University of Trieste
Introduction to Machine Learning Course
Academic Year 2022–2023

Predicting the popularity of food tweets

Enrico Stefanel [SM3500554]*

Last update on 2023–02–02T11:26:45Z.

Abstract

We compare different Machine Learning models that use Text Mining techniques to estimate how popular would a tweet containing the word *food* be, if published on Twitter. In particular, we study application of linear regressors, Decision trees and Random forests to find the best model in terms of MSE and prediction times.

After a little introduction to the problem statement and some preliminary analysis in section 1, we describe assessments index that are used to compare different models in section 2. Next, in section 3 we describe the proposed solution, and the experimental methods in section 4. Finally, we draw conclusions and come up with possible future improvements in section 5.

*Data Science and Scientific Computing Master's student, enrico.stefanel@studenti.units.it

1 Problem statement

We are required to build a tool for predicting *how popular* will be a tweet about food. We are not provided with a dataset, and we are requested to autonomously obtain it.

Formally, given the sets X and Y defined as

$$X = \{x \mid x \text{ is a tweet}\}$$
$$Y = \mathbb{R},$$

our goal is to learn a model $m \in M$ from the function f'_{learn} and use it to predict the response variable using a $f'_{predict}$, where the two functions are defined as

$$f'_{learn} : P^*(X \times Y) \to M$$
$$f'_{predict} : X \times M \to Y.$$

A solution based on Machine Learning is suitable for the proposed problem: building a f_{learn} function can not be done by humans, because

of human costs and solution complexity (we are dealing with tens of thousands of observations, each with hundreds of features). We expect the model M to be not very simple, so also running $f'_{predict}$ on a machine is the only way.

Given the nature of X and Y, we can (and indeed will) use supervised learning techniques for our purposes.

2 Assessment and performance indexes

Recalling that the response variable Y we are dealing with is numeric and continuous, it is natural for us to think of using Mean Squared Error (MSE) for measuring models *effectivness*. As for the *efficiency* of the solutions, we focus on the prediction times rather than the learning ones, since this last are transparent to the final users. We are going to measure the time taken by the model to predict all the response variables $y^{(i)}$ in the test set by repeating the prediction for 1000 times and taking the mean of total time. We are also keeping in consideration that a model that uses a lower number of features for the estimation of the response variable is more *interpretable* by humans than a more complex one.

3 Proposed solution

After retrieval of the dataset, a total of four different learning techniques have been tested for the prediction: (i) Dummy regressor, (ii) Linear models, (iii) Regression trees and (iv) Random Forests. For some of these families (those that are particularly promising), we deepened the study by trying to optimise learning parameters. At the end, we compared models by *effectiveness* and *efficiency*.

4 Experimental evaluation

The whole project is implemented using the R programming language, and all the code is publicly available at https://github.com/enstit/TweetPopularityPrediction.

4.1 Data retrieval

We are not going into details about how we collected the data, because it is not this project purpose. To summarise, we used Twitter APIs [5] to collect a set of 50000 tweets containing the word "*food*", written in english and published during the 2022 Christmas day (in CET time). Each observation includes the tweet text and date-time, tweet metrics (likes, retweets, replies), user metrics (following, followers) as well as media informations for those tweets that also include an image, video, or animated gif, plus other features that has not been used in the study because they clearly are not influential (e.g., the tweet author full name and username). This leads to a total of twenty-two columns for each observation.

As regarding the response variable, we know that the popularity of a tweet is surely correlated with the number of likes, retweets and replies to that. But, on the other hand, we also *expect* this insights to be high for a tweet published by a user with many followers. To take this into consideration, we divide the sum by the number of user followers. We then increase both numerator and denominator by one, to ensure that a finite index exists, and take the natural logarithm.

$$y^{(i)} = \ln\left(\frac{1 + x^{(i)}_{likes} + x^{(i)}_{retweets} + x^{(i)}_{replies}}{1 + x^{(i)}_{followers}}\right)$$

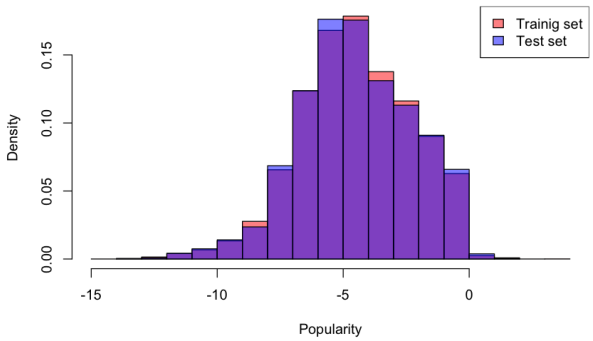

Figure 1: Distribution of *popularity index* in training and test sets.

4.2 Procedure

After reducing the dataset to the tweet text and the attached media type feature, we performed text pre-processing on the tweets corpus (lowercase conversion, punctuation and numbers removal, stemming, stop-words removal). To vectorise documents tests, we used a TF-IDF function [4] and only kept terms that appeared in at least 1% of the documents.

The final dataset counts of 202 features and a response variable.

We then divided the dataset into static training and test dataset, with a 80 : 20 proportion. We decided to do not use *cross validation* since the dataset is big enough and evenly distributed (as shown in figure Figure 1), thus the effective improvements would not be significant to justify degradation in learing-phase efficiency.

All models have been trained on the training set, and evaluated on the test set.

4.2.1 Dummy regressor

The Dummy regressor, used as a baseline to assess all the other most evolved models, is the one that always predict the mean value of $\{y^{(i)}\}_i$.

$$f_{dummy,\{y^{(i)}\}_i}(x) = \frac{1}{|y|}\sum_{i=1}^{|y|} y^{(i)}$$

4.2.2 Linear model

After learning a complete linear model using all the 202 features, we also tried, using a *stepwise algorithm* procedure [3], to reduce the number of that (so that the model *interpretability* increases) only keeping those who are justified by a dropping in AIC index if retained. This model ended by having about 60% of the original predictors (118 versus 205), but scoring a very similar MSE on the test set (even slightly lower) and being about 40% faster on predictions.

4.2.3 Regression tree

The model we tested next has been the Regression tree. Compared with the default values of *R* function `rpart` [1], only the complexity parameter `cp` has been lowered from 0.01 to 0.0001. This led the model to use 189 predictors (compared with only 13 if we left `cp` to its default value).

4.2.4 Random forest

Finally, a Random forest model has been trained. At the beginning, we trained the model using the default values of the `randomForest` function in *R* [2]. Then, we decided to slim down the model reducing the number of trees from 500 to 250 and lastly to 100. Having obtained satisfactory results with the latter option, we also

Model	MSE	$\mu_{p.time}$
Dummy regressor	5.216	0.400
Linear model		
↳ $n_{var} = 118 \ll p$	4.137	**11.048**
↳ $\underline{n_{var} = 205 = p}$	4.138	18.267
Regression tree	3.947	25.502
Random forest		
↳ $n_{tree} = 100$		
↳ $n_{min} = 100$	3.746	396.545
↳ $n_{min} = 50$	**3.741**	465.842
↳ $\underline{n_{min} = 5}$	3.765	718.422
↳ $n_{tree} = 250$	3.748	1705.838
↳ $\underline{n_{tree} = 500}$	3.747	3295.462

Table 1: Models MSE and prediction times (in milliseconds) measured on the test set. Underlined are default values for the learning technique.

proceeded to increase the minimum number of observations in leaf nodes from 5 (default value) to 50 and then to 100 to make the model even simpler (and thus faster in predictions).

4.3 Results and discussion

The results obtained for the individual trained models can be seen in Table 1.

The Dummy regressor is the one with the highest MSE on test set, as we ere expecting. It is also the fastest model in prediction phase, since the related function $f_{predict} = \Theta(1)$ simply returns a constant.

The linear model improves by about 20% in prediction accuracy. The simplest version, with only 118 predictors, takes 11.048 milliseconds on average to predict 10000 observations. It is the fastest prediction considering the complex trained models.

As for the regression tree, it performs better than the linear model. This can come from the presence of non-linear relation between some x_i and the response variable, since we only trained the linear model *assuming* linear relations. Comparing prediction times, the regression tree is more than the double slower than the simple linear model.

The "wisdom of trees" is respected if we analyse results of Random forest regressors: the MSE of the best trained model (with $n_{tree} = 100$ and $n_{min} = 50$) is 3.741, almost 30% lower than the MSE of the Dummy regressor. Complicating the model makes no sense, as we can observe from the result of training models with $n_{tree} = 250$ and $n_{tree} = 500$: the MSE does not improve, and prediction times get exaggeratedly high.

A study on variable importance conducted on the best Random forest model reveals that the factor `media.type` (i.e. the type of media attached to the tweet, that can be *photo*, *video*, *animated gif* or *none*) is the most influential variable for predicting the tweet popularity, with a drop of 147% in mean accuracy if the variable is shuffled through the observations. The second most influential variable is `christma`, with "only" 38% of accuracy decrease on average. `eat` is the first food-related variable in order of importance (fourth in general), with a related drop of 28%.

5 Conclusions

In conclusion, the best model for the prediction of a tweet popularity is a *quite simple* Random forest regressor. But, if time is crucial (let's think about working with hundred of thousand or also millions of observations, or real-time cases), a simple regression tree is eighteen times faster and only 5% more inaccurate.

A little side note: we do not expect the model

to generalise well on every-day data: the fact we collected observations on a special day like Christmas day makes the model very susceptible to festivity-related features (and at Christmas we all try to be better, maybe even leaving likes on not-so-exciting tweets).

Further improvements can be achieved by considering features in the dataset that we do not used (e.g. the tweet date-time). Also, the way to calculate the popularity we proposed is only one of the possible solutions, but it is not necessarily the most representative one.

References

[1] Leo Breiman, Jerome Friedman, Charles J. Stone, R.A. Olshen. "Classification and Regression Trees" (1984). In: *Wadsworth*

[2] Leo Breiman. "Manual On Setting Up, Using, And Understanding Random Forests V3.1" (2002). URL: https://www.stat.berkeley.edu/~breiman/Using_random_forests_V3.1.pdf

[3] Laurie Davies. "Functional Choice and Non-significance Regions in Regression". In: *arXiv* abs/1605.01936 (2016). arXiv: 1605.01936. URL: https://arxiv.org/abs/1605.01936

[4] Felipe Almeida, Geraldo Xexéo. "Word Embeddings: A Survey". In: *CoRR* abs/1901.09069 (2019). arXiv: 1901.09069. URL: https://arxiv.org/abs/1901.09069

[5] Shahab Saquib Sohail, Mohammad Muzammil Khan, Mohd Arsalan, Aslam Khan, Jamshed Siddiqui, Syed Hamid Hasan, M. Afshar Alam. "Crawling Twitter data through API: A technical/legal perspective". In: *CoRR* abs/2105.10724 (2021). arXiv: 2105.10724. URL: https://arxiv.org/abs/2105.10724