CHALMERS
UNIVERSITY OF TECHNOLOGY

Developing Social Media Analytics by the Means of Machine Learning:

The Case of the Diffusion of Virtual Reality Technology

Master's Thesis in the Master's Programme
Management and Economics of Innovations

ADAM BERTHOLD
DANIEL LARSSON

Department of Technology Management and Economics
Division of Science, Technology and Society
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2017
Report No. E 2017:134

MASTER'S THESIS E 2017:134

Developing Social Media Analytics by the Means of Machine Learning:

The Case of the Diffusion of Virtual Reality Technology

ADAM BERTHOLD
DANIEL LARSSON

Tutor: CHRISTIAN SANDSTRÖM

Department of Technology Management and Economics
Division of Science, Technology and Society
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2017

Developing Social Media Analytics by the Means of Machine Learning:
The Case of the Diffusion of Virtual Reality Technology
ADAM BERTHOLD
DANIEL LARSSON

Master's Thesis E 2017: 134

Department of Technology Management and Economics
Division of Science, Technology and Society
Chalmers University of Technology
SE-412 96 Gothenburg, Sweden
Telephone: + 46 (0)31-772 1000

Chalmers Reproservice
Gothenburg, Sweden 2017

Abstract

Social media analytics is concerned with analyzing data generated from social media platforms. It is commonly used within businesses to gain insights in order to improve decision-making. Social media analytics is also used within research, notably innovation research. Using social media data within research often entails reading large amounts of text posts. Since social media datasets can quickly become very large, there is a demand for computerized methods to replace manual analysis. This research is concerned with exploring ways to use machine learning within innovation research to replace manual analysis of social media data.

This study applies machine learning on a case study concerned with the diffusion of virtual reality technology. Virtual reality technology is a technology which has created much online buzz lately. However, sales have been much lower than expected. As such, the case is concerned with why virtual reality technology is not more popular on the market. The case-study will use a social media analytics approach complemented with a method utilizing machine learning. The study makes use of several different researchers' theories on diffusion of innovation to analyze the case.

The dataset used in the case study consists of approximately 6000 public text posts written in Swedish on Facebook, Twitter, forums and other platforms. The dataset was collected between August 2016 and August 2017. To investigate the barriers to diffusion, while also being able to apply machine learning algorithms, the posts are categorized into four different categories based on the topic of the post. The categories, "Technological Utility", "Network Externalities", "Price" and "Trialability" are derived from theories of diffusion of innovation. Also, some posts are marked as "Spam" and not taken into account during analysis.

The categorization is done manually as well as through a machine learning algorithm. The machine learning program used is based on the SVM classifier, which is a supervised binary classifier. The hyperparameter "C" is set to 1.2 and the N-gram to 1. The evaluation metrics used are accuracy and AUROC. Using k-fold cross validation on the dataset, these evaluation metrics reach about 85 % and about 0.8 respectively. Comparing the results of the machine learning categorization and the manual categorization reveals that these evaluation metrics are too low for a practical use in research since it has the potential to significantly change the outcome of the study.

The conclusion of the study is that in order to use machine learning in innovation research, the performance of algorithms needs to be very high, which is hard to do with the classifiers used in the study. In the future more complex algorithms or using different methods for feature selection should be explored. Furthermore, a larger dataset would naturally induce higher performance, and allow for other types of algorithms. The case study however suggests that a small testing set could be useful to apply on Big Data contexts where manual analysis is not feasible, but such a method would always compromise accuracy for time saving.

The study concludes that perhaps the most feasible way to use machine learning in social media analytics innovation research would be either as a filter to pick out data to analyze, or in a temporal analysis that is interested in trends over time. The case concludes that the price of the virtual reality headsets is too high for the majority of customers, which hinders the diffusion of the technology.

Acknowledgments

The students conducting this master's thesis would like to extend a thank you to the supervisor Christian Sandström for his valuable insights during the course of conducting the study. Also, friend and machine learning programmer Henrik Marklund is thanked for his feedback and help during the development of machine learning algorithms.

Adam Berthold
Daniel Larsson
January, 2018

Table of Contents

1. Introduction

Social Media Analytics (SMA) is a field of study that has emerged in the wake of the mass-adoption of social media. Social media has not only been adopted by the general public, but has also attracted businesses, political parties and other actors (Al-Deen & Hendricks, 2011). For many organizations, social media is a promotional tool used for communicating with stakeholders (Moe & Schweidel, 2017), but what characterizes social media is the user's ability to contribute in a conversation as opposed to simply reading the content of others (Ganis & Kohirkar, 2016). Therefore, organizations can use social media as a window into the customers' minds to gain market insights and crowdsource ideas (Moe & Schweidel, 2017).

According to Stieglitz et al. (2014), the primary goal of SMA is "to develop and evaluate scientific methods as well as technical frameworks and software tools for tracking, modeling, analyzing, and mining large-scale social media data for various purposes". These various purposes are often related to research studying public opinion such as public administration, politics, consumer decision-making etc. (Stieglitz et al., 2014). For example, SMA has been applied within innovation research (Laurell & Sandström, 2016; Laurell & Sandström 2017) where behavior and opinions can play a large role.

Today, online information is produced and stored at an unprecedented rate; a dataset collected from social media platforms Facebook, Twitter or Tumblr can quickly amount to thousands of text-based posts, making analysis very time-consuming (Lewis et al., 2013). Such datasets, where the size makes data unmanageable, is commonly referred to as Big Data (Boyd & Crawford, 2012). Traditionally, an analysis of the data entails that the researchers read and categorize each post individually. As such, in order to capture the potential value in social media data, the field of SMA needs ways to shorten the analysis time. Lewis et al. (2013) argue that the massive content flows of social media require computational methods to enable a proper analysis. However, they emphasize that a proper analysis of social media data also needs the systematic rigor and contextual awareness of manual content analysis, creating something of a paradox.

Historically, computational content analysis has relied on what Sebastiani (2002) refers to as knowledge engineering, where data is processed by a manually designed filter. Within text-analysis, such a filter could be designed to identify all text-strings containing positive words such as "great", "yes", "happy" etc. and negative words such as "no", "fail", "sad" etc. to create a sentiment analysis. However, such computational methods often fail to capture the complexity of human language. Instead, machine learning can be a useful tool.

Machine learning is an area within computer science which is popular within text-analysis. Machine learning is a method developed to teach computers to perform tasks that they are not specifically programmed to do. Machine learning programs use more sophisticated patterns to categorize data than traditional computer algorithms do, effectively making a kind of automated reading possible. Properly designed and applied machine learning algorithms process data more reliably and faster than humans, making it possible to analyze ever larger and more complex datasets.

Purpose 1: There seems to be an opportunity to address the shortcomings of SMA by applying machine learning in the field. Thus, *the purpose of this study is to explore ways to further develop the field of SMA within innovation research by adding machine learning, especially in regards to shortening the analysis time.*

To address this purpose, which is concerned with innovation research, and to contextualize the results of the study, the research uses SMA and machine learning in a case study assessing the diffusion of Virtual Reality technology.

The case is used to create a setting in which different types of machine learning options can be experimented with in order to adequately explore how machine learning is best used. The case is also used to simulate a real SMA-based innovation research where the current method of manual reading and categorizing of the data can be compared to the categorization done by a machine learning algorithm. The comparison compares the different case conclusions from each method.

1.1 Case Study

The diffusion of innovations is a research field concerned with how innovations spread in populations. Virtual reality (VR) technology is a medium designed to present media content in an as immersive way as possible. Recently, VR-technology developers have made use of motion tracking to create hand held controls to allow the user to interact with objects in the virtual reality (Oculus, 2016; Vive, 2017). This feature creates opportunities to create novel gaming-experiences. Consequently, technology companies Facebook and HTC are currently developing high-end VR-technology for the gaming industry through their headset-projects Oculus Rift and HTC Vive respectively.

However, even though VR-technology is much anticipated and heavily invested in (Gleasure and Feller, 2016), the sales numbers are so far disappointing. In the first quarter of 2017 HTC Vive reported 95.000 units sold worldwide and Oculus Rift reported 64.000 units sold worldwide (Grubb, 2017). Given that Facebook bought Oculus Rift for $2 billion in 2014 (Gleasure and Feller, 2016), and that the price of a headsets is about $500, these sales are underwhelming. Thus, this research is interested in investigating why the novel and immersive VR-technology is not attracting more users. To do so the research is utilizing SMA and machine learning. Due to limitations in the data gathering process, the research only uses Swedish social media for the analysis.

Purpose 2: Thus, *the purpose of the case study is to study discussions on social media in Sweden to assess the most significant barriers to the diffusion of VR-technology within the gaming community.* The **research question** for the case is: *"What is the most significant barrier to the diffusion of VR-technology?"*

Summing up, this study has two purposes, one (the case purpose) is used in order to fulfil the other. In order to address both these purposes the research is using SMA to analyze a dataset consisting of public social media posts about VR gathered from Swedish social media. The SMA-method is complemented by the use of machine learning algorithms to analyze the posts in the data set. The research uses literature related to the diffusion of innovation and technology ecosystem emergence.

1.2 Virtual reality technology

Virtual Reality-technology is designed to present the senses with a computer generated three-dimensional environment that can be explored and interacted with to some degree (vrs.org.uk, 2017). The illusion is created by alluding to many of the senses, including balance, vision, hearing, and feel. Traditional ways to present media are also designed to immerse the user, but these technologies do not strive to create an ever improving illusion of another reality. That is the ambition of VR-technology (vrs.org.uk, 2017).

In practice, the illusion of VR is created by presenting visual data on screens within a headset that completely obscures the user's vision (Fig 1.1). Headphones present the viewer with sound from the virtual reality he is experiencing in the headset. VR-headsets are equipped with gyroscope technology that tracks head movement to create an illusion of looking around inside the virtual reality. The high-end headsets make use of motion and touch controls to enable more interaction with the virtual reality (Oculus, 2016; Vive, 2017).

Figure 1.1 - HTC Vive and Oculus rift in use.

Today, Facebook and HTC invest heavily into developing their, relatively, advanced headsets Oculus Rift and HTC Vive respectively. Both of the headsets require a high-end personal computer to function. HTC Vive is the more expensive of the two, selling at around $600, while Oculus is sold at around $400 (Oculus, 2016; Vive, 2017). Each of the headsets have some 48-49 % market share of computer based VR headsets (Steam, 2017). Games are often developed for both headsets, although there are also exclusive titles.

While the gaming industry has shown the most significant interest in VR-technology, the technology creates many new possibilities for innovation within other industries. Engineering and design heavy industries (Amend, 2016), healthcare (Lee, 2017), and the defense industry (ClassVR, 2017) are examples of industries that have started to develop their businesses using VR. For example, IKEA has developed an application allowing customers to design and inspect kitchens in VR (IKEA, 2017) and often car manufacturers offer a VR-feature for test sitting and personalizing car models in VR before a purchase (BMW, 2017)

A cheap way to create a VR-experience is utilizing the smartphone as the screen and hardware for the VR-headset. Accordingly, Samsung has developed a headset with a slot for a smartphones. Based on the same principles, Google has developed a cardboard case with two lenses that sells for about $5 - $10. As such, the VR-market is still uncertain and there is no clear dominant design of the products.

2. Theory

The following chapter explains the theories that used for designing the framework used to analyze the case study. Thus, the chapter mainly investigates different views on technology diffusion and technology ecosystems.

2.1 Diffusion of technology

When a new technology enters the market there is hope that it will be adopted by the consumers on that market. That is, the producer is hopeful that the customers who are targeted will start using the product. The process of innovation adoption within a population is famously dubbed the *diffusion of innovation*, a term popularized by Everett M. Rogers. His book Diffusion of Innovation (2003) is dedicated to the most popular and widespread model of how a new technology is adopted by a population.

To Rogers (2003), the diffusion of an innovation is a social process where the channels of interaction between people become important. According to Rogers (2003), mass media is an example of such a channel, and personal contact with a friend is another. The more personal the contact is the more powerful it is in creating a decision to adopt an innovation. Aggregated, many such adoption decisions constitute the diffusion of innovations. Put differently, information about a new technology is spread in a social system via the interaction between people. According to Rogers (2003), the concept of diffusion involves the individual's decision to adopt the technology. The decision is made in five steps that together make up the innovation-decision process. The steps are:

1. **Knowledge** - the individual learns of the innovation
2. **Persuasion** - the individual forms an opinion about the innovation
3. **Decision** - the individual decides whether to adopt or reject the adoption
4. **Implementation** - the individual starts to use the innovation
5. **Confirmation** - the individual seeks information about the value of the adoption-decision

The innovation-decision process described above is to a large extent a process of uncertainty reduction. Rogers (2003) describes five characteristics of an innovation to reduce uncertainty. Thus, if increased, the five characteristics also increase the rate of adoption. The five characteristics are:

1. **Relative advantage** - the degree to which an innovation is perceived as being better than the idea it supersedes. Example: A toy that is more *fun* to use than another toy have a high degree of relative advantage.
2. **Compatibility** - the degree to which an innovation is perceived as consistent with the existing values, past experiences, and needs of potential adopters. Example: When the bar code reader was offered on the Italian market, the digit screen would only fit six digits. The reader was designed for prices expressed in dollars that seldom amounted to more than six digits. As such, it was not compatible with the Italian lire, which was much less valuable than the dollar and prices were often expressed in more than six digits. The bar code reader had a low degree of compatibility with the Italian market (Rogers, 2003).
3. **Simplicity** - the degree to which an innovation is perceived as relatively difficult to understand and use. Example: A new tool with less buttons than existing solutions have a high degree of simplicity.
4. **Trialability** - the degree to which an innovation may be experimented with on a limited basis. Example: A new car can often be taken for a test drive before a purchase. As such, it has a high degree of trialability.
5. **Observability** - the degree to which the results of an innovation are visible to others. Example: A home alarm system keep burglars away. The benefit is not easily detected by observers since they are not living in the house and since the potential burglaries do not happen, causing no event of interest to the observer.

Rogers (2003) classifies the members of social systems into categories based on their innovativeness (Fig, 2.1). He includes five types:

1. **Innovators** are the members of the social system most willing to experiment with new ideas. Innovators are interested with new technology and adopt innovations with high uncertainty, not expecting high rewards. As such, they can be seen as oddballs whose opinions are not taken very seriously by the other members of the social system. Rogers (2003) describes the innovators as venturesome. The first 2.5 % to adopt an innovation are the innovators.
2. **Early adopters** are innovative members of the social system who are more integrated into the system than the innovators. The early adopters hold a role of respect in the community, often asked for advice on new technology. Due to their opinion leadership, early adopters feel the need to adopt innovations early and to form judicious opinions to spread in the social system. Rogers (2003) describes the early adopters as well respected. The 13.5 % of the population to adopt an innovation after the innovators are the early adopters.
3. **Early majority** are individuals who are earlier than the average member. They lack the leadership role of the early adopters. Rogers (2003) describes the early majority as deliberate decision-makers. The 34 % of the population to adopt an innovation after the early adopters are the early majority.
4. **Late majority** members are very similar to early majority members, the late majority is neither first nor last to adopt. They practice a more skeptical approach to the innovation. Rogers (2003) describes the late majority as skeptical. The next 34 % of people to adopt an innovation after the early majority are the late majority.
5. **Laggards** are members who are the most traditional and least interested in change and innovation. Rogers (2003) describes the laggards as traditional. The last 16 % to adopt an innovation are the laggards.

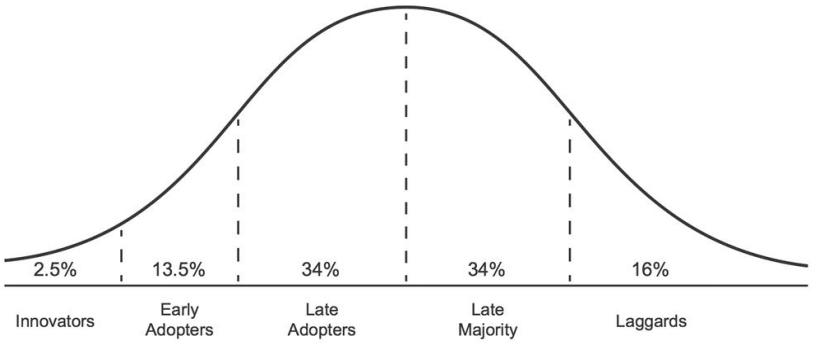


Figure 2.1. The distribution of adopter types (Rogers, 2003)

Rogers' (2003) theories on the diffusion of innovation serve as the basis for many studies, many of which are brought up and discussed in his book. The researchers behind such studies often need to alter Rogers' (2003) frameworks to fit the inherent and unique aspects of the particular innovation of interest. For example, one such study complemented the five characteristics of innovation with five new dimensions that better suited the innovation, totaling to ten characteristics. These new perspectives and complements to Rogers' (2003) theories draw inspiration both from the technology itself and from literature with close relation to the phenomenon of diffusion.

2.2 Ecosystem innovation

Rogers' (2003) approach to diffusion lacks some perspectives of technology that become important when studying more complex technology systems. Rogers' (2003) theories are looking at somewhat one dimensional products, where the product is pushed out on a market and based on its inherent characteristics the members of the social system decides whether to adopt or reject it. Innovations can however be the subject of *network effects* or have a *platform* structured business model that creates an innovation ecosystem. That is, some innovation's value can be expected to increase as other members the social system choose to adopt the innovation (Arthur, 1996). Platforms also create lock-in effects for adopters (Arthur, 1996), creating a general hesitation on the market. Furthermore, a platform-structured business models targets a two-sided market, requiring adoption from two different social systems, innovators and users, that have some degree of interrelation (Magnusson & Nilsson, 2014). The following segments describe theories especially focused on such platform aspects of an innovation.

In their 2015 study of technology changes in semiconductor lithography equipment industry, Ron Adner and Rahul Kapoor develop a framework for how entire *innovation ecosystems* substitute other technology ecosystems. In the study they address the question of why some technology ecosystems are quickly diffused and adopted in a social system while other technology ecosystems are diffused slower or not at all.

Adner and Kapoor (2015) focus on technology ecosystems. They define these as products that constitute a *focal technology* (i.e. an electric vehicle) that is accompanied by *complementary products* that increase performance or utility of the technology (i.e. loading stations for electric vehicles). As such, innovation ecosystems do not only substitute an existing solution to a problem, but also carry a bunch of opportunities that attracts developers to create complementary products for the new focal technology.

The new technology ecosystem benefits if the customers have access to complementary products that add value to the ecosystem. In some cases this access to complements is necessary for diffusion to occur. As such, even if the focal technology is fully developed and functioning, a low attraction for third party developers can slow down the rate of adoption. Such barriers, which slow down the rate of adoption, are by Adner and Kapoor (2015) called *emergence challenges*. These are often challenges inherent in the ecosystem that are required to be solved in order for *emergence* to occur.

Technology S-curves (Fig. 2.2) are often used to describe the diffusion of innovation (Schilling, 2010). The description of an S-shaped performance pattern starts with the technology having a slow development as it struggles to find adopters and thus a reason for further development. The initial development pace is further slowed down by a lack of knowledge of the technology by its developers. As time and effort progress so does the performance of the technology. Eventually it reaches maturity, often bound by physical constraints. Under the threat of substitution, the performance of an incumbent technology can increase significantly, when developers put in extra effort to match the competition of the new technology, a phenomenon popularized as the Sailing Ship Effect (Ward, 1967).

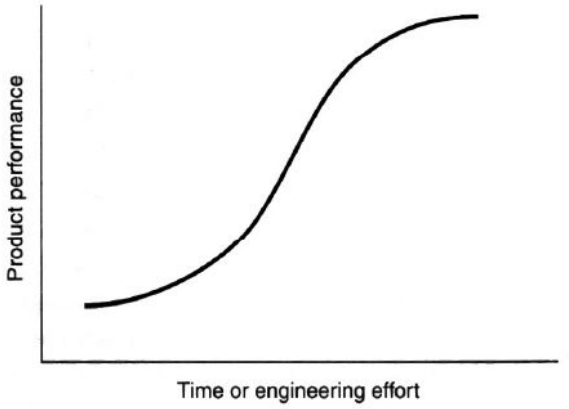


Figure 2.2. The technology S-curve.

Adner and Kapoor (2015) draw inspiration from the technology S-curve and the sailing ship effect, describing an *extension opportunity of the old technology*. These are the opportunities to further develop and improve performance of the old technology ecosystem. Put differently, the eventual success of a new technology ecosystem does not only depend on the performance of the new technology ecosystem itself, but also on whether the old technology ecosystem has opportunities to improve.

Thus, Adner and Kapoor (2015) develop a framework taking into account the two dimensions *the extension opportunities of the old technology* and *the emergence challenges of the new technology*. The dimensions are used to synthesize a framework consisting of four quadrants, and thus four types of technology ecosystem substitution.

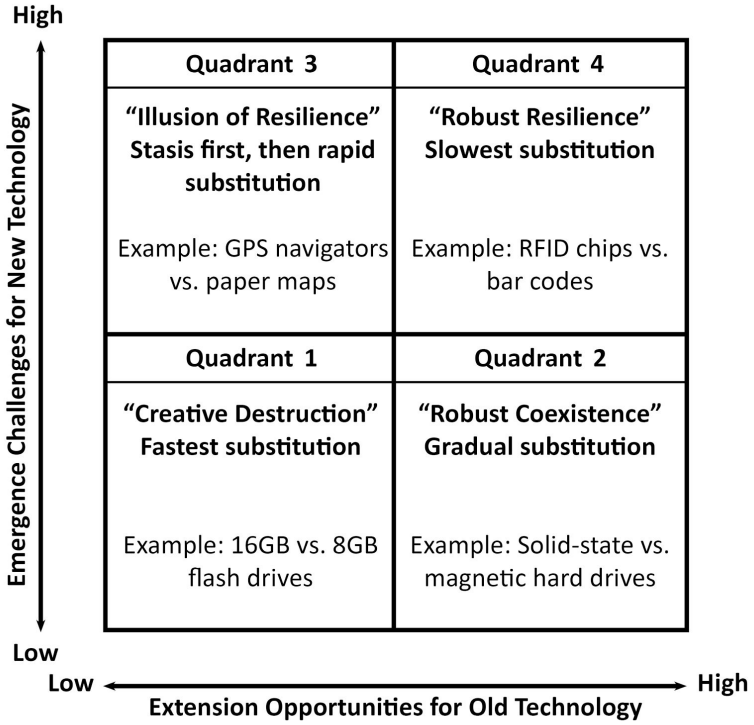


Figure 2.3. Adner's and Kapoor's framework

1. In quadrant one (Fig. 2.3), the emergence challenges of the new technology are low, and the extension opportunities of the old technology are also low. Under these circumstances technology substitution is swift and unrelenting, categorized as creative destruction by Adner and Kapoor (2015).
2. In quadrant two (Fig. 2.3), the new technology is mature enough to be used by the market. The market is however invested in the new technology which is projected to be further developed in the near future. This cause the two technologies to be able to coexist on the market and the pace of substitution gradual.
3. In quadrant three (Fig. 2.3), the low extension opportunities of the incumbent technology make the actors on the market interested in the new technology. However, high inherent challenges must be solved before general adoption can occur, causing a period of stasis before a rapid substitution.
4. In quadrant four (Fig. 2.3), the slowest type of substitution occurs. The extension opportunities of the incumbent technology are high, while the emergence challenges of the new one also are high, resulting in a market that while being aware of the new technology still prefers the old one.

In order to perform an investigation of the diffusion of an new technology ecosystem, Adner and Kapoor (2015) argue that researchers must find data that explain the level of extension opportunities the old technology have, as well as the level of emergence challenges the new technology have. Viewing technology ecosystems as platforms that operates on a two-sided market (Magnusson & Nilsson, 2014), with one side towards developers of complements and one side towards technology users, one type of emergence challenge that can exist is how the user-side of the market value the innovation. Ultimately, their evaluation of the technology makes up their decision to adopt the innovation. As such, it is important to investigate and understand how a technology ecosystem is valued by the customers before a purchase.

2.3 Multiple Dimensions of Value of Technology

Schilling (2010) describe two main dimensions along which a *technology network* can be valued (Fig. 2.4). First, it can be valued on its *stand-alone value.* This value is based on the *technological utility* the innovation bring the user. Examples of parameters upon which a user might evaluate the stand-alone value of a technology are how *fun* it is to use, how *simple* it is or what *image* it brings. As such, the functionality of the technology to the user forms the basis of this type of value.

Second, a new technology can be valued on its *network externalities' value.* That is, the value of an innovation depends on its *installed base* and the access to *complementary goods*. The installed base is the amount of other customers that are using the technology. This value stems from the network effects phenomena described by Arthur (1996). Arthur (1996) describes how a large installed base increase the likelihood of products and services to be developed for the platform. For example, the value of an Android phone is increased when other customers also choose to use Android, since this increases the probability of more applications becoming available. Also, a large installed base is likely to increase the pace of improvement of the focal technology since the developers will put more effort into the technology. As such, a large installed base is likely to increase the access to complementary products. This access heighten the value of a product, the so called network externalities' value.

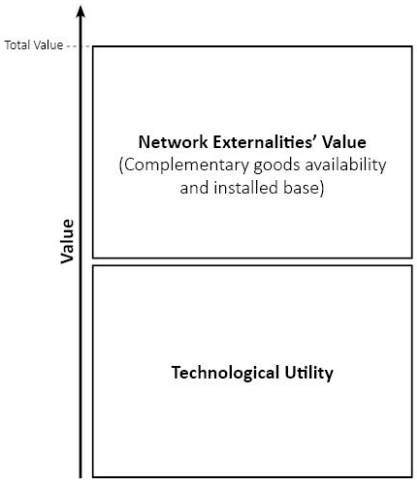


Figure 2.4. The dimensions of value (Schilling, 2010)

Both types of network externalities' values stem from the network created by several users of the technology. As such, network externality value mostly concerns technology ecosystems or platforms, where interaction with the users and interaction with the developers add on to the end-performance of the technology (Magnusson & Nilsson, 2014). The resulting logic is that it is not always enough that a new technology has a great stand-alone value. In order to diffuse in a population, the new technology's value need to exceed the combined value of technological utility, the installed base and the availability of complementary goods of the old technology (Fig. 2.5).

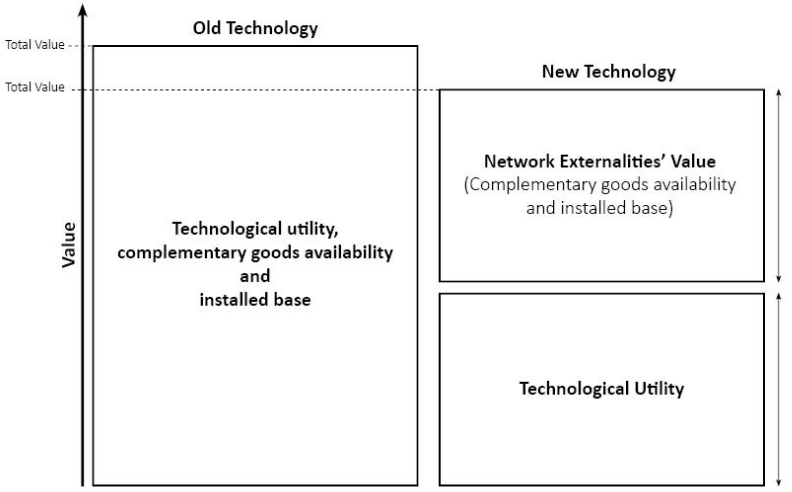


Figure 2.5. New technology's value compared to old. (Schilling, 2010)

Schilling's (2010) discussion takes the end-user's view of the value of the technology. The total value of the technology to the customer is the sum of the value attributed to the stand-alone product and the value attributed to the network externalities. In order to make a purchase decision however, the customer will compare the perceived total value of the technology with the price tag. This comparison is done in the phase Rogers (2003) call the *decision phase.* Put differently, the customers will in this phase try to determine if the cost of the product is less than the perceived total value it brings.

As such, Schilling (2010) describes how it can be important for a developer to increase the perceived value rather than the actual value of the product. In such a strategy, it is common for the developer to exaggerate the value of its network externalities by portraying themselves as having a larger installed base than they really do have. The customers then perceive the total value as higher than the actual value is, leading to a purchase decision. If the tactic is successful, the result is that the installed base indeed becomes large, creating little or no backlash of the tactic.

2.4 Platforms

Magnusson and Nilsson (2014) study technology platforms, which closely resembles the view of technology ecosystems that Adner and Kapoor (2015) take, as well as the view of technology as a network taken by Schilling (2010). Magnusson and Nilsson describe technologies that create two-sided markets, where the technology function as a platform that need to attract and serve a *customer-side* of the market, as well as it needs to attend to the needs of an *innovator-side* of the market. The innovators create applications for the platform, increasing its value to the customer.

The value of the platform is thus created through an interaction between the three stakeholders: the platform owner, the innovators and the customers. The platform owner offer the technological utility, while the innovators offer the complementary products for the platform. For example, a well-functioning governance and maintenance of the platform, as well as a large installed base of customers, create an environment and opportunities that attract innovators to create applications for the platform. Also, well-handled platform governance by the platform owners combined with a large set of quality applications created by innovators will attract more customers. Thus, the platform owners are interested in attracting both customers and innovators to the platform in order for the other side of the market to also grow.

It is notoriously complex to develop price-models for platforms (Magnusson & Nilsson, 2014). The two-sided market generate revenues from both the innovators and the users. Magnusson describe how platform owner identify a *money side* and a *subsidy side*, where the subsidy side is attracted to the platform by the money side. As such, the owner can put a lower price on the subsidy side of the market, and a higher on the money side to compensate. This tactic is dependent on some kind of lock-in from on the money side that lowers that side's price sensitivity. That is, if there is only one platform on the market, or if there are high switching-costs, for the money side, the actors of that side will be locked to the platform and thus less price sensitive.

2.5 Diffusion of VR-technology

The following chapter combine aspects of technology platforms and innovation ecosystems with theories of diffusion and unique aspects of VR-technology to formulate a theory base to address the case's purpose and research question.

2.5.1 VR-technology as an Ecosystem

Technology ecosystems and platforms create value based on the technology itself and the network externalities created through attracting innovators or the access to complementary products it enables. Applying this theory on gaming targeted VR-technology (HTC Vive and Oculus Rift), part of the value of VR for users is based on the quality and performance of the headsets themselves, while part of the value is stemming from the access to games. Thus, the focal technology and complementary products, as described by Adner and Kapoor (2015), would be the headsets and the games developed for the headsets respectively.

This distinction is however not clear-cut. Adner and Kapoor (2015) define complementary products as "the products that users integrate with the focal technology". In doing so, the technology ecosystem based on the electric vehicle is used as an example, where the complementary products are charging stations. This type of complementary product is thus a somewhat necessary complement for the technology to function properly. In the case of VR-technology, such a necessary complement is closer resembling the role of the personal computer, which is the power source for the headset. Schilling (2010) instead takes a view of complementary products as something desirable for the users to access, much resembling applications developed for a platform. In VR-technology such complementary applications would be the games, which are accessed complements. This study makes use of social media data, and as such the opinions in social media is studied, which are opinions from the user-side of the market. Thus, the value of the network externalities, as it is described by Schilling (2010), are stemming from the accessed complements, the games, and not from the necessary complements. Instead, in the perspective of the users (and this study), the value created by necessary complements are part of the value that is stemming from the focal technology. Put simply, in the eyes of the user, the performance of the headset and personal computer combined form the stand-alone value.

Since VR-technology developers are targeting the consumers within the gaming community, the view of this study is that the VR-technology ecosystem is substituting parts of the PC-gaming industry ecosystem. Put in Adner's and Kapoor's (2015) terminology, the PC-gaming industry is in this study viewed as the *old technology* while VR-technology is viewed as the *new technology*. Since VR-headsets offer a more immersive type of display, the display of each ecosystem is viewed as the focal technology. This means that the focal technology of the old technology is the desktop mounted flat screens. However, the two technologies offer quite different experiences for the user, and the adopters might not want to completely substitute one for the other. At a basic level however, the two technologies are competing over the consumer's time, which further motivates the view that VR-technology is a substitution for flat-screen displays as a medium.

2.5.2 Opportunities and Challenges

In taking the view of traditional computer game industry as the old technology, and the VR-technology as the new, Adner's and Kapoor's (2015) framework can be applied to study the diffusion of VR-technology. Thus, in order to understand the diffusion process of VR, the extension opportunities of the old technology, as well as the emergence challenges of the new technology, becomes important to study.

Extension Opportunities of the Old Technology

The PC-gaming industry is a steadily growing industry. The number of active players is huge and the market size and opportunities are increasing yearly (Appendix A). Also, the industry is experiencing changes in business model practice and market structure, showing few signs of maturing. As such, many developers, investors, users and manufacturers are invested in the future of the traditional gaming industry. Collectively, the stakeholders of the industry are seeking to further the technology, the game mechanics and the business models of PC-gaming (Stuart, 2016). Thus, both in regards to market opportunities and in regards to technological opportunities, the incumbent PC-gaming technology ecosystem show signs of having a *high* amount of extension opportunities. Therefore, this research is interested in investigating the emergence challenges of the new technology when assessing the most significant barriers to the diffusion of VR-technology through a social media analysis.

Emergence Challenges of the New Technology

In order to assess the emergence challenges of VR-technology through studying social media data, the challenges are connected to the user-side, or the market demand, of the platform. In assessing the challenges through a social media analysis, the research makes use of Schilling's (2010) two dimensions of value, the stand-alone value and

the network externalities' value, of a new technology. Consequently, customers could either value the focal technology, that is, the performance of the headset, or the network externalities created by owning the headset, that is the access to games. Thus, when evaluating the emergence challenges of the new technology this study is analyzing social media data and assess which dimension of value, the stand-alone value or the network externalities' value, that is currently in most demand for the customers.

2.5.3 Total Value and Price

In the context of this study it is interesting to evaluate the social media user's view of Schilling's (2010) two dimensions of value, However, it is also interesting to assess to what extent the customer find the *total value* of the technology match the price. That is, end-users might evaluate both of the two dimensions equally high while the total value might not match the price.

A VR-headset for PC-gaming cost above the equivalent of $400 in Sweden, which can be a significant sum for a normal household. Also, since the experience of using a VR-headset is quite unique, and the technology is used in the confines of the user's home, the observability can be regarded as low. These two facts combined can be a barrier to diffusion since the user cannot discern how it might benefit from the technology while regarding it as expensive. This would effectively constitute a barrier to diffusion in of itself.

Thus, to increase the rate of adoption VR-developers might benefit from increasing the trialability of the products. That is, the diffusion of VR-technology can be increased if customers can properly evaluate the technology before a purchase decision is to be made. Due to the above discussion the research also investigates how much the price of the headsets and opportunities to try the headsets are discussed among consumers on social media.

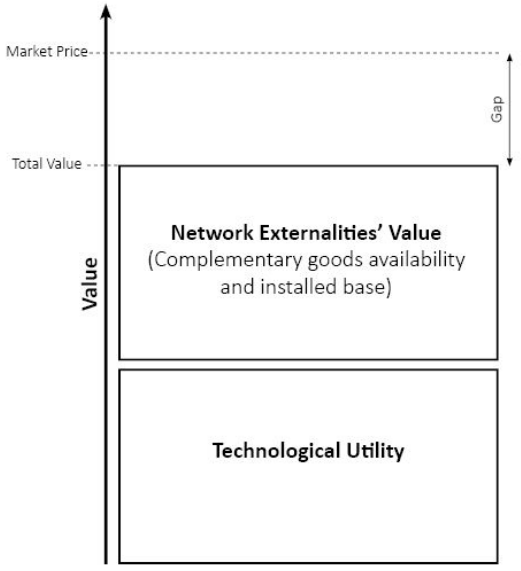


Figure 2.6. A gap between the value and market price. (Schilling, 2010)

2.6 Summary

To summarize, this chapter concludes that in order to study the barriers to diffusion of VR-technology the research uses theories of diffusion and technology systems. Adner's and Kapoor's (2015) framework is used in a combination with Schilling's (2010) dimensions of value in order to investigate emergence challenges and review which challenge is the most significant barrier. The three barriers to diffusion that are studied and compared are the *technological utility*, the *network externalities*' value and *the gap* between the market price and the total value.

3. Method

This chapter is concerned with both creating a basic understanding of SMA and machine learning, as well as explaining the process of developing and applying a machine learning program and how the case is analyzed. As the case in this study is used as an example of an innovation research study upon which machine learning is experimented with, the focus and effort of this study is put more into fulfilling the first purpose. The case study is also used as an evaluation method for this study, where results from the machine learning analysis and a manual content analysis is compared.

The chapter starts with theory on case methodology and SMA. Then the chapter explains how the algorithm of this study is developed and which different aspects of the development the study is concerned with exploring. Also, the method of analysis of the case is explained and the chapter finishes with a critical analysis of the research quality of the study.

3.1 Case methodology

According to Easterby-Smith et al. (2015), cases are in research mostly used to look in depth at organizations, events or individuals. However, they stress that the purposes of cases can be very varied and unique to a single study. Although the emergence of VR-technology could be seen as an event, the purpose of including this case could also relate to what Yin (2014) describe as archival analysis. In this study, using the case aims to explore a how the semantic context within social media can be refined into valuable insights. From these insights, the study aims to explore how machine learning can be applied in other areas within innovation research using SMA. The case is also used as a way to contextualize and highlight how the performance of the machine learning program actually affect the final outcome of the research compared to a manual analysis. Specifically, the case is thus used to show the researchers and readers how much, for example, 80 % accuracy of the machine learning algorithm affect the outcome of the case analysis. Thus, the case has two unconventional goals, to act as an example upon which to explore machine learning solutions and as a context to help understand different metrics.

The study is using a mixed method, as proposed by Easterby-Smith et al. (2015), combining qualitative and quantitative elements. Analyzing the dataset relies on traditional qualitative content analysis to represent the process researchers regularly use. This process is what the algorithms are meant to imitate, learning from the researcher's qualitative judgement to simulate this process digitally. The evaluation of algorithms and change in input data is instead assessed quantitatively by comparing performance metrics. The field of machine learning could take either an inductive or deductive approach, as described by Bryman & Bell (2011), where this study aims to utilize both. For applying the theoretical framework in the case study, the study is using a deductive approach as data has been categorized based on the theoretical framework. On the other hand, this study is using an inductive approach for generating insights about the utility of machine learning within the context of innovation research, and more specifically content analysis within this field.

3.2 Social Media Analytics

The field of SMA is concerned with insights gathered from social media. As such, it is important to understand how social media is used. Some features differ between different social media platforms. For instance, Twitter-posts often contain less words than Facebook-posts. Twitter is also used differently; Twitter's "retweet"-function is to an extent used differently than Facebook's "share"-function. These differences are important to understand before extracting information from social media (Weller, 2015).

Brooker et al. (2016) emphasize that social media data often contain combinations of text, links, images, videos and other media. This fact pressures researchers to adequately render the data and assess what topics such data can address. For established platforms such as Facebook, Twitter and Instagram, researchers have studied how to interpret features such as retweets or check-ins, and such understanding constitutes the basis for many research questions of existing studies (Weller, 2015). Karpf (2012) further elaborates on the flaws of online data within research, arguing that the huge amounts of public data are hard to adequately analyze. Also, social media data often contain a much larger proportion of noise compared to proprietary data, which also contribute to the challenges of analyzing social media data.

Viewing social media as a lens to observe the aggregated public dialog implies the identification of which users are being monitored. Amachai-Hamburger & Vinitzky (2010) find that personality traits correlate with Facebook usage, and argue that other factors, such as social norms, influence individuals' social media participation as well. However, Hughes et al., (2011) conclude that while personalities and motives differ between users, it does not affect the individual's choice of social media platform; instead, the most popular platforms (Facebook, Twitter) are used by most users, but with different intentions. Within this study, platforms are treated equally, but will have different user groups in mind when conducting the analysis.

Brooker et al. (2016) distinguish two main strategies for data analysis within SMA: temporal analysis and corpus analysis. Within temporal analysis the researcher studies narratives over a period of time, to see how language and topics develop. Corpus analysis instead focus on the dataset as a collection of posts from which different topics can be derived.

3.3 Machine Learning

The following chapter explains the design of the machine learning algorithm used within the study. As the field is widely explored, previous research serves as a foundation for design choices. However, the unique traits of the dataset, combined with the research question of this study leaves some choices to be explored by testing. This testing is part of the "exploration" within the research purpose: to explore ways to further develop the field of SMA.

Machine learning algorithms are computerized algorithms that use statistical tools to learn from examples, so called training data, in order to predict new data (Stieglitz et al., 2014). This allows the algorithm to better capture complex data patterns, such as the nuanced meanings of human language (van Zoonen & van der Meer, 2016). Machine learning has seen usage in a wide range of applications, including text classification, recommender systems, and spam filtering (Kumar et al., 2016). When developing machine learning algorithms, the programmer needs to consider both what properties the input data have, and how the output data should be structured to address the research's purpose. Also, the quality of the output data is dependent on the properties of the input data. Karpf (2012) refers to this issue as "garbage in, garbage out", where input of lacking quality never creates good output. For example, the size of the dataset limits the options of categorizing the data into many categories. Thus, there is a triangular relationship between the input data, the output data and the algorithm, which makes the program development a complex and iterative process (Fig. 3.1).

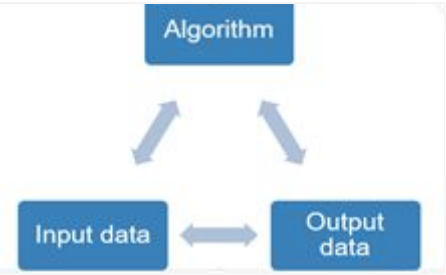


Figure. 3.1. The algorithm, the input and the output data all depend on each other.

3.3.1 Data collection and Dataset

The dataset used in this study was collected using a tool called Notified developed for collecting social media data (www.notified.com). Notified has collected a dataset that contains Swedish public social media posts from platforms such as Twitter, Facebook and Instagram as well as from forums, blogs, and video-sites. The dataset was collected between 2016-08-18 and 2017-08-23 and all posts are made within that time span. The data was collected based on the two market leaders on the VR-PC-gaming market, Oculus Rift and HTC Vive, together constituting 96-98 % of the VR-PC-gaming market (Steam, 2017). As such, each post in the dataset contains either or both of the phrases "Oculus Rift" and "HTC Vive".

The dataset does not contain any statistical metrics such as the number of retweets or number of likes. Instead, the data consists of the extracted text content of each post, also ignoring media such as images, links and videos. This is much due to limitations in the data collection method. As the study aims to identify patterns in the aggregated dataset (corpus analysis), rather than assessing trends over time (temporal analysis), the research disregards the timestamps of the posts. Furthermore, the research question does not incorporate an analysis of the author of each post or make a distinction between which platforms the post is collected from. The result is that the data in the dataset is of text-based nature.

All posts in the dataset are short; more than 85 % of posts contain less than 1000 characters, the longest containing 23058 characters. Also, the dataset contains posts from a wide range of platforms. The distribution between these platforms is presented in Table 3.1.

Table 3.1. Dataset distribution between platforms.

Blog	*Facebook*	*Forum*	*Instagram*	*Twitter*	*Video*
569	*276*	*2398*	*341*	*2381*	*79*

3.3.2 Output Data

The machine learning algorithm used in this study is supposed to emulate a manual content analysis of the data. That is, it should read each post in the dataset and determine something about the content of the particular post. As such, the output data of the algorithm will in this study be an updated dataset where each post is given a label based on what the meaning of the post is.

In order to study the two dimensions of value presented in Chapter 2, as well as the gap between the total value and the market price, the algorithm is designed to label each post based on if it contains information about either of the two dimensions of value - the stand-alone value or the network externalities' value - or the price-value gap. The price-value gap can be discussed in two ways, either the customers are interested in the price tag itself, and as such

discussing the price, or they are hesitant and discussing ways to try the headsets before a purchase. Thus, the algorithm is designed to interpret the content of each post and give it one or more of the four labels:

1. **Technological Utility**- discussions about the stand-alone value of the headsets.
2. **Network Externalities** - discussions about the network externalities' value, that is, the installed base of each headset type or the access to games or other externalities.
3. **Price** - discussions about the price of the headsets or games.
4. **Trialability** - discussion about the extent to which the headsets can be tried before a purchase.

The labeled dataset shows which of the three discussion topics that are the most popular. In order to address the purpose of finding the most significant barrier to diffusion in the dataset, the study sees the amount of discussion of each topic as demand for that topic. This interpretation of amount of discussion is based on that there are three types of discussion-contributing posts that are given labels:

1. There are posts that **express demand**. Example:
   "*I want there to be more games before I purchase*".
2. There are posts that tell of an experience that **create a certain demand**. Example:
   "*I used HTC Vive and quickly got sea-sick*." or "*I bought the Oculus Rift and realized that there are not many games*."
3. There are posts that **directly respond to a demand**. Example:
   "*A new game is released to HTC Vive tomorrow*".

Since the Notified software collects all posts containing the two tags: "HTC Vive" and "Oculus Rift", many posts that are not discussing any of the four topics are also saved. Such posts can be divided into two categories, posts about VR-technology and posts that are not about VR-technology. The study and the algorithm also label the posts that are not about VR-technology as "Spam", while the posts that are about VR-technology, while not being about either of the categories are not given any label at all. Thus, the posts marked as spam can be deleted easily, cleaning the dataset for further analysis.

3.3.3 Algorithm Design

Based on the properties of the data and the theoretical framework, the machine learning algorithm is designed to read text-messages and label the posts based on the text data included in each post. This information forms the basis for how the algorithm will be designed. The program used in the study is based on the scikit-learn toolkit (scikit-learn.org). Scikit is a Python-based open-source library for machine learning development containing tools for both data processing and machine learning. The full source code for this study is found in Appendix C. As the posts to be analyzed are in Swedish, the content is parsed as strings in utf-8 coding.

Machine Learning Methods

Machine learning algorithms are commonly divided into two categories: *supervised* and *unsupervised* learning algorithms (Christensen et al., 2017). Supervised learning algorithms base predictions on data with a predefined set of categories, labelled data. Unsupervised learning is instead based on unlabelled data, where the researcher allows the algorithm to freely identify patterns in the observed data, from which theories can be generated. This study is aimed at making the machine emulate a manual labeling of the data and as such the algorithm of this study is a supervised algorithm.

The predictions, or the output, of machine learning algorithms can either be discrete or continuous, which distinguishes the fields of classification and regression, respectively (Theodoridis, 2015). The purpose of classification algorithms is to predict labels of data. Regression algorithms are instead designed to predict continuous numerical values. For example, classification algorithms can be used to categorize texts into different classes using labels, while regression algorithms can be used to investigate the reach of tweets. Chapmann (2017) notes that classification algorithms are more popular as discrete predictions, such as yes or no, tend to be more useful in decision-making. Since the data of this study is text-based and the framework is designed to divide the data into discrete categories, a machine learning classifier, or a classifier, is used.

Har-Peled et al. (2003) make the distinction between *binary* and *multiclass* classification algorithms. Binary classifiers assign entities to one of two classes, while multiclass classifiers are able to assign any value from a predefined set (Herrera et al., 2016). For example, a binary classifier could determine whether a post is from Twitter or not, whereas a multiclass classifier would determine if a post is from either Twitter, Facebook, or Instagram. Another type of classification is called *multilabel* classification. In this type of classification one entity can be assigned several labels. For example, a text message could be about two different topics. Multilabel classification is often confused with multiclass classification, where the entities only are assigned one class each. As the dataset of this study contain many posts that discuss more than one of the subjects, single posts can receive multiple labels. As such, the study makes use of a multilabel classification logic. However, in order to increase transparency and understanding, the study makes use of a binary classifier. The supervised binary classifier is ran several times, once for every label. This is also how many multilabel algorithms function.

Jebara (2004) highlights another distinction between algorithms: generative and discriminative models. Generative models use probabilistic methods to determine probability distributions within data. Discriminative methods instead only focus on mapping input data to desired outputs, not taking probability distributions into account. While Jebara (2004) notes that discriminative models are more efficient in regards to computational power, they lack the elegant probabilistic concept and structure of generative models. If the researcher has no interest in the underlying statistical methods, either of these models could be used. Figure 3.2 aims to conceptualize the difference between discriminative and generative models.

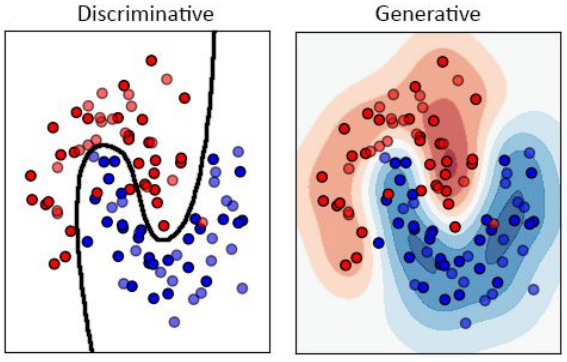


Figure 3.2. Discriminative and generative models

To conclude, the classifiers used in this study are all supervised classifiers. Supervised classifiers are trained using so called *training sets*, which are manually constructed by the researchers. Specifically, the researcher will read a portion of the posts in the dataset and label it according to the four labels, as well as the Spam-label, to create a showcase that the algorithm will use to find patterns in the data. To structure this process, and get more comparable results, there are methods for splitting the dataset known as cross validation (*cross-validation,* 2016). K-fold is a common cross validation method, where the training and testing sets are assigned systematically, iterating over the

whole dataset, each iteration testing a new fraction of the dataset until the whole dataset has been used. This is visualized in Figure 3.3, where four tests (iterations) are made.

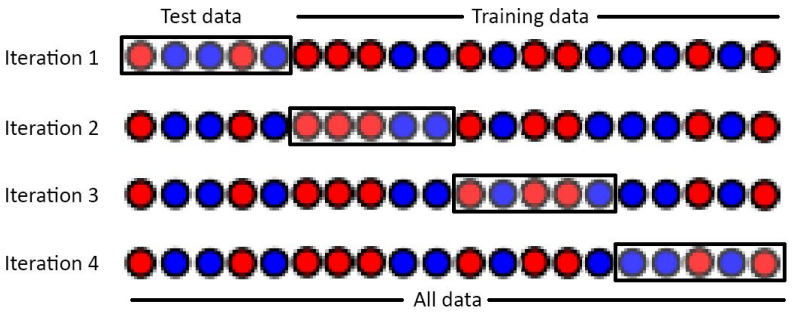


Figure 3.3. K-fold cross validation where k = 4

There are many types of different supervised classifiers, and in order to pick one suitable for the purpose of the particular study, the researcher needs to consider several points. The difference between them is the logic of how to find the pattern and make the divide between different labels. In Figure 3.4 several different algorithms are used to divide the same dataset, consisting of data that are plotted in a two-dimensional space and labeled either blue or red. This pattern is saved by the machine and used to classify new data, as conceptualized in the example of Figure 3.5.

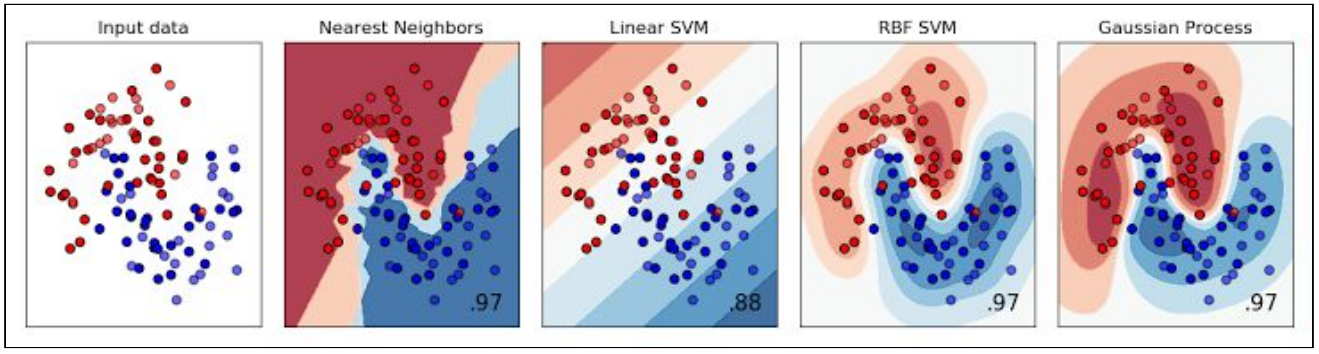


Figure 3.4. Visualization of how four different classifiers find patterns in the same data.

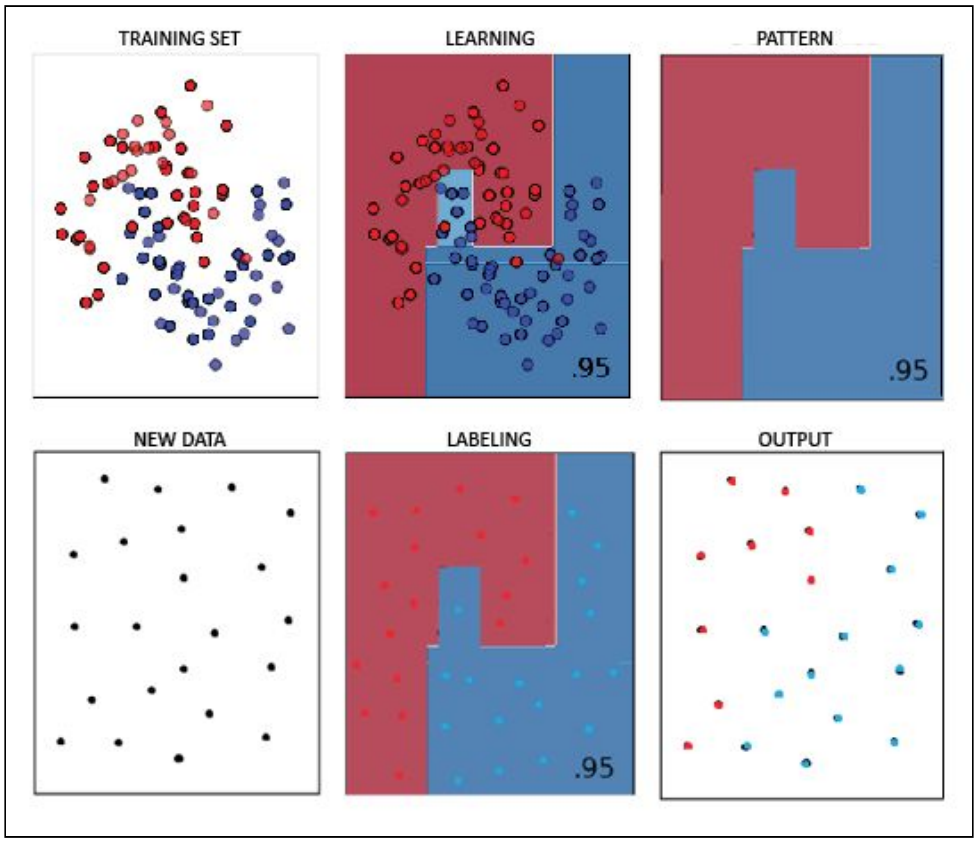


Figure 3.5. Visualization of how the classifier Decision Tree creates a pattern and then use it to classify new data.

Data refinement

Before choosing the adequate learning algorithm the programmer must understand certain properties of the input data. All properties within the input data, such as text length or occurrence of specific words, is referred to as *features* (Herrera et al. 2016). Thus, features are the properties from which an algorithm finds patterns for making predictions. All combined features of a dataset is known as a *feature space*, and the number of features within that feature space determine the *dimensionality* of the feature space. Generally, and especially for smaller datasets, lower dimensionality is preferred, as increased dimensionality requires large amounts of data to make reliable predictions. This phenomenon is often referred to as the 'curse of dimensionality' (Marimont & Shapiro, 1979). As noted by Christensen et al. (2016), high dimensionality is a major issue within text analysis

As machine learning classifiers cannot interpret non-numerical data, such as texts, the researcher must identify ways to transform the data into meaningful units (Sebastiani, 2002). A common approach is to count the occurrence of different words within a text document, known as the bag-of-words approach (van Zoonen, 2016). Bengio et al. (2003) note that there is an exponential increase in dimensionality when taking word combinations into account, as each new word create a new dimension. This issue has resulted in the emergence of more complex word representations than counting term frequency, referred to as word embedding. To represent words word embedding use dense vectors inspired by neural networks language modeling (Levy & Goldberg, 2014). However, since bag-of-words is the most straight-forward and common approach, it is used in this study.

To do this transformation, the program developed for this study will use the scikit TfidfVectorizer. This tool converts the total dictionary of the training set into a normalized vector of word occurrences. To reduce the implications of the curse of dimensionality there are methods for weighing terms within a dictionary, the most common being the Term Frequency: Inverse Document Frequency (TF-IDF) weighting scheme. The TF-IDF

scheme inverses word frequency to give higher importance to less frequent occurring words. Figure 3.6 shows how simple text strings are transformed into normalized vectors using the TfidfTransformer. As seen in the Figure 3.6, 'alpha' with a word index of (0, 0) within this vector, is represented more often than remaining terms.

```
['alpha alpha beta train']      ['alpha alpha beta test']
(0, 0)    0.816496580928        (0, 0)    0.894427191
(0, 1)    0.408248290464        (0, 1)    0.4472135955
(0, 2)    0.408248290464
```

Figure 3.6. The left string is the training set and the right the predicted set. The vectorization of the strings is seen below the strings, where each word is given an index. In the first string, alpha is given (0, 0). The numbers represent a normalization of word occurrence and importance of the word, in this case inverse word frequency. The means by which the machine performs this calculation is not presented. Note how the word 'test' in the second string has no representation. This is due to the word not existing in the training data, resulting in the word not being recognized.

Thus, each post is transformed into a vector that can be plotted in a multi-dimensional space. Simply put, the algorithm will look for patterns of labels among the labelled vectors, and use that pattern to predict the label of new vectors (posts). Thus, each new word in the training set increase the dimensionality of the vector space the algorithm will analyze, quickly giving rise to the curse of dimensionality. Two common tools to reduce the feature space within text mining are *word stemming* and *stopword removal*, both of which require word processing tools containing dictionaries for the desired languages (which exist for Swedish). During word stemming, each word is reduced to its stem, such as 'running' being transformed into its stem, 'run' (Sebastiani, 2002). Stopword removal refers to a process where low-information words, such as 'it', 'for' and 'in', are removed entirely from the feature space. Also, removing special characters such as punctuation marks reduces the risk of words being represented twice which reduces the risk of an unnecessarily large feature space.

Classifiers

As each new word or term in the dataset increases the dataset's dimensionality, text mining requires classification techniques that perform well when dimensionality is high (Christensen et al., 2016). Christensen et al. (2016) summarize ten articles ranking performance of classifiers for text mining, and results show that the algorithm Support Vector Machines (SVM) is the dominating technique for high-dimensional datasets. Other common algorithms within the same study were Naive Bayes classifiers. The characteristics of these classifiers, are explained below. Van Zoonen & van der Meer (2016) study the categorization of *short* social media messages and they find that SVMs and Naive Bayes classifiers outperform logistic regression algorithms. All of these classifiers were based on supervised learning.

Naive Bayes is one of the most common algorithms within text classification, designed for classification problems. Naive Bayes is a generative model, described by Chapmann (2017) as a probabilistic multiclass classifier based on the assumption that features are mutually independent. Mutually independent features implies that there is no relationship between input data. For example, the occurrence frequency of the word 'machine', and the word 'learning' are treated separately, assuming there is no connection between these terms. Friedman et al. (1997), among others, argue that this assumption is too unrealistic as there is often redundancy among features. Lee & Lee (2004) argue that this holds especially within text analysis where there is high dimensionality in the feature space. However, Garreta & Moncecchi (2013) argue that this simplification grants Naive Bayes large practical usage. Due to the reduced computational requirements Naive Bayes becomes efficient and scalable compared to more complex

learning algorithms. The probabilistic distribution used by Naive Bayes can be determined by the researcher, where multinomial Naive Bayes assumes multinomially distributed data, as opposed to other Naive Bayes classifiers which use a different logic. For example gaussian Naive Bayes which assumes a gaussian distribution (scikit-learn.org).

Logistic regression is another popular technique in the statistics community (Jebara, 2004). Logistic regression is a discriminative model with binary outputs, designed for classification tasks (Chapmann, 2017). Logistic regression has proven useful within text classification, as it is similar to Naive Bayes but built on a discriminative method. Menardi & Torelli (2014) notes that logistic regression is not suitable when classes are unbalanced, underestimating the probabilities of the rare class. Logistic regression is however not assuming independent features like the Naive Bayes algorithms.

As seen in the study by Christensen et al. (2016), SVMs are classifiers proven to perform well within the area of text analysis. As logistic regression, SVMs are discriminative models. Chapmann (2017) describe SVMs as a complex algorithm resulting in fine-tuned models. He further notes that the complex calculations of SVM are sensitive to large datasets, and usage should be limited to smaller dataset. Like the Naive Bayes method, there are different alternatives for which SVM to use. Changing the kernel of the algorithm changes the internal logic for data prediction.

To conclude, the study will investigate which one of these classifiers that works best in the program used in this case. Thus, the research will be concerned with fine-tuning the three algorithms to best performance and then be able to compare them. Recently much research within the field of machine learning has focused on algorithms referred to as neural networks. As the name suggests, neural networks do not rely on statistical tools. Instead, neural networks arrange entities into nodes similarly to neurons within the human brain (Haykin, 2009). This allows for analysis of much more complex data structures than possible with a linear regression classifier. These more complex algorithms are deemed to be outside the scope of this study.

Evaluation

When evaluating the performance of an algorithm, the programmers use a test set that is manually categorized. The algorithm is thus trained by using the training set, and then tested through letting the algorithm categorize the test set. There is however no single metric for evaluating performance of machine learning algorithms. Algorithms have to be evaluated based on the specific context in which it is used. The performance of an applied algorithm, such as a recommender system, could be measured by number of clicks on recommended entities, while an algorithm for cancer detection could be measured by how many cases are detected. Testing an algorithm is important to evaluate its performance before applying it in a real life situation. Also, if results of predictions are binary, there are plenty of statistical tools for evaluating the performance of an algorithm. One of the most common and simple metrics of an algorithm's performance is *accuracy*. The accuracy metric simply computes the number of correctly labeled entities divided by the total number of entities.

Two other common performance metrics, popular within binary classification, are precision and recall. These metrics are based on what is known as a confusion matrix (Fawcett, 2006), seen in Figure 3.7. A confusion matrix assumes a binary relation between positives and negatives and the possibility of the classification of the data to be wrong. In such a situation, *precision* is a metric that measures how many of the identified positives actually belong to that class. On the other hand, *recall* is a metric for how many of the total positives were identified by the classifier (Davis & Goadrich, 2006.

	Actual Positive	Actual Negative
Predicted Positive	True Positive	False Positive
Predicted Negative	False Negative	True Negative

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Figure 3.7. Confusion matrix and Precision and Recall formulas

A common tool for assessing how well an algorithm is performing with regards to both precision and recall is to compute a Receiver Operating Characteristic (ROC) curve combining the two metrics (Fawcett, 2006). The area under the ROC curve (AUROC) can then be computed into another single digit accuracy metric. Fawcett (2005) argues that ROC is especially useful for situations when one of the classes is significantly more common than the others. However, the ROC of a classifier is harder to visualize and conceptualize than precision and recall scores. The AUROC is defined as a number between 0 and 1, where 1 is optimal performance. In this study, both AUROC and accuracy is measured in order to explore which of the three classifiers Naive Bayes, SVM and logistic regression that is best applied in this innovation research.

Algorithm tuning

Much of machine learning programming is about picking the right algorithm for the data and context. However, the next step is choosing the right parameters for the algorithm to increase its performance. When training supervised algorithms to find patterns in training data, classifiers tend to account for the noise in the data, finding patterns which are not generalizable within the context. This phenomenon is referred to as *overfitting* (Dietterich, 1995). Put simply, overfitted algorithms takes too detailed information into account, missing out on trends in the large picture, and the created pattern from which the algorithm will make predictions become less useful.

There is also the opposite phenomenon, *underfitting*, where algorithms simplify the model too much, not taking important aspects into account when making predictions. Because of these phenomena, practitioners must find a balance where predictions are as most accurate and general. Much of the work put into developing a good machine learning program is put into "tuning" the algorithm to find the best way to divide the particular dataset. Figure 3.8 aims to visualize how classifiers should find the right balance between underfitting and overfitting.

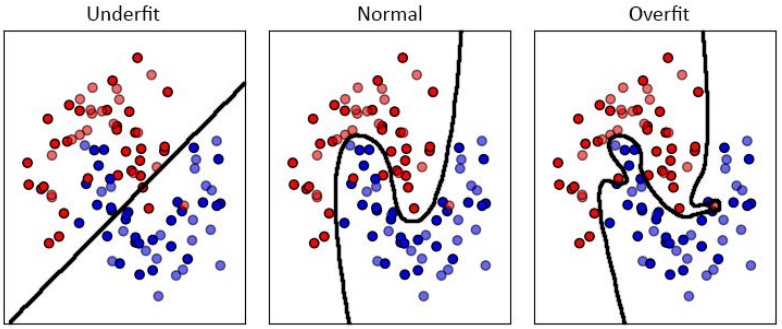


Figure 3.8. The pictures show the concept of overfitting and overfitting. The machine learning algorithms will create a general discrete division pattern between the binary data. The same pattern is used on the unclassified new data to predict labels. If the pattern is too specific to the training set, it becomes overfitted, and cannot accurately classify new unclassified data. The reverse is true for underfitting.

In order to address the research purpose of exploring ways to apply machine learning in innovation research, the best possible algorithm will be developed. The study intends to experiment with the algorithm, tuning different input conditions to yield the highest accuracy. There is no right or wrong method to such *parameter tuning* according to Chapmann (2017). However, a common method for such algorithm tuning is a method called *grid search*, a method which is standard practice within machine learning according to Snock & Larochelle (2012). When performing a grid search, the input parameters are systematically adjusted and the algorithm accuracy systematically evaluated. The grid search result is finding the optimal parameter tuning for the context of the study.

Below follows explanations of the different input conditions that will be tuned. The input conditions to be tuned are: hyperparameters, N-grams, Stopword removal, training set size.

Hyperparameters. Most machine learning algorithms involve variable input parameter, so called *hyperparameters*, which affect the algorithm's performance. For example, the classifier Naive Bayes is accompanied with hyperparameter 'alpha' that determine the degree of generalization for the algorithm. Thus, changing the value of alpha may result in either underfitting or overfitting. The values of the hyperparameters are determined by the practitioner, who can use them to yield better predictions. The study thus change and tune the hyperparameters of the Naive Bayes, SVM and Logistic regression classifiers in order to find the best possible results for the specific research context and data of the case.

N-grams. In this text analysis study, the bag-of-word method is used. Thus, the algorithm will count the number of each term in each post. Using single words as terms is referred to as unigrams, and combining multiple words into terms is known as *n-grams* (Christensen et al. 2017). N-grams are used to take word interactions into account to gather more nuanced meanings of the human language. An analysis using n-grams will capture combinations of words, such as "not good", together with registering these words separately. For example, if the search term "Chicago Bulls" would be treated by Google without a function similar to n-grams, it would yield picture results of pictures of bulls and pictures of the city of Chicago, instead of pictures of the baseball team.

However, including more terms in the analysis, either using single words or n-grams, increases the dimensionality of the feature space, as all unique terms taken into regard add another feature. As Lee & Lee (2004) notes, even moderately sized text-collections can consist of hundreds of thousands of terms. Thus, reducing the feature space is desirable within text classification to get more reliable results and to save computational power. The study explores the accuracy and AUROC differences yielded as the n-gram is changed.

Stopword removal. As the dataset contains text posts that, for the most part, are short, the removal of stopwords could result in that a greater proportion of information in each post is lost than normal. Thus, the study experiments with both including and removing the stopwords while classifying the data.

Training set size. The amount of pre-labelled data that is required for a machine learning program to function best is also interesting to understand. The purpose of this study is also to explore ways to shorten the manual analysis time of SMA, and as such the amount of pre-labelled data to be manually categorized becomes interesting. Thus, the study also explores how large the training set has to be to yield the best results.

3.3.4 Summary

The purpose of the study is to explore ways to further develop the field of SMA by using machine learning. This exploration will be done through finding the most suitable algorithm for the context. The context is innovation research, where multidisciplinary machine learning programmers are rare.

In summary, designing a machine learning algorithm is a multistep procedure where many decisions are made along the way. This is an iterative process where steps in the design are revisited to eventually yield the best results. In this study, some choices were made before the development of the algorithm was initiated based on the context of the study. The algorithm was chosen to be a supervised binary discrete algorithm. These choices were made due to the desire for the algorithm to simulate how a human would classify text into discrete categories based on the contents meaning. Using generative or discriminative models does not necessarily have to be taken into consideration for researchers. If underlying mathematical computations are irrelevant for the outcome of the study, researchers can decide on whichever classifier performs best. Deciding on which algorithm and belonging parameters to use is often a process of trial and error, a process that is used in this study.

The first step in the development of the algorithm after these initial choices is to represent the data in a suitable way with regards to the research question. Which data to include is a complex issue for topics such as text mining, but there are many previously explored methods for these situation. The most popular is the bag-of-words approach combined with a weighting schemes for term frequency, which are used in this study. For these methods, semantic tools like stemming, stopword removal and n-grams exist to make the transition from human language into numerical values a bit more valuable. The study will explore how to best process the dataset for this context.

Due to their popularity, proven usefulness and simpleness, Naive Bayes, logistic regression, and SVM are the classifiers that the study is concerned with exploring. These will be fine-tuned by changing the hyperparameters of each classifier, changing n-grams and removing or including stopwords. The study will also investigate how large the training set is required to be for each algorithm and if there is any difference in analyzing data from different platforms. These investigations are for the most part done through grid searches. To evaluate the algorithms, the study will make use of the two popular metrics accuracy and AUROC. Also, a comparison of the case conclusion using manual or machine categorization will be used as a qualitative metric. The ambition of the chosen method is to define a standardized method based on machine learning that can be easily applied to innovation research in order to lower the time the research takes to perform.

3.4 Manual Data Labeling

In order to fulfil the purpose to investigate the barriers to diffusion, the text-posts in the dataset will be categorized according to the categories defined in the theoretical framework. The framework is based on theories on innovation diffusion as well as the constraints in the data and to fit a machine learning analysis process. The categories, or classes, are “Technological Utility”, “Network Externalities”, “Price” and “Trialability”, as well as “Spam” for the garbage posts. To develop the training set and a test set for the machine learning algorithm each post is given a class label depending on the content of the post. Normally, machine learning is applied to let researchers not manually go through the entire dataset. However, all of the 6000+ posts in the data set are given manual labels. This is done in order for the research to make use of stochastically developed training sets and datasets while developing the best possible algorithm. Also, in order to use the difference between a manual labeling and a machine labeling as a measurement of how applicable machine learning in innovation research, the entire dataset needs to be labeled.

As mentioned, the dataset naturally contains garbage posts. This happens since the posts are collected based on the mentioning of the two phrases “HTC Vive” and “Oculus Rift”. Many posts in social media that are not even about VR-technology are saved. Most of these posts contain either something close to spam (Fig 3.9.) or the poster have one of the tags in his signature when posting (Fig 3.10.) or the post is in the wrong language (Fig. 3.11).

Consequently, all the post not discussing VR-technology were given the class label “Spam”. This is done in order to be able to remove the garbage data.

(Fig. 3.9) “VR battle royale: PS VR vs. HTC Vive vs. Oculus Rift https://t.co/Virs7kQH8x”

(Fig. 3.10) [TRANSLATED FROM SWEDISH] “It is most important that you learn to say no and that you can tell when it starts to become overwhelming. Have been in the same situation and I know that it is not easy, but you have to. [I7 3770K @ 4.6Ghz | Asus 980 Ti 6GB | 16GB Corsair Vengeance DDR3 1600Mhz | Asus P8Z77-V PRO | Samsung 830 128GB | Corsair H100 + 2 x Scythe GT 1850rpm | Cooler Master HAF X | Corsair AX 850W | Samsung DVD±RW | Sony 46″ LCD Full HD | HTC Vive]”

(Fig. 3.11) [NOT SWEDISH] “RT @Virtualne_info: Postapokalyptická stíleka Arktika.1 bude urena exkluzivn pro Oculus Rift #virtualne #VR “

Secondly, not all the posts that are about VR-technology are discussing the technological utility or the network externalities (Fig. 3.12). Such posts are not given any class label at all. These posts are thus saved from being deleted when the garbage posts are removed, making future VR-related research possible on the dataset.

(Fig. 3.12) [TRANSLATED FROM SWEDISH] “Hi! Have for a long time been interested in getting me a VR-headset, guess I’m most into getting HTC Vive. Since a family member is traveling to USA before christmas, I am thinking about ordering a headset. But the question is if one should wait with the purchase now and wait for generation 2, or if one should just close the deal.”

Thirdly, posts that are discussing the price of the products (Fig. 3.13) and posts that are discussing the possibility to test VR-technology without actually purchasing the products (Fig. 3.14) are given the class label “Price” and “Trialability” respectively.

(Fig. 3.13) [TRANSLATED FROM SWEDISH] “The only thing the stops it right now is the price. I and many others that are positive to it but have not yet bought it have not done so precisely due to the price. When I can get a new HTC Vive (version 2?) at max 6000 [SEK] I will close the deal. 6k is not something most have just lying around for fun.”

(Fig. 3.14) [TRANSLATED FROM SWEDISH] “Went and tested HTC Vive for the first time and was completely sold! This will be this year’s christmas present.”

The posts that are discussing the two dimensions of value are thus given labels “Technological Utility” (Fig. 3.15) or “Network Externalities” (Fig. 3.16). With the exception of the label “Spam”, none of the labels are exclusive. This means that a single post can be given several different labels (Fig. 3.17).

(Fig. 3.15) [TRANSLATED FROM SWEDISH] “HTC Vive headset have an OLED-panel with 2160x1200 pixel resolution (1080x1200 per eye) and 110 degrees field of vision. The headsethave 32 motion sensors, a gyroscope and an acceleration meter that renders your motions with extreme precision. The headset has straps that can be adjusted after the user’s head and interchangeable soft padding around the eyes that ensures that it is comfortable to wear. Additionally, it can be adjusted so it fits to have glasses on under inside the headset.”

(Fig. 3.16) [TRANSLATED FROM SWEDISH] "There are 354 VR games and John wick is on the first page before it has even been released so it looks really bright. There are not all to many that own an HTC Vive set but that is increasing all the time. The supply VR games is not all that large so this can become really good in the end."

(Fig. 3.17) [TRANSLATED FROM SWEDISH] "Additionally the technology that is inside (that in many ways resembles that inside a smartphone) is formed in a much better way than corresponding technology. One of the main challenges with today's smartphones is how one should be able to squeeze in more power on a smaller surface. With increasingly tighter margins the risk to make a mistake increases, which also led to Samsung's catastrophic position during last year when all Note 7 phones were pulled back, after they started burning without any reason. In a stand-alone VR-headsets, more components can thus be spread of a large volume, which gives one room for clearly better performance, and larger batteries.
[...]
No matter how good technology and hardware we've got, it is the experiences that are the deciding factor for if VR will succeed or not. The by far biggest AAA-game developers have waited, with reason, seeing that it is virtually impossible to make profit from the 100-million dollar project that are used to. Earlier this spring we saw how one of the first really well produced games was released - Robo Recall from Epic Games. But two other are at the door that together will drive the needs of the customers further. One is Star Trek: Bridge Commander and the other is Fallout 4 from Bethesda."

3.5 Computational Data Labeling

Following the manual data labelling, the next step is to let the machine learning classifier simulate the manual process. By showing the algorithm part of the pre-labelled dataset, the algorithm learns how to label new data. As presented in Chapter 3.3, the text data is processed initially by removing special characters, stemming, and stopword removal. Following this process, words are counted, and the number of occurrences are normalized and inverted. This processed dataset is then split into training and testing sets, utilizing the cross validation method presented in Chapter 3.3.3.

To assess the performance of each task, the algorithm is ran twice: once to remove all "Spam"-posts, and once to classify the four case labels related to the innovation. Naturally, the classifier will never outperform the human content analysis, resulting in some erroneous labels already in the first step, the spam-filter. This implies that for the next step, some of the data will contain spam. The performance of this step is measured in accuracy and AUROC, and by comparing the percentage of data that went through the filter. In the second step the algorithm will try to assign the four class labels derived from the theoretical framework. Similarly to the first iteration, these posts will be labeled based on testing data. Furthermore, there is room for debate regarding how the performance of algorithms compare to the performance of humans. A human researcher does not deliver 100 % accurate labels, why expectations of perfect machine learning algorithms are unreasonable. Also, supervised machine learning algorithms inherit the bias of the researcher in its categorizations.

3.6 Limitations

The study has been limited both by the researchers themselves to make the study more focused and by constraints in time and complexity. The study focus on a dataset about HTC Vive and Oculus Rift to capture the VR-discussions about PC-gaming VR. VR is today also used to view film clips, which can be done much cheaper by using a smartphone and Google Cardboard for about $5-10. Also, VR is used in more innovative ways in many industries. However, the study is limited to the PC-gaming VR due to the focus from the manufacturers themselves, the popularity of the products, and the clear substitution relationship it has with the old technology.

Using these headsets (HTC Vive and Oculus Rift), does not translate directly into assessing VR-technology. Much of the discussion is about these two market leaders, but these terms refer to commercial products rather than the

technology as a whole. Such a search could exclude discussions about just VR as a phenomenon, which could have implications when measuring what topics dominate the online conversation; had discussions been about just VR in general, discussed topics could have differed. Furthermore, limiting the study to Swedish social media results in a significantly smaller dataset and has implications on the language processing tools used in the algorithm.

Using Swedish texts comes with some unique traits compared to analyzing English data. The Swedish language contains compounds to a higher degree than the English language. Naturally, due to basic combinatorics compounds are less frequently occurring in a dataset, resulting in implications for the classifiers. The following is an example found when categorizing the data:

> *Mellansdagsrean är igång och hur man spendera julklappspengen bättre än på ett rejält VR-kit?*
> *Vi prissänker HTC Vive med en tusenlapp! Passa på!*
> *#oskarshamn #kalmar #västervik #nybro #växjö #HTCvive #Vive*

Words like *mellandagsrean, julklappspengen,* and *prissänker* are all uncommon compounds, and for a machine that has not previously encountered these terms will not recognize them, although they are all closely related to Price. An English classifier would not have these problems, as *Christmas sale* (Swedish: *julrea*) would not embed the word *sale* into the compound.

Furthermore, much of the limitations of this study are derived from keeping the study on a beginner's level. When designing algorithms for real life application, much effort is put into optimizing the algorithm's performance. This study instead aims to describe the process and the utility within innovation research, pressuring the study to instead increase transparency and visualizability. Therefore choices regarding algorithms and feature selection are meant to be kept simple, while at the same time remain representative for industry standards.

3.7 Research Quality

Social media is a unique environment for studying large scale social systems without researcher interference. The thoughts and opinions of people, and how and which of these are expressed arguably tell more about human behavior than structured interviews. Social media could therefore be a good candidate for studying the social systems within which the diffusion of innovations occur. Researchers must, however, be vary of the risks and implications of using such a simplified model for a social system.

The characteristics of social media analytics limits the researcher's ability to affect which topics are covered. The inability of asking follow-up questions or asking specifically about a narrow topic limits the researcher to very wide-scale research questions. Social media also implies the inability to observe who the author is, and consequently limits the transparency of to which degree the social platform is generalizable for a larger population. As seen in Roger's (2003) groupings of adopters, the early adopters of a technology is not representative for the social system as a whole.

Yin (2014) notes the criticism against generalizing findings from single case studies, but emphasizes that a case study should not be seen as a sample, but as a theoretical proposition. Accordingly, this study aims to extract insights about machine learning within innovation research utilizing SMA, and proposes that these findings should be generalizable within this context. Thus, the generalizability of the findings from the case study is subordinated as it mainly serves as a simulation of a common research process. However, using social media as a tool for data collection implies the risk of not getting a representative sample of the population of a social system.

The machine learning algorithms on which the study is based are designed to cohere with good practice and industry standards. Algorithms are however coded by the authors, and the process of algorithm design is exposed to subjectivity and programming skills. To reduce these implications, the study aims for a high transparency in order for critics to assess methods and algorithms used. As machine learning algorithms learn from example data presented by the researcher, the machine will also inherit the bias of the researcher. This behavior can be seen as a double-edged sword: on one hand, machine learning algorithms will not deviate from the behavior displayed by the researcher, enabling algorithms to grasp very subtle nuances in language. On the other hand, machine learning algorithms cannot be trained objectively by a single researcher.

4. Results and Analysis

The following chapter will present the results of the study. Each presented result is accompanied by a short analysis that aims at contextualizing the result to the reader. The study conducted has tried many different settings of the algorithms through grid searches to find the correct settings and classifier for the context. As such, the research has gathered much data in the process. This data is presented below. Also, the case results are presented as well as the time consumption of the different styles of categorizing the data.

4.1 Experiment results and analysis

The experiments in the following chapters aim to find a suitable method for processing the data used in the study. Each step is evaluated by comparing the metrics accuracy and AUROC through k-fold cross validation of the dataset. The dataset contains the manually labelled data for training and testing the algorithms. Distribution of classes within the dataset is found in Table 4.1. As seen in the table, some classes occur less frequently than others, having implications on the usefulness of different evaluation metrics.

Table 4.1. Results of manual analysis. The Spam-filtering results in 73 % saved posts, of which the other percentages are based.

Label	Manual no.	Manual %
Spam	4415	73 %
Technological Utility	569	34 %
Network	555	34 %
Price	236	14.4 %
Trial	185	11.4 %

4.1.1 Classifiers

Below the results of the tests using the feature selection presented in Chapter 3. Algorithms Naive Bayes, Logistic Regression and Support Vector Machine (SVM) are tested to examine which algorithms are suitable for this study. Algorithms are tuned by changing associated hyperparameters, either gradually or exponentially. Performance metrics for the testing set are complemented with corresponding metrics of the training set to visualize how the classifier overfits and underfits when parameters are tuned. Associated hyperparameters are tested for each classifier on classes presented in Chapter 3. See Appendix B for table values of figures within this chapter.

Naive Bayes

For Naive Bayes classification this study uses the MultinomialNB classifier found in the scikit-learn toolkit. As presented in Chapter 3.3, Naive Bayes is a rather simple model with relatively quick processing time. It is further categorized as a linear, generative classification algorithm. The MultinomialNB classifier is tuned only with the hyperparameter alpha, which is tested between values of 0.01 and 0.15. These values can all be seen as quite low, as there is no upper limit for alpha values. The results of the tests are presented in Figure 4.1.

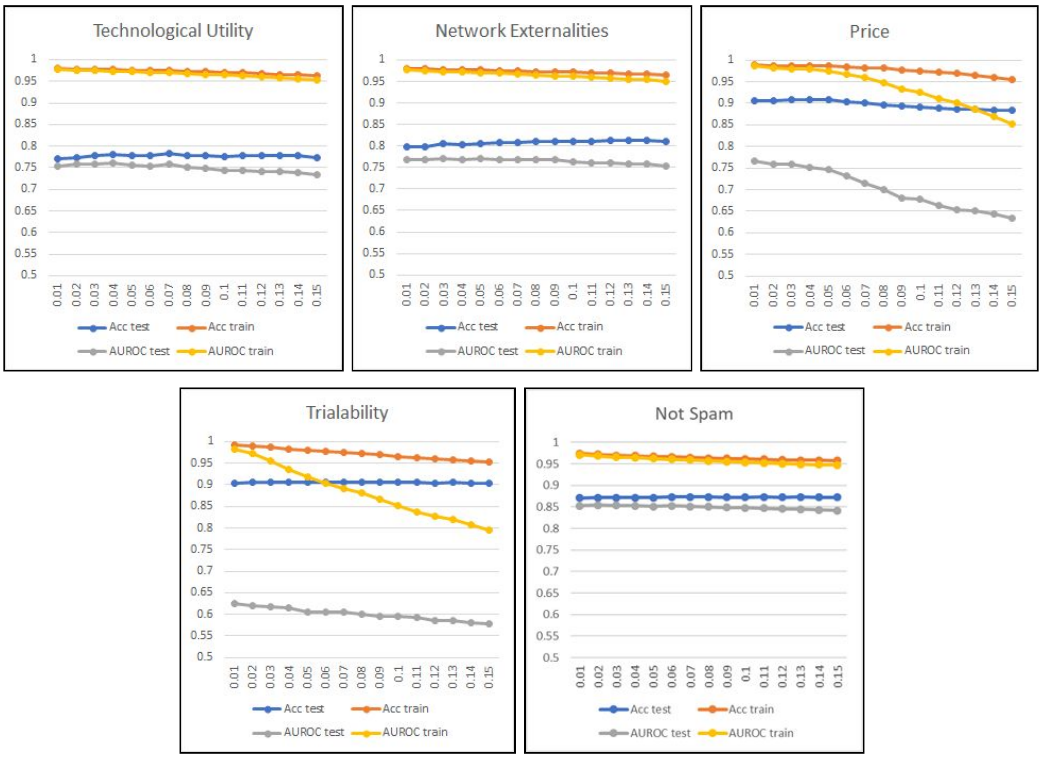


Figure 4.1. Hyperparameter tuning of alpha for MultinomialNB

The parameter tuning shows that the classifier performs worse as alpha is increased. A low alpha induces a low degree of generalization, showing that this classifier is hard to overfit on this classification task. The case of Trialability and Not Spam shows why not to rely solely on accuracy. When alpha increases, accuracy remains the same while AUROC shows that the classifier performs much worse for larger alphas. In the case of Trialability, the classifier could predict all labels as negative but still result in an accuracy of 0.886 due to the few posts with that label. Therefore AUROC would be a better metric in this case.

From these runs it appears that an alpha less than 0.08 is adequate for Technological Utility, Network, and Not Spam, while even smaller (< 0.4) values are better suited for Price and Trialability. A reason for this could be the significantly smaller datasets for Price and Trialability, where an overfitted algorithm more accurately finds specific posts. Thus, an overall alpha of 0.04 should perform well while still retaining some degree of generalization.

Logistic regression

The classifier for logistic regression, LogisticRegression, is imported from the scikit-learn package linear_model. Logistic regression uses the hyperparameter C to adjust the degree of generalization. Logistic regression is like Naive Bayes a linear classifier, but based on a discriminative method. Logistic regression also has a lower processing time than the Naive Bayes classifier. Values of C are tested between 1 and 15, where a smaller C results in a higher degree of generalization.

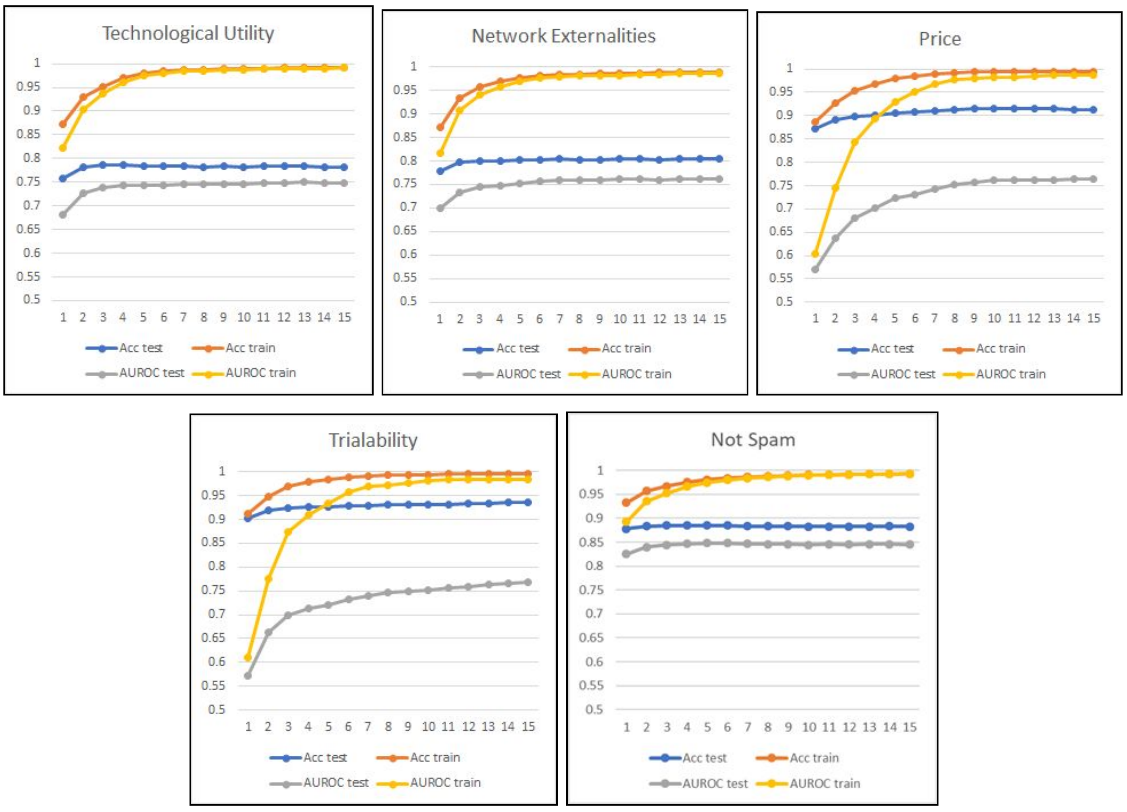


Figure 4.2. C-tuning for LogisticRegression

In all cases for logistic regression (Fig. 4.2) the increase in accuracy subsides at C = 3, while AUROC increases until C is larger than 5. For Price and Trialability, even larger C increases performance. For all labels except for Network the Logistic Regression classifier performs as well or slightly better than the Naive Bayes classifier when comparing accuracy. The same pattern is found when comparing the AUROC, except for Trialability where the Logistic Regression classifier greatly outperforms the Naive Bayes classifier. This could be a hint towards Logistic Regression being better suited for smaller datasets classes with few data points. Indeed, Logistic Regression is a discriminative method, which does not suffer from imbalanced data or lack of data points the way generative models do.

SVM

The SVM classifier (svm.SVC) requires several input parameters, but choosing a linear kernel instead of the standard rbf reduces the hyperparameters to one, 'C'. Tuning C from 0.15 to 2.25 with a linear kernel gives the output shown in Figure 4.3. As mentioned in Chapter 3.3, the SVM is based on more complex calculations than the previous classifiers, resulting in longer processing times, which increase greatly for larger datasets. Like Logistic Regression, SVM is a discriminative method.

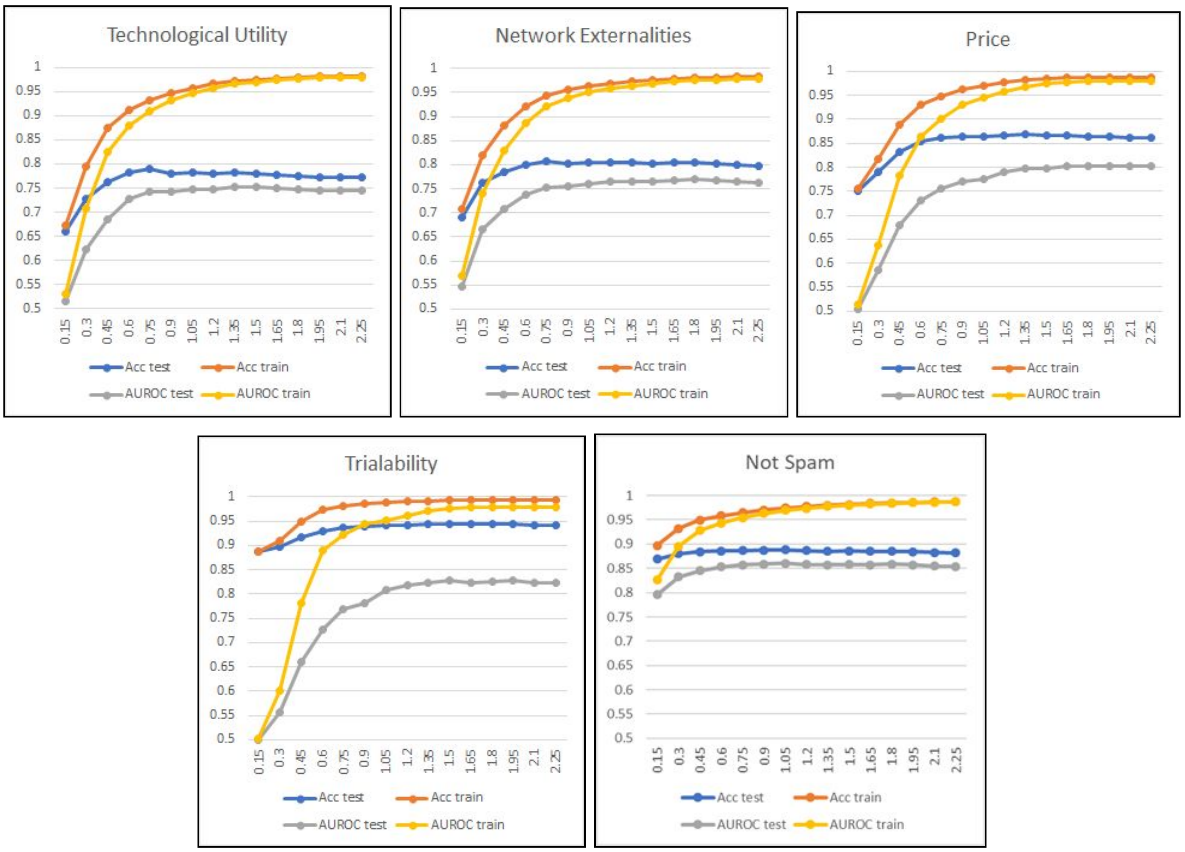


Figure 4.3. C-tuning for linear SVM

The classifier seems to maximize its performance around C = 1.2. The SVM classifier seems to outperform the earlier two, especially with regards to AUROC. Most notably the SVM classifier reaches higher AUROC for the rare classes Price and Trialability.

Using the standard rbf kernel of the SVM classifier requires a more extensive grid search as the classifier requires an additional input parameter compared to the linear kernel. Using an rbf kernel a grid search includes two hyperparameters: C and gamma. Grid searching often includes more than one dimension, making tables and 3D diagrams valuable tools for visualizing results. Compared to only one hyperparameter, including two hyperparameters increases the processing time of a grid search exponentially. Therefore, this grid search is using an exponential growth of hyperparameters to span a larger interval of values, and only includes five values for each hyperparameter. This results in 25 runs compared to the previous 15. In Figure 4.4 C is tested for values between 0.1 and 25.6, while gamma is tested for values between 0.1 and 1.6.

The diagrams for SVM with kernel = 'rbf' (Figure 4.4) display the same plateau pattern seen in earlier algorithms. The algorithm seems to maximize for C around 1.6 and smaller gammas (< 0.4). For rarer labels Price and Trialability the classifier displays the same behavior as earlier algorithms, performing better for less generalizing (overfitted) algorithms. Again, this is probably due to the lack of representative data. To further tune this algorithm, a grid search with a more narrow interval could be conducted. However, the rbf-kernel seems to result in about the same performance as the linear kernel, and the increased processing time and reduced ability to visualize would not be feasible within this study.

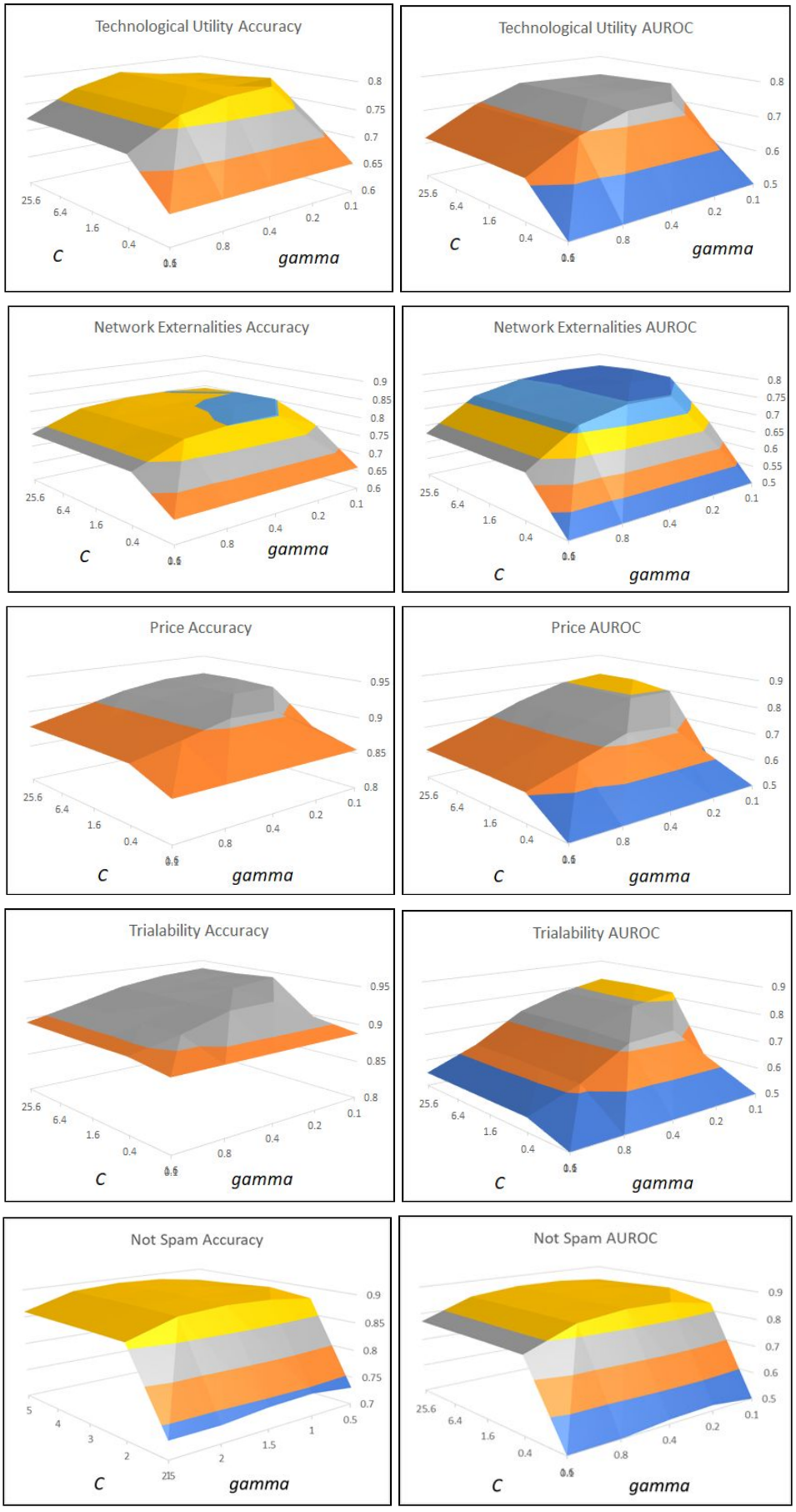


Figure 4.4. Grid search (C, gamma) for SVM kernel = 'rbf'

4.1.2 N-grams

As covered in Chapter 3.3., including multiple words within a single term could capture more nuances in language. Choosing which n-gram to use is often part of the grid search, treating n-gram range in the same way as hyperparameters to tune. As n-grams change the structure of input data it could affect the optimization of hyperparameters, why another grid search is conducted to get comparable results. To visualize how n-grams affect performance this tuning is made separately from the previous parameter tuning.

Naive Bayes

Using the MultinomialNB classifier, and the same values for alpha as in Chapter 4.1. N-gram range is increased from one to two (n-gram = (1, 2)). Test results are seen in Figure 4.5.

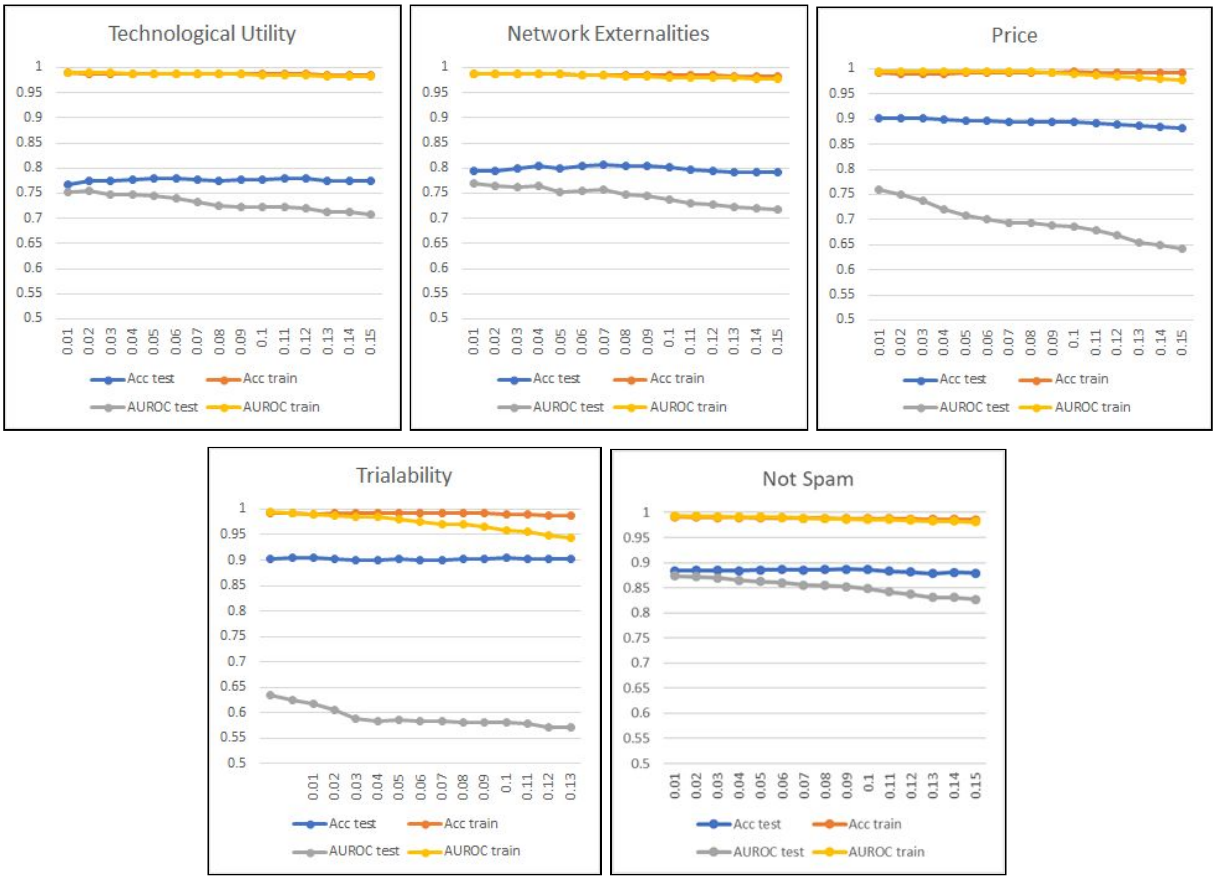


Figure 4.5. Hyperparameter tuning of alpha for MultinomialNB for n-gram = (2)

For Technological Utility and Network Externalities, the results are similar to the previous outcome (n-gram = (1, 1)), meaning that including more words within terms does not reduce performance; as seen for Network and Not Spam, it could even increase performance. This means that there are combinations of words that add valuable information for classification algorithms. Increasing n-gram range does however lower the AUROC for both Price and Trialability, which can be explained by the scarcity of these labels.

SVM Linear

Using the same values for C as in Chapter 4.1. n-gram range is increased from one to two (n-gram = (1, 2)). The results are shown in Figure 4.6.

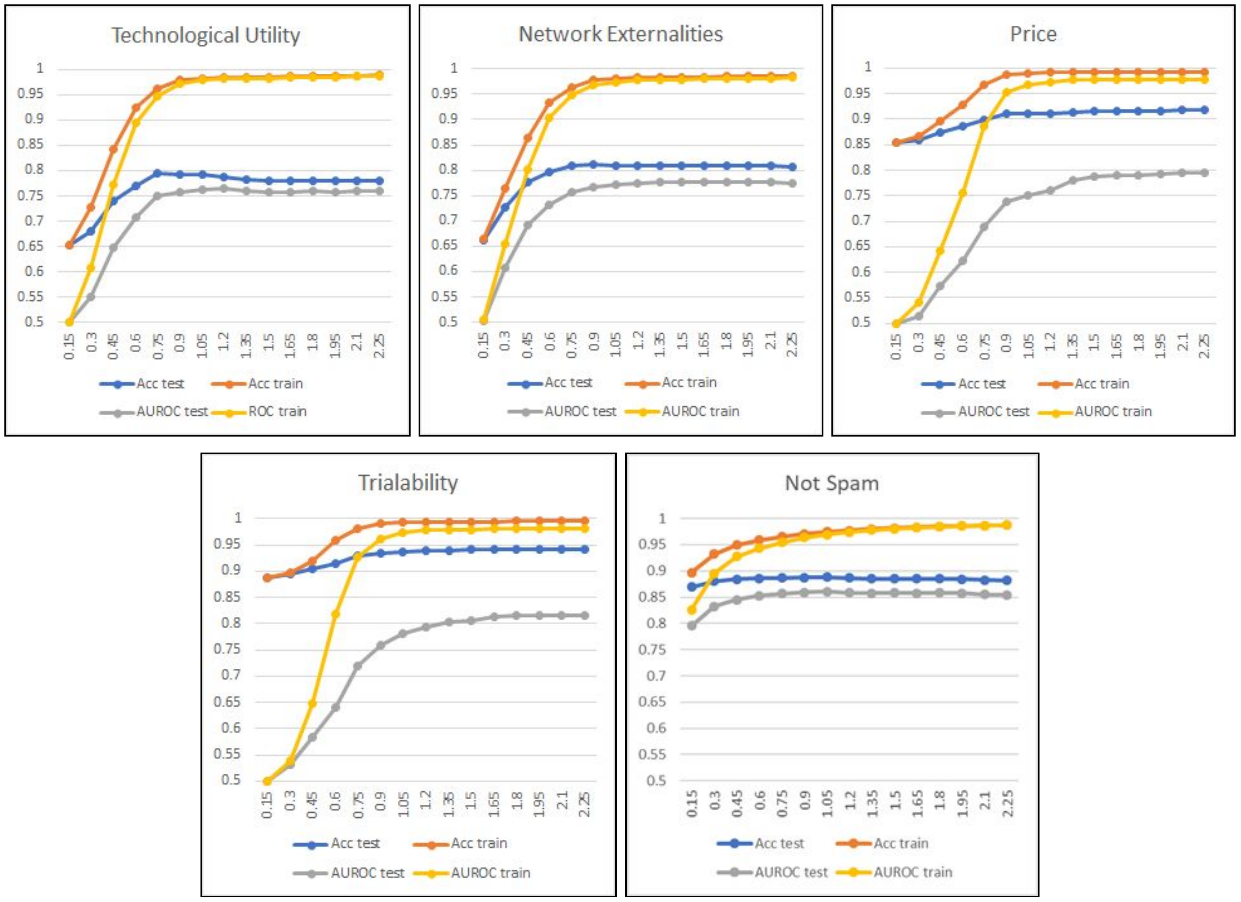


Figure 4.6. C-tuning for linear SVM where ngram=(1,2)

Again, the classifier seems to balance around 1.2., except for Not Spam balancing on a slightly smaller C. Increasing n-gram range from one to two words seems to result in the same performance for most classes. It does however not increase performance significantly and appears to be unnecessary for upcoming tests.

4.1.3 Stopword removal

Although stopword removal is often a standard procedure within text mining it holds information that could be valuable for a study. For a dataset of this size, and where posts often consist of less than 10 words, removing stopwords reduces the dataset greatly. Simultaneously, for smaller datasets reducing dimensionality is often a key objective. To assess this paradox, not removing stopwords is tested for both the Naive Bayes classifier and the SVM classifier with a linear kernel. Again, these classifiers will go through another grid search as changing the input data could affect the optimal hyper parameters. The classifiers are tested for the same hyperparameter values as earlier, and results can be seen in Figure 4.7 and Figure 4.8 for Naive Bayes and SVM respectively.

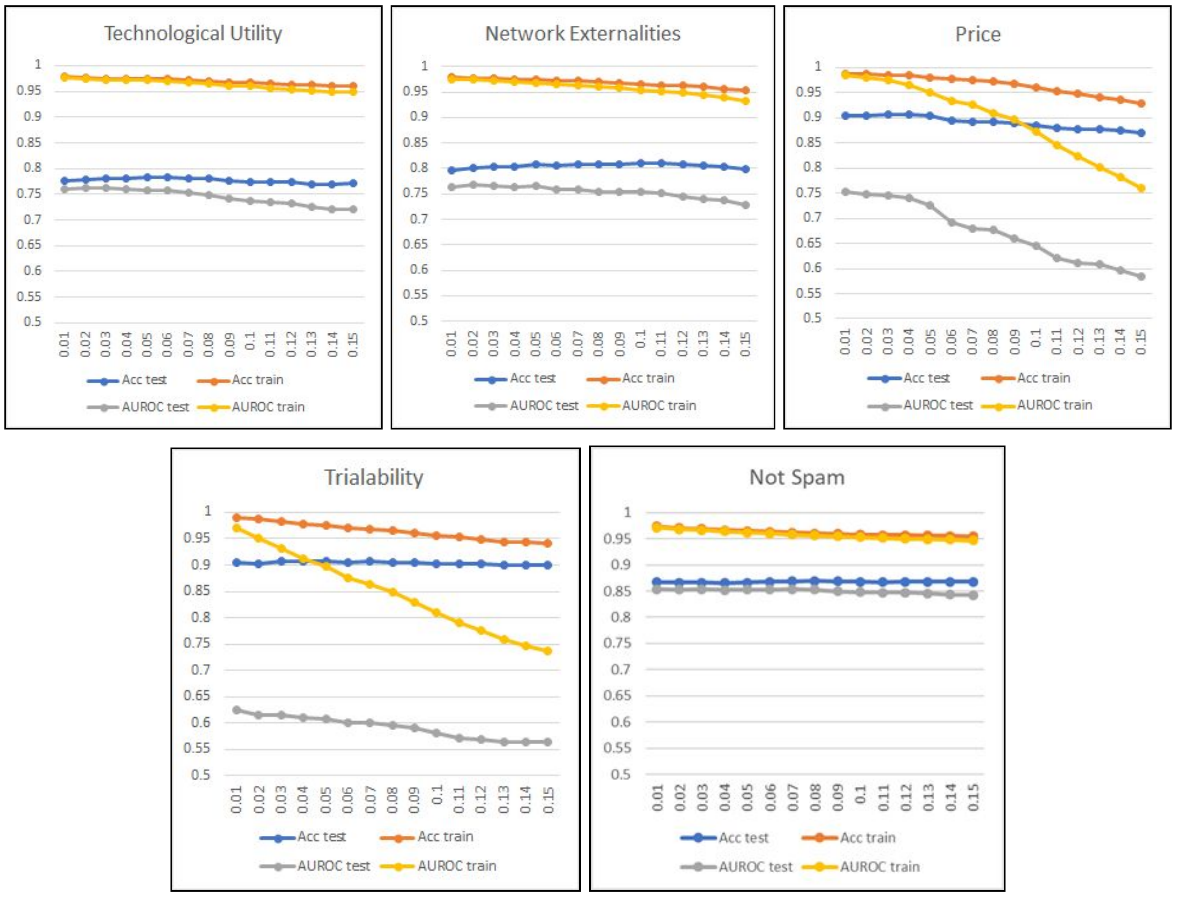


Figure 4.7. Hyperparameter tuning (alpha) for MultinomialNB with no stopword removal

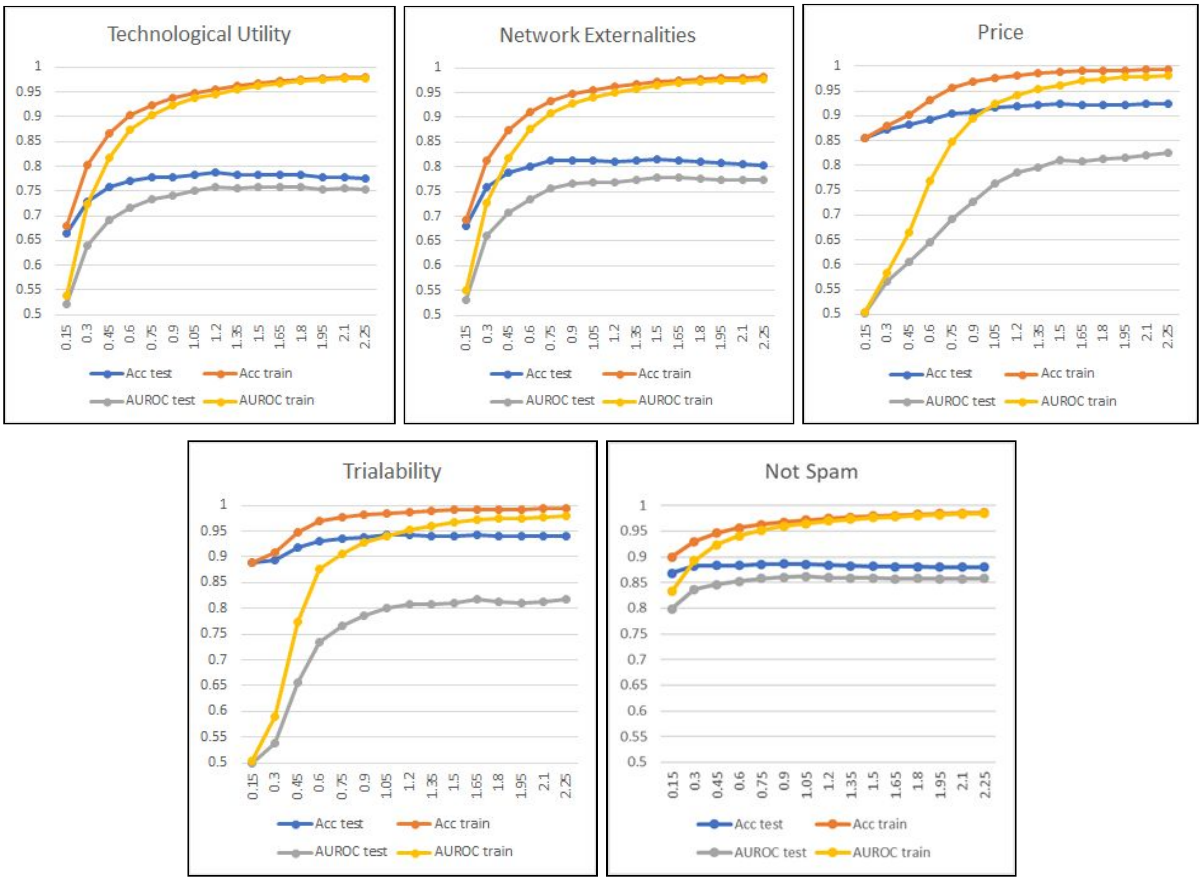


Figure 4.8. C-tuning for linear SVM with no stopword removal

Not removing stopwords does not result in noticeable difference for any of the two classifiers, indicating that these words do not add any additional information, but only increase the dimensionality of the feature space. As this study focuses on which topics are discussed, it appears natural that the phrasing of posts does not determine which class label should be assigned. If the case instead focused on identifying which user wrote a specific post, then stopwords could hold more information due to the personal choice of wordings. Seeing as the algorithm uses an inverse frequency scheme, the interference of stopwords is even further reduced.

4.1.4 Training set size

The definition of Big Data found by Boyd & Crawford (2012), where size makes data unmanageable, can hardly be applied on the dataset used in this study seeing as a manual labeling was feasible. However, dataset size is important in any machine learning context, and as a rule of thumb, more data creates better algorithms. If the amount of training data is gradually adjusted, just like the hyperparameters, an unfolding trend gives a hint about how an increased dataset size would affect results. For this comparison k-fold cross validation is not used as the small fractions would induce a testing set too small. Instead, a set aside testing set of 20 % of the dataset will be used, while the remaining 80 % is randomized into the tested fractions.

Naive Bayes

The Naive Bayes classifier was found in Chapter 4.1 to perform reasonably well at alpha = 0.04 for all labels except Trialability. When increasing dataset size, a generative model like Naive Bayes receives more input data from which to calculate the statistical distribution of data points. Dataset is increased gradually from a fraction of 0.05 until the all training data is consumed at a fraction of 0.8 (Fig 4.9).

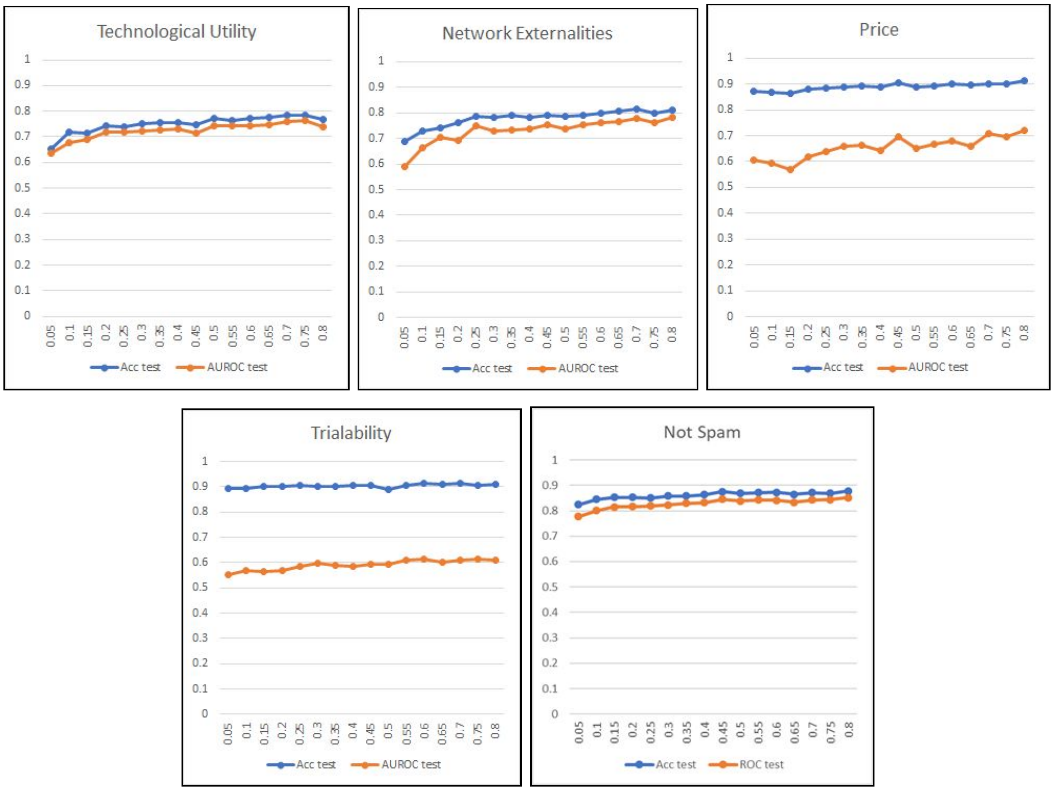


Figure 4.9. Increasing training set size (percentage) for alpha = 0.04

Increasing the amount of training data shows that this classifier would still benefit from more input data. Especially Network and Price does not seem to subside before all training data is consumed. In the cases of Trialability and Not Spam, there is however not much benefit from increasing data size. These two seem to only need about a thousand posts before subsiding. This could be a result of trends in data being very easy to find, such as the word "test" being a dominant feature for Trialability. It should however be noted that this classifier did not perform well for Trialability over all.

SVM

Choosing the linear SVM classifier limits the hyperparameters to only C. As seen in Chapter 4.1. a C of 1.2 would give the most accurate results.

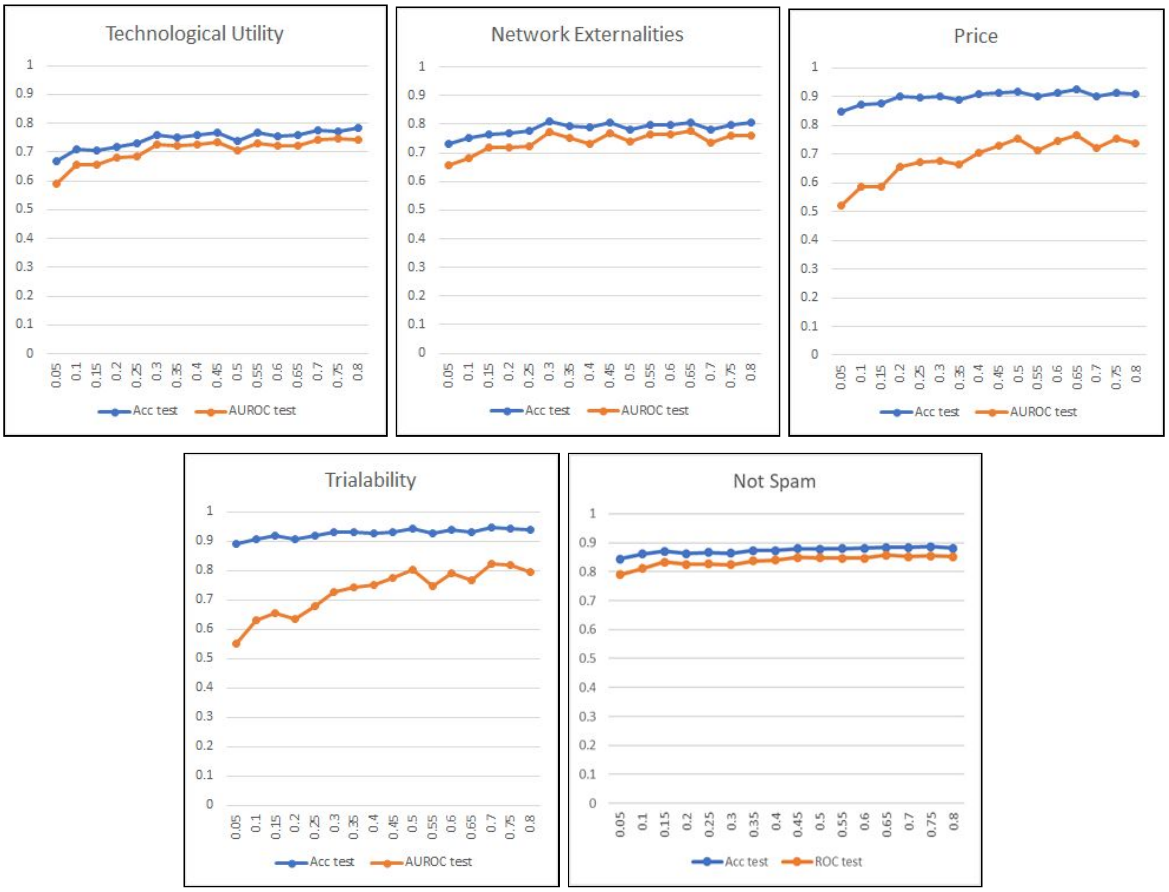


Figure 4.10. Increasing training set size (percentage) for C = 1.2

Once again the classifier subsides at around a thousand posts, and for the SVM, Price and Trialability also appear to come close to a maximum (Fig. 4.10). Thus, increasing the rather small dataset used in this study would not increase performance significantly. The great increase in AUROC comes from the scarcity of these categories, as discriminative methods rely on previous examples to categorize new data. For Not Spam it is evident that the classifier quickly reaches high performance, which is expected as many unwanted posts have an obvious common trait, such as being expressed in another language.

4.1.5 Summary

The final resulting algorithm that works best on the data and framework is based on the SVM classifier. The hyperparameter C is set to 1.2 and the N-gram to 1. Also, the stopword removal is used and the training set size needed for analysis is as high as possible but at least 1000 for filtering and about the same amount for categorizing

the labels derived from the theoretical framework. These are the settings that are applied on the case. As seen in Figure 4.3 in Chapter 4.1.1., this classifier results in an accuracy of 89 % and an AUROC of 0.866 in the Spam-filtering process. For labels Technological Utility and Network Externalities, accuracy is about 80 % with an AUROC of 0.75, while rarer labels Price and Trialability reach a slightly higher accuracy, but a lower AUROC at about 0.8. Together, these two steps result in a lower performance than each step individually.

4.2 Case Results

The case is used an example to apply machine learning on and used to understand how well the algorithm developed function in practicality. Below follows the results of the two methods (Table 4.2).

Table 4.2. The results of the machine categorization compared with the manual categorization. Note the brackets where the amount of correctly saved posts in each category after the Spam-filtering is presented.

Label	Manual no.	Manual %	Machine no.	Machine %
Spam	4415	73 %	4404	73 %
Technological Utility	569	34 %	355 (of 465)	21.7 %
Network Externalities	555	34 %	394 (of 473)	24.1 %
Price	236	14.4 %	143 (of 212)	8.7 %
Trialability	185	11.4 %	117 (of 161)	7.1 %

Manual categorization

The manual categorization is in this study viewed as the actual correct categorization of the data. Out of 1626 posts in the dataset that are about VR-technology and left after the filtering, 569 discuss the technological utility and 555 discuss the network externalities. That means that about a third (34 %) of all discussion in Swedish social media concerning VR-technology for PC-gaming are about the performance of the headsets themselves, while the same amount of discussion (34 %) is about the access to games or about the installed base. 236 of the posts discuss the cost of the headsets, which is about 14.4 % of all discussion while 185 or 11.4 % of the posts discuss possibilities to test VR before a purchase. As such, about 25.8 % of all posts discuss the gap.

Machine categorization

The filtering of the spam posts does not show a different amount of posts than the manual. In both cases 73 % of the posts are saved for further analysis. However, the accuracy of the algorithm while conducting the categorization is about 85 %, and the AUROC at about 0.85, which means that some of the posts saved are not supposed to be saved and some not saved are discarded wrongly. The algorithm label 21.7 % of the saved posts after the filtering into the technological utility and put 24.1 % as discussing the network externalities. 8.7 % discuss price and 7.1 % discuss the trialability, according to the program. The discrepancy between the two analysis methods is large in some contexts and small in other. If the research was aimed at investigating a ratio between the sizes of the categories, the amount of posts categorized right or wrong is of low importance, as long as the accuracy is the same between the different labels. However, in the context of this study, which essentially equates the amount of discussion of each category with the level of demand, the change in actual number of post in each category can be of high significance.

5. Discussion

The following chapter discusses how machine learning should be applied within the area of SMA, and what implications machine learning has when applied within the area of innovation research. Furthermore, this chapter is digging into what implications a machine learning based research method would have had on the case results.

From the case study it is notable that none of the three classifiers truly excels when measuring performance. As suggested by Christensen et al. (2016) the SVM classifier reaches the highest overall performance, but alternative algorithms cannot be excluded from consideration, especially not when considering neural networks. Furthermore, when increasing dataset size, performance of classifiers seems to subside at around a thousand posts for all classifiers, with an exception for rarer labels. Thus, no classifier is the obvious choice even when dataset size increases. The subsiding increase in performance when dataset size increases suggests that even small datasets like the one used in this study could be useful for simpler machine learning algorithms. These observations suggest that improving the model requires a more thoughtful feature selection to capture the valuable information within input data.

5.1 Standardization

The study is conducted in order to investigate how machine learning can be applied within innovation research. More precisely, the study experiments with an example case and some basic algorithms used within text analysis. The exploration results is a somewhat optimized algorithm that is applied in the case. The ambition of the study as such is to find a somewhat standard method for conducting innovation research SMA, in order to save analysis time.

The study highlights that the algorithm design involves many angles and unique aspect specific to the study. The SMA context ensures that the data is from social media. However, the data in social media can consist of several types of data: timestamps, number of likes and text content to mention a few. As such, each type of study would have to choose the specific data to analyze. That choice would significantly alter the structure of the program. This study, for instance, studies the text content of social media posts, and classifiers are chosen accordingly.

Furthermore, the study uses a case example containing a somewhat, within innovation research, normal and expected perspective, research question and choice of theory. The framework is developed to fit with SMA and machine learning. The example case applies a method of text-based corpus analysis. In contrast, another standard method of text analysis is to categorize the posts after sentiment. A third way could be to look for posts that say something unique about VR-technology. The two above examples highlights that even though the practice of reading the text content is the same process for a human, a machine learning program is hard to standardize to do so, but must instead be developed from the ground up for every different type of study.

Thus, a way to develop a standardized way to use machine learning in a corpus analysis SMA would be to avoid the dependency on the framework. An example is a sentiment analysis. In such analyses, as the categories (“positive”, “neutral” or “negative”) are not defined by a particular theory-based framework, a standardized program is much more feasible. Indeed, such analyses already exist within SMA-based innovation research (Laurell & Sandström, 2016) which makes this solution of adding machine learning to SMA even more feasible in innovation research.

However, the conclusion of this discussion is that a standardization of an easily applied program to be used by researchers inexperienced in machine learning is unfeasible. That is, a machine learning program cannot be viewed

in a similar way as a web page which can be updated by an individual without prior experience of programming. The type of program used in research would mostly have to be developed from the start for each different dataset or each theoretical lens chosen.

5.2 Time saving

As the dataset in an SMA research can often rise to a very large size, a Big Data analysis method can be needed. In such cases, it might be worth the effort to either employ a person to develop a functioning algorithm, or invest time in learning machine learning. As seen from the case results, a smaller testing set like the one used in this study could hold enough information to analyze a larger dataset. In other words, categorizing 6000 posts manually could be enough within this context. There is however always the issue of sacrificing accuracy for time, which leaves the researcher to decide where such a method is feasible. The fact that larger training sets always increase the performance of algorithm further emphasizes that manual labeling should only be compromised if necessary, and a certain amount of manual labeling is always a prerequisite for conducting this type of study.

Also, when developing the training set as well as a test set for the study, the labeled subsets can be used in other ways than to train a machine learning algorithm. For instance, one way to analyze large datasets is to use statistical methods to draw general conclusions of the patterns in the data. That is, studying a small portion of a dataset can be enough to draw conclusion about the whole data. Such statistical methods are standard practice, and the cornerstone in many research processes. A standard statistical analysis can often be very useful in SMA, since labeling only a couple of thousands of posts can find the general pattern in the data, saving the time it takes to develop a machine learning algorithm to do the same task. As the statistical method have a ceiling of how large the studied subset needs to be, compared to the as-large-as-possible training set in machine learning, even in Big Data situations, machine learning is not always the go-to method for analysis. Also, traditional statistical methods have more transparent and standardized methods for assessing quality. For that reason, traditional statistical methods are perhaps more desirable within research.

5.3 Application areas

The above discussion begs the question of when machine learning should be applied in innovation research. In temporal analysis (Brooker et al., 2016), topics change over time. In such settings a machine learning program can follow trends in a dataset in a way statistical analysis cannot. For instance, this study could have been analyzed to see if sentiment towards VR changes over the year. This study, however, makes a corpus analysis, treating the whole dataset as one semantic entity. In a corpus analysis, machine learning is only useful if the data is too large, so called Big Data, as previously discussed. Often statistical methods are enough to find patterns in Big Data. Traditional statistical analysis, however, does not allow the researcher to look deeper into data after computing, which is an advantage when using machine learning. That is, the statistical method can find patterns in the data, but does not allow for further analysis of the data. Machine learning picks out individual posts in the dataset, effectively filtering the data. Thus, different types of filtering situations within Big Data contexts is a good application area for machine learning within corpus analysis.

An example of this could be a very large dataset of all social media posts from a certain platform where a researcher wants all negative posts about a certain subject. The subset saved can then be further analyzed, either manually or through traditional statistical methods. Only if the saved subset is considered too large for manual analysis and needs further filtering, machine learning should be applied. Using this case study as an example, the garbage filtering is the best stage to use machine learning. It is, however, also important to note that filtering spam

manually takes much less time than categorizing text, as spam is very easily detectable by just glancing over the data. As such, even in spam filtering, manual filtering can be the best option time-wise.

5.4 Case Study

The following chapter analyzes the results through applying the theoretical framework on the results from the labeled dataset.

5.4.1 Manual Case Analysis

The results show an equal amount, about one third, of discussion about the two dimensions of value, giving no clear indication as to which one is in more demand on the user-side of the market. However, this equality indicates that not only innovators have entered the market, but also early adopters. The early adopters are interested in the utility and value of the whole product when in use, not only the technological utility. As such, feedback received from the social media data can be said to be more balanced and representative than if only innovators were interested in VR-technology.

The price or the option to try the technology before a full purchase are discussed in about one in four posts. This indicates that while the consumers value both the technological utility as well as the network externalities equally, there is hesitation on the market with many customers wondering if the two dimensions of value add up to the price. This indicates that a significant barrier to diffusion is hesitation in the purchase decision in general. In regards to the research question, the most significant barrier to diffusion seen in the data is as such the price-value gap and the concerns around the price.

An increased perceived value could lower the hesitation on the market that stems from the gap between the total value of the technology and the market price. This gap can be closed or, put differently, this emergence challenge can be lowered, by two general methods, either the price is decreased, or the perceived total value of the technology is increased. Also, the hesitation to buy VR-technology could be increased due to the fact that the market is constituted by two producers with similar products. This market constitution creates a question for the consumers: which is going to dominate the market? Before that question is answered, a purchase decision is accompanied by significant uncertainties and risks, especially if the price is perceived to be high.

To lower the price of the product, the platform owners could identify the user-side of the market as the subsidy side, as described by Magnusson. The money side would be the game developers, who should be charged a higher price than on an independent market to compensate for a lower price on the consumer market. However, this tactic requires that the money side is not price sensitive. Game developers have the whole PC-gaming industry to fall back on if the profits of VR become to low, making such a tactic hard to carry through.

Since lowering the price could be hard, undesirable or not entirely close the gap between the perceived total value and the price, developers can also increase the perceived total value. One way to do this is to increase the trialability as it is described by Rogers (2003), by offering more chances for customers to experience the technology without having to make a large investment. This would only work if a trial of the product leaves the users with an increased perception of value, and not a confirmation of the value of the technology. Schilling (2010) describes how the network externalities' value can be increased through increasing the perceived size of the installed base is valued highest by the customers.

The gap of hesitation is as such identified as the most significant barrier to diffusion. The importance of lowering this emergence challenge cannot be overstated since the PC-gaming industry is identified to have high extension opportunities. According to the research made by Adner and Kapoor, this means that if the VR-technology has high emergence challenges, the diffusion cannot occur, or is very slow. If the emergence challenges instead are lowered, gradual diffusion and a steady co-existence with the old technology, is feasible, and as such desirable for VR-developers.

5.4.2 Comparing Manual and Machine Categorization

The results from the computational categorization differ somewhat from the manual analysis. One striking phenomena is that the amount of posts that is given any of the four main labels is significantly lowered. Table 5.1 shows that this loss of categorized posts occur in two steps: first a loss due to the spam filter removing Spam-posts accidentally filtering relevant posts, and then due to the algorithm having a bias towards not assigning values. The first step is natural, as the classifier erroneously filters posts which contains labels. The second step, however, is resulting in another loss. This means that the SVM classifier is biased towards assigning negative (0) values. To assess the reason behind this, further testing could be conducted, and hyperparameters could be tuned to assign more labels.

Table 5.1. The results of both categorization methods.

Label	Manual no.	Manual %	Machine no.	Machine %
Spam	4415	73 %	4404	73 %
Technological Utility	569	34 %	355 (of 465)	21.7 %
Network Externalities	555	34 %	394 (of 473)	24.1 %
Price	236	14.4 %	143 (of 212)	8.7 %
Trialability	185	11.4 %	117 (of 161)	7.1 %

Naturally, an interpretation of these lower values made without knowledge of the actual content of the dataset, could be that there seem to be other subjects that dominate the discussion, or that information within the data is not that valuable for research purposes. Furthermore, only about 15 % of posts are discussing the gap which could show to create a significant change in the conclusions drawn from the results in the previous Chapter 5.4.1. As seen in the results, classifiers have more trouble classifying these these two labels as they occur less frequently than the other two. However, the two categories Technology and Network are still dominating the online discussion with about the same portion of labels each. This conclusion would harmonize with the conclusions in the manual analysis.

The evaluation metrics are on average about 85 % for accuracy and about 0.8 for AUROC for the best algorithms developed (Fig. 4.2). The above discussion results in these metric being regarded as quite low for the machine learning analysis. Metrics at these levels seem to potentially alter the results of the study significantly. One reason behind this could be that the input data does not contain much valuable information, or rather that the feature

selection fails to capture this information. Also, there is much left to explore regarding the choices leading to the final algorithm, which could increase evaluation metrics somewhat.

However, the evaluation metrics need to be put into more perspective. The machine labels on average about 85 % of the posts correctly. This must be compared to how many posts the manual labelers, the researchers themselves, label correctly. That is, in some cases the researchers make mistakes in the labeling process, which can severely affect the algorithms understanding of the dataset. Also, the manual labeling is riddled with decisions of whether a post is or is not about a certain subject depending on different interpretations of the post. Whatever the hesitation to put a label or not results in, the mere fact that there was a hesitation from a human means that a machine will have trouble categorizing that data correctly. Thus, the accuracy of about 85 % is maybe not very low, but yet, it is low enough to alter the case results significantly. This results in a recommendation to use a manual analysis whenever possible.

6. Conclusion

Here follows the conclusions of the study. The first two sections are aimed at addressing the research purposes and answer the research question, and the last to recommend future research topics.

6.1 Machine learning within innovation research SMA

The purpose of this study is to explore ways to further develop the field of SMA within innovation research by adding machine learning, especially in regards to shortening the analysis time. This exploration concludes that machine learning is not very useful to save time in analysis of social media data in innovation research. This is due to several factors. First, standardization of a machine learning program seems unfeasible. Instead a new one would have to be developed for every study. Also, the time spent on manually labeling the dataset will always exceed a certain amount, and more testing data will always bring better performance of algorithms. The case shows that a high accuracy is very important for machine learning in a research context as the loss in accuracy compared to manual labeling compromises the quality of the study. This fact combined with the fact that machine learning algorithms often can be replaced with traditional statistical methods leads to machine learning becoming even more unfeasible to use in innovation research..

Self-labelled machine learning in Big Data situations have advantages over traditional statistical methods in corpus analyses only when the practitioner desires to be able to further analyze the data. As such, machine learning is best applied as different types of filters on the dataset, where the saved subset should be further analyzed. Also, the choice to use machine learning will be accompanied with accuracy problems that can affect the outcome of the study. Indeed, the case of this study shows that around 85 % accuracy of an algorithm significantly alters the data categorization process, and as such harms the research quality. A researcher should carefully define precisely why machine learning should be used in his or her particular study. As a standard practice, there are commonly better methods of data categorization.

6.2 Case Study

The purpose of the case study is to study discussions on social media in Sweden to assess the most significant barriers to the diffusion of VR-technology within the gaming community. The research question for the case is: “What is the most significant barrier to the diffusion of VR-technology?” The research has taken the view that VR-technology constitutes a technology ecosystem, or a platform, that substitutes traditional PC-gaming. The barriers to diffusion, or emergence challenges, found in social media are as such connected to the user-market of VR-technology. The framework was constituted in such a way that the clearest result would be achieved through a comparison between the amount of discussion of either of the two dimensions of value investigated, where a large amount of discussion about one of the dimensions would indicate a demand for such value.

However, the results show no relative demand for either of the two dimensions of value. Instead, the most significant barrier to diffusion from the user perspective stem from the decision-making of the customers, who are hesitant over the high price-tag. As such, in order to lower the emergence challenges of VR-technology, developers need to either lower the price or convince consumers that the value of the technology match the price-tag. The exact way to overcome these emergence challenges could be the subject of further research. Thus, the answer to the research question is that the most significant barrier to diffusion or VR-technology is the price of the VR-headsets.

6.3 Future Research

A valid research purpose for further research would be to assess which types of research questions could benefit from machine learning. The study shows that applying machine learning to mimic a manual theory-based categorization of a dataset in a corpus analysis is unfeasible, and for the most part not even useful. However, the study has shown other potential areas of application within innovation research where it might be of better use. Avoiding the theory-based categorization increases the chances of being able to standardize the program. As such, further research on cases using semantic elements such as sentiment analysis, or more rigorous filtering is regarded as interesting. Also, temporal analyses are identified as potential areas to apply machine learning in future research.

To further develop the field of SMA within innovation research, there are several machine learning tools not covered in this study which can prove useful. First of all, this study could be further elaborated by assessing the utility of neural network classifiers and more refined feature selection methods, such as word embedding. The fact that researchers are able to conduct this content analysis manually implies that the text strings hold enough information for machines to find the same patterns. Another issue to explore is how to successfully remove more noise in data, not only regarding spam filtering, but ways to remove tags, titles, usernames, and other garbage included in the posts.

References

Adner, R. & Kapoor, R. (2015). Innovation Ecosystems and the Pace of Substitution: Re-examining Technology S-curves. *Strategic Management Journal. 37(4), 625-648*

Al-Deen, H. S. N., & Hendricks, J. A. (2011). *Social media: usage and impact*. Lexington Books.

Amend, J.M. (2016). *GM Uses Virtual World to Perfect Future Vehicles.* Retrieved from: http://wardsauto.com/industry/gm-uses-virtual-world-perfect-future-vehicles

Amichai-Hamburger, Y., & Vinitzky, G. (2010). Social network use and personality. *Computers in human behavior*, *26*(6), 1289-1295.

Arthur, W. B. (1996). Increasing Returns and the New World of Business. *Harvard Business Review.*

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, *3*(Feb), 1137-1155.

BMW (2017). *Virtual och Augmented Reality.* Retrieved from: https://www.bmw.se/sv/avdelning/service-tjanster-och-tillbehor/bmw-apps/virtual-och-augmented-reality.html

Boyd, D., & Crawford, K. (2012). Critical questions for big data: Provocations for a cultural, technological, and scholarly phenomenon. *Information, communication & society*, *15*(5), 662-679.

Brooker, P., Barnett, J., & Cribbin, T. (2016). Doing social media analytics. *Big Data & Society*, *3*(2), 2053951716658060.

Bryman, A., & Bell, E. (2015). *Business research methods* (4.th ed.). Oxford: Oxford Univ. Press.

cross-validation (2016). (7th ed.) Oxford University Press.

Chapmann, J. (2017). *Machine learning: Fundamental algorithms for supervised and unsupervised learning with real-world applications* Createspace Independent Publishing Platform.

Christensen, K., Nørskov, S., Frederiksen, L., & Scholderer, J. (2017). In search of new product ideas: Identifying ideas in online communities by machine learning and text mining. *Creativity and Innovation Management*, *26*(1), 17-30.

ClassVR (2017). *Reduce Costs for Military Training with Virtual & Augmented Reality.* Retrieved from: http://www.classvr.com/virtual-reality-industry-work/vr-military-defence-training/

Davis, J., & Goadrich, M. (2006, June). The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning* (pp. 233-240). ACM.

Dietterich, T. (1995). Overfitting and Undercomputing in Machine Learning. *ACM computing surveys (CSUR)*, *27*(3), 326-327.

Easterby-Smith, M., Thorpe, R. & Jackson, P. R., (2015) *Management & Business Research.* SAGE Publications Ltd, London.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern recognition letters*, *27*(8), 861-874.

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. *Machine learning*, *29*(2-3), 131-163.

Ganis, M., & Kohirkar, A. (2015). *Social Media Analytics: Techniques and Insights for Extracting Business Value Out of Social Media*. IBM Press.

Garreta, R., Moncecchi, G., & Ebook Central (e-book collection). (2013). *Learning scikit-learn: Machine learning in python* (1st ed.). Birmingham: Packt Publishing.

Gleasuer, R. & Feller, J. (2016) A Rift in the Ground: Theorizing the Evolution of Anchor Values in Crowdfunding Communities through the Oculus Rift Case Study. *Journal of the Association for Information Systems. 17(19), 708-736*

Grubb, J. (2017). *Vive outsells Rift — but mobile and console VR outsell both.* Retrieved from https://venturebeat.com/2017/05/09/vive-outsells-rift-but-mobile-and-console-vr-outsell-both/

Har-Peled, S., Roth, D., & Zimak, D. (2003). Constraint classification for multiclass classification and ranking. In *Advances in neural information processing systems* (pp. 809-816).

Haykin, S. S. (2009). *Neural networks and learning machines* (3rd ed.). Upper Saddle River: Pearson Education.
Herrera, F., Charte, F., Rivera, A. J., & Del Jesus, M. J. (2016). *Multilabel Classification: Problem Analysis, Metrics and Techniques*. Springer.

Hindman, M. (2015). Building better models: prediction, replication, and machine learning in the social sciences. *The ANNALS of the American Academy of Political and Social Science*, *659*(1), 48-62.

Hughes, D. J., Rowe, M., Batey, M., & Lee, A. (2012). A tale of two sites: Twitter vs. Facebook and the personality predictors of social media usage. *Computers in Human Behavior*, *28*(2), 561-569.

IKEA (2017). *VIRTUAL REALITY - INTO THE MAGIC.* Retrieved from:
http://www.ikea.com/ms/en_US/this-is-ikea/ikea-highlights/Virtual-reality/index.html

Jebara, T. (2012). *Machine learning: discriminative and generative* (Vol. 755). Springer Science & Business Media.

Karpf, D. (2012). Social science research methods in Internet time. *Information, Communication & Society*, *15*(5), 639-661.

Kumar, S., Gao, X., Welch, I., & Mansoori, M. (2016, March). A machine learning based web spam filtering approach. In *Advanced Information Networking and Applications (AINA), 2016 IEEE 30th International Conference on* (pp. 973-980). IEEE.

Laurell, C. & Sandström, C. (2017) The sharing economy in social media: Analyzing tensions between market and non-market logics. *Technological Forecasting and Social Change. 125, 58-65*

Laurell, C. & Sandström, C. (2016) Analysing Uber in social media - Disruptive technology or institutional disruption? *International Journal of Innovation Management, 20(5)*

Lee, B.Y (2017). *Virtual Reality Is A Growing Reality In Health Care.* Retrieved from: https://www.forbes.com/sites/brucelee/2017/08/28/virtual-reality-vr-is-a-growing-reality-in-health-care/#33336c3c4838

Levy, O., & Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems* (pp. 2177-2185).

Lewis, S. C., Zamith, R., & Hermida, A. (2013). Content analysis in an era of big data: A hybrid approach to computational and manual methods. *Journal of Broadcasting & Electronic Media*, *57*(1), 34-52.

Magnusson, J. & Nilsson, A. (2014). *Enterprise System Platforms.* Lund: Studentlitteratur

Marimont, R. B., & Shapiro, M. B. (1979). Nearest neighbour searches and the curse of dimensionality. *IMA Journal of Applied Mathematics*, *24*(1), 59-70.

Menardi, G., & Torelli, N. (2014). Training and assessing classification rules with imbalanced data. *Data Mining and Knowledge Discovery*, 1-31.

Moe, W. W., & Schweidel, D. A. (2017). Opportunities for Innovation in Social Media Analytics. *Journal of Product Innovation Management*, *34*(5), 697-702.

Notified. (2017). https://www.notified.com

Oculus. (2016) *Oculus Touch Launches Today!* Retrieved from: https://www.oculus.com/blog/oculus-touch-launches-today

Rogers, E. M. (2003). *Diffusion of Innovation* (5th ed.) New York: Free Press

Schilling, M. A. (2010). *Strategic Management of Technological Innovation* (3rd ed.) New York: McGraw-Hill Irwin

Scikit Learn. (2017). 1.9 Naive Bayes. http://scikit-learn.org/stable/modules/naive_bayes.html

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, *34*(1), 1-47.

Steam. (2017) *Steam Hardware & Software Survey: December 2017.* Retrieved from: http://store.steampowered.com/hwsurvey

Stieglitz, S., Dang-Xuan, L., Bruns, A., & Neuberger, C. (2014). Social Media Analytics: An Interdisciplinary Approach and Its Implications for Information Systems. *Business & information systems engineering*, *6*(2), 89-96.

Stuart, K. (2016). *The digital apocalypse: how the games industry is rising again.* Retrieved from https://www.theguardian.com/technology/2016/may/17/video-game-industry-changing-virtual-studios

Theodoridis, S. (2015). *Machine learning: a Bayesian and optimization perspective*. Academic Press.

van Zoonen, W., & Toni, G. L. A. (2016). Social media research: The application of supervised machine learning in organizational communication research. *Computers in Human Behavior*, *63*, 132-141.

Vive. (2017) *Vive VR system.* Retrieved from: https://www.vive.com/us/product/vive-virtual-reality-system/

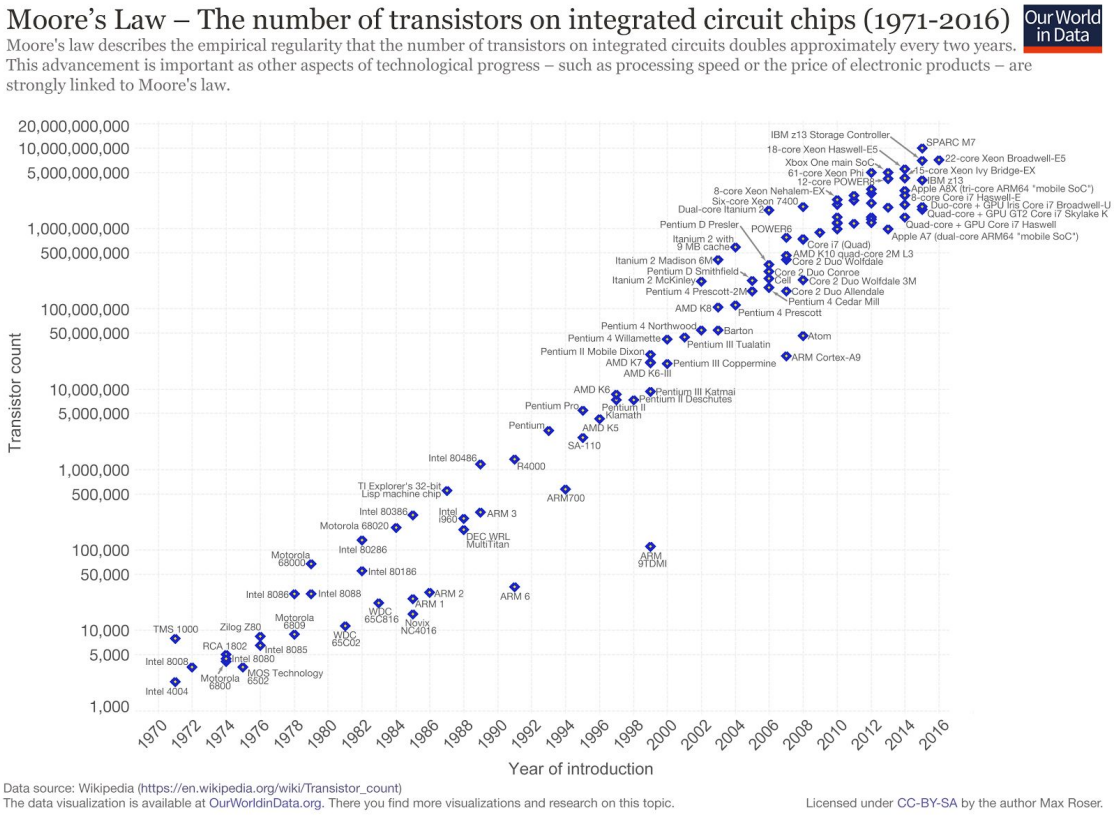
Ward, W. H. (1967). The Sailing Ship Effect. *Physics Bulletin. 18(6)*

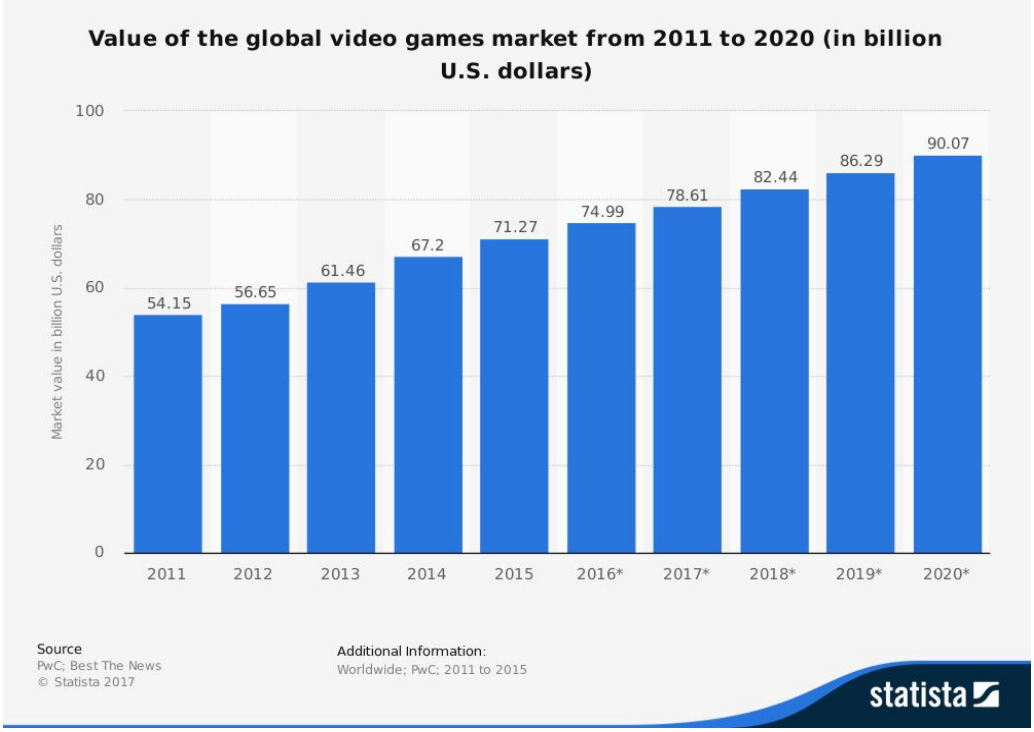
Weller, K. (2016). Trying to understand social media users and usage: The forgotten features of social media platforms. *Online Information Review*, *40*(2), 256-264.

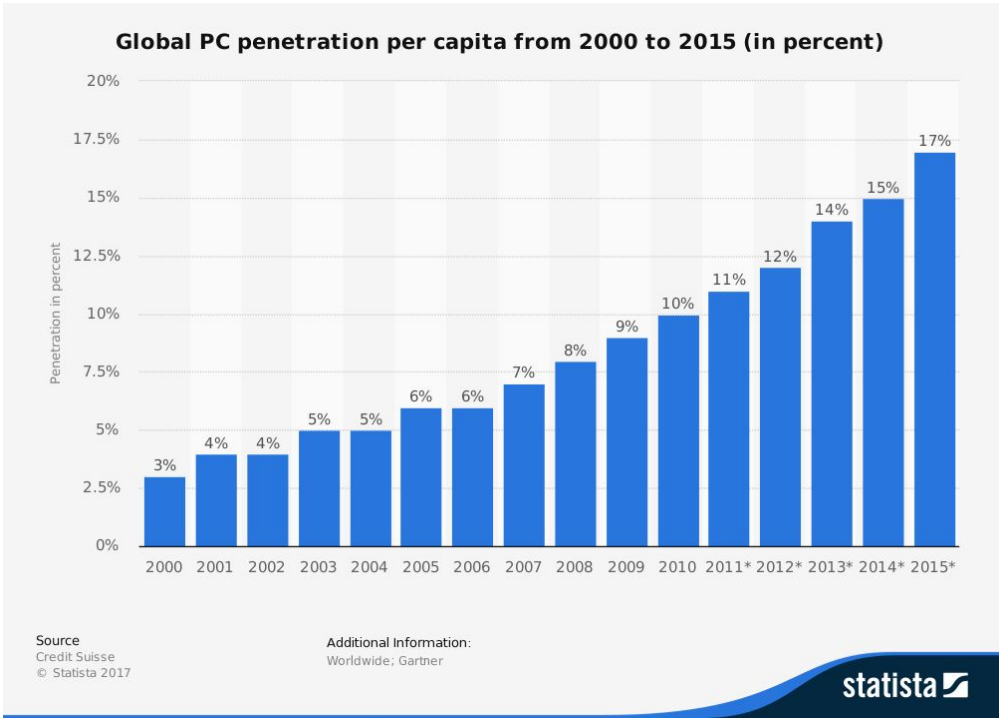
Yin, R. K. (2014). *Case study research: Design and methods* (5.th ed.). London: SAGE.

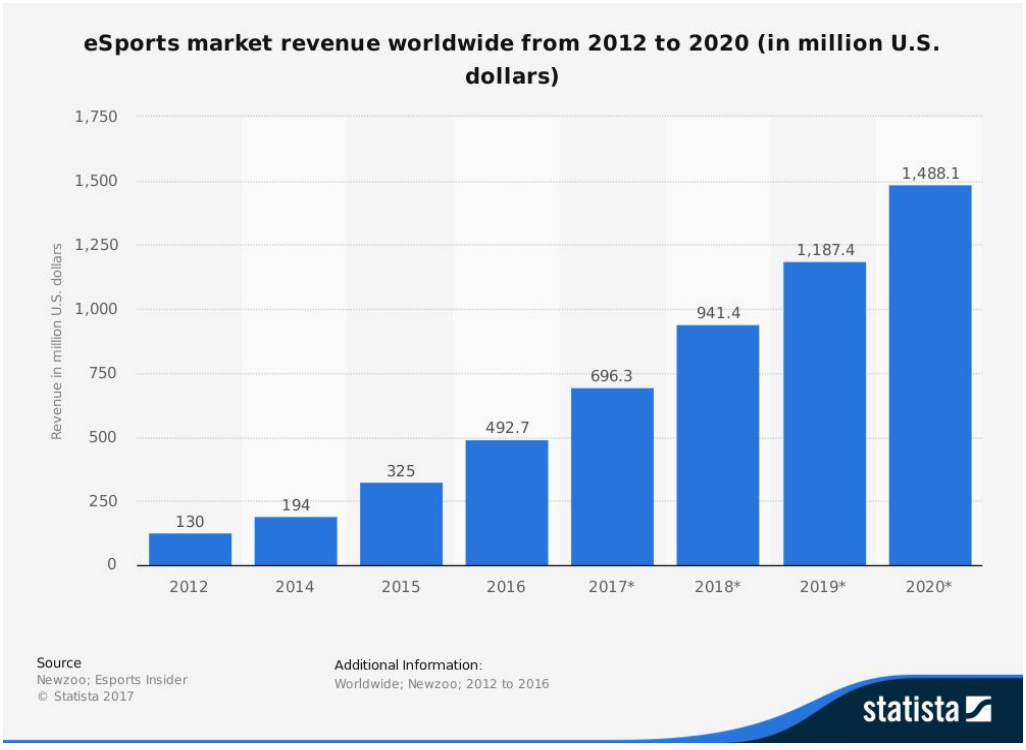
Appendix A - PC Gaming History

Below follows statistics gathered from Esport Insider, PwC and Credit Suisse through Statista, as well as from Wikipedia. The number of games developed increase every year and interests in the gaming community is also increasing seen in the growth of the Esport. Also, there potential market is steadily growing as a larger part of the population own personal computers. The technology development of the computer components is also keeping a steady rise. None of these trends show any signs of plateauing.






Global PC penetration per capita from 2000 to 2015 (in percent)
Penetration in percent
20%
17.5%
15%
12.5%
10%
7.5%
5%
2.5%
0%
3%
4%
4%
5%
5%
6%
6%
7%
8%
9%
10%
11%
12%
14%
15%
17%
2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011* 2012* 2013* 2014* 2015*
Source
Credit Suisse
© Statista 2017
Additional Information:
Worldwide; Gartner
statista

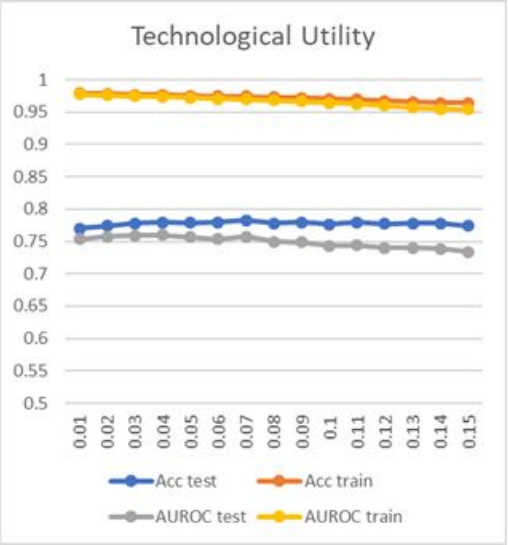

eSports market revenue worldwide from 2012 to 2020 (in million U.S. dollars)
Revenue in million U.S. dollars
1,750
1,500
1,250
1,000
750
500
250
0
130
194
325
492.7
696.3
941.4
1,187.4
1,488.1
2012 2014 2015 2016 2017* 2018* 2019* 2020*
Source
Newzoo; Esports Insider
© Statista 2017
Additional Information:
Worldwide; Newzoo; 2012 to 2016
statista

Appendix B - Table Values

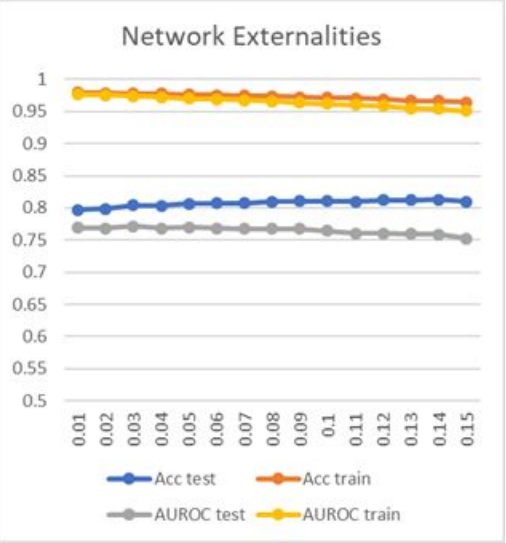
Naive Bayes

Alpha-tuning

	Naïve bayes			
	Acc test	Acc train	AUROC te	AUROC tra
0.01	0.76999	0.979106	0.753526	0.977175
0.02	0.774299	0.978031	0.75768	0.975848
0.03	0.77799	0.976802	0.759081	0.974102
0.04	0.779841	0.976495	0.759849	0.973447
0.05	0.778613	0.975112	0.75639	0.971762
0.06	0.77923	0.97419	0.753911	0.970224
0.07	0.78292	0.974037	0.757282	0.969895
0.08	0.778007	0.972346	0.749885	0.967643
0.09	0.779237	0.971578	0.74885	0.966344
0.1	0.776778	0.970349	0.743669	0.964374
0.11	0.779234	0.969428	0.744479	0.96253
0.12	0.777389	0.967584	0.740463	0.959761
0.13	0.778007	0.965894	0.740154	0.957237
0.14	0.778005	0.964204	0.738903	0.954707
0.15	0.774316	0.963743	0.733506	0.953597

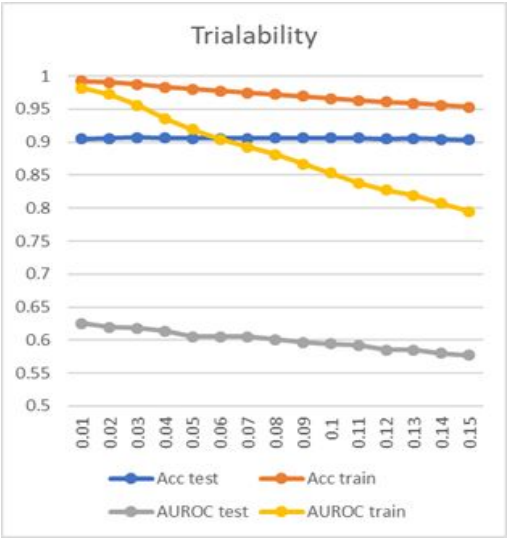


	Naïve bayes			
	Acc test	Acc train	AUROC te	AUROC tra
0.01	0.797059	0.979413	0.769183	0.976403
0.02	0.798905	0.978952	0.768029	0.975516
0.03	0.804438	0.977877	0.771809	0.973611
0.04	0.803213	0.977569	0.768409	0.972499
0.05	0.806286	0.97634	0.769634	0.970236
0.06	0.807515	0.975572	0.768383	0.969
0.07	0.807515	0.97465	0.767439	0.967533
0.08	0.809361	0.973575	0.767497	0.965616
0.09	0.81059	0.972499	0.767272	0.963652
0.1	0.810588	0.971424	0.764002	0.962045
0.11	0.809973	0.970349	0.760321	0.960466
0.12	0.812432	0.96912	0.760678	0.958411
0.13	0.812432	0.967122	0.759253	0.955142
0.14	0.813044	0.966508	0.758532	0.95403
0.15	0.809971	0.964357	0.75271	0.950666

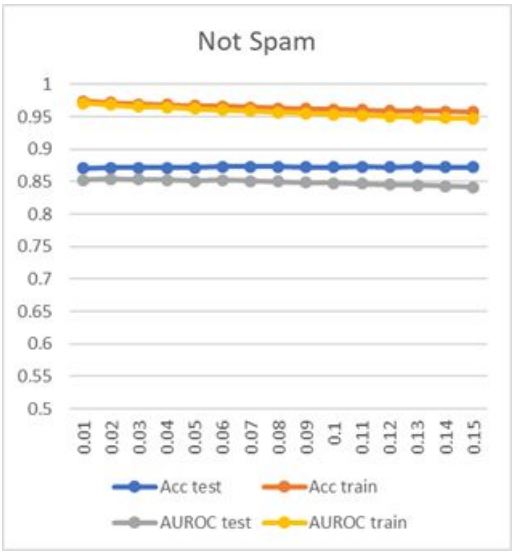


	Naïve bayes			
	Acc test	Acc train	AUROC te	AUROC tra
0.01	0.906497	0.988784	0.766341	0.986478
0.02	0.906497	0.987555	0.759929	0.982382
0.03	0.908953	0.987248	0.760022	0.980472
0.04	0.908951	0.987248	0.751782	0.978686
0.05	0.908336	0.98648	0.745393	0.974585
0.06	0.904646	0.985251	0.732048	0.968187
0.07	0.90034	0.982793	0.714445	0.959009
0.08	0.896649	0.980949	0.699785	0.947661
0.09	0.892963	0.977723	0.681351	0.933964
0.1	0.892347	0.975265	0.676901	0.924333
0.11	0.888659	0.971731	0.662899	0.910774
0.12	0.886813	0.96912	0.65409	0.899969
0.13	0.886195	0.965279	0.649854	0.885887
0.14	0.884351	0.960517	0.64336	0.868873
0.15	0.883118	0.9556	0.633373	0.851999

	Naïve bayes			
	Acc test	Acc train	AUROC te	AUROC tra
0.01	0.904689	0.992164	0.625368	0.9819
0.02	0.905301	0.990628	0.61938	0.972457
0.03	0.907145	0.987248	0.618215	0.955574
0.04	0.906533	0.9831	0.614003	0.935914
0.05	0.905304	0.980181	0.605374	0.919447
0.06	0.905304	0.977108	0.605374	0.903815
0.07	0.905304	0.97465	0.605374	0.892253
0.08	0.90592	0.972039	0.601011	0.880867
0.09	0.90592	0.969273	0.596288	0.866686
0.1	0.905922	0.966201	0.594349	0.853087
0.11	0.90592	0.96282	0.592001	0.837809
0.12	0.904691	0.960516	0.584795	0.827053
0.13	0.905306	0.958673	0.585128	0.819127
0.14	0.904076	0.956061	0.579861	0.807361
0.15	0.90346	0.953296	0.577297	0.794526

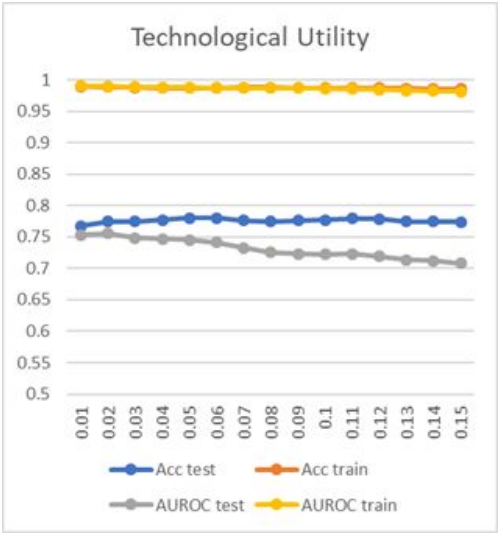


	Naïve bayes			
	Acc test	Acc train	AUROC te	AUROC tra
0.01	0.870842	0.973801	0.852848	0.970784
0.02	0.87167	0.972145	0.853832	0.968657
0.03	0.871836	0.969745	0.853432	0.965705
0.04	0.87167	0.968751	0.852821	0.964489
0.05	0.87134	0.967427	0.85127	0.962704
0.06	0.872996	0.966309	0.852618	0.960917
0.07	0.872995	0.964778	0.851171	0.958854
0.08	0.87283	0.963495	0.85016	0.956903
0.09	0.872498	0.962336	0.848854	0.955037
0.1	0.872168	0.96155	0.847468	0.954006
0.11	0.873161	0.960432	0.847301	0.952371
0.12	0.872333	0.959522	0.845304	0.950558
0.13	0.873161	0.958735	0.845077	0.949296
0.14	0.872665	0.958239	0.843198	0.948324
0.15	0.872665	0.957825	0.841832	0.947361

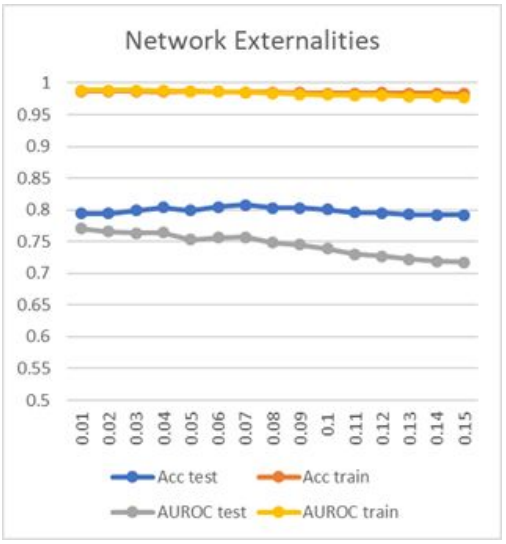


N-gram range = (1, 2)

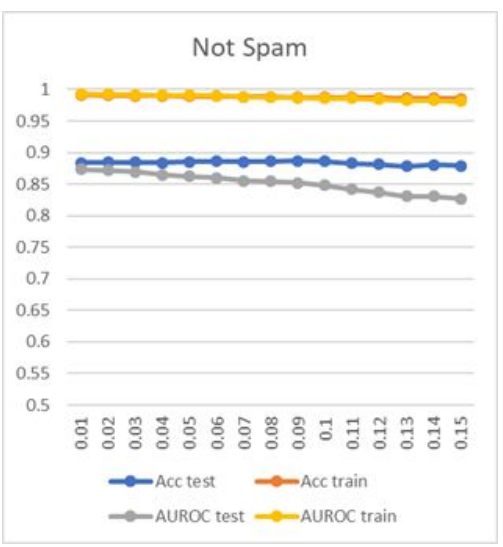
	Naïve bayes (ngram = 2)			
	Acc test	Acc train	AUROC te	AUROC tra
0.01	0.767551	0.989092	0.753399	0.990107
0.02	0.774314	0.988324	0.755644	0.989224
0.03	0.774314	0.987863	0.74835	0.988765
0.04	0.776774	0.987249	0.746631	0.987681
0.05	0.779853	0.987249	0.745502	0.987368
0.06	0.77986	0.987249	0.74103	0.987164
0.07	0.776166	0.987402	0.732279	0.987168
0.08	0.774933	0.987556	0.725113	0.987075
0.09	0.776166	0.987249	0.723423	0.986626
0.1	0.77678	0.986941	0.722434	0.985859
0.11	0.779239	0.986634	0.722789	0.985107
0.12	0.778624	0.98648	0.71951	0.984585
0.13	0.774932	0.985712	0.713366	0.982962
0.14	0.774932	0.985251	0.711711	0.982189
0.15	0.773699	0.984944	0.708387	0.981549



	Naïve bayes			
	Acc test	Acc train	AUROC te	AUROC tra
0.01	0.794609	0.986787	0.770828	0.988434
0.02	0.794609	0.986634	0.76548	0.988106
0.03	0.798907	0.986787	0.763137	0.988115
0.04	0.803823	0.986326	0.764476	0.98722
0.05	0.798905	0.98648	0.753379	0.987115
0.06	0.804442	0.985866	0.756041	0.98565
0.07	0.807511	0.985558	0.756991	0.984899
0.08	0.803209	0.985097	0.748445	0.983322
0.09	0.803211	0.984329	0.745483	0.981968
0.1	0.800751	0.983868	0.738691	0.980842
0.11	0.795832	0.983868	0.730482	0.98061
0.12	0.795218	0.984022	0.726897	0.980174
0.13	0.792755	0.983561	0.722466	0.978949
0.14	0.791524	0.983561	0.719082	0.97862
0.15	0.791526	0.982793	0.717136	0.97738

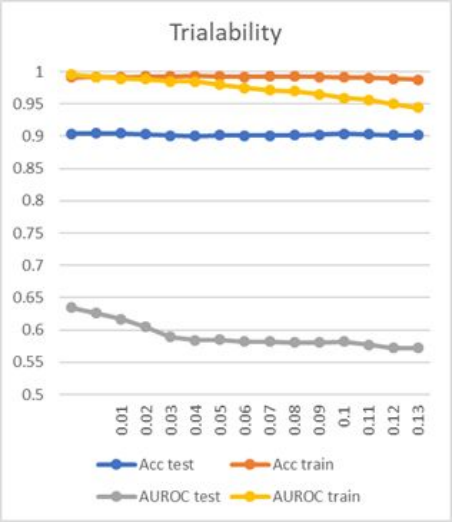


	Naïve bayes			
	Acc test	Acc train	AUROC te	AUROC tra
0.01	0.883425	0.991433	0.873431	0.993086
0.02	0.88475	0.990894	0.872125	0.992528
0.03	0.884419	0.990191	0.869166	0.991559
0.04	0.883923	0.989611	0.86462	0.991016
0.05	0.885248	0.989363	0.862416	0.990552
0.06	0.886242	0.988866	0.860415	0.989661
0.07	0.885083	0.988494	0.85564	0.988625
0.08	0.886408	0.987997	0.854139	0.987542
0.09	0.887071	0.987583	0.852461	0.986482
0.1	0.886078	0.987583	0.847962	0.986243
0.11	0.883097	0.987376	0.841484	0.985623
0.12	0.88144	0.986962	0.837152	0.984608
0.13	0.878626	0.986176	0.830842	0.982893
0.14	0.880282	0.986135	0.830372	0.982674
0.15	0.879122	0.985514	0.826542	0.981



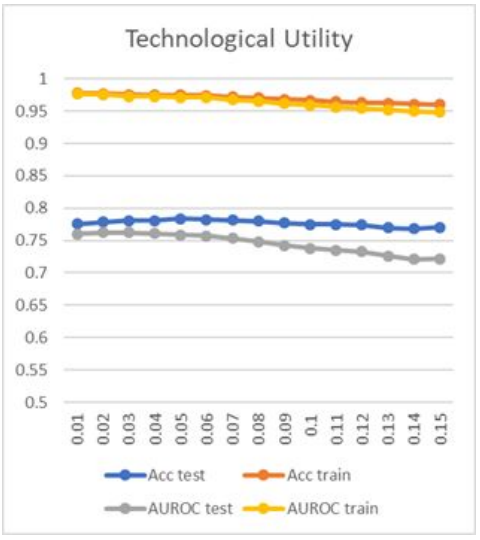
	Naïve bayes			
	Acc test	Acc train	AUROC te	AUROC tra
0.01	0.901578	0.991704	0.759968	0.995136
0.02	0.902801	0.991089	0.749818	0.994779
0.03	0.901571	0.991243	0.736702	0.994868
0.04	0.899724	0.991089	0.721248	0.99478
0.05	0.89788	0.992318	0.707878	0.994695
0.06	0.896034	0.992318	0.701138	0.994695
0.07	0.89542	0.992472	0.69278	0.994786
0.08	0.89542	0.992472	0.69278	0.994329
0.09	0.89419	0.993701	0.689227	0.993635
0.1	0.893574	0.994162	0.685656	0.990434
0.11	0.892347	0.993701	0.679773	0.988031
0.12	0.889888	0.993393	0.668022	0.985642
0.13	0.886197	0.99324	0.65421	0.982856
0.14	0.884966	0.993393	0.648999	0.981294
0.15	0.883124	0.992933	0.64123	0.977083

	Naïve bayes			
	Acc test	Acc train	AUROC te	AUROC tra
0.01	0.903466	0.991396	0.634376	0.995153
0.02	0.904691	0.99155	0.626089	0.991659
0.03	0.904689	0.991243	0.616815	0.989134
0.04	0.903456	0.992626	0.604774	0.988072
0.05	0.900997	0.992472	0.589443	0.984966
0.06	0.900381	0.993086	0.584371	0.984679
0.07	0.901612	0.992318	0.585054	0.980075
0.08	0.900997	0.992011	0.58249	0.974621
0.09	0.900997	0.992626	0.58249	0.971358
0.1	0.90161	0.992318	0.580628	0.969954
0.11	0.902226	0.991704	0.580977	0.964785
0.12	0.90407	0.990935	0.581999	0.959537
0.13	0.902843	0.990167	0.577453	0.956173
0.14	0.901614	0.988784	0.572478	0.949938
0.15	0.901614	0.987555	0.572478	0.94457

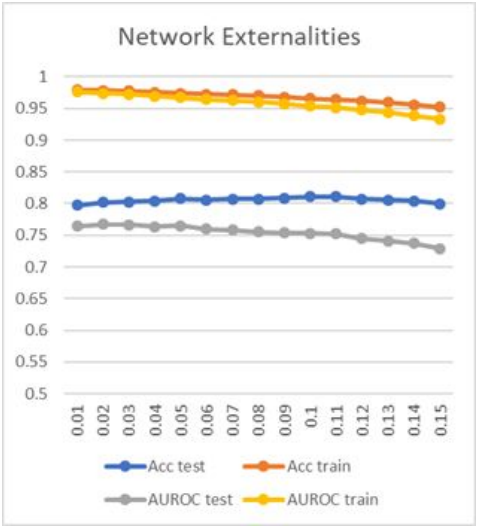


No Stopwords

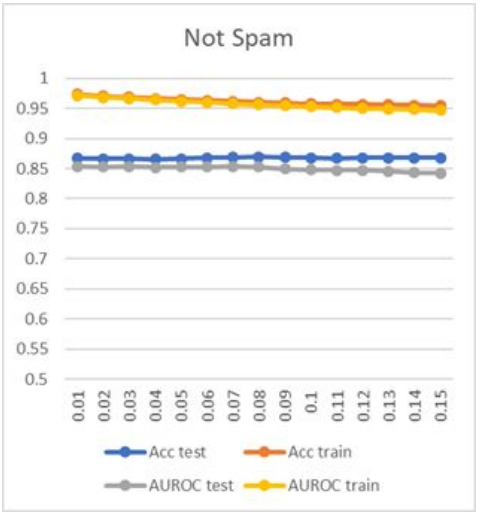
	Naïve bayes			
	Acc test	Acc train	AUROC te	AUROC tra
0.01	0.775524	0.978031	0.760033	0.976346
0.02	0.778601	0.977263	0.762385	0.975249
0.03	0.780455	0.975573	0.762552	0.972827
0.04	0.78107	0.975112	0.7606	0.972282
0.05	0.783536	0.974651	0.758879	0.971522
0.06	0.782303	0.974344	0.756895	0.970984
0.07	0.781691	0.971732	0.753589	0.967305
0.08	0.779849	0.970503	0.748306	0.965413
0.09	0.776774	0.968045	0.742678	0.961595
0.1	0.77493	0.96697	0.737918	0.95963
0.11	0.774928	0.964511	0.734996	0.956179
0.12	0.774311	0.963282	0.733116	0.95397
0.13	0.77001	0.962053	0.726068	0.951768
0.14	0.76816	0.960977	0.720965	0.950009
0.15	0.77062	0.960055	0.721612	0.948568



	Naïve bayes			
	Acc test	Acc train	AUROC te	AUROC tra
0.01	0.797065	0.979259	0.764027	0.975981
0.02	0.801986	0.978184	0.767464	0.974199
0.03	0.802597	0.977416	0.766182	0.972605
0.04	0.804442	0.975419	0.763624	0.969186
0.05	0.808132	0.974343	0.765036	0.967075
0.06	0.805667	0.972499	0.759606	0.964465
0.07	0.806896	0.971424	0.75811	0.962551
0.08	0.806898	0.970349	0.754894	0.960737
0.09	0.808746	0.968352	0.753844	0.957684
0.1	0.81059	0.965894	0.753037	0.953953
0.11	0.811203	0.964204	0.752586	0.951318
0.12	0.806898	0.962206	0.744939	0.947929
0.13	0.805671	0.959748	0.740746	0.944231
0.14	0.804444	0.956061	0.737103	0.938667
0.15	0.799522	0.952528	0.728683	0.933264

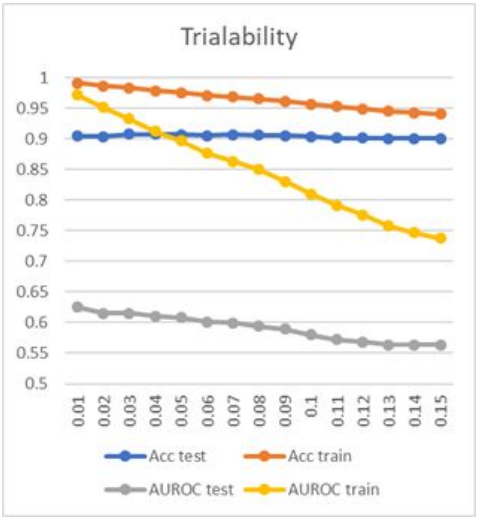


	Naïve bayes			
	Acc test	Acc train	AUROC te	AUROC tra
0.01	0.866703	0.973304	0.85318	0.971127
0.02	0.86604	0.97049	0.852906	0.967789
0.03	0.866371	0.968876	0.853565	0.966198
0.04	0.865544	0.966806	0.852136	0.963839
0.05	0.866372	0.964985	0.852484	0.961913
0.06	0.867697	0.963288	0.853021	0.959934
0.07	0.868856	0.96155	0.853193	0.957754
0.08	0.869684	0.960432	0.852389	0.95621
0.09	0.868359	0.959439	0.849692	0.95464
0.1	0.867697	0.958114	0.848375	0.952813
0.11	0.867034	0.957245	0.846993	0.951318
0.12	0.868028	0.956624	0.846931	0.950155
0.13	0.868194	0.955962	0.845716	0.948953
0.14	0.867697	0.955259	0.843035	0.947843
0.15	0.868028	0.954472	0.842557	0.946379



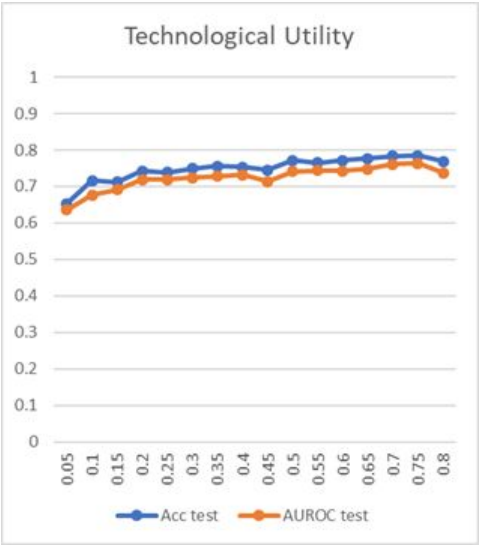
	Naïve bayes			
	Acc test	Acc train	AUROC te	AUROC tra
0.01	0.866703	0.973304	0.85318	0.971127
0.02	0.86604	0.97049	0.852906	0.967789
0.03	0.866371	0.968876	0.853565	0.966198
0.04	0.865544	0.966806	0.852136	0.963839
0.05	0.866372	0.964985	0.852484	0.961913
0.06	0.867697	0.963288	0.853021	0.959934
0.07	0.868856	0.96155	0.853193	0.957754
0.08	0.869684	0.960432	0.852389	0.95621
0.09	0.868359	0.959439	0.849692	0.95464
0.1	0.867697	0.958114	0.848375	0.952813
0.11	0.867034	0.957245	0.846993	0.951318
0.12	0.868028	0.956624	0.846931	0.950155
0.13	0.868194	0.955962	0.845716	0.948953
0.14	0.867697	0.955259	0.843035	0.947843
0.15	0.868028	0.954472	0.842557	0.946379

	Naïve bayes			
	Acc test	Acc train	AUROC te	AUROC tra
0.01	0.904687	0.990628	0.62537	0.971363
0.02	0.903458	0.986633	0.614805	0.951151
0.03	0.907762	0.982793	0.615323	0.932088
0.04	0.907147	0.978644	0.610267	0.911811
0.05	0.906533	0.975418	0.607994	0.896339
0.06	0.905306	0.970963	0.601093	0.876062
0.07	0.906537	0.968198	0.599405	0.863253
0.08	0.905922	0.965279	0.594349	0.8495
0.09	0.905306	0.961131	0.589277	0.830481
0.1	0.903462	0.956675	0.579819	0.809971
0.11	0.901616	0.952374	0.571988	0.790972
0.12	0.901616	0.94884	0.568322	0.775441
0.13	0.900387	0.944999	0.563485	0.757692
0.14	0.900387	0.942695	0.563485	0.746991
0.15	0.900387	0.940544	0.563485	0.73753

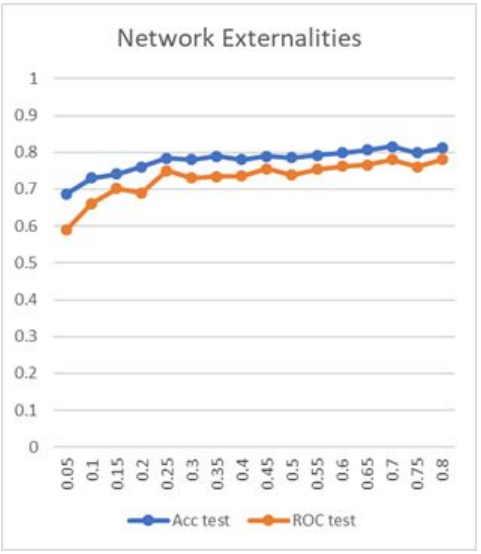


Dataset Size

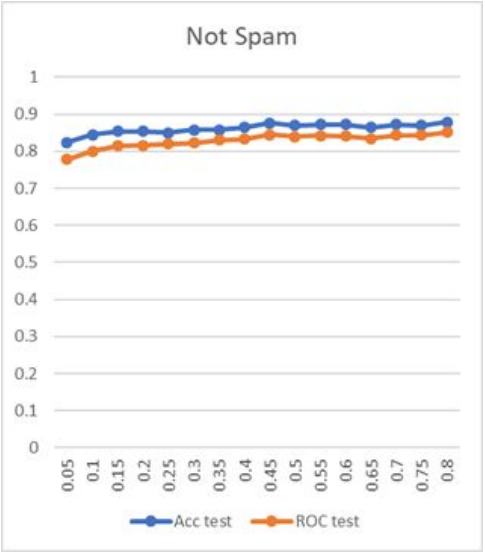
	Naïve bayes	
	Acc test	AUROC test
0.05	0.653412656	0.635958945
0.1	0.716530325	0.676734633
0.15	0.712536668	0.691473047
0.2	0.742587569	0.719222599
0.25	0.739194329	0.718553618
0.3	0.749771347	0.724367511
0.35	0.755996306	0.728446786
0.4	0.753613457	0.731884164
0.45	0.745230687	0.714248462
0.5	0.771595395	0.741912721
0.55	0.765243496	0.744210536
0.6	0.771306286	0.742779143
0.65	0.776835242	0.747907261
0.7	0.783284925	0.761208448
0.75	0.784160849	0.763679659
0.8	0.769574997	0.737618487



	Naïve bayes	
	Acc test	ROC test
0.05	0.68699688	0.590096876
0.1	0.730535167	0.661552274
0.15	0.740872583	0.702396973
0.2	0.760759296	0.690225366
0.25	0.784397243	0.750463649
0.3	0.7807208	0.730006156
0.35	0.789331592	0.734018025
0.4	0.780547285	0.736416192
0.45	0.789737742	0.755206579
0.5	0.785354178	0.738394242
0.55	0.791579124	0.75410128
0.6	0.798241908	0.762522032
0.65	0.806814362	0.766384574
0.7	0.81565313	0.779707418
0.75	0.798585278	0.761168835
0.8	0.811776529	0.780274793

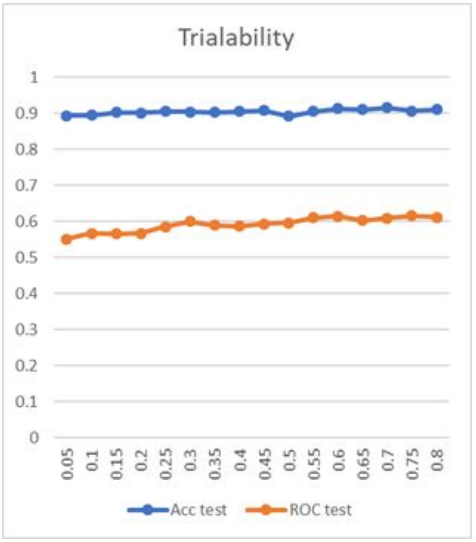


	Naïve bayes	
	Acc test	ROC test
0.05	0.823762318	0.777391443
0.1	0.845029783	0.80043882
0.15	0.853667581	0.814561334
0.2	0.853717154	0.815941602
0.25	0.850098187	0.819572728
0.3	0.85771011	0.822868368
0.35	0.858372636	0.829918535
0.4	0.863972332	0.83234849
0.45	0.875829016	0.844949037
0.5	0.869105193	0.839308503
0.55	0.871899052	0.842374429
0.6	0.872729501	0.841251204
0.65	0.864614828	0.833900077
0.7	0.871672449	0.843175446
0.75	0.869060103	0.843840917
0.8	0.878372882	0.851679596



	Naïve bayes	
	Acc test	AUROC test
0.05	0.872058497	0.606514583
0.1	0.868561342	0.592505706
0.15	0.862227294	0.568321556
0.2	0.879178828	0.617197764
0.25	0.884598255	0.639606966
0.3	0.887120685	0.657690328
0.35	0.890636208	0.66424608
0.4	0.887665229	0.643007803
0.45	0.903892062	0.696923995
0.5	0.88672649	0.652401288
0.55	0.893911104	0.666222329
0.6	0.902068065	0.677908797
0.65	0.894446475	0.659970431
0.7	0.900651638	0.707036197
0.75	0.900037593	0.697516311
0.8	0.913368791	0.721031859

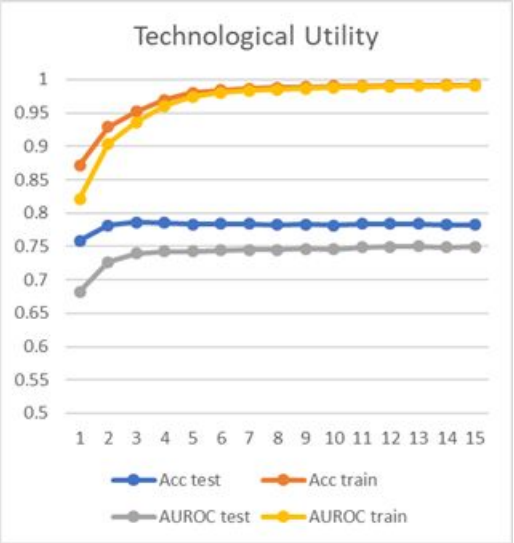
	Naïve bayes	
	Acc test	ROC test
0.05	0.89351047	0.550872335
0.1	0.894505721	0.567137363
0.15	0.901809318	0.565290688
0.2	0.900784645	0.567189226
0.25	0.904412154	0.584993625
0.3	0.903563778	0.599208366
0.35	0.901865819	0.588486835
0.4	0.90499812	0.585995123
0.45	0.907923633	0.593570376
0.5	0.891440751	0.595292468
0.55	0.90523846	0.609637031
0.6	0.913328172	0.613499753
0.65	0.910473923	0.601656744
0.7	0.915896269	0.609505613
0.75	0.906090146	0.614964736
0.8	0.910179553	0.611903333



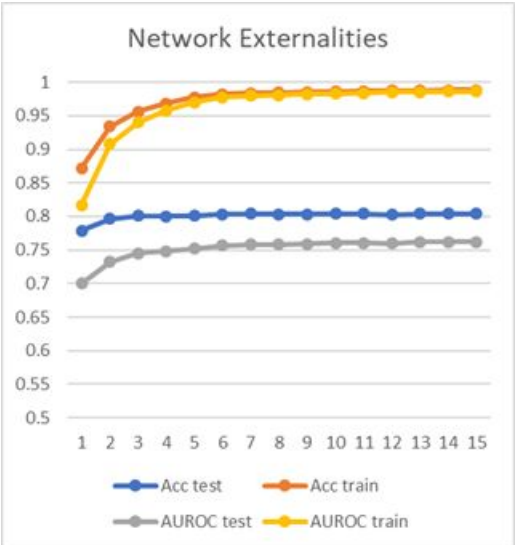
Logistic Regression

C-tuning

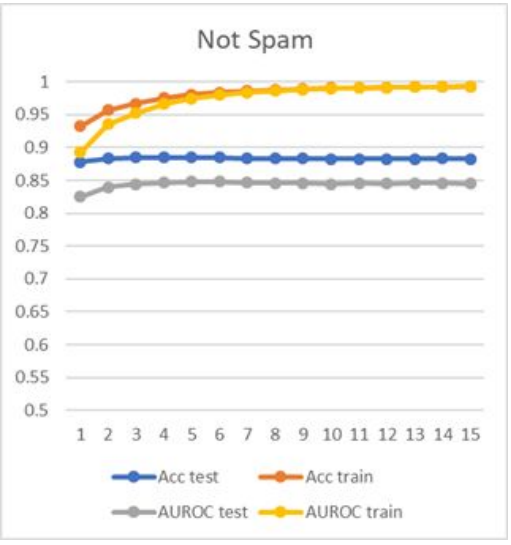
	Logistic regression			
	Acc test	Acc train	AUROC tes	AUROC trai
1	0.758341	0.872024	0.682186	0.821521
2	0.781714	0.929176	0.726545	0.903739
3	0.786018	0.951914	0.739843	0.935938
4	0.785399	0.969888	0.742953	0.960463
5	0.782939	0.980028	0.742816	0.974489
6	0.783551	0.984483	0.74444	0.980568
7	0.783553	0.986634	0.745206	0.983662
8	0.782324	0.987863	0.745184	0.985307
9	0.782939	0.989092	0.746529	0.986856
10	0.78171	0.990014	0.745914	0.988192
11	0.783555	0.990629	0.748502	0.989079
12	0.783556	0.990936	0.749395	0.98952
13	0.784172	0.991397	0.750425	0.990186
14	0.782328	0.99155	0.748947	0.990303
15	0.782326	0.992165	0.749286	0.991205



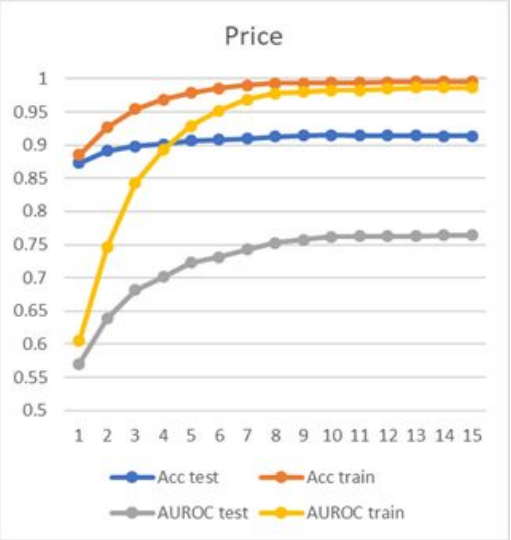
	Logistic regression			
	Acc test	Acc train	AUROC tes	AUROC trai
1	0.779203	0.87187	0.700297	0.817113
2	0.796434	0.934091	0.732319	0.907881
3	0.80074	0.956522	0.74498	0.939981
4	0.800125	0.968813	0.748459	0.957685
5	0.801355	0.977416	0.752227	0.970206
6	0.803198	0.982332	0.75687	0.977481
7	0.803813	0.983868	0.758263	0.979711
8	0.803198	0.98479	0.758329	0.980989
9	0.8032	0.985558	0.758748	0.982005
10	0.804429	0.986019	0.760762	0.982672
11	0.804429	0.986787	0.760762	0.983826
12	0.802582	0.987555	0.759386	0.984957
13	0.804427	0.987863	0.761898	0.985189
14	0.804429	0.98817	0.762317	0.985638
15	0.804429	0.988477	0.762358	0.986096



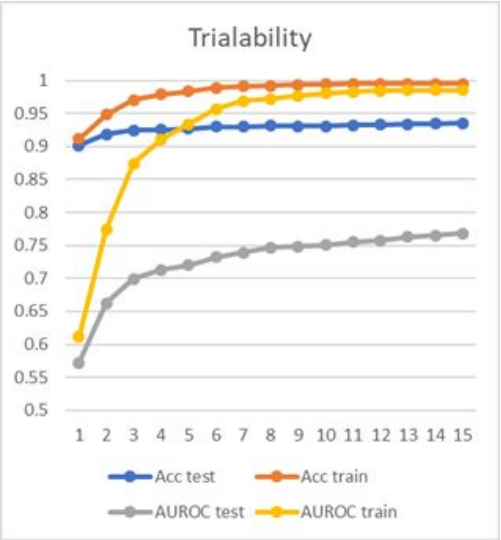
	Logistic regression			
	Acc test	Acc train	AUROC tes	AUROC trai
1	0.878294	0.932826	0.824881	0.892981
2	0.883592	0.956748	0.839578	0.935827
3	0.885082	0.966889	0.844632	0.952346
4	0.884916	0.975539	0.846803	0.966145
5	0.884917	0.980713	0.848054	0.974728
6	0.884751	0.984024	0.847818	0.979978
7	0.883592	0.9863	0.846268	0.983596
8	0.883261	0.987914	0.846253	0.98624
9	0.883095	0.989611	0.846029	0.988767
10	0.882433	0.990605	0.844672	0.990108
11	0.88293	0.990977	0.845689	0.990702
12	0.882433	0.991515	0.845109	0.991221
13	0.88293	0.992012	0.845557	0.991991
14	0.883261	0.992426	0.846119	0.992573
15	0.882598	0.992757	0.844766	0.992901



	Logistic regression			
	Acc test	Acc train	AUROC tes	AUROC trai
1	0.873284	0.885542	0.569985	0.604918
2	0.891741	0.926562	0.638531	0.745938
3	0.898505	0.954524	0.681646	0.843411
4	0.901576	0.968658	0.701074	0.893108
5	0.906499	0.979106	0.722883	0.929225
6	0.908345	0.985558	0.730983	0.952242
7	0.910192	0.990475	0.742603	0.969052
8	0.912651	0.993086	0.752422	0.978236
9	0.914497	0.993701	0.757642	0.980369
10	0.915111	0.994008	0.762102	0.982282
11	0.914496	0.994008	0.763026	0.982282
12	0.914496	0.994776	0.763026	0.984875
13	0.914496	0.995391	0.763026	0.986486
14	0.91388	0.995391	0.763992	0.986486
15	0.91388	0.995391	0.763992	0.986486



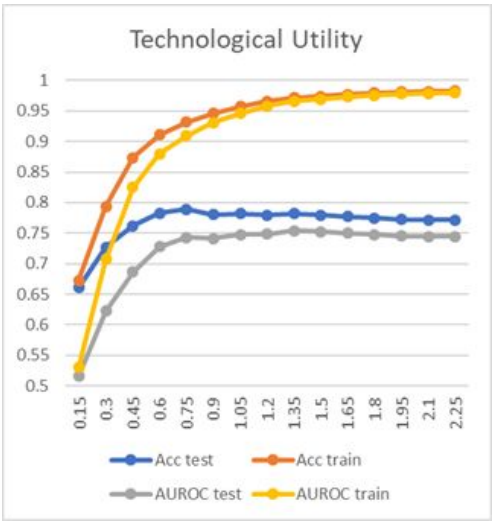
	Logistic regression			
	Acc test	Acc train	AUROC tes	AUROC trai
1	0.901612	0.911354	0.571994	0.610855
2	0.918222	0.948225	0.662723	0.774346
3	0.924989	0.970503	0.699011	0.873715
4	0.925605	0.978799	0.713347	0.910604
5	0.926833	0.983868	0.71962	0.934115
6	0.929907	0.988938	0.732135	0.956775
7	0.929907	0.991704	0.739058	0.969055
8	0.931751	0.992472	0.746598	0.972472
9	0.930524	0.993394	0.748258	0.976573
10	0.931139	0.994315	0.75096	0.980657
11	0.932366	0.99493	0.755367	0.982843
12	0.93298	0.995237	0.75764	0.984212
13	0.934209	0.995391	0.762615	0.984932
14	0.934824	0.995391	0.765318	0.984932
15	0.935439	0.995391	0.768021	0.984932



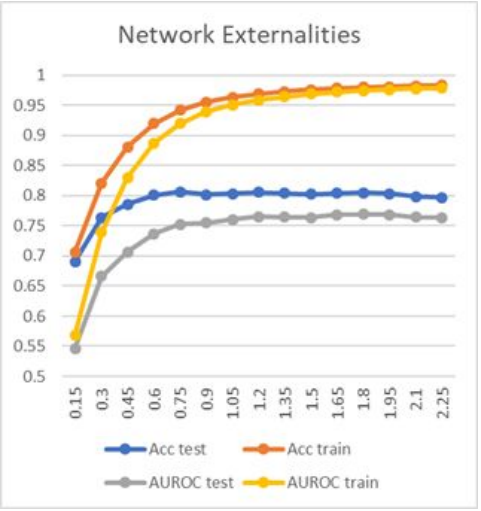
SVM (linear)

C-tuning

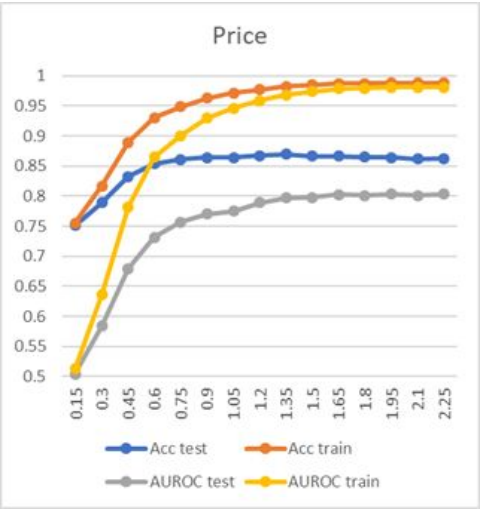
	SVM			
	Acc test	Acc train	AUROC te	AUROC tr
0.15	0.661157	0.672915	0.515793	0.529756
0.3	0.726971	0.794133	0.622465	0.706838
0.45	0.762031	0.873563	0.686451	0.824951
0.6	0.782329	0.910741	0.72746	0.879621
0.75	0.789087	0.931482	0.742989	0.909516
0.9	0.780476	0.94623	0.741554	0.930613
1.05	0.782324	0.957445	0.747702	0.946411
1.2	0.779247	0.965741	0.748469	0.957796
1.35	0.781703	0.971886	0.753726	0.966377
1.5	0.779243	0.974344	0.752386	0.96931
1.65	0.777395	0.977263	0.749849	0.973201
1.8	0.774935	0.979567	0.747976	0.97599
1.95	0.772476	0.981104	0.745438	0.978013
2.1	0.771858	0.981872	0.744549	0.979015
2.25	0.771243	0.98264	0.744846	0.979815



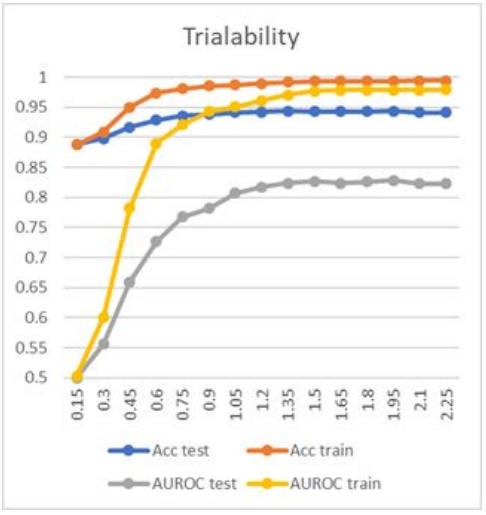
	SVM			
	Acc test	Acc train	AUROC te	AUROC tra
0.15	0.690637	0.706867	0.546272	0.568575
0.3	0.763826	0.820711	0.666076	0.739182
0.45	0.785975	0.881242	0.706882	0.830268
0.6	0.800738	0.920265	0.737089	0.88776
0.75	0.80628	0.942541	0.75281	0.920387
0.9	0.801973	0.9556	0.754716	0.939531
1.05	0.803821	0.963435	0.760294	0.950856
1.2	0.805669	0.96912	0.765157	0.958763
1.35	0.804436	0.972961	0.764956	0.964412
1.5	0.802592	0.976494	0.76398	0.969516
1.65	0.804434	0.978338	0.768316	0.972229
1.8	0.805046	0.980335	0.769679	0.97507
1.95	0.8032	0.981103	0.768194	0.976195
2.1	0.798894	0.982486	0.764407	0.978234
2.25	0.797051	0.9831	0.763599	0.978931



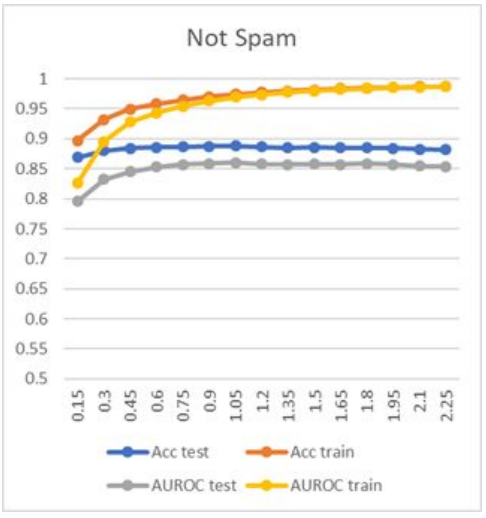
	SVM			
	Acc test	Acc train	AUROC te	AUROC tra
0.15	0.751528	0.755569	0.503861	0.512832
0.3	0.78905	0.817175	0.584309	0.636313
0.45	0.832723	0.889844	0.679223	0.782364
0.6	0.853622	0.930864	0.731617	0.864771
0.75	0.860997	0.948839	0.756656	0.900913
0.9	0.864066	0.963281	0.7701	0.929975
1.05	0.864068	0.97127	0.775406	0.946407
1.2	0.867755	0.977415	0.789	0.959119
1.35	0.869603	0.982332	0.797009	0.968489
1.5	0.866535	0.985097	0.798082	0.974179
1.65	0.866535	0.987094	0.802543	0.978374
1.8	0.864693	0.987555	0.801357	0.979496
1.95	0.864074	0.98817	0.803426	0.980898
2.1	0.862227	0.98817	0.801372	0.981115
2.25	0.862841	0.98817	0.803537	0.981303



	SVM			
	Acc test	Acc train	AUROC te	AUROC tr
0.15	0.888083	0.888002	0.5	0.502878
0.3	0.897925	0.909203	0.556047	0.600274
0.45	0.916993	0.949916	0.659461	0.781995
0.6	0.928674	0.973575	0.726436	0.890141
0.75	0.935437	0.980489	0.767743	0.921072
0.9	0.938509	0.985559	0.782198	0.943039
1.05	0.941586	0.987402	0.807391	0.951009
1.2	0.942201	0.989706	0.817017	0.961276
1.35	0.943428	0.991857	0.823931	0.970886
1.5	0.942815	0.99324	0.827243	0.97719
1.65	0.942815	0.993547	0.823577	0.978507
1.8	0.942815	0.993547	0.825808	0.978507
1.95	0.94343	0.993547	0.828494	0.978507
2.1	0.941588	0.993855	0.823164	0.978645
2.25	0.940974	0.994008	0.822809	0.979278

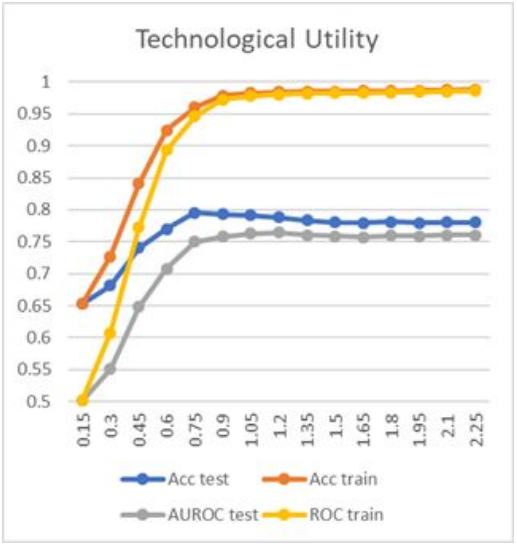


	SVM			
	Acc test	Acc train	AUROC te	AUROC tra
0.15	0.869186	0.896735	0.795705	0.826322
0.3	0.880612	0.932081	0.832788	0.89556
0.45	0.884585	0.949629	0.844918	0.92785
0.6	0.88591	0.958528	0.852851	0.943558
0.75	0.886903	0.964819	0.857082	0.954323
0.9	0.8874	0.970324	0.859032	0.963264
1.05	0.888062	0.974422	0.860373	0.969366
1.2	0.886407	0.977319	0.858228	0.97358
1.35	0.885247	0.980133	0.857397	0.977642
1.5	0.885413	0.981623	0.858259	0.979787
1.65	0.88475	0.983527	0.857278	0.9822
1.8	0.885082	0.984645	0.858313	0.983886
1.95	0.884088	0.985762	0.857191	0.985373
2.1	0.882432	0.986631	0.854895	0.986554
2.25	0.88177	0.987294	0.854044	0.987257

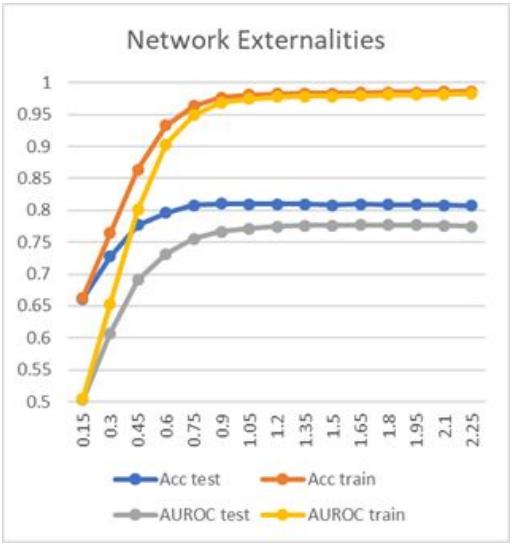


N-gram range = (1, 2)

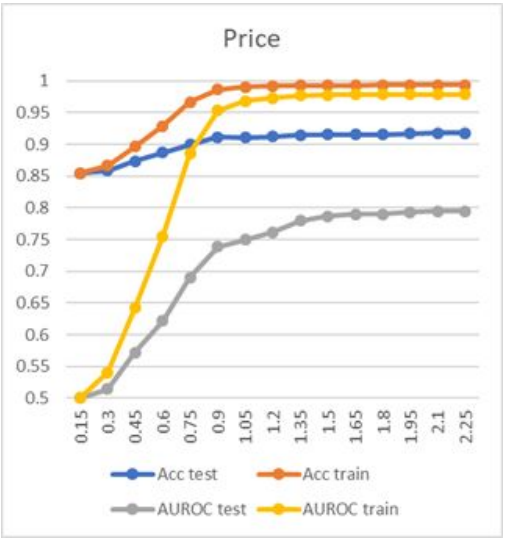
	SVM (ngram = 2)			
	Acc test	Acc train	AUROC te	ROC train
0.15	0.652547	0.652944	0.501031	0.501102
0.3	0.681455	0.727148	0.550644	0.607499
0.45	0.740491	0.841297	0.648226	0.772505
0.6	0.77002	0.924414	0.707667	0.893501
0.75	0.795853	0.960824	0.749726	0.9461
0.9	0.793387	0.978799	0.758109	0.972532
1.05	0.791543	0.982486	0.76293	0.978348
1.2	0.788459	0.984023	0.764083	0.980646
1.35	0.783547	0.985098	0.760798	0.982087
1.5	0.78047	0.985252	0.758443	0.982414
1.65	0.779239	0.985713	0.757467	0.982871
1.8	0.781084	0.98602	0.759352	0.98332
1.95	0.779237	0.986788	0.758425	0.984418
2.1	0.780468	0.987556	0.760183	0.985412
2.25	0.780468	0.988017	0.760183	0.985862



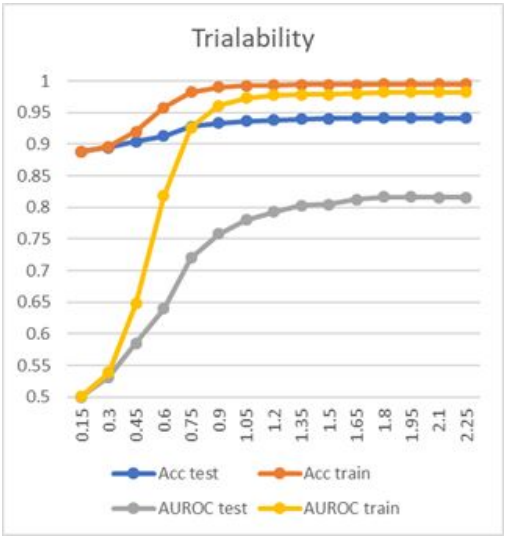
	SVM (ngram = 2)			
	Acc test	Acc train	AUROC te	AUROC tra
0.15	0.661735	0.663388	0.502831	0.504766
0.3	0.727538	0.764482	0.606957	0.653406
0.45	0.776749	0.863883	0.691057	0.800733
0.6	0.795821	0.932864	0.731702	0.903335
0.75	0.808125	0.963435	0.755564	0.949194
0.9	0.810588	0.977262	0.766328	0.969304
1.05	0.809967	0.980642	0.771098	0.974549
1.2	0.809355	0.982639	0.774296	0.97749
1.35	0.809361	0.9831	0.775973	0.978271
1.5	0.808128	0.983561	0.775905	0.978731
1.65	0.809359	0.984329	0.777209	0.979764
1.8	0.808744	0.985251	0.776816	0.981126
1.95	0.808744	0.985405	0.776816	0.981362
2.1	0.808128	0.985712	0.775907	0.981816
2.25	0.807515	0.986326	0.774945	0.982693



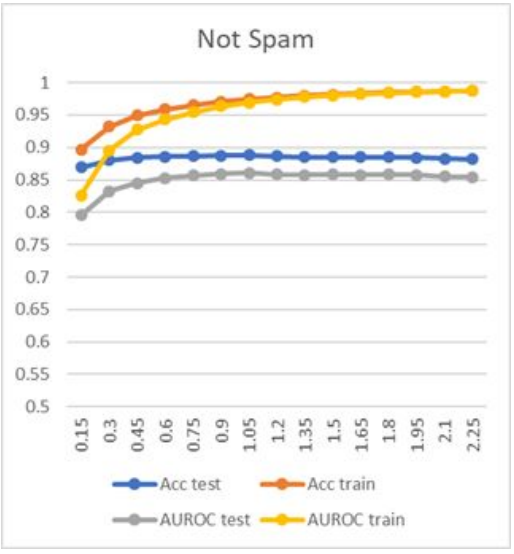
	SVM			
	Acc test	Acc train	AUROC te	AUROC tra
0.15	0.854833	0.854968	0.5	0.5
0.3	0.858522	0.866798	0.514647	0.540627
0.45	0.873899	0.896296	0.571872	0.64256
0.6	0.886816	0.92902	0.62214	0.755452
0.75	0.899736	0.967121	0.690179	0.886677
0.9	0.91142	0.98648	0.739085	0.953563
1.05	0.910803	0.990782	0.749745	0.968669
1.2	0.912032	0.992164	0.761709	0.973348
1.35	0.91449	0.993086	0.779747	0.976913
1.5	0.915721	0.99324	0.78719	0.977963
1.65	0.915719	0.99324	0.789795	0.978368
1.8	0.915719	0.993393	0.789795	0.978848
1.95	0.91695	0.993393	0.793128	0.978848
2.1	0.917565	0.993393	0.795015	0.978848
2.25	0.917565	0.993393	0.795015	0.978848



	SVM			
	Acc test	Acc train	AUROC te	AUROC tra
0.15	0.888083	0.887541	0.5	0.500719
0.3	0.89485	0.896145	0.531479	0.539091
0.45	0.904077	0.920111	0.584978	0.647841
0.6	0.913303	0.958212	0.639647	0.818216
0.75	0.928062	0.982178	0.720524	0.926086
0.9	0.933597	0.990013	0.758315	0.960756
1.05	0.936664	0.992779	0.780607	0.973253
1.2	0.937897	0.993701	0.792947	0.977414
1.35	0.939741	0.994008	0.802855	0.978732
1.5	0.940355	0.994008	0.805128	0.978732
1.65	0.940972	0.994316	0.812393	0.980084
1.8	0.941588	0.994777	0.816393	0.982183
1.95	0.941588	0.994777	0.816393	0.982183
2.1	0.940974	0.994777	0.816039	0.982183
2.25	0.940974	0.994777	0.816039	0.982183

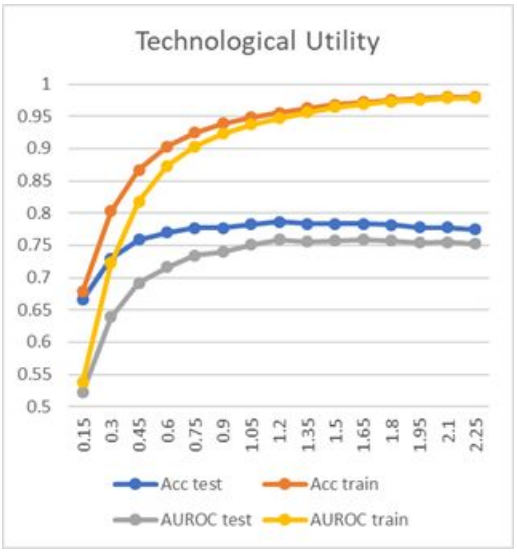


	SVM (ngram = 2)			
	Acc test	Acc train	AUROC te	AUROC tra
0.15	0.869186	0.896735	0.795705	0.826322
0.3	0.880612	0.932081	0.832788	0.89556
0.45	0.884585	0.949629	0.844918	0.92785
0.6	0.88591	0.958528	0.852851	0.943558
0.75	0.886903	0.964819	0.857082	0.954323
0.9	0.8874	0.970324	0.859032	0.963264
1.05	0.888062	0.974422	0.860373	0.969366
1.2	0.886407	0.977319	0.858228	0.97358
1.35	0.885247	0.980133	0.857397	0.977642
1.5	0.885413	0.981623	0.858259	0.979787
1.65	0.88475	0.983527	0.857278	0.9822
1.8	0.885082	0.984645	0.858313	0.983886
1.95	0.884088	0.985762	0.857191	0.985373
2.1	0.882432	0.986631	0.854895	0.986554
2.25	0.88177	0.987294	0.854044	0.987257

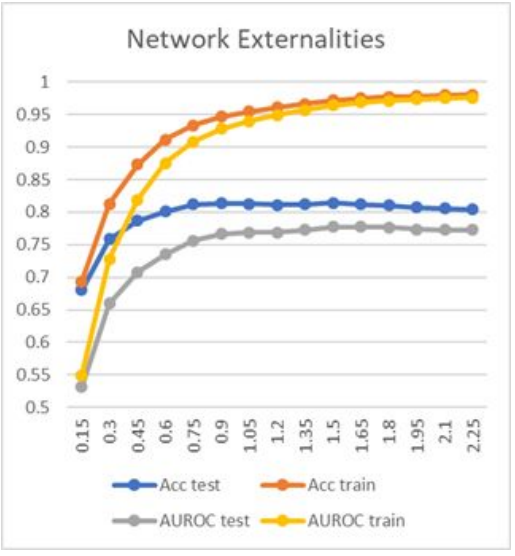


No Stopwords

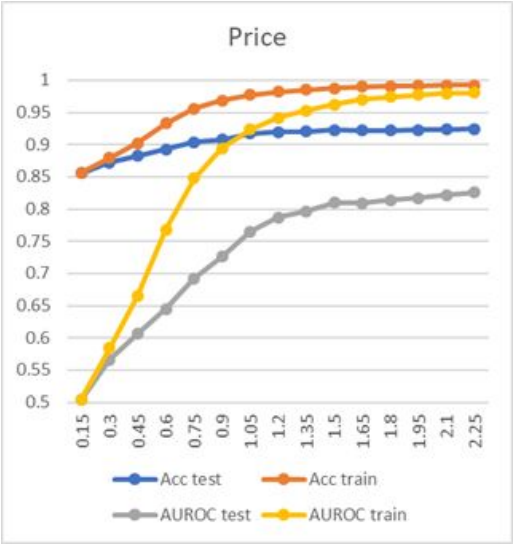
	SVM			
	Acc test	Acc train	AUROC te	AUROC tra
0.15	0.665465	0.6786	0.521834	0.537834
0.3	0.729435	0.802889	0.638992	0.722471
0.45	0.75895	0.867108	0.691433	0.81829
0.6	0.770016	0.903674	0.716404	0.872964
0.75	0.77678	0.92426	0.734173	0.902957
0.9	0.776783	0.938548	0.739905	0.923449
1.05	0.782314	0.94838	0.751083	0.937006
1.2	0.786618	0.955448	0.758511	0.946625
1.35	0.782928	0.962668	0.756046	0.956226
1.5	0.783541	0.968045	0.757011	0.963794
1.65	0.783547	0.971733	0.759022	0.968564
1.8	0.781703	0.975266	0.757382	0.972908
1.95	0.777399	0.97757	0.754085	0.9755
2.1	0.777399	0.979721	0.754582	0.977986
2.25	0.774939	0.980182	0.752778	0.978652



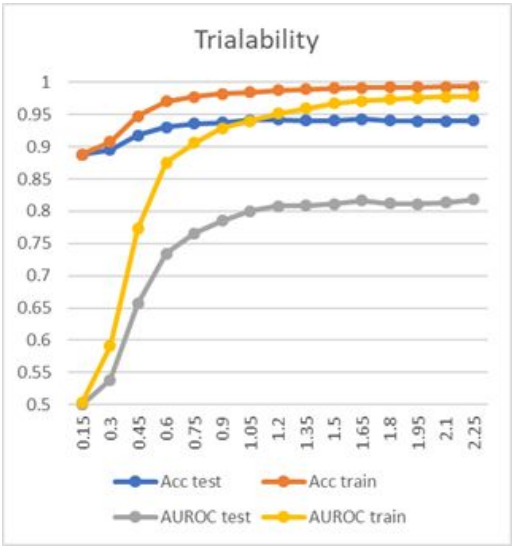
	SVM			
	Acc test	Acc train	AUROC te	AUROC tra
0.15	0.680183	0.693808	0.53102	0.549409
0.3	0.758907	0.812723	0.659608	0.728016
0.45	0.7872	0.872946	0.707865	0.818832
0.6	0.800736	0.911355	0.73555	0.875606
0.75	0.811811	0.933477	0.756519	0.908257
0.9	0.813665	0.946843	0.766183	0.92805
1.05	0.813048	0.954678	0.769192	0.939526
1.2	0.811202	0.961284	0.769264	0.949133
1.35	0.81243	0.966815	0.77248	0.956803
1.5	0.814278	0.972346	0.777926	0.964604
1.65	0.811821	0.975265	0.777602	0.968801
1.8	0.810592	0.977416	0.77638	0.971546
1.95	0.807517	0.978645	0.773915	0.973354
2.1	0.805671	0.980028	0.772501	0.97541
2.25	0.803824	0.980796	0.772612	0.976318



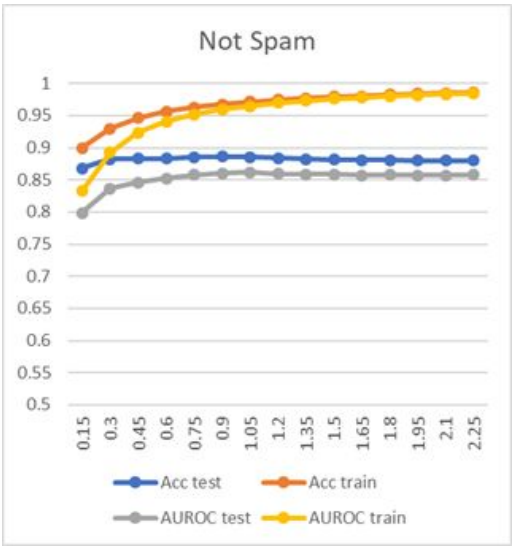
	SVM			
	Acc test	Acc train	AUROC te	AUROC tra
0.15	0.855449	0.856351	0.503571	0.504704
0.3	0.872055	0.879243	0.565625	0.583752
0.45	0.882511	0.902595	0.605952	0.664348
0.6	0.89297	0.932708	0.644962	0.767783
0.75	0.904038	0.955906	0.69193	0.847999
0.9	0.907724	0.968965	0.726198	0.894049
1.05	0.916948	0.977261	0.764955	0.923926
1.2	0.920019	0.982178	0.787045	0.941732
1.35	0.92063	0.985251	0.79687	0.953098
1.5	0.923094	0.987862	0.809796	0.96246
1.65	0.921863	0.98986	0.809088	0.970082
1.8	0.921863	0.990782	0.8138	0.973863
1.95	0.922479	0.991857	0.816756	0.977607
2.1	0.923707	0.992472	0.821584	0.979596
2.25	0.924321	0.992625	0.825828	0.980142



	SVM			
	Acc test	Acc train	AUROC te	AUROC tra
0.15	0.888083	0.888002	0.5	0.502878
0.3	0.894849	0.907359	0.538002	0.590701
0.45	0.918222	0.948072	0.656842	0.772526
0.6	0.93052	0.970656	0.734558	0.875563
0.75	0.936051	0.97757	0.765881	0.906311
0.9	0.937893	0.982639	0.785501	0.928763
1.05	0.941582	0.984944	0.799959	0.940101
1.2	0.942197	0.987555	0.80823	0.951834
1.35	0.940966	0.989245	0.809014	0.959836
1.5	0.940966	0.990935	0.811384	0.96746
1.65	0.942811	0.991857	0.816692	0.971562
1.8	0.940966	0.992318	0.81199	0.973651
1.95	0.939736	0.992779	0.811324	0.975663
2.1	0.939738	0.99324	0.813533	0.977735
2.25	0.940353	0.993394	0.818605	0.97842

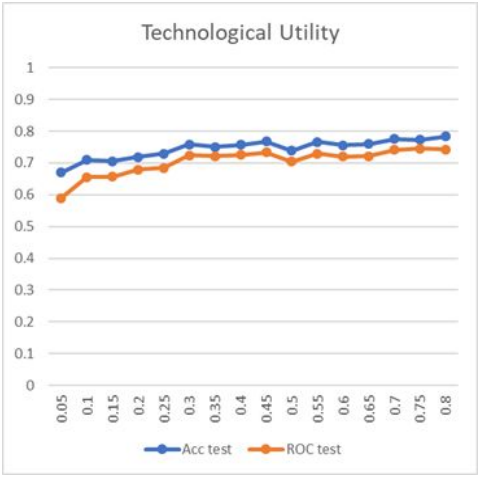


	SVM			
	Acc test	Acc train	AUROC te	AUROC tra
0.15	0.869186	0.896735	0.795705	0.826322
0.3	0.880612	0.932081	0.832788	0.89556
0.45	0.884585	0.949629	0.844918	0.92785
0.6	0.88591	0.958528	0.852851	0.943558
0.75	0.886903	0.964819	0.857082	0.954323
0.9	0.8874	0.970324	0.859032	0.963264
1.05	0.888062	0.974422	0.860373	0.969366
1.2	0.886407	0.977319	0.858228	0.97358
1.35	0.885247	0.980133	0.857397	0.977642
1.5	0.885413	0.981623	0.858259	0.979787
1.65	0.88475	0.983527	0.857278	0.9822
1.8	0.885082	0.984645	0.858313	0.983886
1.95	0.884088	0.985762	0.857191	0.985373
2.1	0.882432	0.986631	0.854895	0.986554
2.25	0.88177	0.987294	0.854044	0.987257

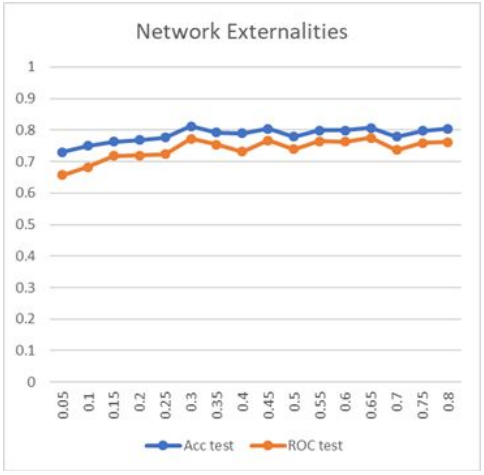


Dataset Size

	SVM	
	Acc test	ROC test
0.05	0.669286846	0.589385606
0.1	0.710024527	0.654904696
0.15	0.70502553	0.656138297
0.2	0.718059747	0.67970173
0.25	0.729045297	0.683750849
0.3	0.758231761	0.724586417
0.35	0.749806571	0.721023688
0.4	0.757226503	0.725394744
0.45	0.767802429	0.733425997
0.5	0.737954204	0.703543834
0.55	0.766062539	0.729567971
0.6	0.755763508	0.720337621
0.65	0.759545905	0.721478762
0.7	0.775553789	0.740484611
0.75	0.772901757	0.744807254
0.8	0.783430226	0.742498255

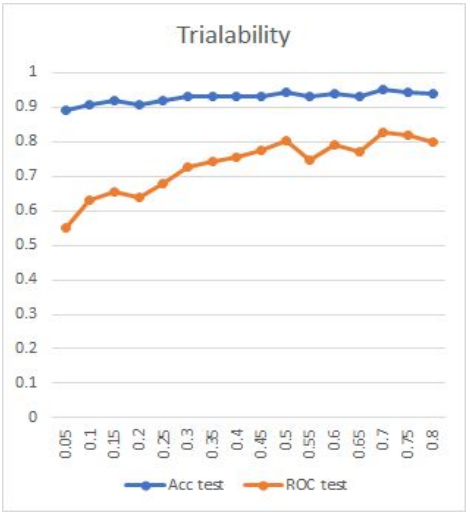


	SVM	
	Acc test	ROC test
0.05	0.730105808	0.656839997
0.1	0.749834801	0.681887538
0.15	0.762614994	0.717646292
0.2	0.768722339	0.718793432
0.25	0.77632751	0.723210872
0.3	0.811545365	0.771737287
0.35	0.792409822	0.753133949
0.4	0.788969606	0.731719117
0.45	0.80377522	0.767349898
0.5	0.779013384	0.738519281
0.55	0.799438035	0.764207599
0.6	0.79851994	0.762913936
0.65	0.806676683	0.775517746
0.7	0.779302599	0.735996414
0.75	0.797693501	0.759661697
0.8	0.803923296	0.761046105

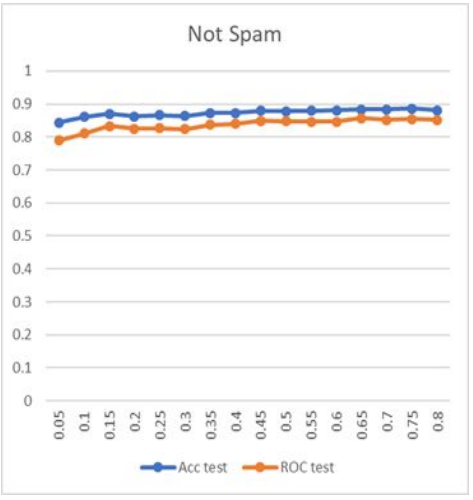


	SVM	
	Acc test	ROC test
0.05	0.847336439	0.520406684
0.1	0.872984408	0.586207626
0.15	0.876888698	0.585922186
0.2	0.901323614	0.654526348
0.25	0.895879207	0.672314634
0.3	0.902935796	0.676449865
0.35	0.890756557	0.664345882
0.4	0.910049925	0.705731037
0.45	0.913274602	0.730470417
0.5	0.917028296	0.755401439
0.55	0.90180273	0.712778937
0.6	0.913680072	0.74670205
0.65	0.925503335	0.764922761
0.7	0.90043083	0.719682816
0.75	0.913662339	0.754831837
0.8	0.907835626	0.738940033

	SVM	
	Acc test	ROC test
0.05	0.889909912	0.549878553
0.1	0.906161666	0.629981039
0.15	0.919268921	0.654644705
0.2	0.908108819	0.636579731
0.25	0.91959728	0.678555861
0.3	0.930320532	0.726581641
0.35	0.929855399	0.743572932
0.4	0.929224783	0.753060727
0.45	0.930639939	0.773995634
0.5	0.943064963	0.804026699
0.55	0.928557016	0.746510541
0.6	0.938553503	0.7912883
0.65	0.930525156	0.768913563
0.7	0.948865371	0.825346764
0.75	0.943412599	0.819635092
0.8	0.939629005	0.797649488



	SVM	
	Acc test	ROC test
0.05	0.843259302	0.789789907
0.1	0.860476438	0.810761618
0.15	0.869729073	0.832966276
0.2	0.862323413	0.824696085
0.25	0.866249232	0.826105154
0.3	0.863170947	0.823592722
0.35	0.87255805	0.837759172
0.4	0.873177951	0.839122004
0.45	0.879860073	0.848695559
0.5	0.878060928	0.847830758
0.55	0.880008792	0.845812169
0.6	0.880337726	0.846968076
0.65	0.883684727	0.857376384
0.7	0.882842259	0.851066303
0.75	0.885779118	0.854181219
0.8	0.880502572	0.85102273

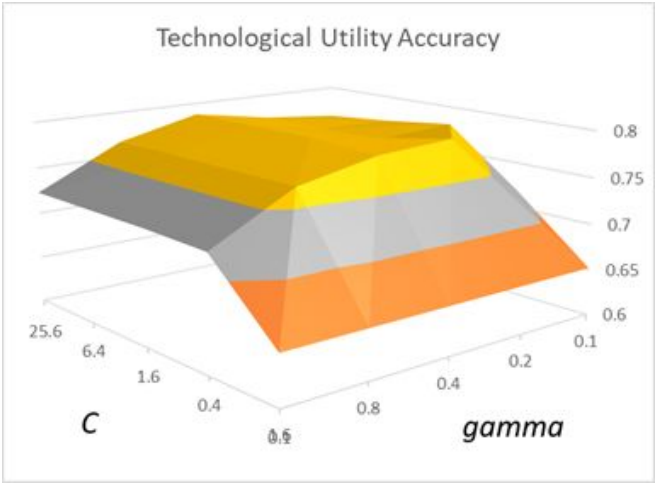


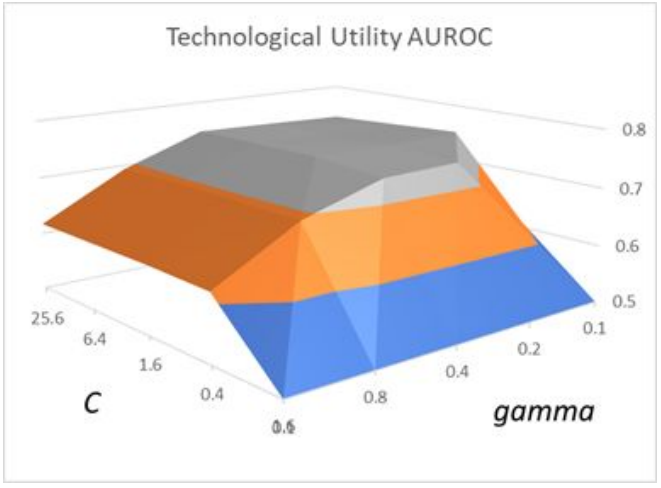
SVM (rbf)

Tuning (C and gamma)

	SVM Acc				
	0.1	0.4	1.6	6.4	25.6
0.1	0.651932	0.719585	0.779857	0.770628	0.763864
0.2	0.651932	0.789095	0.774937	0.772478	0.772478
0.4	0.651932	0.784166	0.787855	0.787857	0.787243
0.8	0.651932	0.766937	0.766935	0.767551	0.76632
1.6	0.651932	0.7208	0.722647	0.722647	0.722647

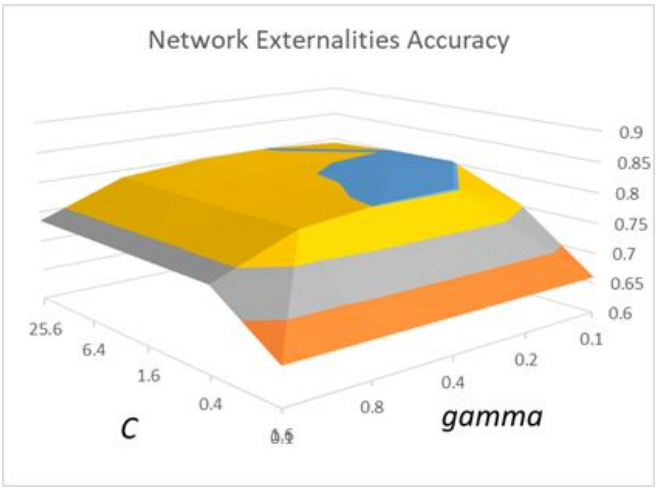
	SVM AUROC				
	0.1	0.4	1.6	6.4	25.6
0.1	0.5	0.609012	0.750379	0.743504	0.737379
0.2	0.5	0.739706	0.744914	0.741785	0.741136
0.4	0.5	0.735856	0.744785	0.744951	0.74407
0.8	0.5	0.693179	0.694597	0.694834	0.693167
1.6	0.5	0.614023	0.616854	0.616854	0.616854

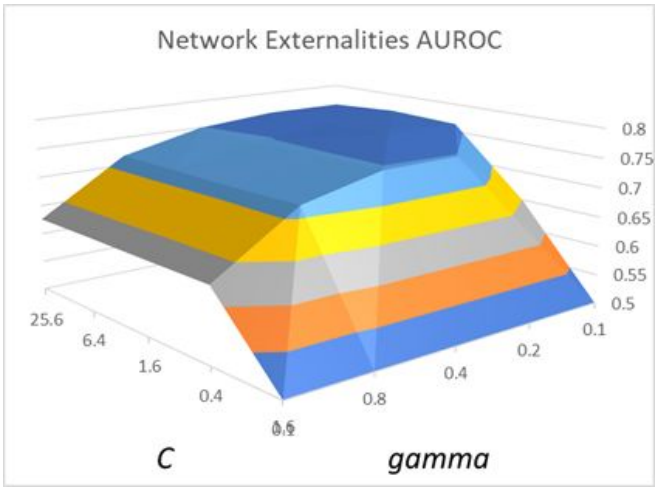




	SVM Acc				
	0.1	0.4	1.6	6.4	25.6
0.1	0.659889	0.75337	0.803205	0.800123	0.79213
0.2	0.659889	0.803203	0.803205	0.799513	0.800126
0.4	0.659889	0.801357	0.798284	0.797053	0.798901
0.8	0.659889	0.787832	0.785986	0.787217	0.787217
1.6	0.659889	0.734309	0.734309	0.734924	0.734924

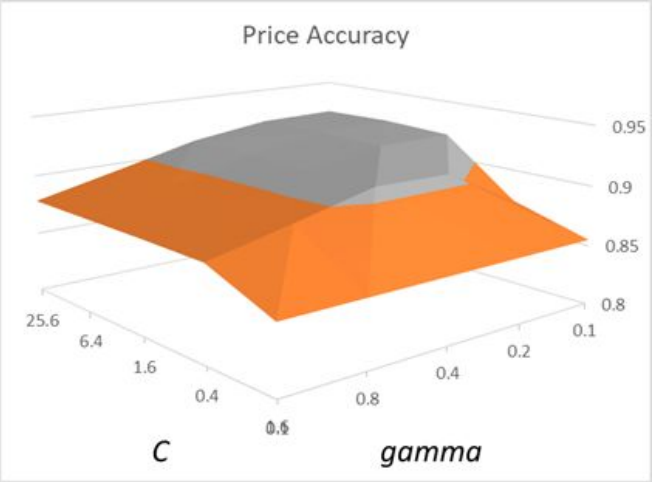
	SVM AUROC				
	0.1	0.4	1.6	6.4	25.6
0.1	0.5	0.649258	0.76354	0.768133	0.761007
0.2	0.5	0.748895	0.76503	0.760456	0.76126
0.4	0.5	0.747984	0.747681	0.747019	0.749371
0.8	0.5	0.716523	0.71491	0.716343	0.716642
1.6	0.5	0.624854	0.624854	0.625763	0.625763

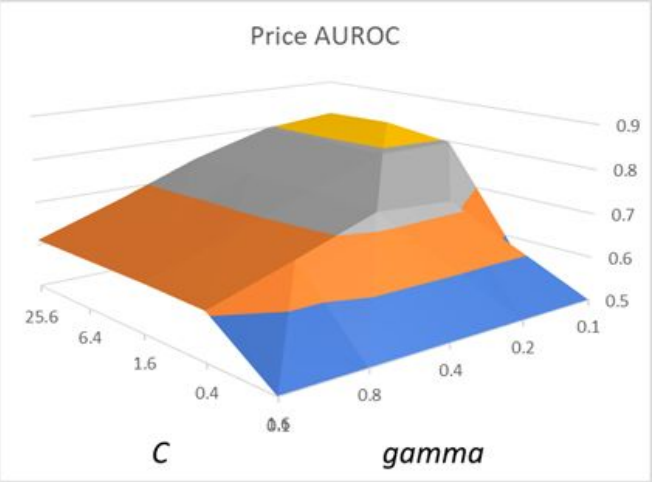




	SVM Acc				
	0.1	0.4	1.6	6.4	25.6
0.1	0.854833	0.873284	0.920025	0.922486	0.921255
0.2	0.854833	0.907724	0.919409	0.920025	0.920025
0.4	0.854833	0.908953	0.90834	0.908955	0.910798
0.8	0.854833	0.892961	0.89419	0.89419	0.89419
1.6	0.854833	0.877591	0.878207	0.878207	0.878207

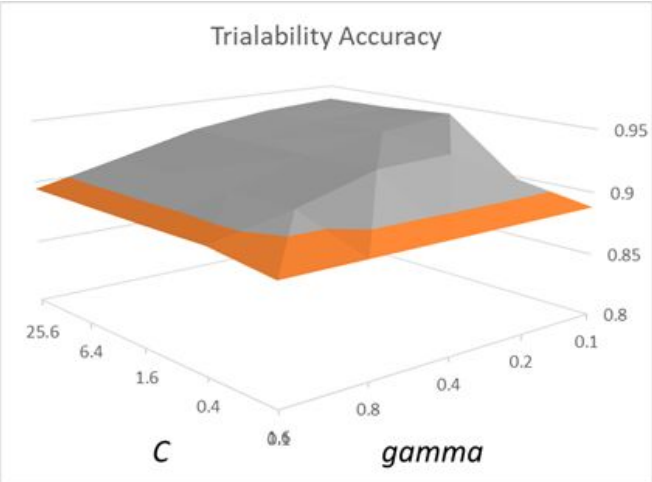
	SVM AUROC				
	0.1	0.4	1.6	6.4	25.6
0.1	0.5	0.5683	0.797615	0.818829	0.815496
0.2	0.5	0.722205	0.796059	0.797725	0.797725
0.4	0.5	0.732652	0.740363	0.74225	0.745911
0.8	0.5	0.668141	0.674654	0.674654	0.674654
1.6	0.5	0.606803	0.610374	0.610374	0.610374

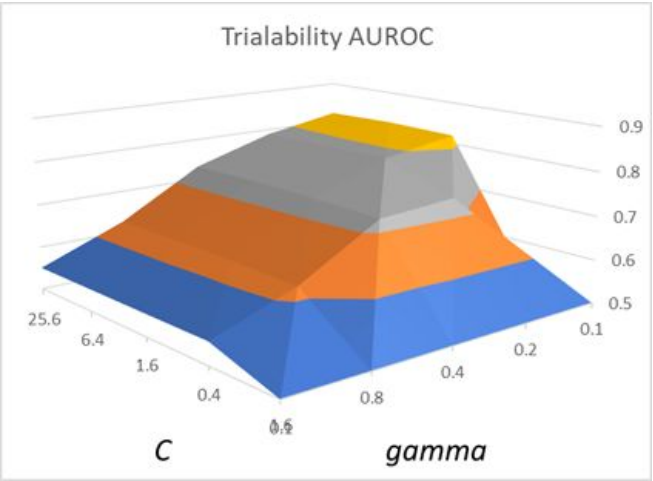




	SVM Acc				
	0.1	0.4	1.6	6.4	25.6
0.1	0.888083	0.897925	0.942815	0.939134	0.937903
0.2	0.888083	0.928674	0.936057	0.934828	0.934214
0.4	0.888083	0.92683	0.925603	0.925605	0.925605
0.8	0.888083	0.908381	0.908997	0.909612	0.909612
1.6	0.888083	0.894852	0.894239	0.894239	0.894239

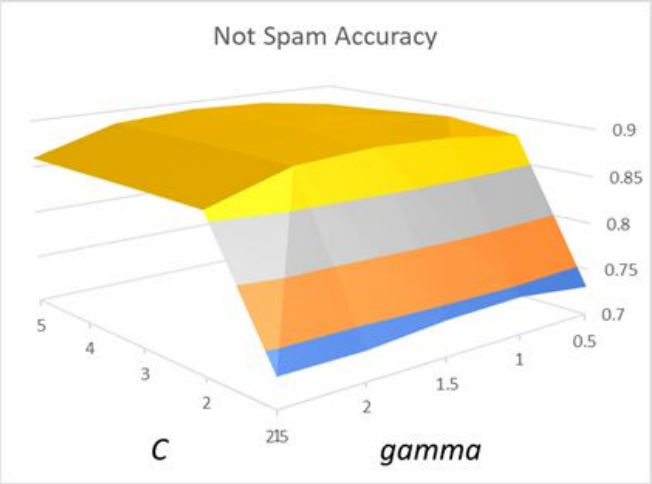
	SVM AUROC				
	0.1	0.4	1.6	6.4	25.6
0.1	0.5	0.549249	0.81991	0.82396	0.820908
0.2	0.5	0.735293	0.791051	0.790363	0.788091
0.4	0.5	0.72251	0.729602	0.73195	0.73195
0.8	0.5	0.618192	0.622813	0.625515	0.625515
1.6	0.5	0.550657	0.550303	0.550303	0.550303

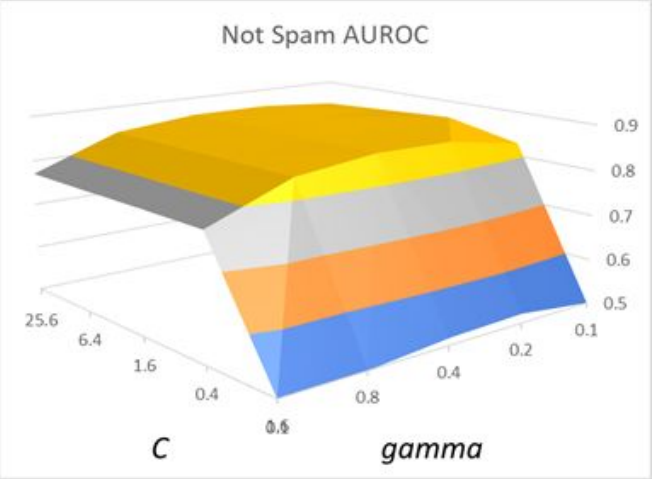




	SVM Acc				
	1	2	3	4	5
0.5	0.730589	0.879618	0.887566	0.88177	0.876803
1	0.740194	0.88889	0.887731	0.889056	0.888725
1.5	0.737875	0.891539	0.892202	0.891706	0.89154
2	0.730589	0.886076	0.886738	0.886572	0.886241
2.5	0.730589	0.859417	0.859914	0.859583	0.859583

	SVM AUROC				
	0.1	0.4	1.6	6.4	25.6
0.1	0.5	0.828613	0.861177	0.85214	0.843394
0.2	0.520805	0.858301	0.857234	0.858352	0.858033
0.4	0.515664	0.860319	0.858524	0.858147	0.858173
0.8	0.5	0.840105	0.840108	0.839994	0.839579
1.6	0.5	0.76971	0.770578	0.769916	0.769916





Appendix C - Source Code for ML algorithm

File: TestingVR

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
'''
Created on 3 okt. 2017

@author: daniel.larsson
'''

import DatasetImport
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn import metrics

### set True to show all predictions in the console
post_predictions = False
starting_fraction = 0.0
fraction_size = 0.2

### change descript to whichever label to be tested
descript = 'Cost'

### used for converting 0 and 1 into text
### in the "post predictions" loop
labels = 'Negative', 'Positive'

### import the uber dataset from excel file (specified in UberImport)
### shuffle all rows
### divide dataset into training and testing sets
### divide training and testing sets into posts and labels
### returns training posts (string[]),
###     training labels (int[]),
###     testing posts (string[]),
###     testing labels (int[])
def dataImport():

        ### get and shuffle all indexes
        dataset_indices = []
        for x in range(0, DatasetImport.countDataset()):
        dataset_indices.append(x)

        ### get dataset
        dataset_posts = DatasetImport.getDataset(dataset_indices)
        dataset_labels = DatasetImport.getLabels(dataset_indices, descript)

        ### define which fraction used for testing
        testing_start = int(starting_fraction * (len(dataset_posts)-1))
        testing_end = int((starting_fraction + fraction_size) * (len(dataset_posts)-1))

        ### assign testing set
        data_test = dataset_posts[testing_start:testing_end]
        labels_test = dataset_labels[testing_start:testing_end]

        ### assign training set
        data_train = list(dataset_posts)
        del data_train[testing_start:testing_end]
        labels_train = list(dataset_labels)
        del labels_train[testing_start:testing_end]

        return data_train, labels_train, data_test, labels_test, dataset_indices
```

```
### vectorizes import data
### trains specified classifier
### if(post_predictions):
###     prints all predictions
### returns accuracy score
def classifyDataset(classifier):

        ### get training and testing sets from dataImport()
        data_train, labels_train, data_test, labels_test, dataset_indices = dataImport()

        ### TfIdf vectorization, choosing ngrams in constructor
        vectorizer = TfidfVectorizer(strip_accents = 'unicode') #ngram_range=(1, 2)
        training_vector = vectorizer.fit_transform(data_train)
        testing_vector = vectorizer.transform(data_test)

        ### fits the specified classifier (input)
        clf = classifier
        clf.fit(training_vector, labels_train)

        ### predict labels of testing set
        pred_test = clf.predict(testing_vector)
        pred_train = clf.predict(training_vector)

        ### Prints predictions and true labels.
        ### Change "post_predictions" in top of document
        ### to print all predictions of the testing set
        if(post_predictions):

        for x in range(0, len(labels_test)):
        str = ":("
        if(pred_test[x] == labels_test[x]):
                str = ":)"
                print str, "Thought", labels[labels_test[x]], "was", labels[pred_test[x]]
                print data_test[x]
                print ""
        else:
                print str, "Thought", labels[labels_test[x]], "was", labels[pred_test[x]]
                print data_test[x]
                print ""

        return pred_test, pred_train, labels_test, labels_train, data_test

#score = cross_validate(estimator, X, y=None, groups=None, scoring=None, cv=None, n_jobs=1, verbose=0,
fit_params=None, pre_dispatch='2*n_jobs', return_train_score='warn')


### Method for making multiple tests
### and getting average score
###
### input = clf (classifier)
### prints average score
def crossValidationTest(clf):

        ### Cross validation accuracy calc
        total_test_acc = 0
        total_train_acc = 0
        total_test_roc = 0
        total_train_roc = 0

        for x in range(0, 5):
        global starting_fraction
        starting_fraction = x * 0.2
```

```
        pred_test, pred_train, labels_test, labels_train, data_test = classifyDataset(clf)

        test_acc = metrics.accuracy_score(labels_test, pred_test)
        train_acc = metrics.accuracy_score(labels_train, pred_train)
        test_roc = metrics.roc_auc_score(labels_test, pred_test)
        train_roc = metrics.roc_auc_score(labels_train, pred_train)

        total_test_acc = total_test_acc + test_acc
        total_train_acc = total_train_acc + train_acc
        total_test_roc = total_test_roc + test_roc
        total_train_roc = total_train_roc + train_roc

        print test_acc, train_acc, test_roc, train_roc

### Naive Bayes classification
from sklearn.naive_bayes import MultinomialNB
clf = MultinomialNB(alpha = 0.04)
crossValidationTest(clf)

### Logistic Regression classification
from sklearn import linear_model
clf = linear_model.LogisticRegression(C = 6)
crossValidationTest(clf)

### SVM classification
#linear
from sklearn import svm
clf = svm.SVC(C = 1.2, kernel = 'linear')
crossValidationTest(clf)

#rbf
clf = svm.SVC(C = 1.2, gamma = 0.8)
crossValidationTest(clf)
```

File: DatasetImport

```
'''
Created on 2 okt. 2017

@author: daniel.larsson
'''

import sys
import pandas
import re
from nltk.stem.snowball import SnowballStemmer
from nltk.corpus import stopwords

### Returns an array of text strings based on
### indexes from input array. Cleans posts from
### dataset before returning cleaned text strings
def getDataset(post_indices):

        ### parse Excel file from project directory
        xlsx = pandas.ExcelFile('VR_dataset.xlsx')
        df = xlsx.parse(xlsx.sheet_names[1])
        dataset = df.to_dict()

        ### declare output array
        social_media_posts = []

        ### clean each post with index x
        for x in post_indices:

        ### get text content of post x (Description)
        unique_post = dataset['Description'][x]

        ### check that the post is valid
        if(type(unique_post) is unicode and (re.search('[a-zA-Z]', unique_post))):

        ### remove any links or numericals
        for word in unique_post.split():
                if word.startswith('http'):
                unique_post = unique_post.replace(word, "")

        ### replace special characters
        unique_post = unique_post.replace("\n", " ")
        unique_post = unique_post.replace("/", " ")
        unique_post = unique_post.replace("%", " % ")

        unique_post = unique_post.lower()
        unique_post = ' '.join(unique_post.split())

        specialchars = "-_.,?!@#:|+=$()[]\"*"
        for char in specialchars:
                unique_post = unique_post.replace(char, "")

        stemmed_post = ""
        stemmer = SnowballStemmer("swedish")
        stop = set(stopwords.words("swedish"))
        removal_words = ""
        #removal_words = ["oculus", "facebook", "oculusrift", "http", "vr", "star", "zenimax", "rift", "googl",
"hahah", "htc", "microsoft"]

        result = []
        for word in unique_post.split():
```

```
                if(word not in stop):
                s = stemmer.stem(word)

                if(s != "" and s not in removal_words):
                        result.append(s)

        stemmed_post = " ".join(result)
        social_media_posts.append(stemmed_post)
        return social_media_posts

def getLabels(post_indices, descript):
        xlsx = pandas.ExcelFile('VR_dataset.xlsx')
        df = xlsx.parse(xlsx.sheet_names[1])
        dataset = df.to_dict()

        labels = []

        for x in post_indices:

        unique_post = dataset['Description'][x]
        if(type(unique_post) is unicode and (re.search('[a-zA-Z]', unique_post))):
            labels.append(int(dataset[descript][x]))

        return labels

def countDataset():
        xlsx = pandas.ExcelFile('VR_dataset.xlsx')
        df = xlsx.parse(xlsx.sheet_names[1])
        dataset = df.to_dict()

        return len(dataset['Description'])
```