Predicting Influencers in Social Media

Aishwarya Mundhe
Dept of Information Science
Drexel University
Philadelphia, USA
aam453@drexel.edu

Anirbhan Das
Dept of Computer Science
Drexel University
Philadelphia, USA
ad3682@drexel.edu

Madhu Anumula
Dept of Computer Science
Drexel University
Philadelphia, USA
mla332@drexel.edu

***Abstract*—In this project, we are comparing two social media users and identifying which amongst them is more influential. The dataset contains the follower count and many other features of each user. By human judgement, the follower count is the only feature to categorize an user as influential or not but it takes more than just a high follower count to make an influencer. In this project, we consider other features to predict which user is more influential. The social media used here is Twitter. We are training the data using different models like Logistic Regression, Naive Bayes, Support Vector Machine and Gradient to predict which user is more influential. We also used techniques like cross validation and a combination of features to enhance the accuracies. We first used the in-built library for predictions and then built a few models from scratch to match or improve the accuracy. Finally, we compared the results we got by different models and different techniques and combinations of models and techniques and also talked about how this project can be helpful in the future.**

***Index Terms*—social influencers, logistic regression, cross validation, gradient descent, support vector machines 6**

I. Background

We have seen a huge rise in social media in the last decade. This rise has resulted in the birth of social media influencers. These influencers have established their credibility in different industries and have a huge following. Influencers can persuade their followers to buy products, donate to charities, follow brands etc. Their opinion also holds a weight among their followers. The followers tend to design their opinion and lifestyle by their influence. The more followers the influencer has, the more minds their opinions can reach. So it is important, in today's world, to know who is an influencer. But the number of followers is not the only feature that makes an influencer. For example, if user A has 30 followers and user B has 50, by just considering the followers feature, user B would be an influencer but when you compare user B to user C who has 100 followers, user B would not be an influencer. Hence, we cannot only rely on the number of followers. Hence, in our project we are considering more features to identify which user is a social media influencer. Dataset used is from Kaggle which is retrieved from Twitter. It includes 11 different features of Twitter users. Each instance of the dataset contains two users with their features and which user is the influencer. The target variable is a binary value with 1 being user A is the an influencer and 0 being user B is the influencer.

II. Related Work

There is a paper "Predict Influencers in Social Media" by a group of Stanford students who have used the same dataset. Using k-means they clustered the users into different classes like film stars. Then they performed Naive Bayes on these clusters. Apart from Naive Bayes, the other non-linear model they used was Neural Networks. They also used linear models like Logistic regression and Support Vector Machine. They set a benchmark by first calculating the accuracy by just using the number of followers the user has. The accuracy they got for that was 70%. Later they used different models that tried a combination of different features and surpassed that accuracy and got a 76% by Logistic Regression and SVM. They used hold-out cross validation and used 30% of the training set as validation set. They also split the data into k folds and picked the set with the smallest cross validation error. They also conducted feature selection. They used the forward selection algorithm to select the most relevant features to optimize the performance. Their results showed that cross validation had an effect on SVM but did not change or improve the test accuracy of Logistic Regression. SVM also performed better than Logistic with fewer features, 4 features giving the best result. Their linear models, Logistic Regression and Support Vector Machine performed better than non linear models with the highest accuracy of 76There have been various other papers using different datasets of different social media like Instagram. Most of these research show that they do not just depend on follower count but various other features.

III. Methodology

We have used the data from Kaggle, which consists of various pre-computed parameters received from the Twitter activity of two distinct individuals, A and B.

A. Data Processing

Before applying various algorithms to classify we wanted to pre-process the data. This helped us in achieving better accuracy. The first step was to delete all the duplicate datas that were present in the dataset. Moreover out of all the 22 features present in the data, we could see that some of the features were of least significance. We have removed those data as well at a later part of our coding.

B. System models

Our basic approach for this project was to try as many algorithms (specially those which were part of our curriculum) as possible on this dataset and to achieve an overall better result than the one obtained previously. We also wanted to explore some new algorithms which weren't part of our syllabus. The thought process behind that was to expand our learning curve. We started with 7 basic classifiers namely: 1. Logistic Regression 2. KNN Classifier 3. Naive Bayes 4. Support Vector Machine 5. Gradient Boosting Classifier 6. Random Forest Classifier 7. MLP Classifier Initially we implemented all of these algorithms using sklearn library. Among all of these classifiers, highest accuracy rate was achieved by Gradient Boost. It has an accuracy rate of 77.51 percent. The accuracy rate achieved by all these classifiers are shown in Table1.

TABLE I
ACCURACIES FROM STANDARD LIBRARIES

Model Name	*Accuracy*
Logistic Regression	0.74316
KNN Classifier	0.74261
Naive Bayes	0.56933
Support Vector Machine	0.5113
Gradient Boosting Classifier	0.7751
Random Forest Classifier	0.77005
MLP Classifier	0.70482

Post this we tried different other ways to achieve better results on accuracy, precision, recall and kappa value. We again ran all these above mentioned classifiers on standardized data. Then we tried them on normalized data. We then scaled each feature to a given range and tried the classifiers on them. We even performed function transformation and got the accuracy for all the above mentioned cases. Our aim was to try various methods and see which would give us the best results. It was evident from these trials that log function transformation provided us the best result using Gradient Boost Classifier. Our next aim was to get the feature importance. We plotted a graph for the same and found out that the columns ’A retweets sent’ and ’B retweets sent’ are the least important features.

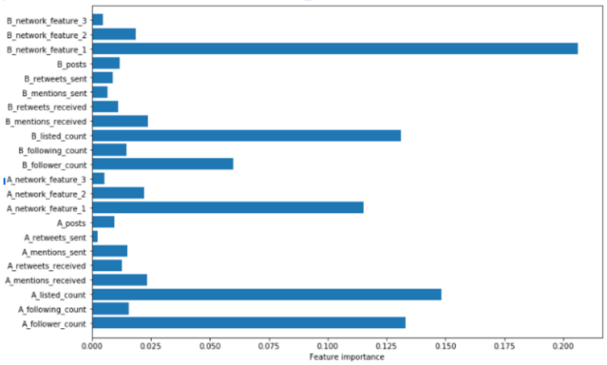

Fig. 1. Example of a figure caption.

So, we deleted those features from our datasets and calculated the accuracy once again. These did improve the result, but it wasn't that significant. We have done an algorithm comparisons and it's graph is shown below:

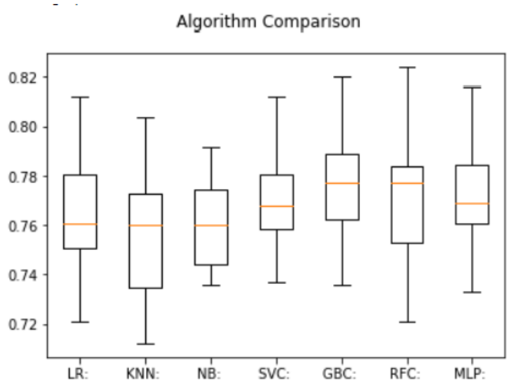

Fig. 2. Example of a figure caption.

But we didn't try to limit ourselves with the obtained results only. We thought of writing as many algorithms as possible from scratch, with the learnings from the class. Various algorithms which we tried from scratch on this dataset are mentioned below:

1) Support Vector Machine
2) Naive Bayes
3) Logarithmic Regression
4) Gradient Descent

1) Support Vector Machine: The main aim here was to form 2 divisions which satisfies the given equations:

$$w.x_i - b >= 1 \quad \text{if} \quad y_i = 1$$

and

$$w.x_i - b <= -1 \quad \text{if} \quad y_i = -1$$

When we combine these two equations it becomes:

$$y_i(w.x_i - b) >= 1$$

The gradient for SVM is calculated based on this:

$$\text{if} \quad y_i.f(x) >= 1 :$$

$$\frac{\partial J}{\partial W_k} = 2 * \lambda * W_k$$

$$\frac{\partial J}{\partial b} = 0$$

Else,

$$\frac{\partial J}{\partial W_k} = 2 * \lambda * W_k - y_i - x_i$$

$$\frac{\partial J}{\partial b} = y_i$$

The accuracy obtained with SVM (without k fold cross validation) is around 47%.

2) Naive Bayes: We applied Naive bayes on discrete data. Since Naive Bayes algorithm assumes that all features are independent of each other, this algorithm was one of our primary choices. The assumption that all features are independent makes naive bayes algorithm very fast compared to other complicated algorithms. In some cases, speed is preferred to higher accuracy. Moreover, it works well with high-dimensional data such as email spam detection. Naive Bayes is simple, intuitive, and yet performs surprisingly well in many cases. This lead us to go for this algorithm
The accuracy obtained with Naive Bayes is around 51%

3) Logarithmic Regression: We know that logarithmic regression is used to model situations where growth or decay accelerates rapidly at first and then slows over time. We use the command "LnReg" on a graphing utility to fit a logarithmic function to a set of data points. This returns an equation of the form:

$$Y = a + b.ln(x)$$

Here,

1) all input values, x, must be non-negative.
2) when b greater than 0, the model is increasing.
3) when b less than 0, the model is decreasing.

Also since the output was discrete classes of either 0 - not being influencer, 1 - being influencer, we have forced the computed output to belong to these classes using a threshold value (0.5), any prediction more that threshold is mapped to class 1 and less than threshold to class - 0. The accuracy obtained with Logarithmic Regression is around 71.2%

4) Gradient Descent: We wanted to use an optimization algorithm and implement it from scratch. As gradient descent is used to minimize some function by iteratively moving in the direction of steepest descent as defined by the negative of the gradient we thought it would be a perfect fit. The three main steps used in gradient descent are:

1) Given the gradient, calculate the change in parameter with respect to the size of step taken.
2) With the new value of parameter, calculate the new gradient.
3) Go back to step (i)

The accuracy obtained with Gradient Descent is around 70.4%

IV. Experiments and Results

The dataset has 5500 instances and 23 attributes. Each instance has features of User A and features of User B and the target variable which tells whether A is an influencer or if B is. This target variable is called Choice and has binary values in which 1 stands for User A is more influential than User B and 0 is for User B is more influential than User A. Each user has 11 features, namely : follower count,

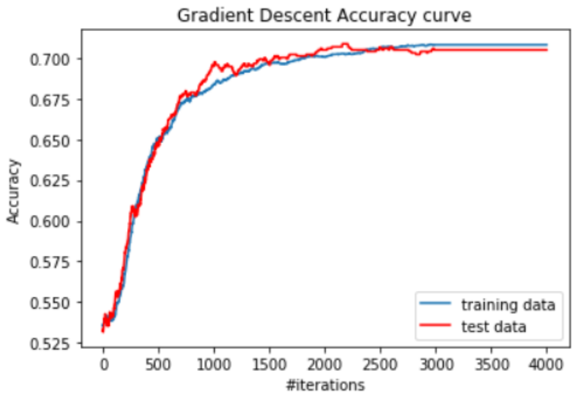

Fig. 3. Accuracy plot of Training and testing datasets using Gradient Descent

following count, listed count, mentions received, retweets received, mentions sent, retweets sent, feature 1, feature 2 and feature 3. Each instance has these attributes for both User A and User B which makes a total of 22 and the target variable. All of the features, apart from the target variable have numeric non negative values. The follower count shows the number of the accounts following the user. There is a good distribution for follower count ranging from 16 followers to 36543194 followers. Follower count plays a significant role but it is not the sole feature the prediction can be based on. The mentions and retweets main twitter functions by which the users communicate on the public front. The listed feature is an interesting one which shows how many accounts have added this user to their private lists of accounts they are interested in and want to see more of. All the features have a huge range and vary a lot. Using these features resulted in accuracies as low as 50%. Hence, we standardized the data to make it easier to deal with. This improved the accuracy. For cross validation, we tried different folds but the modifications did not change the accuracy by much. Comparison of various classifiers with changing data is depicted in table II.

Statistics for classifiers which were executed from scratch are shown in table III.

TABLE II
COMPARISON OF VARIOUS CLASSIFIERS WITH CHANGING DATA

	Raw Data	*Normalized Data*	*Standardized Data*	*Min Max Scaler*	*Function Transformer*
Logistic Regression	74.16	75.19	73.24	69.09	76.39
KNN Classifier	74.26	73.19	71.32	69.19	75.32
Naive Bayes	56.9	74.3	55.76	55.76	75.77
Support Vector Machine	51.13	74.44	73.63	65.81	77.03
Gradient Boosting Classifier	77.51	7719	76.95	76.88	77.53
Random Forest Classifier	77	77.01	76.59	76.41	77.24
MLP Classifier	70.48	75.58	74.43	74.49	76.71

TABLE III
STATISTICS FOR CLASSIFIERS WHICH WERE EXECUTED FROM SCRATCH

	Logarithmic Regression	*Gradient Descent*	*Support Vector Machine*	*Naive Bayes*
Without cross validation	70.3	70.4	51	47
With cross validation	71.2			

V. CONCLUSIONS

We can conclude that apart from the algorithms mentioned in the paper, Gradient Boosting Classifier gave us one of the best accuracy. Moreover, we have noticed that when we have less data, cross validation gives us better results. And when we tried running classifiers from scratch, without preprocessing data, we got accuracies around 50

VI. FUTURE WORK/EXTENSIONS

This project can further be extended as a marketing tool to find which influencers can be targeted for advertising different products. With so many influencers in the market, it is important to target the right ones for a stronger response. It could be done using the tweets of these influencers to conduct NLP techniques and find what the influencer is popular for and then cross match them with the products for advertising.

It can also be used to make sure the influencers are not spreading false information or advertising harmful products. Influencers with most underage followers can be restricted from marketing products like harmful weight loss products, harmful drinks, inappropriate language etc.

References

[1] KushalDave, Rushi Bhatt, Vasudeva Varm, “Identifying Influencers in Social Networks“,

[2] Ruishan Liu, Yang Zhao and Liuyu Zhou, “Predict Influencers in the Social Network“