Finding the infector in a multi-source case:

Retweeting behavior prediction based on user-based features

Candidate Number:RYDC3[1]

MSc Machine Learning

Supervisor:Shi Zhou

Submission date: 12 September 2022

CONTENTS

ABSTRACT

Nowadays, many studies have focused on retweet prediction. These studies can be divided into two directions: 1. predict a tweet will be retweeted by how many users. 2. predict whether a tweet will be retweeted by a given user. In this thesis we study retweet behavior in multi-source cases which are situation that multiple friends of a user post tweets with same hashtags. Our study aims are: 1. explore which informer will be retweeted in a multi-source case. 2. explore what factors affect people's choice of information source. To get answers, we first collect multi-source data with 10099 cases through Twitter API. On the basis of some basic features widely used in previous papers, we also extract interaction features from interaction history between informers and retweeters. We use XGBoost decision tree model to predict which informer will be retweeted by the user and get a good result with 0.67 f1-score. Then features are divided into three parts: informer features, retweeter features and interaction features. The results of experiments based on the combinations of three features group show that informer feature contributes most in the multi-source prediction problem. We can get 0.63 f1-score only with informer features, however 0.51 and 0.49 f1-score for only retweeter and only interaction respectively. Besides, both interaction features and retweeter features can improve the f1-score of model only based on informer features to 0.65. Our contribution in this thesis is twofold: 1. We present a baseline of collecting multi-source data set and solving multi-source prediction problem. 2. We introduce interaction features into retweet prediction problems and prove interaction features play similar role as retweeter features in multi-source prediction problem.

Keywords: Multi-source cases, Retweet Prediction, Feature analysis

ACKNOWLEDGEMENTS

I would like to thank my supervisior Dr. Shi Zhou, who spends at least two hours a week answering my questions and replying my emails even when he caught cold. Furthermore, I would also express my gratitude to my friend Yuheng Jia for the accompany and inspirations during thesis writing. Last but not least, thanks to my parents who provide me the tuition fee and support my decision to study abroad.

I. INTRODUCTION

In the context of web 2.0, information transmission has become easier than ever. Information spreading can bring huge commercial interest and even influence the result of election. Thus it is crucial to study what kind of information is easily propagated. Twitter as a highly active social network, generates millions of information a day and because of the retweet mechanism, we can easily get the information forwarding trace. Therefore Twitter is an excellent platform to study information diffusion.

To facilitate the description of the article, it is necessary to give some significant definitions here:

Definition: If a user have at least two friends post tweets with same hashtag and he/she retweet one of friend's tweet. We call this case as **multi-source case**, these friends as **informers**,the user who retweets that tweet as **retweeter**, the author of the tweet as **infector**. Note: if informers and infectors are mentioned together, in this situation informers refer to all the informers apart from the infector.

Retweet function is the key mechanism for information spreading in Twitter network. There are many studies on retweet prediction. These studies can be roughly divided into two categories1. predict a tweet will be retweeted by how many users. 2. predict whether a retweet will be retweeted by a given user. However, in this thesis, we focus on the retweeting behavior in 'Multi-source' cases. Our study aims are: 1. explore which informer will be retweeted in a multi-source case. 2. explore what factors affect people's choice of information source.

We answer the research questions by collecting a multi-source data set and extract features based on the features we extract. Except some basic features widely used in previous works, we also introduce interaction features, which are

extracted from interaction between informers and retweeters. We train models on the train set based on these features and use f1-score to evaluate the prediction effect on test set. In order to know which factor has the greatest influence on the user's choice of information sources, we groups the extracted features and design experiments to test the predictive effect of different combinations of features groups on the multi-source prediction problem.

What's interesting about this thesis is that we don't just predict the retweet behavior, we care more about the relationship between informers and retweeters. The user's choice of information source reflects the trust level of users for different informers. In this thesis, we're actually looking at what kind of people are more trusted by their friends.

II. BACKGROUND

In this chapter, I will first introduce Twitter, the social network where I collect data; in the second and third sections I will introduce the relevant knowledge of network science and machine learning related to my research respectively, which will help people to better understand this thesis.

1. TWITTER

Twitter is one of the most notable micro-blogging services, which allows users publish tweets with at most 140 characters. Besides, Twitter also provides the social-network function. [1] Due to its ease for real-time information sharing, Twitter has impacted public discourse in the society. [2]

Twitter had a total of 238 million daily active users around the world in July 2022 and around 500 million tweets are sent each day. United States, Japan, India, Brazil, Indonesia, United Kingdom, Turkey, Saudi Arabia, Mexico, Thailand are the top 10 countries with number of twitter users. [3] Twitter is also used as a platform to share news headlines, almost 85% of its trending topics are associated with breaking news. [4].

Tweet is any message posted to Twitter which may contain photos, videos, links, and text, which is the most basic function in Twitter platform.

Timeline displays a stream of Tweets from accounts you have chosen to follow on Twitter. Besides, Twitter platform also recommend tweets to users based on who users already follow and topics users follow.

Follower is a user who is following you. Followers can see the tweets posted by you in their Timeline. Followers will show up in your follower list and you can start a private conversation with them.

Friends are people you are following in the Twitter networks. At the same time, you become a follower of your friend, which means their updates will be added to your timeline.

Mention is a tweet containing another account's Twitter username, preceded by the "@" symbol. A mention can be seen as an interaction between users.

Reply is a kind of mention action happening when a user respond to another person's tweet. Anyone following the user and the recipient of that reply will see the reply in their Home timeline.

Informer is one of the friends of a user, who has tweeted the related hashtags and therefore can potentially inform the user of the outbreaking news under study. A user can have many informers, but the user does not necessarily get infected by any of them.

Original Tweet is a completely original tweet consisting of a string of text or a picture edited independently by the user but excluding any retweet or quote.

Searching is one of the most important way of getting information in Twitter. Users can find Tweets from themselves, friends, local businesses, and everyone from well-known entertainers to global political leaders. By searching for topic keywords or hashtags, users can follow ongoing conversations about breaking news or personal interests.

Lists allow users to organize tweets seen in a user's timeline. You can choose to join lists created by others on Twitter, or choose from your own accounts to create lists of other accounts by group, topic, or interest. Viewing the list timeline will show you the stream of tweets only from accounts on that list.

When a real event occurs, it becomes a **trending topic** in twitter. We can easily collect massive human-generated data through **Twitter developer API**, thus twitter is a popular platform among researchers.

When a new topic becomes popular on Twitter, it is listed as a trending topic, which may take the form of short phrases (e.g., Stephen Curry) or hashtags (e.g. nba).[5]

Hashtag is a simple free form keyword preceded by a character "#" (e.g. #ClosingCeremony). A hashtag is translated into a clickable link for easy searching with tweets with same hashtags. [2]

Retweet is an interesting behaviour in Twitter network, which is a relaying behaviour for a twitter written by another user. This is the key mechanism in Twitter to spread information.

2. INFORMATION DIFFUSION

Information diffusion refers to the process that information or knowledge is spread from one place/node to another through interactions/links. It is a field that encompasses techniques from a plethora of sciences and techniques from different fields such as sociology, epidemiology, and ethnography.[6] With the popularity of the Internet, the speed of information dissemination is unprecedentedly fast and convenient, and social networks have become an indispensable part of people's lives. Lots of latent information can be mined through people's online interaction, which can be used for market predicting, rumor controlling, and opinion monitoring among other things. [7] . Therefore, a lot of research has been carried out on the problem of information diffusion in social network.

We can model the process of information diffusion in three parts: senders,receivers and medium. They play different roles in the process. Senders are responsible for initiating the diffusion process. Receivers receive information from senders. And medium can be viewed as a channel between senders and receivers. If we consider information diffusion through social network, mediums can be Twitter, facebook, weibo and etc..

In Twitter network, the senders are the author of tweets and the receivers are users who have seen these tweets. Information is spread by the retweet mechanism. Social network can be abstractly viewed as a graph, where users are nodes and the the spreading behavior between two users can be seen as edge between two nodes. Therefore a real social group can be mapped by an abstract social network and the process of information can be viewed as information spreading from one node through edges to other nodes. Different nodes play different roles in the network: some of them refuse to spread information, some of them cannot accept information.[8][9] In order to study the complex process of

information diffusion and study the key factors affecting the process, scholars have proposed many explanatory models.

The process of information dissemination can be regarded as the process of epidemic dissemination. In the spread of epidemics, there are three types of people: Suceptible people (S), Infected people (I), Removed People (R). Suceptible people can be infected if they are close to infected people. Removed people will never get infected again by the virus as they are immune. The basic epidemic models are combinations of different stats.

2.1. SI Model

SI (Suceptible Infected) model is built based on the assumption: All the people in the community are susceptible people and can be infected by sick people. We only consider the infection process, which means people cannot be cured and will not get immune. SI model is firstly used for social network study in 2001. [10]

In SI model, we do not need consider the birth rates and death rates. We can represent the total number of people as N. Only two kinds of people are contained under the assumption of SI model: susceptible people and infected people. The susceptible proportion at time t can be represented by $s(t)$ and the infected proportion at time t can be represented by $i(t)$. We can use λ to represent contact rate, which is the proportion of susceptible people infected by infected users. It is clear that every day $\lambda N s(t) i(t)$ users are infected; At time 0, if the proportion of infected people is i_0, we can describe SI model using the following equations:

$$\frac{di}{dt} = \lambda i(1-i),$$

$$i(0) = i_0.$$

2.2. SIS Model

The SI model oversimplified the process of virus transmission so it is not practical. SIS model introduce the process of healing on the basis of the SI model. Excepte

the process of infection described in SI model, SIS model add the daily rates of the cured patients μ. The increment of the change in the number of infected people per day can be described by: $N\frac{di}{dt} = \lambda Nsi - \mu Ni$, where λNsi represents the number of newly infected people and μNi represents the number of newly cured people. As assumed in SI model, the proportion of the infected people at the initial time is i_0, and we can describe the SIS model by the following equations:

$$\frac{di}{dt} = \lambda i(1-i) - \mu i,$$

$$i(0) = i_0.$$

2.3. SIR model

On the basis of SI and SIS model, we also consider removed users in SIR model. If an individual is cured after infection, this person will be an immune person, which means he/she will never get infected again in the future. Under the assumption of SIR model, except suceptible people and infected people, there are also some immune people, represented by R. $r(t)$ represent the proportion. The daily increment of the increasing number of immune users can be expressed by $N\frac{dr}{dt} = \mu Ni$. We can describe the SIR model by the following equations:

$$\frac{ds}{dt} = -\lambda si,$$

$$\frac{di}{dt} = \lambda i(1-i) - \mu i,$$

$$\frac{dr}{dt} = \mu i.$$

2.4. SIRS Model

The SIRS model assume even though a person is cured, the person is possible to be suceptible again with probability α. Other symbols are the same as before. We can describe the SIRS model by the following equations:

$$\frac{ds}{dt} = -\lambda si + \alpha r,$$

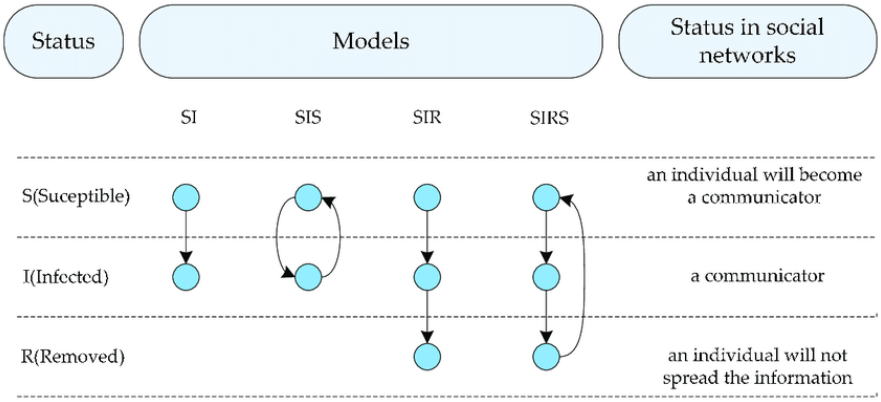

Figure 2.1: Comparison of SI, SIS, SIR, SIRS models.

$$\frac{di}{dt} = \lambda i(1-i) - \mu i,$$

$$\frac{dr}{dt} = \mu i - \alpha r.$$

The comparison of SI, SIS, SIR and SIRS is shown in Figure 2.1, which also illustrate the information diffusion process of a virus in an epidemic, showing how the epidemic model can be used for social network.

3. MACHINE LEARNING

3.1. Distribution Test

We want to test whether a feature of the informer and this feature of the informers come from the same distribution; if they come from the same distribution, which means that this feature cannot distinguish the informer and the informer, so it is less likely to be the key feature in predicting forwarding behavior. Next, I will introduce a few methods to determine whether two columns of data come from the same distribution.

T-test: T-test is the most commonly applied when the test statistic would follow a normal distribution if the value of a scaling term in the test statistic were known. T-test can be used to determine if the means of two sets of data are significantly different from each other.

For two sequences X_1 and X_2, where $\overline{X_i}$, s_i, n_i represent the mean, variance, number of sequence X_i. We need calculate the *t* statistic to test whether the population means are different.

$$t = \frac{\overline{X_1} - \overline{X_2}}{(s_1^2/n_1 + s_2^2/n_2)^{\frac{1}{2}}}$$

And then check table according to the value of t to get p-value.

Kullback–Leibler divergence:The Kullback–Leibler divergence, also called KL divergence, denoted by $D_{KL}(P\|Q)$, is a type of statistical distance: a measure of how one probability distribution P is different from a second, reference probability distribution Q. For distributions P and Q of a continuous random variable, relative entropy is defined to be the integral, where $p(x)$, $q(x)$ denote the probability densities of P and Q:

$$D_{KL}(P\|Q) = \int_{-\infty}^{\infty} p(x) \log(\frac{p(x)}{q(x}) dx$$

Kolmogorov–Smirnov test: The Kolmogorov–Smirnov test (K-S test) is a nonparametric test of the equality of continuous or discontinuous, one-dimensional probability distributions that can be used to compare a sample with a reference probability distribution (one-sample K–S test), or to compare two samples (two-sample K–S test). It can answer the question that "What is the probability that these two sets of samples were drawn from the same (but unknown) probability distribution". It quantifies a distance between the empirical distribution functions of two samples.The two-sample K–S test is one of the most useful and general nonparametric methods for comparing two samples, as it is sensitive to differences in both location and shape of the empirical cumulative distribution functions of the two samples. In the case that test whether two samples are from same distribution, the KS statistic can be writen as followed:

$$D_{n,m} = \sup_x |F_{1,n}(x) - F_{2,m}(x)|,$$

where two samples' size are n, m a, the empirical distribution for them are $F_{1,n}(x)$ and $F_{2,m}(x)$ respectively. For large samples, we can reject the null hypothesis at level α if:

$$D_{n,m} > \sqrt{-\ln(\frac{\alpha}{2}) \frac{1 + \frac{m}{n}}{2m}}$$

3.2. Classification Model

We can view the problem that predicting which informer will be retweeted by a user as a binary classification problem. The informer-retweeter pair will be labelled 1 when user retweet that informer's tweet; 0 when the user ignore the informer's retweet. In this section,

Logistic regression: We need to learn a classification:

$$\hat{y} = \sigma(w^T x)$$

where $\sigma(z) = \frac{1}{1+e^{-z}}$ is sigmoid function. w is the weights for features. We can view the sigmoid function as the probability of the query-passage pair is relevant.

We will calculate cross-entropy loss of the regression model:

$$L(w) = -\frac{1}{n}\sum_{1}^{n}[y_i ln\sigma_w(x_i) + (1-y_i)ln(1-\sigma_w(x_i)]$$

The equation above can also be written as:

$$L(w) = -\frac{1}{n}\sum_{1}^{n}[w^T x_i y_i - w^T x_i - ln(1+e^{-w^T x_i})]$$

We want to get the value of vector w by iterating, so we calculate the gradient of the weights:

$$\frac{\partial L}{\partial w_j} = \frac{1}{n}\sum_{i}^{n}[x_{i,j}(y_i - \sigma_w(x_i)]$$

And we update w_j with a learning rate (a) set by ourselves by:

$$w_j = w_j - a \times \frac{\partial L}{\partial w_j}$$

In general, we initialise weights w then compute the partial deviatives for each w_j, then update w_j using a learning rate and repeat until convergence or reach a certain epoch.

KNN: In statistics, the k-nearest neighbors algorithm (KNN) is a non-parametric supervised learning method which can be used for classification and regression problems. Next I will mathematically describe the application of

KNN in binary classification.

Data $D = \{d_i = (\vec{x_i}, y_i), 0 \leq i \leq n\}$, where n is the size of data set, vector $\vec{x_i}$ is the i_{th} data's features, y_i is label of d_i. Let $N(\vec{x}, k)$ be the set of k nearest inputs to $\vec{x}$ and

$$I_x = \{i : x_i \in N(\vec{x}, k)\}$$

the corresponding index set. And we can get a classifier f,

$$f(x) = 1 \quad if \quad |y_i = 1 : i \in I_x| > |y_i = 0 : i \in I_x|$$

$$f(x) = 0 \quad if \quad |y_i = 1 : i \in I_x| < |y_i = 0 : i \in I_x|$$

Generally, closeness is measured using Euclidean distance.

SVM: Support vector machine (SVM) is a supervised learning model which is widely used for data classification and regression problems. SVM can efficiently deal with linear classification problems and non-linear classification problems by kernel trick. In this section, I will introduce how SVM is used in linear situation.

The data $(\vec{x_i}, y_i) \in R^m \times \{-1, 1\}$, where m is the dimensions of features. A hyperplane can be writen as the sets of data satisfying: $\vec{w} \cdot \vec{x_i} - b = 0$, where $|\vec{w}| = 1$. We want to find the "maximum-margin hyperplane" that can divide the group of points $\vec{x_i}$ with positive labels from the group of points with negative labels. If the data set is totally linearly separable, the question is coverted to:

$$Minimize : \|w\|$$
$$Subjected \quad to : y_i(\vec{w} \cdot \vec{x_i} - b) \geq 1 \qquad i = 1, 2, ..., n$$

If it is impossible to linearly separate the data set, then we need to minimize the following equation:

$$\lambda \|\vec{w}^2\| + \frac{1}{n} \sum_{i=1}^{n} max(0, 1 - y_i(\vec{w} \cdot \vec{x_i} - b))$$

In the above equation $\lambda > 0$ is used to determine the trade-off between increasing the margin size and ensuring that the $\vec{x_i}$ lie on the correct side of the margin. It is worth mentioning that when rbf kernel is used, the SVM algorithms

will be similar to KNN.

Decision Tree: A decision tree is a flowchart-like structure where every node contain a test on a feature, every branch can be used to represent the outcome of the test, and every leaf node represents a class label. The classification rules can be expressed by the path from root to leaf. One of the big advantages of decision tree model is easily understood and very intuitive. Building a decision tree usually takes three steps: feature selection, generating decision tree and pruning. ID3, C4.5 and CART are three most widely used decision tree models, which use information gain, gain ratio and Gini index respectively to divide samples.

Naive Bayes: In statistics, naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naive) independence assumptions between the features. The classifiers can be trained by increasing the metric maximum-likelihood, with linear time cost.

In general, naive bayes model is a kind of conditonal probability model: when data can be represented by a vector $\vec{x} = (x_1, 2_2, ..., x_m)$, it assigns to this instance probabilities:

$$p(C_k|x_1, ..., x_n)$$

for each K possible classes C_k. (We only need to consider binary classification, $K = 2$). By bayesian formula:

$$p(C_k|\vec{x}) = \frac{p(C_k)p(\vec{x}|C_k)}{p(\vec{x})}$$

Where $p(C_k)$ is prior probability, $p(\vec{x}|C_k)$ is likelihood and $p(\vec{x})$ is evidence. For most of cases, people are only interested in the numerator of that fraction, because the denominator does not depend on class C_k ;besides, the values of the features x are given, so that the denominator is effectively constant. The numerator is equivalent to the joint probability model:

$$p(C_k, x_1, ..., x_n)$$

For brevity, we assume features are independent, thus we have:

$$\begin{aligned} p(C_k, x_1, ..., x_n) &= p(x_1|x_2, ..., x_n, C_k)p(x_2, ..., x_n, C_k) \\ &= p(x_1|C_k)p(x_2, ..., x_n, C_k) \\ &= ... \\ &= p(C_k)\prod_{i=1}^{n} p(x_i|C_k) \end{aligned}$$

Therefore, we get the probability of data $\vec{x}$ belong to class k; The bayesian classifier can be written as followed:

$$y = argmax\{p(C_k)\prod_{i=1}^{n} p(x_i|C_k)\}$$

3.3. Evaluation Metrics

A model may have totally different performance on different metrics. Thus we need to use multiple different indicators to comprehensively judge the performance of a model on this data set. In this section, I will introduce 4 metrics used in this thesis.

Accuracy: In binary classification, accuracy is the fraction of correct classifications, which is the most widely used metric to evaluate the algorithm.And accuracy is also the metric used to optimize parameters to increase model performance in this thesis.

Recall: In binary classification, recall is the fraction of all the cases with label 1 that are successfully predicted, which reflects whether our algorithm is comprehensive.

$$Recall = \frac{|Cases\,are\,predicted\,as\,label\,1| \cap |Cases\,are\,actually\,labeled\,1|}{|Cases\,are\,predicted\,as\,label\,1|}$$

Precision:Precision can be expressed by the fraction of the cases are predicted as label 1 that are successfully predicted in the context of binary classification, which reflects whether our algorithm is accurate.

$$Precision = \frac{|Cases\,are\,predicted\,as\,label\,1| \cap |Cases\,are\,actually\,labeled\,1|}{|Cases\,are\,actually\,labeled\,1|}$$

F1-score:The F1 score is the harmonic mean of the precision and recall, which evaluates both the comprehensiveness and accuracy of the algorithm.

$$F1 = 2 * \frac{percision * recall}{percision + recall}$$

III. LITERATURE REVIEW

1. RELATED WORK

1.1. Why do users retweet?

The retweet mechanism in Twitter network is an especially powerful tool for improving the communicablity of tweets. Analyzing the retweet behavior offer insights about how users interact with each others and with different types of information.

Weng et al. [1] found that 70% twitter users follow their followers out of either ‘reciprocity’ or ‘homophily’. They also said users who have similar interest in same topics tend to follow each other.

Anuja Majmundar et al. [11] designed a questionnaire to ask participants why they retweeted. And using PCA to determine the importance of influencing factors, the top four reasons are: 1)To introduce my followers to the tweeter;2) To show my support for the tweeter;3)To show my followers that I like the tweeter;4)because I trust the tweeter.

Xu et al.[12] used the strategy "leave-one-feature-out" to analyze the importance of features and found that retweeting behavior is highly associated with user features. At the same time,they believed users with different intentions would have different retweet patterns. For example, an information seeker will retweet more tweets regarding to his interest, but a user who attach importance to social relationship is more prone to retweet from his close social friends.

Lahuerta-Otero et al. [13] identify which elements of the messages enable tweet diffusion and facilitate, and find that tweets with more information are easily retweeted by other users. For instance, longer messages are disseminated by users much more than the shorter ones; tweets with less hashtags are also easily

forwarding. It's all because tweets with high percentages of valid information are welcome in the context of information diffusion.

Macskassy et al.[14] find that people's retweeting behaviour shows "Anti-Homophily", which means mahority of users do not retweet information on topics they often tweet about or from people who are following them. At the same time, they find take the feature homophily or similarity betweet users into account can greatly improve model prediction accuracy.

1.2. Retweet Prediction

Various studies have been done on the retweet prediction problem because of the important roles the retweet function plays in Twitter network.Most of studies can be divided in two types: one is author-centered and the other is user-centered. [15] The former studies predict the scale of a tweet's retweet volume and the later studies focus on predict whether a given user will retweet a specific or not.

Author-centered studies always focus on factors that makes the author an influential user and the content of tweets. For example, Suh et al. [2] firstly examined a number of features that might affect retweetability of tweets. And they found that the number of followers and friends as well as the age of the account play the most important role. Besides, amongst content features, URLs and hashtags also have strong relationships with retweetability. Although the author-centered studies has achieved good performance in many data set, it ignores users. Wu et al.[16] said different categories of users emphasize different types of content and they also find significant "homophily" within categories. Therefore, the retweeting behavior is not only related to the author, but also to the user, so the research focus has shifted to user-centered.

Most of user-centered studies extract features from users' profile and the tweets and build model to predict retweeting behavior. Lee et al. [17] hope to model a person's likelihood to retweet, so they identified six categories of features such as profile features, social network features, activity features etc. and used multiple classical machine learning methods, such as Logistic Regression models and Naive Bayes, to predict a user's retweeting behavior.

Yang et al.[18] propose a factor graph model to predict users' retweeting

behaviors which combined multiple retweet features and found that social relationship had the greatest promotion on prediction accuracy. They also have some important observation: 1. most of users retweet at a low frequency. 2. if the users are interested in the message, there is a higher probability to retweet it. For the first observation, I have a question: If we conduct experiment on data set where most of users retweet at a high frequency, will the rank of feature importance will change? In my thesis, I want to predict the retweeting behavior on multi-source data set, where most of users are retweetaholics.

Wang et al. [19] propose a deep neural network to user embedding, target tweet embedding,the results of network embedding and user history embedding. And they use these embeddings to predict whether the target tweet will be retweeted by the user and have good prediction results. Although their model have good performance, their results are lack of interpretability.

In conclusion, I hope combine both users and authors, using the multidimensional features for retweet prediction. Besides, current studies always ignore the interactions between authors and users.According to our hypothesis, the higher interaction frequency between two users, the higher probability the tweet will be retweeted. Therefore, I will also add interaction features such as retweet frequency, reply frequency etc. to our model.

Liu et al.[20] believed that it is necessary to accurately distinguish which tweets are seen but not retweeted by users, and which tweets are not seen by users.So they counted the time interval of users' use of Twitter, and regarded the usage time interval as a feature and achieved good results. Their work also inspire us that the post time could also be an important factor as intuitively people prefer retweeting newly post tweets.Additionally, Araujo et al. [21] find that tweets posted on Tuesday through Thursday receive more retweets than those posted on other days of the week. Therefore, in my work, I will also extract time features.

2. SUMMARY

In this chapter, I introduced related works in the following two parts: Firstly, I explored the reasons for retweeting. Both the authors of tweets and the content of

tweets may be reasons for people’s retweeting behavior. Secondly, I introduced some methods of retweet prediction. Previous work always focused on one side of retweet, thus can be classified by author-centered works or user-centered works. I will synthesize previous works, and introduce interaction features in this thesis.

IV. PROJECT PLAN

In this chapter, I firstly state the questions we want to study and then propose the methods that can help us address the problems.

1. RESEARCH QUESTIONS

In past researches, scholars are more interested in predicting whether a post (tweet) is going to be propagated or try to model the level of propagation. [22] In this thesis, I want to study the relationship of trust. To be exact, I want to know when people hear information from different friends, which friend will be trusted. In twitter network, the trust problem reflects on when a user has multiple friends who send tweets with a same hashtag (informers), which friend's tweet will the user retweet. We call the phenomenon that a user have many informers as multi-source case.

In this thesis, we study three related questions: 1. What factors will affect users' choices of sources? 2. Is it possible to predict which informers will be retweeted in multi-source cases? 3. What factors have the greatest impact on users' choice of information sources in multi-source cases? multi-source cases

To answer these questions, firstly we need a data set which contains multi-cases. The first step is collecting multi-source cases data through Twitter API, as we didn't find any previous study foucsing on multi-source cases. It is very convenient to apply for twitter api, but we need notice that because of rate limit of API, data collection is very time-consuming. Therefore it is necessary to apply several accounts and change accounts frequently to avoid reaching rate limit.

Secondly, both the content of tweets and user profiles are all text information, so we need extract features such that we can apply machine learning models on

the data.

Thirdly, features we have extracted may have correlation, so it is necessary for us check correlation between each feature. We can abandon overlapping features to make our model clear and efficient.

Fourthly, we need to check whether informers' feature and the same feature of infectors are from same distribution. This is a preliminary screening of features that may affect users' retweeting behavior, as we believe that if a features are from the same distribution on informers and infectors which means the vast majority of users have no preference for this feature, so this feature cannot be used as the basis for judging the infectors and the informers.

Fifthly, train different models on our data set based on features we extract and compare their classification effect. If the f1-score of the best model exceeds 0.6, we can say it is possible to predict which informers will be retweeted in multi-source cases.

Sixthly, group the features and test which group of features contributes most to the prediction of retweeting behavior in multi-source cases. This can greatly improve the interpretability of the models, allowing people to intuitively know what influences the user's choice of information sources in the Twitter network.

Lastly, due to the time constrains, in this thesis we can only explore the influence of user characteristics on the selection of information sources, but we cannot know the influence of tweet content. The influence of content is measured by the other student of this project. The answer of the question what factors affect people's choice among multiple informers can be answered when combining we two works.

2. METHOD

In this section, I will introduce the methods to address the research problems.

We used machine learning method to study our problem, which implies: 1. each tweet is represented by a set of features; 2. a training set is used to learn the model before model is used on the test set or new tweets. [22] According to our intuition, both the author of the tweet and the tweet content have an impact

on users' choice of information source. But in this thesis, I will only focus on user features. To extract features that can represent users, we borrow the features that posted by previous student and propose new features regarding to retweeting behaviour.

To address question 1, we need extract features from users' profiles and users' behavior. After we get features of informers and infectors, we need compare the distribution of two samples. In our thesis, we choose T-test, which is an inferential statistic to determine if there is a significant difference between the means of two samples. We use the p-value as an indicator for judgment, which is the probability of obtaining test results at least as extreme as the result actually observed. If p-value less than certain value, we reject the null hypothesis (features of informer and infectors come from the same distribution) if the p-value is smaller than certain value.

To address question 2, we need build model based on the feature we have extracted. We plan to build Logistic Regression model, Support-Vector Machine model and Decision Tree model. We need divide the multi-source cases data set into training set and test set. We train models on training set by cross-validation and choose best parameters, then we predict the retweeting behavior on test set based on trained models. We compare the prediction performance (f1-score and accuracy) of models. If both f1-score and accuracy exceed 0.6, we can assert that retweeting behavior among multi-source cases can be predicted. Besides, Logistic Regression models can provide weights of each features, the bigger the weight, the more important of the feature. Decision Tree model can provide us the decision nodes, which are user features in our research. The closer the node is to the root of the tree, the more important the node is. Thus, these two models can also tell us the importance of features.

To address question 3, we should group features we have extracted and design experiments to test the effectiveness of different combination of features groups. We can judge which set of features contribute most by f1-socre and accuracy. If time permits we will also borrow text features extracted by the other student in this project and test the importance of text features for predicting users' choices in multi-source cases.

In summary, I will describe the process of data collection and conduct exploratory data analysis for our multi-source data set in the Chapter V; In Chapter VI, I will extract features and do feature analysis. Then T-test is used to compare the distribution of informers features and infectors features. In Chapter VII, multiple models are used to test the effectiveness of features extracted. Then features are grouped into different categories and prediction models are constructed based on combinations of different groups of features. In the Chapter VIII, we will summary the conclusion we get and give answers for the research questions.

V. DATA

1. METHOD

We collect multi-source cases through Twitter official API, which is the most convenient and powerful method for data collection in Twitter network.[23]

We first choose a hashtag and then we collect latest tweets with that hashtags. We collect authors of tweets and traverse the friend lists of these authors. We consider user B as a informer of user A if and only if: 1. user B is followed by user A; 2. User B post a tweet with that hashtag before A retweet. In particular, if A retweet B's tweet, we consider B to be the infector of A. In our project, we only consider users with multiple informers. Algorithm 1 shows the process of collecting multi-source cases. All the data used in this thesis has been upload to Github

2. DATA DESCRIPTION

Actually whether a tweet be retweeted is highly related to its topic. [24] In order to reduce the bias of our feature extraction, we want to collect data from different topics, making our data set closer to data distribution in real world. The more realistic the data is, the better the outcomes will be. Thus, we need collect data from varieties of hashtags, covering as many fields as possible.

As shown in table 2.1 and table 2.2, We collected 19 topics, a total of 83000 tweets, of which 10099 multi-cases are contained in the data set. The hashtags we selected are all from UK trends. An example of trend is shown in Figure 2.1. The number below hashtag represent the number of newly post tweets. We hope to collect as many recent tweets as possible and trace them back to the first tweet with the hashtag. The number of tweets we select is based on the numbers

Algorithm 1: Get Multi-cases

Data: Recent tweets (reversed) with same hashtags $tweet\_list$; The authors list of these tweets $user\_ids$

Result: A dictionary that store retweet information

```
d = {};
i = 0 ;
while i < len(tweet_list) do
    if tweet_list[i] is not a retweet then
        i ← i + 1;
        continue;
    end
    d[user_ids[i]] = list();
    api = get_api();
    response = api.get_friend_ids(user_id = user_ids[i]);
    j ← 0;
    while j < i do
        if user_ids[j] in response then
            d[user_ids[i]].append(user_ids[j]);
        end
        j ← j + 1;
    end
    i ← i + 1
end
```

marked under trends.

In table 2.1, we list the number of multi-cases and the average number of informers for each tweet. The two hashtags with the highest percentage of multi-cases are taiwan and EnoughsEnough, covering 21.46% and 21.25% respectively. Besides, hashtag onepiece has nearly the lowest percentage of multi-cases, as One Piece is a Japanese manga series which has lots of fans. It's a good example of "Anti-Homophily" phenomenon. [14] Contrary to our intuition the average number of informer and the number of multi-cases are not positively correlated.

In table 2.2, we list cases with one-informer and retweeted from non-friend. It can be seen that the proportion of retweets from non-friend is quite high. Among the 19 topics we counted, the proportion of retweets from non-friends all exceeds 60%, which shows that the recommendation mechanism and trending on Twitter play a more important role in information diffusion in twitter network than forwarding between friends.

Figure 2.1: An example of trending

hashtags	num of tweets	multi-source cases	percentage	ave num of informers
#EnoughIsEnough	2000	425	21.25%	20.51
#ToryScumOut	2000	165	8.25%	34
#VoteThemOut	2000	81	4.05%	5.45
Lisa	2000	223	11.15%	7.60
#crystalpalacevsarsenal	2000	331	16.55%	6.10
#freecodefridaycontest	2000	313	15.65%	6.33
#nba	2000	334	16.70%	6.01
#SunakTheSnake	2000	303	15.15%	6.14
#TheLastLeg	2000	318	15.90%	6.08
#ClosingCeremony	5000	463	9.26%	7.92
#onepiece	5000	231	4.62%	3.55
F1	5000	521	10.42%	7.90
#BookLoversDay	5000	623	12.46%	9.63
#biden	5000	832	16.64%	9.55
#GetBackToworkYouFatPonce	5000	340	6.80%	42.86
#IOS16	5000	691	13.82%	7.84
#taiwan	10000	2146	21.46%	11.62
#CostofLivingCrisis	10000	1173	11.73%	38.42
#EnergyPrices	10000	899	8.99%	35.76

Table 2.1: Stats for multi-source cases in our dataset

hashtags	num of tweets	one-informer cases	non-friend cases	per of non-friend
#EnoughIsEnough	2000	94	1481	74.05%
#ToryScumOut	2000	16	1819	90.95%
#VoteThemOut	2000	67	1852	92.60%
Lisa	2000	85	1692	84.60%
#crystalpalacevsarsenal	2000	449	1220	61.00%
#freecodefridaycontest	2000	411	1276	63.80%
#nba	2000	453	1213	60.65%
#SunakTheSnake	2000	432	1265	63.25%
#TheLastLeg	2000	443	1239	61.95%
#ClosingCeremony	5000	131	4406	88.12%
#onepiece	5000	267	4502	90.04%
F1	5000	254	4225	84.50%
#BookLoversDay	5000	301	4076	81.52%
#biden	5000	493	3675	73.50%
#GetBackToworkYouFatPonce	5000	88	4572	91.44%
#IOS16	5000	455	3854	77.08%
#taiwan	10000	706	7148	71.48%
#CostofLivingCrisis	10000	159	8668	86.68%
#EnergyPrices	10000	289	8812	88.12%

Table 2.2: Stats for others in our dataset

3. DATA OBSERVATION

In this section, I will make a preliminary observation on the collected data and introduce some interesting phenomena in our multi-source cases data set.

3.1. Retweetaholic

I observed 'retweetaholic' behavior in our dataset. In recent 1000 status of retweeters, the average number of tweets and replies are 82.69 and 196.15, however average retweets number is 675.02. We can define retweet frequency as retweet_num/ (retweet_num + tweets_num).In more details, 84.35% users' retweet frequency is higher than 70%; 66.17% users' retweet frequency is higher than 90%; 54.43% users' retweet frequency is higher than 95%. Therefore, we can assert that the vast majority of users in our dataset are retweetaholics.

3.2. The direction of information diffusion

As discussed in previous papers, information propagation is often from higher status users to lower status users. [25]. If this conclusion is true, predicting which informer will be retweeted in a multi-source case can be viewed as finding the most Influential user among informers.

In order to verify this conclusion, we counted whether the user is verified, the number of followes of the user, and whether there is a url in user's profile in the multi-source data set. These three factors can reflect users' influence. Statistics show that 91.61% retweets are from verified users to non-verified users; 71.49% retweets from users with more followers to less followers; 70.56% retweets from users have url in their profiles to users without url. According to these statistical data , we can assert that in our data set, information flows from more influential users to less influential users.

VI. FEATURE MEASUREMENT

1. FEATURES

Our aim is to find the key factors that influence retweet behavior when there are multiple informers. Therefore we need list as many factors as possible that may affect users' choices of information source. We divide the features into 3 categories: basic features, true or false features, interaction features. Next I will introduce these three types of features separately:

Basic Features: These features are extracted from users' profiles and widely used by many other works. They are all integer or float, so it is convenient to put these features into models. For example, the number of users' follower, number of days user has been created.

True or False Features: Features such as is user a verified user or are there any url in user's profile are true or false features. They are all boolean variables. For the convenience of calculation, we convert True to 1 and False to 0. True or False features can also be viewed as basic features. We individually list them as their type of data.

Interaction Features: These features are highly related to retweeting behaviors and relationship between retweeters and informers, but they are rarely mentioned in previous articles. This may be because previous studies always care one side of retweeting behavior: the author of tweets or the user who retweets. In this thesis, I define a mention behavior (retweet, reply, or @ in tweets) as an interaction and count retweeters' interaction with informers and vice versa.

The definition of these features are shown in the table 1.1:

Feature Name	Type	Description
UsM_deltaDays	int	number of days user has been created
UsM_statusesCount	int	number of status posted on Twitter since creation
UsM_followersCount	int	number of followers
UsM_favouritesCount	int	number of favourites on Twitter since creation
UsM_listedCount	int	number of list count
UsM_friendscCount	int	Number of friends
UsM_normalizedUserStatusesCount	float	user status count normalised by creation time
UsM_normalizedUserFollowersCount	float	user followers count normalised by creation time
UsM_normalizedUserFavouritesCount	float	user favourites count normalised by creation time
UsM_normalizedUserListedCount	float	user listed count normalised by creation time
UsM_normalizedUserFriendsCount	float	user friends count normalised by creation time
tweets_num	int	number of tweets in recent 1000 status
retweet_num	int	number of retweets in recent 1000 status
reply_num	int	number of reply in recent 1000 status
norm_tweets	float	tweets count normalised by time interval
norm_retweet	float	retweets count normalised by time interval
norm_reply	float	replies count normalised by time interval
interaction	int	times that the informer or infector mentioned the retweeter
frequency	float	interaction count normalised by time interval
retweeter_interaction	int	times that the retweeter mentioned informer or infector
retweeter_frequency	float	interaction of reweeter count normalised by time interval
is_verified	bool	If True, indicates this user is a verified Twitter User
has_url	bool	If True, indicates this user has at least one url on his/her profile
is_default	bool	If True,indicates this user uses default profile
is_followed	bool	If True, indicates the retweeter is also followed.

Table 1.1: Features description

2. TIME FEATURE

There are a lot of works that consider the relationship between retweets and time. [20] [26] [27] Besides, first source principle, which is an assumption that if a user get infected, the infector is always the first informer who tweeted the related hashtags earlier than other informers is widely used in previous studies. If this assumption is true, then time gap between the time an informer post the tweet and the time user infected may be an important factor that affect people's choice for information source.

We calculated the time gap between the informers post tweets and the infection time and time gap between the infectors post tweets and the infection time. If we count the infection behavior within 24 hours the average time gap between informers and infected time is 24341s, the median of time gap is 8533s; for infectors, the average time gap is 17343s and the median is 8805s. The time gap for informers is lower than the infectors in the median, while the mean is just the opposite. This is because a considerable part of the informers post tweets too early and near the 24 hours boundary.(as shown in figure1) If we limit the

interval between tweets to 12 hours, we found the average and median time gap between informers and the infected is 9396s and 5101s respectively; and for infectors is 11150s and 6558s. This phenomenon is inconsistent with our intuition, which people are more easily infected by newly released information.

In addition, we have studied in more details that the time gap between the retweeters and informers/infectors in different periods of a day. To avoid the effects of jet lag, we choose cases from United Kingdom trends, thus most of users are from same time zone. The result is shown in the figure 2.1, in which x-axis represents the hours in a day (0 to 23) user retweet (the infected time); y-axis represents average time gap (unit: hour) between infected time and the creation time of original tweets.

There are 17 hours within a day when time gap between infectors and retweeters is shorter than time gap between informers and retweeters. It indicates that people are easily infected by newly posted information. Besides, time gap in 12pm, 21pm, 22pm, 23pm is significantly shorter than other periods, which means people in those periods use twitter more frequently. These periods happen to be rest time for most people, which is inline with our expectation for user usage habit.

According to our observation, the retweeting behavior is highly related to time. At different time period of a day, time gaps between informers post tweets and the infected time show significant difference.

Unfortunately, there is a problem with adding time variables: we have no way to know the real time zone of the user, so it is possible to get wrong data because of jet lag. In our data set, the tweets we selected are all from UK Trends, so it is assumed that users are in UTC +1, but for unknown tweets, we cannot make such an assumption. Therefore, it may introduce wrong information due to the time zone, which may have a significant impact on forecast results. Therefore we decided not to include time as a feature in our model;

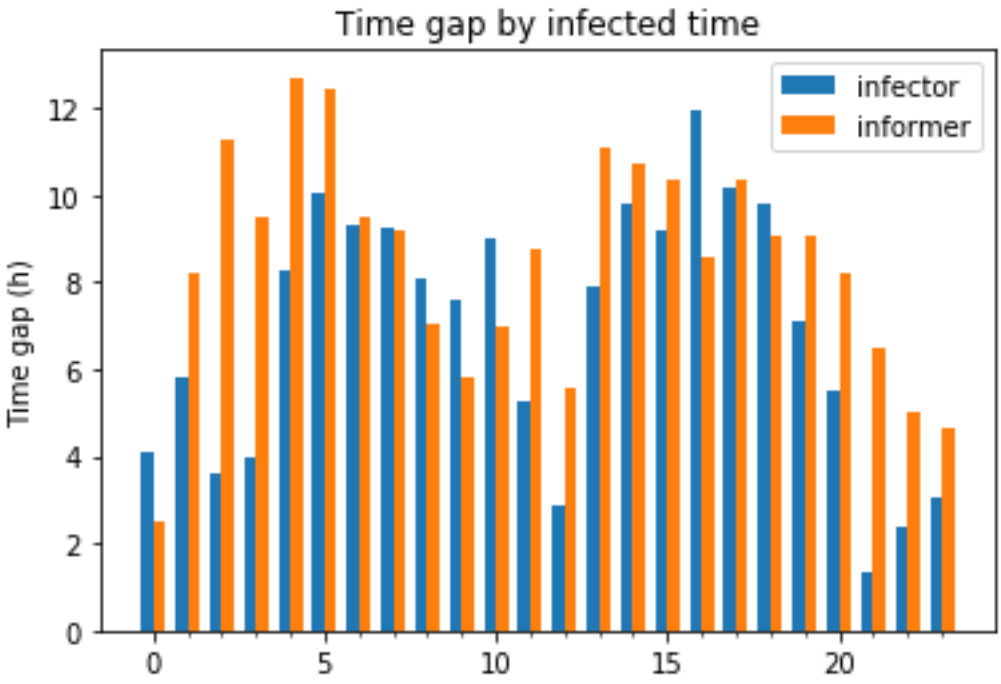

Figure 2.1: Time gap by infected time: x-axis represents the infected hour in a day; y-axis represents the average time gap between the informers post tweets and the infected time.

3. FEATURES CORRELATIONS

In this section, I will explore the correlations between features, as correlated features cannot improve the performance of models and increase the complexity of models. Therefore if two features we extracted are highly correlated, it would be better to only keep one.

As shown in the heat map (Figure 3.1), correlation between different features is all between -0.7 and 0.8, and most of them are between -0.3 to 0.3 so most of them are weak correlation [28]). We need put eyes on those features with correlation higher than 0.5.

Most of the highly relevant features are due to normalized, such as followers count and normalized followers count. The only set of features that interest us are norm_retweet and UsM_statusesCount. This phenomenon just proves that the proportion of retweetaholic in multi-source cases data set is very high. It is because for those retweetaholics, most of statuses are retweets, thus norm_retweet and UsM_normalizedUserStatusesCount are positively correlated.

After analyzing the correlations between features, for those features have

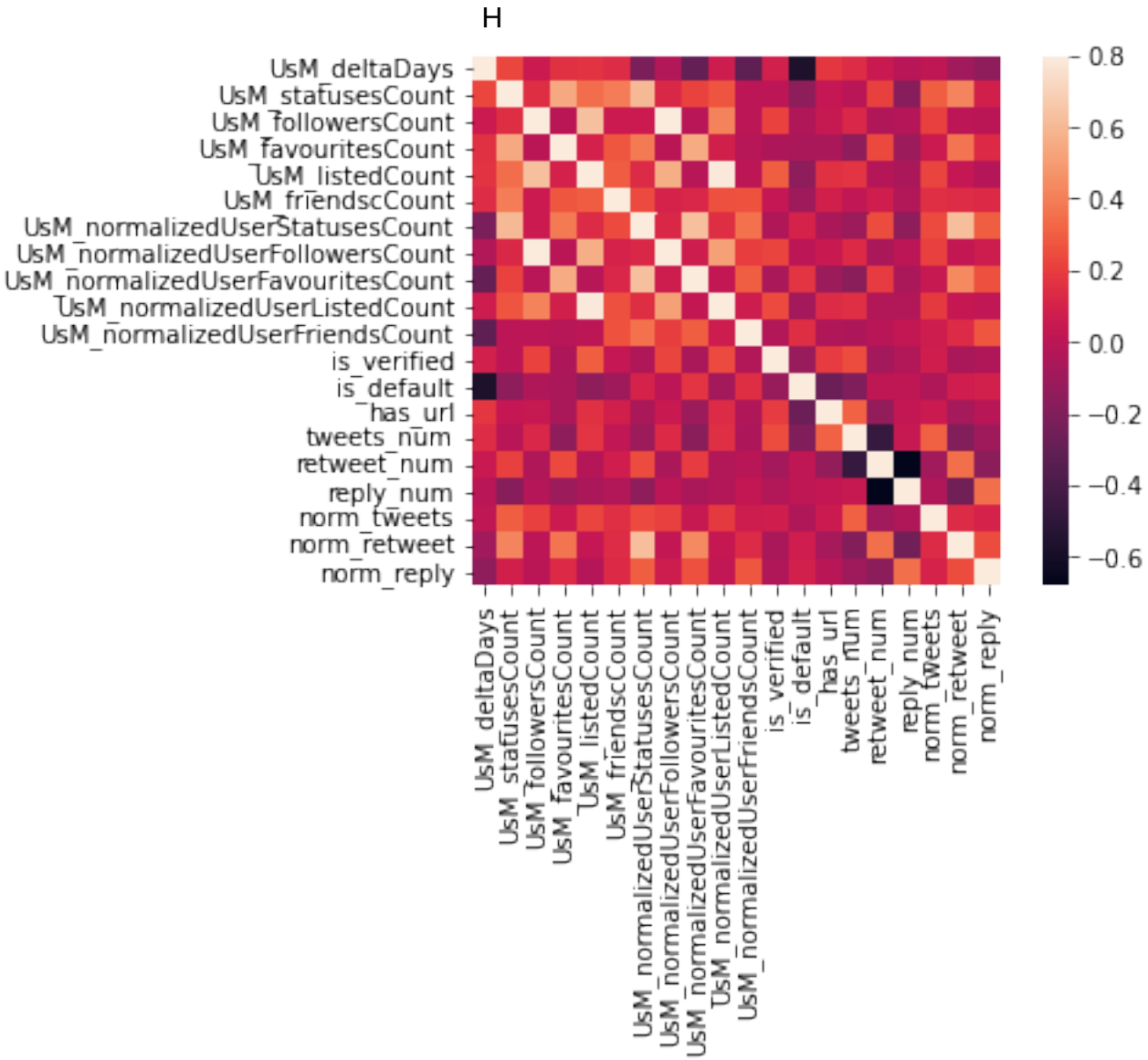

Figure 3.1: Correlation between features

strong correlation, we think they have their own values; In addition, because our data set is not too large, and the number of extracted features is far from the problem that makes us take too long under the existing computing power conditions, so I decide to keep all the features.

4. T-TEST

In this section, I will explore whether the features of informers and infectors come from same the distribution. To achieve this aim, T-test is a useful tool.

T-test is a widely used statistical test to compare the means of two groups of

samples. In this thesis, I use t-test to check whether each feature of informers and infectors is from same distribution. If a feature shows different distribution on infectors and informers, that feature could be the reason for users' choices. I use p-value as the indicator for judgement. If a feature's p-value is smaller than 0.05, we have sufficient reasons to reject the hypothesis that the feature in informers and infectors are from same distribution. In the table 4.1 below I have listed p-value for each features in ascending order.

Features Name	p-value
'UsM_normalizedUserListedCount'	7.8e-10
'tweets_num'	4.4e-09
'retweet_num'	3.0e-08
'UsM_listedCount'	4.8e-08
'is_verified'	5.7e-08
'retweeter_interaction'	0.00020
'UsM_normalizedUserFollowersCount'	0.00039
'UsM_normalizedUserFriendsCount'	0.00087
'UsM_friendscCount'	0.0010
'UsM_normalizedUserStatusesCount'	0.0014
'UsM_favouritesCount'	0.0029
'UsM_normalizedUserFavouritesCount'	0.0045
'is_followed'	0.0045
'UsM_statusesCount'	0.0046
'UsM_followersCount'	0.0057
'norm_tweets'	0.011
'retweeter_frequency'	0.15
'is_default'	0.20
'UsM_deltaDays'	0.21
'interaction'	0.24
'has_url'	0.32
'reply_num'	0.37
'norm_reply'	0.43
'norm_retweet'	0.87
'frequency'	0.91

Table 4.1: p-value (descending rank) when using t-test for different features of informers and infectors (keep 2 significant figures)

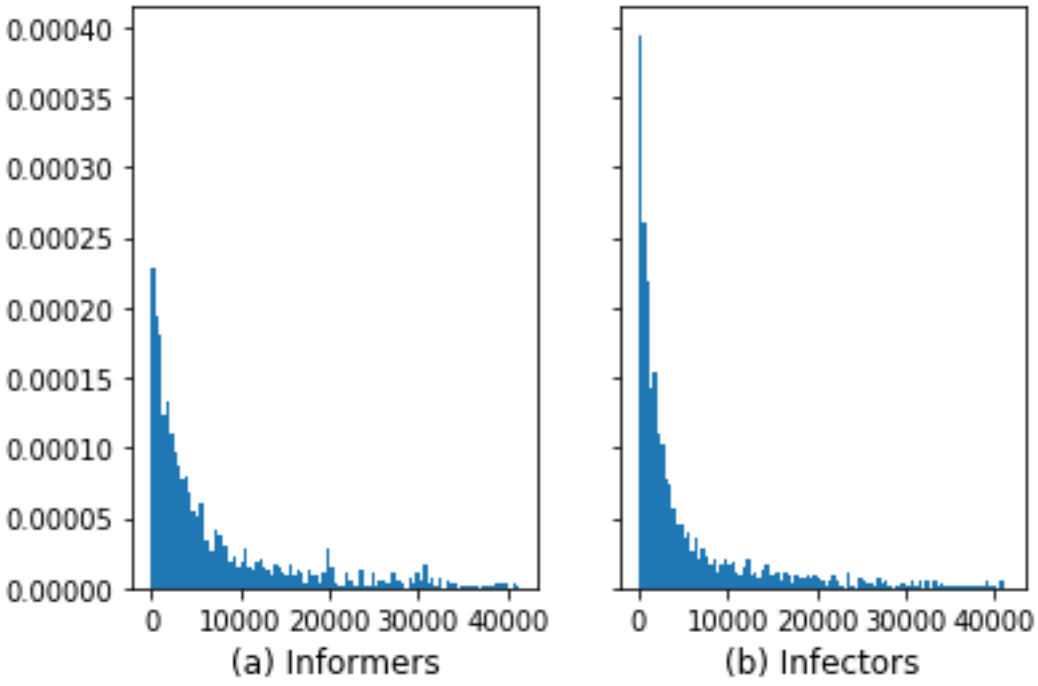

Figure 4.1: Distribution of UsM_followersCount: x-axis represents the UsM_followersCount and y-axis represents the proportion of UsM_followersCount in each interval

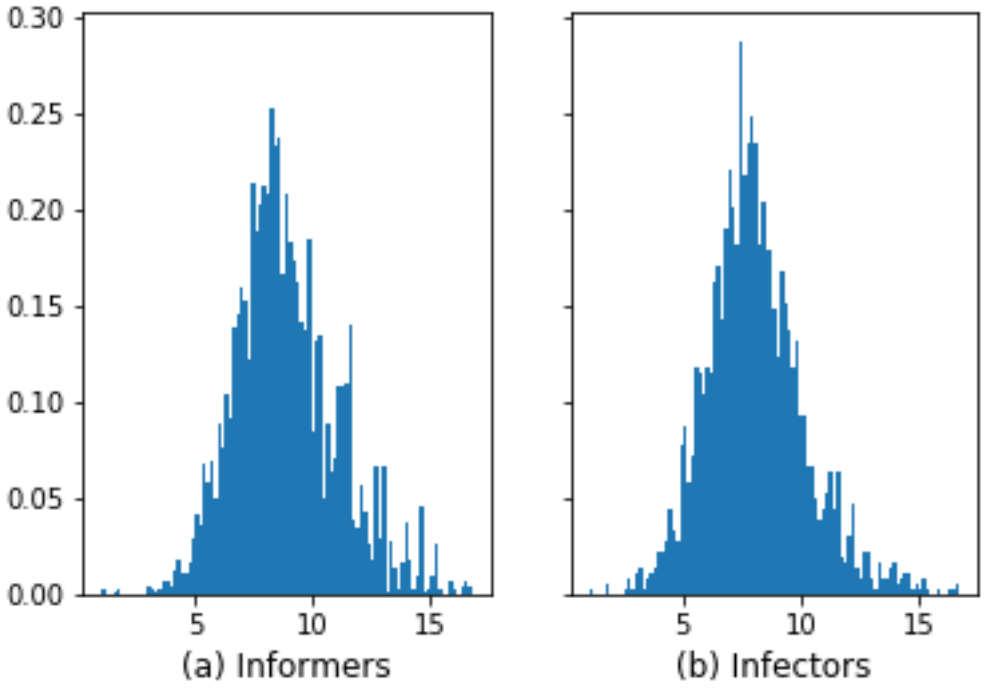

Figure 4.2: Distribution of logarithm of UsM_followersCount: x-axis represents the log(UsM_followersCount) and y-axis represents the proportion of log(UsM_followersCount) in each interval

By our observation, most of features obey exponential distribution. If we

take logarithm on features, they will conform to normal distribution.(As shown in Figure 4.1 and Figure 4.2) According to the assumption of t-test, if both of the two samples follows normal distribution, the results of t-test will be more convincing. Therefore I also do t-test for data after taking logarithm (results are shown in table 4.2).

Features Name	p-value
'UsM_followersCount'	1.1e-47
'UsM_normalizedUserFollowersCount'	1.0e-43
'UsM_listedCount'	2.1e-14
'UsM_normalizedUserListedCount'	2.6e-13
'UsM_statusesCount'	2.4e-10
'UsM_normalizedUserFriendsCount'	1.0e-09
'UsM_normalizedUserStatusesCount'	4.8e-09
'UsM_friendscCount'	6.92e-09
'retweet_num'	7.8e-08
'is_verified'	5.7e-08
'tweets_num'	4.1e-06
'norm_tweets'	1.6e-05
'retweeter_interaction'	2.1e-05
'UsM_favouritesCount'	0.00047
'UsM_normalizedUserFavouritesCount'	0.0015
'is_followed'	0.0046
'norm_retweet'	0.0075
'retweeter_frequency'	0.021
'UsM_deltaDays'	0.17
'is_default'	0.20
'has_url'	0.32
'frequency'	0.71
'interaction'	0.95
'reply_num'	0.97
'norm_reply'	0.99

Table 4.2: p-value (descending rank) when using t-test for logarithm of different features of informers and infectors (keep 2 significant figures)

As shown in the table 4.1, there are 16 features that exhibit different distributions on the two sets of data ($p<0.05$). If we take logarithm of two sets of data and then use t-test, we will find 18 features with p-value less than 0.05 (shown in table 4.2). "norm_retweet" and "retweeter_frequency" should also be considered as features that affect people choice of information source except the original 16 features.

5. RESULT

In this chapter, I have extracted informer and infector features (25 features in total, divided by basic features, True or False features and interaction features). Next I calculate the correlations between features and found the correlations between most of features are weak correlations. After checking correlations between features, t-test is used to check whether a feature of informers is from the distribution as the same feature of infectors, where p-value is used as indicator. There are 16 features come from different distribution, such as UsM_normalizedUserListedCount and tweets_num. According to our observation, most of features are from exponential distribution, so I take logarithm of them and do t-test again. After logarithmizationin addition to original 16 features, retweeter_frequency and norm_retweet are also judged to be from different distribution.

For time features, in the multi-source data set, we found: 1. time gap between a author posts a tweet and time that the tweet is retweeted follows exponential distribution, which means most of retweeting behaviors happens within few hours of original tweets. 2. the informer time gap is longer than infector time gap and time gap is related to user active time. However, if we hope to extract time features. we have to consider the effect of time zone. For example, an American post a tweet on 22 p.m. when his English friend is sleeping. When the English see that tweet, it has been a long time since the tweet was posted by the American. Due to the loss of timeliness, the English will not retweet. It’s hard for us to know the time zone in which the user sends the tweet. If the time feature is introduced rashly, data will be inaccurate. To avoid this indeterminate error I will not introduce time features in this thesis.

VII. MULTI-SOURCE PREDICTION

Multi-source Prediction is a problem of predicting which informer will be retweeted in a multi-source case.

In this chapter, firstly I will introduce the reasons I use different machine learning models to predict retweeting behavior in multi-source cases. Next. I will train Logistic Regression, Support-Vector Machines and XGBoost classifier on the training set and compare their performance on test set. Lastly, I will group the features and test the performance of our model based on different combinations of groups.

1. JUSTIFICATION

We can view the question that predict which informer will be retweeted as a binary classification problem. We concatenate the features of informer and retweeter as features for "informer-retweeter" pairs, whose label is whether this informer was retweeted. In the next section, I will train three general classification models: Logistic regression model, Support-Vector Machine model and XGBoost decision tree on the training set and test their performance on test set based on F1-score and accuracy. We use multiple models on the multi-source cases data set is to determine which model has a relatively good effect on solving the problem of distinguishing between the infecter and other informers. Besides, Logistic regression model can give the weights for features, where weights can represent the importance of features. XGBoost is a decision tree model, which is made up of nodes, each linked by a splitting rule. The splitting rule involves a feature and the value it should be split on. We can get a feature’s importance by adding all the weights of nodes that contain that feature.

Therefore , not only can the models tell us whether it is possible to predict which informers will be retweeted in multi-source cases, but also we will know the importance of each feature.

2. MODELS

2.1. Logistic Regression

Logistic regression allows the analysis of dichotomous or binary outcomes with 2 mutually exclusive levels [29], which is widely used by people for its simplicity, parallelizability, and strong interpretability.

Before we applied logistic regression model on our data, we need normalized data (map data between 0 and 1). This is because different features often have different units and dimensions, which will affect the prediction ability of our models. To eliminate the dimensional influence between the indicators, data normalization is a necessary step before training models, so that each indicator is in the same order of magnitude, which is suitable for comprehensive comparative evaluation. For every feature, we divide every data by the maximum value of this feature. In this case, we can compare the features' importance without considering dimension.

After applying logistic regression, we can get weights of features. The larger weight (absolute value) represents the higher importance of feature. The top 5 important features are:

Features Name	Weights
'UsM_normalizedUserFriendsCount'	-1.60
'r_interaction'	1.41
'norm_tweets'	1.21
tweets_num'	1.16
'UsM_normalizedUserListedCount'	-1.15

Table 2.1: Top 5 important features by Logistic Regression

The model performance is shown in the table 2.2:

Accuracy = 0.6182				
label	precision	recall	f1-score	support
0	0.91	0.63	0.74	592
1	0.15	0.51	0.23	372
avg	0.61	0.58	0.54	964

Table 2.2: Performance Metrics for Logistic regression

As shown in table 2.1 top 5 features are 'UsM_normalizedUserFriendsCount','r_interaction' ,'norm_tweets',tweets_num' , 'UsM_normalizedUserListedCount' . 4 of 5 most important features are informers' feature and only 1 of them belongs to interaction feature, which indicates the identity of the informer is an important factors in the multi-source prediction problem.

2.2. Support-Vector Machine

SVM is an efficient classifier which is widely used in machine learning problems. The aim of the algorithm is finding the maximum-margin hyperplane to divide data into two classes. In this thesis, we use SVM with rbf kernel. The results are shown in the table 2.3:

Accuracy = 0.6224				
label	precision	recall	f1-score	support
0	0.95	0.56	0.70	592
1	0.09	0.62	0.16	372
avg	0.62	0.58	0.49	964

Table 2.3: Performance Metrics for SVM

2.3. Xgboost classifier

XGBoost is a scalable end-to-end tree boosting system, which can achieve state-of-art results on many fields, such as spam classifiers, advertising system and fraud detection system. [30]

LambdaMART is the boosted tree version of LambdaRank, which is based on RankNet.In XGBoost library, we can implement this algorithm by setting the parameter "booster" = "tree". In order for our model to perform well on the multi-source data set, we need to conduct hyper-parameter tuning.

Hyper-parameter tuning

Our strategy for hyper-parameter tuning is keeping other parameters unchanged and then optimize the value of current parameter by grid search. 4-fold validation is used to evaluate the quality of parameters.

Max-depth is the parameter that control the maximum depth of a tree. In tree model, depth means complexity and if the value is set too big, overfitting may occur. We train the classifier by setting the max_depth to 1,3,5,7,9 respectively. The best value of max_depth is 5.

Gamma means the minimum loss reduction required to make a further partition on a leaf node of the tree. We will get a more conservative algorithm when we use a larger gamma.We train the classifier by setting gamma to 0,0.2, 0.4, 0.6,0.8 respectively. And the best value of gamma is 0.0.

Reg_alpha and **Reg_lambda** are L1 and L2 regularization term respectively. Increasing the lambda will reduce the complexity of the model and avoid overfitting. We find the best alpha from 10^{-5}, 10^{-2},0.1, 1, 10 and the best alpha is 1. We find lambda from 0.01, 0.1, 1, 10,100. And the best lambda is 100.

Result of XGBoost classifier

After getting best parameters, we get the prediction results as shown in table 2.4 and table 2.5:

Accuracy = 0.6846				
label	precision	recall	f1-score	support
0	0.71	0.83	0.76	592
1	0.62	0.46	0.53	372
avg	0.68	0.68	0.67	964

Table 2.4: Performance Metrics for XGBoost (depth = 5)

Accuracy = 0.6825				
label	precision	recall	f1-score	support
0	0.70	0.85	0.77	592
1	0.64	0.41	0.50	372
avg	0.68	0.68	0.67	964

Table 2.5: Peformance Metrics for XGBoost (depth = 4)

As shown in the above tables, 4-layer decision tree has similar accuracy and same f1-score as 5-layer decision tree. However, 4-layer decision tree model have stronger interpretability and better generalization ability, more importantly, the computation time of 4-layer tree is roughly half of 5-layer tree. Therefore, it is better to use 4-layer decision tree model in the future. The two figures contain too much information and it's very hard to elegant display them, so I put them in my Github the links are here: 4-layer decision tree and 5-layer decision tree.

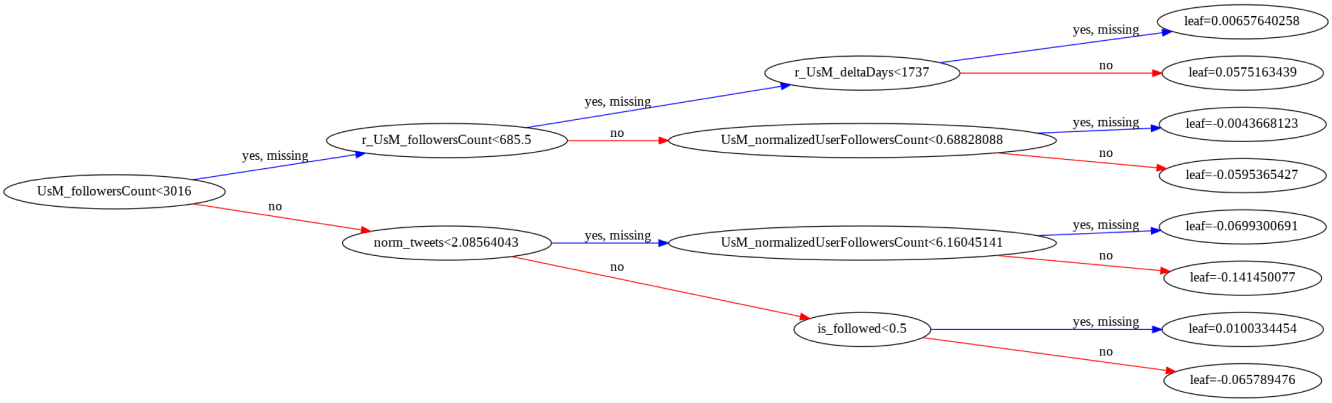

Figure 2.1: 4-layer decision tree

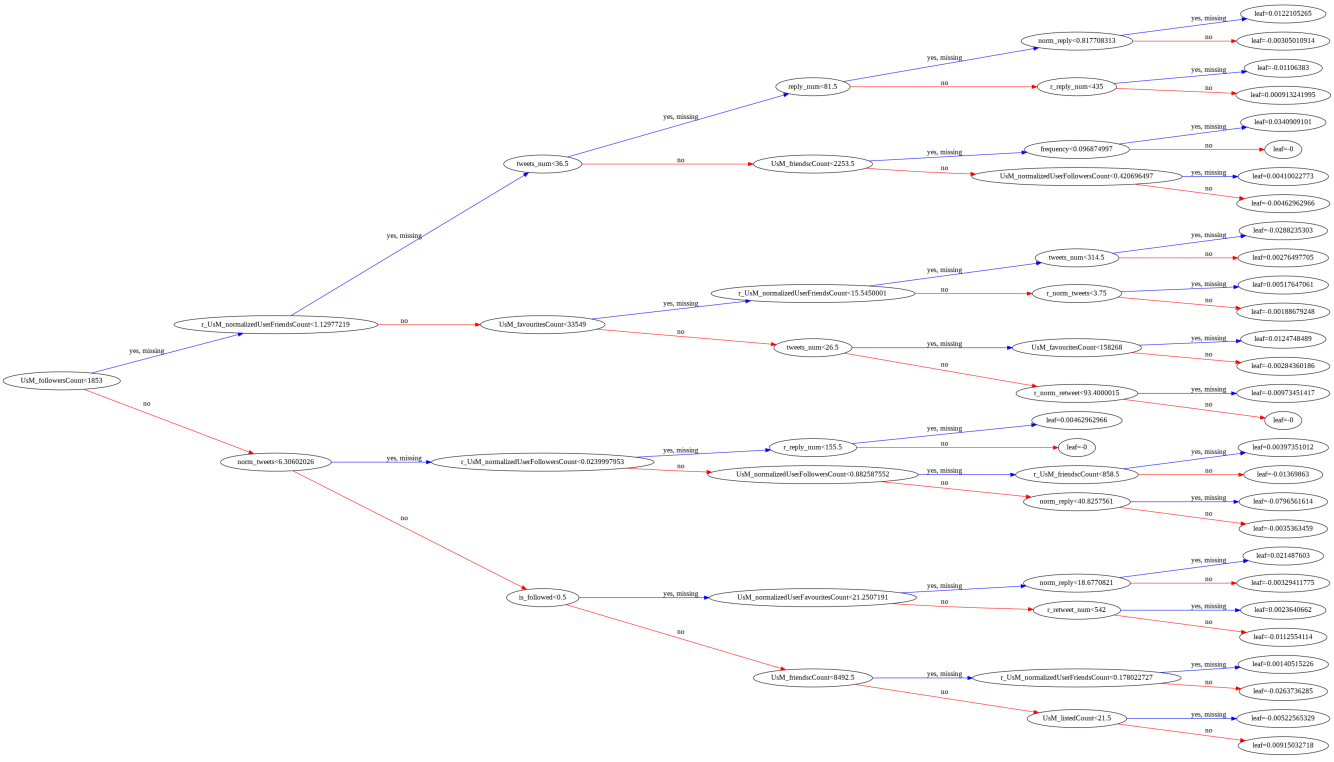

Figure 2.2: 5-layer decision tree

As shown in figure 2.1 and figure 2.2, 'UsM_followersCount' is the most

important feature; besides, 'norm_tweet' and 'r_UsM_followersCount' also play important roles in the prediction problem.

2.4. Results

In this section, we discuss and compare the effectiveness of the three methods in multi-source data set:

As shown in the table 2.6, XGBoost performs better than other two models in precision, recall, f1-socre and accuracy. Thus, in the task of predicting who is the infector in multi-source case XGBoost can get the best result among these three models.

method	ave precision	ave recall	ave f1-score	accuracy
LR	0.61	0.58	0.54	0.6182
SVM	0.62	0.58	0.49	0.6224
XGB(layer =4)	**0.68**	**0.68**	**0.67**	**0.6825**

Table 2.6: Comparison of the prediction results of LR, SVM and XGBoost

3. FEATURE IMPORTANCE

In the previous section I have compared three basic classification models and know that the XGBoost classier performs best on the multi-source data set. In this section, I will divide features into three parts: informer features (20), retweeter features (20) and interaction features (5), and explore which set of features contributes the most to this multi-source prediction problem. I select 6 kinds of feature combinations (only informer feature, only retweeter features, only interaction features, informer and retweeter features, informer and interaction features, retweeter and interaction features) to train the xgboost classifier. The results are shown in table 3.1.

Features	accuracy	f1-score
all	0.6825	0.66
informer	0.6546	0.63
retweeter	0.5985	0.51
interaction	0.6089	0.49
informer + retweeter	0.6701	0.65
informer + interaction	0.6670	0.65
retweeter + interaction	0.6037	0.52

Table 3.1: The performance of 4-layer XGBoost classifier based on different combination of features

We can get a relatively good prediction result using only informer features (with accuracy 0.6825, f1-score 0.63); but if we only use interaction features or retweeter features, the classification effect is similar to random guess. Besides, both interaction features and retweeter features can improve the classification ability of the model on the basis of the informer features; howerver, model based on the combination of interaction features and retweeters features doesn't perform much better than model based only on interaction or only informer.

The experiment results show: informer features play the most important role in the multi-source prediction question; interaction features and retweeter features play similar role. The most interesting point is: we found the 5 interaction features contain all the retweeters' information regarding to multi-source questions, which means we can predict which informer will be retweeted by the user when we know who the informer is and the interaction history between two users without knowing who the retweeter is.

Informer features	Retweeter features	Interaction features
UsM_deltaDays	r_UsM_deltaDays	interaction
UsM_statusesCount	r_UsM_statusesCount	frequency
UsM_followersCount	r_UsM_followersCount	retweeter_interaction
UsM_favouritesCount	r_UsM_favouritesCount	retweeter_frequency
UsM_listedCount	r_UsM_listedCount	is_followed
UsM_friendscCount	r_UsM_friendscCount	
UsM_normalizedUserStatusesCount	r_UsM_normalizedUserStatusesCount	
UsM_normalizedUserFollowersCount	r_UsM_normalizedUserFollowersCount	
UsM_normalizedUserFavouritesCount	r_UsM_normalizedUserFavouritesCount	
UsM_normalizedUserListedCount	r_UsM_normalizedUserListedCount	
UsM_normalizedUserFriendsCount	r_UsM_normalizedUserFriendsCount	
tweets_num	r_tweets_num	
retweet_num	r_retweet_num	
reply_num	r_reply_num	
norm_tweets	r_norm_tweets	
norm_retweet	r_norm_retweet	
norm_reply	r_norm_reply	
is_verified	r_is_verified	
has_url	r_has_url	
is_default	r_is_default	

Table 3.2: Three kinds of features: Informer features, Retweeter features and Interaction features

4. SUMMARY

In this section, we predict which informer will be retweeted in the multi-source cases by three classification models: Logistic Regression (LR), Support-vector machine (SVM) and XGBoost classifier (XGB). According to accuracy and f1-score, XGB performs much better than LR and SVM in our multi-source data set.

Although both LR and XGB can provide features importance, the outcomes of XGB are more reliable as XGB is a better classifier in our multi-source prediction problem. The top three important features given by XGB are 'UsM_followersCount' 'norm_tweet''r_UsM_followersCount', which are all show users' influence in social network. [2]

We have figured out "which features are relatively important" and next we will figure out "which kinds of features are more important". We divide all the features into three categories: informer features, retweeter features and interaction features and test the classification effect based on different combination of three kinds of features. In 6 sets of experiments, we found informer features are most important and the other two groups of features have similar effect. And another interesting finding is when we combine retweeter

features and interaction features, the model didn’t have better performance than use them alone. Therefore, we have hypothesis that in a multi-source case, we can predict which informer will be retweeted only based on who the informer is and the previous interaction between the user and the informer. This hypothesis is important as it can greatly reduces the dimensionality of the features that need to be extracted (from 45 features to 25 features in our thesis).

VIII. CONCLUSION

1. SUMMARY

First, we collect multi-source data set. To the best of our knowledge, we are the first team to collect such a data set. We collected a total of 83,000 tweets from 19 topics, including 10,099 multi-source cases. About $\frac{1}{8}$ tweets are multi-source cases.Through the observation of the data, we found that in the multi-source data set, the vast majority of users are retweetaholics. There are 55.43% of users whose retweets account for more than 95% of their all tweets; in addition, we also observed that information is always from higher status user to lower status users, which means from verfied users to non-verfied users or from users with more followers to users with less followers.

Secondly, we extracted 25 features for informers and calculate the correlations between features and found that correlations between most of features are weak. We also explore whether features of informers and infectors are from same distributions. According our assumption, if a feature is from different distribution on informers and infectors, it could be the factor that affects users' choice of information source. We also analyze time features in detail but because of lack of user location information, time features will not be added to our models.

Thirdly, logistic regression, support-vector machine and XGBoost classifier are used to predict which informer will be retweeted by the user in a multi-source case. XGBoost classifier performs much better than the other two models in this question. To figure out which kinds of features play the most important role in this question, I divide features into three parts: informer features, retweeter features and interaction features. According our result, informers are the most important part in multi-source cases. Retweeters and interaction have similar roles. We

can get a pretty good result if we only consider informer and interaction between informer and retweeter.

2. DISCUSSION ON FUTURE IMPROVEMENT

2.1. Contributions

We present a baseline of collecting multi-source data set and solving the multi-source prediction problem in this thesis. We prove using the XGBoost classifier and features we extracted can predict which informer will be retweeted in a multi-source case with f1-score 0.67, which is an acceptable result. Besides, we introduce interaction features into retweet prediction problems. We also prove that the combination of informer and interaction has the similar effect as the combination of informer and retweeter. However, the feature number of interaction is much fewer than retweeter.

2.2. Limitation

There are three limitations in our thesis:

Firstly, due to the rate limit of Twitter API, we cannot collect more data in the limited time. A larger data set would improve the persuasiveness of this thesis.

Secondly, we cannot get users' real location so the time zones for users are unknown. We cannot add time features to our model, although we believe they may benefit the models.

Thirdly, in multi-source cases, every user have 16 informers on average, so there is a serious data imbalance in our data set. Considering this situation, we select at most two informers for each user. But even so, we still have nearly 60% informers in our data set (with label 0). So our models have better performance on data with label 0. If we conduct data augmentation on data with label 1 and make data with label 1 have the same number as data with label 0, our models may perform better.

2.3. Future

For feature works, we need collect more cases and test our models on the bigger data set. Furthermore, it's necessary to group features in more detail. In this thesis, we only know informer features are most important but don't know which feature within informer features contribute most. With a more detailed classification, we can have a deeper understanding of the process of user selection of information source. Last but not least, in this thesis we only consider the effect of user and interaction, but ignore the content of tweets. Our hypothesis is that the classification of our models can be further improved after adding content features. So it is necessary to combine this work with the work of the other student in this project, who mainly focus on content part.

3. ANSWERING RESEARCH QUESTIONS

Now, let's call back three research questions we asked in the Project Plan chapter:

Q1: What factors will affect users' choice of source?

In the chapter Feature Measurement, we extract 25 features and use t-test to judge whether a feature comes from different distribution on informers and infectors. According to my hypothesis, if a feature show different distribution, this feature may affect users' choices. According to table 4.1 and table 4.2 in Chapter V, features with p-value less than 0.05 are seen as potential factors.

Q2:Is it possible to predict which informers will be retweeted in multi-source cases?

In the chapter of Multi-source prediction, we use XGBoost classifier to distinguish infectors from informers with f1-score 0.67, which means it is possible to solve Q2.

Q3:What factors have the greatest impact on users' choice of information sources?

In the section Feature Importance of Chapter VII, all features are divided into three parts: informer features, retweeter features and interaction between informers and retweeters. Experimental results show that f1-score reaches 0.63 only with informer features, and f1-score for only retweeter features and only

interaction features are 0.51 and 0.49 respectively. Obviously, informer (the author of a tweet) is the most important factor among the three parts. However, it should be noted that the content of tweets also affect users' choice of information source. In my thesis, I only consider the user and interaction between users. In order to solve this question, conclusions of the other student whose study focus on the content of tweets should also be considered.

REFERENCES

1. Weng, J., Lim, E.-P., Jiang, J. & He, Q. *Twitterrank: finding topic-sensitive influential twitterers* in *Proceedings of the third ACM international conference on Web search and data mining* (2010), 261–270.
2. Suh, B., Hong, L., Pirolli, P. & Chi, E. H. *Want to be retweeted? large scale analytics on factors impacting retweet in twitter network* in *2010 IEEE second international conference on social computing* (2010), 177–184.
3. https://datareportal.com/essential-twitter-stats (2022).
4. Letierce, J., Passant, A., Breslin, J. & Decker, S. Understanding how Twitter is used to spread scientific messages (2010).
5. Lee, K. *et al. Twitter trending topic classification* in *2011 IEEE 11th International Conference on Data Mining Workshops* (2011), 251–258.
6. Al-Taie, M. Z. & Kadry, S. in *Python for Graph and Network Analysis* 165–184 (Springer, 2017).
7. Li, M., Wang, X., Gao, K. & Zhang, S. A survey on information diffusion in online social networks: Models and methods. *Information* **8,** 118 (2017).
8. Han, X. & Niu, L. On charactering of information propagation in online social networks. *Journal of Networks* **8,** 124 (2013).
9. Ou, C., Jin, X., Wang, Y. & Cheng, X. Modelling heterogeneous information spreading abilities of social network ties. *Simulation Modelling Practice and Theory* **75,** 67–76 (2017).
10. Pastor-Satorras, R. & Vespignani, A. Epidemic spreading in scale-free networks. *Physical review letters* **86,** 3200 (2001).
11. Majmundar, A., Allem, J.-P., Boley Cruz, T. & Unger, J. B. The why we retweet scale. *PloS one* **13,** e0206076 (2018).

12. Xu, Z. & Yang, Q. *Analyzing user retweet behavior on twitter* in *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (2012), 46–50.

13. Lahuerta-Otero, E., Cordero-Gutiérrez, R. & De la Prieta-Pintado, F. Retweet or like? That is the question. *Online Information Review* **42,** 562–578 (2018).

14. Macskassy, S. & Michelson, M. *Why do people retweet? anti-homophily wins the day!* in *Proceedings of the International AAAI Conference on Web and Social Media* **5** (2011), 209–216.

15. Fu, X., Cheng, S., Zhao, L. & Lv, J. Retweet Prediction Based on Multidimensional Features. *Wireless Communications and Mobile Computing* **2022** (2022).

16. Wu, S., Hofman, J. M., Mason, W. A. & Watts, D. J. *Who says what to whom on twitter* in *Proceedings of the 20th international conference on World wide web* (2011), 705–714.

17. Lee, K., Mahmud, J., Chen, J., Zhou, M. & Nichols, J. *Who will retweet this? automatically identifying and engaging strangers on twitter to spread information* in *Proceedings of the 19th international conference on Intelligent User Interfaces* (2014), 247–256.

18. Yang, Z. *et al. Understanding retweeting behaviors in social networks* in *Proceedings of the 19th ACM international conference on Information and knowledge management* (2010), 1633–1636.

19. Wang, C., Fan, Y., Du, Y. & Sun, Z. Predict Individual Retweet Behavior Based on Multi-feature. *IOP Conference Series: Materials Science and Engineering* **790,** 012046 (Mar. 2020).

20. Liu, W. *et al.* Research on microblog retweeting prediction based on user behavior features. *Chinese Journal of Computers* **39,** 1992–2006 (2016).

21. Araujo, T., Neijens, P. & Vliegenthart, R. What motivates consumers to re-tweet brand content?: The impact of information, emotion, and traceability on pass-along behavior. *Journal of Advertising Research* **55,** 284–295 (2015).

22. Hoang, T. B. N. & Mothe, J. Predicting information diffusion on Twitter–Analysis of predictive features. *Journal of computational science* **28,** 257–264 (2018).

23. Sharma, S. & Gupta, V. Role of twitter user profile features in retweet prediction for big data streams. *Multimedia Tools and Applications,* 1–30 (2022).

24. Zhang, Y., Moe, W. W. & Schweidel, D. A. Modeling the role of message content and influencers in social media rebroadcasting. *International Journal of Research in Marketing* **34,** 100–119 (2017).

25. Cha, M., Mislove, A., Adams, B. & Gummadi, K. P. *Characterizing social cascades in flickr* in *Proceedings of the first workshop on Online social networks* (2008), 13–18.

26. Kupavskii, A. *et al. Prediction of retweet cascade size over time* in *Proceedings of the 21st ACM international conference on Information and knowledge management* (2012), 2335–2338.

27. Lee, K., Mahmud, J., Chen, J., Zhou, M. & Nichols, J. Who will retweet this? detecting strangers from twitter to retweet information. *ACM Transactions on Intelligent Systems and Technology (TIST)* **6,** 1–25 (2015).

28. Bound, J., Jaeger, D. A. & Baker, R. M. Problems with instrumental variables estimation when the correlation between the instruments and the endogenous explanatory variable is weak. *Journal of the American statistical association* **90,** 443–450 (1995).

29. LaValley, M. P. Logistic regression. *Circulation* **117,** 2395–2399 (2008).

30. Chen, T. & Guestrin, C. *Xgboost: A scalable tree boosting system* in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (2016), 785–794.