REGRESSION REPORT

Bonavita Luca, 964220
Borgonovo Giorgio, 971456
Cirignoni Niccolò, 968373

1. Introduction

In this report we will discuss the development steps of a model for predicting how popular (in terms of number of likes) a review will be in the Yoyo.com website. In fact, data analysts have identified product reviews as important driver of customer satisfaction, which is even more true in case of very popular reviews. This report will stick to the steps the team followed to get to the development of the final algorithms, which will be evaluated according to the MAE (mean absolute error), as requested by the assignment. We followed an iterative approach, trying to use different versions of the dataset (given by different decisions on how to treat outliers, transform data, and apply unsupervised techniques) to see which results they generated, and then decided for its best version according to the scores given by each format.

2. Data Exploration & Pre-Processing

The dataset's observations are single product reviews, whose target variable is their number of (positive) likes, and whose descriptors are textual features related to them, such as the rate of positive and negative words, the title's subjectivity, and so on. The given data contains a total of 28000 observations and 38 explanatory variables, 2 of which are categorical. It is also worth noticing that the data does not contain any missing values.

2.1. Categorical variables

The two categorical variables have been used to create dummies, as they are nominal variables, and then dropped.

2.2. Numerical variables

First, we noticed that two groups of numerical variables (the topic-related ones – 6 overall –, and the rate of positive and negative words – 2 overall) had a total sum of 1, therefore one variable from each group was eliminated. After this, we plotted the variables against the target and noticed a strange behaviour: most variables that were supposed to be rates (so, we expected them to be between 0 and 1 or between -1 and 1), were out of that range, and instead were all between 0 and 1000. Therefore, we assumed that these variables were wrongly registered in the dataset, and divided the ones greater than 1, or less than -1 in case of negative values, by 1000, thus adjusting them in the right scale. In addition to that, from these plots, we preliminary noticed a scarce correlation between the descriptors and the number of likes, which was then confirmed (after transformation and standardizing, described below) with the correlation matrix, and confirmed what emerged from the graph.

This can be given by the fact that descriptors might not be useful to predict likes, but also by the very high range (from 5 to 843300) of likes, and by its elevated coefficient of variation (3.69). Moreover, it is also worth noticing that liking a review is a very subjective (and impulsive) process, hence, for example, it might not be easily predictable by looking at objective parameters related to the text characteristics.

2.3. Transformation and standardization

All numerical variables were tested against many distributions (poisson, gamma, exponential, normal, uniform, logarithmic), but often, especially due to their tales, qqplots showed that assumptions on their distributions did not hold (confirmed by the Kolmogorov–Smirnov test). Therefore, we applied boxcox transformations to almost all variables apart from the "age_days", which, if transformed, behaved in a strange way, as well as the four remaining "topics"-related variables. Then, we also transformed the target variable with boxcox, and finally scaled all the numerical variables in the dataset, to offset eventual problems related to the different scales of variables.
After transforming and standardizing, we checked the multicollinearity among variables, and noticed that some groups of variables (e.g., the "self-reference-related" ones) presented a high VIF.

2.4. PCA

During this phase our aim was to explore a way to tackle the multicollinearity problem between the independent variables, and to evaluate if reducing the data variability could bring to improved results. We have tried different approaches, first with all the independent variables, and then considering subgroups, identified due to their similarity and their high mutual correlation. More precisely we selected six different groups: "polarity", "topics", "tokens", "self-references", "title", and "global" variables.
Once created, we made some experiments by substituting the PCs in the initial dataset and fitted some prediction models with them. However, the results in terms of MAE turned to be roughly equal in both situations and so, for sake of interpretation, we decided to keep the initial dataset's variable as they were.

3. Models

3.1. Linear Regression with Backward Selection

To have a preliminary idea of which set of variables were significant to make predictions on the target one, we implemented a linear regression with a backward selection of variables. The parameter to select the best model is based on the Bayesian Information Criterion, which compared to the other criteria was the only one to return a model whose selected variables' coefficients were significant (p-value < 0.05). By fitting with all variables, we learned that the least significant ones are:

- The token-related variables (apart from "n_non_stop_unique_tokens")
- The "sport" dummy
- Num_imgs
- Some polarity-related variables (global_sentiment_polarity, avg_positive_polarity, max_positive_polarity, min_positive_polarity, min_negative_polarity, abs_title_sentiment_polarity).
- Topic_description

All the results of the linear regression with the backward selection can be found in Appendix B.

3.2. Models' development

At this point, we were ready to start the predictive models' construction. We divided this part into 3 main phases:

1) **Functions definition**: as the process of tuning the models' parameters was very repetitive, we defined a function, namely "gs_regression" with the goal of making the code more readable and the process faster. This function preforms a gridsearch (given some models and its parameters) and prints its results (MAE, MSE, R^2, RMSE).
2) **Models' parameters tuning:** with that function, we used them to examine how different algorithms performed by trying out different parameters values. For each algorithm, we followed the same methodology: first, we set very wide ranges of parameters to feed the function grossly understanding which assortments of values brought to adequate results, monitoring both MAE and R^2; then, we changed the parameters one by one within narrower ranges to meticulously tune them.
3) **Stacking**: the goal of phase 2 was to perfect the algorithms' results to compare how they perform with respect to each other, to lay the foundations for building a stacking model, able to consider the prediction of more than one algorithm at the same time.

3.3. Model selection

After the first phase, we understood that algorithms returned variable results according to the number of outliers in the train and in the test set. If for example the observations with the largest number of likes were in the test set, the test's mean absolute error tended to be very significant, and vice versa if they were in the training set. Therefore, to select the best model, we understood that looking at the MAE obtained for only one train-test combination would be a weak decision. So, we created a loop that, for each iteration, resampled the train and test set, executed the algorithm, and kept track of the MAE of the training and of the test set. Then, we computed their mean and standard deviation, to understand how robust the MAE mean was. The results of this can be found in Appendix A. The best model, considering the average MAE and its standard deviation, is the stacking algorithm (that stacks results from random forest regressor, SVR, and K-neighbours regressor). In fact, with respect to the second-best performing algorithm (SVR), it presents a lower mean MAE and a slightly lower MAE standard deviation, resulting in higher performances and robustness.

3.4. Residuals' evaluation

After choosing the best model, we evaluated its residuals by fitting the whole model on the training set, and then plotting its errors. From these plots we noticed that, despite a few observations, residuals seem to behave normally with mean = -0.009 and standard deviation = 0.74. The Shapiro test's p-value is approximatively 0, but by looking at the residuals' plot, we can see that apart from a small tail where the model makes large mistakes, most residuals seem to assume a normal shape.
If residuals would not have been this way, we would have gone back to the model selection step, first tried to change the model parameters and check them again, or else taken the second-best model and evaluate its residuals.

4. Further attempts to improve MAE

To improve the effectiveness of our models, we made some attempts working on the data, described hereinafter

4.1. Clustering

With the k-means clustering algorithm we aimed at fitting different models for different subsets, to reach an overall better MAE. We tried to create different clusters both considering all the independent variables together, and clustering according to some variables subgroups (such as title-related ones, topic-related ones, and so on); the best model was clustering by using the variables whose correlation with the target was > 0.1, and the average MAE with the chosen model (stacking) resulted to be around 2350 likes. For choosing the number of clusters to pick, we looked at the best average silhouette.

As the best MAE reached with clustering was better than our simplest models and competitive for the most complex ones, it was worth considering to cluster observations according to the abovementioned group of variables. However, the standard deviation of MAEs (computed with the same procedure highlighted in paragraph 3.3), was too high with respect to the results gotten with the best models, and thus we decided not to cluster observations.

4.2. New Variables creation

Because variables are poorly correlated with the target, we wanted to understand if, by generating new variables from the ones in the dataset, results could improve. For example, we created 2 new ratios, that represented the percentage of images with respect to the total amount of images and videos and the percentage of videos with respect to the total amount of images and videos. However, by testing the behaviour of this variable when used as a descriptor in the best model, the best performance was not improved.

4.3. Outliers' elimination

As the "likes" variable presents few observations with extremely high values, another attempt to increase the model's performances was to eliminate outliers and fit the model the remaining data; the goal was to reduce the MAE by fitting a model able to grasp the behaviour of the majority of the population, making lower errors on average values but much higher errors on few big values. However, we tested the obtained algorithm on a set which also contained some unusually big observations (like the initial population) and saw that the model performed worse compared to the base case. In the same direction, we tried to eliminate the "likes" observations

5. Appendixes

Appendix A:

	$Mean_{Train}$	$Std\ dev_{Train}$	$Mean_{Test}$	$Std\ dev_{Train}$
Backward	*2395*	*50*	*2402*	*115*
Linear	*2390*	*50*	*2412*	*114*
Ridge	*2378*	*48*	*2441*	*111*
Lasso	*2406*	*44*	*2423*	*102*
Knr	*2329*	*52*	*2394*	*107*
Tree	*2410*	*44*	*2415*	*101*
Random forest	*2398*	*52*	*2378*	*123*
Extra tree	*2404*	*55*	*2392*	*127*
SVR	*2324*	*39*	*2324*	*91*
MLP	*2353*	*61*	*2400*	*144*
Ada	*2355*	*50*	*2419*	*116*
Grad boosting	*2370*	*53*	*2391*	*123*
$Stacking_1$	*2159*	*56*	*2330*	*92*
$Stacking_2$	*2092*	*30*	*2372*	*55*
$Stacking_3$	*2162*	*8*	*2279*	*61*

*$Stacking_1$: composed by knr, forest, svr, mlp, ada, grad.
$Stacking_2$: composed by knr, forest, svr, ada, grad.
$Stacking_3$: composed by knr, forest, svr.

Appendix B:

	coef	std err	t	P>|t|
intercept	0.2705	0.020	13.828	0.000
age_days	0.0228	0.006	3.806	0.000
n_tokens_review	0.3004	0.034	8.903	0.000
n_non_stop_words	-0.3620	0.035	-10.370	0.000
n_non_stop_unique_tokens	-0.0758	0.009	-8.850	0.000
num_hrefs	0.0836	0.007	11.834	0.000
num_self_hrefs	-0.0725	0.010	-7.583	0.000
num_videos	0.0682	0.006	11.415	0.000
num_keywords	0.0222	0.006	3.735	0.000
self_reference_min_shares	-0.2339	0.039	-6.025	0.000
self_reference_max_shares	-0.7617	0.087	-8.763	0.000
self_reference_avg_sharess	1.1239	0.116	9.658	0.000
topic_quality	0.0791	0.010	8.236	0.000
topic_shipping	-0.0182	0.007	-2.467	0.014
topic_packaging	-0.1028	0.011	-9.609	0.000
global_subjectivity	0.0416	0.007	6.193	0.000
global_rate_positive_words	-0.0464	0.013	-3.524	0.000
global_rate_negative_words	0.0769	0.018	4.181	0.000
rate_positive_words	0.0911	0.022	4.194	0.000
min_positive_polarity	-0.0407	0.007	-6.110	0.000
avg_negative_polarity	-0.0259	0.008	-3.079	0.002
max_negative_polarity	0.0246	0.008	3.015	0.003
title_subjectivity	0.0335	0.007	4.685	0.000
title_sentiment_polarity	0.0283	0.006	4.756	0.000
abs_title_subjectivity	0.0432	0.007	6.054	0.000
business	-0.4199	0.026	-16.253	0.000
cleaning	-0.2416	0.028	-8.595	0.000
entertainment	-0.4304	0.022	-19.876	0.000
tech	-0.0797	0.020	-3.993	0.000
travel	-0.2944	0.028	-10.490	0.000
monday	-0.0564	0.020	-2.805	0.005
saturday	0.2806	0.027	10.362	0.000
sunday	0.2624	0.026	10.021	0.000
thursday	-0.1044	0.020	-5.281	0.000
tuesday	-0.1031	0.020	-5.228	0.000
wednesday	-0.1176	0.020	-5.991	0.000