DM - Project report

Group 12
Leonardo Vona
545042

November 7, 2022

1 Data understanding and preparation

1.1 Tweets data set

The tweets data set contains over 13 million tweets. Each tweet has associated 9 attributes.

1.1.1 Characterization of the features

user_id The user_id is a categorical (String) feature.

The tweets that does not have an associated user_id (200.000 entries) have been removed from the data set because they are not useful for the scope of the project, which is to create a profile of the users.

For the same reason, the tweets containing a non numeric user_id (which is not allowed in Twitter), which are other 200.000 elements, have been removed.

retweet_count It is a numerical feature, recognized as Float by Pandas, but actually Int. Pandas recognizes the attribute as Float because it contains some NaN values (which are Float). After the handling of the missing values, the attribute has been correctly converted to Int.

The maximum number of retweets obtained by a tweet is less than 4.1 millions, then all the values above this limit have been removed and considered as missing values.

reply_count It is a numerical (Int) feature. Analyzing the values, it emerges the presence of a 'inf' value, which has been canceled.

favorite_count It is a numerical (Int) feature. The most liked tweet has received less than 7.2 million likes, then the values above this threshold have been removed.

The data set also contains an instance where the value of favorite_count is negative, which has been handled as a missing value.

num_hashtags It is a numerical (Int) feature. The maximum number of hashtags per tweet has been set to a "rough" limit of 140 (the maximum tweet length is 280).

num_urls It is a numerical (Int) feature. Twitter allows to insert at most 10 links per tweet.

num_mentions It is a numerical (Int) feature. Twitter allows to mention at most 10 users per tweet.

created_at It is a numerical (Datetime) feature. The range of allowed dates in which a tweet can been created is between 2006-03-21 (date of publication of the first tweet) and 2022-09-30 (date of publication of the project).

I also searched for tweets which have been published before the creation date of the account which is associated to (retrieved from the users data set).

After cleaning the attribute, it emerged that the range of publication of the tweets in the data set is between 2012-03-11 and 2020-05-03.

1.1.2 Duplicate data

Two tweets are considered equal if they have the same value for the attributes 'user_id', 'created_at' and 'text'. Between two (or more) duplicate tweets, I pick the one with less missing values for the numerical features.

The approach is to order (descending) the tweets by the numerical features, considering NaN as the smallest value, and keep only the first tweet between duplicated ones.

After the process, more than 3 millions tweets were marked as duplicates and removed.

1.1.3 Outlier detection

It is not likely that a tweet has a number of retweets greater than 1000 and a number of likes less than 10. These outliers values for the retweet_count are considered wrong data and removed (replaced with NaN).

1.1.4 Handling missing values

The tweets data set, after applying the phases described above, contains more than 500K elements containing at least a NaN value.

I have chosen to remove the tweets containing no text because for most of them also other features are missing, and so they are not very useful for the analysis (**Note:** I'm evaluating to keep them, they can be useful for example to have a more precise publication rate).

For the numerical features containing NaN values (except created_at), it is possible to fill the missing values using as substitution value the mode (instead of the mean, which is more sensible to outliers) of the feature for each user. This is done grouping the tweets by user_id and then extracting the mode for each numerical attribute.

For the created_at feature instead, the mean is more meaningful, and the outliers have been already removed, so I used it.

After this process, there are still 1K tweets with at least a NaN value (because the users which are associated to have only that tweet in the dataset, so it is not possible to fill the missing values with the approach used above). Having a look at these entries it is likely that these are erroneous data, which are not associated to users recorded in the users data frame, so I have chosen to drop these tweets.

1.1.5 Separate tweets data set

The tweets have a different behavior with respect to retweets (which are naively identified if the 'text' attribute is starting with 'RT @'). In particular they have an high 'retweet_count', and associating them to users who have retweeted them may create problems in the user profiling.

Separating tweets and retweets will hopefully allow to recognize better the behavior of an user.

1.2 Users data set

The users data set contains the information about 11508 users. Each account has associated 5 attributes.

1.2.1 Characterization of the features

name It is a categorical (String) feature. There is one user with no name associated. Since the feature is not particularly significant for the analysis, the missing value has been filled with the id of the user.

lang It is a categorical (String) feature.

The attribute contains two instances of the default value 'Select Language...'. This value has been inferred by the language used by the user to write the tweet, which is English ('en') in both cases.

There are also two pairs of values ('en-GB' - 'en-gb' / 'zh-TW' - 'zh-tw') which actually are the same value and then have been merged.

There are 23 distinct values for the language feature, considering also regional sub-languages ('en-gb', 'en-au', 'zh-ch', 'zh-tw'), which have been kept separated from the principal one.

The most frequent language is 'en' by far, followed by 'it' and 'es'.

bot It is a categorical (String) feature. Admitted values for the feature are '0' and '1', and in the data set there are not invalid or missing values for this attribute.

created_at It is a numerical (Datetime) feature. The range of allowed dates in which a user can been created is between 2006-03-21 (date of publication of the first tweet) and 2022-09-30 (date of publication of the project). There are no instances containing a wrong value for this feature.

statuses_count It is a numerical (Int) feature.

1.2.2 Duplicate data

Two user entries are considered equal if they have the same value for 'user_id'. Using this criterion, in the data set there are no duplicate users.

1.2.3 Handling missing values

For the users data set, there is only one user which has no associated name. Since it is a categorical attribute and it does not influence the users profiling, I have chosen to assign the id as the name for this entry.

For the statuses_count instead I filled NaN values with the mode value.

1.3 Introduction of new features

After the cleaning of the data it is possible now to create new indicators which may be helpful in discriminating users' behavior.

I first merged the tweets and users data frame with a right join, so that only the tweets that have a correspondent user in the users data set are picked.

After the merging, the resulting data set is grouped by user_id, in order to easily extract the interesting new features.

num_tweets Counts the number of tweets associated to the user.

num_retweets Counts the number of retweets associated to each user. To obtain the value I merge the users and the retweets data frames similarly as before, and then retrieve the values.

avg_tweets_per_day Stores the average number of tweets per day for each user. To calculate the average first I retrieve the datetime of the first and the last tweet, and then divide num_tweets by the interval (in days) between the first and last tweet.

avg_[retweets/replies/favorites/hashtags/mentions/urls]_per_tweet For each numerical feature of the tweets data frame (except created_at) I create a new attribute containing the average of that feature per tweet.

cumulative_[retweets/replies/favorites/hashtags/mentions/urls]_per_tweet For each numerical feature of the tweets data frame (except created_at) I also create a new attribute containing the cumulative sum of that feature for each user.

avg_tweet_length Contains the average tweet length for each user.

entropy_hour Measures the entropy in terms of hour of publication of the tweets associated to each user.

To retrieve the entropy with respect to the hour of publication (and analogously with respect to the text length), I first count (for each user) for each distinct value of the hour (0 - 23) the number of occurrences, then I apply the entropy to the occurrences normalized as probabilities, diving them by the total number of tweets associated to the given user.

entropy_text_length Measures the entropy in terms of text length of the tweets associated to each user.

entropy_publication_rate Measures the entropy in terms of publication rate of the tweets associated to each user.

The entropy associated to the publication rate is calculated by, for each user, sorting the datetime of publication of his / her tweets, extracting the difference in seconds between the list of datetimes (excluding the first tweet) and then applying the entropy function.

If some users have just one tweet, the entropy for the publication rate will produce a NaN value, so I fill them with the value 0.

1.3.1 New features distribution and statistics

Most of new features show a distribution still highly skewed towards zero, as expected. They are not very informative, but we can see a more spread distribution for the features num_tweets, avg_text_length, entropy_hour and entropy_text_length.

If we try to compare the distribution of genuine vs bot users with respect to the four promising features listed above, we can see that in fact the distribution are different depending on the bot label.

For example, for the num_tweets feature we can derive that the number of bots with 0 tweets is twice the number of genuine users with 0 tweets.

If instead we consider the entropy characterizing the hour of publication, the bots seems to have a more fixed variability (around 2.8) with respect to genuine users.

Comparing the interquartile ranges of the features for the genuine and bot users we can see a general lower activity for bots. It is possible to see this for example in the number of tweets per user, or in the number of retweets or in the number of likes received, which values are all considerably lower for the bots.

Concerning instead the entropy features, it is interesting to see a slightly higher variability for the bots. If we investigate into it anyway, it is possible to see from the mean value a more realistic ratio, where the genuine users have an higher mean variability. We can also say that the mean is a more appropriate estimate for the entropy features, given the limited presence of outliers for these attributes.

2 Clustering

2.1 Preprocessing

2.1.1 Dividing the data set

I divide the user features into categorical and numerical, into the data frames cat_df and num_df, respectively.

I put the feature 'created_at' into the categorical data frame even if it is not because it can't be used for the clustering.

2.1.2 Elimination of highly correlated features

Highly correlated features may turn into a problem for the clustering analysis. There is an high correlation (greater than 0.9) between:

- avg_favorites_per_tweet and cumulative_favorites
- avg_retweets_per_tweet and cumulative_retweets
- cumulative_favorites and cumulative_retweets

For this reason I decided to drop the features cumulative_favorites and cumulative_retweets.

2.1.3 Normalization

Before starting the clustering analysis, the numerical features are normalized using a MinMaxScaler. I used the min-max method instead of the standard (Z-score) one because it fits better with the data in consideration.

2.2 K-Means

2.2.1 Determine K

Elbow

Silhouette

Comparison

2.2.2 K = 3 evaluation

Evaluation by external metrics

2.3 Density-based

2.3.1 Study of the clustering parameters

2.3.2 Evaluation

Evaluation by external metrics

2.4 Hierarchical

2.4.1 Evaluation

Evaluation by external metrics

2.5 Comparison