Machine Learning project: Twitter food popularity

Gabriele Cimador

Course of AA 2021-22 - Data Science & Scientific Computing

1 Problem statement

The aim of this project was to build a tool based on Machine Learning that can predict the popularity of a tweet about food. The goal was to allow the users to predict the popularity of their tweets before posting them, in order to get a prediction of their success.
The input variable is a single "tweet" which **is not** a "retweet" or a "quote". A detailed description of the input variable is described later.
The output is a label $y \in$ {popular, not popular}; the problem is thus intended as a binary classification problem.

2 Assessment and performance indexes

The assessment of the proposed solution will be done by splitting the dataset: 80% of it will be the training set and the remaining 20% will be the test set.
To assess the performance of the model, there will be used four indexes; the first two are the True Positive Rate and the True Negative Rate for the standard threshold of 0.5. Secondly, since there is not a specific preference in predicting one of the two classes, the Equal Error Rate will be computed. Finally, to get a general overview of how the model behaves with different thresholds, the ROC curve and the AUC value will be computed.
Please do consider that the positive class is "popular" while the negative one is "not popular".

3 Proposed solution

The proposed solution is to use a supervised learning technique to build the model that will be used for the predictions. There have been involved two phases.

3.1 Learning phase

To build the model, it is first necessary to exaclty define the "tweet", i.e. the input variable. A tweet is composed of the following features:

Feature	Type	Description
created_at	integer	Hour of the day at which the tweet has been posted, e.g. 21
source	factor	From which application the tweet has been posted
len_hashtags	integer	The total length of the used hashtags in the tweet
media_type	factor	The type of media inserted
lang	factor	The language of the tweet
followers_count	integer	# of followers of the user who posted
friends_count	integer	# of users which the user follows
listed_count	integer	# of twitter lists in which the user is present
favourites_count	integer	# of liked posts by the user
statuses_count	integer	# of user's posts
verified	boolean	if the user is verified
has_url	boolean	if there is a url in the tweet
has_mentions	boolean	if the user has mentioned somebody in the tweet
class	factor	if the tweet is popular or not

Table 1: Features of the input variable tweet

The output variable will be "class". Since popularity is subjective and might differ from account to account, the tweets of the dataset are labeled according to the following criterion:

$$\text{if: } \begin{cases} \text{tweet\_likes} = \text{likes of the tweet} \\ \text{average\_likes} = \text{average likes received by the user for his/her past 100 posts} \\ \text{sd} = \text{standard deviation of likes received by the user for his/her past 100 posts} \end{cases}$$

$$\text{then: } \begin{cases} \text{y} = \text{popular}, & \text{if tweet\_likes} \geq \text{average\_likes} + 2 * \text{sd} \\ \text{y} = \text{not popular}, & \text{if tweet\_likes} < \text{average\_likes} + 2 * \text{sd} \end{cases}$$

This criterion was chosen because it is used in the study done by Nimish Joseph et. al [1]. The other features of the tweet were the independent variables used for prediction. These features have been chosen based on the studies on popularity of tweets of Nimish Joseph et. al [1] and Kyle A. Jalbert [2].
The machine learning model chosen to build this tool is Random Forest. This choice has been made based on the results of the study of Joseph et. al [1], because of its ability to assess importance to features, and because of the OOB error that will be used for tuning the model.

3.2 Prediction phase

Once the Random Forest is trained, the tool is ready to make its predictions. Before posting, a user can insert as inputs the feature of the tweet he/she want to post and the tool will present in output a label $y \in \{\text{popular, not popular}\}$; this will be the prediction that the tool has made on the tweet. It is worth noting that all the features can be known before a user posts its tweet.

4 Experimental evaluation

4.1 Data

The Twitter API has been used to collect the data. Since this tool is specialized about food topics, the tweets were collected based on the presence of several famous hashtags about food. There have been collected a total of 9661 tweets; however, the dataset was unbalanced, with 8932 "non popular" observations and 729 "popular" observations. For coping with this problem, there has been used an oversampling method for the "popular" observations and a undersampling method for the "not popular" observations in the training set.

4.2 Procedure

4.2.1 Data collection

The data was collected with the Twitter API, on the 10/1/2022; due to limitations imposed by the API, the tweets collected were at most old of 9 days. There have been obtained 9661 tweets containing several famous hasthags about food; for every collected tweet, 100 more tweets from the same account have been retrieved in order to obtain the average amount of likes that a user receives and its standard deviation, which are necessary to label the datapoints.

4.2.2 Data cleaning and preparing

Once all the tweets had been collected, the dataset has been cleaned and prepared. There were first removed unwanted columns that were unuseful for the predictions. Secondly, several columns have been transformed in order to obtain the desidered features; for example, every tweet collected has as an attribute which is a list that contains all the mentioned users in the tweet; this attribute has been transformed to a factor that indicates if a tweet has mentions or not. After that, all the tweets have been labeled and 339 datapoints were removed due to the impossibility of being labeled. Finally, thanks to the "ROSE" package, the training set has been balanced between the two classes of observations. With this method, the training data had 4980 "not popular" tweets and 5020 "popular" tweets.

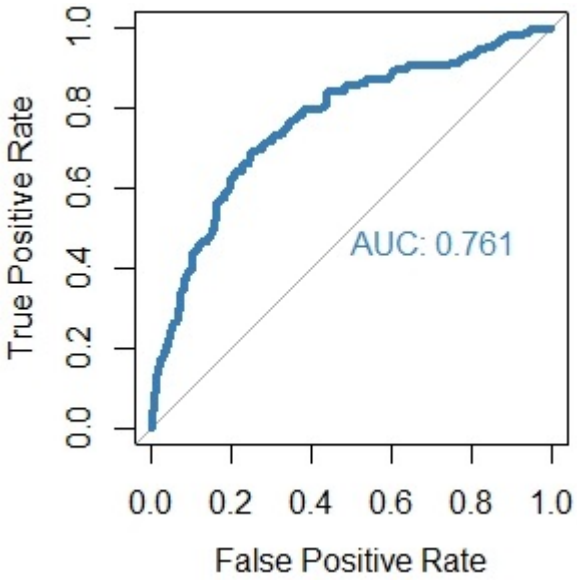

Figure 1: ROC curve and AUC value for the Random Forest model

4.2.3 Learning and assessment of the model

The balanced training set has been used to train the Random Forest model. The number of trees B has been set to 500 and the number of features selected for every tree has been set to $m = 6$ because it minimized the OOB error.
The test set (which is unbalanced) has been used to assess the model performance; in particular, there have been computed the four performance indexes described before.

4.3 Results and discussion

For the standard threshold of 0.5, the rates were: True Positive Rate = 0.340, True Negative Rate = 0.921. The Equal Error Rate computed was EER = 0.289, which indicates that, with the right threshold, less than one over three datapoints are missclassified. *Figure 1* shows the ROC curve with an AUC = 0.761.
Even though the balanced training set, the model seems not to be very accurate in predicting, especially the popular class. The main suspects are that there are missing features that might be important for the popularity, e.g. the sentiment of the tweet, and the difficulty to obtain training data in a long period of time. It has to be said though that overall the model is able the give a raw prediction about the popularity of a tweet about food, so the results are not so bad.

References

[1] Nimish Joseph, Amir Sultan, Arpan Kar, P. Vigneswara Ilavarasan. *Machine Learning Approach to Analyze and Predict the Popularity of Tweets with Images*. 17th Conference on e-Business, e-Services and e-Society (I3E), Oct 2018, Kuwait City, Kuwait. pp.567-576, 10.1007/978-3-030-02131-3_49 . hal-02274178

[2] Jalbert, Kyle A., *Understanding the Factors that Influence Tweet Popularity (2021)*. Honors Theses and Capstones. 588.