HEC Montreal

Social Media Textual Analysis for Facebook: Methods for Predicting Engagement, Emotional Response and Identifying Keywords.

by

Yuan Ping Jin

Master of Science (M.Sc.) in

Business Analytics Thesis

April 2018

Abstract

The marketing ecosystem in the B2C world has changed rapidly and social media has become an integral way for companies to reach consumers. Academic research focused on predictive methods in this area have been sparse. With our research, we hope to provide managers and practitioners a guide on how to leverage sparse textual data from the Facebook ecosystem. Currently, many machine learning methodologies rely on manual classification for calibrating training data. This process is slow and the volume of resulting training data is often small due to human resources constraints. We propose a solution to this problem by exploring the use of two training metrics as well as a method for identifying important keywords. The first is derived from platform users feedback metrics unique to Facebook: the number of reactions, comments and shares. The second is the aggregate emotional polarity expressed by platform users through their comments on a status. Strong results are obtained with the SVM (RBF Kernel) algorithm with good results for engagement prediction and sentiment prediction. The implication of this research opens the door to new training paradigms in the field of social media data mining.

Keywords

Facebook; Machine Learning; SVM; Social Media; Proxy Learning; Predictive Model; Sentiment Analysis

Table of Contents

Table of Figures

Table of Tables

Research Overview

The intent of this research is to provide industry practitioners in B2C companies a toolkit and guide to generate concrete models and insights which directly benefit to better decision making for online communication with consumers, more precisely Facebook. The toolkit presents four major methods which contribute to this objective. They are presented in the recommended order of execution for practitioners to follow when undertaking an analysis for their company.

Before the presentation of the methods, we first look at the current available research in terms of text-mining in the context of social media. We then proceed to present a variety of Facebook specific data problems when performing textual analysis and how to address them. We finish with a presentation of standard textual transformations needed to proceed to text-mining.

The first discussed method is the exploratory analysis. Before performing more complex actions and making use of black-box machine learning techniques, it is wise to first leverage simple methods as they can be source of tremendous value. The section discusses examples of statistical analyses and the tools used to perform them. It also presents the application of two unsupervised machine learning techniques for clustering text and discovering inherent themes in a Facebook corpus composed of company statuses.

From this section, practitioners can get a guide on how to perform reporting and visualization analysis on Facebook data. The number of shares per day of the week is an example of possible visualization. Furthermore, a guide on possible interpretation of popular statuses is presented. This allows businesses to get concrete information on the general statistical patterns of their customers. By acting accordingly, companies can increase their general engagement and communicate subject matter that is more demanded by customers.

The second method presents a more complex process and is used to predict the levels of engagement from consumers depending on the textual content of a company's published status. The implementation and development of the method are explained for practitioners to implement their own versions.

From this section, practitioners can create and build their own machine learning models dedicated to predicting the level of engagement of a status before it is posted. Social media managers can benefit directly from the model by submitting the textual content of their next promotional message into the model and get a prediction on whether it will perform well. This allows for a level of assurance when testing out a social media communication strategy.

It is to be noted that the process used to predict engagement is different from a traditional machine learning model creation process. Indeed, a related example would be analyzing product reviews where the learning is done on a training metric that is well defined (stars out of 5). In our case, engagement is not clearly defined. Part of this thesis will explore how to proceed when our desired target training metric is unclear. To remedy this problem, we make use of existing social media metrics available on Facebook (likes, shares, number of comments) and provide a way to translate those into a training metric ready to be used for machine learning.

The third method makes use of the previous method's process of rating engagement and adds a layer of interpretability for decision makers to gain concrete knowledge. We present three ways to extract keywords from company status messages which contribute in a significant way to engagement either in a positive or negative way.

This section allows for practitioners to identify relevant keywords that affect engagement. With knowledge of these keywords, managers can not only calibrate future posting by creating more engaging posts but also get a good picture of current audience preferences.

The last method presents a similar process to predicting engagement used to predict the emotional response by consumers depending on the textual content of a company's published status. As well, the implementation and development of the method are explained for practitioners to implement their own versions.

Like engagement prediction, it is to be noted that the process used to predict emotional response is also different from a traditional machine learning model creation process. In our case, emotional response can be derived from the comments consumers leave on the company's

Facebook page. To arrive at an emotional response, we make use of existing lexicons and provide a way to translate user comments into a training metric ready to be used for machine learning.

Similarly, to predicting engagement, this method allows for practitioners to create and build their own machine learning models dedicated to predicting the emotional response of a status before it is posted. Social media managers can use also as a method to ensure that posts will be well received and use it as a testing mechanism for a social media communication strategy.

We hope these tools provide insight for improving communication and marketing strategies in enterprise settings. In an increasingly connected world, the proper use of data and social media marketing is essential to succeed in a competitive ecosystem.

Literary Review

1. Introduction

It is to be known that the main concern of our research will be focused on social media (Facebook) textual analysis, therefore, we will focus mainly on the academic literature on this matter as well as quantitative techniques which can be applied to this subject. Beforehand, the basics of text mining will be explained and presented to the reader. Thereafter, as an introductory roadmap, we will lay the ground of a typical academic analysis journey regarding textual analysis. By doing so, we will explore in a logical and coherent way, the multiple discoveries, as well as requirements needed to perform a successful analysis on social media data for insight generation and prediction.

The first concern in textual data analysis is being able to obtain the data. Depending on the scope and interest of the research project, there are many ways to obtain textual data for analysis. In Batrinca, B., & Treleaven, P. C.'s (2014) survey of social media analytics, common sources include social media companies with API access, news companies, public governmental data as well as general data scrapping from the web.

The second concern in textual data analysis is regarding data pre-processing, processing and hardware limitations. From Batrinca, B., & Treleaven, P. C. (2014), common issues regarding social media data is its erratic nature. Indeed, misspellings, slang, foreign words and other anomalies can be common. To get the best results out of text-mining, pre-processing must be done to obtain clean data. As well, depending on the size of the dataset, distributed computing needs to be used to process the data. Common software for distributed processing software include Hadoop, Mahout and Cassandra/hive databases (Batrinca, B., & Treleaven, P. C. 2014).

The third concern in textual data analysis is regarding selection of analysis methods. From Batrinca, B., & Treleaven, P. C. (2014), there are a variety of techniques used for social media analysis. Common approaches include using statistical models and machine learning models.

Amongst machine learning models, we can further classify them into supervised and unsupervised learning algorithms.

The fourth concern in textual data analysis is regarding selection of processing tools. From Batrinca, B., & Treleaven, P. C. (2014), common tools can range from languages such as Java, Python, Math-Lab and R to business toolkits like SAS Sentiment Analysis Manager, RapidMiner and Lexanalytics. Furthermore, programming languages will often have ready to use text-mining libraries from academic groups and non-governmental organizations.

We shall analyse the current literature in-depth to find the current techniques and applications regarding each of these concerns for our purposes of social media (Facebook) text-mining.

Overview of Data Mining

Before talking about text mining, it is worthwhile to first take a short look at data-mining. According to Aggarwal, C. C., & Zhai, C. (2012b), data-mining is the idea of taking a set of organized numerical data and then finding useful patterns from it to make predictions or find relationships. In our context, the objective of these predictions or relationships will be to yield commercial insight with the finality of improving profits. As Aggarwal, C. C., & Zhai, C. (2012b) remarks, a specific configuration for the data is needed to apply the different methods and mathematical techniques to get the desired output.

Overview of Text Mining

Text mining is therefore the extension of this idea applied to textual data. Unlike numerical data, textual data is not made of numbers and therefore cannot be easily transformed to be processed by standard data-mining algorithms. According to Andritsos, P., et al., (2004), data-mining utilises often two types of information; the ordered numerical and the categorical. The typical example of ordered numerical information being a measurement such as height, while an example for categorical would be something like gender. The intuition here is therefore to find patterns from the information contained in the different rows. From our example above, without

getting into the specifics of the algorithmic process of data-mining, one could potentially find a certain pattern regarding the height and gender of a person.

Bag of Words Method

The intuition we therefore need to develop is that of transforming text to a structured numerical form. A popular method is the bag of words which decomposes a coherent text with semantic meaning to a series of word counts and indicators derived from word counts (Aggarwal, C. C., & Zhai, C. 2012a). By this method, we can produce a table format ready to be mined. Each set of texts would be a row and each word would become a column with each numerical value representing either the presence (or absence) of a word, it's frequency or a custom weight derived from a specific method (Jiang, J. 2012). In this sense, each distinct word plays a similar role as a variable in a classical data structure. From Aggarwal, C. C., & Zhai, C. (2012a), we learn that the main characteristics of text data is that it is sparse and of high dimensionality, meaning that textual entries have a large variety of words. In other terms, it is unlikely to find a concentration of specific words in a single entry and more likely to see a diversity of different words, each with low frequency.

It is with these words parameters from the text, that patterns can be detected with analysis. Pushing the intuition further, it is important to understand that this data structure of the text presupposes that the computer possesses no inherent semantic knowledge (Aggarwal, C. C., & Zhai, C. 2012a). Indeed, by decomposing a sentence into a series of unordered words takes away the interpretative information that a human could seize from reading it (Aggarwal, C. C., & Zhai, C. 2012a). Therefore, it is important to note that the way insights are derived in text-mining with a bag of words approach is through statistical methods and significant amounts of data, rather than semantic meaning (Aggarwal, C. C., & Zhai, C. 2012a).

This characteristic property of the text creates most of the preprocessing and cleaning challenges that need to be addressed regarding text-mining versus regular data-mining. This inherent noise needs to be minimized so that the mining produces accurate and actionable insights. These methods will be discussed in depth later, but suffice to understand that we wish to make text as

closely resembling to numerical data as possible to apply similar techniques as those used in data-mining.

2. Data Extraction

Data extraction is the first step to any text mining effort as it is necessary to first choose and obtain the necessary raw data for the analysis. In our case, we are primarily concerned with obtaining data sources that would be classified in the category of social media. According to Kaplan, A. M., & Haenlein, M. (2010), social media can be defined as a platform where all users participate collaboratively to the creation of content to be consumed by the same users of the platform. For our purposes, it can be generally accepted that platforms like Facebook, Twitter, Instagram amongst others fit this definition.

Web-Crawlers and Public Sources

There are many ways to obtain data, according to Batrinca, B., & Treleaven, P. C. (2014), we can either employ the use of a web-crawler to directly extract the textual information from the website we are interested in, or get the information from a relevant database/public source such as Tomson Reuters or the Enron e-mail corpus (Cheng, N., et al., 2011). Such sources have been proven to be effective in analysis dealing with applicable subject matters. Indeed, in Cheng, N., et al., (2011), a paper tackling the problem of author gender identification with textual content, used the Reuters Corpus Volume 1 newsgroup data and Enron e-mail data.

The use of a web-crawler however, is a valid strategy as it is applicable to social media web pages. However, the difficulty comes later in the preprocessing phase, indeed, according to Batrinca, B., & Treleaven, P. C. (2014), web data will contain often language markups like XML and HTML consisting in meta-data that needs to be removed. Furthermore, there is also the problem regarding the choice of a crawler and the scripting instructions associated with it. Referring to Batrinca, B., & Treleaven, P. C. (2014), a primer on text-mining for social sciences researchers, other sources of data can include RSS feeds, blogs, commercial sources and social media APIs.

Commercial Sources

Commercial sources could be used if the researcher possess grants necessary for paying for data licenses. Alternatives to commercial sources also exist and are presented in Batrinca, B., & Treleaven, P. C. (2014). These data sources are in the form of "data grants", meaning an individual arrangement between a company ready to offer data for research purposes under certain conditions to the academic. Some of these programs are publicised such as in the case of Twitter (Batrinca, B., & Treleaven, P. C. 2014), however is it worth mentioning that future academics might even find opportunities by directly reaching out to companies which do not offer such public offerings. If there is a collaborative effort as well as a communication channel between the researcher and the company, it would be the preferred method of obtaining data.

APIs

In other cases, sourcing data from an API is likely the best strategy to be employed, as most popular social media platforms put API at the disposal of users and can result in an output that is more organised and require less metadata wrangling later due to the structural friendliness of data extract formats like JSON (Batrinca, B., & Treleaven, P. C. 2014). As an example, all three mentioned social media platforms earlier have APIs that allow the extraction of public data to varying degrees of freedom.

In Agrawal, H., & Kaushal, R. (2016), researchers could gather sufficient data from the Facebook API Graph v2.5 to conduct analysis and construct machine learning models to detect spam and off-topic messages on public pages. Another analysis done on Twitter by Ghiassi, M., et al., (2013) successfully extracted 300,000 Twitter records with the v1.0 API, although with some slight difficulties regarding the continuity of data because of API defined restrictions on the number of "tweets" able to be extracted per API call. The use of API can even extend to third parties servicing websites. An example is Spinn3r which acts as an API for individual blogs (Corley, C., et al., 2010). In a paper by Corley, C., et al., (2010), a strong correlation between the mentions of flu and influenza illness on online communities and real life incidence is shown, for this research the Spinn3r API was successfully used for data extraction.

Conclusion

We can see that there are various options available for researchers to get the desired data. Depending on the analysis and the subject matter, one type of tool could be preferable to the other. For our purposes of social media text-mining, we can conclude that extracting data from an API is the preferable way to proceed as it directly leverages the tools offered by the company. Indeed, unlike scrapping, API sourced data can be much cleaner, easier to extract and in line with the terms of service of a website. Of course, it would also be preferable if data can be obtained directly by way of an arrangement with a company.

3. Preprocessing

Preprocessing of the data constitutes an essential part of the text-mining process and is often determinant on the quality of the analysis. Indeed, according to Batrinca, B., & Treleaven, P. C. (2014), missing data, incorrect data and inconsistent data can represent major aspects affecting the mining results. Depending on the raw format of our data, processing needs to be done first to remove the associated tags. This is common when dealing with markup languages found from information gathered from the web. Batrinca, B., & Treleaven, P. C. (2014), refers to the common formats such as XML and JSON which are usual data structures used in API extracts. These markups can typically be processed into standard, table form, data structure with a multitude of programming languages. Indeed, JSON itself stands for JavaScript Object Notation and is extended from JavaScript (Batrinca, B., & Treleaven, P. C. 2014). Such processing is common and an example can be seen in Cheng, N., et al., (2011), which made use of the Reuters Corpus Volume 1 which was originally in XML formatting. The data had to be formatted to remove all tags as well as meta data such as data of publication.

Stopwords

Once data is obtained in a purely textual format, we can proceed to apply many useful transformations to the textual content. We would typically start with a series for alterations on the text with the goal of removing noise and standardizing the content. Kahya-Özyirmidokuz, E.

(2014) presents a typical series of such transformations commonly called text normalisation that are effective on social media data. Among these transformations, we take away words that have low significance and meaning. Akaichi, J., et al., (2013) refers to those words as *stopwords* and is a token reduction technique. *Stopwords* are defined (Akaichi, J., et al., 2013) as words with low prediction abilities, typically words like articles and pronouns in the English language (the, a, it, etc.). Another group like *stopwords* are characters which do not have semantic significance, Kahya-Özyirmidokuz, E. (2014) mentions white spaces, punctuation and special characters as part of this group. Many programming languages have available libraries to complete such tasks, the *TM* package on R is an example.

Stemming

Singh, J., & Gupta, V. (2016) define stemming as reducing words to their linguistic stem. The heuristic here is that in the English language, most words have core stems which carry similar meaning for words in that stem family (Singh, J., & Gupta, V. 2016). Therefore, we can effectively create associations for all words in the same stem family by reducing them to their common core. This allows not only for increased significance on the stem family but also feature reduction due to combining multiple groups into one. A popular stemmer to achieve these transformations is the Porter stemmer which is readily available online in multiple programming language implementations.

The Porter stemmer works with a rule based system and an understanding of the linguistic processes of the English language. It is mainly based on the removal of common suffixes to treat words from the same linguistic family as the same group for information retrieval (Porter, M. 1980). Rare suffixes and root words like index/indices are ignored to avoid performance degradation whereas common suffix endings like –ed, -ing, -ion, etc. are treated to get the root of the word (Porter, M. 1980). The algorithm can be described as working in 2 phases. First, endings pertaining to plurals and past participles are treated, then common suffixes are removed (Porter, M. 1980). All of this is done with specific rules and by keeping in mind that suffixes are removed only if the length of a word is sufficient so that the root retains enough meaning (Porter, M. 1980).

Stemming has been deemed to have a moderate positive effect on overall performance (Singh, J., & Gupta, V. 2016) and many research papers have used stemming to various degree of success. (Kahya-Özyirmidokuz, E.'s (2014) paper focused on clustering Facebook pages from similar companies. In this paper, stemming was amongst the tools which helped in enhancing the effectiveness of K-means clustering. Ghiassi, M., et al.,'s (2013) paper focused on evaluating the emotional perception of brands by Twitter users made use of stemming. From a corpus of Tweets referring to the *Justin Bieber* brand, the researchers defined a small and significant lexicon of 187 terms related to sentiment (Ghiassi, M., et al., 2013). With this approach, they could determine non-neutral sentiments on 90% of a corpus of the *Justin Bieber* brand while previous comparable studies only had a non-neutral sentiment on 20% of the corpus (Ghiassi, M., et al., 2013).

Dimensionality Reduction

In the same objective of reducing dimensionality and increasing the predictive ability of features, other techniques exist to filter out words that are either too frequent or too rare. Pang, B., et al., (2002) proposes to simply use the frequency of apparition of words to determine a threshold to reduce features. The justification here is that words that are not very frequent or too frequent will not have a good ability to differentiate between different types of textual interventions. Therefore, it is good to eliminate them to save both on computing costs as well as reducing possible noise.

Weighting

Aside from simply looking at word frequencies, there exist more sophisticated measures of significance and representation. These measures seek to represent with more accuracy the importance of a word in a corpus. The most common of these measures is the TF-IDF, meaning term frequency and inverse document frequency (Kobayashi, M., & Aono, M. 2004). The TF-IDF formula goes as follows:

$$tf - idf(j) = tf(j) * idf(j)$$

$$idf(j) = \log(\frac{N}{df(j)})$$

The j parameter in the function represents a word in the corpus and N represents the total number of documents. Df(j) is the document frequency of the word j, meaning the number of documents in which j appears. Tf(j) is the term frequency of the word j, meaning the number of times the word j appears in the corpus. Idf(j) is the inverse term frequency of the word j, meaning the inverse function of the number of times the word j appears in the corpus.

The inverse document frequency is a measure of weight on how significant a word is in the corpus (Kobayashi, M., & Aono, M. 2004). According to Kobayashi, M., & Aono, M. (2004), a word is very significant if it appears in few documents, this is so because it has a high discriminating power. If it appears in a lot of documents, then it has low discriminating power. This weight is then multiplied with the term frequency of the word to create a weighting that favours words which are in few documents but are present often in documents in which they appear (Kobayashi, M., & Aono, M. 2004).

Research using techniques looking at frequency or using TF-IDF metrics for dictionary reduction as well as defining feature importance have seen good successes. For example, Kahya-Özyirmidokuz, E. (2014) successfully used TF-IDF as part of identifying significant terms needed for clustering different Facebook pages. In another study by Ghiassi, M., et al., (2013), twitter data is used to develop a sentiment analysis model. Traditionally a twitter corpus contains over 10,000 features, with the frequency approach and the TF-IDF approach, as well as a few other techniques, the authors could narrow down the number of significant features to 187 and create an appropriate sentiment lexicon.

4. Analysis Methods: Lexicon vs Machine Learning

We detail here sentiment analysis in a more detailed manner and share the results of selected previous analysis done in the field of text-mining more precisely for applications for social media and related to it. According to Pozzi, F. A., et al., (2017b), sentiment analysis is "[a

process] to define automatic tools able to extract subjective information to create structured and actionable knowledge".

Technically speaking, when performing sentiment analysis, we want to classify subjective sentences by either positive, negative or neutral (Pozzi, F. A., et al., 2017b). Emotional opinions will have targets and to output a successful interpretation of the textual data, we need to first identify the target of an opinion and then associate the right emotional indicators to them (Aggarwal, C. C., & Zhai, C. 2012a). In the case of sentiment analysis of product reviews, this task is often skipped as we can assume with a reasonable degree of confidence that a review is the opinion of the person writing it and the discussed subject is the subject matter being reviewed (Aggarwal, C. C., & Zhai, C. 2012a).

Thus, for documents sentiment classification, two methods are popular, the classification approach based on supervised learning and the lexicon dictionary approach. The classification approach uses algorithms such as SVM, naïve Bayes and others to made a classification on sentiment outcome. These require manually classified training data and is quite like data mining approaches which we will discuss in detail in the following sections. The lexicon approach uses a thesaurus, and leverages the known sentiment of certain words (Aggarwal, C. C., & Zhai, C. 2012a). For example, "bad" and its synonyms are generally negative. The fallback here is that usage of words with non-general sentiment orientation may cause the analysis to be wrong (Aggarwal, C. C., & Zhai, C. 2012a). For example, blue can describe the objective color of a shirt or an emotional response. In that regard, it would be better to define a custom lexicon for specific applications.

In Mostafa, M. M. (2013), a study about consumer brand sentiments on twitter messages, researchers used an expert defined lexicon with 6800 seed adjectives to perform their analysis. By looking at the aggregate tweets by company, researchers could determine a mean sentiment score for 16 companies reflecting an overall consumer impression of these companies. The resulting findings can be used as a benchmark for companies going forward in their social media communication with customers.

5. Machine Learning Algorithms

5.1 Unsupervised Learning

5.1.1 K-Means

The K-Means method is an unsupervised learning algorithm with the goal to group n data points into K clusters (Kanungo, T., et al., 2002). First, K random starting points are chosen in the dataset, these points represent the initial cluster centers. Next, points are added to the cluster centers such that the distance between each point and the cluster center is minimized. The center of a cluster can change after new points are added to minimize the distance of all points to the cluster center (Kanungo, T., et al., 2002). Once no improvements are possible, the algorithm stops. It is also to be noticed that since the starting points are random, different results can arise depending on the starting seed (Kanungo, T., et al., 2002).

Applied to text-mining, we must consider notions of distances and similarity between two document from the bag of words vectors that compose each document. As discussed earlier, the bag of words vector can be based on an TF-IDF measure or a simple frequency measure (Kobayashi, M., & Aono, M. 2004). From there, there are many possible ways of calculating distance such the commonly known Euclidian distance (Kobayashi, M., & Aono, M. 2004).

Using K-means, Kahya-Özyirmidokuz, E. (2014) conducted a study mapping the similarity of 200 Turkish companies' Facebook social network data by clustering them into distinct categories. With the use of cosine distance and a K-means clustering algorithms, researchers separated firms into two distinct categories therefore providing additional insights to practitioners regarding the possible application of K-means text-mining in the context of social media analysis. Furthermore, Irfan, R., et al.,'s (2015) survey on text mining in social networks suggests K-means as a good way to approach problems that require the identification of specific clusters from textual data. Finally, Wu, H., et al.,'s (2014) research focused on analysing local restaurants on Facebook, was able to leverage the K-means algorithm with Euclidian distance and apply it in the context of Facebook customer data for restaurants. With the findings, the researchers mapped out specific characteristics with restaurants. For example, the best tofu

restaurant and the cheapest restaurant from a customer feedback point of view could be easily extracted from this exercise.

5.1.2 Hierarchical Clustering

As K-means, hierarchical clustering is also an unsupervised learning technique producing groupings (Murtagh, F. 1983). It's differentiating factor is its ability to output a grouping with a hierarchy, in the case of text-mining this can often result in finding an underlying taxonomy in the data (Murtagh, F. 1983). All possible pairs of documents are compared to each other and similarity measures are computed, then a hierarchy is defined by grouping the most similar documents together (Murtagh, F. 1983). Typically, a bottom up process is done by pairing up similar documents, then by grouping those documents together, in a recursive fashion until the entire corpus constitutes a single group (Murtagh, F. 1983).

5.2 Supervised learning

5.2.1 Naïve Bayes

The Naïve Bayes algorithm takes its name from Bayes' theorem which states that the prior probability of a state alone can be indicative of the posterior probability of a state (Zhang, H., & Li, D. 2007). It can be represented in mathematical terms by Bayes' formula:

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)}$$

The algorithm naïvely assumes that all used predictors for the likelihood of a state are independent of each other (Zhang, H., & Li, D. 2007). In many studies, Naïve Bayes has been used as a benchmarking method for machine learning prediction because of its simplicity of implementation as well as speed (Weiss, S. M. 2005). In the context of text-mining, we apply the same logic to a vector of words representing documents. As an example, given the presence of specific words, meaning the prior state, the posterior probability of a document to belong to a certain class is calculated.

Researchers in Sun, J., et al., (2015) proposed Naïve Bayes as one of the algorithms in their ensemble method to improving a recommendation engine for TED talks by analysing textual content from comments and reviews. The ensemble model combined a bag-of-words and lexicon approach to suggest new TED talks to users based on textual information left on the previous ones that they had watched.

5.2.2 Max Entropy

Max entropy, also known as multinomial logistic regression (Jurka, T. 2012) is simply an extension of the regular logistical model used to predict cases where we need to account for many classes (Greene, W. H. 2012).

Logistical regression is a technique that is used when the outcome we want to predict is a binary class. Its usefulness comes from the ability to translate independent explanatory variables into a probability to belong to either 0 or 1 of a binary class (Kleinbaum, D. G., & Klein, M. 2010). This transformation is done with the sigmoid function to characterise this probability between 0 and 1 (Kleinbaum, D. G., & Klein, M. 2010).

5.2.3 Decision Tree

A decision tree creates a tree-like structure, starting with a single group and splitting it into smaller groups recursively from characteristics of the members of the group until a stopping condition is met (Apté, C., & Weiss, S. 1997). As an example, we can think of a group of people who are either retired or working. From this information, we would like to build a decision tree that can answer whether a person is under 65 years old. We would split our group of people by their retirement status as it is the characteristic that we are given.

One stopping conditions mentioned by Apté, C., & Weiss, S. (1997) is ending the algorithm after a predetermined number of iterations. For example, we would stop automatically after separating our initial group after x times, where x is decided by the person operating the model. Another stopping condition (Apté, C., & Weiss, S. 1997) could be that all member of the group share the

same characteristics and it is therefore no longer possible to further split the group. For example, if we separate our group into retired people and working people, we cannot further split the two resulting groups as we possess no further differentiating characteristic.

Lastly, we can set an information gain threshold as our stopping condition (Apté, C., & Weiss, S. 1997). Information gain is calculated with the concept of entropy which can be characterised as the degree of impurity of a group. Impurity can be described as the diversity of a group relative to a characteristic (Apté, C., & Weiss, S. 1997). For example, if a group only contains retired people then that group is very pure, if a group contains half retired, half working people then that group is very impure. When creating branch separations, we want to take a group and split it into sub-groups which have low degrees of impurity, meaning minimizing entropy (Apté, C., & Weiss, S. 1997).

As an example, starting with a group of people of 10 people, with 5 over 65 years old and 5 under, if we define two subgroups based on whether a person is retired or not, we could get hypothetically two groups, composed of 4 retired people and 1 non-retired person, as well as 4 non-retired people and 1 retired person. The resulting groups have therefore lower entropy since there is more people of one class, meaning the set is purer (Apté, C., & Weiss, S. 1997). In the context of text-mining, text features are utilized as separators to increase pureness in sub-sequent partitions, therefore the functioning remains essentially the same (Aggarwal, C. C., & Zhai, C. 2012a).

In Irfan, R., et al.,'s (2015) survey of text-mining on social networks, it is shown from a variety of existing studies that decision tree in the context hybrid models can reach good performance and can serve as a viable compliment to artificial neural networks and support vector machines. Sun, J., et al., (2015), a study on the implementation of a recommendation engine based on Ted Talk comments also made successful usage on decision trees. Furthermore, Cheng, N., et al.,'s (2011) study focused on author gender identification from text made use of the boosting variant of decision trees, a concept we will discuss in the following section, and obtained results with up to 74% of accuracy when classifying a text to the gender of its author. The study made use of the

Reuters and Enron email corpus and exploited features were character-based, word-based as well as meta-data based (such as the number of paragraphs) (Cheng, N., et al., 2011).

5.2.4 Random Forests

Random forests represent a form of ensemble learning that also uses multiple decision trees as a method to improve overall performance (Breiman, L. 2001). Random forests are created from a series of decision trees with the major difference being, a random factor selection at each tree split (Breiman, L. 2001). For example, when building a tree, we randomly select a subset of variables from our overall pool to consider when going through the process of choosing on which variables to perform the branching split for the most information gain.

5.2.5 Tree Bagging

Bagging is a special case of random forests. Like random forests, bagging is defined by (Aggarwal, C. C., & Zhai, C. 2012a) as being part of a family of classification techniques called Meta-Algorithms. This family of algorithms have the characteristic of leveraging an ensemble of models and then combining them with a voting method (Aggarwal, C. C., & Zhai, C. 2012a). An example of a voting method for a case involving binary classification is simply taking the classification result of all the models and going with the popular vote (Aggarwal, C. C., & Zhai, C. 2012a). The principle of bagging is simply the idea of training multiple models from multiple samples drawn randomly with replacement from our data set (Aggarwal, C. C., & Zhai, C. 2012a). This random sampling process is what enables the creation of different decision trees. These models are then submitted to a voting method and we are given one overall result.

5.2.6 Logistic Boosting

Logistic boosting is a variant of boosting used for binary outcomes (Friedman, J., et al., 2000). Like bagging, this method is often used in conjunction with classification trees matched with a voting method. Boosting can be described as an iterative process that assigns weights to training data (Aggarwal, C. C., & Zhai, C. 2012a). Weighting methods are various, but the intuition is to adjust the weighting in such a way to favor the learning of data which is erroneously classified

(Aggarwal, C. C., & Zhai, C. 2012a). Weights are readjusted after each iteration and the learning continues.

5.2.7 Support Vector Machines

Support Vector Machines is a method based on separating a set of data using "support vectors". The optimization is done by maximizing the space between the support vectors. In that sense, the data points that are closest to the support vector play a great role in separating the dataset (Joachims, T. 1998). These points represent intuitively the data that is at the fringe of their respective category. This separation is done in a linear way, if a dataset requires non-linear separation, a kernel trick can be employed. A kernel-trick is the process of increasing dimensionality of data to obtain a distribution that is linearly separable in the hyperplane (Joachims, T. 1998). The intuition behind this method is that by finding the largest linear empty region, we will allow for the best separation for the fit of the two classes of data. For text mining, a TF-IDF or frequency matrix with a kernel transformation can be used to perform essentially the same type of SVM as used for regular data-mining.

SVM in the context of text-mining have shown to produce many good results. In a study by Akaichi, J., et al., (2013) with the goal of extracting useful information on social media relating to the "Arabic Spring" era, sentiment analysis was done on comments coming from Facebook. By proceeding with unigrams only, SVM achieved a 72% accuracy in classifying sentiment of user comments. Irfan, R., et al.,'s (2015) survey on text-mining also shows that SVM performs very well and is a strong addition to multiple hybrid approaches especially with neural networks. The previously referred study (Ghiassi, M., et al., 2013), was also able to leverage SVM as a viable method which produced moderate results with 64% accuracy in determining the sentiment of a Tweet and served as benchmark for the neural network model used by the researchers. In Sun, J., et al.,'s (2015) TED talk recommendation system, SVM was the best performing algorithm in the ensemble method. Finally, in Cheng, N., et al.,'s (2011) study on author gender identification also had success using SVM. Indeed, for their application case and both corpuses of Reuters and Enron, SVM was shown to perform better than boosting decision trees and logistic regressions.

5.2.8 Artificial Neural Networks

A neural network is an algorithm inspired by the biological process of the brain (Sebastiani, F. 2002). A neural network is composed of an input layer, a hidden layer and an output layer. Each layer is composed of a series of neurons which are connected in a bi-partite fashion with the next layer. Each neuron has inputs which it sums up using a specific function, the sum is then adjusted for bias and its output is processed by an activation function as input for the next connected neuron (Sebastiani, F. 2002). The resulting network thus produces a classification. Training is done by minimizing the prediction error at the output layer. This error is gradually spread out to previous neurons in a process called backpropagation (Sebastiani, F. 2002). The application in text mining can be understood with starting out with a single neuron with a linear weighting function working on a binary classification from several features from the text (Aggarwal, C. C., & Zhai, C. 2012a). From there, we can perform a classification on a dataset that is linearly separable. By adding further neurons, we can shape our separation space further into non-linear regions and therefore provide potentially performant classifications (Aggarwal, C. C., & Zhai, C. 2012a).

We have seen the use of ANNs in applications related to text-mining. In Irfan, R., et al.,'s (2015) survey of on text-mining, several hybrid models from different studies making use of ANNs have shown good performance. Another example of application can be found in Ghiassi, M., et al.,'s (2013) research on twitter brand sentiment analysis. By using ANN, the researchers could obtain better results than SVM on the classification of sentiment on tweets. Indeed, the category with the lowest accuracy (Strongly positive) had a reasonable performance of 67% of accuracy for classifying the sentiment of a tweet (Ghiassi, M., et al., 2013).

5.2.9 Generalized Linear Models

General Linear Models are a well-known statistical tool. They are an extension of the linear regression model to accommodate a range of independent variables as model inputs (Pozzi, F. A., et al., 2017a). They are composed of predictive variables, a random component representing natural noise and a function linking the predictive variables to a result (Pozzi, F. A., et al.,

2017a). The underlying optimization is performed on minimizing the errors with the least squares fitting method (Pozzi, F. A., et al., 2017a). For a text-mining situation, predictive variables would typically be word features.

5.2.10 Linear Discriminant Analysis

LDA finds the linear projection for the data which maximizes the discriminating power between two classes (Torkkola, K. 2011). It uses the averages of the individual classes and their covariance to set a score for each factor. Once the score is set, we simply perform scoring on each data point and find out its associated class (Torkkola, K. 2011). For text-mining a variant called linear scoring can be employed where each word is given a score representative of their discriminating quality (Weiss, S. M. 2005).

Data Treatment

1. Choice of Companies

Five companies are chosen to be part of this thesis. The reason behind choosing five companies, is to have a diverse number of analysis cases, in which we could possibly detect variability in the results and performance of different analysis methods as well as make some form of generalisation when dealing with B2C Facebook pages. More companies would be preferable; however, computational resources are limited and each company must go through multiple model tuning processes.

The chosen companies with industry sector market share are:

- The Hudson Bay [9% CAN] (Cohen, A. 2017b)
- La Senza [Part of L Brands Holdings 39.4% US] (Cohen, A. 2017c)
- Best Buy [43.6% US] (Guattery, M. 2017)
- Chapters Indigo [60% CAN] (Cohen, A. 2017a)
- Lululemon [8.7 Billion Market Cap, Fragmented industry 86.7% made from non-major players] (Hurley, M. 2017)

Statuses and associated comments were extracted through the Facebook API for the following periods:

- The Hudson Bay: First Post to 2017-11-16
- La Senza: First Post to 2017-11-12
- Best Buy: First Post to 2017-11-16
- Chapters Indigo: First Post to 2017-11-16
- Lululemon: First Post to 2017-11-19

The specific choice for these companies is made according to market share in their specific industries. Indeed, they serve as proxies of their activity sectors. The Hudson Bay represents departments stores, La Senza represents lingerie and swimwear stores, Best Buy represents

consumer electronic stores, Lululemon represents women clothing stores and Chapter Indigo represents book stores.

All five companies are client facing, as this represents an important factor in social media analysis. Indeed, the very nature of social media is one of interactivity between people, therefore B2B businesses were not considered because of the generality that they have a lower number of total interactions with end customers.

Furthermore, all five companies represent relatively distinct business offerings. This has the potential of finding differences in the performance of various analysis methods regarding the respective business categories. Lastly, the amount of Facebook data also played a factor in choosing these companies. The intent being, to have companies without too much Facebook data like Amazon as those could prove to be difficult to analyze computationally with the resources on hand. Furthermore, we also want to have companies that are not too small, which would have insufficient data for the mining process.

2. Raw Data Extraction

The analysis of social media text will be limited to Facebook. The main reasons being accessibility and richness of the content and the “like” button feature. Twitter and Instagram were also considered to be part the analysis. The reason for choosing exclusively Facebook is because there is low potential for cross-referencing between platforms. We would therefore end up with separate analyses for each platform. Furthermore, Twitter’s inherent data organisation is quite different to that of Facebook, meaning it does not have a structured comment section for each posting feedback but rather works as a network graph with each tweet acting as a node. Instagram is mostly focused on images rather than text and would hence require the usage of techniques in computer imaging which would go beyond the scope of this thesis. Therefore, Facebook is the best suited choice for text-mining.

Facebook has an API allowing users to extract information from public pages such as those of companies. Its capabilities go beyond extracting comments and statuses from pages, according to

the Facebook Graph API reference page, other information such as image and video metadata can also be extracted.

For our purposes, the needed information fields related to **statuses** and **comments** are presented with their descriptions in the two below tables.

Table 1: Facebook Status Field Description

Field	Description
status_id	Unique identifier for the status.
status_message	Message written in the status.
status_published	Publication time of the status in the format YYYY-MM-DD HH:MM:SS.
num_reactions	Number of reactions. Reactions are the sum of likes, loves, wows, hahas, sads and angrys on the status.
num_comments	Number of comments on the status.
num_shares	Number of times the status has been shared.
status_type	Format of the status. The possibilities are event, link, photo, status, video.

Table 2: Facebook Comment Field Description

Field	Description
comment_id	Unique identifier for the comment.
status_id	Unique identifier of the status, on which the comment is posted.
parent_id	Unique identifier of the parent comment. The parent comment is a comment which is being replied to, therefore only applicable to comments which reply to another comment.
comment_message	Message written in the comment.
comment_author	Author of comment in the form: FIRST NAME LAST NAME
comment_published	Publication time of the comment in the format YYYY-MM-DD HH:MM:SS.
comment_like	Number of likes on the comment.

Placing API calls can be done in many ways including PHP, JavaScript, Android SDK, etc. Max Woolf's Github page offers an off-the-shelf python script allowing us to extract the necessary information described above in a CSV UTF-8 encoded format. To connect to the Facebook API, a Facebook Developer account is needed as well as an secret app token which can be obtained by creating an application in the user's Facebook developer account.

3. Preprocessing

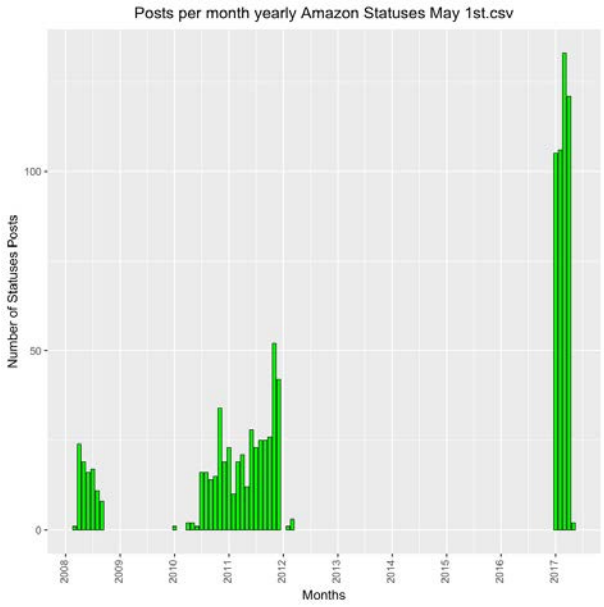
When calling the Facebook API for information extraction, many inconveniences and inconsistencies can arise. As a test, multiple companies' pages are scrapped to test the

consistency of the API and find potential errors that need to be addressed. We only address in this section issues which we have encountered, however this is not an exhaustive list and future issues might arise. We therefore advise academics to proceed with caution when working with the Facebook API.

Facebook API Issue 1: Missing information

The first issue is that of data loss, on a API call for statuses on Amazon's Facebook page on May 1st 2017, 381KB of raw data was obtained. An API call is done again on June 5th 2017, the raw data obtained was 2.7MB. This shows a clear discrepancy and missing data on the first API call. Significant data loss such as this one can be detected with a visual representation of the statuses. In the following case, we use the R language with ggplot2 library to obtain a simple monthly histogram view of the number of statuses vs time of posting.

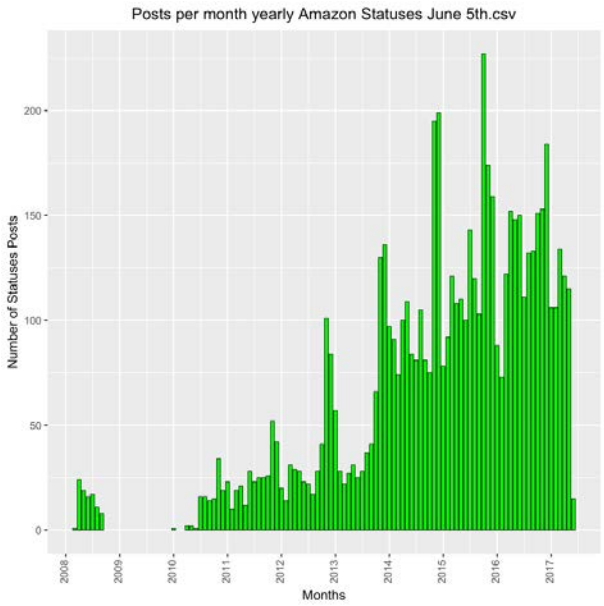
Figure 1: Example of Monthly Histogram View



From this first graph it can be observed that there are no posts from the start of 2012 until 2017. This is no doubt a suspicious result. Upon a simple manual verification on the Amazon Facebook page, it can be concluded that data is missing in this extract. Regarding the earlier missing data,

upon investigation, the empty period between September 2008 and January 2010 is found to be the real situation as reflected on the Amazon Facebook Page.

Figure 2: Example of Monthly Histogram View (2)



The graph of the second extract shows a more representative picture as there are status posts all throughout the timeline (except September 2008 and January 2010). Unfortunately, it is to be noted that there is no way to verify the complete integrity of our API extract other than through a full manual verification by checking every status post. This process would be too time consuming and defeat the purpose of using the API extraction method. It is therefore assumed from this point on that there is an acceptable degree of missing values in our data compared to the real data. This is one of the shortfalls of the Facebook Graph API.

We can conclude that the behavior of the API regarding the completeness of the data is erratic. The two Amazon extracts are done about a month apart. However, this test does not represent a guarantee for other pages. Indeed, it is possible to never get a full extract for a company, hence the companies that can be analysed are limited to the ones whom full extracts can be obtained conditional to the good behavior of the API.

Facebook API Issue 2: Special Timestamps

The second issue is special timestamps. This is not a bug, but rather a property of the Facebook platform. Special timestamps refer to the field “status_published” which can contain timestamps on some posts that would seem unrealistic or the result of a bug. This is better illustrated with an example. When looking at Best Buy’s statuses, some are dated before the inception of Facebook which in theory would be impossible. However, upon closer inspection, those statuses are in fact not the result of a bug, but a feature in Facebook that is used to describe some event in a company’s history.

Table 3: Examples of Special Timestamp Statuses from Best Buy

status_published	**status_message**
1966-01-01 3:00	Founded in 1966
1983-01-01 3:00	Sound of Music is renamed Best Buy

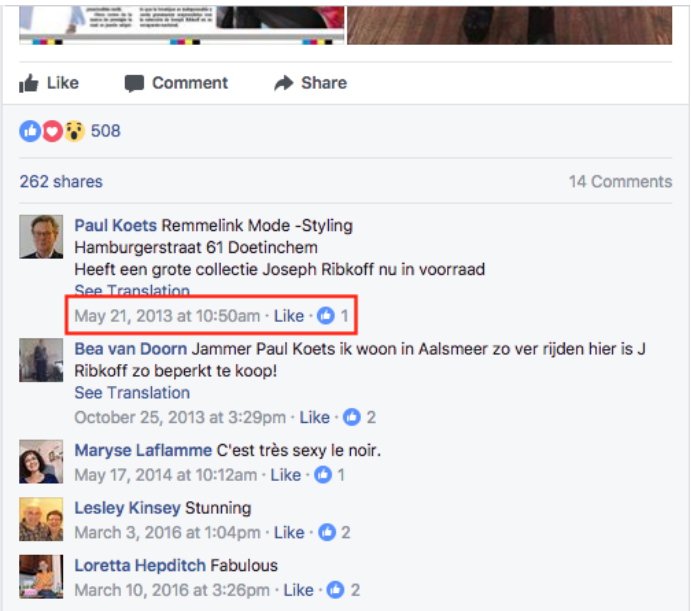
Referring to the above table, an example of two statuses which have publication dates preceding the inception of Facebook can be seen. Upon closer inspection of the message content, these refer to historical events in the company’s development rather than actual timestamps for statuses. In the prepossessing phase, such status entries will have to be treated as they are unlike regular statuses.

Facebook API Issue 3: Comment Duplication API bug

While doing manual verifications, several inconsistencies are noticed regarding some public pages. The Joseph Ribkoff page is taken as an example.

Figure 3: Screenshot of Joseph Ribkoff Status: "Recent media coverage of our styles"

Figure 4: Screenshot of Joseph Ribkoff Status: "Recent media coverage of our styles" Comment Section



The first issue here is regarding comment association. Upon first look, the comment section of this status might seem normal, however the date ordinance does not work. Examining the date bordered in red in picture 1, the date of publication is 10 March 2016. Examining the date bordered in red in picture 2, the date of publication of the first comment is 21 May 2013. This is

impossible as a comment on a status cannot be physically posted before the publication of said status. Therefore, there are wrongly assigned comments on Facebook public pages, upon inspection of the csv generated for this page by the Graph API, it can be observed that this same dissonance is present in the csv and this finding is not merely visual.

Table 4: Extract of Statuses of the Type "Recent media coverage of our styles".

status_id	status_message	status_published
61908299206_10154420817714207	Recent media coverage of our styles	2016-08-05 8:06
61908299206_10154378496979207	Recent media coverage of our styles	2016-07-21 9:59
61908299206_10154375745029207	Recent media coverage of our styles	2016-07-20 10:55
61908299206_10154330977104207	Recent media coverage of our styles	2016-07-04 15:08
61908299206_10154282164104207	Recent media coverage of our styles	2016-06-16 12:02
61908299206_10154255865929207	Recent media coverage of our styles	2016-06-06 16:08
61908299206_10154242122969207	Recent media coverage of our styles	2016-06-01 15:43
61908299206_10154227198964207	Recent media coverage of our styles	2016-05-26 16:03
61908299206_10154191381814207	Recent media coverage of our styles	2016-05-12 10:15
61908299206_10154177871244207	Recent media coverage of our styles	2016-05-06 15:58
61908299206_10154158256754207	Recent media coverage of our styles	2016-04-28 12:35
61908299206_10154134716879207	Recent media coverage of our styles	2016-04-18 15:37
61908299206_10154122550944207	Recent media coverage of our styles	2016-04-13 12:42
61908299206_10154117664194207	Recent media coverage of our styles	2016-04-11 15:34
61908299206_10154052404054207	Recent media coverage of our styles	2016-03-23 14:51
61908299206_10154048139284207	Recent media coverage of our styles	2016-03-22 14:47
61908299206_10154043649514207	Recent media coverage of our styles	2016-03-21 13:50
61908299206_10154024285639207	Recent media coverage of our styles	2016-03-15 13:32
61908299206_10154009637609207	Recent media coverage of our styles	2016-03-11 12:55
61908299206_10154004905974207	Recent media coverage of our styles	2016-03-10 14:27
61908299206_10153986375059207	Recent media coverage of our styles	2016-03-03 12:55
61908299206_10153967816374207	Recent media coverage of our styles	2016-02-24 10:53
61908299206_10153950252969207	Recent media coverage of our styles	2016-02-16 16:43
61908299206_10153938792089207	Recent media coverage of our styles	2016-02-11 16:13
61908299206_10153932060494207	Recent media coverage of our styles	2016-02-08 12:09
61908299206_10152437825464207	Recent media coverage of our styles	2014-05-14 9:20
61908299206_10151965392909207	Recent media coverage of our styles	2013-10-24 10:41
61908299206_10151620050509207	Recent media coverage of our styles	2013-05-21 9:42

We find a variety of statuses like in picture 1 with the same message as shown in the above table. The status presented in the screenshots published on 10 March 2016 refers to status ID 61908299206_10154004905974207 as shown in the above table. The earliest of these statuses is published on 21 May 2013. This is the only status which occurs before the publication date of the first comment, which would make the existence of the first comment valid. Upon further investigation, it can be found that these statuses are in fact part of the same "album" named "Joseph Ribkoff in the Media" as shown in picture 3. This is further confirmed when looking at the comment section of the album shown in picture 4, which shows the same first comment.

However, this comment replication is not observed for all albums when performing an API extract. Therefore, it can be concluded that the database architecture of Facebook makes it such, that there is occasionally a comment duplication bug when taking extracts of statuses which belong to a same album. It can also be observed that these replicated comments share the same comment ID as shown in the table below. Furthermore, they are all assigned to a different status ID. When analyzing the dates of the associated status ID, it is observed that they are by default, inversely ordered by time of publication. Meaning, the last status ID associated with the comment corresponds to the earliest status which the comment can be associated with. The comment_published field is of no help here as the API extract displays the original publication time for all duplicates. These facts will be helpful when dealing with this particularity in the preprocessing step.

Table 5: Part of Extract of the First Repeated Comment in the Album

comment_id	status_id	comment_message	comment_author
10151620036684207_26283591	61908299206_10154420817714207	Remmelink Mode -Styling Hamburgerstraat 61 Doetinchem Heeft een grote collectie Joseph Ribkoff nu in voorraad	Paul Koets
10151620036684207_26283591	61908299206_10154378496979207	Remmelink Mode -Styling Hamburgerstraat 61 Doetinchem Heeft een grote collectie Joseph Ribkoff nu in voorraad	Paul Koets
10151620036684207_26283591	61908299206_10154375745029207	Remmelink Mode -Styling Hamburgerstraat 61 Doetinchem Heeft een grote collectie Joseph Ribkoff nu in voorraad	Paul Koets
10151620036684207_26283591	61908299206_10154333166854207	Remmelink Mode -Styling Hamburgerstraat 61 Doetinchem Heeft een grote collectie Joseph Ribkoff nu in voorraad	Paul Koets
10151620036684207_26283591	61908299206_10154330977104207	Remmelink Mode -Styling Hamburgerstraat 61 Doetinchem Heeft een grote collectie Joseph Ribkoff nu in voorraad	Paul Koets

Figure 5: Joseph Ribkoff Photos, Album Section, Joseph Ribkoff in the Media

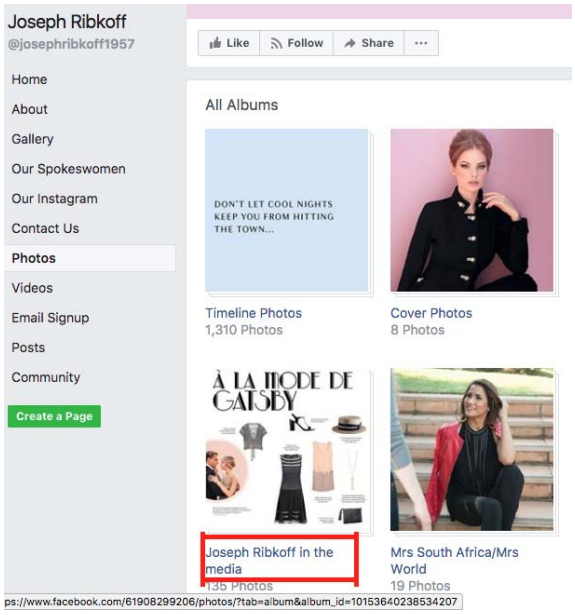
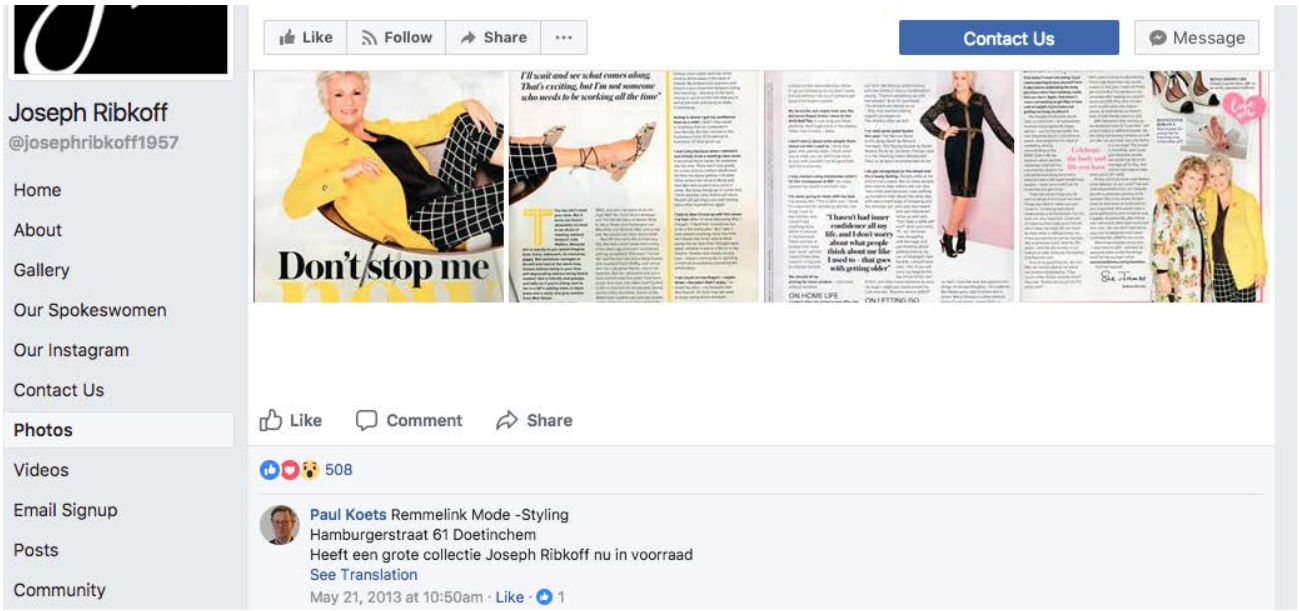


Figure 6: Joseph Ribkoff in the Media Album



Addressing Missing Data

Before proceeding any further to processing special characters and arranging our data in a form adapted for text-mining. It is first important to verify the integrity of our data. This will be done with a visual inspection using the ggplot2 library in R.

Figure 7: Best Buy Number of Posts per Month

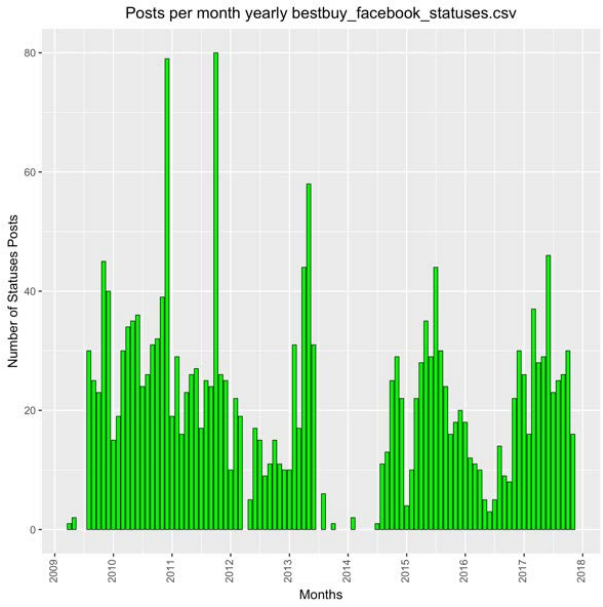
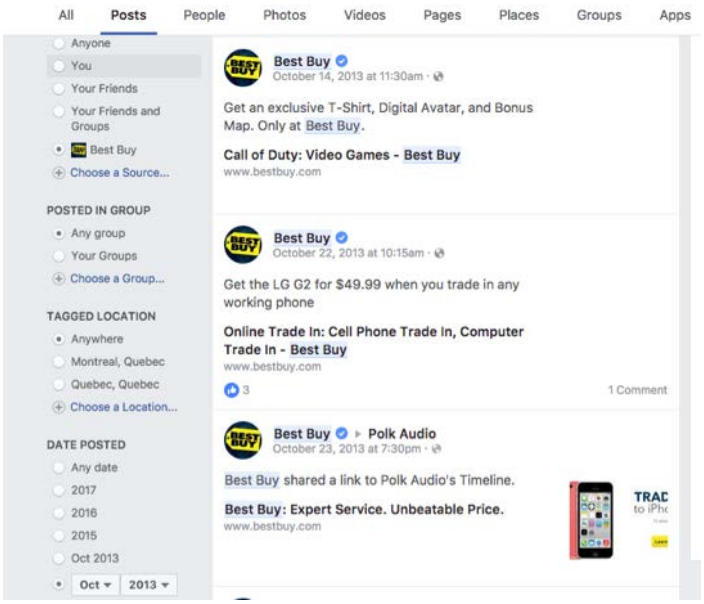


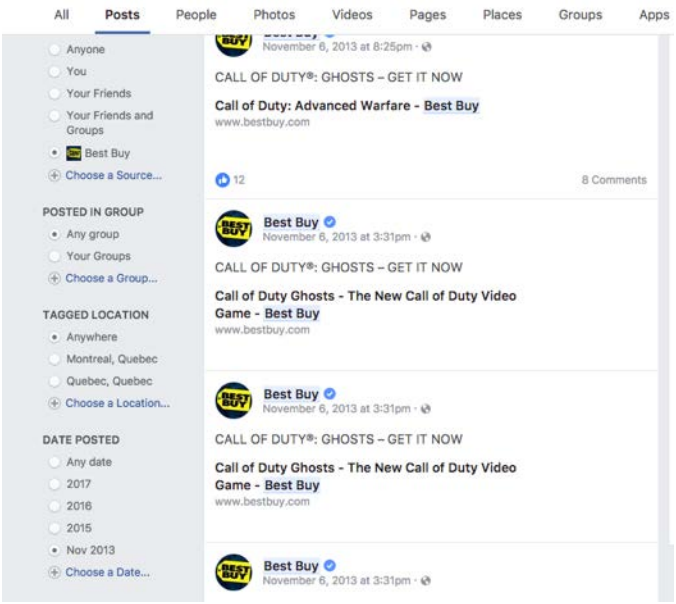
Figure 8: Facebook Search Functionality



As seen in the above graph, there are many months where there is no posting in the Best Buy timeline. Each of these empty months is verified manually with the Facebook post search functionality as shown in the above picture with the term “Best Buy” and a temporal filter on the

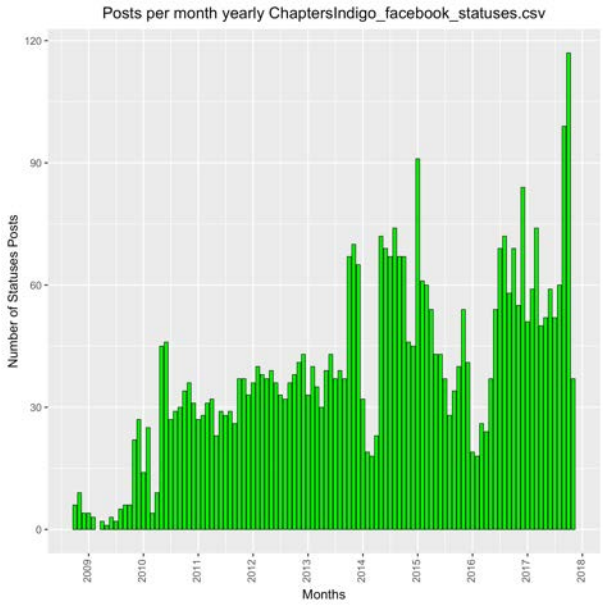
desired month that needs to be verified. All months with no posts in the graph are confirmed to have no posts in the Facebook platform, except for November 2013 as shown on the picture below.

Figure 9: November 2013 Best Buy Statuses



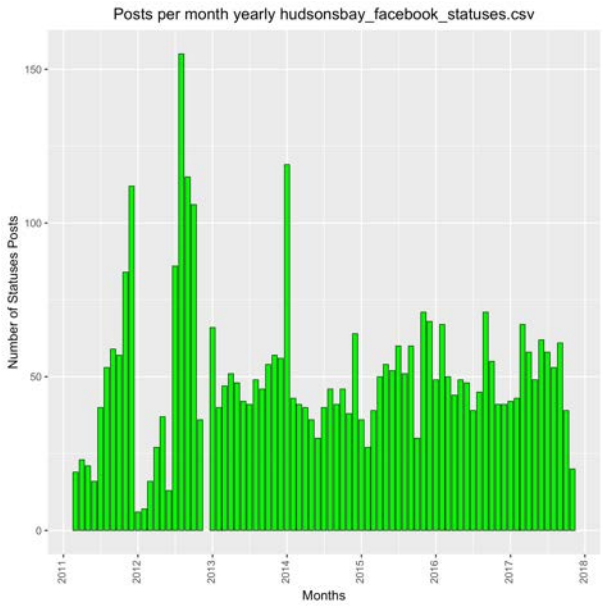
A new request is made to the API to obtain the data for November 2013. The obtained result is unfortunately the same. Since the missing data is only for one month and the month only has 10 posts, this does not affect significantly the analysis, as there are over 2000 statuses for the Best Buy page. We perform the same process for all the other companies.

Figure 10: Chapters Indigo Number of Posts per Month



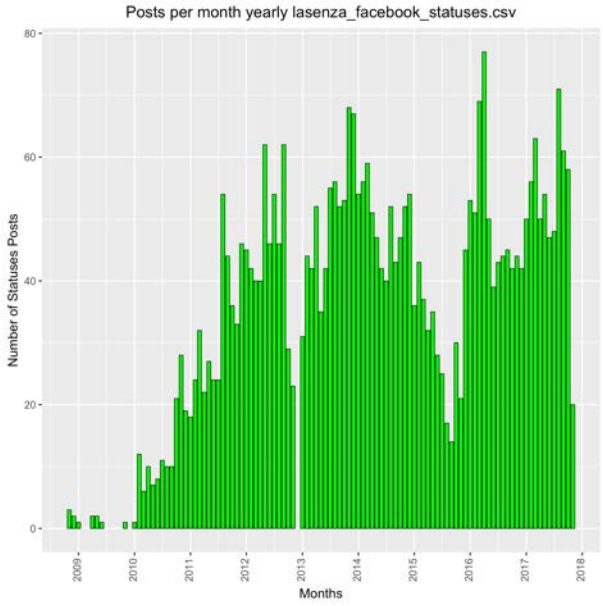
As shown in the graph above, the month with no posts in Chapters Indigo's case is March 2009, this is manually confirmed to be true on the Facebook platform.

Figure 11: Hudson Bay Number of Posts per Month



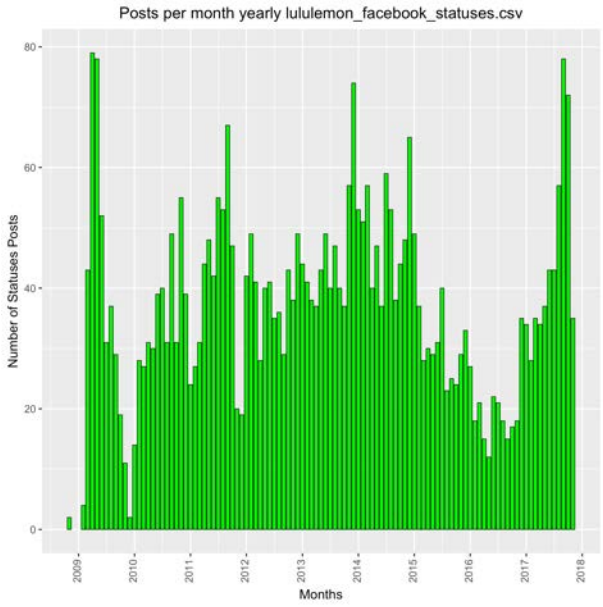
As shown in the graph above, the month with no posts in Hudson Bay's case is December 2012, 100 posts were found for this month after manual validation. A new request is made to the API to get the data for December 2012. The result is unfortunately the same. Since the missing data is only for one month and the month only has 100 posts, this does not affect significantly the analysis, as there are over 4000 statuses for the Hudson Bay's page.

Figure 12: La Senza Number of Posts per Month



As shown in the graph above, there are multiple months with no posts in La Senza's case. After manual verification, there is 2 missing statuses in August 2009, 2 missing statuses in September 2009 and 1 missing status in December 2009, as well as 37 missing statuses in December 2012. The other months are confirmed to have no posts. A new request is made to the API to get the data for the missing months. The result is unfortunately the same. Since the missing data is only for a total of 42 statuses, this does not affect significantly the analysis, as there are over 3700 statuses for the La Senza's page.

Figure 13: Lululemon Number of Posts per Month



As shown in the graph above, the only month with no posts is January 2009 in Lululemon's case, this month contains 100 missing statuses. A new request is made to the API to get the data for January 2009. The result is unfortunately the same. Since the missing data is only for one month and the month only has 100 posts, this does not affect significantly the analysis, as there are over 4000 statuses for the Lululemon page.

Addressing Special Timestamp Statuses

There are special event timestamps that need to be treated since they are not part of the regular statuses posted by companies. The treatment here will be to eliminate them from our dataset. This is reasonable since they do not represent a regular status posting and therefore their textual data cannot be used as variables to detect patterns through text-mining. Since these types of statuses are not numerous and can be easily found by filtering by publication date, the deletion will be completed manually.

Addressing Possible Comment Duplication.

The Facebook API has an occasional bug affecting comments that are made on statuses with images, which are part of the same album. As mentioned before, these comments share the same comments ID. As a first step, we will check if there are any duplicated comment ID in our csv extract. This is done using an R script and the base library function “duplicated”. With this function, we will return the number of duplicated comment ID that is can be found in each company. We get the following result:

- For Best Buy, there are 241 duplicated comments out of 175434
- For Chapters Indigo, there are 188 duplicated comments out of 111882
- For Hudson’s Bay, there are 368 duplicated comments out of 63804
- For La Senza, there are 3916 duplicated comments out of 83782
- For Lululemon, there are 1296 duplicated comments out of 89013

All five companies will have their duplicated comments removed. We can use the fact that the status IDs of the comments are in reverse chronological ordering. This allows us to keep the “original” (the one first posted) comment of the duplicated series by simply keeping the last comment from a group of comments sharing the same ID. This is a standard type of operation which can be done in SQL, Python, etc. We have performed it using R and the base library functions “duplicated” and “order”.

4. Text Treatment

4.1 Replacing Emojis

Emojis are a central part of modern social media communications. The emoji dictionary is part of the Unicode encoding and is therefore hard to analyse by some libraries or languages which takes UTF-8 as its standard encoding for processing of data. The goal in this step is to replace emojis by a written word counterpart. By doing so, we preserve the original meaning of the textual data as well as gain precious features which can be very good predictors of engagement.

Indeed, emojis are often associated with emotional response which unlike a stop-word, hold significant meaning and should therefore not be discarded from the preprocessing phase.

This step is performed by using the *emoji4j* package in Java created by Krishna Chaitanya Thota which can be found on his GitHub. The package allows to transform emojis into their word equivalent. For example, □is transformed into ":cat:" and □ is transformed into ":dog:". To perform the emoji transformation in Java, we recommend using creating a scanner to read the csv and then parsing it into individual words and then passing the words first into the "isEmoji" function to determine if it is an emoji. If the word is an emoji then, we will use the "shortCodify" function to convert emojis into their word counterparts. Words are then recombined into their original comment and outputted into csv form for further steps.

4.2 Encoding

Following the conversion of emojis from Unicode to UTF-8 word equivalents, we are ready to proceed to a series of standard cleaning operations on the corpus. These cleaning procedures are done in the R language with the help of the qdap and the tm libraries, which contain in their implementation most of the transformation we need to proceed to cleaning. Indeed, functions like tm_map from the tm library allow for the use of regex expressions to remove a variety of special textual entities such as hyperlinks. Furthermore, the tm library has a set of predefined stop words lexicon for the English language which we utilize here.

First, symbols such as the dollar sign ($) or the euro (€) sign are replaced by their textual counterparts. This is done to allow further standardization in the text as well as having an input that is entirely made of textual content for the analysis. Next is the removal of hyperlinks and non-ASCII characters, these features do not add any meaning to the text as they are references to other webpages. After we transform all characters to lowercase format, this is simply for standardization of the text. As an example, two words which are the same, for example, "Dog" and "dog" are perceived differently if one has a capitalization, lowercasing all characters allows for words which are the same to be considered so by the algorithm. Next is the removal of numbers, this operation has the same logic as hyperlinks, in of themselves they do not possess

any meaning and are assumed therefore to have low predictive ability. Following that, we remove all punctuation for the same reason as hyperlinks and numbers. Then, stop-words are removed with the help of the tm library defined English stop-words. Finally, whitespaces are removed, this is done because by definition, whitespaces have no meaning. Removing them reduces the size of the files and provide standardization to the text.

It is worthwhile noting that the order of each cleaning operation here is important. For example, replacing symbols should be done before removing punctuation as doing the latter before the former can result in some symbols like the dollar sign ($) being deleted. These symbols could have significant importance since they bring about the subject of money. Another good example is removal of whitespaces which should be done last, this is because the steps before removes characters from the text and therefore creates multiple new whitespaces not present originally.

4.3 Stemming

Stemming is performed with the widely-recognised Porter stemmer. We use the R implementation because of its ease of use. Indeed, we can simply pass individual words through a “Stem” function and get its stemmed counterpart. We therefore recommend proceeding in a similar manner as with the emoji transformer and define a simple algorithm to parse each message into individual words and then feed them to the stemmer, the output is then reconstructed into the same message format as before. Stemming is done for all status text. Stemming is not done for comments as they are not used as predictors for our machine learning models. Indeed, comments are in the same category as the number of shares, comments or likes. This because comments made by customers are the direct response to a post and is therefore dependent on the content of that post and is more akin to a training metric.

4.4 Feature Reduction

Feature reduction is our last preprocessing step. We use the concept of sparseness to eliminate non-relevant features. The sparseness is defined by the sparseness term which can take values between 0 and 1. For example, if 0.995 is chosen, all terms which are sparser than a 0.995 score is removed. A sparseness of 1 would indicate that a word is only present in one document, while

a sparseness of 0 would mean that a word is present in all documents. It can also be thought mathematically as a probability of a word being in a document relative to the size of the corpus and the occurrence of the term within individual documents as well as the corpus (Croft, W., & Harper, D. 1979). The below formula represents that probability, where *t* is the term, *d* an individual document and *N* the total number of documents in the corpus.

$$P(t|d) = \frac{|\{d \in D: t \in d\}|}{N}$$

This is done with the *TM* package in R. The sparseness threshold argument in the *TM* package refers to document frequency of a word by default. When creating a document term matrix (DTM) with our corpus, we can choose to enter a sparseness factor to get a DTM with less than features than the corpus initially contained. In our case, we use 0.995 as our sparseness factor because our corpuses are relatively sparse to begin with, therefore we do not want to eliminate too many terms. Indeed, 0.995 seems to be reasonable choice, because this threshold basically translates to eliminating words which are very rare to the point of being in only a miniscule number of documents, which consequently have low predictive power and can add noise to our predictions.

Method 1: Exploratory Analysis

1. Statistical Analysis

Before proceeding to more sophisticated methods like text-mining, it is worthwhile noting that it is possible to obtain a variety of actionable and useful insights simply from doing a series of standard statistical calculations from the data. An example of this type of analysis based on simple descriptive statistics for the company **Best Buy** is done here to show the potential and results that such an analysis can produce.

First, let us look at the univariate statistics for the number of reactions, the number of comments and the number of shares shown in the table below.

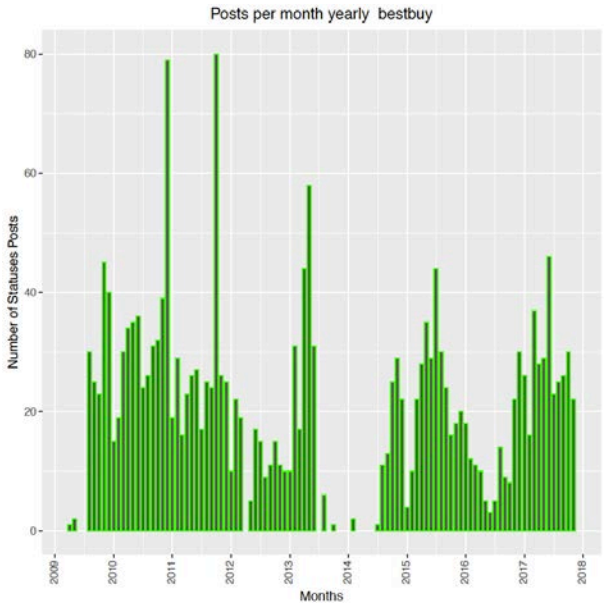
Table 6: Univariate Statistics for Engagement Metrics

Metric	Min	1st Qu.	Median	Mean	3rd Qu.	Max
Reactions	0	87	149	472	304	46915
Comments	0	12	32	79	75	1597
Shares	0	0	5	34	18	4408

This give us a general idea of the distribution of each of the customer engagement metrics. We can refer to these numbers to give us a quick relative comparison point when considering the performance of a status post.

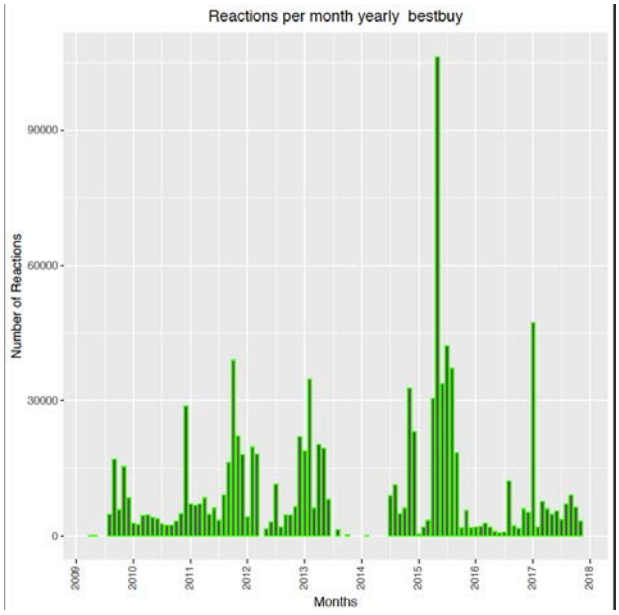
Next, we can generate some visualization graphics to observe the general trends that are happening on the company's social media page.

Figure 14: Posts per Month for Best Buy



We can see in the above graph, that there are periods where there is no activity from the company. From an analyst's point of view, this is worth asking questions to the person in charge of social media. Questions could be of the type; "Why were there no posts in those months?", "Was there an employee managing the social media page at that time?", etc. This way we can understand the social media strategy and the reasoning for these inactivity periods.

Figure 15: Reactions per Month for Best Buy

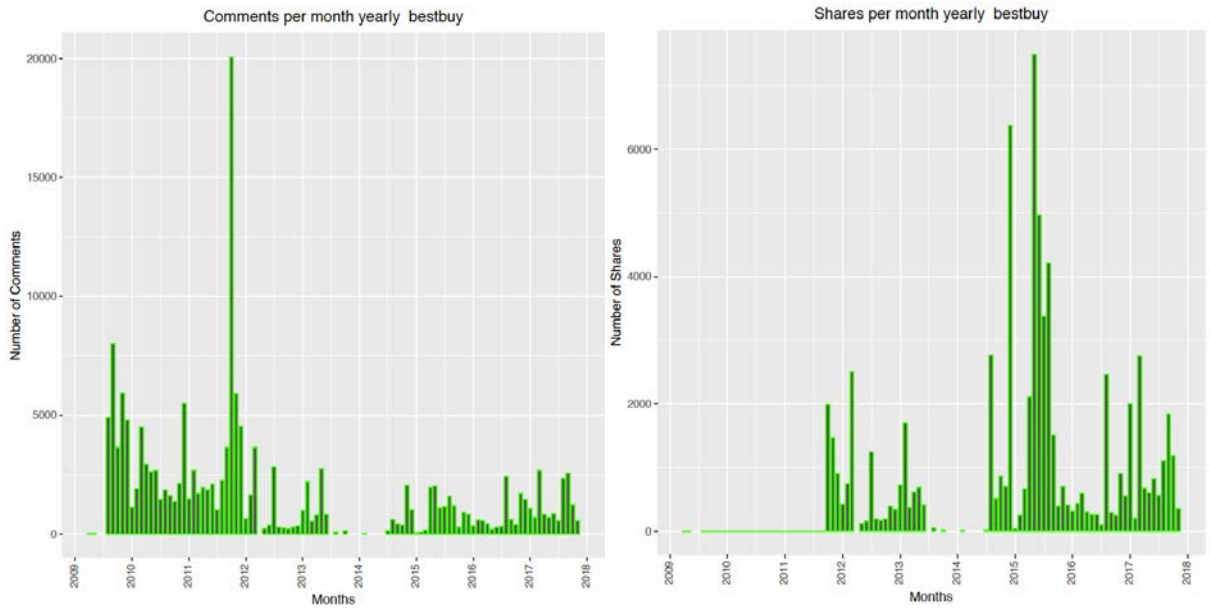


The graph above gives us a summary of which months had the most cumulative reactions. We can see that there are many peaks present that would all be worthy of analysis. We will look in depth at May 2015 which is by far the month with the most reactions. This will serve as an example of a possible analysis that can be done when investigating metrics peak periods. When looking in detail at the posts in May 2015, we notice that three status posts are mainly responsible for the bulk of the reactions. We see below, the images of the statuses which have performed very well in May 2015.

Figure 16: Popular Statuses in May 2015

First, a post on the movie Pitch Perfect 2 gaining 46k likes, next a post on the movie Fifty Shades of Grey gaining 27k likes and finally a post on the minion's movie gaining 25k likes. It is immediately apparent that there is a clean theme revolving around movies here. Therefore, it would be beneficial to inquire with the social media manager if they have continued this practice of increasing engagement with the help of the likability of movies to promote the retail brand. The same exercise can be repeated for the other months where there are significant performance peaks.

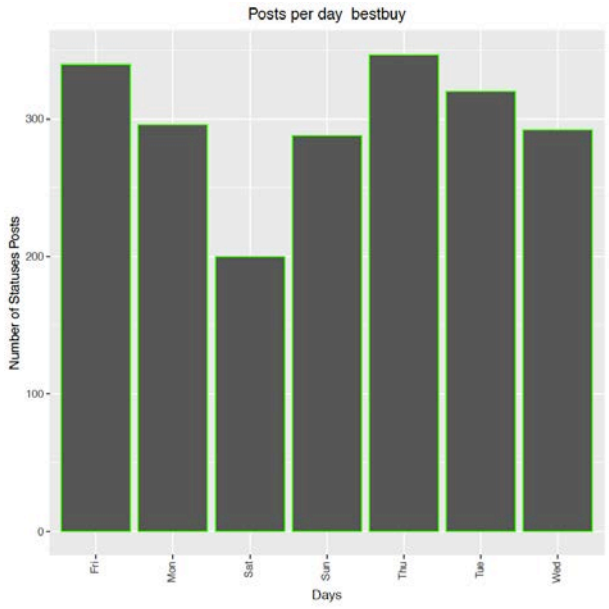
Figure 17: Comments and Shares per Month for Best Buy



Again, this same exercise can be repeated for comments and shares which are also metrics which indicate engagement from the customer. As it is seen in the graphs above, performance peaks are also present in certain months and engagement is not constant, but rather fluctuates dramatically over different time periods.

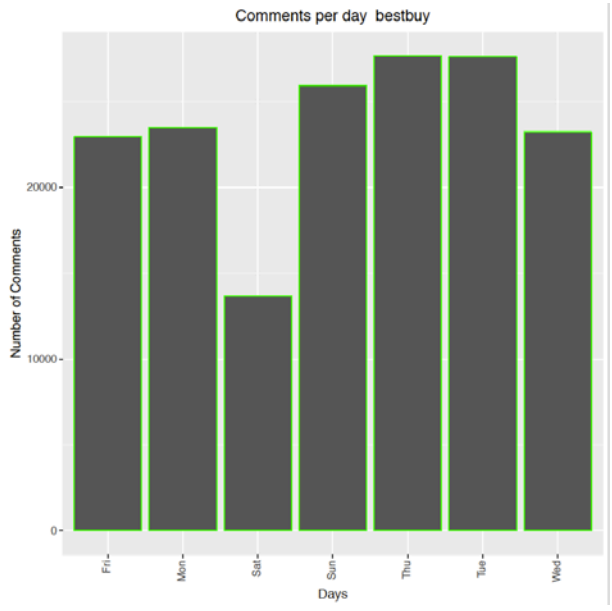
Here we explore examples of further analyses that can done with visualization. In the following graphs, we have the number of posts for each day of the week.

Figure 18: Posts per Day of the Week Best Buy



From this graph, we compare the number of posts versus the engagement on a specific day of the week. For example, let us look at the number of comments per day of the week in the below graph.

Figure 19: Comments per Day of the Week Best Buy



We see that for the most part, the number of comments is well in line with the number of statuses posted. There is perhaps Friday that should be investigated further as there are less comments than there are on Monday, which has less statuses posts than Friday. The same comparative analysis can be done for reactions and shares as well. Visualisations with ratios, such as total number of comments divided by total number of statuses posts with a day of the week breakdown could also be generated. Further visualisation could be done by selecting different differentiating categories, like the type of post (text, image, video). In brief, the possibilities are various. Depending on the objective of the company, multiple insights can be attained by simply looking at the dataset in different ways.

This is by no means an exhaustive list of possible analyses that can be conducted on a social media dataset, but it rather serves as an example that managers and practitioners can already obtain significant and actionable insights without necessarily using text-mining or data-mining techniques that can be prohibitive if the company does not have the technical knowledge or capacity to develop such competencies.

2. Clustering Data

K-Means

We can also perform a K-means clustering as an exploratory method to see if there are any strong themes which are present in the statuses. As an example, for Best Buy, we conducted a K-means analysis with a K of 5 to see if the strongest themes that emerged. We must keep in mind that practitioners and managers may find it better to set K to a number which corresponds more appropriately to their data.

The method employed is by using R and the *stats* package and a frequency DTM of our corpus. The *stats* package uses the Hartigan and Wong variation of the K-means algorithm by default. We have also left all other arguments as default aside from k=5.

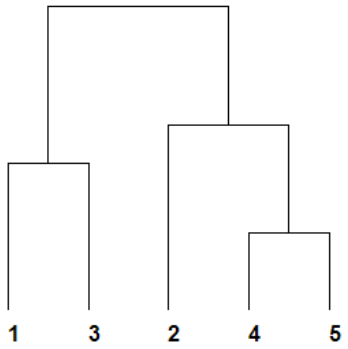
Table 7: K-Means with 5 Groups and Average Reactions, Comments and Shares.

Group	Description	Average Reactions	Average Comments	Average Shares
1 (n=203)	Posts with an invitation to "get" things.	437	52	56
2 (n= 74)	Posts involving "Phones"	334	78	74
3 (n= 1477)	Other Posts	480	87	34
4 (n=15)	Posts about Best Buy and holding a contest	647	99	0
5 (n=320)	Posts with an questioning to "want" things.	483	59	27
Total (n=2089)	All Posts	472	79	34

With a preliminary analysis like this one, we can see from the above table that there are strong themes that arise. With these information, it is possible to benchmark certain themes versus the average performance of all posts. For example, we see that posts advertising "phones" underperform relative to the average. Furthermore, K-means allows us to reveal the social media strategy of a company and see if they have self-defined hashtag words or themes. Indeed, in the case of Best Buy, we see a recurring theme with some posts with the formulation "get", "want" or involving the promotion of "phones".

Hierarchical Clustering

Figure 20: Illustrative Example of a Dendrogram with 5 groups



We performed a hierarchical clustering on our Best Buy corpus and trimmed our dendrogram to 5 end groups as represented above. The method employed is by using R and the *hclust* function from *stats* package and a frequency DTM of our corpus. We used the Euclidian distance and Ward as the agglomeration method. The remainder of parameters were left to their default values. The "cutree" function was used to only leave us with 5 groups.

We can perform an analysis like K-means by looking at the categories that are outputted and examining those. The table below shows a description of each category as well as the average reactions, comments and shares.

Table 8: Hierarchical Clustering with 5 Groups and Average Reactions, Comments and Shares.

Group	Description	Average Reactions	Average Comments	Average Shares
1 (n=1545)	Other Posts	522	85	41
2 (n=193)	Posts about movies, phones and tvs	454	50	24
3 (n=284)	Posts about games and apple products	250	68	10
4 (n=55)	Posts relying on non textual information	208	59	16
5 (n=12)	Posts about Best Buy and holding a contest	786	111	0
Total (n=2089)	All Posts	472	79	34

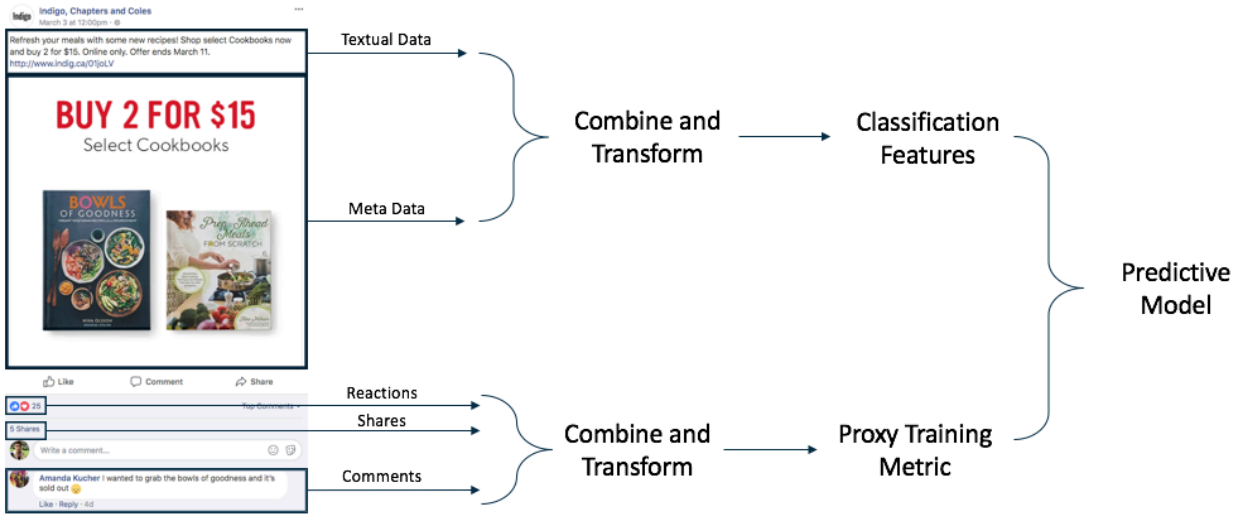
3. Business Recommendations

We recommend businesses to first proceed with the above exploratory analysis as it is interpretable and easy to perform. From the statistical analysis, we recommend posting at the best times for increased engagement. We also recommend manually investigating peaks and lows and asking questions to social media managers regarding events that occurred during those periods. From the clustering analysis, we recommend identifying as many themes as possible from the corpus and evaluating the performance of each theme. Managers can then identify themes which are high performers and get a better picture of their audience. We then suggest adjusting the marketing strategy appropriately and develop new themes which are related to existing ones which are popular.

Method 2: Predicting Engagement Performance

Currently, there is no easy way of implementing a machine learning process for predicting engagement performance for social media, more precisely Facebook, from textual content. We provide guidelines and suggest a procedure to get good performance for mining social media posts. What we propose can be characterized as "Proxy Learning" and consists in using the RSC (reaction/shares/comments) metrics left by users on Facebook posts to assign a level of engagement to a Facebook post. We provide a standard way to translate RSC into a binary training class expressing high or low engagement. This addresses a known problem in social media mining called the evaluation dilemma, which can the described as the presence or absence of a ground truth in the training data to make data mining possible (Zafarani, R., et al., 2014). Such problem can be solved by using humans to tag the training data with platforms such as the Amazon Turk Platform (Zafarani, R., et al., 2014). However, this is an inefficient process as we are limited by human speed and scalability. By using the RSC metrics, we can save enormously on human costs and lift limitations on our training sample sizes. The process can be summarised in the diagram below.

Figure 21: Summary of Proxy Learning for Engagement Prediction



As well, we can test whether the use of metadata such as time of posting and the type of status (text, photo, video, etc.) has an incidence on performance. Finally, we look at a multitude of readily available machine learning algorithms and see which ones perform best for our application case which involves social media statuses which are typically limited in textual length.

1. Method Development

1.1 Meta Variable Definition

First let us define what are meta-variables in the context of our analysis. Traditionally, data mining analyses require a set of well tagged data to train a classification algorithm. For example, if we want to classify whether a person will buy a pair of shoe when going into a store based on certain characteristics, we would have a set of training data in which we know for certain if that person has bought shoes or not. The assumption here is that the data given to us is accurate and trustworthy.

For text mining analysis, it is common that models rely on human classification first and then use that data to train the machine learning algorithm (Zafarani, R., et al., 2014). As an example, in the case of sentiment analysis, a reader would have to interpret whether a phrase is positive, neural or negative based on his human semantic understanding of the phrase. This type of human classification would be both costly and inadequate for our analysis. Indeed, our objective is to classify statuses in two categories, high performers and average/low performers. Performance within the Facebook ecosystem is measured regarding engagement, meaning how responsive are readers of a status. Facebook provides for three measures which we assume are good proxies for engagement, namely the number of reactions, shares and comments. This is a rather fair and logical assumption as these scores are directly proportional to interactions that people have with a status post.

The difficulty therefore is to derive the equivalent of a “human classification” or ground truth (Zafarani, R., et al., 2014) to define whether a status is a high performer or not. Indeed, this

classification is necessary to train our machine learning algorithm. For prediction purposes, we do not know the level of engagement a post will receive before its posting. In deriving an appropriate pre-training classification method equivalent to “human classification”, there are many considerations to contemplate.

First, we cannot simply use the raw number of reactions/shares/comments because of a lack of standardisation. Indeed, Facebook pages evolve over time, usually accumulating a larger audience. Therefore, there is a bias in the engagement levels of later posts since they usually have a larger audience and thus a larger reach. We must consequently find the best transformation to correct this bias and lead us to a high rate of correct classification.

Second, there are three ways users can react to a post: reactions/shares/comments. There is no empirical evidence on the relative importance of one type of feedback in relation to another. We therefore must find the right combination function that will yield the best classification.

Finally, there is also useful additional meta-data contained in a status such as the time of posting and the type of post. This information could be beneficial to the classification and we have to find the correct way to incorporate them in our model.

Choice of Machine Learning Algorithms

As with meta-variables selection, many machine leaning algorithms are available for use. We use the R package *RTextTools* for its simplicity of use and large choice of algorithms. We test 7 different algorithms which are available in the *RTextTools* package namely: SVM (Support Vector Matrix), SDLA (Scaled Linear Discriminant Analysis), Tree Bagging, Tree Logit Boost, General linearized models (GLMNET), Max Entropy, Random Forest. We use the default setting of the *RTextTools* package for every algorithm listed here. Results from all the algorithms using their default configurations are compared and the best performing ones are kept and adjusted for the final model.

1.3 Use of Additional Meta-Data

Let us start by discussing the addition of additional meta-data to our dataset to increase its performance. For our analysis, we consider three meta-data variables, namely the weekday of publication, the hour of publication and the type of post. The assumption being that each of these variables can be beneficial on the performance of a status. Additions will be done in the form of additional words to be added to the existing textual data of a status.

First, we add the day of the week and hour of the status posting into the textual data. As seen in the descriptive statistics section, temporal data could indicate the level of performance of a status. We do not use the month or the year of the status to keep the practicality of the model. Indeed, even if the month or year of the status had a significant impact on predictive performance, we could not realistically leverage in real use cases, as it would be unreasonable for companies to wait months to post a status or to go back a year in time.

Secondly, we add the status type into the textual data. This meta-data variable indicates the nature of the status. The options are link, photo, status and video. This variable is used because it clearly separates each status into four categories. If any of the categories were to have significantly more engagement over the other format, this use would allow us to capture this in our text mining.

The additions are done in R. Using the status type is trivial as it amounts to simply appending further data to the text. The day of the week and hour of posting are extracted from the status published time which has a YYYY-MM-DD HH:MM:SS format. This can be done using base R and the weekdays function.

Finally, all added words have a marker appended in front of them as they do not have the same signification as original textual data. For example, if a status mentions the word “Tuesday” but is published on Monday, by using “Monday” directly, we would cause confusion as both words are not equivalent. We use the characters “inj” and append it to all the meta-data used text. Therefore, the “Monday” in the example becomes “injMonday”.

We test each meta-data variable by incrementing the number of uses while excluding the other variables assuming therefore independence between meta-variables. The settings that are tested are 0 to 3 uses for each variable, this represents a total of 12 possible combinations. Doing this for each company dataset for a total of 60 tests. For example, we can add the status type once, the weekday once and the hour once as a possible combination. For each meta-data variable, the configuration yielding the highest classification performance as per configuration F-Score determines the value to be used in our final model.

1.4 Training Classificatory Metric (Based on Shares, Reactions, Number of Comments)

As mentioned earlier, reactions/shares/comments need standardization to adjust for bias and to transform statuses metrics into a training class. We propose four methods for calculating a standard score for the number of RSC (reactions/shares/comments). The transformations used here is rather simple and complicated transformation function should not be considered. Indeed, we must keep in mind that the prediction that we will obtain will be on this training class. We therefore should keep interpretability of this transformation to a maximum, otherwise we end up predicting something that we cannot really interpret.

First, we use a ratio of the current number of RSC divided by the mean of all RSC available up to the date of the current RSC used in the calculation. For clarification, as an example, if we have started to post on our page on the 1st of January 2017 and we get 10 comments on a post on the 7th of January, then the comment score for this status would be 10 divided by the mean of all comments on all statuses that were published before this one.

Secondly, we use a moving average window of previous statuses rather than using all the previous statuses. For the size of the moving window, we shall use 7, 14 and 28. Which corresponds to roughly a week, two weeks and a month if statuses are posted daily.

Thirdly, we use a more complicated method based on using a moving window of previous statuses. We assume that this moving window follows a normal distribution and calculate the

standard deviation of the latest entry relative to the distribution of the moving window. For the size of the moving window, we shall use 7, 14 and 28, the reasoning being the same as in the second method.

Finally, we use the standard deviation score based on a normal distribution defined by all RSC available up to the date of the current RSC used in the calculation. This is like the first mentioned approach, the difference being the use a standard deviation score.

We test each classification metric with a standard configuration that is detailed in the following sections. The best performing metric as per configuration F-Score is used for our final model.

1.5 Relative Importance of Reactions/Shares/Comments

As mentioned earlier, the optimal relative importance of RSC is still unknown. As all three metrics are the result of direct customer feedback, we assume that all three are relevant in forming our final training class. Following that logic, we combine all three metrics with a weighted sum. We test a variety of weight combinations such as equal weighting, 1.5 times importance of some metrics and 2 times importance of some metrics. We do not go beyond 2, as that would indicate that one of the RSC elements is significantly more important relative to another. It is assumed that this is not the case, as they are all valid ways for users to express engagement. We test multiple combinations of RSC weighting with a standard configuration that is detailed in the following sections. The best performing metric as per configuration F-Score is used for our final model.

1.6 Separation Point

After calculating the weighted final score of the combined RSC, we still need to separate the dataset into a binary class composed of high performers and average/low performers. We do this by separating along the x^{th} percentile. Thus the top x^{th} scores is of the category high performers and the remaining $(1\text{-}x)^{th}$ percentile is of the category average/low performers. We determine this by attempting to find a strong separation in scores between a cut-off percentile.

1.7 Data Partitioning for Validation and Minority Class Adjustment

For all analyses methods that we try, we split the data set into 80% training and 20% validation. We perform a randomization on our dataset prior to selecting training and validation groups. We prefer this method to keeping the data in chronological order because of the nature of Facebook data. For certain companies, recent data performance is mediocre leading to a possibly biased validation score. Indeed, if our model does well in detecting negative posts but poorly in detecting positive ones, then we will have an artificially high accuracy due to the lack of positive class cases in the validation set. We want to therefore have a fairer picture for model performance and randomizing the chronological order of posts before proceeding to the separation of training and validation samples.

Because of the nature of our data, performant posts will be a minority class and therefore some adjustment is needed for the training set to properly create a model. If a group of the binary class is underrepresented, we perform random oversampling with return to balance the training data. We then have a 50/50 distribution of the positive and negative class in the training set. The ratio of positive and negative class data for the validation set is not changed.

1.8 Performance Evaluation

Performance evaluation is done by comparing the best F-measure amongst the algorithms used for each configuration. In cases of ties, we favor the simplest configuration with the least processing. A more complete range of measures is only used once the best algorithms are selected and their hyper-parameters are optimized. We only use F-measure since this is an easy measure on which we can compare the different configurations (Zafarani, R., et al., 2014). Furthermore, it is a good indicator as it summarises precision and recall in the following formula:

$$F = 2 * \frac{precision * recall}{precision + recall}$$

Where precision represents the percent of times we are right when we make a positive prediction (Zafarani, R., et al., 2014) and can be defined with the following formula:

$$Precision = \frac{True\ Positives}{True\ Postives + False\ Positives}$$

Where recall represents the percent of positive class elements our model can extract over all the positive class elements in the population (Zafarani, R., et al., 2014) and can be defined with the following formula:

$$Recall = \frac{True\ Positives}{True\ Postives + False\ Negatives}$$

2. Method Calibration

2.1 Definition of a Standard Configuration for Iterative Testing

Since we must find the best configuration for many meta-variables, we define a standard configuration first, to have a starting point on which to find individual optimal results for the respective meta-variables. The following table summarizes the meta-variables to optimize as well as the value that we define to be the standard configuration.

Table 9: Meta-Variable Standard Configuration

Meta-Variable	Standard Configuration Value
Separation Point	X^{th} Top Percentile (To be determined first)
Training Classificatory Metric	Current score divided by mean up to current score (Configuration 1)
Meta-Data Use	No Use
Relative Importance of RSC	Equal
Algorithm Choice	ALL

The data partitioning is of 80% training and 20% test and the sparseness factor is set to 0.995 as detailed in the previous sections.

2.2 Separation Point Optimization

We attempt to determine for each of our dataset, the appropriate quantile separation to separate high performers from average/low performers. Intuitively this means finding the quantile at which scores see a drastic increase relative to previous quantiles.

This is the exact approach that we use to determine the optimal cut-off point between our two classes. More precisely, we first define an appropriate quantile granularity to observe. For our purposes, we use steps of 5%. From each of these steps we calculate the increase in score from one quantile to another.

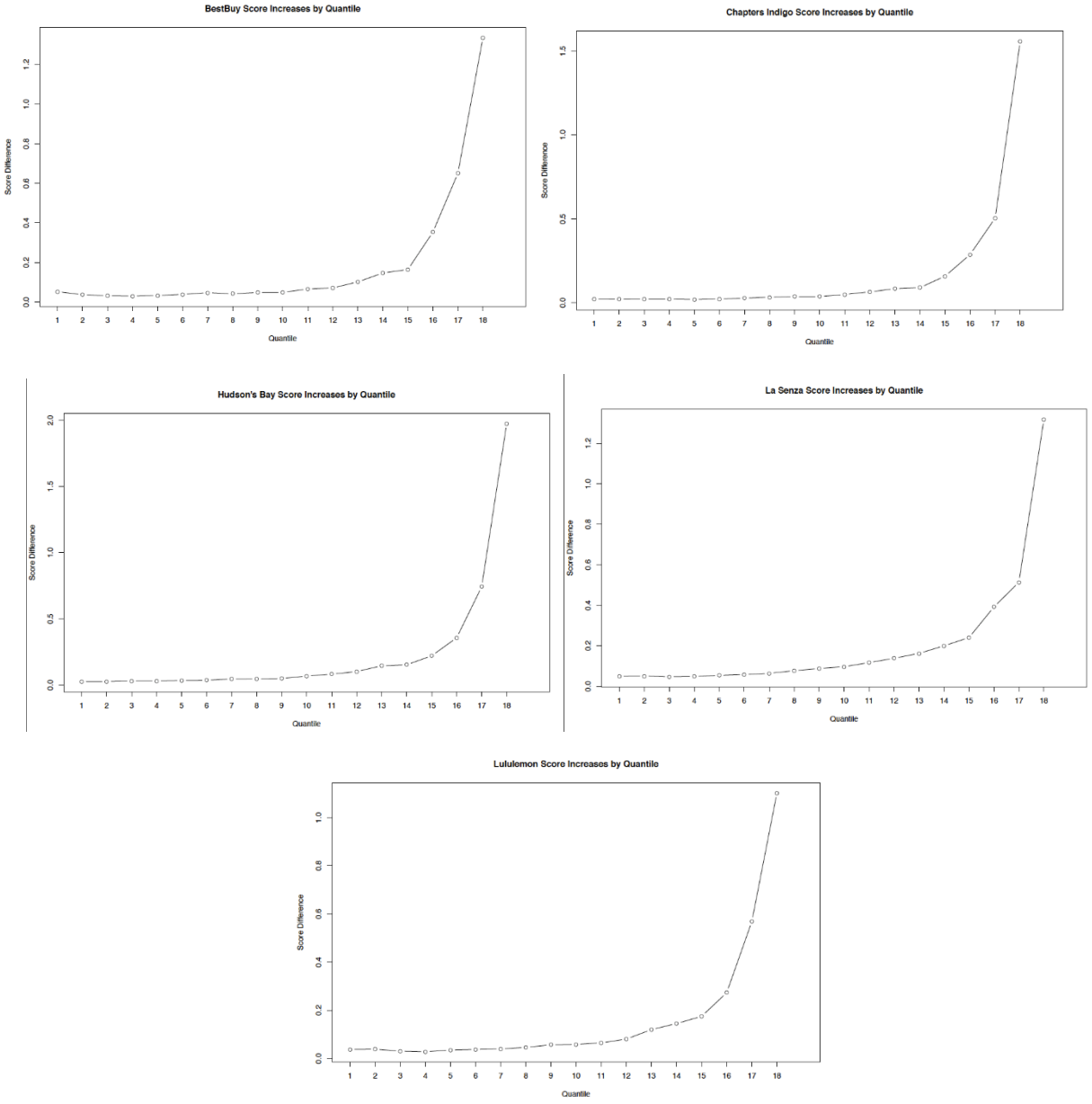
We define the cut-off point as the quantile which has a difference in score of more than 50% than that of the previous quantile difference, meaning a ratio superior to 1.5. Since this is a measure that is relative and scales according to any binary training class configuration, we can find a definite x^{th} top percentile cut-off point to use going forward for each of our datasets.

Table 10: Cut-Off Point at 50% Increase

Company	Cut-Off Point at 50% Increase
Best Buy	80^{th} Percentile
Chapters Indigo	75^{th} Percentile
Hudson's Bay	80^{th} Percentile
La Senza	80^{th} Percentile
Lululemon	80^{th} Percentile

All five companies have very similar cut-off points at the 80^{th} percentile, with Chapters Indigo being the only one at 75^{th} percentile. If we look at the plot of the percentile differences below, we see that the increase in differences all rise sharply near the 80^{th} percentile mark. This give us an interesting insight, that on average 1 in 5 status posts perform significantly better than the rest.

Figure 22: Percentile Differences of each Company



Note that the last quantile difference has been cut-off from the visualisation since it is of several hundred times larger than the rest of the data points, they represent the ultra-viral status posts.

As the results are rather uniform, we use an 80th percentile cut-off point for all 5 companies to have better standardization and comparison when analyzing results. This gives us the following configuration going forward.

Table 11: Meta-Variable Configuration (After Setting Separation Point)

Meta-Variable	Standard Configuration Value
Separation Point	80th Top Percentile
Training Classificatory Metric	Current score divided by mean up to current score (Configuration 1)
Meta-Data Use	No Use
Relative Importance of RSC	Equal
Algorithm Choice	ALL

2.3 Training Classificatory Metric

We now optimize the training classificatory metric. We employ the standard configuration for the other variables with a separation point of 80th top percentile. For the training classificatory metric there are four defined types of configurations, two of them having three sub configurations each for a total of 8 configurations to try. The following table lists the configurations to be tested.

Table 12: Training Classificatory Metric Configurations

Configuration	Description
1	Current score divided by mean up to current score
2	Moving average window of previous scores: Window size 7
3	Moving average window of previous scores: Window size 14
4	Moving average window of previous scores: Window size 28
5	Standard deviation of moving average window: Window size 7
6	Standard deviation of moving average window: Window size 14
7	Standard deviation of moving average window: Window size 28
8	Standard deviation defined by up to current score values

The results are the following:

Table 13: Top F-Score by Configuration and Company for Training Classificatory Metric

Configuration /Company	BestBuy	Chapters Indigo	Hudson's Bay	La Senza	Lululemon	Mean
1	0.625	0.64	0.625	0.63	0.54	0.612
2	0.6	0.6	0.575	0.585	0.545	0.581
3	0.62	0.595	0.595	0.575	0.545	0.586
4	0.625	0.635	0.625	0.615	0.53	0.606
5	0.57	0.62	0.585	0.585	0.55	0.582
6	0.625	0.605	0.61	0.56	0.575	0.595
7	0.6	0.615	0.615	0.605	0.555	0.598
8	0.645	0.635	0.625	0.62	0.555	0.616

We see that overall configuration 8 has performed the best with the highest F-score. Configuration 1 also performed well being in second place. The top F-scores of both configurations are very similar, although depending on the company, one configuration performs slightly better than the other. What we mostly see here is that taking scores derived from recent periods rather than having a global relative store turns out to be slightly worse for classification performance.

2.4 Use of Additional Meta-Data

Table 14: Meta-Variable Configuration (After Training Classificatory Metric)

Meta-Variable	Standard Configuration Value
Separation Point	80th Top Percentile
Training Classificatory Metric	Standard deviation defined by up to current score values (Configuration 8)
Meta-Data Use	No Use
Relative Importance of RSC	Equal
Algorithm Choice	ALL

Next, we optimize the number of meta-data words use. We can set the training classificatory metric to configuration 8 and go forward with the configuration shown above. The following table lists the configurations to be tested.

Table 15: Use of Additional Meta-Data Configurations

Configuration	Hour Use	DOW Use	Type Use
Null	0	0	0
H1	1	0	0
H2	2	0	0
H3	3	0	0
D1	0	1	0
D2	0	2	0
D3	0	3	0
T1	0	0	1
T2	0	0	2
T3	0	0	3

We then compare results by section. Meaning for example, {Null, H1, H2, H3} is compared within their own category to find the best number of meta-data hour words that should be used. The same is done for the two other categories. The final configuration used is a combination of the finding of the three-test consisting in the best number for each of the category.

The results are the following:

Table 16: Top F-Score by Use of Additional Meta-Data Configurations (Weekday)

	BestBuy	Chapters Indigo	Hudson's Bay	La Senza	Lululemon	Average
Null	0.62	0.65	0.63	0.62	0.545	0.613
D1	0.62	0.65	0.63	0.62	0.545	0.613
D2	0.62	0.65	0.63	0.62	0.545	0.613
D3	0.64	0.65	0.645	0.62	0.53	0.617

Table 17: Top F-Score by Use of Additional Meta-Data Configurations (Hour)

	BestBuy	Chapters Indigo	Hudson's Bay	La Senza	Lululemon	Average
Null	0.62	0.65	0.63	0.62	0.545	0.613
H1	0.64	0.65	0.645	0.615	0.53	0.616
H2	0.64	0.65	0.645	0.625	0.535	0.619
H3	0.645	0.635	0.625	0.62	0.555	0.616

Table 18: Top F-Score by Use of Additional Meta-Data Configurations (Type)

	BestBuy	Chapters Indigo	Hudson's Bay	La Senza	Lululemon	Average
Null	0.62	0.65	0.63	0.62	0.545	0.613
T1	0.625	0.645	0.63	0.62	0.59	0.622
T2	0.625	0.645	0.63	0.62	0.59	0.622
T3	0.625	0.645	0.63	0.625	0.59	0.623

We can see that the meta-data use mostly have very little effects on the recall score. The null configuration performs generally as well as the other configurations. Furthermore, if we take a closer look at any of the configurations, the difference for an dataset with meta-data and null are very similar, if not insignificant.

Indeed, there seems to be only a significant difference for the type of posts, which performs slightly better than null. For the other categories, the increase does not seem to be convincing enough for us to go with their configuration over null. Furthermore, for the hour, H2 performed better than H3 which would seem counter-intuitive since adding more information should result in a more precise model. We attribute this to statistical randomness and is a further sign that the temporal meta-data is unreliable in prediction and does not add sufficient value.

Therefore, T1 is the configuration used going forward, as T2 and T3 produce essentially the same effect. This is a surprising result as we would think that temporal information would be a

good contributor in performance. However, the results here show that most of the gain can be attributed to the textual content of the status itself, meaning text mining is even more relevant as a tool versus the use of only statistical analysis.

3.5 Relative Importance of Reactions/Shares/Comments

Table 19: Meta-Variable Configuration (After Use of Additional Meta-Data)

Meta-Variable	Standard Configuration Value
Separation Point	80th Top Percentile
Training Classificatory Metric	Standard deviation defined by up to current score values (Configuration 8)
Meta-Data Use	T1
Relative Importance of RSC	Equal
Algorithm Choice	ALL

Next, we optimize the weighting of the reactions/shares/comments. We can set meta-data use at T1. The configuration going forward is shown above. The below table lists the configurations for relative importance of RSC to be tested.

Table 20: RSC Weighting Configurations

Configuration	Reaction Weight	Shares Weight	Comments Weight
Equal	1	1	1
C1.5	1	1	1.5
C2	1	1	2
R1.5	1.5	1	1
R1.5 + C1.5	1.5	1	1.5
R1.5 + S1.5	1.5	1.5	1
R1.5 + S2	1.5	2	1
R1.5+C2	1.5	1	2
R2	2	1	1
R2 + C1.5	2	1	1.5
R2 + C2	2	1	2
R2 + S1.5	2	1.5	1
R2 + S2	2	2	1
S1.5	1	1.5	1
S1.5 + C1.5	1	1.5	1.5
S1.5 + C2	1	1.5	2
S2	1	2	1
S2 + C1.5	1	2	1.5
S2 + C2	1	2	2

The results are the following:

Table 21: Top F Score by RSC Weighting Configuration

	BestBuy	Chapters Indigo	Hudson's Bay	La Senza	Lululemon	Average
Equal Weight	0.625	0.645	0.63	0.62	0.59	0.622
C1.5	0.645	0.65	0.64	0.61	0.595	0.628
C2	0.64	0.665	0.64	0.62	0.615	0.636
R1.5	0.625	0.675	0.62	0.62	0.61	0.63
R1.5 + C1.5	0.64	0.63	0.63	0.62	0.6	0.624
R1.5 + C2	0.63	0.655	0.62	0.62	0.595	0.624
R1.5 + S1.5	0.615	0.66	0.65	0.605	0.605	0.627
R1.5 + S2	0.625	0.65	0.645	0.62	0.61	0.63
R2	0.625	0.68	0.64	0.625	0.625	0.639
R2 + C1.5	0.62	0.665	0.63	0.615	0.62	0.63
R2 + C2	0.64	0.65	0.63	0.615	0.6	0.627
R2 + S1.5	0.64	0.65	0.63	0.625	0.61	0.631
R2 + S2	0.62	0.665	0.65	0.615	0.62	0.634
S1.5	0.645	0.67	0.635	0.595	0.59	0.627
S1.5 + C1.5	0.65	0.64	0.63	0.64	0.585	0.629
S1.5 + C2	0.655	0.65	0.64	0.64	0.58	0.633
S2	0.625	0.66	0.65	0.61	0.61	0.631
S2 + C1.5	0.61	0.655	0.635	0.61	0.59	0.62
S2 + C2	0.655	0.65	0.64	0.625	0.57	0.628

The best average configuration in terms of F-Score is R2, meaning reactions carry twice as much weight as comments and shares. Relative to the equal weighting, R2 performs slightly better by almost 2%. Close seconds are C2 and S1.5 + C2, this seems to indicate that comments and shares can also provide value depending on the company. Indeed, shares and comments do a better job of prediction for Best Buy and La Senza. However, the same can be said for Chapters Indigo and Lululemon for reactions.

What is clear here is that equal weighting performs the second worst of all the tested configurations. These results indicate that in general, it is better to use imbalanced weights regarding RSC when using these metrics to create training metrics for Facebook status performance prediction. We therefore go forward with the R2 configuration as it is a reasonable choice and could be indicative that reactions are the most significant influencer of engagement, which would fit into an intuitive narrative since it is the engagement action most frequently performed on Facebook with the least effort and therefore would be the most reliable.

3. Choice of Algorithms and Hyper-Parameter Optimization

3.1 Choice of Algorithms

From all the configurations that we have previously tested, we see which algorithms have performed generally better, by doing a count on the number of times they have been the best performer per configuration and dataset. From our 37 configurations across 5 datasets, this give us a total of 185 instances to work with. We use the F-Score to evaluate which is the best performing algorithm for a dataset.

Table 22: Count of Top Performing Algorithm per Test

Algorithm	Count
Logit Boost	2
GLMNET	8
SVM	23
Random Forest (Bagging)	27
Random Forest	123
Max Entropy	1
SDLA	1

We see that in general, random forest outperforms all other algorithms by a wide margin, this may be due to its inherent strength in performing well under minimal tuning and non-standard data (Breiman, L. 2001). Indeed, we have used the default configurations of the *RTextTools* package for all algorithms. We therefore also consider SVM and bagging as valid options that are worth exploring.

3.2 Hyper-Parameter Optimization

After looking at the top performing algorithms, we go ahead with SVM, random forest and bagging and have the following configuration for optimizing hyper-parameters using grid search in the three models. We make use of 5-fold cross validation for SVM to avoid overfitting.

Table 23: Meta-Variable Configuration (After Choice of Algorithm)

Meta-Variable	Standard Configuration Value
Separation Point	80th Top Percentile
Training Classificatory Metric	Standard deviation defined by up to current score values (Configuration 8)
Meta-Data Use	T1
Relative Importance of RSC	R2
Algorithm Choice	SVM, RF and Bagging

For SVM, we will be using the *e1071* R package implementation (which is also used in the *RTextTools* package) and going to tune three parameters, the cost of constraints violation (C), the gamma and the kernel type. The C parameter determines the penalty for violating a constraint, having a higher cost can force boundary to be smoother and perform better in training, however, these precise separations might produce worse results in testing (Duan, K., et al., 2003). The gamma represents the reach of each data point relative to the support vectors (Duan, K., et al., 2003). A low value of gamma means that data points further away from the separation region have more of an effect in determining the position of the support vectors. A high value of gamma means that the closer points, meaning the values which are more fringe relative to their data group, has a stronger influence on the model (Duan, K., et al., 2003). As for the kernel, the *e1071* package has a selection of linear, polynomial, radial basis and sigmoid configurations.

For random forests, we are using the *randomForest* R package implementation (which is also used in the *RTextTools* package). The three parameters we focus on are the number of decision trees, the number of candidate variables at each split and whether sampling would be done with replacement or not. Performance usually increases up to performance stability as more decision trees are added , however this takes longer to compute (Genuer, R., et al., 2010). The number of candidate variables at each split can be tricky, we want a good balance where significant variables are selected to represent their importance in the model, but less frequently selected variables could also be useful for specific cases within the dataset (Genuer, R., et al., 2010). To use replacement or not is another similar issue of having the correct variable representation (Genuer, R., et al., 2010) in cases where subsampling is used.

For bagging, we are using the *ipred* R package implementation (which is also used in the *RTextTools* package). The most important hyper parameter in bagging is the number of bootstrap

replications (Liu, B., & Zhang, L. 2012). In theory, the more replications of the tree, the more successful the model will be up to performance stability (Liu, B., & Zhang, L. 2012). We therefore try several numbers of replications and observe the result.

SVM Test Space

For SVM, a paper from Hsu, Chih-wei et al., (2003) suggests proceeding with values of C and gamma which differ exponentially in scale. We therefore use the following configurations:

Table 24: SVM Hyper-Parameter Test Space

Hyper-parameter	**Range**
C	2^-5 to 2^5
Gamma	2^-5 to 2^5
Kernel	linear, polynomial (degree 3), radial basis and sigmoid

This represents a total of 484 hyper-parameters configurations for a total of 2420 tests. Ideally a more exhaustive grid would be used, but this is sufficient relative to the computational resources at hand.

RF Test Space

For random forest, the hyper-parameter space can be more readily defined since parameters are mostly a consequence of the dataset like the number of candidate variables, in our case there are roughly 300+ after dimensionality reduction using a sparseness factor. As for the number of trees, it is mentioned earlier that the performance generally increases as the number of trees goes up. We therefore use the following configurations:

Table 25: Random Forest Hyper-Parameter Test Space

Hyper-parameter	**Range**
Number of Trees	2^6 to 2^8
Candidate Variables	All/2^2 to All/2^6
Replacement	With Replacement, Without Replacement

This represents a total of 30 hyper-parameters configurations for a total of 150 tests.

Bagging Test Space

For bagging, the most important hyper-parameter is the number of replication which should improve performance as the number goes up. We therefore use the following configurations to asses this:

Table 26: Bagging Hyper-Parameter Test Space

Hyper-parameter	Range
Number of Trees	2^4 to 2^10

This represents a total of 7 hyper-parameters configurations for a total of 35 tests.

3.3 Results Post Hyper-Parameter Optimization

SVM Results

Because of the number of configurations (484) of SVM tested, we show here only the most effective configurations for each company. First, it is to be noted that multiple configurations yielded the same results, therefore the configurations presented in the below table are ranges rather than unique configurations.

Table 27: SVM Results for Best Hyper-Parameter Configurations (by accuracy) by Company

Company/Parameter	C	Gamma	Kernel	Recall	Precision	Accuracy
Best Buy	2	1	radial	71.64%	96.77%	85.74%
Chapters Indigo	4 to 32	0.5	radial	70.67%	98.03%	85.85%
Hudson Bay	1 to 16	4 to 32	radial	73.93%	93.68%	85.69%
La Senza	1 to 32	2 to 32	radial	70.58%	97.90%	85.77%
Lululemon	2 to 32	1	radial	75.52%	87.41%	83.72%

What we notice immediately is that the radial kernel outperforms all the other ones, this an interesting result indicating to us that the type of data found in social media text responds best to this kind of projection for separating linearly highly engaging statuses vs regular/lowly statuses. As for C, we see that in general a value of 2 would be good as it is part of all the best configuration ranges. For gamma, this could be generalised to trying small values of gamma

could yield good results. Furthermore, we see that precision is very impressive when using SVM, this can have implications regarding business decisions when using the model in practice. This means that when the model makes a positive prediction, it is very likely to be true. Although we should keep in mind that the recall is not exceptional, meaning that we can still have a good status, even if the model says otherwise.

If we take a close look at the full range of results obtained by our grid search, we find that the biggest contributor is kernel choice, this comes to no surprise, as discussed previously, the kernel allows to project data points into a new space that is better suited for linear separation. Here is the best performing configuration by kernel:

Table 28: SVM Results for Best Hyper-Parameter Configurations (by accuracy) by Company for Linear Kernels

Company/Parameter	C	Gamma	Kernel	Recall	Precision	Accuracy
Best Buy	32	0.03125 to 32	linear	74.82%	72.23%	75.15%
Chapters Indigo	1	0.03125 to 32	linear	74.16%	74.33%	76.33%
Hudson Bay	32	0.03125 to 32	linear	64.67%	67.62%	69.44%
La Senza	8	0.03125 to 32	linear	69.17%	66.50%	69.77%
Lululemon	32	0.03125 to 32	linear	79.72%	67.74%	73.19%

Table 29: SVM Results for Best Hyper-Parameter Configurations (by accuracy) by Company for Polynomial (degree 3) Kernels

Company/Parameter	C	Gamma	Kernel	Recall	Precision	Accuracy
Best Buy	0.03125	0.25 to 1	polynomial (3)	75.71%	85.61%	82.84%
Chapters Indigo	0.125	0.5	polynomial (3)	77.23%	88.75%	85.01%
Hudson Bay	0.125	1	polynomial (3)	79.96%	80.28%	81.66%
La Senza	0.125	1	polynomial (3)	76.67%	79.61%	80.24%
Lululemon	0.0625	0.125 to 1	polynomial (3)	83.38%	77.37%	81.10%

Table 30: SVM Results for Best Hyper-Parameter Configurations (by accuracy) by Company for Sigmoid Kernels

Company/Parameter	C	Gamma	Kernel	Recall	Precision	Accuracy
Best Buy	16	0.03125	sigmoid	65.75%	67.34%	69.45%
Chapters Indigo	0.5	0.03125	sigmoid	57.20%	71.71%	69.97%
Hudson Bay	1	0.03125	sigmoid	48.65%	71.32%	67.38%
La Senza	0.5	0.0625	sigmoid	61.30%	61.57%	64.07%
Lululemon	4	0.03125	sigmoid	77.93%	65.62%	71.03%

We see that in general, the sigmoid kernel performs the worst, followed by the linear kernel and then polynomial (degree 3) kernel. We also see that C and Gamma adjustments are highly dependent on individual kernel choice and even on the dataset.

RF Results

Because of the number of configurations (30) of RF tested, we will show here only the most effective configurations for each company.

Table 31: RF Results for Best Hyper-Parameter Configurations (by accuracy) by Company

Company/Parameter	Number of Trees	Candidate Variables	Replacement	Recall	Precision	Accuracy
Best Buy	128	21.9375	FALSE	76.32%	86.27%	81.69%
Chapters Indigo	64	32.3125	TRUE	82.35%	87.33%	86.78%
Hudson Bay	128	52.375	TRUE	83.55%	80.74%	83.32%
La Senza	64	74.5	TRUE	79.16%	80.39%	81.20%
Lululemon	128	27.25	TRUE	86.82%	75.95%	81.79%

It is to be noticed that even though these specific configurations produce the most effective accuracy rates, the range of rates are sensibly very similar across different configurations. We can therefore say that the following results are an upward threshold for performance when using random forests. However, performance does start to go down in a significant fashion when the number of candidate variables at each separation is lower than the total number of variables/32.

Bagging Results

The results for bagging are the following:

Table 32: Bagging Results for Different Hyper-Parameter Configurations (Best Buy)

Number of Trees	Recall	Precision	Accuracy
16	59.81%	88.07%	75.32%
32	61.68%	88.79%	76.43%
64	60.75%	89.45%	76.27%
128	59.50%	89.25%	75.64%
256	61.06%	88.69%	76.11%
512	59.81%	88.89%	75.64%
1024	59.19%	89.20%	75.48%

Table 33: Bagging Results for Different Hyper-Parameter Configurations (Chapters Indigo)

Number of Trees	Recall	Precision	Accuracy
16	83.42%	73.47%	79.14%
32	86.10%	71.66%	78.58%
64	85.38%	73.81%	79.94%
128	86.81%	73.56%	80.18%
256	85.92%	73.59%	79.94%
512	85.92%	74.27%	80.41%
1024	86.10%	73.74%	80.10%

Table 34: Bagging Results for Different Hyper-Parameter Configurations (Hudson Bay)

Number of Trees	Recall	Precision	Accuracy
16	54.48%	86.88%	75.36%
32	59.60%	84.46%	76.45%
64	57.95%	86.14%	76.45%
128	58.14%	86.89%	76.78%
256	57.59%	87.02%	76.61%
512	57.77%	86.81%	76.61%
1024	57.04%	85.95%	76.03%

Table 35: Bagging Results for Different Hyper-Parameter Configurations (La Senza)

Number of Trees	Recall	Precision	Accuracy
16	75.14%	74.01%	76.01%
32	64.44%	75.22%	73.41%
64	87.00%	67.41%	74.22%
128	72.85%	74.41%	75.56%
256	78.20%	71.25%	75.02%
512	78.01%	73.51%	76.54%
1024	76.86%	72.96%	75.83%

Table 36: Bagging Results for Different Hyper-Parameter Configurations (Lululemon)

Number of Trees	Recall	Precision	Accuracy
16	89.45%	67.47%	75.97%
32	89.83%	67.66%	76.22%
64	89.45%	68.05%	76.48%
128	89.45%	67.95%	76.39%
256	89.64%	67.52%	76.05%
512	89.64%	67.71%	76.22%
1024	89.64%	67.71%	76.22%

It seems that for bagging the number of trees does not help enormously in the performance in our case. Looking at accuracy, we see that none of the best scores are from the configuration with 1024 trees, rather the best scores are spread out amongst different configurations in a rather random way. It is to be noted that the difference in accuracy between different configurations is rather small and can mostly be attributed to randomness of the bagging process. We can

therefore say that the following results are an upward threshold for performance when using bagging.

3.4 Comparison of the Different Models

In conclusion, we can see that bagging is no match for random forests and SVM methods. RF performed rather well in our preliminary analysis where hyper-parameters were not optimized. However, SVM is the clear winner in terms of performance after adjusting hyper-parameters. Indeed, SVM produced better accuracy scores for 4 of our 5 companies. It only loss is for the Chapters Indigo dataset by a margin of less than 0.5%, whereas for the other companies, SVM performs almost up to 4% higher than RF.

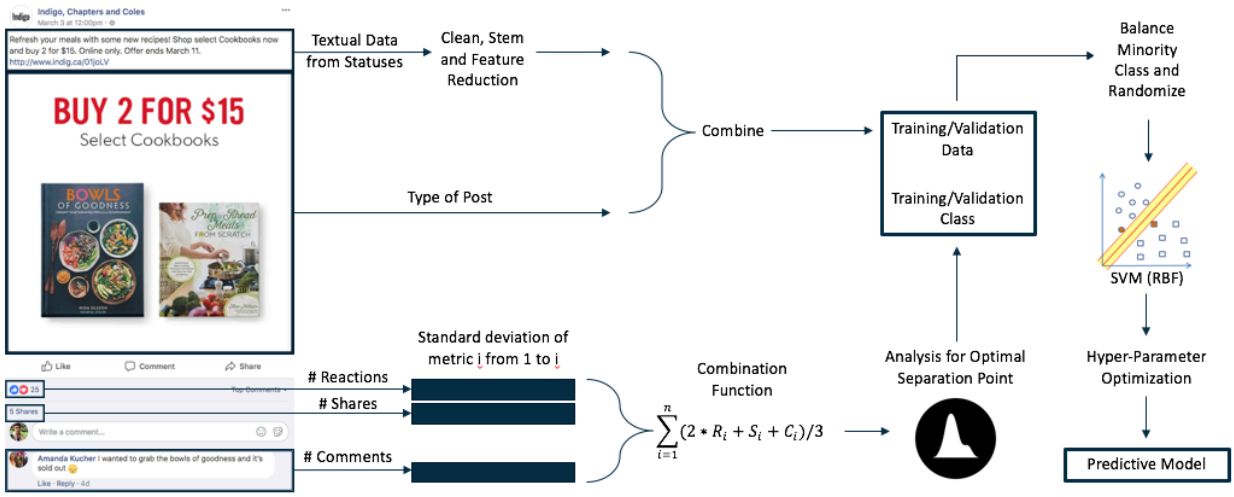
Table 37: Comparison of Top Accuracy Scores by Company and Algorithm

Company	Accuracy SVM	Accuracy RF	Accuracy Bagging
Best Buy	85.74%	81.69%	76.43%
Chapters Indigo	85.85%	86.78%	80.41%
Hudson Bay	85.69%	83.32%	76.78%
La Senza	85.77%	81.20%	76.54%
Lululemon	83.72%	81.79%	76.48%

4. Summary of Analysis

From our analysis, we recommend proceeding in a similar fashion as shown in the below diagram for future applications of proxy learning applied to social media data from the Facebook platform.

Figure 23: Summary of the Proxy Learning Process for Engagement Prediction



To summarise, for any company page we want to analyse, we would first take the number of RSC and then transform them into a standardized score. Following that we would combine them into a single score. From this score distribution, we determine the best separation point for engaging statuses versus non-engaging statuses. This constitutes our training and validation class. For the textual data of statuses, we first proceed to cleaning, stemming and feature reduction. Then we use the type of the status post into the text. This constitutes our training and validation data. From there we balance the minority class and randomize our data. We split it into a training set and validation set. We use the SVM algorithm and adjust its hyper-parameters thus obtaining our predictive model.

5. Business Recommendations

We recommend using the developed model to enhance confidence when posting new statuses. Social media managers can leverage this tool in their day-to-day operations and gain foresight before posting new status messages. The model can be leveraged in two ways. The first is identifying posts which are non-engaging. Because the model has been training on existing statuses, future statuses resembling past statuses which have shown to be significantly non-engaging will be caught by the model. The second is identifying posts which are engaging. The same logic is valid here, future statuses resembling past statuses which have shown to be significantly engaging will also be caught by the model. Therefore, the model should serve to

measure engagement in new statuses by considering that it only has knowledge of past statuses. This means that managers should continue innovating and creating new marketing concepts and campaigns, but also enhance previously popular themes by improving upon them.

Method 3: Investigation of Relevant Features

From our pool of selected features after eliminating features by sparseness, we attempt to gain interpretability by identifying features which are strong contributors to a status being in a high or low engagement class, as defined with the configuration of our earlier analysis. For this we use three methods which allow a degree of interpretability of chosen features, namely naïve Bayes, decision trees and chi-squared selection.

1. Using Naive Bayes

We used our reduced pool of features to create a naïve Bayes model for all 5 companies. We use the naivebayes R library and default settings. The naïve Bayes model gives us the mean contribution for each feature on the probability of the overall status to belong to the positive or negative class. We will observe in the two tables below, the top 5 words which contribute most to positive engagement and the top 5 words that contribute most to negative engagement.

Table 38: Top 5 Positive Contributors of Engagement

Best Buy	Chapters Indigo	Hudson Bay	La Senza	Lululemon
Favorite	Resident (Refers to contest rules)	Canada	Injphoto	Injphoto
Know	Like	Olympics	Shop	Injvideo
Want	Book	Share	Color	Run
Injvideo (Refers to Video Statuses)	Canadian	Win	Arrival	Tank (Refers to tank top)
Song	Contest	Stripe	Hot	Bra

Table 39: Top 5 Negative Contributors of Engagement

Best Buy	Chapters Indigo	Hudson Bay	La Senza	Lululemon
Injlink	Shop	Shop	Dollar	Injlink
Get	Percent	Injlink	Get	Yoga
Check	Save	Fashion	Offer	Ambassador
Store	Offer	Make	Purchase	Injstatus
Deal	End	Style	Percent	Love

For positive words, we see that in general, statuses having multimedia support such as pictures and videos help increase engagement. Contests are also source of engagement in the case of

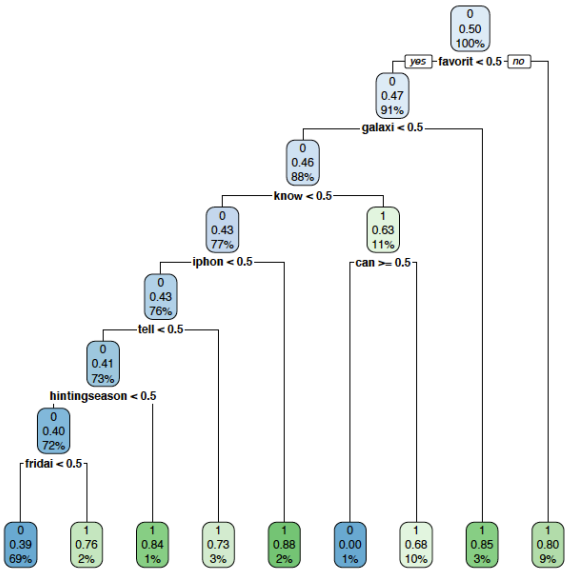
Indigo with two words referring to contests. We also see that words which are descriptive like book, hot, color, tank and bra are good contributors.

For negative words, we see that statuses with links do worse than statuses with pictures and videos. This shows that people prefer interactivity and visual supports. As well words advertising offers, deals and promotions do not do well. Furthermore, words which encourage people to perform an action like get, have the reverse effect, as they are negative contributors to engagement. What is surprising though is that for Lululemon the word “yoga” is a negative contributor. We therefore must wonder for that even though this company’s main offering is athletic apparel has in fact a low number of yoga practitioners amongst its fans.

2. Using Decision Tree

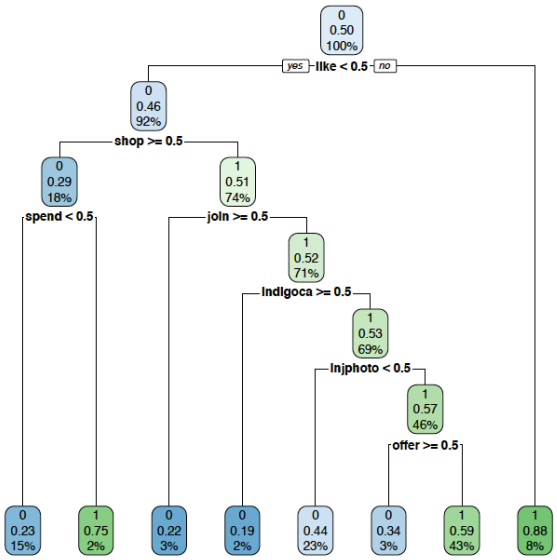
For the decision tree, we use the rpart R library. The settings are for a classification tree with a complexity parameter of 0.01. We obtained graphical decision trees for all five of our companies. It is to be noticed that the word features which appear on the decision trees have been stemming. We look at the relevant features selected for each company and make sense of them. We do not look at the entirety of the tree as we lose a lot of direct interpretability as the levels go down, because we are in fact looking at conditional decisions.

Figure 24: Best Buy Decision Tree



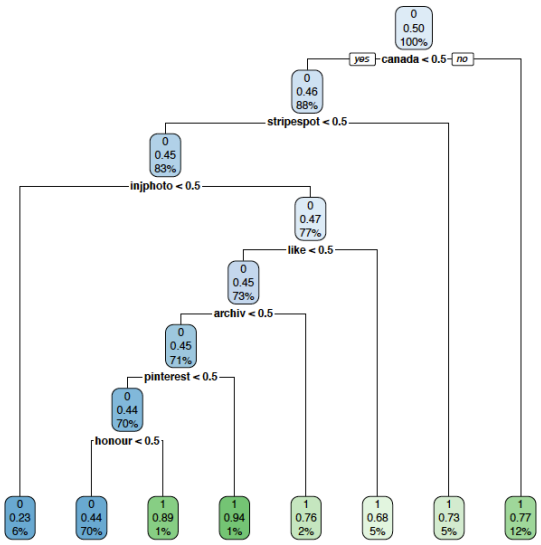
First, let us look at Best Buy's decision tree displayed in the above diagram. First, we see that the word "favorite" is the top differentiator between our binary engagement class. This is consistent with our naïve Bayes analysis. This confirms that people are indeed more inclined to engage with a post mentioning something that is a favorite or can be qualified as favorite. Followed by that is the word "galaxy", which is a reference to the galaxy smartphone by Samsung. It is shown that people tend to like this brand and like engaging with such posts. This is a difference from our naïve Bayes analysis which did not accord as much importance to this word. Finally, we find the word "know" again, followed by "IPhone" which is another cell phone brand which increases engagement. The word "can" seems to have a negative effect and omitting it is better for engagement.

Figure 25: Chapters Indigo Decision Tree



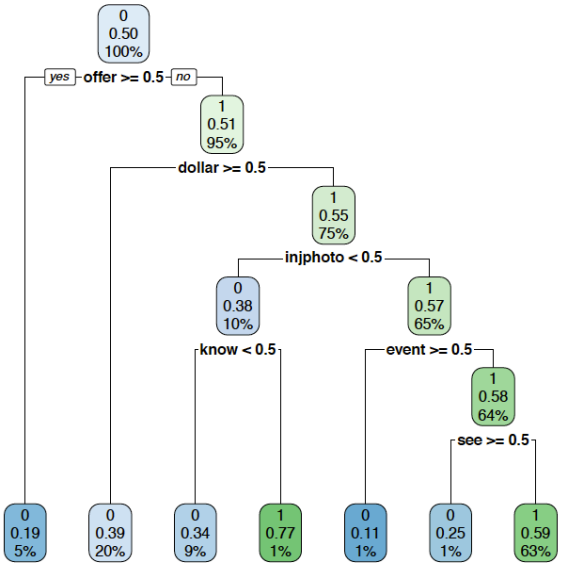
Next, let us look at the decision tree for Chapters Indigo in the above diagram. We see that the word “like” is our most significant feature. This is somewhat consistent with the naïve Bayes as “like” is also an important feature determined by the NB method. Next is the word “shop” which has a negative influence on engagement. This is consistent with our previous NB results. We gain further confidence that words that have a prescriptive aspect have negative effects. The hypothesis being that people do not like being ordered to do an action.

Figure 26: Hudson's Bay Decision Tree



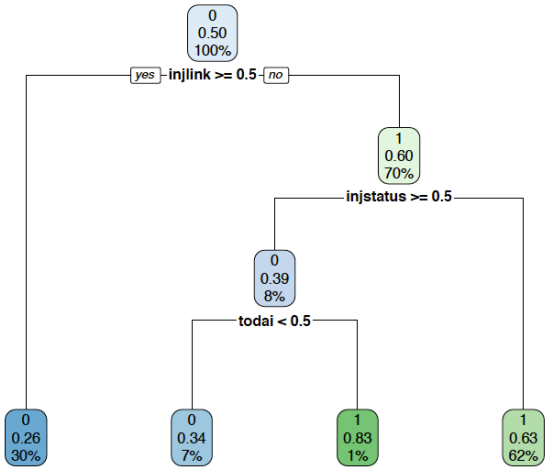
Next, let us look at the decision tree for Hudson's Bay in the above diagram. We see that the three most important factors are "canada", followed by "stripespotting" and "injphoto". This is consistent with our NB analysis. It reveals that people are more inclined to engage with posts with a patriotic aspect. We also see that people like pictures. What is perhaps revealing here is that "stripespotting" is a significant factor when it comes to engagement. This word happens to be a self-defined marketing hashtag by the Hudson's Bay. It shows that the marketing strategy has positive impacts and people are willing to interact with this theme.

Figure 27: La Senza Decision Tree



Next, let us look at the decision tree for La Senza in the above diagram. We see that the words "offer" and "dollar" are the most relevant factors and they contribute negatively to engagement. This is consistent with our NB and reinforces the hypothesis that people do like to be submitted to prescriptive ordering and are rather insensitive to offers with monetary characteristics. The factor contributing the most to positivity are the presence of images. Indeed, people want to engage with interactive elements.

Figure 28: Lululemon Decision Tree



Finally, let us look at the decision tree of Lululemon. With our complexity parameter of 0.01 we only obtained tree relevant factors in our decision tree. Two of the most relevant factors are references to the type of post. Certainly, we see that people are in fact averse to interaction when the status does not contain multimedia elements such as pictures or videos. Indeed, statuses with only text or a link affect engagement negatively. The last word “today” refers to a multitude of statuses with no real thematic connection usually referencing an event or feeling of the day.

3. Using Chi Squared

For the Chi-Squared, we use the *chi.squared* function from the R package *FSelector* with default settings. We obtained for each word an attribution importance. This score does not differentiate between positive attribution to a class versus negative attribution but rather classes features by discriminating power. The words we obtain are therefore a mix of words which both favor and disfavor engagement of a status. We look at the top 5 words for each company by attribution importance in the table below.

Table 40: Top 5 Contributors by Attribution Importance

Best Buy	Chapters Indigo	Hudson Bay	La Senza	Lululemon
Favorite	Like	Canada	Number	Injlink
Song	Residence	Share	Offer	Injphoto
Galaxy	Prohibited	Olympic	Dollar	Injvideo
Know	Rulesdisclaimer	Like	InjPhoto	Ambassador
Store	Void	Post	Canada	City

We cannot comment on whether the shown word are positive or negative contributors from the attribution score. But from the obtained words, we largely see the same group of words as seen in the two first analysis. This serves as a confirmation that these features are indeed important ones when it comes to discriminating between engagement classes.

4. Business Recommendations

We recommend using positive keywords to boost engagement in future posts as they have shown to be positively associated. We also recommend avoiding negative keywords in the future. Furthermore, these keywords also reflect themes that consumers are particularly appreciative of. Using this information, marketing can create future campaigns based on these themes and enhance the probably of success. Additionally, these keywords can serve to construct and enhance existing customer knowledge and develop potential customer personas. Let us give as example, recommendations for our five companies from the obtained keywords.

First for Best Buy, we see that customers are in general appreciative of statuses referring to words which have formulations which include words which promote engagement such as *Favorite*, *Know*, *Want*. *Favorite* refers to posts which the social media manager would talk about a product they deem is a staff favorite. *Know* refers to posts which often convey a sharing of information that would go beyond promotional meaning. For example, knowing the annual household dust quantity and then tying in a promotion for vacuums. *Want* can refer to a variety of messages which have a pull type effect by inciting the customer to want something. *Song* and *Injvideo* are also popular. *Song* refers of course to when the subject is music. *Injvideo* is our code word for the status type video, meaning videos are popular. As for the negative contributors, we see that statuses composed only of links are unpopular. *Get* and *Check* are also unpopular, it seems consumers do not like these verbs as they have a connotation of ordering people to

perform an action. *Deal* is also unpopular as it is perhaps simply too promotional and has only a connotation of advertisement. If we look at the decision tree, we see that the stemmed words *galaxi* and *iphon* which refer to smartphones (Samsung Galaxy and Apple Iphone) are also positive contributors.

For Chapters Indigo, we see that people love the word *Resident* which actually refers to the terms and conditions of contests (eg. You have to be resident of Canada to enter). This reveals us that customers are very engaged by contests and are entering regularly. *Like* also refers to contests as the mechanism to enter the contest is to like the post. The word *Book* is no surprise as Chapters Indigo sells mainly books. *Contest* refers again to the contests they organise. *Canadian* is an interesting word as it indicates that customers like to interact with posts referring to Canadian sentiments. On the negative end, we see that all five words (*Shop, Percent, Save, Offer, End*) paint a strong picture that people do like posts simply referring to a promotion. The tree reveals similar information.

For Hudson Bay, we see that Canadian sentiment related to the Olympics perform very well. Indeed, Hudson Bay is a sponsor of the Canadian Olympic team which would explain the prominence of these posts. *Stripe* is also an interesting word, it refers to Hudson Bay`s in house line of products which have a defining characteristic of donning a green, red, yellow and navy pattern. *Stripespot* from the classification tree also refers to these stripes. As for the negatives, we see a similar picture to Chapters Indigo, customers do not enjoy posts which are simply referring to promotions. The words *Shop, Fashion* and *Style* refer to this as they are often used in conjunction to promote a sale on fashion items such as clothing, bags, shoes, etc.

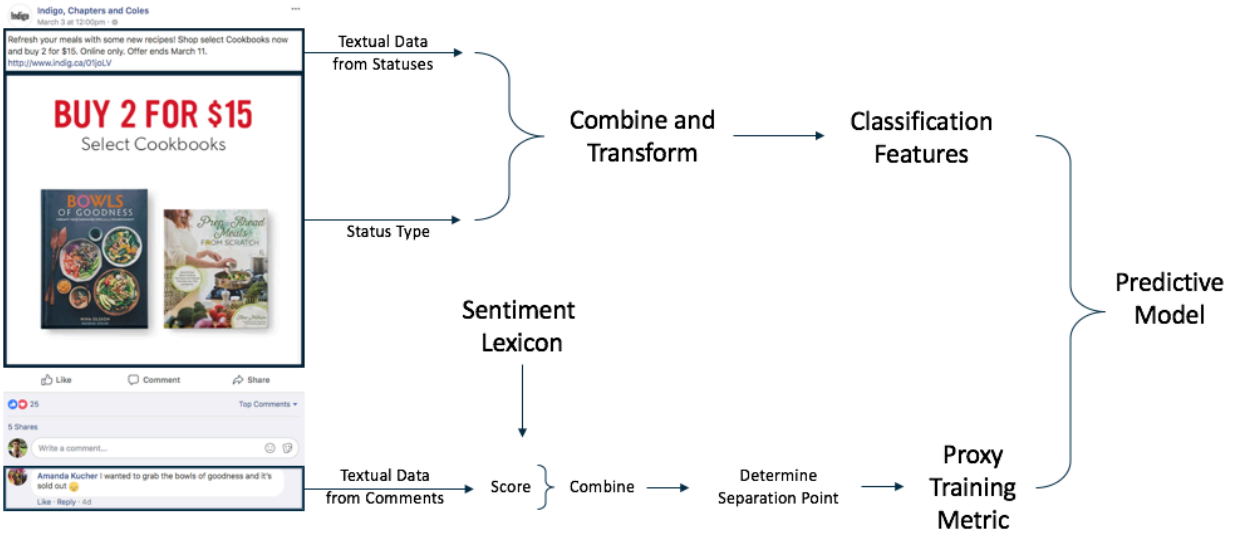
For La Senza, we see that any post with a picture performs very well from the word *injphoto.* We see that promotional posts indicating the arrival of new items are in fact a positive in this case with words like *Shop, Arrival, Color, Hot*. However, when sales are advertised, customers are in fact less likely to engage, the words *Dollar*, *Get, Offer, Purchase*, Percent are all keywords referring to sales (eg. Exclusive offer, get x percent off your next purchase).

For Lululemon, customers like multimedia in the form of pictures and videos and do not like statuses with only links. The products *Tank (Tank Top)* and *Bra* are also appreciated by consumers. The activity of running (*Run*) is also popular among consumers while *Yoga* actually performs poorly. This is a surprising result indicating that the consumer base is perhaps more likely to perform running activities versus yoga. This does make intuitive sense as running as an activity is much more accessible than yoga. Finally, Lululemon`s ambassador (*Ambassador*) program also performs badly.

Method 4: Emotional Proxy Learning

1. Process Description

Figure 29: Summary of Proxy Learning for Emotion Prediction



Because of our positive results with engagement prediction, we decide to extend the idea to sentiment prediction. The process of sentiment prediction or emotional proxy learning can be summarised in the above graph. The concept remains similar to engagement prediction, as we use a proxy to create a training metric. In this case, we use the textual data from the comments to derive a binary class to qualify the emotional response that a status generates.

We proceed by first taking every individual comment and scoring it with a sentiment lexicon. The sentiment scoring for every comment is done either with the embedded function provided by the used R package or by simply taking the average aggregate emotional score for every single comment when no embedded function is provided. For example, if a comment has two words deemed positive and one word deemed negative, the average aggregation will result in a positive score. Once every comment is scored, we group them by status id and output an average score per status. This represents the aggregate emotional score for the status. For example, if a status has 5 comments on it and 4 are positive and 1 is negative, then the aggregate emotional score

will be positive. By giving equal weighting to every comment in a status, we are preventing long comments with multiple words from taking disproportional influence versus shorter comments, when qualifying the emotional response of a status. From the sentiment score of statuses, we can define a threshold to define a binary class of positive versus negative comments.

For the following analysis, we split the data set composed of status text and their corresponding sentiment binary class into 80% training and 20% validation. The same settings are used as in the previous engagement analysis and no hyperparameters tuning is done. We perform a randomization on our dataset prior to selecting training and validation groups. We prefer this method to keeping the data in chronological order because of the nature of Facebook data. If a group of the binary class is underrepresented, we perform random oversampling with return to balance the training data. We then have a 50/50 distribution of the positive and negative class in the training set. Of course, the validation set is never changed.

2. Choice of Lexicons

We test 4 distinct lexicons. First, we use the *SentimentAnalysis* package in R which makes use of the *General Inquirer* lexicon from Harvard. Second, we use the *tidytext* package in R. This package contains three lexicons which we use: *Afinn*, *Bing* and *Nrc*. For all lexicons, we use the positive and negative connotations and attribute a numerical score to each comment.

The *General Inquirer* lexicon contains 1915 positive words and 2291 negative words. We use the *analyzeSentiment* function from *SentimentAnalysis* to attribute a numerical score to each comment.

For the *Afinn*, *Bing* and *Nrc* lexicons, we will set a numerical score to each comment by averaging the polarity of the textual content present in the text. The *Bing* and *Nrc* lexicons have a pool of positive and negative words. For our purposes, we will weight a positive word as 1 and a negative word as -1. The *Afinn* lexicon however, has an embedded score range from -5 to 5 (negative to positive).

3. Separation Point Analysis

We analyze the quantiles of sentiment scores for our five companies for all four lexicons. We observe that in general, sentiment scores are rarely negative. This represents our first insight, when it comes to retail companies, comments are in general neutral and positive rather than negative. This result is rather expected, as it would be abnormal if there were a significant amount of negative comments. Indeed, that would indicate that the business itself has very bad public image and we would question its very survival. Therefore, it is reasonable for businesses to have simply more positive comments as an objective. We take this into account when selecting our separation point for binary class creation.

We will proceed with an arbitrary fixed threshold to set our binary class. This is justified by several factors. First, we observe that the variability of quantile sentiment score between companies can be quite different. Second, unlike engagement, which can be considered as a relative metric, sentiment scores make use of a lexicon, therefore the logic is slightly different. Indeed, it makes sense to characterize engagement as the relative engagement within a company as it is inherently unfair to compare two companies. However, for sentiment score, we make use of uniform lexicons which have pre-set interpretability to them. For example, the Afinn lexicon tells us that 5 represents absolute positivity, while -5 represents absolute negativity. Therefore, it makes little sense to set a relative cut-off point as we want to predict positive sentiments in the comments and those can only be represented with a certain fixed range of positive scores. To put it simply, what can be characterized as a nice comment can't be different because we are on a different Facebook page.

For the GI, Bing and Nrc lexicons which have ranges between -1 and 1, we set the threshold as 0.25. Meaning, we want to separate anything that would be considered bad to neural from the slightly positive to very positive. For the Afinn lexicon which has a range between -5 and 5, we set a threshold of 1.25, the intent here is similar and we have simply adapted the threshold to the range of Afinn scores. Statuses with no comments and therefore no sentiment feedback, are excluded from the analysis.

4. Analysis

We use the same configuration for treating status text as in engagement prediction. We use a SVM model with the radial kernel and adjust its hyper-parameters with a grid search and the same search space for C and Gamma as used previously. We present below the best configuration for each lexicon for each company.

Table 41: SVM Sentiment Prediction Model for Best Buy

Best Buy						
Lexicon	**C**	**Gamma**	**Recall**	**Precision**	**Accuracy**	**Accuracy Rank**
Afinn	0.5	2	87.38%	99.62%	94.14%	2
Bing	2	1	57.08%	96.48%	79.81%	3
GI	0.125	0.25	93.25%	99.32%	96.68%	1
Nrc	1	1	56.64%	96.67%	78.44%	4

Table 42: SVM Sentiment Prediction Model for Chapters Indigo

Chapters Indigo						
Lexicon	**C**	**Gamma**	**Recall**	**Precision**	**Accuracy**	**Accuracy Rank**
Afinn	16	0.25	59.89%	88.26%	76.85%	2
Bing	1	0.125	72.41%	58.33%	59.36%	4
GI	1	1	82.21%	94.52%	89.83%	1
Nrc	1	0.125	66.67%	64.60%	62.61%	3

Table 43: SVM Sentiment Prediction Model for Hudson's Bay

Hudson's Bay						
Lexicon	**C**	**Gamma**	**Recall**	**Precision**	**Accuracy**	**Accuracy Rank**
Afinn	4	1	37.08%	86.27%	66.53%	2
Bing	1	0.0625	95.97%	64.25%	63.69%	3
GI	4	1	50.93%	85.65%	72.97%	1
Nrc	1	0.0625	89.61%	58.96%	58.76%	4

Table 44: SVM Sentiment Prediction Model for La Senza

La Senza						
Lexicon	**C**	**Gamma**	**Recall**	**Precision**	**Accuracy**	**Accuracy Rank**
Afinn	32	2	68.37%	88.47%	81.14%	2
Bing	2	0.03125	73.09%	57.18%	60.15%	4
GI	32	2	74.94%	88.95%	84.41%	1
Nrc	1	2	51.39%	86.08%	73.41%	3

Table 45: SVM Sentiment Prediction Model for Lululemon

Lululemon						
Lexicon	**C**	**Gamma**	**Recall**	**Precision**	**Accuracy**	**Accuracy Rank**
Afinn	1	4	38.36%	78.13%	65.28%	3
Bing	1	0.125	99.57%	67.14%	67.37%	2
GI	1	2	75.67%	88.65%	84.42%	1
Nrc	2	0.0625	77.15%	57.98%	58.62%	4

The first insight that we discover is that the GI lexicon performs in general much better than the other ones. For all five companies, GI had the best accuracy. It also had good precision and recall versus the other lexicons. The Afinn lexicon also performed adequately coming in second for accuracy, four times out of five. However, if we compare its accuracy percentages versus that of the GI lexicon, we find that the difference in performance can be quite large, as much as 19% for Lululemon. We therefore recommend using the GI lexicon when analysing the subject matter of social media text in the context of retail companies. The second thing we discover is that performance can vary enormously between companies. This is somewhat normal as we have set a fixed threshold and we are dealing with different datasets. Finally, we can say that the process of emotional proxy learning is successful as even for Hudson's Bay which only has an accuracy of 72.97%, this process can be used to generate real business value as we can utilize the model to improve customer emotional feedback over time.

5. Business Recommendations

Like the model predicting engagement, we recommend using the developed model to enhance confidence when posting new statuses. Social media managers can leverage this tool in their day-to-day operations and gain foresight before posting new status messages. As with the previous model, we must consider that entirely new concepts might not be accurately predicted by the model as it feeds off historical data. Therefore, managers should continue innovating and creating new marketing concepts and campaigns, but also enhance previously popular themes by improving upon them.

Conclusion

In conclusion, our analysis has shown several possibilities in the realm of social media text analysis. First, we have shown that it is possible to use user feedback scores such as “reactions” as well as other measures and use these scores to derive classes for training machine learning algorithms instead of relying on human classifications. Second, we have shown that it is possible to use the sentiment polarity of user comments as a training a training metric. By doing so we can effectively predict consumer sentiment feedback from status textual content. Third, we have shown the importance of textual content for social media managers when sending out a status. Indeed, statuses in the context of social media have few words and are short in length, therefore to reach actionable levels of accuracy means that special attention regarding posting statuses should be employed going forward. Fourth, we have offered a standard approach and guidelines in determining certain meta-parameters as well as optimizing these same meta-parameters. Fifth, we have shown that SVM performs best amongst our tested algorithms for text-mining in the context of social media analysis.

Going forward, we propose a few ways researchers can expend on proxy learning in the context of social media. First, in the sphere of social media, images and videos are also prominent ways of engaging people. In our current research, we have only focused on textual content, however, it is evident that images and videos are also determinant in the level of engagement. Second, researchers who are particularly knowledgeable in linguistics can map further meta-features. The possibilities are quite large, but as an example, we can detect the number of adjectives, nouns, etc. Lastly, the scope of this kind of analysis can be pushed to different companies by combining similar companies, pooling companies in a same industry in one analysis, etc.

Bibliography

Aggarwal, C. C., & Zhai, C. (2012a). A Survey of Text Clustering Algorithms. Mining Text Data,77-128. doi:10.1007/978-1-4614-3223-4_4

Aggarwal, C. C., & Zhai, C. (2012b). An Introduction to Text Mining. Mining Text Data,1-10. doi:10.1007/978-1-4614-3223-4_1

Agrawal, H., & Kaushal, R. (2016). Analysis of Text Mining Techniques over Public Pages of Facebook. 2016 IEEE 6th International Conference on Advanced Computing (IACC). doi:10.1109/iacc.2016.12

Akaichi, J., Dhouioui, Z., & Perez, M. J. (2013). Text mining Facebook status updates for sentiment classification. 2013 17th International Conference on System Theory, Control and Computing (ICSTCC). doi:10.1109/icstcc.2013.6689032

Andritsos, P., Tsaparas, P., Miller, R. J., & Sevcik, K. C. (2004). LIMBO: Scalable Clustering of Categorical Data. Advances in Database Technology - EDBT 2004 Lecture Notes in Computer Science,123-146. doi:10.1007/978-3-540-24741-8_9

Apté, C., & Weiss, S. (1997). Data mining with decision trees and decision rules. Future Generation Computer Systems,13(2-3), 197-210. doi:10.1016/s0167-739x(97)00021-6

Batrinca, B., & Treleaven, P. C. (2014). Social media analytics: a survey of techniques, tools and platforms. Ai & Society,30(1), 89-116. doi:10.1007/s00146-014-0549-4

Breiman, L. (2001). Random Forests. Mach. Learn. 45, 1 (October 2001), 5-32. DOI: https://doi.org/10.1023/A:1010933404324

Cheng, N., Chandramouli, R., & Subbalakshmi, K. (2011). Author gender identification from text. Digital Investigation,8(1), 78-88. doi:10.1016/j.diin.2011.04.002

Cohen, A. (2017a). Book Stores in Canada. IBISWorld Industry Report 44311. Retrieved from IBISWorld database.

Cohen, A. (2017b). Department Stores in Canada. IBISWorld Industry Report 44311. Retrieved from IBISWorld database.

Cohen, A. (2017c). Lingerie, Swimwear & Bridal Store in the US. IBISWorld Industry Report 44311. Retrieved from IBISWorld database.

Corley, C., Cook, D., Mikler, A., & Singh, K. (2010). Text and Structural Data Mining of Influenza Mentions in Web and Social Media. International Journal of Environmental Research and Public Health,7(12), 596-615. doi:10.3390/ijerph7020596

Croft, W., & Harper, D. (1979). Using Probabilistic Models of Document Retrieval Without Relevance Information. Journal of Documentation,35(4), 285-295. doi:10.1108/eb026683

Duan, K., Keerthi, S., & Poo, A. N. (2003). Evaluation of simple performance measures for tuning SVM hyperparameters. Neurocomputing,51, 41-59. doi:10.1016/s0925-2312(02)00601-x

Friedman, J., Tibshirani, R., & Hastie, T. (2000). Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors). The Annals of Statistics,28(2), 337-407. doi:10.1214/aos/1016120463

Genuer, R., Poggi, J., & Tuleau-Malot, C. (2010). Variable selection using random forests. Pattern Recognition Letters,31(14), 2225-2236. doi:10.1016/j.patrec.2010.03.014

Ghiassi, M., Skinner, J., & Zimbra, D. (2013). Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network. Expert Systems with Applications,40(16), 6266-6282. doi: 10.1016/j.eswa.2013.05.057

Greene, W. H. (2012). Econometric analysis. Boston: Prentice Hall.

Guattery, M. (2017). Consumer Electronics Stores in the US. IBISWorld Industry Report 44311. Retrieved from IBISWorld database.

Hsu, Chih-wei & Chang, Chih-chung & Lin, Chih-Jen. (2003). A Practical Guide to Support Vector Classification Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin.

Hurley, M. (2017). Women's Clothing Stores in the US. IBISWorld Industry Report 44812. Retrieved from IBISWorld database.

Irfan, R., King, C. K., Grages, D., Ewen, S., Khan, S. U., Madani, S. A., . . . Li, H. (2015). A survey on text mining in social networks. The Knowledge Engineering Review,30(02), 157-170. doi:10.1017/s0269888914000277

Jiang, J. (2012). Information Extraction from Text. Mining Text Data,11-41. doi:10.1007/978-1-4614-3223-4_2

Joachims, T. (1998). Text categorization with Support Vector Machines: Learning with many relevant features. Machine Learning: ECML-98 Lecture Notes in Computer Science,137-142. doi:10.1007/bfb0026683

Jurka, T. (2012). Maxent: An R Package for Low-memory Multinomial Logistic Regression with Support for Semi-automated Text Classification. The R Journal,4(1).

Kahya-Özyirmidokuz, E. (2014). Analyzing unstructured Facebook social network data through web text mining. Information Development,32(1), 70-80. doi:10.1177/0266666914528523

Kanungo, T., Mount, D., Netanyahu, N., Piatko, C., Silverman, R., & Wu, A. (2002). An efficient k-means clustering algorithm: Analysis and implementation. IEEE Transactions on Pattern Analysis and Machine Intelligence,24(7), 881-892. doi:10.1109/tpami.2002.1017616

Kaplan, A. M., & Haenlein, M. (2010). Users of the world, unite! The challenges and opportunities of Social Media. Business Horizons,53(1), 59-68. doi: 10.1016/j.bushor.2009.09.003

Kleinbaum, D. G., & Klein, M. (2010). Logistic regression: A self-learning text. New York: Springer.

Kobayashi, M., & Aono, M. (2004). Vector Space Models for Search and Cluster Mining. Survey of Text Mining,103-122. doi:10.1007/978-1-4757-4305-0_5

Liu, B., & Zhang, L. (2012). A Survey of Opinion Mining and Sentiment Analysis. Mining Text Data,415-463. doi:10.1007/978-1-4614-3223-4_13

Mostafa, M. M. (2013). More than words: Social networks' text mining for consumer brand sentiments. Expert Systems with Applications,40(10), 4241-4251. doi: 10.1016/j.eswa.2013.01.019

Murtagh, F. (1983). A Survey of Recent Advances in Hierarchical Clustering Algorithms. The Computer Journal,26(4), 354-359.

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - EMNLP 02. doi:10.3115/1118693.1118704

Porter, M. (1980). An algorithm for suffix stripping. Program,14(3), 130-137. doi:10.1108/eb046814

Pozzi, F. A., Fersini, E., Messina, E., & Liu, B. (2017a). Sentiment analysis in social networks. Cambridge, MA: Morgan Kaufmann.

Pozzi, F., Fersini, E., Messina, E., & Liu, B. (2017b). Challenges of Sentiment Analysis in Social Networks. Sentiment Analysis in Social Networks,1-11. doi:10.1016/b978-0-12-804412-4.00001-2

Sebastiani, F. (2002). Machine learning in automated text categorization. ACM Computing Surveys,34(1), 1-47. doi:10.1145/505282.505283

Singh, J., & Gupta, V. (2016). A systematic review of text stemming techniques. Artificial Intelligence Review,48(2), 157-217. doi:10.1007/s10462-016-9498-2

Sun, J., Wang, G., Cheng, X., & Fu, Y. (2015). Mining affective text to improve social media item recommendation. Information Processing & Management,51(4), 444-457. doi: 10.1016/j.ipm.2014.09.002

Torkkola, K. (2011). Linear discriminant analysis in document classification.

Weiss, S. M. (2005). Text mining: predictive methods for analyzing unstructured information. New York: Springer.

Wu, H., Liu, K., & Trappey, C. (2014). Understanding customers using Facebook Pages: Data mining users feedback using text analysis. Proceedings of the 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design (CSCWD). doi:10.1109/cscwd.2014.6846867

Zafarani, R., Abbasi, M. A., & Liu, H. (2014). Social media mining: an introduction. New York, NY: Cambridge University Press.

Zhang, H., & Li, D. (2007). Naïve Bayes Text Classifier. 2007 IEEE International Conference on Granular Computing (GRC 2007). doi:10.1109/grc.2007.4403192

R Packages, Other Libraries and Lexicons

Ingo Feinerer and Kurt Hornik (2017). TM package: Text Mining Package. R package version 0.7-1. https://CRAN.R-project.org/package=tm

R Core Team (2017). Stats package: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Timothy P. Jurka, Loren Collingwood, Amber E. Boydstun, Emiliano Grossman and Wouter van Atteveldt (2014). RTextTools package: Automatic Text Classification via Supervised Learning. R package version 1.4.2. https://CRAN.R-project.org/package=RTextTools

David Meyer, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel and Friedrich Leisch (2017). e1071 package: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien. R package version 1.6-8. https://CRAN.R-project.org/package=e1071

Andrea Peters and Torsten Hothorn (2017). Ipred package: Improved Predictors. R package version 0.9-6. https://CRAN.R-project.org/package=ipred

Piotr Romanski and Lars Kotthoff (2016). FSelector package: Selecting Attributes. R package version 0.21. https://CRAN.R-project.org/package=FSelector

Silge J and Robinson D (2016). Tidytext package: "tidytext: Text Mining and Analysis Using Tidy Data Principles in R." _JOSS_, *1*(3). doi: 10.21105/joss.00037 (URL: http://doi.org/10.21105/joss.00037), <URL: http://dx.doi.org/10.21105/joss.00037>.

Porter, M., The Porter Stemming Algorithm: https://tartarus.org/martin/PorterStemmer/

Krishna Chaitanya Thota, emoji4j: https://github.com/kcthota/emoji4j

Max Woolf, facebook-page-post-scraper: https://github.com/minimaxir/facebook-page-post-scraper

General Inquirer Lexicon: Stone, P. J., Bales, R. F., Namenwirth, J. Z., & Ogilvie, D. M. (2007). The general inquirer: A computer system for content analysis and retrieval based on the sentence as a unit of information. *Behavioral Science,7*(4), 484-498. doi:10.1002/bs.3830070412

Afinn Lexicon: Finn Årup Nielsen "A new ANEW: Evaluation of a word list for sentiment analysis in microblogs", Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages 718 in CEUR Workshop Proceedings : 93-98. 2011 May. http://arxiv.org/abs/1103.2903

Bing Lexicon: Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. *Proceedings of the 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 04*. doi:10.1145/1014052.1014073

NRC Lexicon: Mohammad, S. M., & Turney, P. D. (2012). Crowdsourcing A Word-Emotion Association Lexicon. *Computational Intelligence,29*(3), 436-465. doi:10.1111/j.1467-8640.2012.00460.x