Reliability of Social Media Platform

Xiaotong Diao
Computer Science
University of Colorado Boulder
Boulder Colorado United States
xidi9223@colorado.edu

Yu-Lin Chou
Computer Science
University of Colorado Boulder
Boulder Colorado United States
yuch2589@colorado.edu

Vandana Sridhar
Computer Science
University of Colorado Boulder
Boulder Colorado United States
vasr6141@colorado.edu

ABSTRACT

The social media platform is the best marketing tool in this generation. If you can find the right people to promote your product on social media, you would reach out to the right customer base and create more profit. However, if you select a user with a lot of fake followers, it would be a waste of time. In order to help companies build a more reliable marketing list, we want to build a model which can analyze the user's Twitter account, find similarity between real followers and analyse the features of fake followers. Based on this we can estimate the percentage of fake followers of the user's account and leverage it as a reference for social media marketing.

In order to solve this issue, this paper presents a model to obtain user accounts that are influential to the public, with a focus on the middle-tier accounts that are more prone to high fake followers percentage. The project is aimed to estimate the percentage of the given account's followers that fit one or more of the buckets namely: spam accounts, inactive accounts, bot accounts and propaganda accounts. We would like analyze the fake follower estimate through several different features such as user profile contents such as name, number of tweets, URL issues, location issues etc. We have also considered features such as tweet contents, tweets duplication, retweet number and hashtag number into our analysis. We would be utilizing tools such as Twitter API for data querying, and the sklearn module to implement different algorithms for training the model and finally classify fake users. Lastly, we have used the Twitter Audit tool to test the correctness of our predictions.

INTRODUCTION

With the development of online social media, online advertising is becoming more and more popular nowadays. Companies tend to promote their new products through social media platform, like Instagram, Twitter or Facebook, so that advertisements could be broadcasted all over the world with a relatively lower cost. Companies would select a group of users who are relatively influential and let them advertise the new products. Basically, the selection is based on the number of followers and the advertising cost for each influential user. Highly popular celebrities such as Justin Bieber, Tom Cruise etc don't need their accounts checked for fake followers because most of their followers are real. One the other hand, middle-tire influential users leverage unethical methods to increase their follower and fan bases which would result in misleading marketing companies. So, a system to show the reliability of online social media platforms would aid companies in better decision making.

RELATED WORKS

There are two problem that we need to understand in order to explore the reliability of social media platform. One is for identifying middle-tire influential users, the other is for identifying fake followers.

1 Identification of Fake Followers

There are plenty of research about Identification of fake followers has been proposed. Most methods for identifying fake followers are based on data mining methodology, like feature engineering, feature reduction and neural networks.[1]

Wang and Wilson[2] used a clickstream dataset provided by RenRen, a social network used in China, to cluster user accounts into similar behavioral groups, corresponding to real or fake accounts. Viswanath[3] used groundtruth provided by RenRen to train an SVM classifier in order to detect fake accounts. Using simple features, such as: frequency of friend requests and fraction of accepted requests. Cresci[4] used a ground truth provided by Twitter; the data have been processed using two main approaches:

Single classification rules and feature sets proposed in the literature for detecting spammers.

Feature reduction can be used to reduce computational cost and noise. S. Adikari and K. Dutta [5] using PCA as a dimension reduction produce better accuracy results than using all the features without any selection. However, Viswanath[3] collected their data from three social networks to illustrate that normal user behavior is low dimensional. So feature reduction for identifying fake followers is still controversial.

Neural networks is becoming a more useful tools in classification problems. S. Adikari and K. Dutta [5] extracted the profile features using PCA, and then applied Neural Networks, and Support Vector machines to detect legitimate profiles.

Aso, there are several mature tools for identifying fake followers, such as TwitterAudit, Amazon Fakespot, Statuspeople, and Social Baker.[6] Some of these tools can even tell whether the account is a bot account or a spam account.

2 Identification of middle-tire influential users

There are several algorithms for identifying influential users. However, in order to compare the reliability of different social media platforms, we will concentrate more on middle-tire influential users like some rising Youtuber because fake followers wouldn't have a great effect on top influential users. Since Youtube is a growing industry and their followers count isn't much, we would like to analyze their data and see their composition of followers.

Since the identification of influential users is a very conceptual problem and the measurement on influence is defined by researchers, researchers have proposed methods from very different perspective. Influence measures can be computed by centrality-based algorithms, PageRank, topical-sensitive algorithms, and deep learning methods.[7] Centrality-based algorithms are based on social networks' topology and graph theory.For example, closeness centrality is based on the length of the shortest paths from a node *i* to everyone else, proposed by Haijian and White[8]. PageRank is a way previously for identifying the pages' relevance for search engine, and can be used as a measurement for influence. Topical-sensitive methods are based on natural language processing and sensitive analyzing of users profiles. And neural networks can also be used to find influential users in a social network.

SIGNIFICANCE & DIFFERENCE GIVEN PRIOR WORK

While the aforementioned work focuses on detecting fake followers through centrality techniques and topical sensitive algorithms, our project would be to analyse middle-tier influential users such as Youtubers by querying their tweet and user contents and understanding the importance of certain features in the process. Certain features provide more clarity in detecting fake followers and we would like to analyse the accuracy of our prediction model by highlighting/dropping selective features. In this way we can comprehend the features required to determine and differentiate fake/genuine followers for an account. We also intend to use multiple machine learning algorithms to potentially find the most optimum solution to the problem. The result from the above study would provide potential marketing companies a reliability analysis of social media platforms such as Twitter and this would further allow them to choose the right platform to showcase their ideas and projects.

The biggest difference between our work and prior work is that we are going to analyze the contents of the tweet. Since Twitter provide a new API to query the latest 5 tweets for the specific user, we can evaluate the activity time for the user and tweet contents. Another new feature that we would like to incorporate would be to create a benchmark statistic for Twitter in terms of fake followers percentage on account of unavailability of a comparison measure. This would help further researchers who show interest in social media analytics. We would also like to focus on evaluating accounts with lesser influential reach since they are accounts which are prone to have more fake followers.

METHODOLOGY

1 Problem formulation

We want to analyse the major large scale big data analytics platforms namely Twitter by estimating the fake followers percentage of several influential accounts in order to help companies choose the most reliable platform to advertise their products and ideas.

We are using the MIB dataset for the training purpose. Some questions that arise here are:

- How to choose the correct attributes which are correlated to fake accounts.
- How to choose an appropriate classification method to build the model.

Moreover, before the data training, we need to consider how to clean the initial data, how to integrate the data from the different resources, how to convert nominal ones into numeric ones, and how to remove the outliers.

2 Data querying

For the dataset collection, since we want to build our training model, we need a dataset including the information of fake followers and real followers. Although there's no dataset like that available online. We found a resource called MIB Dataset[9] which includes the information we want. It has a information of real followers and fake followers, including user profile information like banner or personal information, private or public user, the detail of their tweets content, how many tweets they produce, and what hashtag and url resource in their data. It is really helpful for us to build the training model since they include a lot of information of fake followers and real followers.

After getting the mIB dataset, we needed to decide which data attributes we are going to select in this project, the first is the user profile part, based on the normal user behavior of social media,more information contained in the user profile means more opportunities that the user may be real, so we checked the API provided by Twitter and found that we can see whether this user has banner, has profile, has url for their profile, since the url may contains other social media they are using, like Instagram, LinkedIn or Facebook, so if they have the url information, it would be more helpful for us to decide whether they are real user in Twitter, besides of the user profile setting part, we also want to get the followers and friends amount number, since more followers and friends means the content of this user generated are more interesting, or this user is famous enough to explore more different relationship and let other people follow his account. Figure 1.1 is an example of our user profile data query,

表格 1

user_id	user_name	has_url	has_banner	has_location	has_extended_profile	followers_amount	friends_amount
1244663707	GDFLSH	0	0	0	0	0	39
822561822356230100	pet_idle	0	0	1	0	13	40
104138787	vahid_r	1	1	1	0	224	115

Figure 1.1: queried data

However, we do not want to just analyze the user profile, it is the same as prior project and example. In our project we would like to dive into another data attributes which may play an important role while you are trying to figure out who is the fake followers, it is the tweet content generated by a specific user. From the tweets content, we can check whether this user is running this tweet and whether he is devoting his time into the social media platform.

First we should see the duplication, there are many fake accounts just generate same content everyday and try to the spam or marketing stuff. We need to find how many duplicate tweets this user has and put into our training model.

Second thing that we are going to analyze is the retweet number, a tweet would be retweet because other user agree with the content or because the content is funny or interesting. We also query that data into our model to see the total amount of retweet that this user has.

Third part is about the hashtag, if the tweet contains hashtag, it may include more information. so it would also be a good reference for us to verify fake followers.

We also analyze the total number of the tweets which only contains url. If the user only put a tweets with url. Then it may be an account which only use for marketing or advertising. Then it may not be a real user. Figure 1.2 is an example for the tweets content data querying.

user_id	total_tweet	total_retweet	total_duplicate	only_url_tweet	contains_hashtag_tweet
1244663707	5	0	0	0	0
822561822356230100	-1	-1	-1	-1	-1
104138787	199	877499	0	0	13

Figure 1.2: queried data

So far we have mentioned what kind of data attributes that we think are valuable for the analysis, but we want to select some Youtubers to do the test, we need to get the followers information from the Youtuber's twitter account, how can we get these data from Twitter? We did a research on twitter API and found the tools that we need. It basically require the information of the project purpose and then we can get the authorization from Twitter. After we have the authorization from Twitter, it would be able for us to make a HTTP request to Twitter and then fetch the data we want.

While querying the data, we faced one issue during the development, that is Twitter actually has a limitation on query time and query request. It only allows you to make a 15 request in 15 minutes. However, we want to write all the data into our csv file at once without making same command twice. We set an interval in the server and make it call the Twitter API request once a minute. This gives us a way to keep querying data without violating the twitter API rule.

3 Data cleaning

The MIB dataset we got contains a lot of different types of accounts and features. In order to make it feasible for model training, we cleaned the MIB dataset based on the feature we determined before.

Generally, we used fake followers, which contains about 3000 fake user accounts, and gennuin accounts, which also contains about 3000 real user accounts. By combining the two datasets, we got a balanced training dataset. There are two files called users.csv and tweets.csv in both folder. The user.csv file included user profile features, like user id, locations, has_banner, and etc. The tweets.csv file included all tweets tweeted by the user listed in the user.csv profile, and contained features like text, has_url, has_hashtag, and etc. The main job that we did was to extracted features we needed and combined all csv file together to form our

training dataset. Figure 1.3 shows the sample data for user.csv. Figure 1.4 shows the sample data for tweets.csv

id	name	screen_name	statuses_count	followers_count	friends_count
1502026416	TASUKU HAYAKAWA	0918Bask	2177	208	332
2492782375	ro_or	1120Roll	2660	330	485
293212315	bearclaw	14KBBrown	1254	166	177

Figure 1.3: sample for user.csv

id	text	source	user_id	truncated	in_reply_to_status_id	in_reply_to_user_id
593932392663912449	RT @morningJ	<a href="http://tapbots.co	678033		0	0
593895316719423488	This age/face	<a href="http://twitter.com	678033		0	0
593880638069018624	Only upside of	<a href="http://twitter.com	678033		0	0

Figure 1.4: sample for tweets.csv

The tool we used for dataset manipulation is pandas in python. Since pandas is easy to manipulate data in row and column. It's also allows custom functions to clean data. For example, we could use pandas to group tweets attributes by user id and join to the user.csv.

The main challenges in this process included tweets text analyzation and dataset rebalancing. First we analyzed each tweets about whether it only has url, whether it has duplicated and whether it has been retweeted. Then we group these attributes by user id and join the use.csv. However, after we joined the two csv file. There only 4000 accounts left, which is due to the private accounts existing in user accounts. What's more, the combined dataset is no longer balanced, because there are 3000 real accounts and only 1000 fake accounts. In order to make the training dataset for public users balances again, we deleted 2000 real accounts randomly and shuffle the whole training dataset. But we had to pay attention to the rebalanced training dataset, because the size of it is much smaller than that of public user, which has 6000 user accounts. So we created another cleaned dataset which only contained public users with only user profile attributes to eliminate the effect of size problem mentioned before.

The cleaned dataset contains 3 csv file listed as Table 1.1:

dataset name	**size**	**feature**	**function**
user_all.csv	6000	user profile	predict private user
user_public.csv	2000	user profile	comparison
tweet_public.csv	2000	tweets & user profile	predict public user

Table 1.1: cleaned dataset

So after data cleaning, we got the cleaned csv file, which could used as input for model training and analysis directly.

4 Data analysis

After obtaining the cleaned MIB datasets, we classified them as public and private sets. The public dataset included all of the tweets' contents along with the followers' data. The private dataset contained just the user's profile. We analysed each dataset separately. Since both public and private datasets had genuine and fake users in them, we separated the data and classified them into 4 categories namely: Public/real users, Public/fake users, Private/real users and Private/fake users. Those users who have an assigned label of 0 is classified as fake and the ones who have a label of 1 are genuine followers. We then analysed each of four datasets using Matplotlib's Histogram function. Histograms help analyse the underlying frequency distribution of a set of continuous data. Through the histogram graphs we understood the frequency of occurrence of certain features, how symmetric features are, identification of outliers and how skewed the data is. Through the graphs we understood that the 'has_banner' feature has a good priority for separating fake from genuine followers. The has_banner feature had a sharp column of zeros for fake followers and a column of one for genuine users. Features such as location, friends and followers were pretty skewed to one direction. None of the features exhibited symmetry.

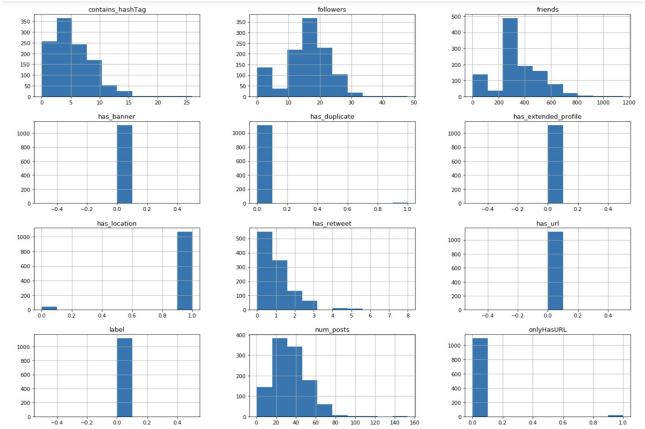


Figure 1.5: Histogram analysis of Public/Fake dataset

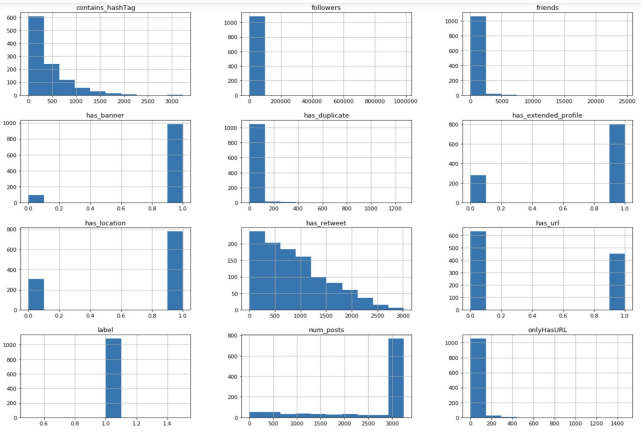


Figure 1.6: Histogram analysis of Public/real dataset

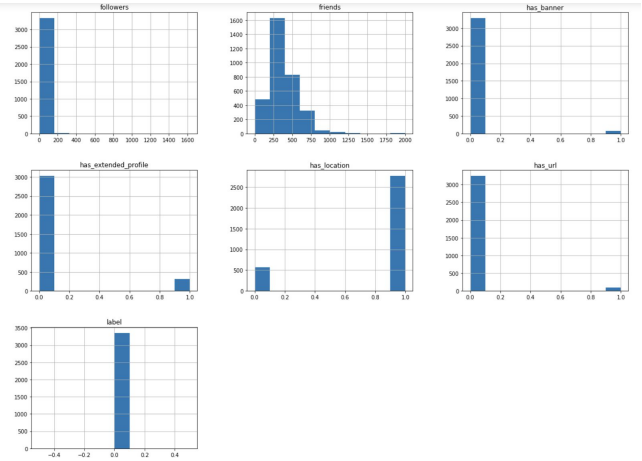


Figure 1.7: Histogram analysis of Private/fake dataset

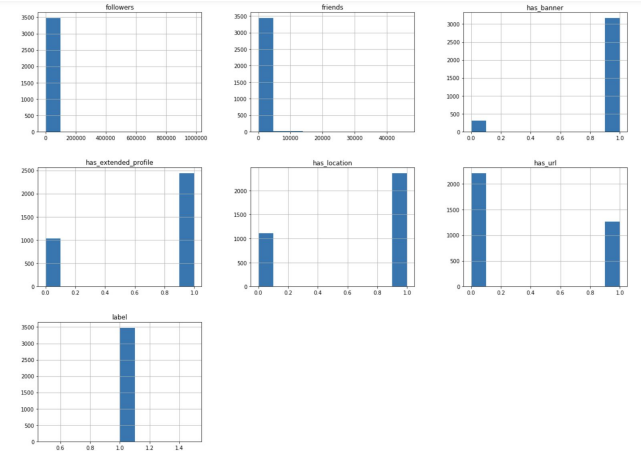


Figure 1.8: Histogram analysis of Private/real dataset

5 Model training

Based on the cleaned data from the MIB dataset, we trained a model to identify fake followers in Twitter. There are three steps for training and analyzing the model. First we need to prove that after adding tweets attributes, the model has a better performance than before. So that we could use the new model to predict public users. Socond, we need to select the proper machine learning model to predict our results. At last, we need to figure out the importance for each feature and prove that the features that we used are feasible.

The tool we used for model training and analyzation included sklearn, which encapsulates a lot of classic machine learning algorithms, and matplotlib, which is used to generate figures to analyze the training results more clearly.

5.1 Model analyzation

In order to figure out that whether the tweets attributes contribute to the model performance, we generated a figure to show the accuracy score under different test size. Each training process is iterated for 1000 times to reduce accidental error. Figure 1.5 shows the results of the accuracy score with different test size for three cleaned dataset.

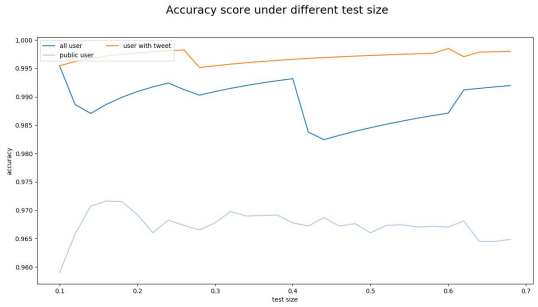


Figure 1.9: Accuracy under different test size

We could see that The public user has lower accuracy score than all user model. It's obvious that if we has less training data, the results would be worse. So after rebalancing the public user dataset, we got a worse performance. However, even if we only had 2000 data in tweet_public.csv, we still got a higher accuracy score than the model trained with all user accounts, which only included user profile attributes. So that we could say after adding tweets attributes, the model for predicting public user is better than before and these features contribute a lot to the model performance.

For the test size, actually 0.2-0.3 is usually a reasonable test size for most machine learning model. We could see that the model for public user adding tweets attributes perform higher at this test

size, which also prove our conclusion before. So for the following model training, we use 0.3 as the test size and included both user profile and tweets attributes to predict public users, and use only user profile to predict private user.

5.2 Model selection

This model mainly used seven machine learning algorithms: K-nn, Random Forest, Boosting, Bagging, Naive bayes, SVM, and quadratic discriminant. All these algorithms are good for classification problems..

We select K-nn(K nearest neighbors) because that it's easy to train, and can have a good performance on small dataset. Although it has some drawbacks, like not robust to noisy data and slow on large dataset. But our large-scale system will reduce these drawbacks by using spark to clean data in advance and scale training model among a cluster of nodes.

In addition to common machine learning algorithm, we mostly selected ensemble learning algorithm, including random forest, boosting, and bagging. This algorithms use multiple weak learner to create a strong learner, which improve the performance significantly. Random forest create multiple decision trees and is famous for its performance in classification problem, especially binary classification problem. Boosting make use of multiple different model, which avoid low performance of a particular model. Bagging make use of multiple same model, which avoid overfitting. In this case, we used K-nn for bagging algorithm.

We split the benchmark into two dataset:train dataset and test dataset. The ratio for train dataset is 70%. The model is trained with python and scikit-learn. We iterated the training process for each algorithms for 1000 times. The accuracy score for each algorithms is shown as Figure 1.6.

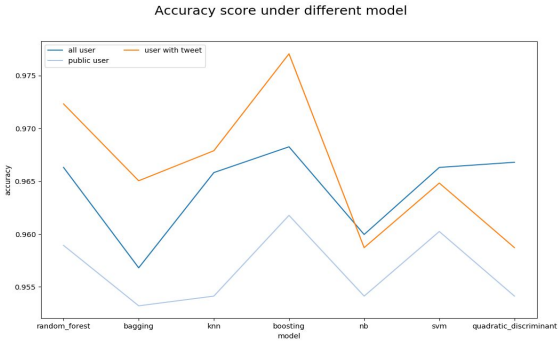


Figure 2.1: Accuracy under different model

We could see that the public user adding tweets attributes mode is still perform best for most algorithms. The random forest algorithm and the boosting algorithms perform better among all these classification model. And boosting has the best performance for both public user model and private user model. So we select bossing as our training algorithms in both public user model and private user model to predict the results.

5.3 Feature importance

Then we tried to figure out the importance for each features, and proved that the features we selected are feasible and useful.

First, we plot the accuracy score when we delete each feature. Figure 1.7 shows the results once we deleted the user profile feature for three training csv file.

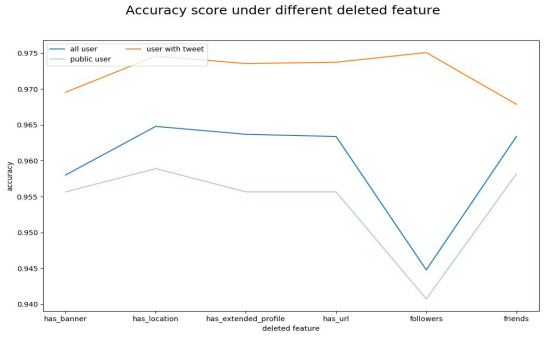


Figure 2.0: Accuracy under different deleted feature

We could see that once we deleted followers feature, the accuracy for models that only use user profile attributes fell a lot. But for model that using both user profile and tweets attributes, the accuracy score didn't fell a lot. That's because for private user model, the followers attribute is the most important attribute, but for public user model, it's no longer the most important feature.

Figure 1.8 shows the feature importance for both user profile and tweets attributes.

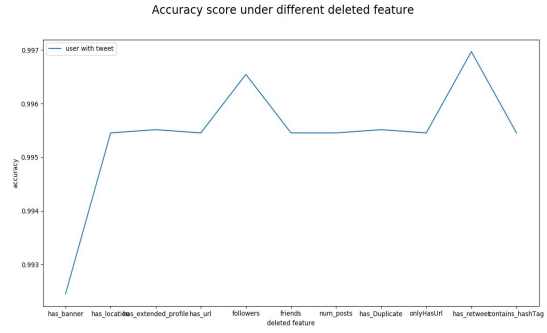


Figure 2.2: Accuracy under different deleted feature(adding tweets attributes)

We could see that even though the has_banner feature is the most important feature in public user model, all tweets feature still contribute a lot to the performance, because once we deleted each tweets attributes, the accuracy score also fell a lot. So all the feature we selected could contribute to our model and were used to predict the data we queried before.

EVALUATION

1 Dataset for training

The MIB dataset is labeled. So the evaluation is pretty easily. We used sklearn to train our model, and printed out the confusion matrix and accuracy score to see the performance of our model.

Basically, the sklearn package split the cleaned dataset into training dataset and testing dataset. Then it trained the model based on the training dataset and predict the results for each user item in testing dataset. Finally, it compared the results predicted with the labels in testing dataset and got the confusion matrix and accuracy score for each model.

The accuracy score has been shown in the model training part, which shows the performance for our model. Moreover, our model has high efficiency. Although we only have limited training dataset, it only takes less than 10s to analyze our model with 50 times iteration. So if we want to predict more than 1 million user account, it will only take about 30s to finish predicting. Although it seems like a long time, if we deploy our model on cloud computing platform and distribute training dataset to predict the results, it will only take seconds to get how much fake followers for a specific user.

2 Dataset for exploring

The queried data has no label. However, there are existing tools like SparkToro and HypeAuditor which could be used to test whether our model is correct and see the difference. We compared our results with the tools online and get a roughly understanding of whether our model is reliable.

RESULTS ANALYZATION

Once the model predicted the respective values for the queried data, we analysed the results using matplotlib. We again separated public and private data into real(1) and fake(0) and fed this to the Pie chart generator. The chart resulted in estimated percentages for real and fake followers. This was done for the five Youtubers we picked. Below shown is an example for a youtuber named "DeedsSonny". Here it says that 24.3% of the private users are fake and 75.7% of the private users are genuine followers of Deeds Sonny. Similarly 5.9% of the public followers are fake and 94.1% of the public followers are genuine.

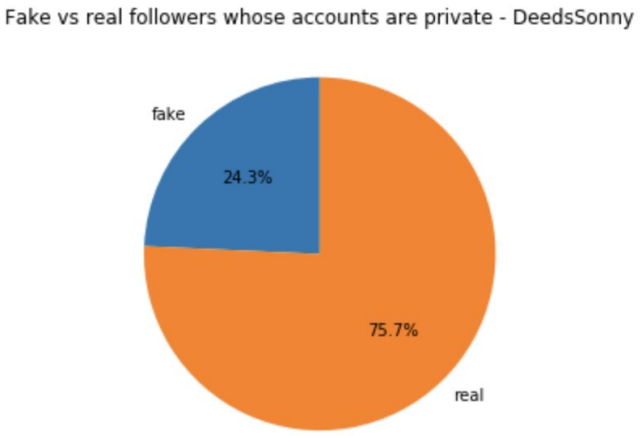


Figure 2.3: Pie chart estimating percentage of fake vs real followers for Deeds Sonny - Private dataset.

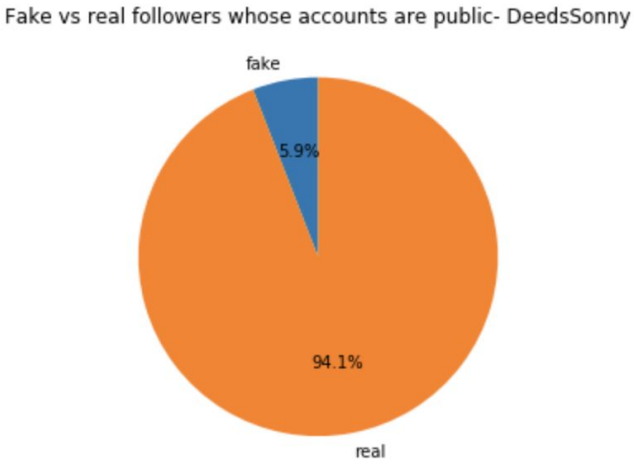


Figure 2.4: Pie chart estimating percentage of fake vs real followers for Deeds Sonny - Public dataset.

Youtuber	Public Fake %	Public real%	Private Fake%	Private Real%
Patrick Shyu	5.1%	94.9%	29.6%	70.4%
Living Bobby	2.1%	97.9%	24.0%	76.0%
Joma	3.3%	96.7%	28.4%	71.6%
BoldBebo	0.4%	99.6%	57.8%	42.2%
DeedsSonny	5.9%	94.1%	24.3%	75.7%

Figure 2.5: Estimated percentages of fake vs real followers for five Youtubers.

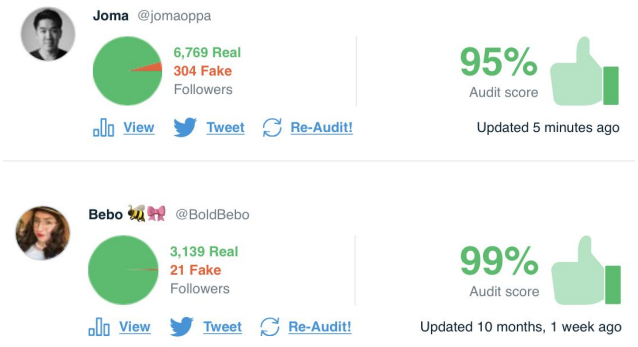
Next, we analysed certain features for each Youtuber's public and private datasets. For the public dataset we analysed the number of followers who have a banner, have a location, have an extended profile, have a URL, the average number of friends followers, the public followers had, the average number of posts they posted in their timeline. Apart from the above, we also peeked into the users who have duplicate tweets, those who only have URL as their primary features, the number of retweets and average number of hashtags the followers used. Since the private datasets only contain user profile data, we couldn't mine and analyse tweet contents.

From the tabulated results we found that fake followers of public accounts don't contain banner or extended profile, only some of them have location and URL, the average number of friends and followers are low in number. This could account to the fact that these accounts could be spam, bot or propaganda accounts only available in Twitter to raise the popularity of certain Youtubers. The average number of retweets and hashtags are significantly less. Genuine public followers have appreciable number of friends and followers, all of them have banners,location, extended profile, URL, posts and retweets and the model has correctly separated fake from genuine followers.

For private datasets, we couldn't really comprehend much difference between the fake and real followers since the model only operates on user profile data and doesn't take into consideration the tweets' data. One distinguishing factor between the two is the average number of followers for private fake users are low in number and they range between 20-30 whereas the private genuine followers have a massive follower rate ranging from 300-1500. The values under the private fake users are significantly lesser than the private genuine users.

Comparison with Twitter Audit

Finally we compared our model's results with the Twitter Audit tool. Twitter Audit takes a sample of up to 5000 Twitter followers for a user and calculates a score for each follower. This score is based on number of tweets, date of the last tweet, and ratio of followers to friends. Using these scores, the tool determines whether any given user is real or fake. This scoring method is not perfect but it is a good way to tell if someone with lots of followers is likely to have increased their follower count by inorganic, fraudulent, or dishonest means. We input the Youtuber's twitter names into the tool and the results are as follows:



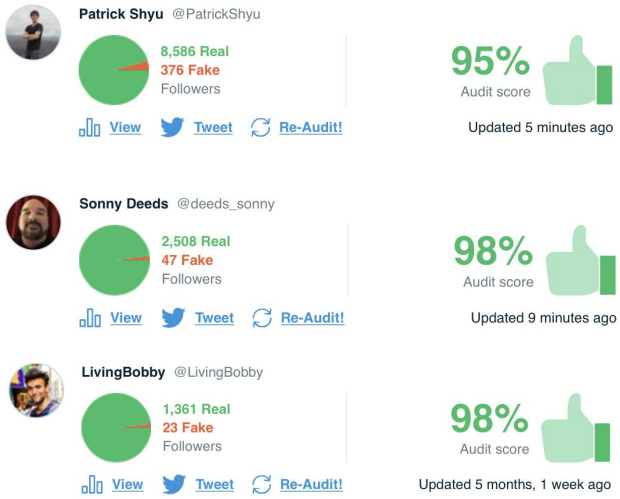


Figure 2.6: Twitter Audit score for several Youtubers

From this we conclude that our model provides similar results to the public datasets. It determines the same amount of fake/real followers as Twitter Audit does. This is because the model gets a holistic view of the users follower data. It takes into consideration both the user profile and the tweets content to do the analysis. One the other hand, Twitter Audit's results don't particularly match the private data results owing to the fact that private data has minimal number of features and the model doesn't take into consideration, the tweets' data of followers.

Twitter Audit additionally provides a quality score for the followers of a user. Twitter Audit scans followers and flags the ones that look low quality. The tool outputs a graph of the average quality score per follower. For example, in the below graph the tool provides a score of 0 for 26 followers of the Youtuber Living Bobby. This shows that 26 followers are 100% fake and they have low quality content in their profiles. The graph slowly descents and around 2-3 followers have a score of 0.9 which is slightly better than the 26 followers. Followers having 0.9 as their score might have lower red flags and might contain certain important features in their account such as location, banner etc. We performed similar analysis for each follower.

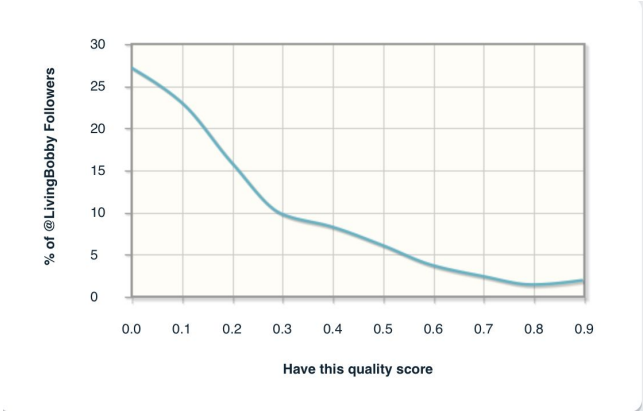


Figure 2.7: Twitter Audit graph for follower's quality score

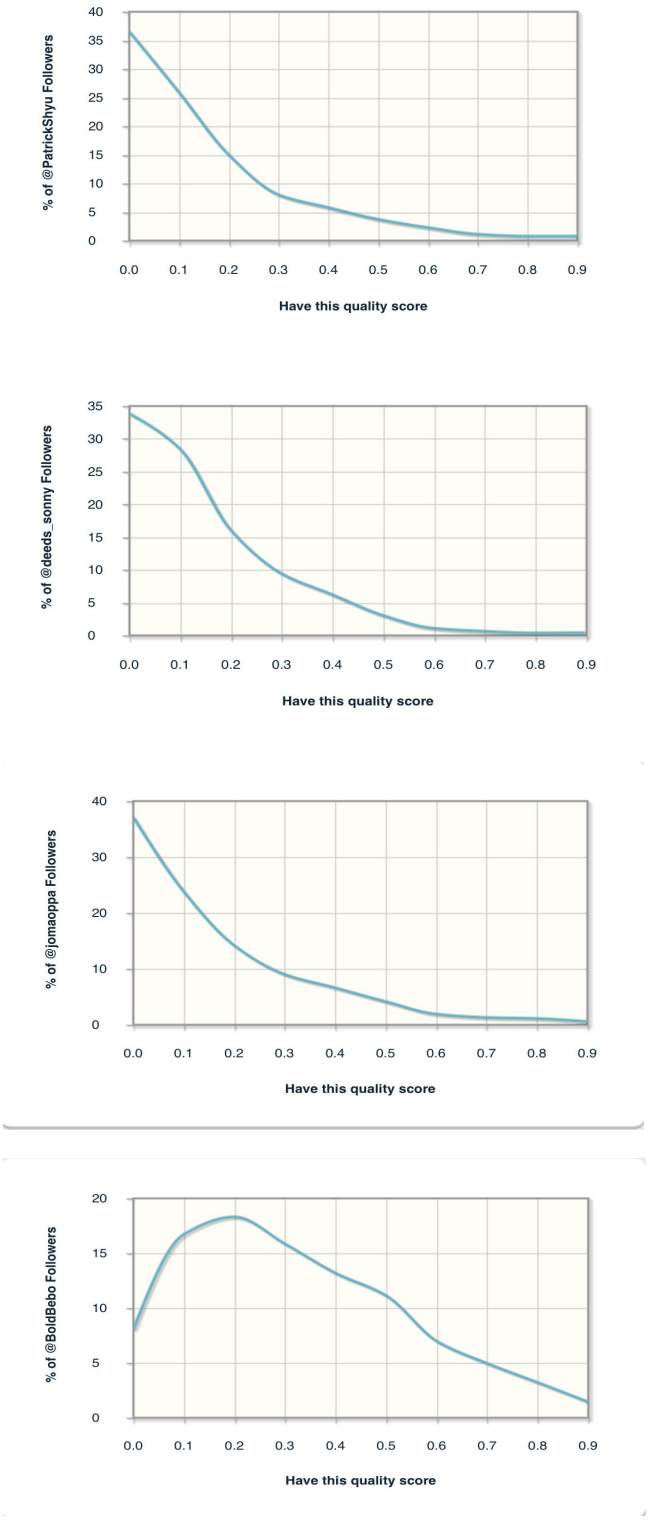


Figure 2.8: Twitter Audit quality score graph

In the case of the Youtuber BoldBebo, the quality score graph seems to raise and then fall. Our interpretation of this graph is that around 7 followers of BoldBebo are 100% fake, 17 followers seem to have a score of 0.2 which could mean that some of their features aren't void but could still seem suspicious to the tool. This could imply that these accounts could be Bot accounts which don't have much data but have enough to fool other users and increase the popularity of an account. Bot accounts are made for the sole purpose of following a particular user's account and promoting him/her. The graph declines from that point and around 3-4 users have a score of 0.9.

FUTURE WORK

1 Data quantity

In this project, we used MIB dataset to train our model. However, there are only 2000 data after we cleaned the dataset. And we may get better performance if we got enough data sample. Another problem is that because we can not get tweets data for private users, it limit the performance for the private user's model. So we need higher data quality for future improvement.

2 Scalability

Now our model runs on local machine, because the dataset is not large. However, if we are going to predict more user accounts, we definitely need faster speed. We could achieve this by deploying our code on AWS Sagemaker to distribute the training tasks. So that we could scale our project and gain higher efficiency.

3 Algorithm improvement

As the results shows, the predicting performance for private users is not as good as that of that for public users, because of lacking tweets attributes. So for private users, we may try different method like deep learning or social networks algorithms to get an equivalent performance as the public user's model.

REFERENCES

[1] Khaled, Sarah, Neamat El-Tazi, and Hoda MO Mokhtar. "Detecting Fake Accounts on Social Media." In *2018 IEEE International Conference on Big Data (Big Data)*, pp. 3672-3681. IEEE, 2018.

[2] G.Wang,T.Konolige,C.Wilson,X.Wang,H.Zheng,andB.Y.Zhao, "You are how you click: Clickstream analysis for sybil detection." in *USENIX Security Symposium*, vol. 9, 2013, pp. 1–008.

[3] B. Viswanath, M. A. Bashir, M. Crovella, S. Guha, K. P. Gummadi, B. Krishnamurthy, and A. Mislove, "Towards detecting anomalous user behavior in online social networks." in *USENIX Security Symposium*, 2014, pp. 223–238.

[4] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, “Fame for sale: efficient detection of fake twitter followers,” *Decision Support Systems*, vol. 80, pp. 56–71, 2015.

[5] S. Adikari and K. Dutta, “Identifying fake profiles in linkedin.” in*PACIS*, 2014, p. 278.

[6] Simon, Nitin T., and Susan Elias. "Detection of fake followers using feature ratio in self-organizing maps." In *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, pp. 1-5. IEEE, 2017.

[7] Riquelme, Fabián, and Pablo González-Cantergiani. "Measuring user influence on Twitter: A survey." *Information Processing & Management* 52, no. 5 (2016): 949-975.

[8] Hajian, Behnam, and Tony White. "Modelling influence in a social network: Metrics and evaluation." In *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, pp. 497-500. IEEE, 2011.

[9] Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, M. Tesconi. *WWW '17 Proceedings of the 26th International Conference on World Wide Web Companion, 963-972, 2017*