Predicting Number of Likes On Instagram

Jiahua Chen, Jennifer Liu, Glenn Xu

Motivation

In the past decade, the amount of internet users has increased rapidly, replacing other traditional forms of media and entertainment with a skyrocketing trend among all ages known as social media. Specifically, social media is a website or application that enables users to create and share content or to participate in social networking, which is the form of communication through social media. With social media capturing the public's attention, as many as 65% of American adults use social networking sites.

Among the many social media applications, Instagram, released in 2010, grows quickly and currently has become one of the most popular ones in the US. As a photo and video-sharing social networking service, it allows users to upload photos and videos to the service, which can be edited with various filters and organized with tags and location information. Users are able to browse other users' content by tags and locations, and view trending content as well. Users can also "like" photos and follow other users to add their content to a feed.

Specifically, many people have started to view the likes they receive for the posts as a major measurement of their personal influence and even their success in social interactions. As a result, as more and more people pay attention to how many likes they may receive for each post, they have always been looking for a program that would do this job for them, which would take their past posting information and make predictions about the upcoming ones. Consequently, our group decides to take on this quest and utilize the innovative machine learning knowledge that we acquired in the 349 course to build a program for this purpose.

Dataset

We have written a web crawler based on Selenium to get data from Instagram. Throughout the project, we collected 2285 data points for building the model. Specifically, each data point includes 10 attributes and 1 label, including:

Attribute:

(1) the number of posts associated with the account
(2) the number of followers associated with the account
(3) the number of other accounts that the account follows
(4) if there is a location associated with the post
(5) for each post, the number of people "ated" in the post
(6) for each post, the number of hashtags in the post's description
(7) for each post, the total number of posts of hashtags in the post's description
(8) for each post, the total number of followers of the accounts "ated" in post

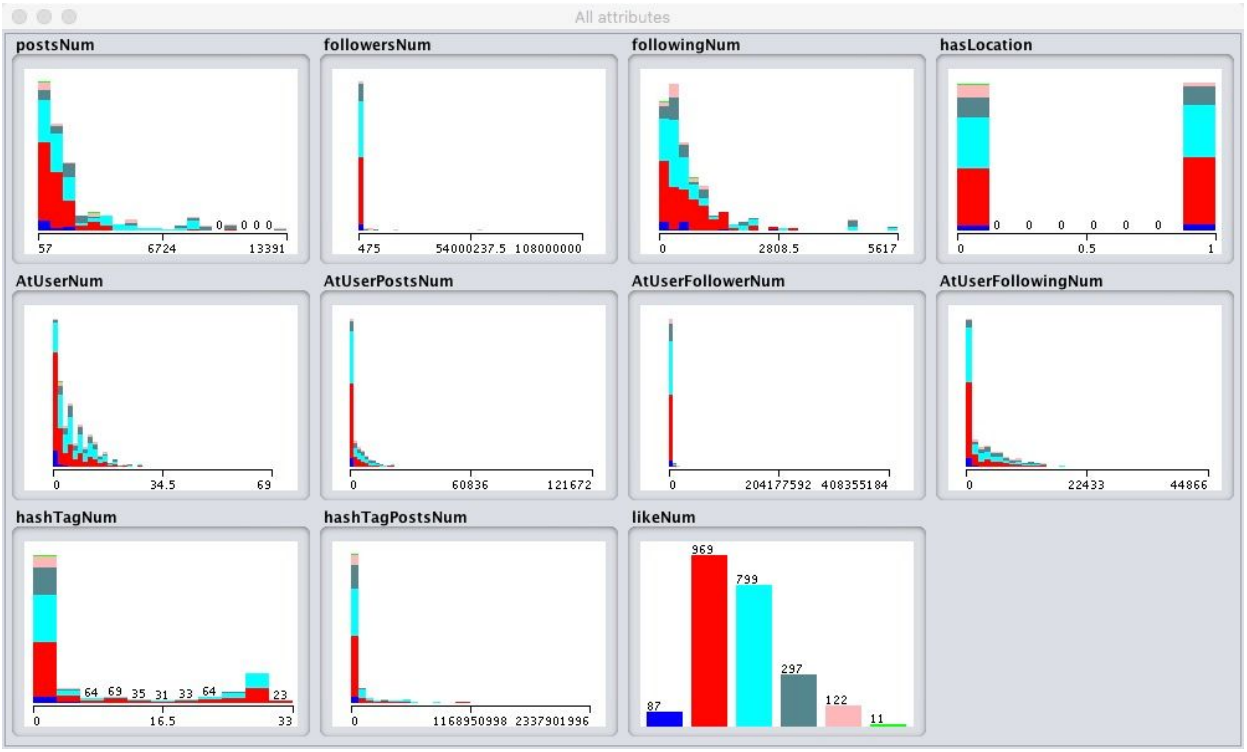
(9) for each post, the total number of followings of the accounts “ated” in the post

(10) for each post, the total number of posts of the accounts “ated” in the post

Label: the number of likes received for the post

Before doing the training, we processed the data in various ways, including categorizing the number of likes from the raw numbers to 6 categories using log10 (class 0: 1-9 likeness; class 1: 10~99 likeness; class 2: 100~999 likeness; etc.). We used all the data to train the model and 10-fold cross validation to validate the model.

The following figure shows the distribution of the attributes and label from the dataset.



We can see from this figure that overall the number of likes forms a distribution close to normal distribution. Most posts from our dataset have 10 to 99 likes or 100 to 999 likes.

Training and Testing

After getting the data from our web crawler and conducting the preprocesses, all the data were put into a csv file. This file was then loaded into Weka to train using different models via 10 fold cross validation. (Please see the result section below for details on the models used and the results.)

Result and Analysis

We trained our dataset with 43 models from Weka. The accuracy result is listed in the table below.

Algorithm	Accuracy
Filtered Classifier	81.53%
Bagging	81.27%
Decision Table	80.83%
Random SubSpace	80.70%
Attribute Selected Classifier	80.53%
OneR	80.48%
Classification Via Regression	80.44%
BayesNet	79.87%
Random Forest	79.69%
JRip	79.56%
J48	79.52%
REP Tree	79.21%
Random Committee	79.17%
PART	78.47%
LMT	77.81%
Iterative Classifier Optimizer	77.77%
LogitBoost	77.77%
Random Tree	75.62%
Simple Logistic	72.74%
Logistic	72.47%
IBK	72.21%
Multi Class Classifier	70.15%

Randomizable Filtered Classifier	67.97%
Multilayer Perceptron	64.38%
LWL	63.46%
Ada Boost M1	63.46%
Decision Stump	63.46%
SMO	52.91%
KStar	48.27%
Multi Class Classifier Updateable	43.33%
Naive Bayes Multinomial Text	42.41%
CV Parameter Selection	42.41%
MultiScheme	42.41%
Stacking	42.41%
Vote	42.41%
Weighted Instances Handler Wrapper	42.41%
Input Mapped Classifier	42.41%
ZeroR	42.41%
Hoeffding Tree	39.34%
Naive Bayes	34.35%
Naive Bayes Updateable	34.35%
Naive Bayes Multinomial	9.80%
Naive Bayes Multinomial Updateable	9.80%

As we can see from the table, ZeroR produced an accuracy of 42.41%. The model with the highest accuracy is Filtered Classifier with accuracy of 81.53%. Filtered Classifier is 39.12% better than ZeroR given this dataset.

Future Steps

After the meeting with the professor for the status report, our group has collected more data. At the same time, due to the updates on Instagram, we updated the web crawler and improved its performance as well. We have also updated the preprocess modules and successfully applied many more additional machine learning algorithms on the data to achieve a much prominent accuracy (60% to 80%).
Since accuracy increased when we collected more samples, our group can keep working on updating the learning program by scraping more features and data (expected: > 10,000 samples) for future steps. Since we only used the algorithms available in Weka, we can also try to develop machine learning algorithms that are not available in Weka to construct a better model to improve accuracy.

Contributions

Jiahua Chen: data scraping

Jennifer Liu: data processing, website, report

Glenn Xu: running models on Weka, report