Tweet Virality Prediction on Streamed Twitter Data

Ameya Potey
Computer Science
The University of Texas at Dallas
Richardson, TX, USA
anp200000@utdallas.edu

Koushik Grandhi
Computer Science
The University of Texas at Dallas
Richardson, TX, USA
kxg190042@utdallas.edu

Vedant Nibandhe
Computer Science
The University of Texas at Dallas
Richardson, TX, USA
vxn200012@utdallas.edu

***Abstract*—This report explores the use of Map-Reduce based K-nearest neighbors model predict the virality of streamed live twitter data based on attributes like the retweet count, tweet length, number of hyperlinks in the tweet, the number of followers of the user posting the tweet, the number of hashtags in the tweet and the actual words used in the tweet. For the sake of this project, we shall define a viral tweet as the one with a number of retweets higher than the median number of retweets per tweet in the entire dataset.**

***Index Terms*—Machine Learning, Spark streaming, Tweet streaming, K-nearest neighbors classifier, Map-Reduce, Big Data**

I. Introduction

Even though Twitter has been at the receiving end of a considerable amount of flak these past few weeks, it would not be a stretch to claim that Twitter, like other micro-blogging sites around the world has become an indispensable source of information with respect to any field one might possibly think of. Hence, we feel it is crucial to be able to understand what are the factors that make certain tweets/ pieces of information spread quicker than others.

These insights would be extremely handy in many different research areas, like analyzing opinion formation in the general public based off of information read online, and many other opinion mining tasks in the field of Big Data. In this paper, we attempt to address the important questions surrounding the information dispersal on twitter, such as is possible to predict the virality of a tweet, i.e., whether or not a tweet will be viral, and if so, to what extent? What features of a tweet most affect the virality of a tweet, as a model off of the features which have a high impact on the virality will yield very good results and accuracy.

Opinion formation [8] and opinion mining insights have a lot of significance in today's fast growing world especially in the field of prediction tools, which can the used by politicians, organizations etc. For example, once a new political initiative is proposed, it can be very helpful for politicians to understand what the general sentiment of the public is about the proposed scheme and it can then be implemented or scraped based on the feedback received. Similarly, public opinions on the beta version of a tech product could determine what features a tech company decides to keep or scrap from the final release.

A possible use case of this work can also be to answer the question as to whether it is possible to "design" a viral tweet, which could then shape the general opinion of the public on any given matter. In today's fast paced world where our primary source of information is the internet, it is crucial for us to be aware of the biases strategies that are being to shape and structure public opinions, especially on social media. It has been shown in many studies that there are in fact many inherent biases in most of the tweets and if these play a part in the formation of public opinion, it becomes extremely paramount to study how and to what extent these biases affect the spread of information on the internet.

II. Background Work

A lot of research has already been done in similar areas and have developed valuable insights. Twitter, as any other social network with a follower based structure can be classified as a graph, so intuitively it makes sense that the Pagerank [3] algorithm could be used to determine the importance of a user, but unlike one might be tempted to assume, having a higher pagerank, which usually correlates to a higher follower count does not automatically ensure that the tweets from the user in question will go on to become viral.

Interestingly though, a viral tweet might lead to increased followers for a user, as suggested by Lymperopoulos [9], as a viral tweet leads to a very high exposure, the user in question is exposed to a huge number of twitter users. Lymperopoulos [10] also had another interesting observation that for any given tweet, the activity of the tweet itself, like, retweets, likes , comments etc. start off rapidly immediately after the tweet is posted and then dies down into an asymptotic limit. Another useful aspect to note from prior research is that, unlike one might intuitively think, the virality of a tweet depends much more just the user and content of the tweet itself. Jiménez-Zafra [11] and Co find that things like sentiments in the tweet, hyperlinks, hashtags, mentions etc. have a significant effect on the virality of a tweet.Another interesting observation that we observed from prior work, like the one done by Pancer and Poole [10] is that positive semantic terms hurt the retweet count of a tweet in general,

whereas negative terms tend to boost the same.

Now, we shall put all these insights to good use in our own analysis. We will treat this as a classification problem, with two classes, viral and non-viral tweets. We shall take into account the retweet count, tweet length, number of hyperlinks in the tweet, the number of followers of the user posting the tweet, the number of hashtags in the tweet and the actual words used in the tweet as features in order to classify the tweets based on the K-nearest neighbors classification algorithm, we discuss in detail in the next section.

III. Theoretical and Conceptual Framework

A. *Introduction*

K-Nearest neighbors is a supervised learning algorithm which does not have any parameters as it depends on local approximation of the target function. We shall be using kNN for classification, wherein the algorithm assigns a class membership based on the plurality vote of its k nearest neighbors with respect to any distance metric defined by the user, like euclidean distance. A common way to improve the performance of the kNN classifier is to assign weights to the neighbors of any data point, where the weight of a neighbor is proportional to 1/d, where d is the distance of the neighbor from the node in question.

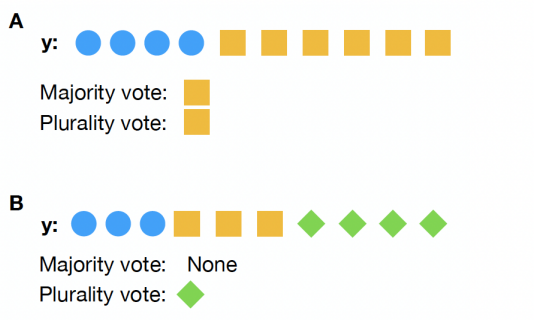

Fig. 1. Difference between majority and plurality vote, Image source: [1]

kNN just stores the labelled training data during the training phase (this algorithm does not have an explicit training step),

$$\langle x_i, y_i \rangle \in D \qquad (|D| = n) \tag{1}$$

B. *Mathematical formulation*

KNN defers the entire computation to the testing phase, and for this reason it is also known as a *lazy* learning algorithm. During predictions, the kNN fetches the k closest neighbors of the query point, where k is a model hyper-parameter, and computes the class label based on the majority of the assigned classes of the neighbors, as we can see the actual computation is done during the test phase. Since, there is no actual approximation of the target function involved, and the algorithm tried to locally approximate the target function every time, kNN is a **non-parametric** model.

For the sake of argument, let us assume that we have an approximation function *f(x) = y*, where y is the class label which can have n values.

$$f : \mathbb{R} \rightarrow \{1, ..., n\} \tag{2}$$

Now, we identify the k nearest neighbors $(D_{\|} \subseteq D)$ of the given query point $x^{[q]}$, we can now define kNN hypothesis as,

$$h(x^{[q]}) = arg \max_{y \in \{1,...,n\}} \sum_{i=1}^{k} \delta(y, f(x^{[i]})) \tag{3}$$

Where δ denotes the Kronecker delta function, which is defined as follows:

$$\delta(x, y) = \begin{cases} 1, & \text{if } x = y \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

For the distance metric, one of the most common metrics used is the euclidean distance, which is defined as follows:

$$d(x^{[a]}, x^{[a]}) = \sqrt{\sum_{j=1}^{t} (x_j^{[a]} - x_j^{[b]})^2} \tag{5}$$

which calculates the distance between the given data points over the *t* input features which the model takes into consideration.

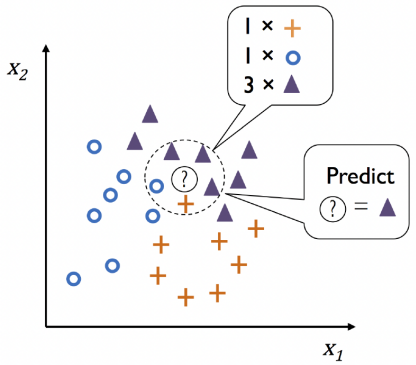

Fig. 2. Working of a kNN classifier for n=3 and k=5, Image source: [1]

Fig 2 illustrates how the class of a query data point is calculated based on the model hyper-parameter k and the number of classes n. As simple and effective as the kNN classifier may sound, it is prone to dimensionality issues. As the dimension of the dataset increases, the neighbors become more and more "distant" or different from each other, owing to all the different dimensions that need to be included when evaluating the distances. Hence, generally feature selection or feature extraction is done in order to mitigate this issue.

C. *Evaluation Techniques*

Confusion matrix is a really good evaluation technique for classification tasks. The confusion matrix gives us the following values:

*1) **True Positives (TP):*** The number of times the classifier classified the data point as True when its class was True.

*2) **True Negatives (TN):*** The number of times the classifier classified the data point as False when its class was False.

*3) **False Positive(FP):*** The number of times the classifier classified the data point as True when its class was False.

*4) **False Negative (FN):*** The number of times the classifier classified the data point as False when its class was True.

The confusion matrix can then be used to calculate more

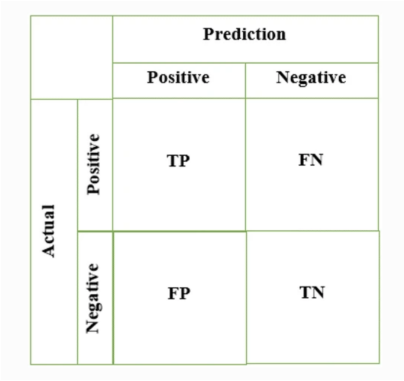

Fig. 3. Outline of a confusion matrix, Image source [4]

metrics like *Accuracy*, *Precision*, *Recall* and *F-score*. To test the performance even more comprehensively, K-fold cross-validation can be used in conjunction with the confusion matrix.

IV. Results and Analysis

A. Model Evaluation

We implemented a parallelized kNN using map reduce on a twitter dataset from kaggle as mentioned in the progress report, which had 11099 tweets and 31 features for every tweet, namely, created at, tweet id , tweet text, truncated, tweet entities, tweet metadata, tweet source, tweet reply fields, user profile data, geographical data, contributors ,retweeted status, quote status, retweet count, favorites count, favorited, retweeted, tweet language , sensitivity, quoted information, extended entities, quoted status, withheld in countries.

For our classifier, we used the number of tagged users in the tweet, number of urls in the tweet, the number of hashtags in the tweet,the length of the tweet and the number of followers of the tweet poster as the features. We tested the model with various test-train splits and k values to see for what model hyper parameters our model performance was optimum. We used the sklearn library to test our mdoel and found that for k = 7 and a test-train split of 80/20 our model performed marginally better than other combinations, with very similar performance across the board.

Table 1 shows the confusion Matrix for k=7 and a Test/Train split of 80/20. Table 2 shows the Classification report for the same set of hyper-parameters. Figure 4 goes on to illustrate

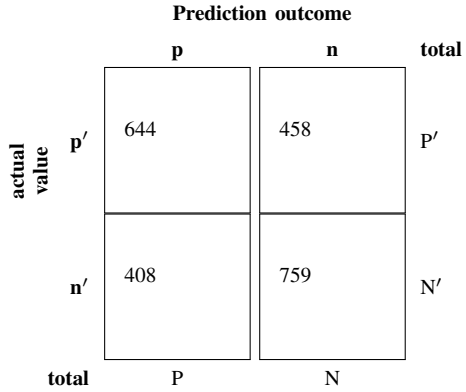

TABLE I
Confusion Matrix for K=7

how the model performance varies with k, and finally, Table 3 shows a summary of results for all the experiments performed.

TABLE II
Classification report for k=7

	precision	recall	f1-score	support
0 (class)	0.61	0.58	0.60	1102
1 (class)	0.62	0.65	0.64	1167
accuracy			0.62	2269
macro avg	0.62	0.62	0.62	2269
weighted avg	10.62	0.62	0.62	2269

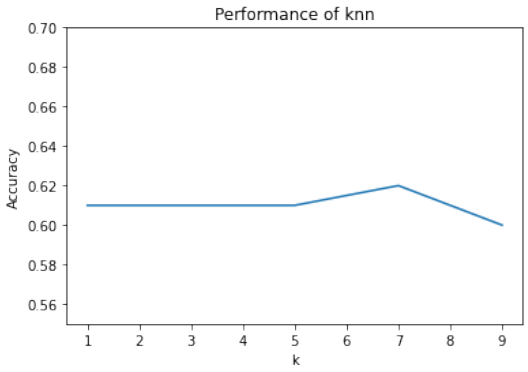

Fig. 4. Model Performance vs k

B. Insights Obtained

After pushing model results on the streamed tweets into Kafka and Elasticsearch and creating a few visualizations, we obtained a few interesting insights. Fig 5 shows the pie chart distribution of viral vs non-viral tweets obtained over a period of time. The dataset we have has randomly collected tweets over a certain timeline which has approximately equally divided target classes. Also, the Fig. 5 shows that the distribution of the two classes is more or less equal. The non-viral tweets

TABLE III
SUMMARY OF EXPERIMENTS PERFORMED

Experiment Number	Parameters	Accuracy
1	k=1 , test/train = 80/20	0.61
2	k=3 , test/train = 80/20	0.61
3	k=5 , test/train = 80/20	0.61
4	k=7 , test/train = 80/20	0.62
5	k=9 , test/train = 80/20	0.60

compose 51.85% and the viral tweets are around 48.15% of the total tweets obtained over a significant period of time.

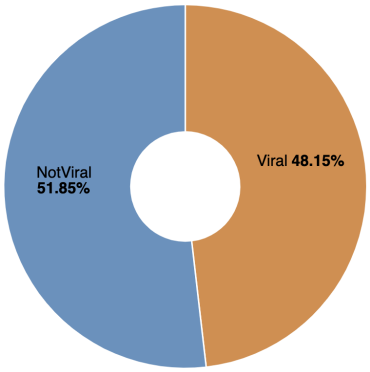

Fig. 5. Pie Chart of Tweet Classification

Fig 6 and 7 depict the distribution of tweets over the different contexts retrieved from the tweet text.The following categories show that in around 1250 tweets majority of them are tagged as Business taxonomy as the highest number of tweets owing to the current economic and market conditions surrounding employment setbacks, market recession, twitter privatisation etc. which all revolve around the business entities. This deduction helps to understand that the most of the tweets

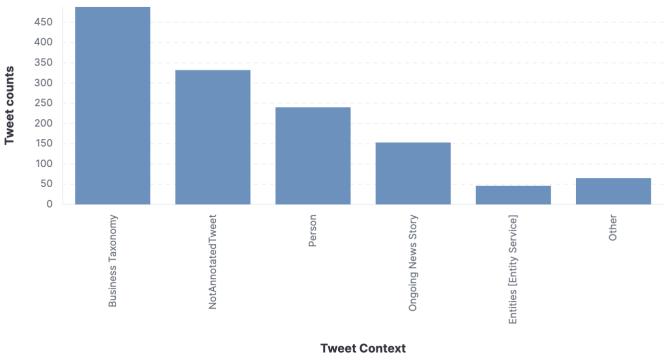

Fig. 6. Tweet contexts vs counts

nowadays are getting retweeted and have high possibility of getting retweeted.Also, we did obtain quite a few non-annotated tweets, as Twitter does not annotate tweets which are not comprised entirely of text, like tweets containing images, hyperlinks etc.

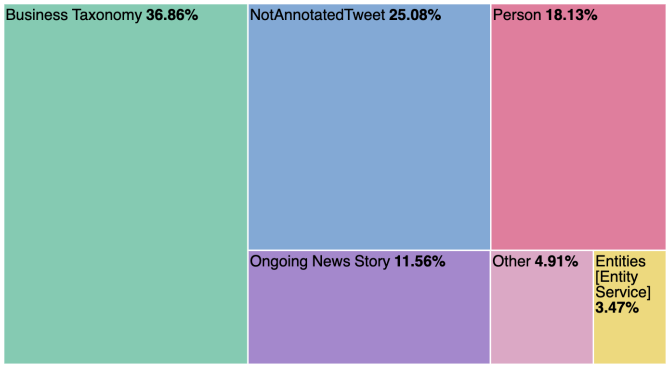

Fig. 7. Heat Map of Tweet Contexts

V. CONCLUSION AND FUTURE WORK

We got some very interesting and extensive insights from the tweet virality prediction using a parallelized kNN classifier. One aspect of this analysis that really intrigued us and made us choose this project in the first place is the amount of scope, potential and the impact these insights would have across disciplines. This work has immense future research potential, a few related areas that we can explore are:

1) Evolving Training Data and classification criteria: Trends, topics, median retweet counts and other relevant information would evolve over time, so we can constantly incorporate new data into our training data corpus to make sure the model remains relevant over time.Also, we can improve on our tweet classification criteria, considering a combination of factors rather than just considering the retweet count's median.

2) Natural Language Processing: We could also explore how the linguistic context of the posted tweets affects its virality, by using NLP techniques on the text of the tweets.

3) Tweet sequence Analysis: : Instead of treating every tweet as independent entity, we could also examine the relationships between the different tweets posted by users, like does a viral tweet have an effect on the upcoming tweet by the same user etc.

4) Multimedia Processing: We could also implement image/video processing on the images of the tweets to see if there is any correlation between different types of the media included in the tweet and its virality.

5) Ensemble Classification: We could also experiment with ensemble classifiers as they are often found to have higher accuracy than standalone classifiers.

6) Opinion Mining tasks: The data and insights obtained from this analysis can be used to perform opinion mining tasks like finding out the general sentiment over a new policy/product, observing how viral tweets affect opinion formation among the general public etc.

REFERENCES

[1] Notes on kNN - https://sebastianraschka.com/pdf/lecture-notes/stat479fs18/02_knn_notes.pdf

[2] Racial Bias in Hate Speech and Abusive Language Detection Datasets - arXiv:1905.12516 [cs.CL], https://doi.org/10.48550/arXiv.1905.12516

[3] What is Twitter, a Social Network or a News Media? - WWW '10: Proceedings of the 19th international conference on World wide web, April 2010 Pages 591–600, https://doi.org/10.1145/1772690.1772751

[4] Evaluation metrics for kNN classification algorithm- https://link.springer.com/article/10.1007/s42452-019-1356-9/figures/2

[5] https://tex.stackexchange.com - For Debugging

[6] https://stackoverflow.com/ - For Debugging

[7] Analyzing and Predicting Viral Tweets - by Maximilian Jenders, Gjergji Kasneci and Felix Naumann.

[8] ViralBERT: A User Focused BERT-Based Approach to Virality Prediction by Rikaz Rameez, Hossein A. Rahmani and Emine Yilmaz.

[9] RC-Tweet: Modeling and predicting the popularity of tweets through the dynamics of a capacitor by Ilias N. Lymperopoulos- https://doi.org/10.1016/j.eswa.2020.113785

[10] The popularity and virality of political social media: hashtags, mentions, and links predict likes and retweets of 2016 U.S. presidential nominees' tweets by Ethan Pancer and Maxwell Poole - https://doi.org/10.1080/15534510.2016.1265582.

[11] How do sentiments affect virality on Twitter? by Salud María Jiménez-Zafra, Antonio José Sáez-Castillo, Antonio Conde-Sánchez, and María Teresa Martín-Valdivia - https://doi.org/10.1098/rsos.201756.