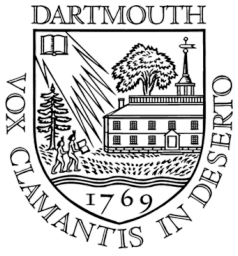
6-1-2020

Predicting Influencer Virality on Twitter

Danah K. Han
Dartmouth College



Dartmouth College
Department of Computer Science

Predicting Influencer Virality on Twitter

Danah Han

Dartmouth Computer Science Technical Report TR2020-888
Advisor: Professor Soroush Vosoughi
Senior Honors Thesis
June 2020

Abstract

The ability to successfully predict virality on Twitter holds great potential as a resource for Twitter influencers, enabling the development of more sophisticated strategies for audience engagement, audience monetization, and information sharing. To our knowledge, focusing exclusively on tweets posted by influencers is a novel context for studying Twitter virality. We find, among feature categories traditionally considered in the literature, that combining categories covering a range of information performs better than models only incorporating individual feature categories. Moreover, our general predictive model, encompassing a range of feature categories, achieves a prediction accuracy of 68% for influencer virality. We also investigate the role of influencer audiences in predicting virality, a topic we believe to be understudied in the literature. We suspect that incorporating audience information will allow us to better discriminate between virality classes, thus leading to better predictions. We pursue two different approaches, resulting in 10 different predictive models that leverage influencer audience information in addition to traditional feature categories. Both of our attempts to incorporate audience information plateau at an accuracy of approximately 61%, roughly a 7% decrease in performance compared to our general predictive model. We conclude that we are unable to find experimental evidence to support our claim that incorporating influencer audience information will improve virality predictions. Nonetheless, the performance of our general model holds promise for the deployment of a tool that allows influencers to reap the benefits of virality prediction. As stronger performance from the underlying model would make this tool more useful in practice to influencers, improving the predictive performance of our general model is a cornerstone of future work.

Contents

Chapter 1

Introduction

1.1 Motivation

Social media is embedded in the daily lives of many Americans. As of 2015, 65% of all American adults used social networking sites [24]. Social media usage is even more widespread among young adults: for Americans aged 18 to 29, 90% use social media [24]. Now more than ever, understanding how to capture the attention and interest of audiences on social media is a valuable resource for monetization and information spreading.

Twitter remains one of the most prominent social media platforms. Twitter has 330 million monthly active users and sees 500 million tweets sent daily [23]. 22% of U.S. adults use Twitter [23]; for reference, 37% of U.S. adults use Instagram [13], and 71% of U.S. adults use Facebook [12]. Many influential individuals and organizations have Twitter accounts, including celebrities, politicians, activists, and companies. Influential individuals and organizations use social media platforms like Twitter to capture new followers, grow their consumer audience, and direct attention to new products and services. Twitter, with its retweeting feature, is particularly well-suited for audience engagement, as it allows stakeholders to spread information at a high speed and scale. Moreover, if a tweet goes viral, the speed and scale of both information spreading and audience engagement can be accelerated. Musicians looking to promote a new album, companies looking to market a new product, and politicians looking to garner support for a new policy proposal are just some examples of influential users that would greatly benefit if their tweets on said subject went viral. Influential Twitter users stand to benefit from viral tweets by seeing, among other results, their promotional goals realized.

1.2 Defining Virality and Influencers

We define virality as a measure of the number of retweets a given tweet receives. In this work, we focus on the degree of virality, as we are interested in the relative

virality of tweets compared to that of other influential users, rather than just raw retweet numbers. Retweets are commonly used as a virality metric in the literature ([15],[11], [22], [20]), as retweeting facilitates the propagation of information to new users and enables information cascades ([16], [35], [21], [34], [28]). In contrast, the other mechanism through which users can interact with a given tweet, known as "favoriting" a tweet, does not necessarily spread the information in a given tweet to new users, such as the followers of a given user who favorited said tweet. Instead, favorites are at most merely a symptom of an object going viral.

We study virality in the contained context of Twitter alone, without factoring in potential spread to other social media platforms (e.g. screenshots of tweets being posted to Instagram or Facebook). We also limit our focus to tweets posted by influencers, defining influencers as Twitter users who are verified or have at least 12,000 followers on the platform.

1.3 Our Goal

We seek to successfully predict the degree of virality for tweets posted by Twitter influencers. The capacity to predict the virality of influencer tweets would allow influencers to (1) better quantify the engagement of their messages, and (2) evaluate the engagement of potential messages before publicly sharing them with their followers. The ability to evaluate tweets before they are posted could allow influencers to test out different versions of their messages and compare virality scores of tweet drafts, thus enabling influencers to select the tweets that achieve the greatest degree of virality before officially posting and sharing said tweets with their folllowers. This capability would allow influencers to utilize more advanced audience engagement, audience monetization, and information sharing strategies.

Retweet prediction has been studied before in the literature, but to our knowledge, it has not been applied to the specific context of predicting virality for influencer tweets. In addition to studying retweet prediction in a novel context, we seek to develop a unique approach to retweet prediction. Retweet prediction models commonly leverage content-based information from tweets, such as hashtag count, or tweet sentiment, to aid in prediction. Others incorporate information about the user that posted the tweet, such as follower count.

A defining feature of our work is expanding the scope of traditionally considered features to include information from users exposed to influencer tweets when predicting retweet amount. We hypothesize that influencers on Twitter have distinct audiences, and that distinct audiences respond to the same content differently. We suspect that incorporating audience information through exposed users will allow us to make better predictions, by enabling us to better discriminate between virality classes. We thus present an approach that is unique with regards to its influencer-specific context and its investigation of the role of influencer audiences in predicting virality.

1.4 Overview of Results

We first approach virality prediction using traditional methods, examining the predictive capability of individual feature categories intrinsic to tweets. We examine content-based features extracted from the message body of tweets, and temporal features (i.e. when a given tweet was posted) individually, building a content-based feature model and a temporal feature model. We find that the feature categories do not hold much predictive power individually: the maximum accuracy of our content-based feature model is just 34.7% while the maximum accuracy of our temporal feature model is only 31.55%. We find that the predictive performance increases slightly if we combine our content-based and temporal features into a single model: the maximum accuracy of our combined temporal-content model is 35.75%, roughly a 1% improvement on our content-based model and a 4% improvement on our temporal model.

We then build a general predictive model that combines all of our feature categories intrinsic to given tweets (i.e. content-based features, temporal features, and influencer account features). The influencer account feature category includes features like follower count that are extracted from the influencer accounts in our dataset. The predictive performance of our general model achieves promising results: with this suite of extracted content, temporal, and account features, we are able to predict the degree of virality for influencer tweets with 68% accuracy. The improved performance of our general model compared to our individual models and combined temporal-content model support previous work that suggest models that combine feature categories covering a range of information perform better than models only incorporating individual feature categories.

With a predictive baseline established through our general model, we investigate our hypothesis that incorporating audience information through exposed users will improve our ability to predict influencer tweet virality. We define exposed users as Twitter users that have been exposed to the influencer tweets of interest (i.e. the influencer tweets for which we seek to make virality predictions). We explore two different approaches to incorporate exposed user information into our general model, building a series of 10 different predictive models that utilize influencer audience information. We find that incorporating exposed user information does not improve the predictive performance of our general model, concluding that we are unable to find experimental evidence to support our claim that incorporating influencer audience information will improve our ability to predict influencer virality. With this result acknowledged, the performance of our general model nonetheless suggests promise for improvement with sophisticated hyperparameter tuning. Improving the accuracy of our general model is a compelling area of future work, as it would enable the successful deployment of a tool that allows influencers to determine the predicted virality of a tweet they would like to post, and receive suggested edits that improve the predicted tweet virality. Stronger predictive performance from the underlying model would make our tool more useful in practice to influencers, hence improving

the predictive performance of our general model is an integral component of proposed future work. We also propose investigating differences in audience behavior at higher levels of granularity as another avenue for future work. Capturing differences between influencer audiences at the influencer category level as well as at the influencer-specific level could provide more informative discriminative information for making virality predictions.

Chapter 2

Related Work

Our primary objective was predicting the degree of virality for influencer tweets. Variants of this task and related tasks have been examined in the literature by the scientific community. An important distinction between our work and the works discussed in this section is that, for making virality predictions, we focus exclusively on tweets posted by Twitter influencers. Unless specified, the following works discussed do not make this distinction in their methodology or their data collection. We first address the related task of extracting features from tweets to examine their predictive power.

2.1 Feature Extraction and Analysis

Prior work has focused on extracting features from tweets and studying their respective impact on virality. Suh et al. examined a collection of features that might impact the retweet proneness of tweets, finding that the use of hashtags and URLs in a tweet have a strong impact on the retweet frequency of that tweet [28]. Can et al. studied multimedia and image-based features (e.g. the distribution of color intensities, the responses of individual object detectors for objects like dog, car, etc.) to predict the retweet count of tweets that share links to images, finding that the inclusion of multimedia and image-based features improves their ability to predict retweet count [4]. Pfitzner et al. analyzed the relationship between the inferred sentiments of tweets and their retweet probability [26]; similarly Mahdavi et al. analyzed sentiment features (e.g. funniness, amount of positive/negative sentiment, etc.) and their impact on the number of retweets a tweet garners, finding that the "socialness" of the tweet content, defined as a measure of how related the tweet is to "public issues and societal problems", is the most informative feature in predicting retweet amount [20]. Mahdavi et al.'s analysis also demonstrates that tweets centering on individual issues and the private life of a given author are not likely to be frequently retweeted [20]. Like Pfitzner et al. and Mahdavi et al., Hansen et al. also examine sentiment effects on retweet frequency, finding that for tweets carrying news-related information, negative sentiment increases virality, but that negative sentiment does not impact virality for tweets

carrying non-news information [8]. Nesi et al. investigated the most representative metrics for predicting the actual number of retweets a tweet will receive, identifying publication time, listed count (the number of Twitter lists a given user is a part of), and mentions count as the most representative features [22]. Jenders et al. provide a comprehensive analysis of different features including user features (e.g. number of user followers), tweet features (e.g. tweet length, number of hashtags in tweet), and the sentiment and emotional divergence of tweets, outlining their respective impact on and correlation with retweet frequency [15]. The findings from these works were useful reference points in our own feature extraction process for the content-based, temporal, and account features we incorporated into our predictive models.

2.2 Retweet Prediction

We now address the literature related to the broader task of retweet prediction, which includes a much wider scope of investigation compared to our specific task of predicting the degree of virality (i.e. the relative retweet amount) for tweets posted by Twitter influencers. Both Jiang et al. and Zhang et al. approached retweet prediction in a general sense, treating retweet prediction as a binary classification problem with two outcomes: retweeted or not retweeted [16], [35]. While Zhang et al. focused on Twitter [35], Jiang et al. conducted their study on Sina Weibo [16], a Chinese social media site similar in form to Twitter with 503 million registered users as of March 2013 [36]. Morchid et al. had a similar approach to Zhang et al. and Jiang et al., training classifiers to detect if a tweet is "massively retweeted" or "low retweeted", defining "massively retweeted" tweets as tweets that received greater than 100 retweets in a short period of time, and "low retweeted" tweets as tweets that received up to 30 retweets [21]. Hong et al. engage with both binary classification (whether or not a tweet will be retweeted) and multiclass classification (estimating the degree of retweets grouped by amount category), notably investigating structural properties of the users' social graph in addition to more commonplace features like message content, temporal information, message metadata, and user metadata [11]. The social graph structural properties investigated by Hong et al. include features such as PageRank, local clustering coefficient, and degree distribution [11]. Jenders et al. explore virality on Twitter, defining virality as whether or not a tweet will receive more retweets than a certain threshold T [15]. Jenders et al. study virality prediction for T values of 50, 100, 500, and 1,000, concluding that viral tweets can indeed be predicted in this context [15]. Liu et al. constructed a two-phase model to predict the number of retweets of messages on Sina Weibo [18]. While not conducted on Twitter, this study is notable because Liu et al. call attention to the importance of grouping users by relative influence for accurate retweet prediction [18]. They propose a two-phase model that first classifies messages into categories by retweet number, then performs regression on each category, achieving better prediction performance than traditional regression without intricate feature extraction [18]. Liu et al. conclude that their two-phase approach addresses the problem of extreme imbalance in retweet number

across different messages in Sina Weibo [18]. Twitter also faces this problem of extreme imbalance. For example, an analysis by Jenders et al. found that the retweet distribution across their tweet dataset follows a Pareto distribution, with only 4% of all tweets receiving more than 50 retweets [15]. Their dataset was composed of over 21.8 million tweets that are not retweets, and 4.2 million tweets that are retweets, collected across roughly 15,000 users [15]. Focusing our scope exclusively on influential users thus reduces the interference of the Twitter platform's imbalance in retweet number on our study.

2.3 Incorporating Audience Information

In addition to specifically considering tweets posted by influential Twitter figures, incorporating influencer audience behavior is a defining feature of our work. We specifically concentrate on Twitter users that have been exposed to given influencer tweets, to include audience behavior for predicting retweet amount. To our knowledge, using this approach to predict virality classes of influencer tweets is novel. Indeed, there is minimal work in the literature investigating the role of user audiences in retweet prediction. We seek to contribute to the literature regarding the role of user audiences in retweet prediction with our approach to predicting virality classes of influencer tweets. We review the most closely related prior work.

Zaman et al. raise the consideration of users exposed to a target tweet for retweet prediction (referring to these users as "retweeters"), but the features they extract from these users to aid in prediction are limited, consisting of just the name of the exposed user, the exposed user's number of followers, and the number of users the exposed user follows [34]. Additionally, Zaman et al. focus on determining the probability that a given tweet will be retweeted by a specific user, rather than on predicting the overall retweet amount of a given tweet [34].

Similar to Zaman et al., Luo et al. present a learning-to-rank framework for retrieving the top followers most likely to retweet a given tweet [19]. While Luo et al. investigate a task distinct from ours, focusing on *who* will retweet a given tweet rather than on the total amount of retweets a given tweet will garner, their work is notable due to the feature categories they consider [19]. Luo et al. consider feature categories similar to our approach, including shared interests between the target tweet creator and a given follower, the retweet history of a given follower, information about a given follower account (e.g. number of posts, number of followers, etc.), and overlap between the posting time of the target tweet creator and a given follower [19]. Luo et al. analyze the ranking performance of each feature category individually, finding the shared interest feature family to be the most informative, and finding posting time overlap to be not useful for their objective of retrieving the top followers likely to retweet a given tweet [19]. Despite this finding regarding posting time overlap, Luo et al. also conclude, similar to Jenders et al., that using all feature categories together achieves the best performance for their task [19], [15].

Zhao et al. propose a retweet prediction model for the Chinese microblog service

Sina Weibo that incorporates information from both direct and indirect followers of the user who created the target post [36]. Zhao et al. define indirect followers as Sina Weibo users that do not directly follow the target post creator [35]. Their model incorporates both the probability that direct followers of the post creator will retweet the post of interest as well as a retweet number estimate for that post from the post creator's indirect followers [36]. Zhao et al. estimate retweet number from indirect followers by calculating the weighted mean number of retweets from direct followers' retweets of the original post, where the weight is derived from a weight function meant to capture the influence of time on retweeting [36]. While Zhao et al. focus on a social media platform besides Twitter, their work is notable as they incorporate the role of user audiences for retweet prediction, albeit on Sina Weibo [36].

2.4 Influence on Twitter

Our work falls directly into the broader categories of retweet prediction and studying virality on Twitter. Related prior work that falls outside of these categories include studies of influence on Twitter, studies of information diffusion on Twitter, and studies of social media sites besides Twitter. We first address studies of influence on Twitter. Rosenman defined influence as "the ability to, through one's own behavior on Twitter, promote activity and pass information to others" [27]. Unlike virality, which we treat as a measure of relative retweet count, Rosenman's influence metric measures behavior change in users that interact with a given target influencer [27]. Rosenman focused exclusively on a group of 60 celebrities, and examined multiple types of influence: retweet-based influence, hashtag and link adoption, word adoption, and emotion adoption, declaring retweet-based influence to be "fundamentally different" from the other types of influence [27]. Rosenman's investigation of retweet-based influence indicates there is a strong correlation between retweets and both follower count and mention count [27]. Between the two, Rosenman concludes that mention count better demonstrates interest in an influencer's message as compared to follower count [27]. Notably, Rosenman also presents a discussion of how some celebrity types become influential on Twitter [27]. Cha et al. also studied influence on Twitter, finding that Twitter influence is not evenly dispersed: instead, "the most influential individuals are many orders of magnitude more influential than the average user" [5], [27]. Additionally, Cha et al. compare indegree, mentions, and retweets as metrics of Twitter influence, presenting a number of important conclusions, including the observation that having a large following does not always result in influence with regards to generating mentions or retweets [5].

2.5 Information Diffusion on Twitter

Information diffusion focuses on the information flow through a social network, rather than on changes in other users' behavior due to a given user (influence) or the amount

of retweets a certain post receives (virality). Pezzoni et al. studied information diffusion on Twitter, investigating the respective effects of network features and user behavior [25]. Pezzoni et al. call attention to the relationship between a message's visibility level and the probability of information dissemination [25]. Hoang et al. addressed "viral" information diffusion, meaning information diffusion that occurs "widely and quickly", using the adoption of specific items like hashtags and URLs as a measure of viral information diffusion [10]. Hoang et al. study a mix of item and user factors that contribute to viral diffusion including user susceptibility, defined as the likelihood of a user to adopt items introduced to her [10]. Yu et al. studied diffusion in the context of predicting the scale of information dissemination, using regression to determine the number of times a tweet is forwarded (retweeted) [33]. Lastly, Vijayan et al. study the early detection of fake news, particularly fake news concerning elections, incorporating network diffusion features in addition to tweet-level features for this task [31].

2.6 Studies on Other Social Media Platforms

Prior work has also focused on social media platforms besides Twitter. We've already discussed a number of studies focusing on the Chinese microblog platform Sina Weibo. Li et al. also studied retweet prediction on Sina Weibo, achieving a prediction accuracy of 86.63% with a Support Vector Machine (SVM) approach [17]. Gao et al. compared behavior between Sina Weibo and Twitter users, analyzing how users access each site and their respective writing styles through textual feature analysis [7]. Gao et al. also compared sentiment polarities and topics of Sina Weibo and Twitter posts, and investigated changes in posting behavior over time such as shifts in user interests [7]. Bakhshi et al. studied photo content engagement on Instagram, analyzing 1 million Instagram images, and finding that photos with human faces are "38% more likely to receive likes and 32% more likely to receive comments [than photos without human faces], even after controlling for social network reach and activity" [2]. Deza et al. also studied viral images, but on the social media site Reddit [6]. Deza et al. introduce classifiers that can predict individual image virality as well as relative virality for pairs of images, achieving a 68.1% prediction accuracy for predicting the relative virality of Reddit image pairs [6]. Aswani et al. studied content virality on Facebook, specifically focusing on the impact of different semantics on Facebook post virality, finding that post virality is positively correlated with promotional offers, direct user mentions, freebies, and direct brand engagement [1]. Lastly, Heimbach et al. examined the likelihood of sharing across different social media platforms including Google+, Facebook, and Twitter, specifically for news articles, finding both common features and evidence for distinct sharing behaviors of each network's users [9].

Chapter 3

Data Collection and Methods

3.1 Data Collection

Our dataset consists of (1) influencer tweets, (2) influencer user objects, and (3) exposed user tweets, where exposed users are defined as Twitter users that have been exposed to influencer tweets of interest (i.e. a given influencer tweet for which we seek to make a virality prediction). Figure 1 depicts our overall dataset breakdown.

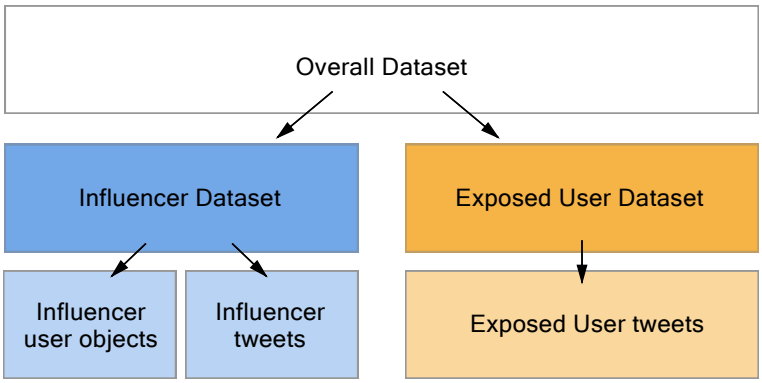


Figure 1: Overall dataset breakdown

3.1.1 Influencer Dataset

Our influencer data collection process began with compiling a list of the major influencer categories on Twitter. We ended up with a list of 14 different categories. Once these categories were established, we collected a list of influencers for each category, defining influencers as users who are either verified or have at least 12,000 followers (with the exception of the "Regular People" category, which was composed of normal Twitter users with follower counts numbering either under one hundred or in the low hundreds). Table 1 lists our 14 different influencer categories and the number of users collected for each category. The majority of influencers in our dataset are American

or based in the United States. For further details about our influencer dataset, see Appendix A.

Influencer Category	Number of Users
Federal Government Figures	24
Celebrities	28
Local/State Government Figures	4
Activists	4
Public Service	3
Bots	2
Musicians	12
Regular People	6
Companies	6
Miscellaneous Topics	15
TV Shows	8
Magazines	4
Sports	18
News	28

Table 1: The number of users per category for which data was collected.

We used Tweepy, a Python library for working with the Twitter API, to collect user and tweet objects, which are objects that contain metadata on Twitter users and tweets, respectively. We gathered 162 user objects in total, one for each of the influencers in Table 1. For each influencer, we also gathered up to 600 tweet objects. For influencers with less than 600 tweets posted to their accounts, the total number of tweets they produced were collected instead. We collected 95,316 influencer tweets in total, with posting dates ranging from January 2007 to March 2019.

As previously mentioned, we chose to focus on predicting degree of virality. Degree of virality translates to the degree of retweets a given tweet has received. Measuring the degree of retweets instead of raw retweet number allows us to more clearly understand virality relative to other influencers in our dataset. Predicting degree of virality also transforms the regression problem of predicting raw retweet numbers into a classification problem of predicting categories of virality (degree of virality/degree of retweets). We created six different classes representing different degrees of virality. Our six classes follow a power law, as previous work in the literature has found that the number of retweets users receive for their tweets follow a power-law distribution [5], which corroborates Jenders et al.'s analysis that the retweet distribution across their dataset follows a Pareto distribution [15]. The six virality classes are detailed below:

(a) **Class 0:** retweet count $= 0$

(b) **Class 1:** $0 <$ retweet count ≤ 10

(c) **Class 2:** $10 <$ retweet count ≤ 100

(d) **Class 3:** 100 < retweet count ≤ 1,000

(e) **Class 4:** 1,000 < retweet count ≤ 10,000

(f) **Class 5:** > 10,000 retweets

We labeled each tweet in our dataset with the appropriate virality class based on its raw retweet number.

3.1.2 Exposed User Dataset

To generate our exposed user tweet dataset, we first calculated the survey sample size for each of our influencers. We used the following formula:

$$sample\_size = \frac{\frac{z^2 \times p(1-p)}{e^2}}{1 + (\frac{z^2 \times p(1-p)}{e^2 N})} \tag{1}$$

where N = population size (i.e. the total follower count for each given influencer), e = margin of error (specifically, the error percentage in decimal form), and z = z-score, to calculate the sample size for each influencer [29]. We used the SurveyMonkey sample size calculator [29] to run these calculations, with a z-score of 1.96 (for a desired confidence level of 95%) and a 3% margin of error. Then, for each influencer, for n of their followers, where n = the influencer's calculated sample size, we collected 100 tweets per follower. If a given follower had less than 100 tweets, we collected the total number of tweets that follower produced. We used the Tweepy library for this data collection, which notably, when retrieving followers, returns followers ordered by when they were added. For example, if an influencer had a sample size of 1,067 followers, Tweepy would return the 1,067 earliest followers of that influencer. We repeated this process for all 162 of our influencers, resulting in an exposed user tweet dataset of 6,743,941 tweets, with posting dates ranging from April 2007 to February 2020.

3.2 Influencer Data Processing

We extracted a number of different features from our influencer dataset, including (1) content-based features from the message body of influencer tweets (e.g. presence or absence of hashtags, tweet sentiment, tweet topic, etc.), (2) temporal features from influencer tweets (i.e. when a given tweet was posted), and (3) account features from the influencer accounts in our dataset. In the following sections, we discuss these features in detail; for a summary of these features, see Appendix B.

3.2.1 Content-Based Features

The majority of the content-based features extracted from our dataset of influencer tweets involved minimal amounts of data processing, only needing to be transformed from their original form into a numerical boolean representation (i.e. 0 or 1).

Binary Content Features

These binary features include *hashtags*, if the influencer tweet contains a hashtag, *symbols*, whether or not the tweet contains a financial symbol, which the Twitter API defines as "a dollar sign ($) followed by a word identifier" (e.g. $MSFT, for discussing Microsoft stock) [3], *urls*, if the tweet includes a URL, *user_mentions*, whether or not the tweet includes a mention (@) to another user (e.g. @LeoDiCaprio), *in_reply_to_user*, if the tweet was in reply to another Twitter user, *in_reply_to_status*, whether or not the tweet was in reply to another tweet, *possibly_sensitive*, if the tweet includes a URL that Twitter has identified as potentially containing sensitive content [30], and *truncated*, which indicates if the tweet text content was truncated.

Tweet Source Content Features

In addition to these binary features, our content-based features also include the *twitter_app_source* and *non_twitter_app_source* features. The *twitter_app_source* and *non_twitter_app_source* features are derived from the 'source' attribute in tweet objects returned by the Twitter API, which we retrieve using Tweepy. The 'source' attribute is a text value that indicates the platform source of a given tweet (e.g. Twitter for Android). We determined that there were three main categories that the 'source' attribute fell into:

(a) Twitter applications: the Twitter application itself, running on different platforms (e.g. "Twitter for iOS", "Twitter for Android", "Twitter Web App", "Twitter for iPad", etc.)

(b) Non-Twitter applications/websites: applications or websites that allow content to be shared directly from their platform onto Twitter via tweets posted by users (e.g. "Instagram", "GeekWire", "WordPress.com", etc.)

(c) Third party applications: applications that post tweets for users, either on a pre-scheduled basis, with bots, or by some other mechanism (e.g. "Cheap Bots, Done Quick!", "CoSchedule", "Twitter Ads Composer", etc.)

A tweet falling into the Twitter applications category is indicated by a *twitter_app_source* feature value of 1, while a tweet falling into the non-Twitter applications/website category is indicated by a *non_twitter_app_source* feature value of 1. A tweet falling into the third party applications category is indicated by feature values of 0 for both the *twitter_app_source* and *non_twitter_app_source* features.

Emoji Content Features

Our content-based features also include binary features that indicate the type of emoji present in a given tweet, specifically *has_joy_emoji*, *has_anger_emoji*, *has_disgust_emoji*,

has_fear_emoji, *has_sadness_emoji*, and *has_surprise_emoji*. These features were created using demoji, a Python library that extracts emojis from text. We first compiled all the emojis present in our dataset of influencer tweets, then categorized each emoji into one of the six basic emotions (joy, anger, disgust, fear, sadness, surprise). For each influencer tweet, we then assigned each of the six previously mentioned emoji features a value of 1 or 0, depending on if that emoji type was present in the given tweet, with 1 indicating present and 0 indicating not present. Each influencer tweet was also assigned a value for the *has_misc_emoji* feature, which indicated if a miscellaneous emoji that did not fall into one of the six basic emotion categories was present in the tweet. If there were no emoji present in the tweet, all of the binary emoji feature values were set to 0.

Punctuation Content Features

In order to extract additional content-based features, the text of each influencer tweet was tokenized (i.e. transformed from a single string into a list of strings called tokens, where each token is an individual word or punctuation mark) using the Natural Language Toolkit (NLTK) library. We then created the *has_question_mark* and *has_exclamation_mark* features by checking each token to determine if it was in the Python punctuation set, and from there, if it was a question or exclamation mark.

Tweet Topic Content Features

We then cleaned the text of each tokenized influencer tweet by removing stopwords using the NLTK stopwords list, removing punctuation using Python's punctuation set, stemming the remaining tokens using NLTK's PorterStemmer, and transforming all tokens to lowercase. Once the influencer tweets were cleaned, we performed Latent Dirichlet Allocation (LDA) topic modeling using the gensim library. We set the number of latent topics to be extracted from the training corpus to 100, and for each influencer tweet, we extracted the topic distribution across the 100 topics to create 100 features, *topic_0_distribution* through *topic_99_distribution*.

We trained our LDA model on a random sample of 100,000 tweets from our overall tweet dataset (6,839,257 tweets in total), which included tweets from both our influencer tweet dataset (95,316 tweets) and our exposed user tweet dataset (6,743,941 tweets). After we trained our LDA model on this 100,000 tweet sample, we saved the model and re-used it for all other instances of LDA topic distribution calculation, so that there would be consistency in our topic distribution results. We also saved the dictionary generated from that same random sample (this dictionary was also used to generate our initial LDA training corpus), so that we could re-use it to generate future corpora while ensuring consistency across LDA distributions. Since our LDA model was trained on a random sample from our overall tweet dataset, it is a general model that we can apply to other accounts for evaluative purposes.

Tweet Sentiment Content Features

Our last content-based feature, the *sentiment_category* feature, was determined by utilizing the VADER (Valence Aware Dictionary and sEntiment Reasoner) Sentiment Analysis library. Using polarity scores determined by VADER's Sentiment Intensity Analyzer, each tweet was labeled a category 1, 2, or 3 (positive, neutral, or negative sentiment). Scores ≥ 0.05 were determined to be positive sentiment, scores < 0.05 and scores > -0.05 were determined to be neutral sentiment, and scores ≤ -0.05 were determined to be negative sentiment. These thresholds were recommended by VADER based on thresholds typically used in the literature [14].

3.2.2 Temporal Features

We also extracted temporal features from our dataset of influencer tweets. Each tweet object returned by the Twitter API through the Tweepy library has a 'created_at' attribute, which is a string that indicates when the tweet was created in Coordinated Universal Time (UTC). We processed this string data into *hour_of_posting*, *day_of_posting*, and *month_of_posting* features by splicing the original 'created_at' string three times to retrieve the respective characters for hour, day, and month. From there, we converted the strings into the appropriate numeric equivalent (0 - 23 for hour, 0 - 6 for day, and 0 - 11 for month). We performed this data processing on all tweets in our influencer dataset.

3.2.3 Account Features

Account features are features concerning the influencer accounts that posted the given tweets of interest. We extracted account features from the metadata provided by the Twitter API's user object data type, which we retrieved using Tweepy. The data processing for account features was minimal.

Our binary account features include *has_location*, whether or not a given user has a location in association with their account, *has_profile_banner_url*, if a given user has customized their profile banner, *has_url*, whether or not a given user has provided a URL in association with their profile, *is_verified*, if a given user is verified on the Twitter platform, and *uses_default_profile*, whether or not a given user has altered the theme or background of their user profile.

Our non-binary, but still numeric, account features include *listed_count*, the number of public Twitter lists a given user is a part of, *tweet_count*, the total number of tweets a given user has posted to their account, *friends_count*, the total number of users a given user is following, *favourites_count*, the total number of tweets a given user has liked in their account lifetime, *followers_count*, the total number of followers a given user has, and *account_age*, the number of years since a given account's creation.

3.3 Exposed User Data Processing

In addition to extracting features from our influencer dataset, we also extracted features from our exposed user dataset. We claimed that incorporating information about the audiences engaging with influencer content may lead to better virality predictions. Following this train of thought, we suspected that incorporating exposed user features would enable us to capture meaningful audience information. We discuss these exposed user features in detail in the following sections, but for a summary of these features, see Appendix B.

3.3.1 Measuring Differences in Topic and Posting Time

We pursued two different approaches to incorporate exposed user information. Our first attempt centered around capturing the differences between influencers and their exposed users with regard to (1) the topics of their tweets and (2) the posting times of their tweets. Our hypothesis was two-fold: (1) individuals would be more likely to retweet influencer tweets featuring topical content similar to their own tweets, and (2) individuals would be more likely to retweet influencer tweets that were posted close to when they were active on the platform.

K-L Divergence Features

To capture differences between the topics of influencer tweets and the tweets of their exposed users, we utilized Kullback-Leibler divergence, henceforth referred to as K-L divergence. In mathematical statistics, K-L divergence is "a measure of how one probability distribution is different from a second, reference probability distribution" [32]. In other words, it measures the deviation between two different distributions. We applied the K-L divergence to influencer and exposed user LDA topic distributions to examine the role tweet content similarity plays in whether or not an exposed user will retweet an influencer tweet.

For each influencer tweet, we calculated the K-L divergence between that tweet's LDA distribution and each of the LDA distributions of that influencer's exposed users. We then took the mean and variance of the resulting K-L scores, thus creating *LDA_KL_mean* and *LDA_KL_variance* features for each influencer tweet. Consider the following example:

> For the influencer @Malala (the prominent activist Malala Yousafzai), we had tweet information in our exposed user dataset for 1,067 of her earliest Twitter followers. Given a tweet by @Malala, we first calculated the LDA distribution for the content of that tweet. Next, for each of her 1,067 associated exposed users, we retrieved that exposed user's LDA distribution, and calculated the K-L divergence between the LDA distribution of @Malala's tweet and that exposed user's LDA distribution. This produces 1,067 distinct K-L scores, one for each exposed user, which we

then take the average and variance of, resulting in a single *LDA_KL_mean* and *LDA_KL_variance* feature for the original tweet sample posted by @Malala.

We repeat this procedure to calculate *LDA_KL_mean* and *LDA_KL_variance* features for all the influencer tweets we wish to make virality class predictions about. Equation 2 lists the formula used to calculate the K-L divergence between a given influencer tweet and each of that influencer's exposed users. $infl_{LDA}$ refers to the LDA distribution of a given influencer tweet, and $exposed_{LDA}$ refers to a given exposed user's LDA distribution.

$$KL = \frac{entropy(infl_{LDA}, exposed_{LDA}) + entropy(exposed_{LDA}, infl_{LDA})}{2} \tag{2}$$

While *entropy* refers to the scipy.stats library's implementation of K-L divergence, which they named *entropy*, we utilize Equation 2 to calculate the K-L divergence in order to have a symmetrical result, as the K-L divergence on it's own is not symmetric, meaning that $entropy(A, B) \neq entropy(B, A)$.

As previously mentioned, each exposed user has its own LDA distribution, represented by $exposed_{LDA}$ in Equation 2. We calculate the LDA distribution of each exposed user as follows:

For each exposed user, we retrieve all of the tweets in our exposed user dataset associated with that user (we have up to 100 tweets for each exposed user). Let n be the total number of tweets in our dataset posted by that exposed user. We combine all n of that exposed user's tweets into one document. We then perform LDA topic modeling on that single document, resulting in a single LDA topic distribution for all of the tweets collected for that exposed user. For LDA topic modeling, we use the same dictionary and model described in Section 3.2.1, subsection *Tweet Topic Content Features*.

Posting Time Difference Features

To examine if individuals are more likely to retweet influencer tweets posted at times they are active on the platform, we studied posting time difference. Posting time difference is the difference between influencer posting times and exposed user posting times. An individual is active on Twitter if they have just posted a tweet, making posting time an appropriate metric for measuring the activity of an exposed user. Calculating the posting time difference between influencers and their exposed users allows us to inspect the role temporal activity plays in whether or not an exposed user will retweet an influencer tweet.

We calculated the posting time difference for each of the following three temporal variables associated with when a tweet is posted: hour, day, and month. Consider the following example for calculating the posting time difference for hour of posting.

> Given a tweet by the influencer @Malala, to calculate the posting time difference for hour of posting, we first consider the exposed users associated with @Malala. @Malala has 1,067 exposed users in our dataset. For each of these 1,067 exposed users, we retrieve that exposed user's average hour of posting. We then take the difference between each exposed user's average hour of posting and the hour that the given tweet by @Malala was posted. This results in a collection of 1,067 distinct difference values, one for each exposed user, which we then take the average and variance of, producing a single *hour_posting_time_difference_mean* and *hour_posting_time_difference_variance* feature for the original tweet sample posted by @Malala.

We perform this calculation for each temporal variable (hour, day, and month) for each influencer tweet. This produces 6 features for each influencer tweet: mean of posting time difference and variance of posting time difference for hour, day, and month.

To calculate the previously mentioned average temporal variable for each exposed user, we first retrieve all of the tweets in our exposed user dataset associated with that exposed user (we have up to 100 tweets for each exposed user), then extract the hour, day, and month of posting from each of the n tweets in our dataset posted by that exposed user. This produces a collection of hours, days, and months for each exposed user. We simply take the average of each temporal variable, resulting in the average hour of posting, day of posting, and month of posting for each exposed user in our dataset.

3.3.2 Leveraging Temporal Information from Retweeters

We elaborate on this point further in Section 4.5.1, but our first approach of measuring differences in tweet topic and posting time did not yield promising results for improving influencer virality predictions. As a result, we launched a second attempt to incorporate meaningful information about the audiences engaging with influencer content.

We decided to shift our source of audience information. Recall that our exposed user dataset was composed of followers of influencers. We assumed that followers of influencers would be exposed to influencer tweets, deeming followers appropriate exposed user candidates. Rather than relying on influencer followers for audience information, we decided to focus on the actual retweeters of influencer tweets instead. Focusing on the retweeters of influencer tweets allowed us to directly collect information from the users that were engaging with influencer content.

We decided to create a temporal retweet background for each influencer by extracting and processing temporal features from each influencer's retweeters. This methodology allowed us to shift our source of audience information while also allowing us to explore at least one of our hypotheses from our initial exposed user approach that

individuals are more likely to retweet influencer tweets posted at times close to when they are active on the platform.

For a given tweet, the Tweepy library can extract information for up to 100 of the first retweets of that tweet. So, for each influencer tweet in our dataset, we used Tweepy to extract the hour, day, and month of the first 100 retweets, collecting up to 300 temporal data points for each influencer tweet (up to 100 each for hour, day, and month). As our influencer dataset had up to 600 tweets for each influencer, each influencer had up to 180,000 temporal data points collected from the retweeters of their tweets.

We used these data points to create three distributions for each influencer, one for hour, day, and month. Each influencer's hour distribution was created by retrieving their associated hour data points, then binning each data point into the appropriate hour bucket (24 buckets). This was repeated for each influencer's day distribution (7 buckets), and each influencer's month distribution (12 buckets). Each of these distributions was then normalized to 1 for every influencer.

These distributions represented the temporal activity of each influencer's retweeters, separated by hour, day, and month. Each distribution reported, for a given influencer, the probability that a retweet would occur at a given hour, day, or month.

Retweeter Hour Activity, Day Activity, and Month Activity

We used these temporal activity distributions to create retweeter activity features. Each influencer tweet in our dataset was assigned a *retweeter_hour_activity* feature, a *retweeter_day_activity* feature, and a *retweeter_month_activity* feature. Each influencer tweet's *retweeter_hour_activity* feature was determined by first extracting that influencer tweet's hour of posting. Once the hour was retrieved, using that influencer's temporal distribution for hour, we retrieved the corresponding probability that a retweet would occur at the extracted hour of influencer tweet posting.

Put more concretely, if an influencer posted a tweet at 8AM UTC, and the influencer's temporal distribution for retweeter hour activity had a value of 0.14 for hour 8 (meaning that the probability for that influencer that a retweet would occur at 8AM UTC was 0.14), the *retweeter_hour_activity* feature for that influencer tweet would be 0.14. The same procedure was followed to determine the *retweeter_day_activity* and *retweeter_month_activity* features, except extracting the respective temporal variable of posting (i.e. day or month of posting) from the influencer tweet and utilizing the influencer's temporal distribution for day or month instead of hour.

Combined Day-Hour Retweeter Activity

In addition to *retweeter_hour_activity*, *retweeter_day_activity*, and *retweeter_month_activity*, we wanted to create a feature to capture the idea that a user's behavior at 5PM UTC on a Friday may be different from their behavior at 5PM UTC on a Monday. In other words, we wanted to capture temporal activity taking into account the combined impact of a given day on a retweeter's behavior at a given hour.

We did this by creating a new temporal distribution that combined hour and day buckets. Rather than having 24 hour buckets and 7 day buckets, this new distribution had 28 day-hour buckets. For each day, we created 4 time bucket categories: (1) midnight UTC up to, but not including 6 AM UTC, (2) 6 AM UTC up to, but not including 12 PM UTC, (3) 12 PM UTC up to, but not including 6 PM UTC, and (4) 6 PM UTC up to, but not including midnight UTC. With 7 days in a week, this resulted in 28 day-hour buckets.

Recall that as our influencer dataset had up to 600 tweets for each influencer, each influencer had up to 180,000 temporal data points collected from the retweeters of their tweets. Like how we previously used this data to create distributions for each influencer's retweeter hour, day, and month activity, we used these data points to create a distribution of retweeters' combined day-hour activity for each influencer. For each influencer, we retrieved the temporal data points collected from the retweeters of their tweets, specifically focusing on the hour and day data points from each retweet, and ignoring the month data points. For each hour and day data point pair (the hour and day of posting extracted from the same retweet), we binned the data point pair into the appropriate day-hour bucket (out of 28 total buckets). This produced a distribution of combined day-hour activity (with 28 total buckets) for each influencer, each of which was then normalized to 1. Each combined day-hour activity distribution represented the probability that a retweet would occur at a given day-hour bucket for a given influencer.

Given these combined day-hour activity distributions, we were then able to create a *combined_day-hour_retweeter_activity* feature for each influencer tweet in our dataset. For each influencer tweet in our dataset, we first extracted the hour and day the tweet was posted. We then determined the appropriate day-hour bucket for this day and hour of posting pair. Once the day-hour bucket of the influencer tweet was determined, we retrieved that influencer's corresponding combined day-hour activity distribution. Using this day-hour distribution, we then determined the corresponding probability for the influencer tweet's day-hour bucket. More concretely, if the influencer posted a tweet at 8PM UTC on a Friday, they would be in day-hour bucket 19 (as midnight UTC up to, but not including 6 AM UTC on Monday is bucket 0). If the influencer's temporal distribution for day-hour buckets had a value of 0.36 for bucket 19 (meaning that the probability that a retweet would occur at 8PM UTC on a Friday was 0.36), the *combined_day-hour_retweeter_activity* feature for that tweet would be 0.36.

Chapter 4

Predicting Virality of Influencer Tweets

4.1 Content-Based Model

In our process of predicting the virality of influencer tweets, we began by first examining the predictive capability of content-based features. Section 3.2.1 further describes our selection of content-based features and the data processing involved. We built a content-based model that predicts virality classes for influencer tweets using only our content-based features. We tested a range of different classifier types for our content-based model. Table 2 lists the accuracy results of our content-based model across classifier types.

Classifier Type	**Accuracy Score**
Dummy Classifier - Most Frequent	28.03%
Multinomial Naive Bayes	34.70%
Random Forest	30.13%
Support Vector Machine (SVM) with Radial Kernel	33.21%

Table 2. Results of our content-based model. Each classifier was trained using 5-fold cross validation.

4.2 Temporal Model

We then studied the predictive capability of temporal features extracted from influencer tweets, building a temporal model that predicts virality classes for influencer tweets using only our temporal features. We elaborate on our selection of temporal features and the data processing involved in Section 3.2.2. Like our content-based model, we tested different classifier types for our temporal model, the results of which are listed in Table 3.

Classifier Type	Accuracy Score
Dummy Classifier - Most Frequent	28.03%
Multinomial Naive Bayes	29.34%
Random Forest	29.0%
SVM with Radial Kernel	31.55%

Table 3. Results of our temporal model. Each classifier was trained using 5-fold cross validation.

4.3 Combined Temporal-Content Model

Recognizing that the performance of our individual content-based and temporal models were low, we sought to examine if the observed predictive performance would improve if we combined our content-based and temporal features into a single model. To study this, we built a combined temporal-content model that predicts virality classes for influencer tweets using both our temporal and content-based features. Like our preceding models, we tested our temporal-content model on a range of classifier types. Table 4 lists the accuracy results of our combined temporal-content model across classifier type.

Classifier Type	Accuracy Score
Dummy Classifier - Most Frequent	28.03%
Multinomial Naive Bayes	33.74%
Random Forest	30.66%
SVM with Radial Kernel	35.75%

Table 4. Results of our combined temporal-content model. Each classifier was trained using 5-fold cross validation.

4.4 General Predictive Model

We then built a general predictive model that combined all of our feature categories (content-based, temporal, and account-based), capturing all of our defined features intrinsic to influencer tweets. Section 3.2.3 describes our influencer account features and the data processing involved. Table 5 lists the accuracy results for our general predictive model across a range of classifier types.

Classifier Type	Accuracy Score
Dummy Classifier - Most Frequent	27.8%
Multinomial Naive Bayes	20.9%
Random Forest	38.6%
SVM with Radial Kernel	68.0%

Table 5. Results of our general predictive model. Each classifier type was trained using 5-fold cross validation.

4.4.1 Discussion

We find that the content-based and temporal feature categories hold limited predictive promise individually: our content-based model achieves a maximum accuracy of 34.7%, while our temporal model achieves a maximum accuracy of just 31.55%. We see that predictive performance slightly increases as we combine feature categories: our combined temporal-content model achieves a maximum accuracy of 35.75%, improving upon the results of our individual content-based and temporal models by 1.05% and 4.2%, respectively. We found this improvement, while minimal, to suggest that including additional feature categories might further boost performance.

Indeed, our general predictive model achieved the best performance by far, with a maximum accuracy of 68%. Our general model outperforms the preceding combined temporal-content model by 32.25%, and our individual content-based and temporal models by 33.3% and 36.45%, respectively. Our results suggest that, with the appropriate classifier type, virality predictions improve with an increasingly diverse range of informative features. Our results are in line with the findings of related prior work, such as that of Jenders et al. [15] and Luo et al. [19], both of which find that using all of their feature families in combination achieves the best performance for their respective tasks. Our results also indicate that incorporating information about the influencer posting a given tweet improves our ability to predict virality for that tweet.

4.5 Measuring Differences in Topic and Posting Time

With a promising predictive baseline established by our general model, we sought to investigate our claim that incorporating audience information via our exposed user dataset would lead to improvements in our virality predictions. As Section 3.3.1 explains, our first attempt to incorporate exposed user information was two-fold, involving (1) capturing the difference between the topics of influencer tweets and the topics of their exposed users, and (2) capturing the posting time difference between influencers and their exposed users.

To test the predictive capability of incorporating the topical content difference between influencers and their exposed users, we built a topical difference model. Our

topical difference model extracts the *LDA_KL_mean* and *LDA_KL_variance* features described in Section 3.3.1 from influencer tweets, then uses these features to make virality class predictions.

To study the predictive capability of incorporating the posting time difference between influencers and their exposed users, we built a posting time difference (PTD) model that extracts the *hour_PTD_mean*, *hour_PTD_variance*, *day_PTD_mean*, *day_PTD_variance*, *month_PTD_mean*, and *month_PTD_variance* features described in Section 3.3.1 from influencer tweets, then makes virality class predictions using these features.

In addition to studying the predictive performance of topical content difference and posting time difference in isolation, we also evaluated the predictive performance of these features in combination with each other, and in combination with our general predictive model. We built a combined topical difference and posting time difference model that makes virality predictions using only the combination of topical difference and posting time difference features. Next, we tested our general model combined with the topical difference model features, then our general model combined with our posting time difference model features. We finally evaluated our general model combined with both our topical difference and posting time difference features.

For each of these models (topical difference, posting time difference, topical and posting time difference combined, and the three expanded general models), multiple classifier types were tested, namely: Random Forest, Most Frequent Dummy classifier, and Support Vector Machine (SVM) with a radial kernel. Each classifier was tested using 5-fold cross validation. Table 6 summarizes the results of these experiments, only listing the results of the best performing classifier type for each model. Across all six models, SVM with a radial kernel was consistently the best performing classifier type.

Model	**Accuracy Score**
Topical Difference	29.74%
Posting Time Difference	50.74%
Topical and Posting Time Difference combined	51.20%
General Model with Topical Difference	61.31%
General Model with Posting Time Difference	60.96%
General Model with Topical and Posting Time Difference	61.11%

Table 6. Best performing classifier results across model types.

4.5.1 Discussion

We find that, when considered in isolation, posting time difference holds more predictive power than topical difference, with our posting time difference model outperforming our topical difference model by 21%. We also find that topical content difference and posting time difference, whether it be in isolation, in combination with each

other, or in combination with our general predictive model, do not yield promising results for improving influencer virality predictions. The performance of our general predictive model actually decreased by about 6.7% to 7% with the inclusion of topical and/or posting time difference features. We did not anticipate this result, particularly given our previous finding from Section 4.4 that predictive performance improved as we incorporated a wider ranger of feature categories. We had anticipated this trend of improvement to continue with the inclusion of audience information via topical content difference and posting time difference, however, we observed a decline in predictive ability. Due to this observed decline, we concluded that the exposed user features we provided in an attempt to capture topical content difference and posting time difference do not provide helpful information to our models.

4.6 Using Temporal Information from Retweeters

Given these findings from Section 4.5, we decided to shift focus to the actual retweeters of influencer tweets, aiming to directly collect audience information from the users that were actually engaging with influencer content, rather than from users we were assuming were exposed to influencer content. Using this retweeter-centric approach, we revisited our earlier hypothesis that individuals are more likely to retweet influencer tweets posted at times close to when they are active on the platform. As Section 3.3.2 explains, we attempted to capture the temporal activity of each influencer's retweeters to investigate if capturing the difference between influencer and retweeter temporal activity would would lead to improvements in our virality predictions.

To study the predictive capability of incorporating the temporal activity of each influencer's retweeters, we built a series of retweeter activity models. For these models, we extract *retweeter_hour_activity*, *retweeter_day_activity*, *retweeter_month_activity*, and *combined_day_hour_retweeter_activity* features from influencer tweets. Section 3.3.2 further describes these features and the data processing involved.

Our suite of retweeter activity models includes four different types of models: (1) classifiers with just *retweeter_hour_activity*, *retweeter_day_activity*, and *retweeter_month_activity* as features, (2) classifiers with just *combined_day_hour_retweeter_activity* as a feature, (3) our general predictive model combined with the *retweeter_hour_activity*, *retweeter_day_activity*, and *retweeter_month_activity* features, and (4) our general predictive model combined with just the *combined_day_hour_retweeter_activity* feature. For each of these models, multiple classifier types were tested, namely: Random Forest, Most Frequent Dummy classifier, and Support Vector Machine (SVM) with a Radial kernel. Table 7 summarizes the results of these experiments, only listing the best performing classifier result per model type, and Table 8 details the best performing classifier type per model. The models were all run with 5-fold cross validation.

Model Type	Accuracy Score
RA for hour, day, and month only	34.3%
combined_day_hour_retweeter_activity only	32.1%
General Model with RA for hour, day, and month	61.13%
General Model with *combined_day_hour_retweeter_activity*	61.15%

Table 7. Best performing model results across classifier types. RA for hour, day, and month refers to the *retweeter_hour_activity*, *retweeter_day_activity*, and *retweeter_month_activity* features. RA stands for **r**etweeter **a**ctivity.

Model	Classifier Type
RA for hour, day, and month only	Random Forest
combined_day_hour_retweeter_activity only	Random Forest
General Model with RA for hour, day, and month only	SVM with Radial Kernel
General Model with *combined_day_hour_retweeter_activity*	SVM with Radial Kernel

Table 8. Best performing classifier type per retweeter activity model.

4.6.1 Discussion

We find that our retweeter-centric approach does not improve the performance of our predictive models. Like our previous exposed user approach detailed in Section 4.5, our retweeter-centric approach decreases the predictive performance of our general model. We conclude that our approach to capturing the difference between influencer and retweeter temporal activity does not provide helpful information to our models. In retrospect, this finding does align with Luo et al.'s conclusion that temporal overlap between a target tweet creator's activity and that creator's follower activity is not helpful for their distinct task of ranking followers most likely to retweet a given tweet [19]. It is unclear how much importance to place on this commonality though, given the distinctions between our two tasks, and given Luo et al.'s additional conclusion that capturing shared interests between a given target tweet creator and follower provided helpful information for their ranking task [19]; in contrast, our attempt to capture topical content difference (Section 4.5) did not yield improvements in our predictions. As Table 9 summarizes, our attempts to incorporate meaningful audience

Model	Accuracy Score
General Predictive Model	68.0%
General Model with Topical Difference	61.31%
General Model with Posting Time Difference	60.96%
General Model with Topical and Posting Time Difference	61.11%
General Model with retweeter activity for hour, day, and month	61.13%
General Model with *combined_day_hour_retweeter_activity*	61.15%

Table 9. Summary of performance across different attempts to predict virality for influencer tweets. The best performing classifier result for each model type is listed.

information from users exposed to/engaging with influencer tweets plateau at an accuracy of approximately 61%. We conclude that we were unable to find experimental evidence to support either our claim that incorporating audience information will improve our ability to predict influencer virality or our two-fold hypothesis that (1) individuals are more likely to retweet influencer tweets posted at times they are active on the platform and (2) that individuals would be more likely to retweet influencer tweets featuring topical content similar to their own tweets.

Chapter 5

Conclusions

5.1 Our Findings

Our overarching goal was to successfully predict the virality of influencer tweets. We achieve promising results to that end, building a general predictive model that predicts degree of virality with an accuracy of 68%. Through our initial virality prediction experiments, we find that combining feature categories covering a range of information performs better than models only incorporating individual feature categories. Indeed, we find that our content-based and temporal feature categories hold limited predictive promise individually: achieving a maximum accuracy of just 34.7% and 31.55%, respectively. Our findings are in line with related prior work, such as that of Jenders et al. [15] and Luo et al. [19], both of which find that using all of their feature families in combination rather than in isolation is critical for achieving strong performance for their respective tasks. We also find that our account-based features provide meaningful information for the task of predicting influencer virality: their inclusion boosts our predictive performance by 32.25%, from 35.75% with our combined temporal-content model to 68% with our general predictive model.

Our primary hypothesis was that influencers on Twitter have distinct audiences, and that distinct audiences respond to the same content differently. We claimed that incorporating influencer audience information would improve our ability to predict influencer tweet virality. We did not find evidence to support this claim. We first focused on capturing topical content difference and posting time difference between influencers and users exposed to their tweets, based on our two-fold hypothesis that (1) individuals would be more likely to retweet influencer tweets featuring topical content similar to their own tweets, and (2) individuals would be more likely to retweet influencer tweets that were posted at times close to when they were active on the platform. However, incorporating differences in tweet topic and posting time did not yield promising results for improving influencer virality predictions. Given the results from our general predictive model (Section 4.4), which suggested virality predictions improve with an increasingly diverse range of features, and given our two-fold hypothesis, we had anticipated that the inclusion of audience information via topical content

difference and posting time difference would improve our predictive results. Instead, we observed a decline in prediction accuracy. The performance of our general predictive model actually decreased by approximately 7% with the inclusion of topical and posting time difference features (both individually, and in combination). Given this observed decline, we conclude that our topical content difference and posting time difference feature categories do not provide helpful discriminative information to our models.

Following this finding, we sought to find a meaningful source information about the audiences engaging with influencer content that would actually improve our predictive results. We decided to utilize actual retweeters of influencer tweets as our source of audience information, instead of followers of influencers that we assumed would be exposed to influencer tweets (i.e. our exposed user dataset). With this new approach, we were able to collect information directly from the users that were actually engaging with influencer content. With our retweeter-centric approach, we decided to re-investigate our hypothesis that individuals are more likely to retweet influencer tweets posted at times they are active on the platform. We created a temporal retweet background for each influencer to capture the difference between the temporal activity of each influencer and their retweeters.

Despite our shift in methodology, our new retweeter-centric approach did not improve the performance of our predictive models. Like our topical content and posting time difference approach, our retweeter-centric approach ended up decreasing the predictive performance of our general model, indicating that capturing the difference between influencer and retweeter temporal activity does not provide helpful discriminative information for virality prediction.

Both of our attempts to incorporate audience information plateaued at an accuracy of approximately 61%. As a result, we conclude that we were unable to find experimental evidence to support either our claim that incorporating audience information will improve our ability to predict influencer virality or our two-fold hypothesis that (1) individuals are more likely to retweet influencer tweets posted at times they are active on the platform and (2) that individuals would be more likely to retweet influencer tweets featuring topical content similar to their own tweets.

5.2 Future Work

It is possible that the collection of more data, either with regard to influencer retweeters or influencer exposed users, may have improved the predictive performance of our respective retweeter and exposed user models that incorporated audience information. Recall that our retweeter activity approach utilized up to 180,000 temporal datapoints per influencer, and that our sample size approach to collecting exposed users (described in Section 3.1.2) collected up to 1,067 exposed users per influencer. That being said, it is also possible that our approaches to incorporating audience information were just fundamentally flawed. A more decisive conclusion on this subject could be reached through future work re-examining these approaches on a much

larger dataset of retweeters and/or exposed users.

Returning to our initial goal of virality prediction for Twitter influencers, we do have a promising baseline established through our general predictive model. The accuracy result of our general model (68%) could likely be improved through more advanced hyperparameter tuning. Improving the accuracy of our model would allow for the successful completion of a tool built to optimize influencer virality. The work regarding this tool is not a part of my thesis, as it was conducted during an independent study (COSC 94) this spring quarter (20S), but in general terms, our tool allows influencers to determine the predicted virality of a tweet they would like to post, and suggests edits that improve the predicted tweet virality. Our tool allows influencers to test out different versions of their tweets prior to posting, enabling influencers to select the tweets that achieve the greatest degree of predicted virality to officially share with their audience. We hope that our tool will allow influencers to improve their strategies for audience engagement, information sharing, and audience monetization (if applicable). However, in order to make our tool more useful in practice to influencers, we would like to boost the predictive performance of the underlying model, which is currently our general predictive model. Hence, a compelling area of future work is improving the predictive performance of our general model.

If hyperparameter tuning does not provide sufficient improvement in prediction accuracy, another option for future work is to investigate differences in audience behavior at higher levels of granularity. Specifically, targeting differences between influencer audiences at the category level (e.g. audiences of musician accounts versus audiences of sports accounts) as well as at the influencer level (e.g. the audience of the YouTuber Jeffree Star versus that of the prominent activist Malala) could prove to hold more promising results for capturing distinctions among influencer audiences and providing informative discriminative information for making virality predictions.

Appendix A

List of Influencers

In the table below, Tweet Count refers to the number of tweets collected per influencer. Exposed User Count refers to the number of exposed users collected for that influencer (i.e. that influencer's sample size).

Influencer	Username	Category	Tweet Count	Exposed User Count
Donald Trump	realDonaldTrump	Federal Govt.	400	1,068
Mark Sanford	MarkSanford	Federal Govt.	597	1,023
Joe Walsh	WalshFreedom	Federal Govt.	598	1,063
Joe Biden	JoeBiden	Federal Govt.	600	1,067
Elizabeth Warren	ewarren	Federal Govt.	600	1,067
Bernie Sanders	BernieSanders	Federal Govt.	600	1,067
Senator account	SenSanders	Federal Govt.	600	1,067
Amy Klobuchar	amyklobuchar	Federal Govt.	600	1,066
Kamala Harris	KamalaHarris	Federal Govt.	600	1,067
Pete Buttigieg	PeteButtigieg	Federal Govt.	600	1,067
Alexandria Ocasio-Cortez	AOC	Federal Govt.	600	1,067
Mikie Sherrill	MikieSherrill	Federal Govt.	600	1,039
Representative account	RepSherrill	Federal Govt.	600	996
Matt Gaetz	mattgaetz	Federal Govt.	600	1,062
Vincente Gonzalez	RepGonzalez	Federal Govt.	600	907
Tammy Baldwin	tammybaldwin	Federal Govt.	600	1,045
Ben Cardin	BenCardinforMD	Federal Govt.	599	915
Maggie Hassan	SenatorHassan	Federal Govt.	600	1,060
Mark Warner	MarkWarner	Federal Govt.	600	1,065
Campaign account	MarkWarnerVA	Federal Govt.	595	991
Mike Pompeo	SecPompeo	Federal Govt.	600	1,066
Stephanie Grisham	PressSec	Federal Govt.	600	1,067
Mick Mulvaney	MickMulvaneyOMB	Federal Govt.	252	1,037
GOP	GOP	Federal Govt.	600	1,067
Ryan Reynolds	VancityReynolds	Celebrities	600	1,068
Mindy Kaling	mindykaling	Celebrities	600	1,068
Leonardo DiCaprio	LeoDiCaprio	Celebrities	600	1,068

Lana Condor	lanacondor	Celebrities	600	1,065
Timothée Chalamet	RealChalamet	Celebrities	574	1,066
Kim Kardashian	KimKardashian	Celebrities	600	1,068
Kylie Jenner	KylieJenner	Celebrities	594	1,068
NeNe Leakes	NeNeLeakes	Celebrities	596	1,067
Snooki	snooki	Celebrities	591	1,067
Stephen Colbert	StephenAtHome	Celebrities	600	1,068
Show account	colbertlateshow	Celebrities	600	1,066
Ellen	TheEllenShow	Celebrities	600	1,068
Jimmy Kimmel	jimmykimmel	Celebrities	600	1,068
Show account	JimmyKimmelLive	Celebrities	598	1,067
Dr. Phil	DrPhil	Celebrities	600	1,067
Show account	TheDrPhilShow	Celebrities	600	1,056
Dr. Oz	DrOz	Celebrities	600	1,067
Chrissy Teigen	chrissyteigen	Celebrities	600	1,068
Ashley Graham	ashleygraham	Celebrities	600	1,064
Karlie Kloss	karliekloss	Celebrities	600	1,067
Elon Musk	elonmusk	Celebrities	600	1,068
Tim Cook	tim_cook	Celebrities	600	1,068
Jeffree Star	JeffreeStar	Celebrities	598	1,067
James Charles	jamescharles	Celebrities	597	1,067
Jenna Marbles	Jenna_Marbles	Celebrities	600	1,067
Grace Helbig	gracehelbig	Celebrities	600	1,067
Lele Pons	lelepons	Celebrities	594	1,067
King Bach	KingBach	Celebrities	600	1,067
Bill DeBlasio	BilldeBlasio	Local/State Govt.	600	1,061
Ben Allen	BenAllenCA	Local/State Govt.	599	953
London Breed	LondonBreed	Local/State Govt.	600	1,043
Bill Weld	GovBillWeld	Local/State Govt.	600	1,055
Greta Thunberg	GretaThunberg	Activists	600	1,066
Joshua Wong	joshuawongcf	Activists	600	1,067
Malala Yousafzai	Malala	Activists	600	1,037
Gloria Steinem	GloriaSteinem	Activists	600	1,067
MTA	MTA	Public Service	600	1,067
NYC Public Schools	NYCSchools	Public Service	600	1,061
CalTrain	Caltrain	Public Service	600	1,060
Tiny Care Bot	tinycarebot	Bots	600	1,060
Magic Realism Bot	MagicRealismBot	Bots	600	1,058
J Cole	JColeNC	Musicians	600	1,068
Lil Wayne	LilTunechi	Musicians	600	1,068
Drake	Drake	Musicians	590	1,068
Florida Georgia Line	FLAGALine	Musicians	599	1,067
Thomas Rhett	ThomasRhett	Musicians	600	1,067
Kacey Musgraves	KaceyMusgraves	Musicians	596	1,066
Yo-Yo Ma	YoYo_Ma	Musicians	598	1,041
Lizzo	lizzo	Musicians	599	1,067
Halsey	halsey	Musicians	600	1,068
Britney Spears	britneyspears	Musicians	598	1,068
Justin Timberlake	jtimberlake	Musicians	600	1,068
Ariana Grande	ArianaGrande	Musicians	600	1,068
rebbeca	brleman99	Regular People	224	32
andy zajac	ndyzajac	Regular People	509	193
evan	eshawd	Regular People	562	189

Zoë	zoej_anderson	Regular People	479	58
Ashleigh Brady	Ashleigh1225	Regular People	554	148
Tyler Thierry	tylerthierry7	Regular People	546	223
Pepsi	pepsi	Companies	600	1,067
Samsung Mobile	SamsungMobile	Companies	600	1,068
Anastasia Beverly Hills	ABHcosmetics	Companies	600	1,066
JetBlue Airways	JetBlue	Companies	600	1,067
JetBlue Cheeps	JetBlueCheeps	Companies	600	1,065
Riot Games	riotgames	Companies	600	1,067
Comedy Account	caucasianjames	Miscellaneous Topics	600	1,066
Comedy Account	prasejeebus	Miscellaneous Topics	598	1,019
Comedy Account	mistachrish	Miscellaneous Topics	596	1,064
The Onion	TheOnion	Miscellaneous Topics	600	1,068
Inspirational Quotes	unlockmindset	Miscellaneous Topics	600	1,067
Girl Notes	SmiIe	Miscellaneous Topics	599	1,067
Tasty	tasty	Miscellaneous Topics	600	1,067
Goodful	goodful	Miscellaneous Topics	600	1,040
UberFacts	UberFacts	Miscellaneous Topics	600	1,068
Muscle Strength	Muscle_Strength	Miscellaneous Topics	599	1,050
bodybuilding.com	Bodybuildingcom	Miscellaneous Topics	600	1,066
Luxury and Travel Blog	melandjake99	Miscellaneous Topics	600	1,008
Parenting & Money Saving	Katykicker	Miscellaneous Topics	600	984
Koreaboo	Koreaboo	Miscellaneous Topics	600	1,067
Catholicism for teens	LifeTeen	Miscellaneous Topics	600	1,043
The Bachelor	BachelorABC	TV Shows	596	1,066
The Bachelorette	BachelorettеABC	TV Shows	597	1,066
Bachelor in Paradise	BachParadise	TV Shows	594	1,065
Kardashians on E!	KUWTK	TV Shows	600	1,065
Modern Family	ModernFam	TV Shows	598	1,067
Riverdale	CW_Riverdale	TV Shows	599	1,067
Bravo	BravoTV	TV Shows	600	1,067
ABC	abcnetwork	TV Shows	600	1,066
The New Yorker	NewYorker	Magazines	600	1,067
The Economist	TheEconomist	Magazines	600	1,068
Vogue	voguemagazine	Magazines	600	1,068
GQ	GQMagazine	Magazines	600	1,067
Kevin Durant	KDTrey5	Sports	600	1,068
Steph Curry	StephenCurry30	Sports	600	1,068
Lebron James	KingJames	Sports	600	1,068
Kobe Bryant	kobebryant	Sports	600	1,068
Clayton Kershaw	ClaytonKersh22	Sports	414	1,065
Alex Rodriguez	AROD	Sports	600	1,067
Drew Brees	drewbrees	Sports	600	1,067
Tom Brady	TomBrady	Sports	312	1,066
Cam Newton	CameronNewton	Sports	595	1,067
Russell Wilson	dangerusswilson	Sports	600	1,067
Rafael Nadal	RafaelNadal	Sports	600	1,068
Serena Williams	serenawilliams	Sports	600	1,068
Roger Federer	rogerfederer	Sports	600	1,068
Novak Djokovic	DjokerNole	Sports	600	1,067
NY Yankees	Yankees	Sports	600	1,067
Seattle Seahawks	Seahawks	Sports	600	1,067
LA Dodgers	Dodgers	Sports	600	1,067

Miami Heat	MiamiHEAT	Sports	600	1,067
CNN Breaking News	cnnbrk	News	600	1,068
CNN	cnn	News	600	1,068
MSNBC	MSNBC	News	600	1,067
CBS	CBSNews	News	600	1,067
Fox News	FoxNews	News	600	1,068
NY Post	nypost	News	600	1,067
Boston Globe	BostonGlobe	News	600	1,066
The Washington Post	washingtonpost	News	600	1,068
Breibart News	BreitbartNews	News	600	1,067
Paul Joseph Watson	PrisonPlanet	News	597	1,066
The Atlantic	TheAtlantic	News	600	1,067
Jake Tapper	jaketapper	News	600	1,067
Chris Cuomo	ChrisCuomo	News	600	1,067
Megyn Kelly	megynkelly	News	593	1,067
Tomi Lahren	TomiLahren	News	594	1,067
Sarah Huckabee Sanders	SarahHuckabee	News	600	1,065
E! News	enews	News	600	1,068
Daily Mail	MailOnline	News	600	1,067
Spotify	Spotify	News	600	1,067
SoundCloud	SoundCloud	News	599	1,067
Apple Music	AppleMusic	News	600	1,067
WIRED	WIRED	News	600	1,068
GeekWire	geekwire	News	600	1,058
Reuters Tech News	ReutersTech	News	600	1,061
Wall Street Journal	WSJ	News	600	1,068
Business Insider	businessinsider	News	600	1,067
Buzzfeed	BuzzFeed	News	600	1,067
Elite Daily	EliteDaily	News	600	1,062

Appendix B

Feature List

B.1 Content-based features

Feature	Minimum Value	Maximum Value
hashtags	0	1
symbols	0	1
urls	0	1
user_mentions	0	1
in_reply_to_user	0	1
in_reply_to_status	0	1
possibly_sensitive	0	1
truncated	0	1
sentiment_category	1	3
twitter_app_source	0	1
non_twitter_app_source	0	1
has_joy_emoji	0	1
has_anger_emoji	0	1
has_disgust_emoji	0	1
has_fear_emoji	0	1
has_sad_emoji	0	1
has_surprise_emoji	0	1
has_misc_emoji	0	1
has_exclamation_mark	0	1
has_question_mark	0	1
topic_0_distribution	0.000454546	0.752498806
topic_1_distribution	0.000454546	0.669999301
topic_2_distribution	0.000454546	0.752498746
topic_3_distribution	0.000454546	0.669999719
topic_4_distribution	0.000454546	0.752499759
topic_5_distribution	0.000454546	0.66999954
topic_6_distribution	0.000454546	0.66999948
topic_7_distribution	0.000454546	0.669999838

topic_8_distribution	0.000454546	0.66999954
topic_9_distribution	0.000454546	0.669999659
topic_10_distribution	0.000454546	0.01
topic_11_distribution	0.000454546	0.66999954
topic_12_distribution	0.000454546	0.752499461
topic_13_distribution	0.000454546	0.858571291
topic_14_distribution	0.000454546	0.752499282
topic_15_distribution	0.000454546	0.669999182
topic_16_distribution	0.000454546	0.752499044
topic_17_distribution	0.000454546	0.801995277
topic_18_distribution	0.000454546	0.75249958
topic_19_distribution	0.000454546	0.752499223
topic_20_distribution	0.000454546	0.75249958
topic_21_distribution	0.000454546	0.669999003
topic_22_distribution	0.000454546	0.752499163
topic_23_distribution	0.000454546	0.504998744
topic_24_distribution	0.000454546	0.801999867
topic_25_distribution	0.000454546	0.669999421
topic_26_distribution	0.000454546	0.75249958
topic_27_distribution	0.000454546	0.80199945
topic_28_distribution	0.000454546	0.752498507
topic_29_distribution	0.000454546	0.75249958
topic_30_distribution	0.000454546	0.669999838
topic_31_distribution	0.000454546	0.752499282
topic_32_distribution	0.000454546	0.669999599
topic_33_distribution	0.000454546	0.752499819
topic_34_distribution	0.000454546	0.858570039
topic_35_distribution	0.000454546	0.669999599
topic_36_distribution	0.000454546	0.66999948
topic_37_distribution	0.000454546	0.752499521
topic_38_distribution	0.000454546	0.669999421
topic_39_distribution	0.000454546	0.669999719
topic_40_distribution	0.000476222	0.66999948
topic_41_distribution	0.000454546	0.752498269
topic_42_distribution	0.000454546	0.669999123
topic_43_distribution	0.000454546	0.669998765
topic_44_distribution	0.000454546	0.669999063
topic_45_distribution	0.000454546	0.752499759
topic_46_distribution	0.000454546	0.834999502
topic_47_distribution	0.000454546	0.669999659
topic_48_distribution	0.000454546	0.669999778
topic_49_distribution	0.000454546	0.752499878
topic_50_distribution	0.000454546	0.66999948
topic_51_distribution	0.000454546	0.669999301
topic_52_distribution	0.000454546	0.669999123
topic_53_distribution	0.000454546	0.752499819
topic_54_distribution	0.000454546	0.7524997
topic_55_distribution	0.000454546	0.669999421
topic_56_distribution	0.000454546	0.66999954
topic_57_distribution	0.000454546	0.752498627
topic_58_distribution	0.000454546	0.752499461
topic_59_distribution	0.000454546	0.669999182
topic_60_distribution	0.000454546	0.669998527

topic_61_distribution	0.000454546	0.669998944
topic_62_distribution	0.000454546	0.752499402
topic_63_distribution	0.000454546	0.752499819
topic_64_distribution	0.000476222	0.669999838
topic_65_distribution	0.000454546	0.733083785
topic_66_distribution	0.000454546	0.66999954
topic_67_distribution	0.000454546	0.752499342
topic_68_distribution	0.000454546	0.669999897
topic_69_distribution	0.000454546	0.7524997
topic_70_distribution	0.000454546	0.669999421
topic_71_distribution	0.000454546	0.669998109
topic_72_distribution	0.000454546	0.669999897
topic_73_distribution	0.000454546	0.669999599
topic_74_distribution	0.000454546	0.752499759
topic_75_distribution	0.000454546	0.669999659
topic_76_distribution	0.000454546	0.669999003
topic_77_distribution	0.000454546	0.66999948
topic_78_distribution	0.000454546	0.75249809
topic_79_distribution	0.000454546	0.669997156
topic_80_distribution	0.000454546	0.66999954
topic_81_distribution	0.000454546	0.669999838
topic_82_distribution	0.000454546	0.801999927
topic_83_distribution	0.000454546	0.669999421
topic_84_distribution	0.000454546	0.669997454
topic_85_distribution	0.000454546	0.504998803
topic_86_distribution	0.000454546	0.752499402
topic_87_distribution	0.000454546	0.752499163
topic_88_distribution	0.000454546	0.669998527
topic_89_distribution	0.000454546	0.752499521
topic_90_distribution	0.000454546	0.75249958
topic_91_distribution	0.000454546	0.572859406
topic_92_distribution	0.000454546	0.752498865
topic_93_distribution	0.000476222	0.834999382
topic_94_distribution	0.000454546	0.66999954
topic_95_distribution	0.000454546	0.752499521
topic_96_distribution	0.000454546	0.669998944
topic_97_distribution	0.000454546	0.601999938
topic_98_distribution	0.000454546	0.669999003
topic_99_distribution	0.000454546	0.669999659

B.2 Temporal features

Feature	Minimum Value	Maximum Value
hour_of_posting	0	23
day_of_posting	0	6
month_of_posting	0	11

B.3 Account features

Feature	Minimum Value	Maximum Value
followers_count	32	78,719,258
listed_count	0	182,945
tweet_count	224	593,031
friends_count	0	380,815
favourites_count	0	139,704
has_location	0	1
account_age	1	13
has_profile_banner_url	0	1
has_url	0	1
is_verified	0	1
uses_default_profile	0	1

B.4 Topical Content Difference features (Section 3.3.1)

Feature	Minimum Value	Maximum Value
LDA_KL_mean	0.760184	4.77036
LDA_KL_variance	0.0145822	2.74149

B.5 Posting Time Difference features (Section 3.3.1)

Feature	Minimum Value	Maximum Value
hour_posting_time_difference_mean	-13.6837	13.1303
hour_posting_time_difference_variance	2.95683	28.6499
day_posting_time_difference_mean	-3.49248	3.26912
day_posting_time_difference_variance	0.0856763	1.6124
month_posting_time_difference_mean	-5.99769	9.3622
month_posting_time_difference_variance	1.09812	13.2298

B.6 Retweeter Activity features (Section 3.3.2)

Feature	Minimum Value	Maximum Value
retweeter_hour_activity	0.0	0.5
retweeter_day_activity	0.0	0.762
retweeter_month_activity	0.0	1.0
combined_day_hour_retweeter_activity	0.0	0.595

Bibliography

[1] Reema Aswani, Arpan Kumar Kar, Shalabh Aggarwal, and P Vigneswara Ilavarsan, *Exploring content virality in facebook: A semantic based approach*, Conference on e-Business, e-Services and e-Society, Springer, 2017, pp. 209–220.

[2] Saeideh Bakhshi, David A Shamma, and Eric Gilbert, *Faces engage us: Photos with faces attract more likes and comments on instagram*, Proceedings of the SIGCHI conference on human factors in computing systems, 2014, pp. 965–974.

[3] Twitter Developer Blog, *Symbols entities for tweets*.

[4] Ethem F Can, Hüseyin Oktay, and R Manmatha, *Predicting retweet count using visual cues*, Proceedings of the 22nd ACM international conference on Information & Knowledge Management, 2013, pp. 1481–1484.

[5] Meeyoung Cha, Hamed Haddadi, Fabricio Benevenuto, and Krishna P Gummadi, *Measuring user influence in twitter: The million follower fallacy*, fourth international AAAI conference on weblogs and social media, 2010.

[6] Arturo Deza and Devi Parikh, *Understanding image virality*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1818–1826.

[7] Qi Gao, Fabian Abel, Geert-Jan Houben, and Yong Yu, *A comparative study of users' microblogging behavior on sina weibo and twitter*, International Conference on User Modeling, Adaptation, and Personalization, Springer, 2012, pp. 88–101.

[8] Lars Kai Hansen, Adam Arvidsson, Finn Årup Nielsen, Elanor Colleoni, and Michael Etter, *Good friends, bad news-affect and virality in twitter*, Future information technology, Springer, 2011, pp. 34–43.

[9] Irina Heimbach, Benjamin Schiller, Thorsten Strufe, and Oliver Hinz, *Content virality on online social networks: Empirical evidence from twitter, facebook, and google+ on german news websites*, Proceedings of the 26th ACM Conference on Hypertext & Social Media, 2015, pp. 39–47.

[10] Tuan-Anh Hoang and Ee-Peng Lim, *Virality and susceptibility in information diffusions*, Sixth international AAAI conference on weblogs and social media, 2012.

[11] Liangjie Hong, Ovidiu Dan, and Brian D Davison, *Predicting popular messages in twitter*, Proceedings of the 20th international conference companion on World wide web, ACM, 2011, pp. 57–58.

[12] Hootsuite, *33 facebook stats that matter to marketers in 2020*.

[13] ———, *37 instagram stats that matter to marketers in 2020*.

[14] C.J. Hutto, *Vader-sentiment-analysis*.

[15] Maximilian Jenders, Gjergji Kasneci, and Felix Naumann, *Analyzing and predicting viral tweets*, Proceedings of the 22nd international conference on world wide web, ACM, 2013, pp. 657–664.

[16] Bo Jiang, Jiguang Liang, Ying Sha, Rui Li, Wei Liu, Hongyuan Ma, and Lihong Wang, *Retweeting behavior prediction based on one-class collaborative filtering in social networks*, Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, 2016, pp. 977–980.

[17] Ying-le Li, Hong-tao Yu, and Li-xiong Liu, *Predict algorithm of micro-blog retweet scale based on svm*, Application Research of Computers **30** (2013), no. 9, 2594–2597.

[18] Gang Liu, Chuan Shi, Qing Chen, Bin Wu, and Jiayin Qi, *A two-phase model for retweet number prediction*, International Conference on Web-Age Information Management, Springer, 2014, pp. 781–792.

[19] Zhunchen Luo, Miles Osborne, Jintao Tang, and Ting Wang, *Who will retweet me? finding retweeters in twitter*, Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, 2013, pp. 869–872.

[20] Mohammad Mahdavi, Masoud Asadpour, and Seyed Morteza Ghavami, *A comprehensive analysis of tweet content and its impact on popularity*, 2016 8th International Symposium on Telecommunications (IST), IEEE, 2016, pp. 559–564.

[21] Mohamed Morchid, Richard Dufour, Pierre-Michel Bousquet, Georges Linares, and Juan-Manuel Torres-Moreno, *Feature selection using principal component analysis for massive retweet detection*, Pattern Recognition Letters **49** (2014), 33–39.

[22] Paolo Nesi, Gianni Pantaleo, Irene Paoli, and Imad Zaza, *Assessing the retweet proneness of tweets: predictive models for retweeting*, Multimedia Tools and Applications **77** (2018), no. 20, 26371–26396.

[23] Omnicore, *Twitter by the numbers: Stats, demographics & fun facts.*

[24] Andrew Perrin, *Social media usage: 2005-2015: 65% of adults now use social networking sites–a nearly tenfold jump in the past decade*, Pew Research Trust, 2015.

[25] Fabio Pezzoni, Jisun An, Andrea Passarella, Jon Crowcroft, and Marco Conti, *Why do i retweet it? an information propagation model for microblogs*, International Conference on Social Informatics, Springer, 2013, pp. 360–369.

[26] René Pfitzner, Antonios Garas, and Frank Schweitzer, *Emotional divergence influences information spreading in twitter*, Sixth international AAAI conference on weblogs and social media, 2012.

[27] Evan TR Rosenman, *Retweets—but not just retweets: Quantifying and predicting influence on twitter*, Ph.D. thesis, Ph. D. thesis, Bachelors thesis, applied mathematics. Harvard College, Cambridge, 2012.

[28] Bongwon Suh, Lichan Hong, Peter Pirolli, and Ed H Chi, *Want to be retweeted? large scale analytics on factors impacting retweet in twitter network*, 2010 IEEE Second International Conference on Social Computing, IEEE, 2010, pp. 177–184.

[29] SurveyMonkey, *Sample size calculator.*

[30] TechCrunch, *Twitter adds possibly sensitive designation to tweets.*

[31] Raghavendran Vijayan and George Mohler, *Forecasting retweet count during elections using graph convolution neural networks*, 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA), IEEE, 2018, pp. 256–262.

[32] Wikipedia, *Kullback–leibler divergence.*

[33] Haihao Yu, Xu Feng Bai, ChengZhe Huang, and Haoliang Qi, *Prediction of users retweet times in social network*, International Journal of Multimedia and Ubiquitous Engineering **10** (2015), no. 5, 315–322.

[34] Tauhid R Zaman, Ralf Herbrich, Jurgen Van Gael, and David Stern, *Predicting information spreading in twitter*, Workshop on computational social science and the wisdom of crowds, nips, vol. 104, Citeseer, 2010, pp. 17599–601.

[35] Qi Zhang, Yeyun Gong, Jindou Wu, Haoran Huang, and Xuanjing Huang, *Retweet prediction with attention-based deep neural network*, Proceedings of the 25th ACM international on conference on information and knowledge management, 2016, pp. 75–84.

[36] Huidong Zhao, Gang Liu, Chuan Shi, and Bin Wu, *A retweet number prediction model based on followers' retweet intention and influence*, 2014 IEEE International Conference on Data Mining Workshop, IEEE, 2014, pp. 952–959.