Online Video Data Analytics

Jefferson Lai
Benjamin Le
Pierce Vollucci
Wenxuan Cai
Yaohui Ye
George Necula, Ed.
Don Wroblewski, Ed.

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2015-72
http://www.eecs.berkeley.edu/Pubs/TechRpts/2015/EECS-2015-72.html

May 13, 2015

Acknowledgement

We would like to thank George Necula for advising us throughout the entirety of this Capstone project. We would also like to thank Jibin Zhan from Conviva for introducing the problem space to us and Pat McDonough from Databricks for providing extensive technical support throughout our use of Databricks.

University of California, Berkeley College of Engineering

MASTER OF ENGINEERING - SPRING 2015

Electrical Engineering and Computer Sciences

Data Science and Systems

Online Video Data Analytics

Jefferson Lai

This **Masters Project Paper** fulfills the Master of Engineering degree requirement.

Approved by:

1. Capstone Project Advisor:

Signature: _____ Date _____

Print Name/Department: George Necula/Electrical Engineering and Computer Sciences

2. Faculty Committee Member #2:

Signature: _____ Date _____

Print Name/Department: Don Wroblewski/Fung Institute for Engineering Leadership

Abstract

This capstone project report covers the research and development of Smart Anomaly Detection and Subscriber Analysis in the domain of Online Video Data Analytics. In the co-written portions of this document, we discuss the projected commercialization success of our products by analyzing worldwide trends in online video, presenting a competitive business strategy, and describing several approaches towards the management of our intellectual property. In the individually written portion of this document, we discuss our implementation of two machine learning models, k-Nearest Neighbors and Random Forest, and evaluate them as a means of identifying subscription churners, the primary goal of Subscriber Analysis. In particular, we show that the performances achieved by these models using our initial, restricted feature set are promising and warrant future exploration of these models.

Contents

* Co-written with Benjamin Le, Pierce Vollucci, Wenxuan Cai, and Yaohui Ye

I. Introduction

This report documents the Online Video Data Analytics capstone project completed in the course of the Data Science and Systems focus of the Master of Engineering degree at UC Berkeley. Through the collective efforts of Benjamin Le, Jefferson Lai, Pierce Vollucci, Wenxuan Cai, and Yaohui Ye, our team has not only characterized the need for effective data analysis tools in the domain of online video data, but has also developed analysis tools which attempt to address this need. As we will describe in detail in our Individual Technical Contributions, our work has produced many important findings and we have made significant strides towards a complete implementation of these tools. However, at the time of the writing of this report, additional work is required before our tools can be considered complete. That being said, our substantial progress has allowed us to form a very clear vision of what our finished tools will look like and how they will perform. Our vision leads us to believe that, once finished, our tools can be of great potential value to entities within the online data analytics industry. In order to understand how best to cultivate this value, we have extended our vision to depict tools to marketable products and we have evaluated the potential for our team to establish a business offering these products. In doing so, we have performed extensive research of the current market and industry which our potential business would be entering. The remainder of this report presents our findings and is divided into seven sections. First, we introduce our industry partner, Conviva, in the Our Partner section. Second, we present the objective of our work and the motivation behind the resulting products in the Products and Value section. Third, we introduce and describe the dataset leveraged by our products in the Our Dataset section. Fourth, our team characterizes our industry as well as our competitive strategy in the Trends, Market, and Industry section. Fifth, in our Intellectual Property section, we describe how we plan to protect the value of our work. Sixth, the Individual Technical Contributions section of this report details our specific contributions toward the goals of our project. Finally, the Concluding Reflections section contains a retrospective analysis

of the significance of this work and provides an outlook on the potential for continuation of our work in future endeavors.

II. Our Partner

This project is sponsored by Conviva, a leading online video quality analytics provider. Conviva works with video content providers, device manufacturers, and developers of video player libraries to gather video quality metrics from content consumers. Through our partnership with Conviva, we have access to an anonymized portion of their online video quality metric dataset for the development of our products. We also have access to Conviva engineers for collaboration purposes who provide domain knowledge and on site support. For the purpose of the business analysis forthcoming, the entity, “we”, will refer to our capstone team as a separate entity from Conviva. Furthermore, we consider Conviva to be a close partner to our capstone team on whom we can rely for continuous access to their dataset.

III. Products and Value

A vast and painfully prevalent gap exists between the amount of data being generated around the world and the global tech industry’s ability to utilize it. According to IBM, “every day, we create 2.5 quintillion bytes of data — so much that 90% of the data in the world today has been created in the last two years alone” (“Bringing Big Data”). While the already monumental quantity of data continues to grow, scientists and engineers alike are just beginning to tap into the power of this data. This is not to say that data does not already pervade nearly every imaginable aspect of life today; it does. Large amounts of data crunching and predictive analysis go on behind the scenes of numerous activities, from returning search queries, to recommending movies or restaurants, to predicting when and where the next earthquake will occur. However, there remains a massive body of questions and problems in both academia and industry that researchers have been unable to use data to answer. One domain in which better utilization of data could yield tremendous benefit is that of online media. Our team aims

to serve this niche by building tools that address two critical challenges of online video data analysis: accurate real-time anomaly detection on large scale data and subscriber churn analysis.

Online video providers struggle to consistently serve a TV-like experience with high quality video free of buffering interruptions. Many factors within the "delivery ecosystem" affect the throughput of a video stream and, ultimately, the end user's viewing experience (Ganjam et.al 8). These factors include "multiple encoder formats and profiles, CDNs, ISPs, devices, and a plethora of streaming protocols and video players" (Ganjam et al. 8). An automatic anomaly detection and alert system is necessary in order to both inform a video content provider when their customers are experiencing low quality and, among the many possible factors, diagnose the primary cause of the problem. For example, if all customers experiencing frequent buffering belong to a certain ISP, then the alert system should flag that ISP as the root of the problem. The challenge that plagues many current solutions, however, is related to the aforementioned growth in the amount of collected data. While it is easy to detect when and why predictable measurements misbehave at small scale, it is hard to do so with high accuracy at large scale, across a range of system environments. To meet this challenge, our team has developed our Smart Anomaly Detection system to detect when and why truly anomalous and interesting behavior occurs in measured data. Such a system would greatly help content providers improve both their operational performance and efficiency. This value will be passed down to the viewers who benefit from higher service quality.

A second problem for subscription-based online video content providers is the ability to retain their subscribers. While the problem of diagnosing and eventually reducing subscriber churn has existed as long as the subscription service model, only recently has the tech industry developed the capacity and means to use big data to do so (Keaveny). Furthermore, largely due to the fact that online video hosting and distribution is a relatively new service, nearly all previous works in the area have

focused on other domains such as telecommunications or television service subscriptions (Keaveny; Verbeke 2357-2358). Our team's Subscriber Analysis toolset aims to fulfill this unmet need by developing predictive models of viewer engagement and churn based on viewing activity and service quality data. Being able to predict churners and identify characteristic predictors from the data allows companies to focus on addressing the problems most critical to their viewers, thereby reducing churn rates. As proven by Zeithaml, there is a real, high cost associated with subscriber churn (Zeithaml). Thus by aiding in the reduction of churn, our Subscriber Analysis product can both help content providers increase revenues and result in higher overall satisfaction for those who purchase online video subscriptions.

For the reasons described above, our team is confident that our Smart Anomaly Detection and Subscriber Analysis products are important and valuable to both content providers and their customers, the viewers.

IV. Our Dataset

Conviva provided 4.5 months of session summary data from a single anonymous content provider for our research and development. 73,368,052 rows of session summaries are in this dataset. Each session summary represents a single instance of a viewer requesting a video object. In addition to service quality data, the type of device used by the viewer, the approximate location of the viewer, and metadata about the video content being accessed are collected into 45 columns. Subscription and demographic information about a viewer beyond their location are not available within this dataset. Fields that might otherwise help identify the anonymous content provider such as video content metadata were also anonymized by Conviva prior to data transfer to protect their customer.

Although the data was preformatted by Conviva before being transferred to our capstone team, we identified two important challenges implicitly encoded within this data through exploratory data analysis and follow-up communication with Conviva

engineers. First, several of the fields in the session summaries are not as reliable as we initially believed. For example, fields such as `season` and `episodeName` are often empty. Second, our initial dataset included data generated by an artificial “viewer” that was used by Conviva for testing purposes and exhibited very strange, abnormal behavior. This was very important to keep in mind as we developed and evaluated our tools based on this data.

For our Smart Anomaly Detection product, Conviva informed us of the two most important metrics in assessing QoS that they wished to detect anomalies for. First is the number of attempts in watching online video over a time period. Low number of attempts indicates that users may be unable to access the content due to a datacenter failure. A high number of attempts signals the presence of a viral video. Second is the video start failure (VSF) rate. The VSF rate is the percentage of attempts that have failed to begin properly. VSFs may be caused by bugs in the video player software or by improper encoding/decoding of video content. Unlike attempts, low VSF rate is not a concern for video content providers. However, high VSF rate indicates major issues in the content delivery pipeline. To determine if an attempt has ended in VSF, we look at the `joinTimeMs` and `nrerrorsbeforejoin` columns in the data. The table below provides description about these 2 columns. An attempt ends in VSF if `joinTimeMs < 0 AND nrerrorsbeforejoin > 0`.

Column Name	**DataType**	**Description**
joinTimeMs	int	How long this attempt spent joined with the video stream. If this attempt has not yet joined, then this value will default to -1.
nrerrorsbeforejoin	int	How many fatal errors occurred before video join

For Subscriber Analysis, on the other hand, the nature of the problem is that we cannot know beforehand which fields within the session summary are useful in distinguishing viewers who are likely to churn. At the same time, as a consequence of

the first of the challenges mentioned above, the Subscriber Analysis product should not indiscriminately use all fields of the session summary, including both reliable and unreliable fields. Thus, a central component of the work in Subscriber Analysis revolves around selecting a subset of these fields to use to form “features” to be used by the product.

V. Trends and Strategy

Having defined our team’s product and established both how they generate value and for whom they are valuable, we can focus on how we plan to bring these products to market from the standpoint of a new business. Amidst an era of rapid information and especially within the technology-abundant Silicon Valley, bringing such innovations to market requires understanding the market and having a well-formed competitive strategy. In this section, we describe the social and technological trends relevant to our product as well as the market and industry our business would be entering. We then describe the strategy we have developed that would allow our business to be successful in this competitive environment.

Why Us, Why Now

In the past five years, the number of broadband internet connections in the United States has grown from 124 million in 2009 to 306 million in 2014, leading to a compound annual growth rate of 19.8% per year (“Num. of Broadband Conns.”). This growth is indicative of the ever-growing role the Internet plays in daily life. Along with the growth of the Internet, as both a cause and effect, comes the spread of online services. In her article for Forbes, Erika Trautman, CEO of Rapt Media, states that “each year, more and more people are ditching cable and are opting for online services like Netflix and Hulu.”

The emergence of online video services has been so disruptive a shift in video distribution, that it incited a 2012 public hearing concerning public policies from the

Senate Committee on Commerce, Science, and Transportation. In the hearing, leaders from technology juggernauts and state senators alike echoed the same viewpoint: online video services are the future of video distribution. Susan D. White, the Vice Chair for Nielsen, a leading global information and measurement company, reported that "the use of video on PCs continues to increase—up 80 percent in the last 4 years…Consumers are saying, unequivocally, that online video will continue to play an increasing role in their media choices" (U.S. Sen. Comm. on Commerce, Sci. & Trans. 9).

Of course, similar to other industries, a business seeking to enter today's online video industry must meet a myriad of both business and engineering challenges. Unlike many of these industries, however, our industry is well-positioned to easily collect and analyze vast amounts of data to meet these challenges. Out of these conditions, the online video analytics (OVA) industry emerged, helping to translate and transform this data into useful insights that can be directly used by online video providers. A report by Frost & Sullivan summarized the rapid growth in the market:

> Still a largely nascent market, online video analytics (OVA) earned $174.7 million in revenue in 2013. It is projected to reach $472 million in 2020 as it observes a compound annual growth rate (CAGR) of 15.3%....The growth of OVA is largely attributed to the high demand for advanced analytics from online video consumption (Jasani).

Spurred by the massive opportunity in this market, our team has worked with Conviva to identify two of the most significant technical challenges faced by content providers: real-time detection of anomalies in a rapidly changing, unpredictable environment and efficiently reducing subscriber churn.

The challenge of retaining subscribers has existed as long as the subscription-based business model itself. As the competitive landscape of the online video market continues to evolve, the ability to diagnose and mitigate subscriber churn is a crucial component for business success. Sanford C. Bernstein estimated Netflix's

average annual churn rate at 40-50%, which translates to 24-30 million subscribers (Gottfried). Reducing this churn rate by even a small fraction and keeping the business of these subscribers could mean significant increases in revenue. Just as critical for success is the ability to detect and respond to anomalies or important changes in metrics such as network usage and resource utilization. On July 24, 2007, 18 hours of Netflix downtime corresponded with a 7% plummet in the company's stock (Associated Press).

As previously described, our team provides solutions to these challenges through our Subscriber Analysis and Smart Anomaly Detection products. We believe that while these solutions, which use a combination of statistical and machine learning techniques, are powerful, our primary value and competitive advantage lies in our use of the unique dataset available to us through our partnership with Conviva. In the following sections, we discuss in detail how we plan to establish ourselves within the industry. In particular, we describe how we will position ourselves towards our buyers and suppliers as well as how we will respond to potential new entrants and existing competitors to the market.

Buyers and Suppliers

One of the most important components of a successful business strategy is a deep and accurate understanding the different players involved in the industry. In particular, an effective strategy must define the industry's buyers, to whom businesses sell their product, and its suppliers, from whom businesses purchase resources. In this section, we provide an overview of important entities related to our industry and present an analysis of our buyers and suppliers.

Potential customers for the global online video analytics market include content providers, who own video content, and service providers, who facilitate the sharing of user-generated video content (Jasani; Smith). Among the content providers are companies such as HBO, CCTV, and Disney, who all bring a variety of original video content to market every year. These businesses serve a huge user base and are able to

accumulate large amounts of subscriber data. HBO alone was reported to have over 30 million users at the beginning of 2014 (Lawler). This abundance of data presents massive potential for improving these companies' product quality and, correspondingly, market share. Our Subscriber Analysis product can realize some of this potential by helping understand the experience and behavior of their users. Furthermore, with our Smart Anomaly Detection product, content providers can be made aware when significant changes occur in viewer behavior, system performance, or both. These tools can lead to a more valuable product, as seen from the content provider's viewers. While service providers such as Twitch or Vimeo differ from content providers in that they tend to offer free services, the success of these companies is still highly dependent on retaining a large number of active users. Thus, we target service providers in much the same way as we target content providers. Overall, we find that content and service providers, as buyers, are at an advantage in terms of business leverage over us, as sellers. This is primarily due to low switching costs, which arise from the fact that other businesses such as Akamai and Ooyala offer products for processing video data similar to ours (Roettgers). Because buyers ultimately make the choice choosing where to send their data on which both Smart Anomaly Detection and Subscriber Analysis depend, it can be difficult to deter customers from switching to our competitors. However, as we describe later in this paper, our unique approach towards churn analysis may differentiate us from our competitors and decrease buyer leverage over us.

On the other end of the supply chain, we also must consider who our suppliers will be and what type of business relationship we will have with them. Because our product exists exclusively as software, we require computing power and data storage capacity. Both of these can be obtained through the purchase of cloud services. Fortunately, the current trends indicate that cloud services are becoming commoditized, with many vendors such as Amazon, IBM, Google and Microsoft offering very similar products (F. Hanley). Though our buyers benefitted from low switching costs between us and our competitors, we face even lower switching costs between our suppliers. This is because while there is a considerable amount of effort involved with integrating a

monitoring or analytical system with a new set of data, migrating the services between the machines which host them is almost trivial, involving only a transfer the data and minor machine configuration. In addition to cloud services, to a certain extent, we are dependent on device manufacturers and developers of video player libraries. We require them to provide an Application Program Interface (API) which we can use to gather online video analytics data from users. Fortunately, prior relationships with these device manufacturers and developers have been established through our partner Conviva. Conviva can help us open APIs for new devices and video players to maintain the flow of data required for our products.

As Porter argued, strategic positioning requires performing activities either differently or more efficiently than rivals ("Five Competitive Forces" 11). Our partnership with Conviva affords us a large quantity of high quality data for our algorithms to utilize, giving us a slight advantage compared to other services. In order to maintain and build upon this advantage, however, we must focus on developing our products to utilize this data and yield results in a superior manner. Thus, it is clear that our ability to differentiate from competing products and outperform them is key to our business strategy and the following sections describe how we can do so.

New Entrants

"Know yourself and know your enemy, and you will never be defeated" (Sun Tzu 18). This proverb can be applied to almost any competitive situation, from warfare to marketing. Interpreting this teaching in the context of business strategy, we identify that understanding the rivalry among existing and potential competitors is essential to a lasting competitive advantage. This interpretation fits well within the framework of Michael Porter's five competitive forces. We now examine new entrants through the incumbent advantages and barriers to entry that work to keep this force as a low threat to both of our products. Porter recognized seven incumbent advantages ("Five Competitive Forces" 4-6). The first is supply side economies of scale in which established incumbents have tremendous strength. The code behind a given analysis

program is a fixed cost which scales well with an increased number of users, thus reducing the marginal cost of the code with each customer. The servers that receive and process the various users' data are linear, but scale with the number of customers acquired. The real advantage comes from the exponential power of the data supplied by these same customers, a theme we have come back to repeatedly in this paper. As the breadth and quantity of data increase with the combined user base of our customers, our algorithms become increasingly powerful and allow the incumbent product to outperform new entrants. This leads into our second advantage, demand side benefits of scale. As the authority in the field of providing content providers with analytics, incumbents can encourage customer demand by using their data on content quality improvements to provide hard evidence of the bottom line improvement new users can expect. "Increasingly powerful predictive analytics tools will unlock business insights [and drive revenue]" (Kahn 5). Demonstrating that our tools provide access to increases in revenue is key to nurturing demand.

Switching from an incumbent's service provides another barrier to entry, customer switching costs. While switching from one online service to another is not prohibitively expensive considering the benefits offered, the most impacting loss is in the past data the incumbent analysis provider's algorithms had of user's performance. "As we increase the training set size L we train on more and more patterns so the test error declines" (Cortes et al. 241). Via additional training examples, the incumbent's algorithm would consistently outperform the new entrant as the new entrant slowly acquires a pool of data comparable to that of the incumbent.

Just as it does not appear expensive for a customer to switch, it appears feasible for new entrant to join due to minimal physical capital requirements. With Platform as a Service (PaaS) providers, a new entrant merely needs a codified algorithm and a client or two to get started. Still, it is again the data that proves key to providing value to our customers. Importantly, new entrants cannot attain this data until they acquire clients, a classic catch-22 which serves as an inhibiting capital requirement for new entrants.

The global reach of our data partner, Conviva, provides both a size independent advantage as well as an unequal access to potential distribution channels in that it allows for direct international sales in the form of immediate integration of our tools with the systems of our partner's customers. The last relevant advantage as discussed by Porter, concerns restrictive government policy. Privacy concerns do arise when personal data is used, however there are standards for anonymization to be employed when using such data (Iyengar). While governments do allow the use of such data, it has to be acquired by legal means, which means a new entrant is restricted in its means of gathering new data for its algorithms. Thus, after a thorough analysis of the potential new entrants of our industry, the incumbents' advantages suggest that the threat of new entrants is a relatively weak force in our industry.

Existing Rivals

Another category of threats that a successful business strategy must address is that of existing rivals. As Porter described, the degree to which rivalry drives down an industry's profit potential depends firstly on the intensity with which companies compete and secondly on the basis on which they compete ("Five Competitive Forces" 10). We analyze these two parts for each of our products separately.

As machine learning grows in popularity, research into anomaly detection and other analyses of time series data is receiving greater attention both in academia and in industry. A survey of anomaly detection techniques shows a variety of techniques applied in a diverse range of domains (Chandola). Our strategy must take into account the threat of commercialization of technologies into industry competitors. For example, in 1994 Dipankar Dasgupta used a negative selection mechanism of the immune system to develop a "novelty" detection algorithm (Dasgupta). In addition to these potential competitors, there already exist several important industrial competitors working on anomaly detection. In January, 2015, Twitter open-sourced *AnomalyDetection*, a software package that automatically detects anomalies in big data

in a practical and robust way (Kejariwal). Our Smart Anomaly Detection product is comparable to products from industry competitors such as Twitter; it is able to integrate with various sources of data, perform real-time processing, and incorporate smart thresholding with alerts. Although our competitors may try to research and develop a superior anomaly detection algorithm, we believe that our superior quantity and quality of data provided by Conviva gives us an edge over our competitors. Thus, we characterize competitive risk for Smart Anomaly Detection as weak. To a large extent, the competitors of Subscriber Analysis include the content providers themselves. Netflix spends $150 million on improving content recommendation each year, with the justification that improving recommendations and subscriber retention by even a small amount can lead to significant increases in revenue (Roettgers). These content providers have the advantage that they have complete access and control over the data they collect. If most companies were able to build an effective churn predictor in-house, the industry would be in trouble. However, we are confident that the quality of our Subscriber Analysis product will overwhelmingly convince content providers facing the classic “buy versus build” question, that building a product of similar quality would demand significantly more resources than simply purchasing from us (Cohn). This confidence is further supported by Porter in the context of the tradeoffs of strategic positioning (“What Is Strategy?” 4-11). In addition to content providers, there also exist competitors such as Akamai and Ooyala, who offer standalone analysis products to content and service providers. These competitors tend to focus on the monitoring and visualization of the data. In contrast, Subscriber Analysis focuses on performing the actual analysis to identify the characteristics and causes of subscriber churn.

Still, our most important advantage over these competitors remains our ability to perform in-depth churn analyses based on the abstraction of session summaries, which consist of a unique combination of metrics exclusively related to service quality. To the best of our knowledge, this is unique to previous and existing works in subscriber churn analyses. Our research has shown that the most prominent existing analysis approaches all incorporate a significant amount of information, often involving direct

customer surveys or other self-reported data. Because service quality data is abundant and easy to obtain compared with demographic data, our Subscriber Analysis product can appear extremely appealing to potential customers. This easy to collect and consistent subset of video consumption data means our product has the potential to scale much better than existing approaches which require highly detailed, case-specific, and hard to obtain datasets. However, we cannot guarantee that this algorithmic advantage be sustained as our competitors continue their own research and development. Thus, we conclude that threat of competition to Subscriber Analysis is moderate.

Substitutes

The final element of our marketing strategy concerns the threat of new substitutes. Porter defined substitutes as products that serve the same purpose as the product in question but through different means (“Five Competitive Forces” 11). We first discuss potential substitutes for our Smart Anomaly Detection product.

The gold standard for most alert systems is human monitoring. Analogous to firms hiring security monitors to watch over buildings, video content providers can hire administrators to keep watch over network health. A more automated substitute is achieved through simple thresholding, in which hardcoded thresholds for metrics such as the rate of video failures trigger an alarm when exceeded. Content providers can also utilize third party network performance management software from leaders like CA, Inc. This type of software alerts IT departments of potential performance degradation within the companies' internal networks (CA Inc. 4). Similarly, content providers can pursue avenues besides Subscriber Analysis to reduce churn rates. Examples include utilizing feedback surveys and consulting expert market analysts. Feedback from unsubscribers is an extremely popular source of insight into why customers choose to leave and can go a long way in improving the product and reducing churn rate. These often take the form of questionnaires conducted on the company’s website or through email. In addition, content providers commonly devote many resources towards

consulting individuals or even entire departments with the goal of identifying marketing approaches or market segments that generate lower churn rates.

Porter classified a substitute as a high threat when the substitute offers superior price/performance (“Five Competitive Forces” 12). With this in mind, we found that the overall threat of substitutes for Smart Anomaly Detection product is low. In contrast to human monitoring, our product offers a superior value proposition to our buyer. According to Ganjam et. al, many factors, including "multiple encoder formats and profiles, CDNs, ISPs, devices, and a plethora of streaming protocols and video players," affect the end user’s viewing experience (Ganjam 8). The complexity of this delivery ecosystem requires equally complex monitoring with filters to isolate a specific ISP, for example, and to determine if its behavior is anomalous. Such large scale monitoring does not scale efficiently when using just human monitoring. Similarly, simple thresholding poses little threat as a substitute because fine tuning proper thresholds over multiple data streams is difficult and time consuming. Many false positives and negatives still occur, despite such fine tuning (Numenta 11). Network performance management software, on the other hand, poses a considerable threat to us. However, while they are excellent at detecting problems within a content provider’s internal network, they alone cannot increase the quality of service. Xi Liu et al. argue that an optimal viewing experience requires a coordinated video control plane with a “global view of client and network conditions” (Liu 1). Fortunately, thanks to our partnership with Conviva, we have the data necessary to obtain this global view.

Just as with Smart Anomaly Detection, the threat of substitutes for Subscriber Analysis is also low. Although feedback surveys are direct and easy to implement, there are several inherent issues associated with them. Perhaps most prominently, any analysis that uses this data format must make a large number of assumptions in order to deal with uncontrollable factors such as non-response bias and self-report bias (Keaveny). Expert opinion, whether gathered from a department with the company or through external consult, is the traditional and most common approach towards

combating subscriber churn. This method, while very effective, tends to be extremely expensive. Still, as demonstrated by Mcgovern's Virgin Mobile case study, expert opinion can lead to identifying the right market segment, lower churn rates, and ultimately a successful business (McGovern 9).

To mitigate the threat of substitutes, Porter suggests offering "better value through new features or wider product accessibility" ("Five Competitive Forces" 16). For Smart Anomaly Detection, there are several avenues to pursue to provide a better value proposition to our buyers. For example, we can develop more accurate predictors with additional data from Conviva and explore new machine learning algorithms. For Subscriber Analysis, the threat of substitutes continues to be low because, unlike the examples given above, our product can perform effective analyses and generate valuable insights in an automated, efficient fashion. Data obtained through direct customer surveys, while potentially cheap, come bundled numerous disclaimers and can lead to a certain stigma from the subscriber's perspective. Furthermore, although data obtained through surveys, such as demographic information, might be more helpful in characterizing churners, by focusing on providing churn analysis based only on service quality data, our Subscriber Analysis product has at least one significant advantage. Service quality data from content consumers can be more easily gathered compared to data such as demographic information. Consequently, our product can be more appealing and accessible to content providers, especially those who do not have access to, or would like to avoid the cost of obtaining, personal data about their users. We also point out that both Subscriber Analysis and the substitutes such as those described above can be used in combination with each other. In such a case, our Subscriber Analysis product becomes even more appealing. This is because it can use the data from customer feedback to yield further improved performance. Our product would also make tasks such as identifying appropriate market segments much easier and cheaper to accomplish for content providers.

Strategy Summary

In summary, there are several social and technological trends which make now the right time for commercializing our Subscriber Analysis and Smart Anomaly Detection products. The most prominent among these are the rapid growth in internet connectivity and the spread of online services. In order to evaluate how well positioned we are to capitalize on the opportunity created by these trends, we developed a business strategy through competitive industry and market analysis from several different perspectives. From the perspective of buyers and suppliers, though we find that buyer power is significant, over time we expect to differentiate ourselves from our competitors by leveraging both the superior size of our dataset and our more efficient overall use of the data. We find that supplier power is low for our industry because the only significant resource we require is available through cloud services, an industry in which we have high buyer power and which is quickly becoming commoditized. From the perspective of rivals, the threat of new entrants is low due in large part to the superior quantity and quality of our data as well as the benefits of scale we would stand to benefit from as incumbents. Similarly, while existing competitors do present a threat, we find that our use of superior data and unique approach gives us a significant competitive advantage over them. Finally, we see a weak threat from the perspective of substitutes because we offer superior value at a cheaper price to our customers that only improves in combination with other techniques. Taken together, our evaluations lead us to believe that there is significant potential for a sustained competitive advantage over competitors, and that now is an opportune time to pursue it.

VI. Intellectual Property

Equally important to a team's ability to build a valuable product and bring it to market is its ability to protect that value. In this section, we explain how we, as a business pursuing the strategy above to bring Subscriber Analysis and Smart Anomaly Detection to market, intend to sustain and protect the value of our work.

The traditional method for protecting the value of a new technology or innovation is obtaining a legal statement regarding ownership of intellectual property, IP, in the form of a patent. Indeed, patents have performed well enough to remain a primary mechanism for IP protection in the US for more than 200 years (Fisher). Unfortunately, when it comes to software, the rules and regulations regarding patents become dangerously ambiguous. The recent influx of lawsuits involving software patents has been attributed to the issuance of patents that are unclear, overly broad, or both (Bessen). Despite software patent laws being an active and controversial topic, these discussions have simply left more questions unanswered. The *Alice Corporation v. CLS Bank* Supreme Court case in 2013 is oft cited as the first source of information about software patentability, and even this case has been criticized for the court's vagueness (*Alice Corporation v. CLS Bank*). As noted by patent attorney and founder of IPWatchDog.com Gene Quinn, a definitive line should be drawn by the courts: a patent describing only an abstract idea, without specific implementation details, is invalid and cannot be acted upon (Quinn).

Thus, faced with the question of patentability, our team must examine the novelty of our Subscriber Analysis and Smart Anomaly Detection products. The goals of Subscriber Analysis and Smart Anomaly Detection are to diagnose the causes of subscriber churn and intelligently detect important changes in measured data respectively. Because these goals are rather broad, there exist a number of existing implementations, both old and new, with similar objectives. As a team considering patentability, we look towards the novelty of our specific approach and implementation. In the course of this introspection, we note that our implementation amalgamates open source machine learning libraries such as SciKit-Learn, published research from both industry and academia, programming tools such as those offered by Databricks, and finally the unique data afforded to us through our partnership with Conviva. With this in mind, we conclude that current patenting processes are flexible enough such that by defining our implementations at an extremely fine granularity, we would likely be able to

obtain a patent on our software. However, we strongly believe that there exist several significant and compelling reasons against attempting to obtain a patent for our work. In this section, we elaborate on these reasons and describe an alternative method for protecting our IP which better suits our situation and business goals.

There is an abundance of existing anomaly detection patents of which we must be wary. Several of these patents are held by some of the largest companies in the technology sector, including Amazon and IBM. For example, *Detecting anomalies in Time Series Data*, owned by Amazon, states that it covers “The detected one anomaly, the assigned magnitude, and the correlated at least one external event are reported to a client device” (U.S. Patent 8,949,677). One patent owned by IBM, *Detecting anomalies in real-time in multiple time series data with automated thresholding*, states that in the submitted algorithm, a “comparison score” is calculated by comparing “the first series of [observed] normalized values” with “the second series of [predicted] normalized values” (U.S. Patent 8,924,333). In observance of these patents, we must be wary of litigation, especially when it concerns large technology companies. Recently, many companies in the tech industry, both small and large, have come under fire with a disproportionate number of patent infringement lawsuits (Byrd and Howard 8). Some optimists argue that most companies need not worry, because large technology companies are likely filing patents defensively. However, these companies are often the ones who play prosecutor in these patent infringement cases as well. For example, IBM, a holder of one of these anomaly detection patents, has a history of suing startups prior to their initial public offerings (Etherington). More recently, Twitter settled a patent infringement lawsuit with IBM by purchasing 900 of IBM’s patents (Etherington). In a calculated move by IBM, Twitter felt pressured to settle to protect their stock price in preparation for their IPO. Thus, we must be extremely careful in how we choose to protect our intellectual property. If this means filing a patent, then we must be prepared to use it defensively. This is likely to require a very large amount of financial resources. As we do not currently have these resources to spare and cannot guarantee that the protection offered would be long lasting or enforceable, we seek an alternative to patenting.

The goal of our Subscriber Analysis product is to predict the future subscription status of users based on past viewing behavior. Despite our research on existing patents, our team has been unable to find many patents which pose a legal threat to Subscriber Analysis. Most active patents on video analytics focus on video performance and forecast, such as Blue Kai Inc's *Real time audience forecasting* (US Patent App. 20120047005). In contrast, the patent field of quantization and prediction of subscriber behavior remains largely unexplored. Despite several commercial solutions on the market, there has not been a corresponding number of patents. Thus, Subscriber Analysis does not face the same level of risk of litigation compared to Smart Anomaly Detection. However, there are a handful of patents in other domains that we need to be wary of. *System and method for measuring television audience engagement*, owned by Rentrak corporation, describes a system that measures audience engagement based on the time he or she spends on the program (US Patent 8,904,419). In short, it constructs a viewership regression curve for different video content and measures the average viewing length. For a new video, the algorithm infers the level of viewer engagement based on the video content and the duration the viewer watched. While viewer engagement is a critical component for predicting behavior in Subscriber Analysis, we also incorporate additional data. These include viewing frequency, content type, and video quality. Under such circumstances, we do not see it as necessary to license patents such as the one above for two reasons. First, and perhaps most importantly, we apply churn analysis in the domain of online video, whereas most relevant patents apply to other older domains. Second, our algorithm incorporates a unique set of features corresponding to the data provided by Conviva.

The decision to pursue and rely on a patent in the software is an expensive one in both time and financial resources as well as a risky one due to the tumultuous software patent environment. As such, while we may pursue a patent, it will not be relied upon for our business model. As such, we have two additional IP strategies to investigate, open sourcing and copyrighting.

Open source software is software that can be freely used, changed, and shared (in modified and unmodified form) by anyone, subject to some moderation (Open Source Initiative). Open sourcing has become increasingly popular; both the total amount of open source code and the number of open source projects are growing at an exponential rate (Deshpande, Amit et al). For the purposes of our endeavor, it is not the novelty of our approach but our dataset and partner provided distribution network that distinguishes us. As the algorithms used are already publicly available, open sourcing our code does not cost us anything but provides us the shield of using open source software for our business and the badge having our code publically exposed and subject to peer review. Our business model would entail providing a value-added service company, dedicated to helping customers integrate their existing systems with our anomaly detection library. Through our partnership with Conviva, we have an established distribution network to our potential customers who we can offer immediate integration with Conviva's existing platform. This is a significant advantage as while open source is openly available to all users, they are primarily for experienced users. Users have to perform a significant amount of configuration before they begin using the code, which can pose quite a deterrent. While we will use the open source codebase as a foundation for our service, we will additionally provide full technical support in designing a customized solution that meets the customer's needs. By pivoting towards this direction, we add additional monetary value to the product that we can sell and bridge the technical gap for unexperienced users, relying on a SAAS implementation style for our business model instead of on a patent.

Copyright for software provides another IP Strategy option. While debate continues to surround software patents, copyrights are heavily applied in software. As expressed by Forbes's Tim Worstall, "there's no doubt that code is copyright anyway. It's a specific expression of an idea and so is copyright." There are several differences in the protection offered by copyrights compared to that of patents. While a patent may expose a very specific invention or process to the public and protect for 20 years, a

copyright offers much broader protection while still providing the threat of lawsuit for enforcement. The copyright lasts 90 years past the death of the author and offers statutory damages (Copyright.gov). In addition, the scope of what it encompasses proves more relevant to our endeavor. “Multiple aspects of software can qualify for copyright protection: the source code, the compiled code, the visual layout, the documentation, possibly even the aggregation of menu commands” (Goldman). By protecting the numerous aspects of our project, copyright provides us adequate security. Besides the advantages of the protection offered, the process is affordable and efficient. Copyright is automatic as soon as a work is completed, though to file for statutory damages, one must formally register for a fee of less than $100 and an application turnaround time of under a year (Copyright.gov). In addition, even prior to completion of the work, we can preregister with a detailed explanation of the work in progress.

All IP strategies come with risks and copyright is no different. While pursuing a strategy of trade secrets would make our code more private, we would risk losing our protection should the secret be compromised. Also, as a general security principle in the computer science field, only the bare minimum should be relied upon to be kept secret to minimize risk of loss. However, completely publicizing our code for our copyright can be equally dangerous as the competition could copy our code with only slight rewrites. To remedy this, we can limit access to the raw code and only publish the required first and last 25 pages of code needed to attain a copyright on the entire work. In addition to this measure, it is our unique dataset that is the source of our code’s advantage over our competitors, and this is already protected by our partner, Conviva, in its aggregated form as a trade secret,

We believe that the novelty of our code and the application of our techniques to our unique dataset would allow us to obtain a software patent. However, while a patent may be most effective at reducing our risk of litigation, we look to alternatives due to the current complexity of filing a software patent and the immense amount of financial

resources required to do so. Our research has led us to two very appealing alternatives: open sourcing and copyrighting. For the reasons stated above, we believe that while each of these alternatives have their own risks, their respective merits make them more appropriate for our use than patenting. Moving forward, we plan to employ open sourcing, as we expect that building a large, open community of support will encourage adoption and most benefit our products.

VII. Technical Contributions

Overview

Online Video Data Analytics refers to the process of using data generated by the consumption of online video to generate useful insights, which in turn create value for the content consumer, for the content provider, or for both. Having obtained online video consumption data through our partnership with Conviva Inc., our team, consisting of Benjamin Le, Pierce Vollucci, Wenxuan Cai, Yaohui Ye, and myself, approaches performing such analyses through two different bodies of work. The first of these comes in the form of our Smart Anomaly Detection toolset, which, given a stream of data, attempts to intelligently detect true anomalies in a way that is robust to noise and potential false alarms. The second comes through our work in Subscriber Analysis, which uses data from past content consumption to identify which subscribers are likely to “churn,” or discontinue their subscriptions.

There exists considerable overlap between the methodologies behind Smart Anomaly Detection and Subscriber Analysis, as both involve applying machine learning and statistical techniques to the same domain. However, they ultimately are two different problems that operate on different views of the data and have very different applications. Thus, in order to increase our overall productivity along these two distinct paths of work, yet still benefit from both teams’ findings, we have divided our five person team into two sub-teams. While Benjamin, Pierce, and Wenxuan have collaborated on Smart Anomaly Detection, Yaohui and myself have focused our efforts on Subscriber

Analysis. The majority of our work in Subscriber Analysis focuses on implementing and evaluating four different machine learning models: a k-Means clustering approach, Logistic Regression, k-Nearest Neighbors regression, and Random Forest. While Yaohui's responsibilities include overseeing the first two of these models, my responsibilities include the latter two. In addition, my responsibilities within the Subscriber Analysis subteam include learning how to use our tools, performing exploratory data analysis and extracting features. The following section focuses on my individual technical contributions towards the goals of both Subscriber Analysis and the team as a whole and is organized as follows. The Literature Review subsection provides a survey of existing related works and how they relate to our work in Subscriber Analysis. The Methods, Materials and Results subsection presents a breakdown of each of my individual tasks and elaborates on the methodology behind and challenges involved with each task. This subsection also displays and explains the most insightful performance results achieved by our models. The last subsection, Technical Contribution Conclusions, concludes with a summary of the collective findings and insights drawn through my work, as well as how it fits into the big picture of current churn analysis research.

Literature Review

The intuitive notion that improving customer retention leads to increased profits has been thoroughly verified in marketing literature (Zeithaml 84). While this is definitely true for subscription-based services in the rapidly growing online video industry, there has been little published research concerning subscriber churn in this specific industry (Trautman). Indeed, the majority of work around subscriber retention and churn analysis has been conducted in older industries such as newspaper publication and telecommunications. Adding another degree of separation between our work and existing research, we have found no existing research which attempts to address subscriber churn using only data about service usage and quality. To the best of our knowledge, all existing works rely on information such as viewer demographics and

information gathered through interactions between the subscriber and the service provider. For example, in their study of customer churn in the Korean mobile telecommunications market, Kim and Yoon had access to information about customer age and income (Kim and Yoon 758). Keaveny and Parthasarathy evaluated hypotheses for churn predictors across several online services, which closely relates to our industry of interest (Keaveny and Parthasarathy). However, their method of research consisted of analyzing information related to customer-company interactions, demographics, subscription types, and even self reports of satisfaction gathered through direct surveys. In contrast, the data we have available to us contains neither explicit information about the subscription nor demographic information. Thus, as a major contribution of this work, we infer and construct formulations of critical subscriber information such as viewer engagement and the definition of churning from this limited data set. In the ideal case, our implicitly-derived formulations accurately capture the semantics of their explicitly-given counterparts in existing work. Building upon these formulations, we apply regression models which go beyond the binary classification of churners, which existing work focuses on, and attempt to predict actual values for viewer engagement on a continuous scale.

Despite the considerable distance between our specific goals and those of existing research, there are several generally applicable techniques which fit well with our problem formulation. The majority of existing work identifies either Logistic Regression or Random Forest based models as the current state-of-the-art in the binary churn classification setting (Breiman; Chen et al.; Coussement and Van den Poel). While the Logistic Regression model falls under Yaohui's responsibilities, Random Forest falls under mine. Random Forest is a relatively young machine learning model that has quickly become extremely popular and is very widely used in practice today. This model, whose operation is explained in the following subsection, is extremely appealing due to its simple implementation, robustness, and the ease of interpretation of results (Coussement). The following Methods, Materials, and Results subsection describes my individual work as part of the Subscriber Analysis subteam.

Methods, Materials, and Results

My individual technical contributions within Subscriber Analysis can be categorized into four broad tasks: tool familiarization and Exploratory Data Analysis, feature extraction, implementation of k-Nearest Neighbors regression, and implementation of a random forest classifier. This section describes the work involved with completing each of these tasks.

Tool familiarization and Exploratory Data Analysis

The primary goals of this stage are to become familiar with our capstone team's specific toolset and to learn about the data. In particular, the former of these goals involves learning how to effectively use Apache Spark and Databricks Cloud. Apache Spark is an open source framework and engine for big data processing originally developed at UC Berkeley (Zaharia). We use Spark for its ability to simplify and optimize iterative distributed computation by, in addition to other things, handling the scheduling of jobs to different "worker" machines. Databricks Cloud is the primary product of our second industrial partner, Databricks. Through our partnership with the startup, we have access to their beta version of this new tool. In short, Databricks Cloud is an easy to use web console and programming environment for use with Spark that abstracts away the complexity behind the configuration and management of the worker machines. While Databricks periodically holds bootcamps and publishes videos of these bootcamps online that are quite useful, we find that the best way to learn about the tools and the programming model is to dive in and experiment with them.

The traditional approach when tackling a data science or analytics problem is to perform Exploratory Data Analysis (EDA) (Tukey). EDA is an unstructured investigation of the data with the aim of discovering characteristics and patterns in the data that become useful later. As previously mentioned, our dataset consists of a large number of "session summaries" generated by the viewers of a single anonymous content provider over several months. Our EDA and follow-up communication with Conviva engineers

led to our previously mentioned findings that several session summary fields are often empty and that the dataset included data generated by an artificial viewer used by Conviva for testing purposes. Discoveries like these justify our time spent in EDA and help us avoid making incorrect assumptions about the data that may otherwise lead to strange behavior and invalid results.

Feature Extraction

Having characterized the data and become familiar with the programming model, the next task is to identify and extract the elements of the data we will use in our models. Feature extraction is the critical process of identifying and pulling out information encoded within a dataset that are beneficial to the performance of a predictive model (Sun et al.). This stage is often one of the most time consuming and challenging parts of real machine learning problems. In theory, by combining and transforming the finite number of fields in the data, there are an infinite number of possible features and feature sets. Therefore, some method of pruning this search space is needed. Typically, at this stage, domain experts are consulted who are able to identify and help generate important features from the data. However, owing to our extensive interaction with the data and our own knowledge of online video services, we possess enough domain knowledge to act as the domain experts and manually extract the features ourselves. The resulting feature list is shown in Figure 1.

Each of the features shown is computed on a per-week basis for each of the four weeks preceding the week for which we want to predict viewer engagement. For the remainder of this paper, we shall refer to the week for which we are predicting engagement as week 5 and the four weeks preceding it as weeks 1 to 4 respectively, in sequence. Figure 2 illustrates this five-week “time window.” We use the granularity of one week to capture the extremely common case of weekly released content such as sports and television series (Marvin). As Marvin points out, weekly aggregations also incorporate viewing behaviors of the viewers themselves, such as watching videos at home during the weekends or during the week at work. While including a longer history

with more weeks of data would likely provide more information to our predictors, using longer histories is more computationally expensive for our models. With this tradeoff in mind, we choose to use four weeks of viewer history, reasoning that this is a long enough history to pick up temporal trends in viewer behavior and reveal the value of our features, while still being reasonable in terms of computational cost; we leave experimentation with longer histories in combination with a more refined model as future work.

Feature Name	Description	Hypothesis/Justification
totalplaysec	Total playing time	Engagement Metric. Included for autoregressive purposes.
nsessions	# sessions initiated	The number of sessions reflects how often a viewer watches
avgplaysec	Avg. playing time per session	Indirect measure of content type and quality of service
avgbuffersec	Avg. buffering time per session	Buffering time a per session directly affects viewer experience
avgpausedsec	Avg. paused time per session	Pause time could represent content type or viewing quality
avgjoinsec	Avg. join time per session	How long viewer has to wait for video start affects viewer experience
avgstopsec	Avg. stop time per session	Stop time could represent content type or viewing quality
avgsleepsec	Avg. sleep time per session	Sleep time could represent content type or viewing quality
ndevices	# unique devices	More engaged viewers may use multiple devices
nconntypes	# unique connection types	More connection types could mean more engagement
ncountries	# unique countries	Could show interesting viewer behavior and/or signal loyalty
fvsfs	Fraction of video start failures	Proportion of videos which failed to start reflects service quality
febvs	Fraction of exits before video start	Proportion of videos closed by the viewer before the video starts reflects service quality
fjoined	Fraction of successful video starts	Proportion of videos which started successfully reflects service quality
flives	Fraction of livestream sessions	Distribution of content types correlates with tendency to churn

Figure 1. Extracted Features (computed on a per week basis).

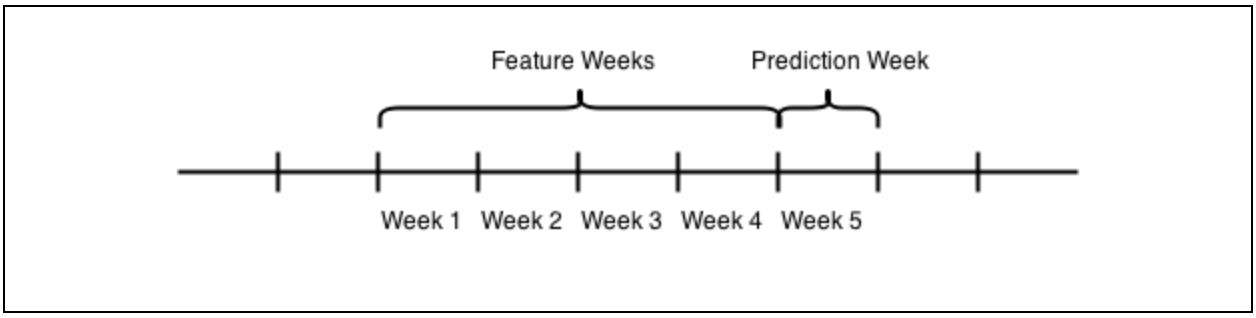

Figure 2. Partitioned “time window” for featurization and prediction

As shown in the top row of Figure 1, `totalplaysec` is used as both the output metric, whose week 5 value our models predict, and a feature type, whose values for weeks 1 to 4 our models use. A crucial choice we make here is to use total weekly playing time as our metric for viewer engagement. Total playing time over all of the videos a viewer watches in week is a good measure of how “engaged” a viewer is because it shows how much of the content actually ends up on the viewer’s screen. This definition, made in cooperation with Conviva engineers, is robust and accurate compared to the other possible metrics we have considered. For example, the use of weekly session counts would be vulnerable to misrepresentations because a session summary is generated each time a video begins to be loaded, regardless of whether the whole video played, an error occurred that prevented any video showing, or anything between. In order to help capture and quantify situations like this, we also include features such as the average buffering time and the fraction of sessions in which the video successfully started.

One non-obvious issue that must be addressed is what values the features above should take when a viewer has no sessions for a given week. Due to the fact that our data consists entirely of information generated when videos are viewed, we must infer when a viewer does not view any content. Because k-Nearest Neighbors, to be explained in the next subsection, requires each feature to have a continuous value, we must impute the values for the features for weeks when a particular viewer does not have any session summaries. While it is straightforward that we should impute a value

of zero for playing time, the same is not true for other features. For example, it is unclear what value to impute for average buffering time when no content was consumed. While a value of zero is technically accurate, this may be counter-productive and “confuse” a model because we would typically associate low buffering time with high quality of service and a tendency towards retaining viewers. As a result, the model may end up placing viewers with very low engagement closer to a cluster of highly engaged viewers than we desire, simply because we have imputed values that indicate these viewers experience low buffering time. On the other hand, imputing positive values would mean artificially inflating these values with inaccurate data. One reasonable approach is to use ratios of metrics like these rather than aggregate values. The issue with this however, is that a high-engagement viewer with many high quality sessions will be placed extremely close to low-engagement viewers which experience similar quality for the very small amount of sessions they have. It can be just as difficult, if not more difficult, to choose valid values for other feature types where there is little a priori knowledge about how the feature might segment different types of uses. Such is the case with the counts and fractional features which compose the bottom seven rows of the feature list above. Citing the lack of an established solution to this problem, we opt for the first option and impute values of zero for all of the features above. We note that this is not nearly as much of an issue for Random Forests, as they can simply ignore, i.e. never choose to split on, features that would increase or perpetuate such “confusion.” Furthermore, in general, models which allow categorical features—features whose values can take on a finite number of values which do not require an explicit ordering—are better suited to handle these situations, because a special value, e.g. “Not Available,” can be incorporated.

Another related issue concerns the implicit weighting of the features. The fact that the features shown take on extremely different ranges of values, features with very large variances, total playing time for example, will dominate the distance computation and the features with smaller variances will essentially become ignored. This is clearly undesirable, because this makes it difficult to distinguish helpful predictor variables from

unhelpful ones. To address this problem, we run one iteration of each experiment with raw features and another with normalized features, where each feature is brought down to the same range by dividing by the maximum value.

One final item to note is that several of the features shown above are highly correlated. For example, it is clear that when the *nsessions* feature, which represents the number of sessions per week, has a high value, the `totalplaysec` feature, which represents the total playing time per week, will also tend to be high, and vice versa. Fortunately this fact is not a critical issue for k-Nearest Neighbors, because clusters are preserved along correlated features (the addition of correlated features can be thought of as a translation of these clusters in the feature space). Similarly, Decision Trees and Random Forests are minimally affected because they incorporate relationships between variables into the sequence of splits defined by each query path. However, when considering other models or processing of these features, this correlation should be taken into account.

k-Nearest Neighbors Regression

Having established a richer feature set, the next task is to utilize these features by actually implementing the k-Nearest Neighbors regression model (Altman). k-Nearest Neighbors regression predicts a value for a sample by taking the average of the k "nearest" neighbors to the sample. Exactly which neighbors are "nearest" is determined by a distance function that is computed in the space defined by the features. k-Nearest Neighbors (kNN) is a non-parametric model; it does not compute any explicit weights or probabilities, because this information is implicitly encoded in the observed data, which the model must keep track of. While this means that there is essentially no training time, kNN is expensive in terms of storage and computation during prediction since the model must locate the k closest points in the feature space. Still, as described by Altman, non-parametric models are appealing when there is little a priori knowledge about the structure or distribution of the samples in the feature space (Altman 175). This is appropriate for Subscriber Analysis because we expect viewers to cluster into

archetypal subpopulations that exhibit similar behavior, but we do not know what all of these clusters look like or where they are located in the feature space ahead of time. Ideally, one or more of these subpopulations would consist primarily of churners and these viewers would be relatively far from other subpopulations of non-churners. Rather than reimplement this well-known algorithm ourselves, we utilize the existing implementation of kNN made available by the Scikit-learn Python library. Although this saves some redundant work for us, the considerable task of configuring this model specifically for Subscriber Analysis still remains. We perform this through hyperparameter tuning and feature selection.

Evaluating Performance

Tuning our kNN model requires evaluating performance on the training data. Modeling a real world use case, all training and validation done during tuning and feature selection are performed on data gathered for a fixed "time window," where the term "time window" refers to a five-week span as defined above. The training data, consisting of about 440,000 distinct viewers, is split into a training set and a validation set through a randomized partitioning of viewers. We train on all viewers for a single time window and test by sliding the time window forward up to ten weeks. That is, each time we slide the window forward, week i in the previous window becomes week $i-1$ in the next. For each of these windows, we generate features on the first four weeks and evaluate performance on the last week. Our final performance numbers are obtained by averaging the performance on each of these ten test weeks. Our primary performance metric for kNN in the following methods is the R^2 standard coefficient of determination (Jones). The R^2 score, shown in Figure 3, is one of the most commonly used metrics for evaluating regression models. The metric compares the performance of the predictor with the performance achieved by simply predicting the mean of the observed values. A score of one is achieved by a perfect predictor with zero error on every test sample, a score of zero means a predictor performs equally as well guessing the mean, and lower scores are worse.

$$R^2 = 1 - \frac{\Sigma (y_{true} - y_{predicted})^2}{\Sigma (y_{true} - y_{mean})^2}$$

Figure 3. Formula for R^2 regression evaluation metric.

Hyperparameter Tuning of *k*

As with most other machine learning algorithms, kNN has case-dependent hyperparameters that need to be manually tuned for the data. The most prominent parameter for kNN is k, the number of neighbors to consult when predicting engagement for a sample. This value is critical to the success and correctness of kNN, because a value too small may lead to a model too sensitive to noise and a value too large may disperse clusters we might otherwise achieve. In order to find the optimal value of k while avoiding overfitting, we tune this parameter using multifold cross-validation (Kohavi, Zhang). Figure 4 shows the results from the tuning of hyperparameter k.

The two plots correspond to our two preprocessing steps: raw feature values and normalized or "normed" feature values. To avoid overfitting on a specific subset of features, for both preprocessing steps, we fix three different subsets of weekly features and evaluate the performance over different values of k. The first feature subset contains all features for all four weeks, the second contains only `totalplaysec` for each of the four weeks, and the final feature subset gives the model no information to train on. While the first two feature subsets approximate the minimum and maximum number of features, the third subset serves as a sanity check for our model. In this case, the feature subspace consists of a single point and a sample's k nearest neighbors are just a random subset of the points. As we would expect, as k increases, this essentially "blind" model more closely emulates simply predicting the mean and its R^2 score approaches zero. The other two feature subsets significantly outperform predicting the mean and yield more interesting results.

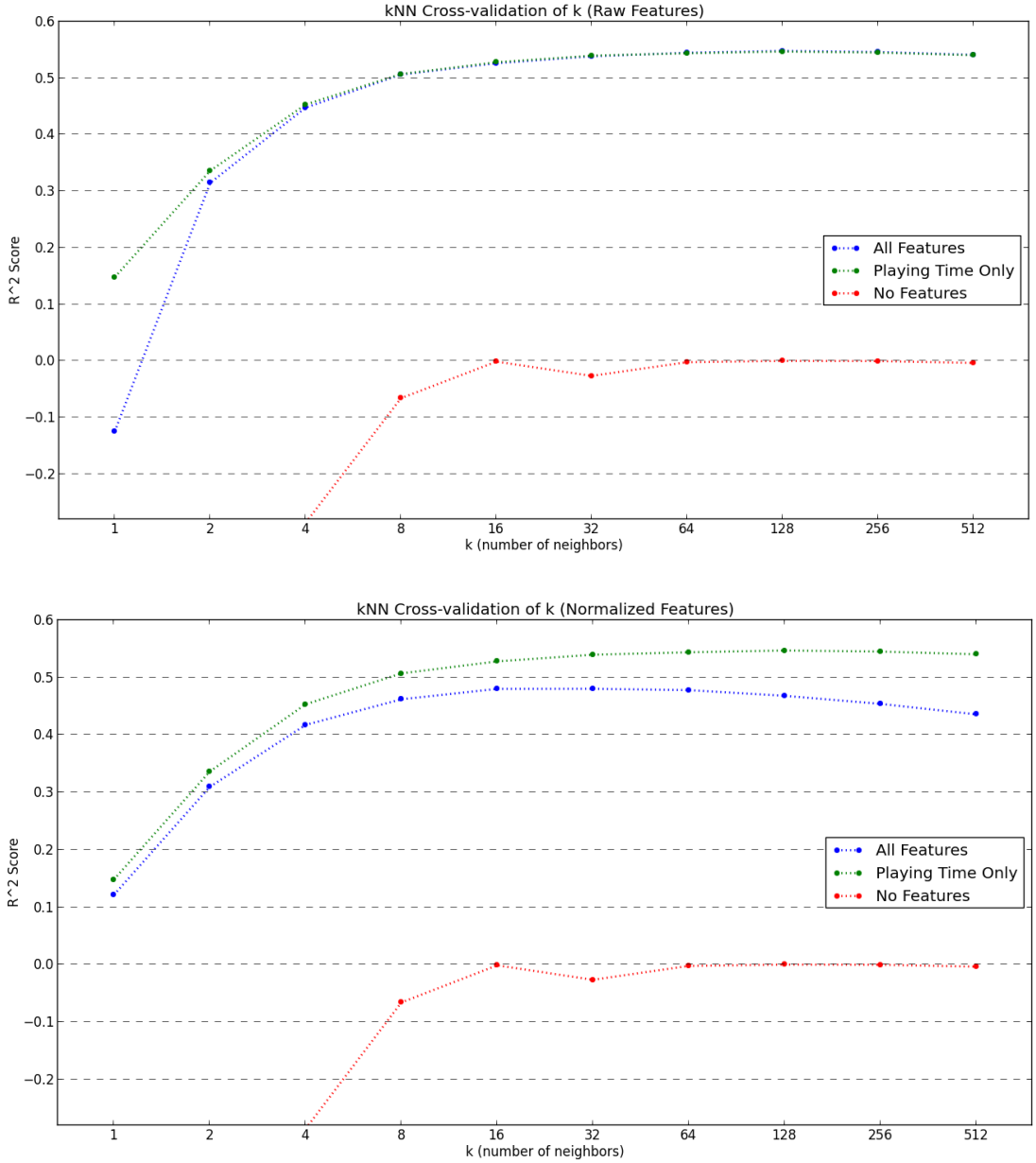

Figure 4. Hyperparameter Tuning of k for raw (top) and normalized (bottom) feature values.

We can see that for both normalized and raw feature values, using playing time alone yields scores close to those yielded by using all of the features. With raw feature values, the two algorithms exhibit almost exactly the same performance. With

normalized feature values, however, using playing time alone actually outperforms using all of the features. This suggests that past playing time is a helpful predictor for future playing time and that simultaneously using all feature types adds detrimental noise. The nearly identical performance of the two feature subsets when using raw feature values can be explained by the previously mentioned fact that both the variance and value of total playing time are typically magnitudes larger than those any of the other features. This essentially gives the feature a very large implicit weight in the calculation of distance in the feature space. This means that when we do not normalize, total playing time dominates the distance calculations, so the noise introduced by the other features essentially does not have much effect. We note that at small values of k, performances increase significantly with k, but the curves quickly level off. This is expected behavior, because at small values of k, predictions are very sensitive to noise. In both preprocessing types, for the roughly 440,000 viewers in our training set, the best cross-validation performance is achieved when k is 128. Past this optimal value of k, performance slightly decreases.

Feature Selection

One issue that kNN is vulnerable to is the curse of dimensionality (Indyk and Motwani). This problem arises when the dimension of the data representation grows too large and causes the data to become very sparsely distributed in the feature space. If it so happens that some of the features that we extracted are not helpful predictors, these features may simply add an extra dimension of noise and disperse the type of clusters we want. Since we produce 15 features for each of the four feature weeks, for a total of 60 features, the curse of dimensionality poses a real threat for us. In order to address this problem, we perform dimensionality reduction through feature selection.

Feature selection prunes the set of features by removing non-contributing features. Because a brute-force, exhaustive search of every possible subset of the feature set would require an impractical amount of resources, the search space must be pruned. Guyon and Elisseeff suggest using the method of Forward Feature Selection

(Guyon and Elisseeff). This greedy algorithm, shown in pseudocode in Figure 5, incrementally builds a “best” feature set by iterating over the features and keeping exactly those that lead to an improvement in performance.

```
function forwardFeatureSelection(model, data, full_feature_set)
{
	best_feature_subset = []
	best_score = 0
	for feature in full_feature_set
	{
		cur_feature_subset = concatenate(best_feature_subset, feature)
		cur_dataset = featurize_dataset(cur_feature_subset)
		cur_score = model.train_and_cross_validate_score(cur_dataset)
		if cur_score > best_score
			best_score = cur_score
			best_feature_subset = cur_feature_subset

	}
	return best_feature_subset
}
```

Figure 5. Forward Feature Selection

While quick to perform and quite popular, Forward Feature Selection alone explores a very narrow search space and its result is highly dependent on the order in which the elements are traversed. To slightly improve on these weaknesses, we employ the slightly modified version of forward feature selection shown in pseudocode in Figure 5. For each of the 15 feature types, we obtain the cross-validation score for a model which uses only that feature for each week from week 1 to week 4. We point out that although we describe a feature vector that includes one of these feature types as having a subset size of one, in reality, it adds four elements to the feature vector (one for each week). From these subsets of size one, we select the best performing subset and evaluate each feature subset of size two that includes the best subset of size one. We continue increasing the size of the feature subset by always adding the feature type that yields the best performance, with the final iteration evaluating the performance of the full feature set. Although this modified feature selection process explores a larger space than standard forward feature selection, it is still far from exhaustive and it is possible that the optimal feature subset is missed.

```
function modifiedForwardFeatureSelection(model, data, full_feature_set)
{
	best_feature_subset = []
	best_score = 0
	cur_feature_subset = []
	repeat full_feature_set.size() times
	{
		remaining_features = full_feature_set - cur_feature_subset
		best_feature_to_add = NULL
		best_score_for_size = 0
		for feature in remaining_features
		{
			cur_feature_set = concatenate(cur_feature_subset, feature)
			cur_dataset = featurize_dataset(data, cur_feature_set)
			cur_score = model.train_and_cross_validate_score(cur_dataset)
			if cur_score > best_score_for_size
				best_score_for_size = cur_score
				best_feature_to_add = feature
		}
		cur_feature_subset.add(best_feature_to_add)

		if best_score_for_size > best_score
			best_score = best_score_for_size
			best_feature_subset = cur_feature_subset

	}
	return best_feature_subset
}
```

Figure 6. Modified Forward Feature Selection

Figures 7a and 7b show the results of our modified version of forward feature selection for weekly features for raw and normalized feature values respectively. For each subset size x along the x-axis, we plot the R^2 score of every possible subset of that size that includes the best performing subset of size $x-1$. We can see that for both pre-processing types, while there is a very open spread of R^2 scores for the different subsets of size one, after selecting this best single feature there is no significant improvement in performance over the rest of the evaluated subsets. Unsurprisingly, though not explicitly shown in the plots, this best feature is invariantly `totalplaysec`. For raw feature values, the results are so similar that it is hard to discriminate the individual markers on the plot. Again, we can attribute this to the large implicit weight placed on total weekly playing time; the performance of the model is only minimally affected by the presence of the other features. In the case of normalized feature values, we see a slightly larger spread of performances at each step. These points are representative of the varying amounts of noise each additional feature adds to the

model. The decrease in performance at the largest subset sizes is a result of our forcing the subset size to grow. Thus, more noisy variables are continuously added to the feature set.

We can see that there are quite a few points in the single feature subset case with positive R^2 scores, representing hopeful performance for features besides `totalplaysec`, the top point. Some of the top performing single features include the `fvsfs` and `fjoined` features. However, the absence of significant performance increases in larger subsets shows that, when combined with the information given by past total playing time, these predictors do not add information in a way that improves predictions. We obtain the final models we use for evaluating on the test set by taking the highest performing feature subset for each preprocessing type. As the graphs suggest, however, there is almost no difference in performance between the best subsets for each subset size. The implies that the differences in composition of our feature subsets are likely due to noise. This is reflected by the fact that the only consistently appearing feature across the best performing feature subsets is the total weekly playing time.

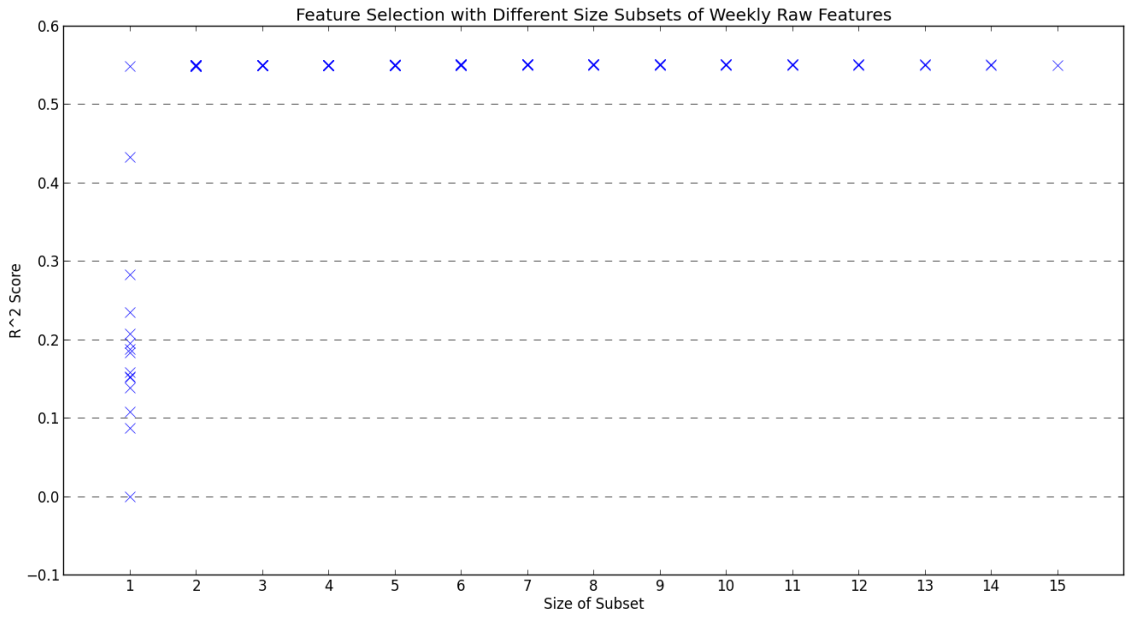

Figure 7a. Modified Forward Feature Selection for raw features

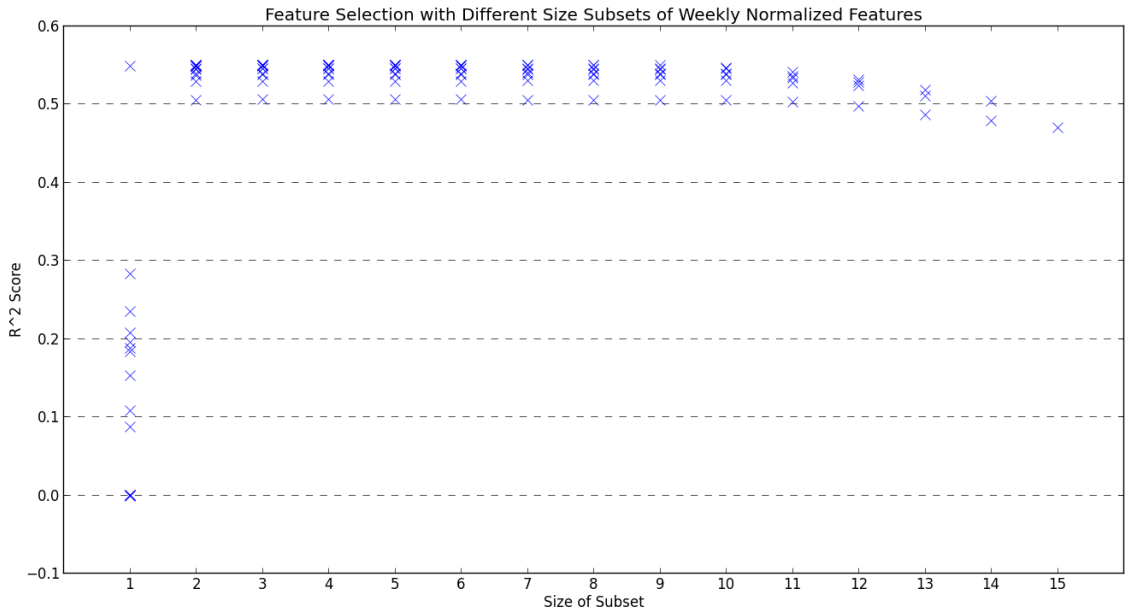

Figure 7b. Modified Forward Feature Selection for normalized features

In order to gain further insight into how “good” our feature types are and how worthwhile searching for the optimal combination of our current feature types would be, we also perform a standard forward feature selection process over the 60 possible features in the feature vector, considering each feature from each week separately from any other. The resulting plot is shown in Figure 8. For both processing types, the plot shows four large steps with little change in performance over the 15 feature indices between each step. These steps correspond to large increases in performance resulting from the inclusion of the `totalplaysec` value for each of the four weeks of viewer history we include in our feature vectors. Thus, these results provide convincing evidence for what the previous results suggested: playing time is by far the most helpful predictor.

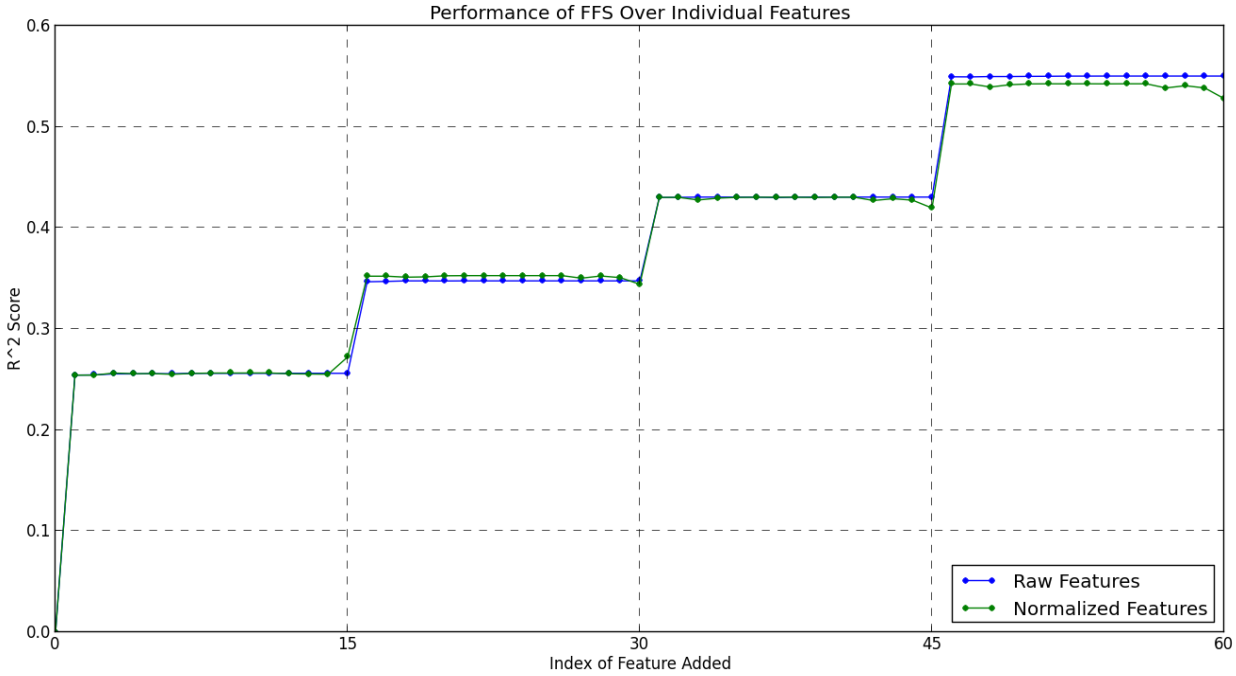

Figure 8. Forward Feature Selection Considering Each Feature Individually

Figures 9 and 10 below present the best models found through our two feature selection approaches and their performances on the test set. Interestingly, despite the very different feature spaces used by each of these models, the models' R^2 scores are nearly identical at around 0.54-0.56. This means that the models perform much better than simply predicting the mean, but they are still far from perfect. For a more concrete idea of their performance, the figure also shows the average error in the models' predictions both in terms of seconds and as a percentage of the mean total playing time per week. For numerical stability, the latter of these values is calculated by dividing the mean error by the mean true value. Again, the models are nearly identical with respect to these numbers. Each model yields slightly over 11 thousand seconds average error, which translates to mispredicting weekly playing time by over 3 hours. According to the percentage error, this average is over 73% off of the average true weekly playing time of 4.6 hours. Thus, while better than the most naive of models such as predicting the mean, our best kNN models are still quite inaccurate.

Statistic / Preprocessing Type	Best Feature Subset	R^2	Mean Error (seconds)	Average Percentage Error
Raw	totalplaysec *avgplaysec* *ncountries* *flives*	0.542	11645	73.5%
Normalized	*totalplaysec* *nsessions* *avgjoinsec* *avgstopsec*	0.559	11332	72.9%

Figure 9. Performance of Best Models from our modified Forward Feature Selection on Weekly Features

Statistic / Preprocessing Type	Best Feature Subset (Weeks features are included)	R^2	Mean Error (seconds)	Average Percentage Error
Raw	*totalplaysec* (1 2 3 4) *nsessions* (1 2 3 4) *avgplaysec* (2 3 4) *avgbuffersec* (2 3 4) *avgpausedsec* (2 3) *avgjoinsec* (4) *ndevices* (2 3 4) *nconntypes* (4) *fvsfs* (4) *febvs* (4) *fjoined* (4) *flives* (4)	0.542	11679	73.3%
Normalized	*totalplaysec* (1 2 3 4) *avgplaysec* (1 2 3 4) *avgbuffersec* (2 3) *avgpausedsec* (1 2) *avgjoinsec* (1 2 3 4) *avgstopsec* (1 2) *nconntypes* (4) *fvsfs* (4)	0.553	11358	75.4%

Figure 10. Performance of Best Models from Forward Feature Selection on Individual Features

Importantly, by inspecting the best feature subsets resulting from feature selection, we can see that the only feature that appears in all experiments is total weekly playing time. Because of the implicit weight of total playing time, the outcomes of feature selection with raw feature values should be taken with a grain of salt; the features that appear are more likely to have appeared due to noise. The result of using normalized feature values, on the other hand, does not require such a disclaimer. Interestingly, we see slightly better performance in terms of all three metrics, with the exception of percentage error for individual features, for normalized feature values compared with raw feature values. This suggests that there is in fact at least one other feature besides total playing time that, when allowed equal weight to total playing time, helps predictions slightly. While the intersection between the weekly features case and individual features case is small for normalized features, there is one feature that appears hopeful. The `avgjoinsec` feature, representing average join time, appears in the best feature set for weekly features and is included for every week in the individual features case. This suggests that average join time, the time it takes for a video to begin playing, may in fact be a useful feature.

In order to better understand the predictive behavior of these kNN models, we present a confusion matrix and a histogram of predicted values for each of these models over a single test week in Figure 11. A confusion matrix is a tabular histogram generally used to visualize the characteristics of a classification model. It shows how many times each true class is classified as every possible class. The confusion matrix for a perfect model would show large counts along the diagonal and zero counts elsewhere, while a classifier that assigns class labels randomly would yield a confusion matrix with counts uniformly distributed across all cells. We adapt this slightly for regression and our value range by taking the log (base 2) of our predicted values and then discretizing the resulting values by rounding to the nearest integer. Because these plots look nearly identical for both raw and normalized feature values, and in the interest of space, we show only the plots for raw feature values.

When using a k value of 128, the confusion matrix shows two "patches" of high counts. We note that gaps between these patches are due to the fact that we use a log scale and if not zero, total playing time tends to be large, at least around 1000. We see high counts in the log range of 10 to 15 because these encompass a larger range of values than the smaller numbers and in most cases, playing time does not exceed 2^{15} seconds. The traditional histogram provides a better idea of the distribution of actual playing times. It also shows the distribution of predictions the model generates. The first of the patches occurs along the bottom row and shows that this kNN model inaccurately predicts the playing time for the large number of viewers who actually have zero playing time in week 5 (furthermore "zero viewers"). The second patch shows a rough alignment along the diagonal of the matrix. This indicates that aside from the severe overestimation of engagement for zero viewers in the first patch, the model actual generates reasonable predictions.

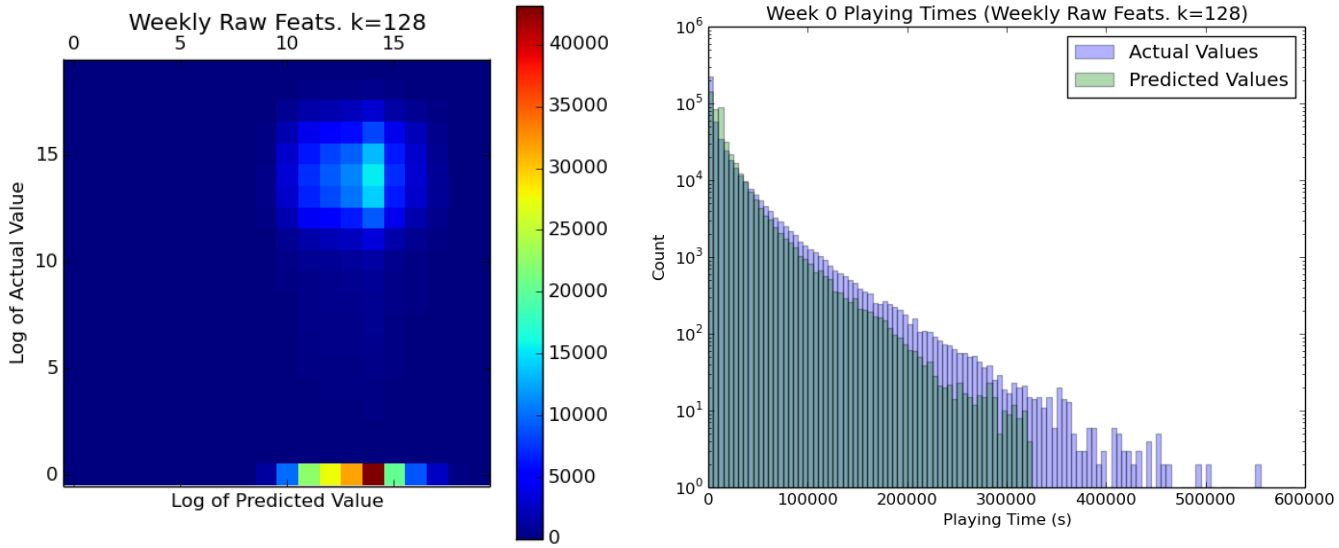

Figure 11. Confusion Matrix and Playing Time Histogram for raw feature values and k = 128

After manually inspecting the data and predictions, we conclude that this dichotomy in performance can be explained by the following hypothesis: individual

viewing behavior is often sporadic and our features do not indicate when viewers will have a week with no viewing time. While we acknowledge that there is a variance inherent to viewer behavior, we cannot conclude that this is the limiting factor to our models based solely on their performance. The most we can say is that viewers behave sporadically when represented in the feature space we have defined. In other words, in our feature space, the zero viewers are collocated with non-zero viewers with large week 5 playing times; the zero viewers look just like the non-zero viewers and they are deeply integrated into the clusters we have in our feature space. In order to verify that this collocation of zero viewers with non-zero viewers is happening at a fine granularity, we repeat the experiment the experiment with smaller values of k. Figure 12 shows the plots resulting from using a k value of 4.

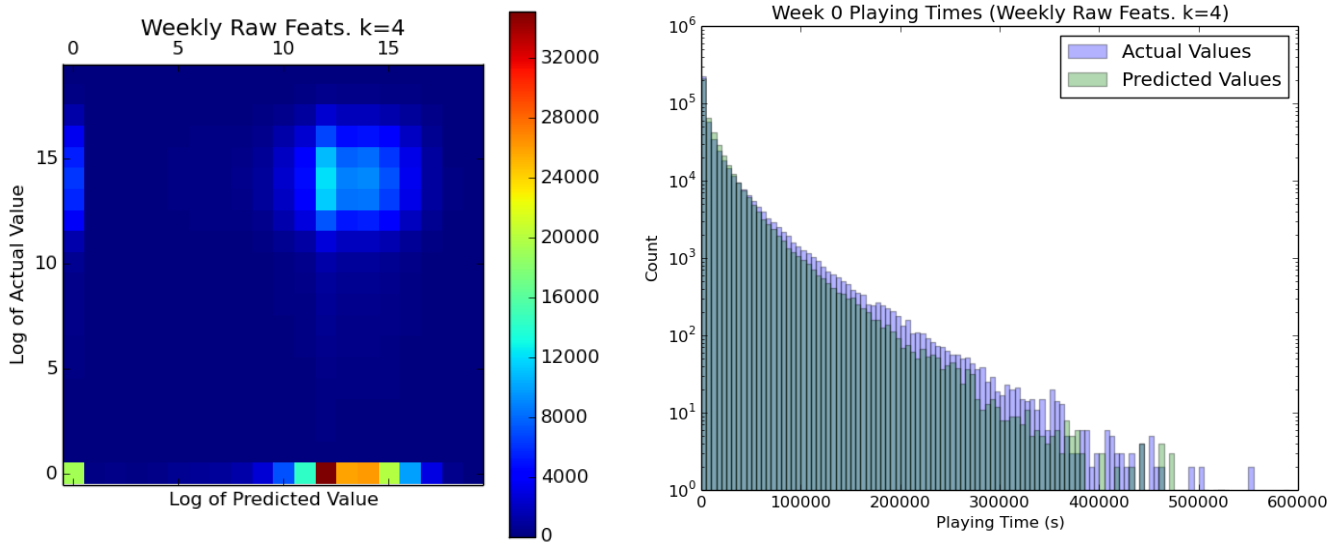

Figure 12. Confusion Matrix and Playing Time Histogram for raw feature values and k = 4

We can see that by using this smaller k value, the model is able to predict a large number of zero playing times. However, even at this granularity, the first of the patches described above is still very prominent. This demonstrates that these zero-viewers are indeed being collocated right alongside the non-zero viewers and verifies our second hypothesis from above. One final item to note is that both the

confusion matrix and the playing time histogram for the show that there is considerably more variance in the predicted values when compared with the higher k value case, and yields an R^2 score of 0.426558. This verifies the previous statement that lower k values translate to models that are more sensitive to noisy data.

Regression to Classification

For a final perspective on our kNN model, we examine its performance when used for the classification of low-engagement viewers. While it is possible and in fact quite common to use kNN directly as a classifier (by using the labels of neighbors as votes rather than taking a mean of some value), there are some benefits to using the regression version as a means for classification. These include obtaining a better idea of the distribution of the predicted value, total playing time in our case, as well as being able to easily modify how classes are defined and view the corresponding change in performance even after the regression predictions are made. This latter benefit is exhibited when we choose a simple labeling technique such as applying a threshold value to binarize the predicted total playing time.

For example, we can choose a threshold, e.g. 10 minutes, and declare that any viewers with a week 5 total playing time below this threshold value, furthermore "`min_engagement`," are low-engagement viewers. A correct classification would mean that our regression model predicted a value on the same side of this threshold value as the actual value. Similarly, an incorrect classification would mean that our regression model predicted a value that exists on the other side of the threshold. If we define low-engagement users as the "positive" class and high-engagement users as the "negative" class, we can divide correct classifications into True Positives, in which we correctly predict low-engagement viewers, and True Negatives, in which we correctly predict high-engagement viewers. The analogous division for misclassifications divides the cases into False Positives, in which we predict low-engagement for an actual high-engagement viewer, and False Negatives, in which we predict high-engagement for an actual low-engagement viewer. These definitions allow us to evaluate the

performance of our classification process for a given `min_engagement` value. However, the choice of this value is critical not only because it defines whom the user of the model, e.g. an online content provider, is targeting, but also because it highly affects the classification performance. Therefore, we want to evaluate our model's performance over a number of `min_engagement` values. In particular, examine the True Positive Rate (TPR), the fraction of all positive class members correctly labeled, and the False Positive Rate (FPR), the fraction of all negative class members incorrectly labeled.

We note that a widely used plot for evaluating the performance of a binary classifier over different values of a decision threshold is the Receiver Operating Characteristic (ROC) plot (J. Hanley). While very appealing and almost perfect for our setting, there is one fundamental and critical reason we cannot use a standard ROC plot: in our case, changing the value of `min_engagement` actually changes the true labels of our samples. ROC plots are intended to be used in a setting where the threshold being changed only affects the predictions made by the classifier. Thus, we present a simple plot of the TPR and FPR curves, based on predictions for a single test week, in Figure 13. Again, because the plots for normalized and raw features are nearly identical, we show only the plot for raw features.

The two curves shown are the result of plotting the TPR and FPR for different values of `min_engagement`, with each generated point separated by 10 minutes. The first point in the bottom left corner corresponds to labeling with a `min_engagement` of zero. This labeling translates to a highly selective definition of low engagement that includes only viewers with zero playing time. The low TPR shows that at this high selectivity, the classifier is not able to recognize many of the low-engagement viewers. At the same time, the low FPR shows that the classifier does not falsely label high-engagement viewers as low-engagement viewers. As we incrementally increase `min_engagement`, we see that though both TPR and FPR rise, TPR rises much more quickly than FPR. This shows that when the goal is to detect viewers with very low viewer engagement, our kNN model appears to be reasonably accurate. More

concretely, if the goal is to predict which viewers will have less than 180 minutes of playing time in the coming week, our kNN Regression-Classifier will correctly identify 70.9% of these low-engagement viewers while misclassifying only 20.4% of high-engagement viewers as low-engagement viewers. Furthermore, if the goal is more selective, for example, to identify which viewers have less than 60 minutes of playing time in the coming week, we would expect TPR and FPR values to be about 42.6% and 7.3% respectively. These results suggest that overall, our kNN Regression-Classifier is effective at avoiding False Positives. In more technical terms, our classifier exhibits fairly high precision.

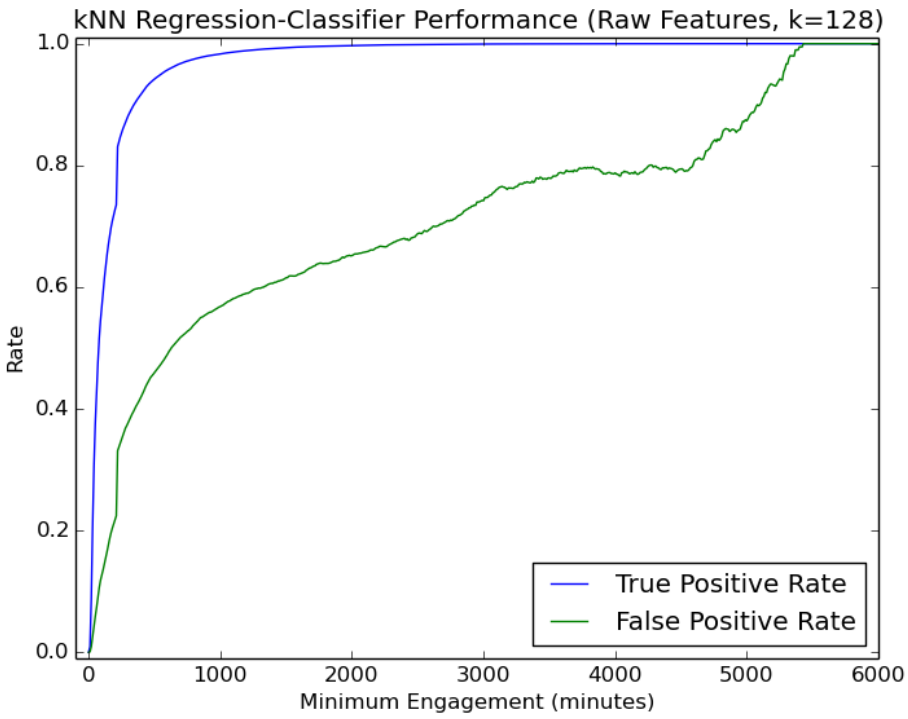

Figure 13. Performance of kNN Regression-Classifier

In summary, though our kNN Regression model has not shown to be accurate in terms of predicting actual total playing time, we have seen promising results from using the model's predictions as a basis for classification. In particular, for a wide range of "minimum engagement" thresholds, our classifier has shown high precision and can

identify a relatively large portion of low-engagement viewers while misclassifying a small portion of high-engagement viewers.

Random Forest

My final task involves implementing a Random Forest classifier. The Random Forest classifier is an ensemble method which uses multiple Decision Tree classifiers as “weak learners” and produces an output classification based on the outputs from these weak learners (Breiman). Decision trees are well known for their ability to construct a series of logical predicates or “splits,” each of which further partitions the data. These splits partition the viewers based on the features of the data and compose the nodes of the tree, such that more informative features appear closer to the root of the tree. One extremely important benefit of this is that these splits can be examined after the tree is constructed to identify the most valuable predictors. In a Random Forest, each decision tree is constructed in a randomized manner so that the forest uses the results from a diverse set of trees. This diverse population of trees has proven to be extremely effective in reducing overfitting, which is the primary weakness of individual decision trees.

Evaluating Performance

In contrast with regression techniques, which predict a continuous variable that already exists in the data, there is no explicit information in the data about whether a given viewer churned. Thus, in order to apply the Random Forest model, we must first define what these two classes are and label the viewers accordingly. Through careful deliberation over what is meaningful and important to users of Subscriber Analysis, we define likely churners for a given week as active viewers who exhibit a dramatic drop in engagement near that week. The drop in engagement at a given week is defined by the ratio of mean viewing engagement over that week and the three following weeks to the mean viewing engagement over the four preceding weeks. Pseudocode for the algorithm is shown in Figure 14.

```
function isChurner(viewer, week_number)
{
        engagement_after = mean(total play time for weeks [week_number to week_numer+3])
        engagement_before = mean(total play time for weeks [week_number-4 to week_number-1])
        engagement_drop = engagement_after / engagement_before

        if engagement_before >= active_requirement and engagement_drop <= churn_cutoff
                return true
        else
                return false
}
```

Figure 14. Pseudocode for finding likely churners.

The two bolded variables above, `active_requirement` and `churn_cutoff`, represent the minimum for average weekly playing time for “active” and the minimum relative engagement decrease to say that an “active” viewer has become a churner respectively. The exact values of these two parameters directly define how selective the definition of churners is. As such, these two values can be thought of as parameters to the labeling process and it is up to the user of Subscriber Analysis to define. For our evaluation of the Random Forest model, we use an **`active_requirement`** of five minutes and a `churn_cutoff` of 0.3.

To obtain our training set, we apply the above labeling algorithm to every distinct viewer across an eight week window. Just as for kNN, we only include data from the first four weeks of this window to generate feature vectors. We only use the latter four weeks to obtain the viewing activity we need during this labeling process, so we are careful not to use information from the “future” to make predictions. Again, just as with kNN, we obtain our test set by sliding this eight week window forward in time up to ten weeks, evaluating the performance of our model for each of these weeks. Each of these time windows, of which there is one in the training set and ten in the training set, contains roughly one million distinct viewers. Due to the large size of this dataset, instead of performing cross-validation for each iteration of tuning as we did for kNN, we exclude a portion of the training set to serve as a dedicated validation set. Our

configuration of `active_requirement` and `churn_cutoff` mentioned above results in roughly 17% of viewers being labeled as churners in each data set.

To build our model, we use an existing implementation of Random Forest classification provided by MLlib, Spark's specialized machine learning library. For the Random Forest model, there are two primary hyperparameters: the maximum depth of each tree and the number of features that will be considered for splitting at each node. Technically, the number of trees to grow in the forest is another hyperparameter. However, the implementation of Random Forest integrates sufficient randomization such that adding additional trees can not hurt predictive accuracy of the forest; the additional weak learners can only help make the model better or not affect it when there is already a very large number of trees. Thus, in order to bound the computational load as we explore the model, we set the size of our forests to be 10 trees and keep in mind that we can always increase this number for slightly better performance later. In addition, by following convention and utilizing existing work, we find it unnecessary to manually tune the number of features to split at each node. In his work, Breiman has shown that fixing this hyperparameter to the square root of the number of features is very effective and near optimal in many cases (Breiman). By adopting this approach, the task of configuring our Random Forest model is reduced to determining the best maximum depth to grow each tree.

Our primary performance metric for our Random Forest Models is the misclassification rate. This metric is calculated by dividing the number of misclassifications by the total number of test samples. In the ideal case, we would generate a ROC plot and obtain the coupled Area Under Curve (AUC) metric to gauge performance (Coussement and Van den Poel). However, since the current version of Random Forest in MLlib does not allow retrieval of the prediction probabilities these plots require, we leave this as future work. As a baseline model for comparison, we use one which predicts that all viewers are non-churners. While this might seem an unreasonably naive baseline, our labeling process classified only 17% of viewers as

churners. As a result, this baseline would achieve a respectable 17% misclassification error rate and beating this performance is nontrivial.

Tuning Maximum Tree Depth

Figure 15 shows the performance of our Random Forest models from training with different values of maximum tree depth.

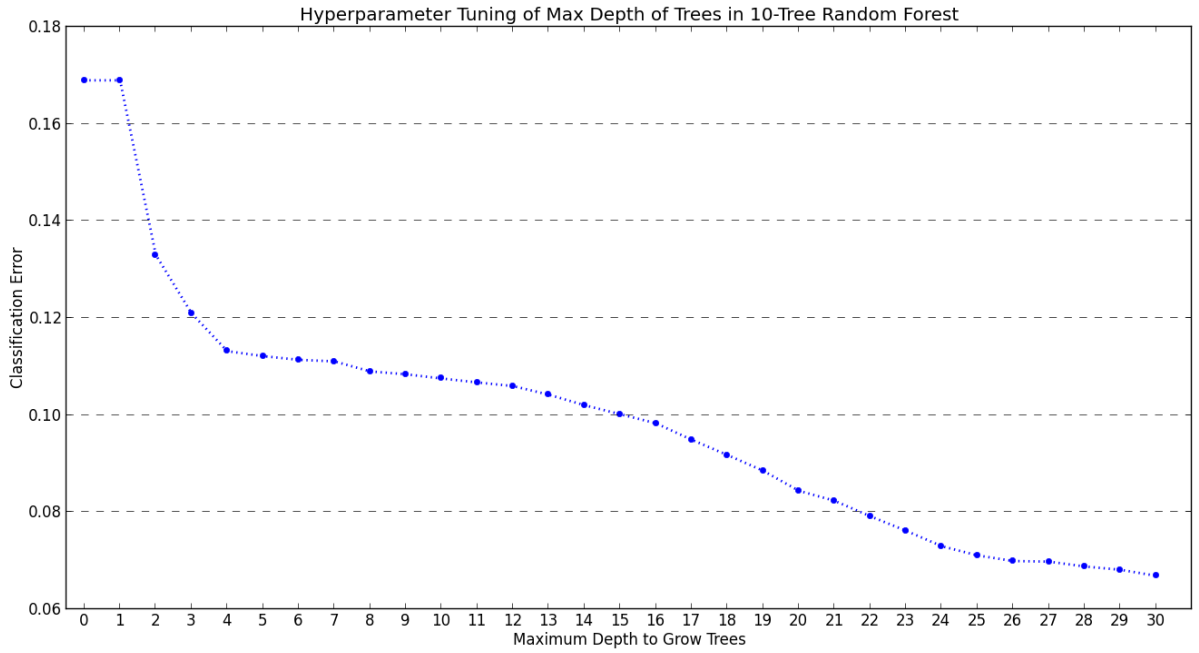

Figure 15. Max Depth Tuning for Random Forests

As the plot shows, when the maximum depth is zero or one, the error rate is the same as the equal to the distribution of churners. It turns out that for these depths, the trees are not able to perform enough splits to separate churners from non-churners. The result is that almost every leaf node is composed primarily of the majority class, non-churners, so the model predicts every viewer to be a non-churner. Since only 17% of the viewers are actually churners, the error rate for these models is 17%. However, as we allow the trees to grow larger, the trees are able to query more features and better separate churners from non-churners. The largest maximum depth we attempt is

30 because this is the largest value for the parameter allowed by MLlib's implementation of Random Forests.

Up to this maximum depth, the error rate continues decreasing. Although Random Forests are intended to avoid overfitting, the continuously decreasing training error suggests that overfitting is still occurring. Indeed, this model's classification error of about 12.3% on the test set shows that there is some level of overfitting on training data. Still, at 5% more accurate than the baseline, this is an impressive error rate. The confusion matrix in Figure 16 verifies that the model is in fact correctly labeling the majority of viewers. The "0" class represents non-churners while the "1" class represents churners. Although there is a much larger number of non-churners than churners, the high counts along the diagonal squares show that the model is in fact classifying correctly.

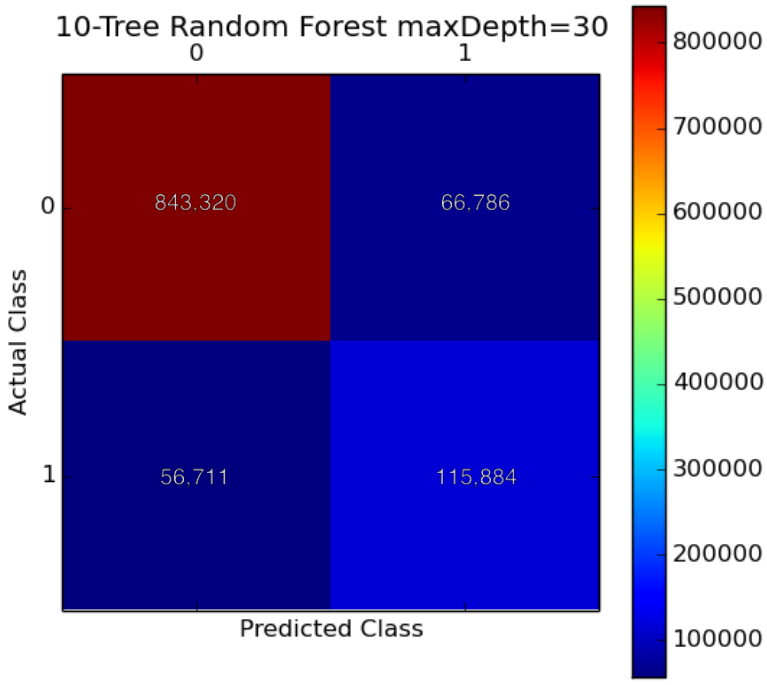

Figure 16. Confusion Matrix with counts for our best Random Forest model

Inspecting Trees

In order to understand how the model is able to separate churners from non-churners, we might examine the splits which define the nodes of these trees. One complicating issue with directly examining the trees of a Random Forest, however, is that we cannot know which random subset of features were considered when splitting at each node. This makes it difficult to interpret which features are the most important when examining the trees. To avoid this complication, we instead construct a single decision tree for the purpose of investigation. We use the same construction algorithm as for the trees in our Random Forest models, with the exception that we consider every feature for splitting at every node instead of a random subset of features. The diagram in Figure 17 shows the structure of such a single tree, grown to a maximum depth of four in the interest of space.

```
If (Week 1 totalplaysec <= 0.0)
   If (Week 2 totalplaysec <= 0.0)
      If (Week 3 totalplaysec <= 0.0)
         Predict: 0.0
      Else (Week 2 totalplaysec > 0.0)
         If (Week 4 avgjoinsec <= 8.75E-4)
            Predict: 1.0
         Else (Week 4 avgjoinsec > 8.75E-4)
            Predict: 0.0
   Else (Week 2 totalplaysec > 0.0)
      If (Week 4 avgjoinsec <= 8.75E-4)
         Predict: 1.0
      Else (Week 4 totalplaysec > 8.75E-4)
         Predict: 0.0
Else (Week 1 totalplaysec > 0.0)
   If (Week 4 avgjoinsec <= 0.0)
      If (Week 2 avgjoinsec <= 9.285714285714286E-4)
         Predict: 1.0
      Else (Week 2 avgjoinsec > 9.285714285714286E-4)
         If (Week 1 flives <= 0.39285714285714285)
            Predict: 1.0
         Else (Week 1 flives > 0.39285714285714285)
            Predict: 0.0
   Else (Week 4 avgjoinsec > 0.0)
      Predict: 0.0
```

Figure 17. A Decision Tree of Depth 4

Color-coding split nodes by depth and denoting leaf nodes with green, this tree shows the sequences of splits being made. This diagram reveals something very important. Although the model considers every feature at each split, the entire tree in fact only uses three different feature types: playing time, average join time, and fraction of sessions that are live streams. Further examination shows that total playing times are queried along the path to every leaf node where a prediction is made. This finding reinforces almost exactly what was suggested by examining the best features found through feature selection on the normalized features for kNN: both total playing time and average join time are helpful predictors of playing time.

Technical Contributions Conclusion

Through Subscriber Analysis, we attempted to develop a system for accurately predicting behavior in engagement, and in particular, identify viewers likely to churn. In contrast to existing work, we attempted to answer this question using a dataset consisting of only viewing activity and service quality metrics. My contributions towards this goal consisted of exploring the data, identifying potential predictors of churn, defining a method for inferring viewer engagement and churn, and finally implementing two different predictive models: k-Nearest Neighbors and Random Forest. In applying and evaluating these predictive models, I made use of established, well-known techniques and formulae from machine learning and statistics such as multifold cross-validation and the R^2 coefficient of determination.

Our experiences with k-Nearest Neighbors indicated that past playing time is a very helpful predictor of future playing time, our measure of viewer engagement. With this feature alone, kNN achieved an R^2 score greater than 0.5. This translates to performing more than twice as well as predicting the mean in terms of minimizing the sum of squared errors. However, on average, these predictions were off by several hours, over 72% of the average value itself. Thus, while our kNN regression models were an improvement over a naive model, we concluded that they are not accurate. Our

results and observations led us to attribute the source of inaccuracy to the lack of quality features. Beyond past playing time, the most important predictor of future playing time, our use of normalized features suggested that average join time may be a promising feature for future exploration. Using our regression results as a basis for classification, however, we were able to construct a classifier of low-engagement viewers that exhibited fairly high precision. By binarizing viewers based on a configurable threshold value of playing time, we showed our classifier was able to identify a significant proportion of the low-engagement viewers while misclassifying a much smaller proportion of the high-engagement viewers. Our results also showed that precision was best at lower values of the `min_engagement` threshold.

The results from our Random Forest models argue further for the importance of past playing time as a predictor and show that this feature type is the most important of our features in the context of classifying churners. However, they also support the argument that average join time is a good predictor of viewer engagement as well. By partitioning on these two metrics, and to a small extent on the fraction of sessions that target live content, the model was able to obtain about roughly 88% classification accuracy. Although a model that predicts only non-churners would achieve 83% accuracy for our data and definition of churners, our achieved accuracy is a significant improvement on this naive model.

VIII. Conclusion

Capstone Summary

The high level goal of the Online Video Data Analytics project was to develop analysis tools that help extract useful insights from the online video data we have obtained through our partnership with Conviva. In conjunction with Conviva, we have formulated two separate toolsets: Smart Anomaly Detection and Subscriber Analysis. In order to increase focus and productivity along these two products, we divided into two sub teams that worked alongside each other throughout the year. Along the way, we

adopted the perspective of a new business preparing to enter the online video data analytics industry. After researching the industry and market, we developed a competitive market strategy for our prospective business along with a plan for protecting our intellectual property. Together, our findings indicate that our products are valuable and there is substantial potential for establishing ourselves in the market and sustaining a competitive advantage.

As part of the Subscriber Analysis team alongside Yaohui Ye, I was responsible for developing models that predict viewer engagement and identify churners. The specific models we chose to fulfill these tasks were a kNN regression model and a Random Forest model, respectively. In the process of constructing these models, we encountered the intermediate tasks of extracting features and using the data to formulate definitions of both viewer engagement and churn. Together, Yaohui and I accomplished these tasks by acting as domain experts to formulate logical definitions for these concepts, citing assumptions and shortcomings where appropriate. Finally, we performed extensive configuration and tuning of our models. As we did so, we used sound machine learning techniques to avoid suffering from bias, variance, overfitting, and other known machine learning issues to which our specific models are vulnerable.

Our results with kNN regression demonstrate that, while better than a naive model, our currently implemented models are not accurate predictors of weekly playing time, our measure of viewer engagement. These results further indicate that the limiting factor for our models is our lack of good, predictive features. While helpful, past playing time alone is not enough to achieve an accurate kNN regression model. Despite relatively poor performance, our work with kNN suggested two possible helpful predictors: past weekly playing time and average video join time. In addition, we were able to extend our regression model to perform classification of low engagement. This classifier showed high precision and is a promising model to pursue. We believe that with more work, this classifier can provide significant value to, among other possible users, content providers. For example a content provider might seek to identify a list of

viewers it believes are likely to drop in engagement and possibly churn in the near future. By using our high-precision classifier, this content provider may significantly narrow down this list, thereby allowing it to focus its retention resources on these viewers.

By formulating our own definition of churners, we were able to employ a Random Forest classifier and achieve high accuracy on our test set. Our experiments showed that the Random Forest model is able to build trees that accurately detect viewers whom we defined as churners. By inspecting the splits that occur within these trees, we found that the two most promising predictive features suggested by kNN were in fact being heavily used to separate churners from non-churners.

Though the original definition of the Subscriber Analysis problem was vague, through our research and data exploration, our team was able to narrow this definition into the following challenge: develop a predictive model that can accurately predict churn and can be analyzed to understand the causes of churn. Through our work, we have implemented several machine learning models and, in the process, produced valuable findings for our problem space. While three of our originally planned four models—k-Means, kNN Regression, and Logistic Regression—did not perform consistently well enough to be considered accurate, we found that a Random Forest model is very promising. In addition, we discovered that a secondary model that performed classification based on the results from kNN Regression demonstrated similar promise. However, there remains research and development which must be before our models can be viewed as finished products. We now provide a summary of the areas in which future work Subscriber Analysis should focus.

Future Work

The most limiting and critical factor in our work has been the quality of our features. We found no existing work that has tried to perform churn analysis using only viewing activity and service quality metrics. Future work should prioritize exploring

different features. Some possible approaches to this include consulting true professional domain experts, applying a kernel function to a subset of the data, extending the viewer history, preprocessing steps, and tree induction (Verbeke). We note that equally important to the exploration of these features is the exploration of appropriate methods for imputing data when necessary.

While we believe our formulations of viewer engagement and churn are reasonable, we have no formal argument for why they are “correct.” We believe that the application of statistical tests, particularly “goodness of fit” tests such as Chi-Squared or Two-sample KS tests would be appropriate here. Such statistical validation of our definitions would contribute significantly to the validity of our own findings as well as those of future work in churn analysis in general.

We believe that, given the right feature space, kNN as a regression algorithm may be an appropriate model and future work should not discount its potential. One issue we dealt with heavily in our implementation of kNN was numeric stability. For example, overflow issues prevented us from using a weighted average of neighbors’ playing times rather than a simple mean. By weighing closer neighbors more, the kNN model may be more robust to the previously mentioned zero viewer collocation problem.

One issue that Random Forests are vulnerable to, especially in the customer churn setting, is class imbalance (Burez). Class imbalance arises when the vast majority of the data set is dominated by a majority class, leading to a marginalization of the minority class. Class imbalance tends to result in poor predictive performance in the real world. Chen et. al have suggested multiple ways to address this problem, including selective sampling and weighting the votes from the “weak learner” trees.

Finally, given the success shown by our kNN Regression-Classifier and Random Forest model, we believe both of our models warrant future development. For example, future work may compare the performance of kNN with a more complex labeling process, such as the one used in our Random Forest model. In addition, both kNN and Random Forests are easily adapted to be used directly as either regression or

classification. An interesting experiment would compare the performance of these models when used in these settings. Similarly, although we have used our models in isolation, we believe that applying these models in sequence can lead to interesting results. For example, a hypothesis worth testing is that higher regression accuracy can be achieved by applying a regression model to the output of a classification or clustering algorithm.

IX. Acknowledgements

We would like to thank George Necula for advising us throughout the entirety of this Capstone project. We would also like to thank Jibin Zhan from Conviva for introducing the problem space to us and Pat McDonough from Databricks for providing extensive technical support throughout our use of Databricks.

References

Alice Corporation v. CLS Bank. 573 U.S. Supreme Court. 2014. Print.

Altman, Naomi S. "An introduction to kernel and nearest-neighbor nonparametric regression." *The American Statistician* 46.3 (1992): 175-185.

Associated Press. “Netflix reeling from customer losses, site outage.” *MSNBC*. MSNBC. 24 July 2007. Web. 15 Feb. 2015.

Bessen, James. “The patent troll crisis is really a software patent crisis.” *Washington Post*. The Washington Post. 3 Sept. 2013. Web. 27 Feb. 2015.

Biem, Alain E. “Detecting Anomalies in Real-time in Multiple Time Series Data with Automated Thresholding.” International Business Machines Corporation. US Patent 8,924,333. 30 Dec. 2014.

Breiman, Leo. "Random forests." *Machine learning* 45.1 (2001): 5-32.

"Bringing Big Data to the Enterprise." IBM. N.p., n.d. Web. 13 Apr. 2015.

Brundage, Michael L., and Brent Robert Mills. “Detecting Anomalies in Time Series Data”. Amazon Technologies, Inc., assignee. U.S. Patent 8,949,677. 3 Feb. 2015.

Burez, Jonathan, and Dirk Van den Poel. "Handling class imbalance in customer churn prediction." *Expert Systems with Applications* 36.3 (2009): 4626-4636.

Byrd, Owen, and Brian Howard. 2013 Patent Litigation Year in Review. Rep. Menlo Park: Lex Machina, 2014. Print.

CA Inc. “Manage Your Network Infrastructure for Optimal Application Performance.” *CA Technologies*. n.p. n.d. 13 Feb. 2015.

Chandola, Varun, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey." *ACM Computing Surveys (CSUR)* 41.3 (2009): 15.

"Coefficient of Determination." Wikipedia. Wikimedia Foundation, n.d. Web. 23 Apr. 2015.

Cohn, Chuck. “Build vs. Buy: How to Know When You Should Build Custom Software Over Canned Solutions.” *Forbes*. Forbes Magazine, 15 Sep. 2014. Web. 7 Apr. 2015.

Conviva. "About Us." *Conviva*. n.p., n.d. Web. 28 Feb. 2015.

Connelly, J.P., L.V. Lita, M. Bigby, and C. Yang. “Real time audience forecasting.” US Patent App. 20120047005. 23 Feb. 2012.

Cortes, Corinna, Lawrence D. Jackel, and Wan-Ping Chiang. "Limits on learning machine accuracy imposed by data quality." *KDD*. Vol. 95. 1995.

"Covariance." Wikipedia. Wikimedia Foundation, n.d. Web. 21 Apr. 2015.

Cleveland, Robert B., et al. "STL: A seasonal-trend decomposition procedure based on loess." Journal of Official Statistics 6.1 (1990): 3-73.

Dasgupta, Dipankar, and Stephanie Forrest. "Novelty detection in time series data using ideas from immunology." *Proceedings of the international conference on intelligent systems*. 1996.

Deshpande, Amit and Riehle, Dirk. "The total growth of open source." *Open Source Development, Communities and Quality*. Springer US, 2008. 197-209.

"Engineering Statistics Handbook." NIST/SEMATECH E-Handbook of Statistical Methods. NIST, n.d. Web. 14 Mar. 2015.

Etherington, Darrell. "Twitter Acquires Over 900 IBM Patents Following Infringement Claim, Enters Cross-Licensing Agreement." TechCrunch. N.p., 31 Jan. 2014. Web. 25 Feb. 2015.

"F1 Score." Wikipedia. Wikimedia Foundation, n.d. Web. 13 Apr. 2015.

Fisher, William W. “Patent.” *Encyclopaedia Britannica Online*. Encyclopaedia Britannica Inc.

Ganjam, Aditya, et al. "Impact of delivery eco-system variability and diversity on internet video quality." IET Journals 4 (2012): 36-42.

Goldman, Eric. "The Problems With Software Patents (Part 1 of 3)." *Forbes*. Forbes Magazine, 28 Nov. 2012. Web. 01 Mar. 2015.

Gottfriend, Miriam. “Bullish Investors See New Hope for Netflix Profit Stream.” *The Wall Street Journal*. The Wall Street Journal. n.d. Web 14 Feb. 2015.

Guyon, Isabelle, and André Elisseeff. "An introduction to variable and feature selection." *The Journal of Machine Learning Research* 3 (2003): 1157-1182.

Hanley Frank, Blair. "Amazon Web Services Dominates Cloud Survey, but Microsoft Azure Gains Traction - GeekWire." *GeekWire*. Geekwire, 18 Feb. 2015. Web. 02 Mar. 2015.

Hanley, James A., and Barbara J. McNeil. "The meaning and use of the area under a receiver operating characteristic (ROC) curve." Radiology 143.1 (1982): 29-36.

Harvey, Cynthia. "100 Open Source Apps To Replace Everyday Software." *Datamation*. N.p., 21 Jan. 2014. Web. 28 Feb. 2015.

Indyk, Piotr, and Rajeev Motwani. "Approximate nearest neighbors: towards removing the curse of dimensionality." *Proceedings of the thirtieth annual ACM symposium*

on Theory of computing. ACM, 1998.

Iyengar, Vijay S. 2002. “Transforming data to satisfy privacy constraints.” *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining* (KDD '02). ACM, New York, NY, USA, 279-288. Web. 12 Feb. 2015.

"Jarque Bera Test." Jarque Bera Test. NIST, n.d. Web. 15 Mar. 2015.

Jasani, Hiral. “Global Online Video Analytics Market.” *Frost & Sullivan*. n.p. 5 Dec. 2014. Web. 12 Feb. 2015.

Jones, James. “Stats: Coefficient of Determination.” Richland.edu. n.p. n.d. Web. 29 Apr 2015.

Kahn, Sarah. "Business Analytics & Enterprise Software Publishing in the US." IBISWorld (2014): 5. Web. 11 Feb. 2015.

Kandel, Sean, et al. "Enterprise data analysis and visualization: An interview study." Visualization and Computer Graphics, IEEE Transactions on 18.12 (2012): 2917-2926.

Keaveney, Susan M. "Customer switching behavior in service industries: An exploratory study." The Journal of Marketing (1995): 71-82.

Kejariwal, Arun. "Introducing Practical and Robust Anomaly Detection in a Time Series." Twitter Engineering Blog. Web. 15 Feb. 2015.

Kim, Hee-Su, and Choong-Han Yoon. "Determinants of subscriber churn and customer loyalty in the Korean mobile telephony market." *Telecommunications Policy* 28.9 (2004): 751-765.

Kohavi, Ron. "A study of cross-validation and bootstrap for accuracy estimation and model selection." *Ijcai*. Vol. 14. No. 2. 1995.

Lawler, Richard. "Netflix Tops 40 Million Customers Total, More Paid US Subscribers than HBO." *Engadget*. N.p., 21 Oct. 2013. Web. 15 Feb. 2015.

Liu, Xi, et al. "A case for a coordinated internet video control plane." Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication. ACM, 2012.

Marvin, Ginny. “Mobile Video Views Surge 300%; Tablets Fuel Growth And Engagement.” *MarketingLand*. n.p. 04 April 2014. Web. 10 April 2015.

Mcgovern, Gale. Virgin Mobile USA: Pricing for the Very First Time. Case Study. Boston. Harvard Business Publishing, 2003. Print. 9 Jan. 2010.

Metz, Cade. "Facebook's 'Deep Learning' Guru Reveals the Future of AI." Wired.com. Conde Nast Digital, n.d. Web. 16 Mar. 2015.

Moogk, Dobrila Rancic. "Minimum viable product and the importance of experimentation in technology startups." *Technology Innovation Management Review* 2.3 (2012).

“Number of Broadband Connections.” *IBISWorld*. IBISWorld. 3. Web. 12 Feb. 2015.

Numenta. “The Science of Anomaly Detection.” *Numenta*. n.p. n.d. 13 Feb. 2015.

Numenta. “Hierarchical Temporal Memory.” *Numenta*. n.p. n.d. 13 Feb. 2015.

Open Source Initiative. "Welcome to The Open Source Initiative." *Open Source Initiative*. N.p., n.d. Web. 28 Feb. 2015.

"Ordinary Least Squares." Ordinary Least Squares. N.p., n.d. Web. 14 Mar. 2015.

"Partial Autocorrelation." Partial Autocorrelation. N.p., n.d. Web. 23 Apr. 2015.

Plackett, Robin L. "Karl Pearson and the chi-squared test." *International Statistical Review/Revue Internationale de Statistique* (1983): 59-72.

Porter, Michael. "The Five Competitive Forces That Shape Strategy." *Harvard Business Review Case Studies, Articles, Books*. N.p., Jan. 2008. Web. 12 Feb. 2015.

Porter, Michael. "What is Strategy?." *Harvard Business Review Case Studies, Articles, Books*. N.p., Jan. 2008. Web. 12 Feb. 2015.

Quinn, Gene. “A Software Patent Setback: Alice v. CLS Bank.” *IP Watch Dog*. n.p. 9 Jan. 2015. Web. 27 Feb. 2015.

"Receiver Operating Characteristic." Wikipedia. Wikimedia Foundation, n.d. Web. 13 Apr. 2015.

Roettgers, Janko. "Netflix Spends $150 Million on Content Recommendations Every Year." *Gigaom*. N.p., 09 Oct. 2014. Web. 15 Feb. 2015.

Seo, Songwon. A review and comparison of methods for detecting outliers in univariate data sets. Diss. University of Pittsburgh, 2006.

Shelby County v. Holder. 570 U.S. Supreme Court. 2013. Rpt. in Dimensions of Culture 2: Justice. Ed. Jeff Gagnon, Mark Hendrickson, and Michael Parrish. San Diego: University Readers, 2012. 109-112. Print.

Smith, Sarah. "Analysis of the Global Online Video Platforms Market." -- *LONDON, Jan. 5, 2015 /PRNewswire/* --. Reportbuyer, n.d. Web. 02 Mar. 2015.

Stanway, Abe. "Algorithms.py." GitHub. Etsy, n.d. Web. 13 Mar. 2015.

Stanway, Abe. "Analyzer." GitHub. Etsy, n.d. Web. 13 Mar. 2015.

Stanway, Abe. "Skyline." GitHub. Etsy, n.d. Web. 13 Mar. 2015.

Sun, Zehang, George Bebis, and Ronald Miller. "Object detection using feature subset selection." *Pattern recognition* 37.11 (2004): 2165-2176.

Sun Tzu, and James Clavell. *The Art of War*. New York: Delacorte, 1983. Print. 17-18.

Trautman, Erika. “5 Online Video Trends To Look For In 2015.” *Forbes*. Forbes Magazine, 08 Dec. 2014. Web. 14 Feb. 2015.

Tukey, John W. "Exploratory data analysis." *Reading, Ma* 231 (1977): 32.

United States. Cong. Senate. Committee on Commerce, Science, and Transportation. *The Emergence of Online Video : Is It the Future? : Hearing Before the Committee on Commerce, Science, and Transportation*. 112th Cong., 2nd sess. Washington: GPO, 2014. Web. 15 Feb. 2015

Vallis, Owen, Jordan Hochenbaum, and Arun Kejariwal. A Novel for Long-Term Anomaly Detection in the Cloud. Proc. of {6th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 14). Philadelphia: USENIX Association, 2014. Print.

Verbeke, Wouter, et al. "Building comprehensible customer churn prediction models with advanced rule induction techniques." Expert Systems with Applications 38.3 (2011): 2354-2364.

Vinson, Michael, B. Goerlich, M. Loper, M. Martin, and A. Yazdani. “System and method for measuring television audience engagement.” US Patent. 8,904,419. 26 Sep. 2013.

“What Does Copyright Protect? (FAQ) | U.S. Copyright Office.” *What Does Copyright Protect? (FAQ) | U.S. Copyright Office*. N.p., n.d. Web. 01 Mar. 2015.

Wold, Svante, Kim Esbensen, and Paul Geladi. "Principal component analysis."*Chemometrics and intelligent laboratory systems* 2.1 (1987): 37-52.

Worstall, Tom. "The Supreme Court Should Just Abolish Software Patents In Alice v. CLS Bank." *Forbes*. Forbes Magazine, 29 Mar. 2014. Web. 01 Mar. 2015.

Zaharia, Matei, et al. "Spark: cluster computing with working sets." *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*. 2010.

Zeithaml, Valarie A. "Service quality, profitability, and the economic worth of customers: what we know and what we need to learn." Journal of the academy of marketing science 28.1 (2000): 67-85.

Zhang, Ping. "Model selection via multifold cross validation." *The Annals of Statistics* (1993): 299-313.