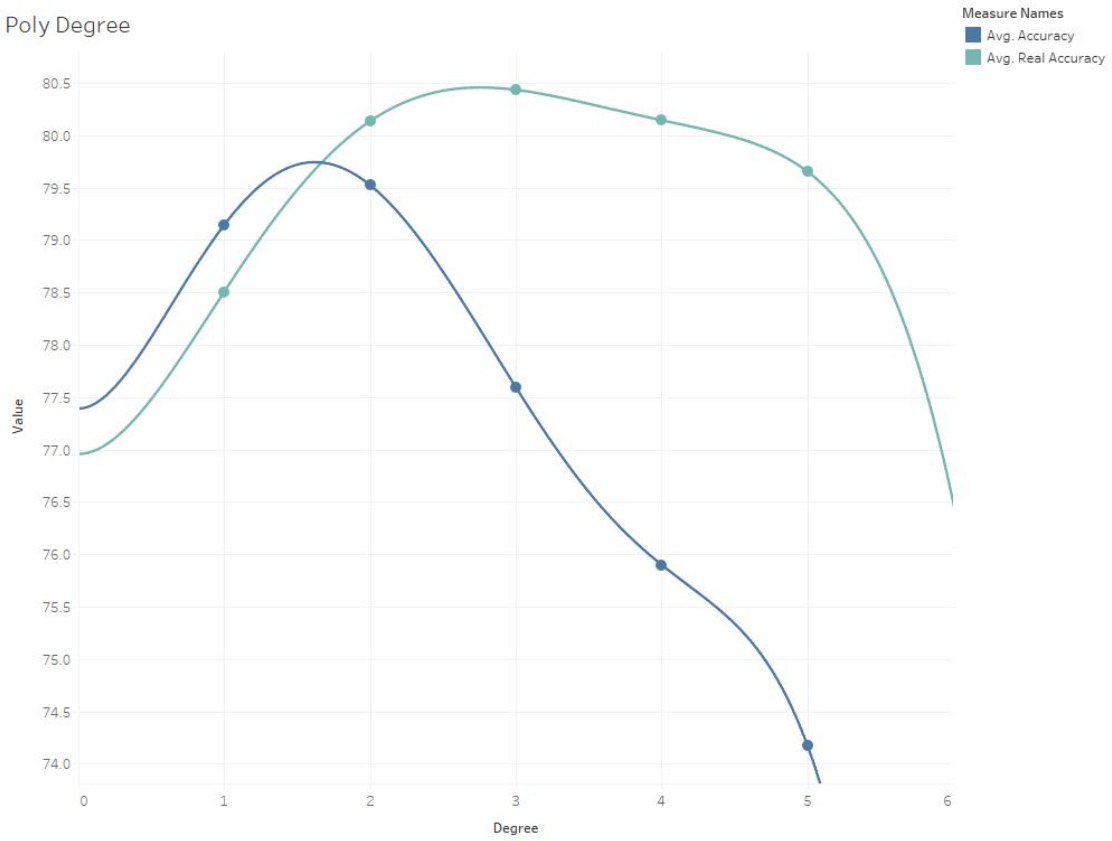

Figure 46. Grid Search Polynomial Degree Run 2 Accuracy

For the RBF kernel, figure 44 shows a C of 2 being the best performance. Even though a C of 3 has a slightly higher cross validated accuracy, the real accuracy drops considerably. This results in a best case of 79.09 and 80.55. Like with the linear kernel, figure 45 shows that ν of 0.2 is the best choice. This has an accuracy of 79.08 and 80.80.

This time, because of the data being restricted to fewer parameters, the first graph that was analyzed for the polynomial kernel was the degree graph shown in figure 46. It is clear in this case that the best degree this time around was 2. Taking that, figures 47 and 48 were restricted to showing only data at degree 2. This showed that C peaked at 1 and ν peaked at 0.3. This results in a best case for C of 80.66 and 81.15 while ν managed 80.52 and 80.96.

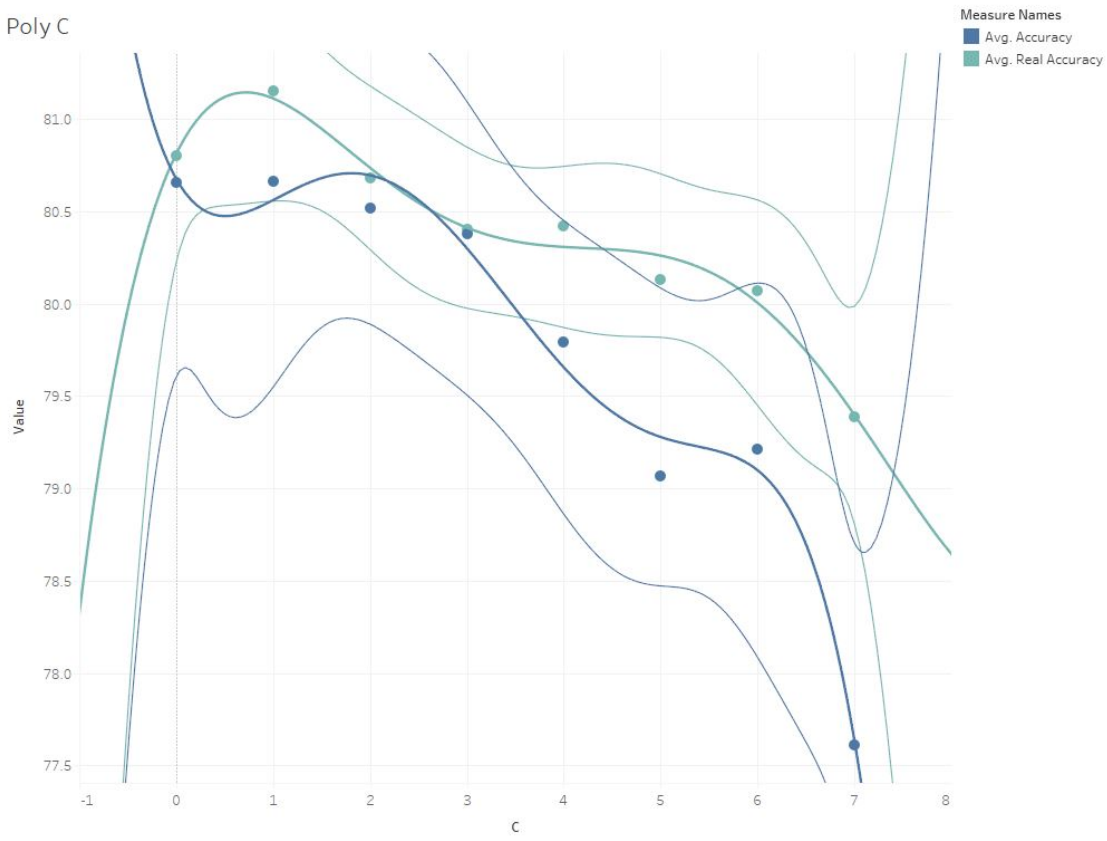

Figure 47. Grid Search Polynomial C Degree 2 Run 2 Accuracy

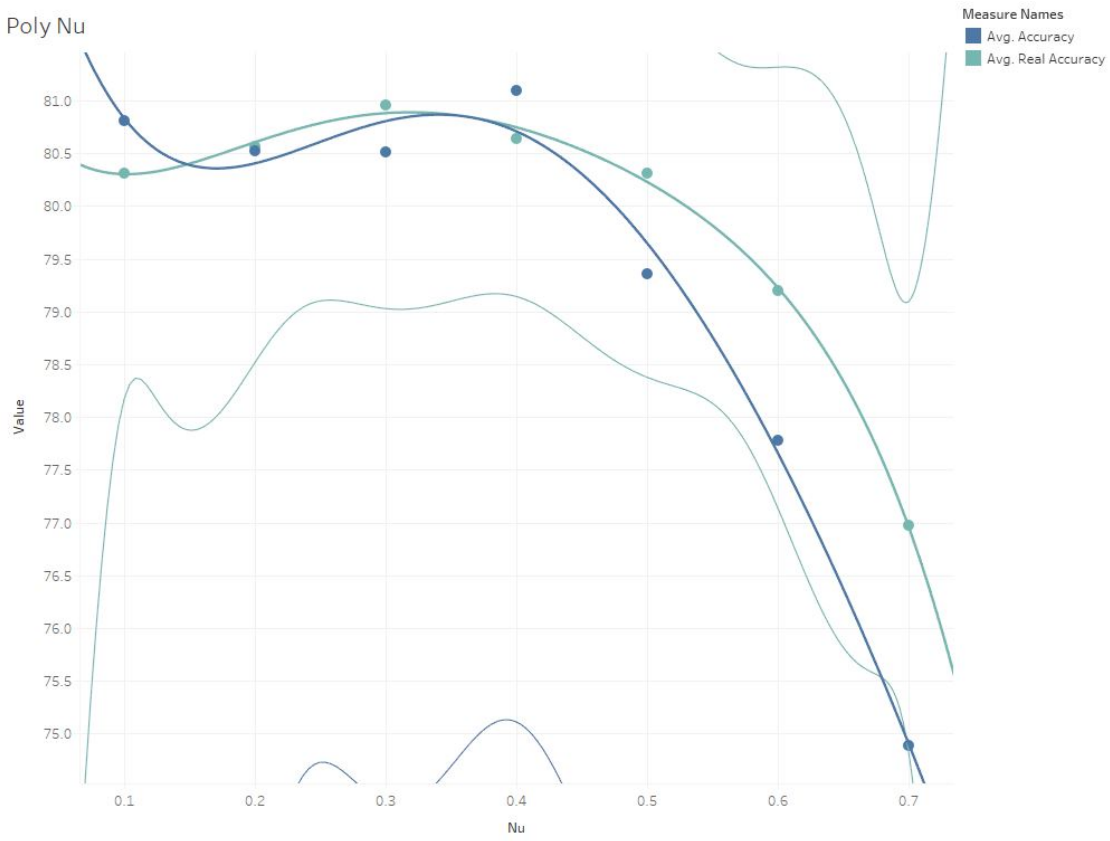

Figure 48. Grid Search Polynomial ν Degree 2 Run 2 Accuracy

Picking the best option from the grid search is one of the most subjective tasks of the dissertation. However, either the linear ν or the polynomial C options would be a good selection. In this case, the choice used from here on out is the polynomial C because the additional flexibility of the polynomial kernel may aid on differing sets of data.

4.2 Testing the Model

This section will test what the best case accuracies are given different groups of comments. In each case, at least 5 runs of the data will be used in order to ensure a good average, but the standard deviation will still be calculated to check if more than 5 are required. For each of these tests, the parameters used were 350 comments, 7% of n-grams up to a length of 3, a knn level of 4%, and the polynomial kernel with a C of 1, a degree of 2 and a Coef0 of 1.

Along with the cross-validated accuracy and the real accuracy that were reported on all other figures, when testing the model the .632+ bootstrap values were also calculated[2]. In this method 200 training sets were randomly generated from the training set with replacement, and then after training the model on those comments, the remaining comments that were not randomly selected were used as the testing set. The error from this calculation, Err_{boot}, is then averaged with the training error from the original training set $\overline{err}$ to produce a range of error estimations:

$$Err_{.632} = 0.368 * \overline{err} + 0.632 * Err_{boot} \tag{11}$$

It is from these estimations that the 95% confidence intervals were calculated.

Accuracy	Std Dev	Train Time	Test Time	Positive	Negative
81.76[80.37,89.61]	0.96	3071	4656	74.27	89.43

Table 29. Legal Method Average Accuracy

Accuracy	Std Dev	Train Time	Test Time	Positive	Negative
73.43[65.51,79.97]	3.59	4118	9887	74.74	81.35

Table 30. Terms of Service Method Average Accuracy

4.2.1 Legal Method

With all of the optimization finished, it is time to get a good reading on the maximum accuracy that can be gained on the legal method that tuned the original optimization.

After 5 runs, the standard deviation is 0.96 which means that 0.14 runs were needed for a good result.

As both the data in table 29 and figure 49 show, the cross validated accuracy is above 80%. The real accuracy also agrees with that, although there is a slight bias towards correctly identifying the banned comments.

4.2.2 Terms of Service Method

After getting the average accuracy of 81.76 on the legal method, all of the comments used were reclassified as to whether they contained politics or religion in them. This will test how accurate the optimized parameters can be on training sets that are significantly different from the set used to optimize the parameters.

After 5 runs, the standard deviation is 3.59 which means that 1.98 runs were needed for a good result.

The terms of service method performed lower than the legal method as seen in table 30 and figure 50. The cross validated accuracy hovers around 70% in this

Figure 49. Legal Method Accuracy

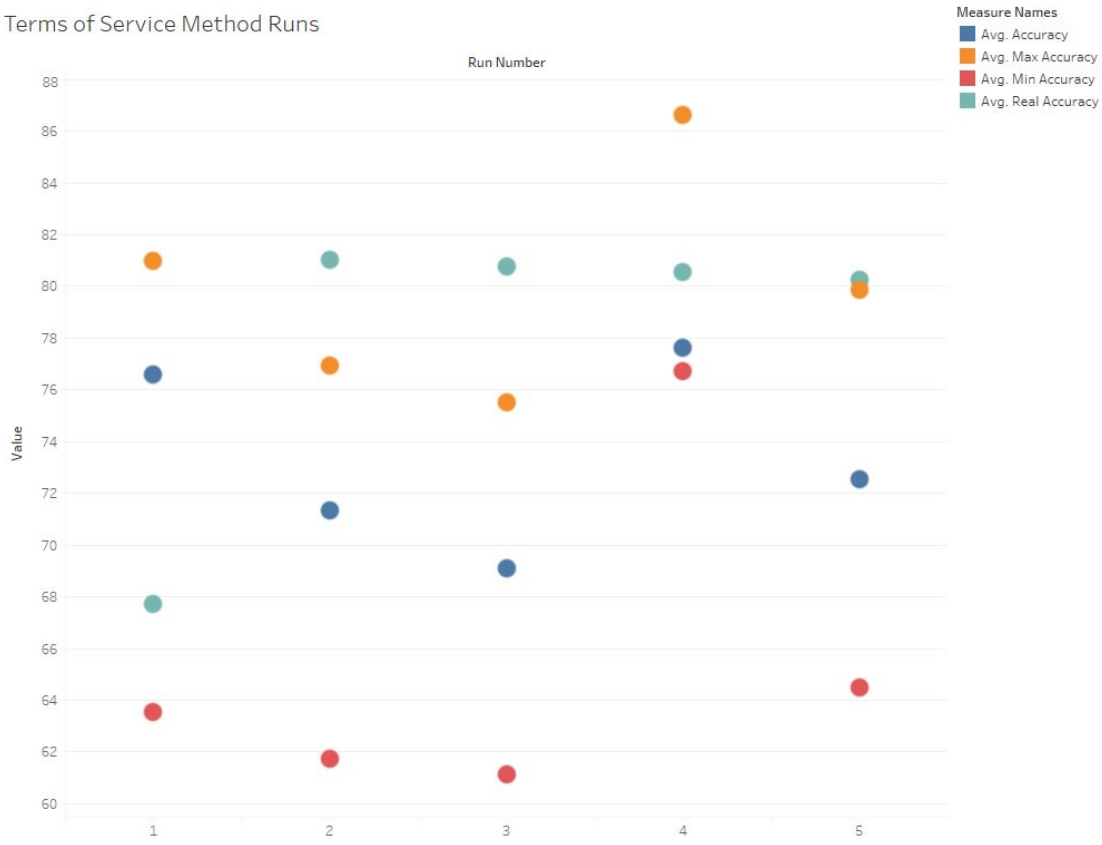

Figure 50. Terms of Service Method Accuracy

Accuracy	Std Dev	Train Time	Test Time	Positive	Negative
69.92[58.34,73.63]	2.19	4242	8935	69.52	77.84

Table 31. Overall Method Average Accuracy

test while the real accuracy in this case is slightly higher reaching in to the 80% range.

4.2.3 Overall Method

With the accuracy of the legal and the terms of service method found, the last test for the YouTube comments is to see how well the model handles an integrated method combining both of the prior methods. For this test, any comment that was marked as restricted in either of the prior tests is now restricted in this one.

After 5 runs, the standard deviation is 2.19 which means that 0.74 runs were needed for a good result.

Like the terms of service method, the overall method does perform worse than the legal method that all of the parameters were optimized on. Table 31 and figure 51 both show the cross validated accuracy again hovers around 70% while the real accuracy of both the positive and negative class are lower although still acceptable.

4.2.4 Twitter Run

Finally, for the last accuracy test, the YouTube comments are set aside and a test run is done utilizing comments taken from Twitter. This will test whether the results up until this point were only based on the type and length of comment that is posted on YouTube, or if it can generalize well to other sites including ones with forced character limits.

After 5 runs, the standard deviation is 1.19 which means that 0.22 runs were needed for a good result.

Figure 51. Overall Method Accuracy

Accuracy	Std Dev	Train Time	Test Time	Positive	Negative
83.76[82.30,91.39]	1.19	1808	2805	77.29	92.48

Table 32. Twitter Method Average Accuracy

Figure 52. Twitter Method Accuracy

Surprisingly, the Twitter comments performed on par with the original YouTube comments using the legal method. The average cross-validated accuracy is 83.76% which is slightly higher than the YouTube comments and much higher than the terms of server or overall method. The real accuracy is even more skewed towards the negative classes, however.

4.2.5 Retraining

With the maximum accuracies established, two different retraining methods were tested to see which has a better performance in increasing the accuracy. For the first one, the comments that were incorrectly classified were added to the training set allowing the set to grow. For the second method, the training set was held at the 350 comments, but the training set was built out of the comments most often misclassified. For both of these test, the original set of YouTube comments was used with the legal method classification scheme.

Adding Comments Retraining

For the first method of retraining, after each run all comments that were misclassified were marked by changing their training value from a 1 to a 2 or a -1 to a -2. The next time the training set was built, all of the -2 values were added to the training set and then an equivalent number of 2 values were added as well. This was done because there were much fewer bullying comments than normal ones, so this was the only method to keep the set balanced. This was repeated until such a time as no more bullying comments were added to the set.

As figure 33 shows, at each step the accuracy improved, with the sole exception being the fourth run. In the end, only 6 runs were completed, at which point all of the misclassified bullying comments were in the list. This led to an average standard deviation of 4.74 which justified doing this test in only 3.46 runs.

Run	Positive	Negative
1	73.32	78.11
2	80.81	87.12
3	83.04	90.80
4	82.39	90.77
5	83.44	92.31
6	83.36	93.02

Table 33. Adding Retraining

Run	Positive	Negative
1	74.55	89.81
2	76.64	87.88
3	73.92	88.64
4	71.61	90.91
5	78.00	90.38
6	74.84	90.70
7	78.73	89.19
8	74.57	90.63
9	76.44	89.66
10	77.81	92.00

Table 34. Training Size Comparison

As this result was not entirely suprising given the earlier observation that simply using more comments resulted in a higher accuracy, 10 additional runs were computed utilizing the same number of comments, but chosen at random instead of relying on pulling in misclassified comments.

Figure 34 clearly shows that while having more comments did improve above the starting point of the retraining method, at no point did the accuracy meet or beat the final retrained set even though the same number of comments were utilized. The average standard deviation of this set is 3.39 and only requires 1.76 runs. This shows that the method of adding misclassified comments to the training set does outperform simply utilizing more comments.

While not exactly the same method used, this strategy is similar to the boost-

ing algorithm created at the University of Ottawa in order to handle data with imbalanced data sets[3]. In their case, rather than keep the data sets balanced, they put all of the training data in and then modified the weights on the minority class so that the misclassified points had more of an effect on the final model. This would begin with a model that was heavily skewed towards the majority class, with most, if not all, of the minority class data being misclassified. Then as the weights on the minority class were raised the model would approach the optimum. In our case, rather than modifying the weights we are simply adding the outlying points that may not have been addressed by the existing model while still maintaining a balanced set.

Priority Comments Retraining

For this method of retraining, the number of comments was set to the fixed 350 total comments that was chosen during the parameter optimization. Each time a run was completed the TrainValue was incremented by one if the classification did not match the training value. Then the new training set was built by randomly selecting the comments, but always taking the highest value (most misclassified) training value first.

Figure 35 clearly shows that this method did not work as expected. The average standard deviation was 13.64 which required 28.57 runs. This method did show one weakness of the system in which some comments could be misclassified even when always in the training set. The worst case was the comment "fucking brilliant" which out of 30 runs was misclassified 23 times as bullying even though it is not.

Run	Positive	Negative
1	81.02	74.72
2	37.60	81.06
3	32.96	80.68
4	35.54	90.91
5	45.47	77.36
6	41.44	88.37
7	76.12	75.68
8	32.05	87.50
9	69.23	72.41
10	40.00	88.00
11	69.12	73.91
12	76.92	71.43
13	34.72	89.47
14	61.42	77.78
15	79.12	68.75
16	83.69	68.75
17	42.47	85.71
18	68.27	76.92
19	42.35	84.62
20	79.68	61.54
21	80.90	66.67
22	71.01	72.73
23	40.74	81.82
24	80.00	70.00
25	85.23	66.67
26	42.66	88.89
27	76.09	66.67
28	40.60	87.50
29	64.06	75.00
30	74.19	75.00

Table 35. Priority Retraining

Average	95% Average	Median
1434[3,323193]	1073[23,7650.075]	586

Table 36. Dual Core Processing Stats

4.3 System Speed

Now that the accuracies of the individual methods are determined, the final step is to determain how well the parallel nature of the algorithm created functions, and what sort of throughput can be gained from each section on the hardware available. All tests were run in a virtual machine running Windows Server 2012 R2 and were run on a desktop with an Intel i7 5930k processor overclocked to 4.2 GHz, 32 GB of quad channel Crucial DDR4-2400 RAM and a Samsung 850 EVO SSD. Depending on the test, a variable number of cores and 27.9 GB of RAM were assigned to the virtual machine. For additional machines, 2 laptops were utilized, the first with an Intel i7 740QM processor clocked at 1.73 GHz and 8 GB of dual channel DDR3 RAM, and the second with an AMD A6-3400M APU clocked at 1.4 GHz and 8 GB of single channel DDR3 RAM. In total 15,000 YouTube were used for each test to ensure there was enough data for a consistent result.

4.3.1 Dual Core Speed

The purpose of this test is to establish the average speed that can be achieved by a dual core computer. A dual core is used instead of a single core because both the processing and analyzing programs were designed to create a thread for one less than the total number of cores available. This is to ensure that there remains processing power available both for the database and for the primary thread of the programs.

Average	95% Average	Median
29[16,173]	29[26,33]	30

Table 37. Dual-Core Analysis Stats

Processing Speed

The processing run on the 15,000 comments took 6 hours, 9 seconds and 200 ms to complete or 1,440 ms per comment. The standard deviation of the processing was 4,217.33 ms which equates to a coefficient of variance of 294.10%. This means a minimum of 13,291 comments were required. As seen in table 36, the average was 1,434 ms per comment but it was 1,440 ms per comment overall showing that there is some overhead processing involved between the processing of a comment and the start of the next. The median time was only 586 ms which points to a few outliers skewing the data, so the 95% confidence level was also calculated and shown in the table. This reduced the average to 1,073 ms.

Analyzing Speed

The analysis run took 8 minutes, 31 seconds and 960 ms to complete or 34 ms per comment. The standard deviation of the analysis was 2.58 ms which equates to a coefficient of variance of 8.90%. This means only 13 comments were required. As seen in table 37 the average is 29 ms while the median is 30 ms which shows how consistent the data is. Even at the 95% confidence level the average remains 29 ms.

4.3.2 Multi-Core Speed

This test will show how well the algorithm scales as the number of cores increases. As mentioned in the last test, the services are designed to create one less thread than the number of cores available in the system. For this test, the system will have a hexacore processor assigned to it in VMWare.

Average	95% Average	Median
2123[6,320740]	1591[33,11523.175]	833

Table 38. Multi-Core Processing Stats

Average	95% Average	Median
41[16,203]	40[26,80]	36

Table 39. Multi-Core Analysis Stats

Processing Speed

This processing run took 1 hour, 46 minutes, 41 seconds and 103 ms or 426 ms per comment. This is 2.38 times faster than the dual core method while using 3 times the number of cores. However, table 38 shows that the average processing time each comment takes has actually increased and both the 95% confidence level average and the median agree. However, because it is now able to handle 5 comments at the same time it actually reduces the effective average to 426 ms per comment. The standard deviation is also reduced to 5,579.81 ms which means a 262.83% coefficient of variance. This means that 10,615 comments were required.

Analyzing Speed

The analysis run took 6 minutes, 16 seconds and 823 ms to complete which is only 0.36 times faster than the dual core method. Again table 39 shows that the average and median time per comment increased while the per comment time was reduced to 25 ms. The standard deviation increased to 15.42 ms which is a coefficient of variance of 37.62%. This means only 218 comments were required.

4.3.3 Multi-Computer Speed

This final speed test is designed to show how well the system design scales when additional computers are added in. This will allow for servers to be brought online to scale the throughput required either based on expected workload or in

Average	95% Average	Median
4434[3,776183]	3351[76,23168.35]	1800

Table 40. Multi Computer Processing Stats

Average	95% Average	Median
67[16,1236]	63[30,140]	60

Table 41. Multi Computer Analysis Stats

response to a sudden increase in the frequency of comments. Note that in these tests, the two additional computers utilized are much lower power laptops to the primary machine that has been used in all other tests. So, while an increase in performance is expected, the expected increase will not be linear.

Processing Speed

This processing run took 1 hour, 15 minutes, 18 seconds and 976 ms or 301 ms per comment. This is only .42 times faster than the multi-core method while table 40 again shows an increased average time per comment. The standard deviation was 11,800.82 ms which is a coefficient of variance of 266.14% and requires 10,885 comments.

Analyzing Speed

Finally, the analysis run took 7 minutes, 22 seconds and 750 ms or 29 ms per comment. This is actually 0.18 times slower than the multi-core method. Table 41 shows that again the average and median are higher, but in this case the average over time is increased as well. The standard deviation is 47.91 ms with a coefficient of variance of 71.51% with a minimum of 786 comments required.

List of References

[1] Tableau. “Tableau software.” [Online; accessed 28-December-2016]. 2016. [Online]. Available: https://www.tableau.com/

[2] B. Efron and R. Tibshirani, "Improvements on cross-validation: The .632+ bootstrap method," *Journal of the American Statistical Association*, vol. 92, no. 438, pp. 548–560, June 1997, [Online; accessed 16-April-2017]. [Online]. Available: https://www.jstor.org/stable/2965703?seq=1#page_scan_tab_contents

[3] B. X. Wang and N. Japkowicz. "Boosting support vector machines for imbalanced data sets." [Online; accessed 15-April-2017]. 2008. [Online]. Available: http://www.csi.uottawa.ca/~nat/Papers/29-Wang.pdf

CHAPTER 5

Conclusion

5.1 Analysis of Goals

This section will analyze each of the goals to show that the research was able to meet the goal.

5.1.1 Legal Definition

While the analysis of the laws regarding cyberbullying showed them to be highly subjective and designed with juries making the final determination, the research was able to narrow it down to some simple rules that could be deployed with a minimal subjective requirement. The first test is if the comment is sexual in nature. While this is stricter than the law requires, these comments are present on sites with minors present. The second test is if the comment was intended to seriously alarm, annoy, or bother the subject. The final test is does the comment serve a legitimate purpose. This third rule is the most subjective of the rules, but is still simple enough to work for this research.

One thing the creation of this definition showed was that at this time there are no specific laws for cyberbullying on its own. Instead, it falls under the broader laws for harassment. For this reason, although the research was primarily aimed at combating the increasing cyberbullying, the end point proved successful against a much broader range of restricted speech.

5.1.2 Distinguish Cyberbullying

After optimizing all of the parameters, the optimum was found to be a training set with 350 comments, 7% of n-grams up to length 3, a knn level of 4%, and the polynomial kernel with a C of 1, a degree of 2 and a Coef0 of 1. The system was capable of identifying cyberbullying 81.8% of the time. This means even if a

human moderator has to check all of the misclassified comments manually as users flag them as incorrect, it would still drastically cut down on their workload. This will allow fewer moderators to handle an increased load of commentators without having to sacrifice the safety of the users.

Switching the moderation from matching the legal definition of cyberharrasment to a method based on a terms of service decreased the overall accuracy to 73.4% and using both methods resulted in an accuracy of 69.9%. Utilizing the legal method on the comments taken from Twitter resulted in an accuracy of 83.8%.

5.1.3 False Positives and Negatives

From the comments that were gathered off of YouTube, there were less than 10% of the comments classified as cyberbullying. This means even if the algorithm marked all comments as positive, it would have achieved a 93.4% overall accuracy. In practice, however, the algorithm generally had the negative class accuracy within 10% of the positive class due to the balanced training file.

5.1.4 Allow Retraining

After testing several different retraining strategies, it was shown that the best way to retrain is to add all of the misclassified negative comments and then balance the class with misclassified positive comments. While this will cause the training set to grow beyond the optimized 350 comments, the added time in generating the training set will be more than made up for with the increased accuracy. In testing, just 6 iterations increased the accuracy by more than 10%.

5.1.5 Speed and Parallel Operation

In 2016, Twitter averaged around 6,000 comments per second[1]. With a dual core processor and a single thread, the algorithm was only able to process 2 comments every 3 seconds. When the system was scaled up to a hexa-core processor,

that same system was able to handle 2 comments per second. Adding on 2 additional laptops brought the final speed to 3 comments per second. In total, the 3 comments per second represented approximately 18 logical processors. Some of the poor scaling in these tests is due to the three computers communicating over wifi, and everything utilizing a single ssd leading to multiple points of bottlenecking. Thus, since even popular YouTube channels can afford 36 core, 72 logical core servers for rendering[2], assuming a linear scaling puts them at handling 12 comments per second even without accounting for the raid disks and running on a host OS instead of a VM. This means that without finding additional optimization angles, it could be assumed that 500, 36 core Xeon servers may be able to handle the 6,000 comments per second. Given that as of 2010, technical presentations put Facebook as having over 60,000 servers[3], that is not out of the realm of feasibility for a company expecting to handle thousands of comments per second.

5.2 Future Work

While the scalability of the system was tested as part of the research, it was only done on enthusiast consumer grade hardware. The first test that should be performed is to test the performance of the system on a high speed dedicated Xeon system, properly setup for handling high I/O databases. This will allow for a better estimation of the number of comments per second a computer can scale to. With that, a second server should be added with a 10 GB fiber connection to see how well it continues to scale across data center hardware.

The accuracy estimations in the research are all based on the classification of a single researcher, and while it does show the method is able to accurately classify in the same method as a single moderator, it does not necessarily correlate to how well it will do in an actual system. Thus, the algorithm should be run in tandem on a large system side-by-side with a moderation team, and utilizing retraining,

test how well it can keep up with those moderators.

List of References

[1] Internet Live Stats. "Twitter usage statistics." [Online; accessed 23-February-2017]. Aug. 2013. [Online]. Available: http://www.internetlivestats.com/twitter-statistics/

[2] Linus Tech Tips. "Our 36 core video rendering server – finally explained." [Online; accessed 23-February-2017]. Oct. 2015. [Online]. Available: https://linustechtips.com/main/topic/464037-our-36-core-video-rendering-server-%E2%80%93-finally-explained/

[3] Data Center Knowledge. "The facebook data center faq (page 2)." [Online; accessed 23-February-2017]. Sept. 2016. [Online]. Available: http://www.datacenterknowledge.com/the-facebook-data-center-faq-page-2/

.1 Appendix

Example code can be found on GitHub[1] at https://github.com/danielducharme/Machine-Learning-for-the-Automated-Identification-of-Cyberbullying-and-Cyberharassment.

List of References

[1] GitHub. [Online; accessed 17-April-2017]. 2007. [Online]. Available: https://github.com/

BIBLIOGRAPHY

"Miller v. california, (1973)," June 1973, [Online; accessed 1-May-2016]. [Online]. Available: http://caselaw.findlaw.com/us-supreme-court/413/15.html

"Confidence levels and sample size." [Online; accessed 18-February-2017]. 2000. [Online]. Available: http://download.ctpp.transportation.org/training/mod_8_part1.pdf

"k-nearestneighbors." [Online; accessed 1-October-2016]. 2013. [Online]. Available: http://www.statsoft.com/textbook/k-nearest-neighbors

Aftab, Cheung, K. T.-Y. "Information theory: Information theory and the digital age." [Online; accessed 12-August-2016]. 2001. [Online]. Available: http://web.mit.edu/6.933/www/Fall2001/Shannon2.pdf

Brants, T., Popat, A. C., Xu, P., Och, F. J., and Dean, J., "Large language models in machine translation," pdf, Google, 2007. [Online]. Available: http://research.google.com/pubs/MachineTranslation.html

Broderick, R. BuzzFeed. "9 teenage suicides in the last year were linked to cyber-bullying on social network ask.fm." [Online; accessed 25-January-2014]. Sept. 2013. [Online]. Available: http://www.buzzfeed.com/ryanhatesthis/a-ninth-teenager-since-last-september-has-committed-suicide

Carter, T. "An introduction to information theory and entropy." [Online; accessed 12-August-2016]. 2011. [Online]. Available: http://astarte.csustan.edu/~tom/SFI-CSSS/info-theory/info-lec.pdf

CBC News. "Cyberbullying-linked suicides rising, study says." [Online; accessed 25-January-2014]. Oct. 2012. [Online]. Available: http://www.cbc.ca/news/technology/cyberbullying-linked-suicides-rising-study-says-1.1213435

Chang, C.-C. and Lin, C.-J. "Libsvm – a library for support vector machines." [Online; accessed 12-September-2012]. Apr. 2012. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm/

Chih-Chung Chang, C.-J. L., "Training nu-support vector classifiers: Theory and algorithms," *Neural Computation*, vol. 13, pp. 2119–2147, 2001, [Online; accessed 11-June-2016]. [Online]. Available: http://ntur.lib.ntu.edu.tw/bitstream/246246/155217/1/09.pdf

Chih-Wei Hsu, C.-C. C. and Lin, C.-J. "A practical guide to support vector classification." [Online; accessed 3-January-2016]. Apr. 2010. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

Costa, L., "Voting neighbours: Svm border selection algorithm based on knn," [PhD Thesis, to appear].

Data Center Knowledge. "The facebook data center faq (page 2)." [Online; accessed 23-February-2017]. Sept. 2016. [Online]. Available: http://www.datacenterknowledge.com/the-facebook-data-center-faq-page-2/

Department of Justice. "Citizen's guide to u.s. federal law on obscenity." [Online; accessed 20-October-2015]. July 2015. [Online]. Available: http://www.justice.gov/criminal-ceos/citizens-guide-us-federal-law-obscenity

Dredze, M., "Machine learning finding patterns in the world," pdf, Johns Hopkins University, 2009. [Online]. Available: http://old-site.clsp.jhu.edu/workshops/ws09/documents/machine-learning-overview.pdf

Efron, B. and Tibshirani, R., "Improvements on cross-validation: The .632+ bootstrap method," *Journal of the American Statistical Association*, vol. 92, no. 438, pp. 548–560, June 1997, [Online; accessed 16-April-2017]. [Online]. Available: https://www.jstor.org/stable/2965703?seq=1#page_scan_tab_contents

Erhan, C. "Libsvmsharp." [Online; accessed 15-September-2015]. Apr. 2015. [Online]. Available: https://github.com/ccerhan/LibSVMsharp

GitHub. [Online; accessed 17-April-2017]. 2007. [Online]. Available: https://github.com/

Google. "Google translate." [Online; accessed 20-October-2015]. 2015. [Online]. Available: https://translate.google.com/

Green, L. "Mean, mode, median, and standard deviation." [Online; accessed 18-February-2017]. 2008. [Online]. Available: https://www.ltcconline.net/greenl/courses/201/descstat/mean.htm

Hamel, L., *Knowledge Discovery with Support Vector Machines*. 111 River St, Hoboken, New Jersey 07030: John Wiley & Sons Inc., 2009.

Hodges, A. The Alan Turing Internet Scrapbook. "The turing test, 1950." [Online; accessed 20-October-2015]. 1997. [Online]. Available: http://www.turing.org.uk/scrapbook/test.html

Hsuan-Tien Lin, C.-J. L. "A study on sigmoid kernels for svm and the training of non-psd kernels by sm o-type methods." [Online; accessed 14-May-2016]. 2003. [Online]. Available: www.csie.ntu.edu.tw/~htlin/paper/doc/tanh.pdf

Hu, J. "History of machine learning." [Online; accessed 6-April-2014]. Apr. 2013. [Online]. Available: http://www.aboutdm.com/2013/04/history-of-machine-learning.html

III, M. L. R. "Basic concepts of statistics - class 23." [Online; accessed 1-October-2016]. 2010. [Online]. Available: http://www.unc.edu/~rls/s151-2010/class23.pdf

Internet Live Stats. "Twitter usage statistics." [Online; accessed 23-February-2017]. Aug. 2013. [Online]. Available: http://www.internetlivestats.com/twitter-statistics/

Jill F. DeVoe, Sarah Kaffenberger, K. C., "Student reports of bullying. results from the 2001 school crime supplement to the national crime victimization survey." National Center for Education Statistics, Tech. Rep., 2005, [Online; accessed 7-February-2016]. [Online]. Available: http://nces.ed.gov/pubs2005/2005310.pdf

José Pinheiro Neves, L. d. O. P., "Cyberbullying: A sociological approach," *International Journal of Technoethics*, vol. 1, pp. 24–34, 2010, [Online; accessed 2-January-2016]. [Online]. Available: http://www.scribd.com/doc/134607212/Cyberbullying-a-Sociological-Approach#

Jurafsky, D. and Martin, J. H., *Speech and Language Processing*. Upper Saddle River, New Jersey 07458: Pearson Education, Inc., 2009.

Kamen, M. Wired. "Twitch now blocks trolls and hatespeech in real-time." [Online; accessed 3-February-2017]. Dec. 2016. [Online]. Available: http://www.wired.co.uk/article/twitch-introduces-anti-troll-automod-for-game-streams

Laird, E. M., "The internet and the fall of the miller obscenity standard: Reexamining the problem of applying local community standards in light of a recent circuit split," *Santa Clara Law Review*, vol. 52, 2012, [Online; accessed 1-May-2016]. [Online]. Available: http://digitalcommons.law.scu.edu/cgi/viewcontent.cgi?article=2735&context=lawreview

Legal Information Institute. "U.s. code." [Online; accessed 20-October-2015]. [Online]. Available: https://www.law.cornell.edu/uscode/text

Legal Information Institute. "U.s. code: Title 18 - crimes and criminal procedure." [Online; accessed 20-October-2015]. Oct. 1970. [Online]. Available: https://www.law.cornell.edu/uscode/text/18

Legal Information Institute. "18 u.s. code § 1470 - transfer of obscene material to minors." [Online; accessed 20-October-2015]. Oct. 1998. [Online]. Available: https://www.law.cornell.edu/uscode/text/18/1470

Legal Information Institute. "18 u.s. code § 2266 - definitions." [Online; accessed 20-October-2015]. Aug. 2006. [Online]. Available: https://www.law.cornell.edu/uscode/text/18/2266

Legal Information Institute. "18 u.s. code § 1514 - civil action to restrain harassment of a victim or witness." [Online; accessed 20-October-2015]. Dec. 2012. [Online]. Available: https://www.law.cornell.edu/uscode/text/18/1514

Legal Information Institute. "18 u.s. code § 2261a - stalking." [Online; accessed 20-October-2015]. Mar. 2013. [Online]. Available: https://www.law.cornell.edu/uscode/text/18/2261A

Legal Information Institute. "47 u.s. code § 223 - obscene or harassing telephone calls in the district of columbia or in interstate or foreign communications." [Online; accessed 20-October-2015]. Mar. 2013. [Online]. Available: https://www.law.cornell.edu/uscode/text/47/223

Legal Information Institute. "U.s. code: Title 47 - telecommunications." [Online; accessed 20-October-2015]. Mar. 2013. [Online]. Available: https://www.law.cornell.edu/uscode/text/47

Linus Tech Tips. "Our 36 core video rendering server – finally explained." [Online; accessed 23-February-2017]. Oct. 2015. [Online]. Available: https://linustechtips.com/main/topic/464037-our-36-core-video-rendering-server-%E2%80%93-finally-explained/

Nuance. "Dragon speech recognition software." [Online; accessed 20-October-2015]. 2015. [Online]. Available: http://www.nuance.com/dragon/index.htm

Patchin, J. W. Cyberbullying Research Center. "Cyberbullying research: 2013 update." [Online; accessed 2-January-2014]. Nov. 2013. [Online]. Available: http://cyberbullying.us/cyberbullying-research-2013-update/

Patterson, S. CNBC. "Man vs. machine: Seven major players in high-frequency trading." [Online; accessed 6-April-2014]. Sept. 2010. [Online]. Available: http://www.cnbc.com/id/39038892

Pope, T. P. New York Times. "Parents often unaware of cyber bullying." [Online; accessed 25-January-2014]. Oct. 2008. [Online]. Available: http://well.blogs.nytimes.com/2008/10/03/parents-often-unaware-of-cyber-bullying/?_php=true&_type=blogs&_r=0

Reddit. "Reddit content policy." [Online; accessed 14-December-2015]. 2015. [Online]. Available: https://www.reddit.com/help/contentpolicy

Riot Games, Inc. "Exploring player behavior design values." [Online; accessed 20-October-2015]. Nov. 2014. [Online]. Available: http://na.leagueoflegends.com/en/news/game-updates/player-behavior/exploring-player-behavior-design-values

Riot Games, Inc. "Instant feedback powers up." [Online; accessed 20-October-2015]. Sept. 2015. [Online]. Available: http://na.leagueoflegends.com/en/news/game-updates/player-behavior/instant-feedback-powers

Riot Games, Inc. "New player reform system heads into testing." [Online; accessed 20-October-2015]. May 2015. [Online]. Available: http://na.leagueoflegends.com/en/news/game-updates/player-behavior/new-player-reform-system-heads-testing

Shapiro, L. "Information gain." [Online; accessed 25-January-2016]. 2010. [Online]. Available: https://courses.cs.washington.edu/courses/cse455/10au/notes/InfoGain.pdf

Silva, Y. N. Arizona State University. "Bullyblocker: Towards the identification of cyberbullying in facebook." [Online; accessed 2-January-2014]. [Online]. Available: http://www.public.asu.edu/~ynsilva/BullyBlocker/index.html

Souza, C. "Accord.net framework." [Online; accessed 14-May-2016]. 2012. [Online]. Available: http://accord-framework.net

Stanglin, D. and Welch, W. M. USA Today. "Two girls arrested on bullying charges after suicide." [Online; accessed 25-January-2014]. Oct. 2013. [Online]. Available: http://www.usatoday.com/story/news/nation/2013/10/15/florida-bullying-arrest-lakeland-suicide/2986079/

State of Rhode Island. "Title 11 criminal offenses." [Online; accessed 20-October-2015]. 1992. [Online]. Available: http://webserver.rilin.state.ri.us/Statutes/TITLE11/INDEX.HTM

State of Rhode Island. "Title 11 criminal offenses, chapter 11-42 threats and extortion, section 11-42-2 extortion and blackmail." [Online; accessed 20-October-2015]. 1992. [Online]. Available: http://webserver.rilin.state.ri.us/Statutes/TITLE11/11-42/11-42-2.HTM

State of Rhode Island. "Title 11 criminal offenses, chapter 11-52 computer crime, section 11-52-4.2 cyberstalking and cyberharassment prohibited." [Online; accessed 20-October-2015]. 1992. [Online]. Available: http://webserver.rilin.state.ri.us/Statutes/title11/11-52/11-52-4.2.htm

State of Rhode Island. "Title 11 criminal offenses, chapter 11-59 stalking, section 11-59-1 definitions." [Online; accessed 20-October-2015]. 2002. [Online]. Available: http://webserver.rilin.state.ri.us/Statutes/TITLE11/11-59/11-59-1.HTM

State of Rhode Island. "Title 11 criminal offenses, chapter 11-59 stalking, section 11-59-2 stalking prohibited." [Online; accessed 20-October-2015]. 2002. [Online]. Available: http://webserver.rilin.state.ri.us/Statutes/TITLE11/11-59/11-59-2.HTM

State of Rhode Island. "State of rhode island general laws." [Online; accessed 20-October-2015]. Dec. 2014. [Online]. Available: http://webserver.rilin.state.ri.us/Statutes/

Tableau. "Tableau software." [Online; accessed 28-December-2016]. 2016. [Online]. Available: https://www.tableau.com/

Twitch Interactive. "How to use automod." [Online; accessed 26-December-2016]. Dec. 2016. [Online]. Available: https://help.twitch.tv/customer/portal/articles/2662186-how-to-use-automod

Twitter. "Twitter." [Online; accessed 15-September-2015]. Jan. 2015. [Online]. Available: https://twitter.com/

Visell, Y., "Lecture 12: Decision trees," pdf, McGill University, 2006. [Online]. Available: www.cim.mcgill.ca/~yon/ai/lectures/lec12.pdf

Wang, B. X. and Japkowicz, N. "Boosting support vector machines for imbalanced data sets." [Online; accessed 15-April-2017]. 2008. [Online]. Available: http://www.csi.uottawa.ca/~nat/Papers/29-Wang.pdf

Warner, W. and Hirschberg, J., "Detecting hate speech on the world wide web," pdf, Columbia University, June 2012, [Online; accessed 25-May-2014]. [Online]. Available: http://aclweb.org/anthology//W/W12/W12-2103.pdf

Wikipedia, The Free Encyclopedia. "League of legends." [Online; accessed 20-October-2015]. Oct. 2015. [Online]. Available: https://en.wikipedia.org/wiki/League_of_Legends

YouTube. "Popular right now." [Online; accessed 20-October-2015]. 2015. [Online]. Available: https://www.youtube.com/playlist?list=PLrEnWoR732-BHrPp_Pm8_VleD68f9s14-