Machine Learning and Data Mining project: Prediction of Tweets Popularity

M. Danese[1], D. Roznowicz[1], and N. Tosato[1]

[1] problem statement, solution design, solution development, data gathering, writing

Course of AA 2020-2021 - Data Science and Scientific Computing

1 Problem under Analysis

A fairly well-known kitchenware manufacturing company is willing to increase its influence through twitter. In order to sponsor its own set of knives, it would like to partner with some food bloggers/influencers who have already been chosen by the firm itself. Needing to avoid excessive costs, it has not opted for famous people, but medium-profile individuals who have built a little fan base on twitter by showing their expertise, and by publishing recipes and photos of remarkable dishes.

The company has made a deal with the bloggers: it agrees to pay a fixed sum for every **popular tweet**, in which its brand is explicitly mentioned in the text or at least one of its knives appears in the attached photos. To maximise the return, the company will grant the payment **if and only if** the tweet is predicted to get more than 50 likes (also called **favorite_count**), a threshold imposed by the firm itself after proper customer researches.

2 Problem Description

The firm turned to our expertise to determine if a future tweet will become **popular** or not, giving us some blogger profiles to dig into:
"heyadamroberts", "smittenkitchen", "foodwishes", "nomnompaleo", "Food52", "balancedbites", "CookingChannel", "bflay", "GDeLaurentiis", "shakeshack", "ottolenghi", "testkitchen", "therealweissman", "thedomesticman".

It is known that the popularity of a tweet can be modelled as a function of various features [3, 1, 4], which should, in principle, relate to the appeal of the post in the reader's eyes. Therefore, our goal is to build a machine able to predict whether a tweet will be popular or not, i.e. compute a binary classification (more or less than 50 likes).

3 Data Gathering and Pre-processing

We gathered our dataset through twitter API interfaced with "rtweet", querying in each user timeline all the tweets older than a week, in order to assure that their popularity evaluation metrics reached a stable value. The result is a 30'000-elements set of tweets, each presenting 90 features, either numeric or in the form of an ordered character collection. Lastly, being the images in huge amount ($\sim 1.6GB$) and indirectly given through an url, we used "aria2c" to parallelise the download phase, and retrieve them in a few minutes.

Since direct mapping of each pixel in its 3-dimensional RGB space provides a too large amount of features (i.e. $256 \cdot 256 \cdot 256 \sim 16.8M$), the colour space was grouped into 50 distinct colour-bins proven robust to variations [2, 3]. Approximating each pixel to its nearest colour trough euclidean norm, a normalised frequency of these colours is extracted for each image, and rationalised into a vector in $\mathbb{R}([0,1])^{50}$. Because image processing is computationally intensive, a first attempt using python resulted in a forecasted processing time of 300 hours. Thus, to reach an acceptable timing of few minutes, we developed a C program and then implemented a parallel version of it.

By means of an ad-hoc created food dictionary, a topic filtering was performed on the row data, selecting only tweets presenting at least a word from a food inherent vocabulary. The remaining observations formed a significant dataset of more than 15'000 tweets. We performed stemming and removed stop words in order to compensate for the high dimensionality, and because a sentiment analysis falls outside of our purposes, additional punctuation removal and lowercasing were done, being lossless in information.

In-tweet links, hashtags and mentions were independently associated to flag variables and removed from the main text, allowing to keep their occurrences as factors, but avoiding additional noise and raise in the dimensionality. Thus, the used predictors were: **hashtags**, **mentions**, **links**, the **text length**, the **amount of followers**, the **number of friends**, the **amount of lists** to which the user is enrolled, a 50-dim **image** vector, and **word features**, which were included adopting a bag-of-words approach with variable numbers of elements.

4 Data Analysis

Two binary classifiers were built and compared: Support Vector Machine (SVM), with package "e1071" and Random Forest (RF) with "superML", allowing for multithreading computations. The training dataset was set to 80% of all available data, while the remaining 20% was used as test set to check the accuracy of the prediction.

On one hand, direct input of all the observed words does not represent a suitable choice, being the curse of dimensionality extremely effective on a dataset with such concise texts. Td-idf analysis provides a possible solution, but also this technique shows no predisposition for our dataset, whose elements report an average length of only 141 characters. Thus, we based our selection on

(a) (b)

Figure 1: Word cloud of (a) food related vocabulary, and (b) all the remaining words.

occurrences, while in order to study the impact of different foods in the tweets, we trained our model setting 2/3 of the bag-of-words to be food-related, and 1/3 associated to all the other matters (Figure 1).

Since the data look slightly unbalanced between the two classes, the adopted metrics for the assessment procedure fall into FPR and FNR, defined as following:

- FPR = the rate of unpopular tweets wrongly classified as popular;
- FNR = the rate of popular tweets wrongly classified as unpopular.

Furthermore, we decided to see how these metrics change while only varying the number of word-related features used as predictors. More in depth, the features we extracted by using the classical bag-of-word approach were saved in decreasing order according to their total number of occurrences: Then, both SVM and RF were computed, by adding 50 word-related predictors each time, starting from 50. This was performed up to no more than 500 word-related features, as the SVM was getting increasingly slow to fit our purposes.

5 Conclusions

Our reasoning in identifying the better model is a trade-off between FPR and FNR. A high FNR could probably get the bloggers into losing faith in the firm as many promising tweets would be wrongly predicted as unpopular; instead, a high FPR would make the company waste money in useless tweets.

Looking at the results (Figure 2), even though none of the two applied techniques provides strong reliability, RF definitely outperforms SVM: FPR values are not particularly different, while FNR values are much lower for RF compared to SVM. For this reason, we decide to dump the SVM and keep the RF binary classifier. It is worth noticing that, since we have been hired by the firm, our main goal is to maximise the return for the company itself, and so we tend to appreciate a lower FPR, at the cost of higher FNR.

We see that, by increasing the number of features in the model, FNR increases but not that dramatically for our purposes; at the same time, FPR decreases. Being the data unbalanced (the number of popular tweets is about

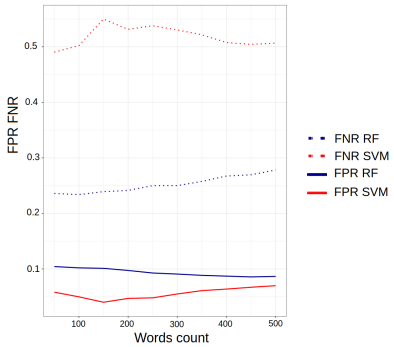

Figure 2: FPR and FNR for RF and SVM as functions of the number of words

half the amount of unpopular ones), a 1% variation in FPR might seem negligible, but it actually provides a major increase in absolute values (i.e. number of misclassified tweets), than a 1% variation in FNR:having a low FPR is way more important to let the company save a lot of money.

For these reasons, the RF classifier with about 500 features is chosen and sold to the company.

5.1 Outlooks

Possible improvements of our analysis can be achieved through more accurate pre-processing of word-related features, such as n-grams or neural network analysis of the images. Lastly, one can consider varying the sensitivity threshold in the RF, in order to reduce the misclassification of non-popular tweet.

References

[1] Liangjie Hong, Ovidiu Dan, and Brian D Davison. Predicting popular messages in twitter. In *Proceedings of the 20th international conference companion on World wide web*, pages 57–58, 2011.

[2] Rahat Khan, Joost Van de Weijer, Fahad Shahbaz Khan, Damien Muselet, Christophe Ducottet, and Cecile Barat. Discriminative color descriptors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2866–2873, 2013.

[3] Aditya Khosla, Atish Das Sarma, and Raffay Hamid. What makes an image popular? In *Proceedings of the 23rd international conference on World wide web*, pages 867–876, 2014.

[4] Nelson Joukov Costa de Oliveira. *Retweet Predictive Model in Twitter*. PhD thesis, Universidade de Coimbra, 2018.