Online News Popularity Prediction: From Noisy Data To Regression Model

Luca Scalenghe
Politecnico di Torino
Student id: s303452
s303452@studenti.polito.it

***Abstract*—This report proposes an approach to building a predictive system for the popularity of online news. The proposed solution to the problem contains an analysis of the given dataset and provides insights on the dynamics that drive diffusion of information having a big impact in the media industry. The explanation of the regression pipeline is then presented and a comparison between several known models that led to achieve good performances following the RMSE metric.**

I. Problem overview

We are currently living in the Digital Age where increasingly more time is spent online and the consumption of information has shifted from paper-based to the use of electronic devices. In the digital world there is currently a paradigm shift were news are consumed through social media and therefore their visibility is not uniform but based on their public appealingness. This mechanism shifts the production of the content. The understanding of the factors that contribute to the sharing of news and articles is critical for the players in the media space. The prediction of sharing by analysing the components and metadata relative to an article can be of extreme benefit for industry insiders.

A. Data Exploration

The provided dataset is composed of two parts: *Development dataset* and *Evaluation dataset*. The Development dataset contains 31715 records, described by 50 features and the Evaluation dataset contains 7917 records. An exhaustive list of all features can be seen in Table I. The first exploration of the dataset showed that there were no duplicate records and that all columns had no missing values except for num_imgs (20.1%), num_videos (19.9%) and num_keywords (19.9% missing). URL column has been used as input to a script for scraping the webpages and fill the missing values for images and videos. This operation took 260 minutes in total. It was taken into account that some pages where not available anymore. This case was handled by filling with zero values. In order to handle the most important features for this task a Random Forest Regressor was used to asses the feature importance, see results in Table I. It is important to note that only the most important features are presented in this report, for additional informations visit the repository.

B. Feature Analysis

The first analysis was performed on the target feature, which presented a very skewed distribution as shown in Fig. 1.

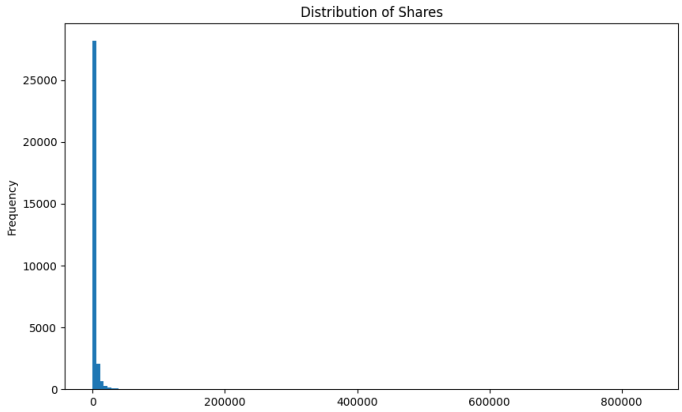

Fig. 1. Distribution of target feature

There is the presence of outliers as shown in Fig. 2 that are distributed sparsely on a wide range of values. A zoom on the actual boxplot is shown in Fig. 3 that shows how 95% of the data is distributed. An attempt of parsing the URL link was made searching for additional information as year, month and numerical day. The new features did not seem to be very relevant and where discarded afterwards. The univariate analysis of all important features was then performed in order to obtain more information on their distribution and their peculiarities.

After the first overview the idea was to select from groups of similar features the most important ones. A multivariate analysis was performed and correlations have been taken under scrutiny in order to eliminate through feature selection most of the correlations present within each group. An example of the procedure can be seen in Fig. 4 where similar features in meaning like n_tokens_content, n_non_stop_unique_tokens, n_unique_tokens, n_tokens_title have been evaluated both by their importance and by their correlation. From this group n_tokens_content has been selected for the next stage of the pipeline because most uncorrelated and most important. Similar considerations have been done for all other groups, for more details on each selection visit the repository. The selected features that were used for the model are:

- timedelta
- n_tokens_content

- num_hrefs
- num_imgs
- num_videos
- average_token_length
- kw_max_min
- kw_avg_max
- kw_avg_avg
- self_reference_avg_shares
- LDA_00
- LDA_01
- LDA_02
- LDA_03
- LDA_04
- global_subjectivity
- global_rate_positive_words
- global_rate_negative_words
- avg_positive_polarity
- avg_negative_polarity
- title_subjectivity
- title_sentiment_polarity
- data_channel
- weekday
- shares

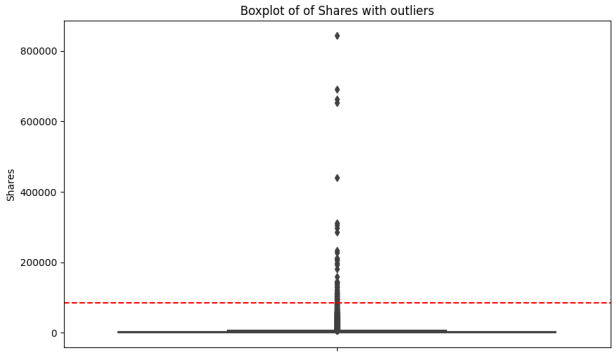

Fig. 2. Boxplot of Shares with outliers, red line shows the capping of the data at 85000, approximately 0.2% of dataset is present above this line

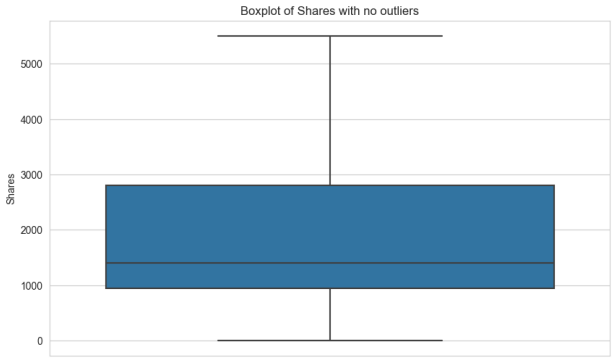

Fig. 3. Boxplot of Shares without outliers

II. Proposed approach

In this section the description of the pre-processing of the data will be presented and how the selection of the regressor was performed. The transformations described are applied both on development and evaluation datasets.

TABLE I
Feature Importance and Correlation sorted by Feature Importance

Feature	Importance	Correlation
n_tokens_content	0.08098739994018335	0.006407
kw_avg_avg	0.07425034926820087	0.103006
kw_max_avg	0.07251012013164254	0.056123
average_token_length	0.044988097928351185	-0.018259
self_reference_avg_shares	0.04334711233939437	0.062653
LDA_00	0.04254155069915452	-0.003499
self_reference_min_shares	0.03512707829639118	0.061501
LDA_04	0.030257527041578476	-0.014132
kw_avg_min	0.029527120800856364	0.015748
LDA_03	0.02951955248414905	0.080471
n_non_stop_unique_tokens	0.026520576140195307	-0.018692
kw_avg_max	0.02649839064377973	0.045563
avg_negative_polarity	0.025028278168222073	-0.034730
num_hrefs	0.02413606705063245	0.046224
timedelta	0.024070365248518327	0.008731
LDA_01	0.02339902320117783	-0.009844
kw_max_min	0.022225314297665704	0.004461
n_unique_tokens	0.021895296560365242	-0.005761
self_reference_max_shares	0.021852174582169617	0.051441
LDA_02	0.020190784983559293	-0.058666
title_subjectivity	0.019013135341712475	0.023279
kw_min_avg	0.018452493899312535	0.038692
kw_min_max	0.016754229994028775	0.004461
global_sentiment_polarity	0.016475146721152228	0.003121
global_subjectivity	0.016134780549531554	0.035555
min_negative_polarity	0.015325121925019263	-0.022413
avg_positive_polarity	0.014587970591865188	0.016275
global_rate_positive_words	0.014449586848428361	0.000719
title_sentiment_polarity	0.013699389418188131	0.011451
num_self_hrefs	0.012202304636452787	-0.001116
global_rate_negative_words	0.011962197626153298	0.010116
max_negative_polarity	0.011555915303293068	-0.019667
n_tokens_title	0.011057538996979732	0.007482
weekday_monday	0.010707109666062583	0.010064
data_channel_bus	0.009018307881813714	-0.009304
num_imgs	0.00820892281892419	0.035279
rate_positive_words	0.00812152823934287	-0.013572
num_videos	0.00653398695716251	0.018142
rate_negative_words	0.005870326764958904	-0.000889
num_keywords	0.0046028356844532384	0.023405
abs_title_sentiment_polarity	0.00459940282065607	0.027853
min_positive_polarity	0.00389359678983924	0.001528
abs_title_subjectivity	0.0037827774448156324	-0.003274
max_positive_polarity	0.0034674046171566026	0.011894
kw_max_max	0.0034663501100246894	0.009432
weekday_wednesday	0.003370567200517444	-0.003998
data_channel_lifestyle	0.0026039195439370036	0.084398
weekday_friday	0.0017426753007613513	-0.003364
weekday_tuesday	0.0016039326905701507	-0.005280
weekday_thursday	0.0014773751528176131	-0.009047
data_channel_tech	0.0014184310734543743	-0.012680
weekday_saturday	0.001209312186744578	0.011327
data_channel_socmed	0.0009446695413634634	0.002389
weekday_sunday	0.0007824051744841903	0.007096
data_channel_entertainment	0.0007139614029015877	-0.018575
data_channel_world	0.0006795454519043652	-0.047170
kw_min_min	0.0005999517192545755	-0.002783
n_non_stop_words	0.00003871210777403102 4	-0.016120

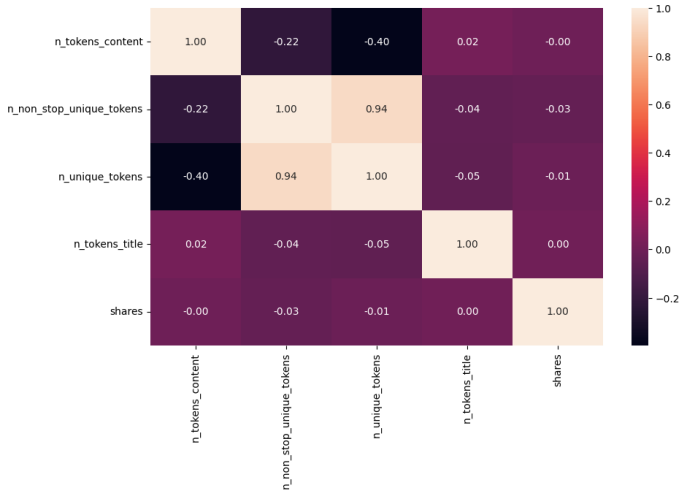

Fig. 4. Heatmap of correlations between tokens features

A. Preprocessing

The development dataset was capped where the shares were bigger then 85000 in order to eliminate 0.2% of the data by cutting out noisy instances, see Fig. 2. The 947 instances where columns n_tokens_content, avg_positive_polarity, self_reference_avg_shares, avg_negative_polarity, average_token_length, global_subjectivity were all zeros concurrently have been imputed since in the evaluation set the same phenomena was present in 231 instances. The last three columns were imputed by using the Scikit-learn SimpleImputer using the median because of their skewed distribution. Columns n_tokens_content, avg_positive_polarity and self_reference_avg_shares were also imputed using SimpleImputer but with the mean strategy. The categorical columns i.e. weekday and data_channel have been encoded using DummyEncoding. Numerical columns have been rescaled using StandarScaler and the LDA columns have then been aggregated into a single one called PC1 through the PCA method. This choice was made because each LDA represented a coordinate in a 5-dimensional space so choosing a single dimension were maximal variance could be preserved seemed a natural dimensionality reduction in order to reduce the dimension of the design matrix.

B. Model selection

The model selection was performed by using two groups of models specifically:

- **Traditional models**
  - Linear Regression
  - Lasso
  - Ridge
  - Random Forrest Regressor
  - K-Neighbors Regressor
- **Ensembles and more complex models**
  - Bayesian Ridge
  - ARD Regressor
  - Extra Trees Regressor
  - AdaBoost Regressor
  - Gradient Boosting Regressor

All the trainings used Cross Validation using three partitions. The training of the traditional models gave the opportunity through the Random Forest Regressor to measure the feature importance. It was noted that the categorical columns seemed to be potential candidates for removal. To check this hypothesis a feature selection training has been performed obtaining the results in Table II. This step showed that maintaining this features gave a better predictor with respect to discarding them. The best predictor of this group was Lasso. The second group of predictors was then trained again on all the features giving the results in Table III, where Bayesian Ridge and ARD Regressor proved to be the best models. Those two and the Lasso were trained on the complete dataset in order to make a submission on the platform and the worst performance was given by Bayesian Ridge which was dropped from the group for the next stage of the process.

TABLE II
TRADITIONAL REGRESSION MODELS

Model	All features	No weekday	No channel	No weekday and channel
Linear Regression	5506.90	5507.98	5516.70	5517.98
Lasso	5506.88	5507.97	5516.65	5517.92
Ridge	5506.95	5507.98	5516.75	5517.98
Random Forest Regressor	5653.13	5674.06	5665.03	5671.83
K-Neighbors Regressor	5912.67	5918.38	5904.97	5929.08

TABLE III
COMPLEX REGRESSION MODELS

Model	All features
Bayesian Ridge	5506.31
ARD Regression	5507.16
Extra Trees Regressor	5623.15
AdaBoost Regressor	9079.07
Gradient Boosting Regressor	5517.34

C. Hyperparameters tuning

A Grid Search was performed using the values in Table IV. The best parameters are highlighted in bold.

TABLE IV
MODEL PARAMETERS TESTED

Model	Parameter	Value
ARDRegressor	n_iter	100, 200, **500**
	alpha_1	1e-5, 1e-4, **1e-3**
	alpha_2	1e-5, 1e-4, **1e-3**
	lambda_1	1e-5, **1e-4**, 1e-3
	lambda_2	1e-5, 1e-4, **1e-3**
Lasso	alpha	0.0001, 0.001, 0.01, 0.1, **1.0**, 10.0, 100

III. RESULTS

Lasso and ARD Regressor were trained using the best parameter configuration and used for a submission on the

evaluation platform. The best performer was the second that scored 5959.446 while Lasso scored 5962.454. It has to be noted that Lasso is a far easier model. The score obtained for the ARD Regressor was selected as final evaluation for the submission.

IV. Discussion

The cutting out of outliers in the dataset in this case is definitely an over-simplification of reality because some news could become instantly much more viral then others and show some remarkable high value in shares. The study of this problem brought to the usage of two approaches that were not successful but are worth mentioning.

A. Log-normalization of the dataset

This was hinted by the distribution of the target. The skewed distribution suggested the log-normalization and after applying the transformation the shape was more similar to a Gaussian as shown in Fig. 5

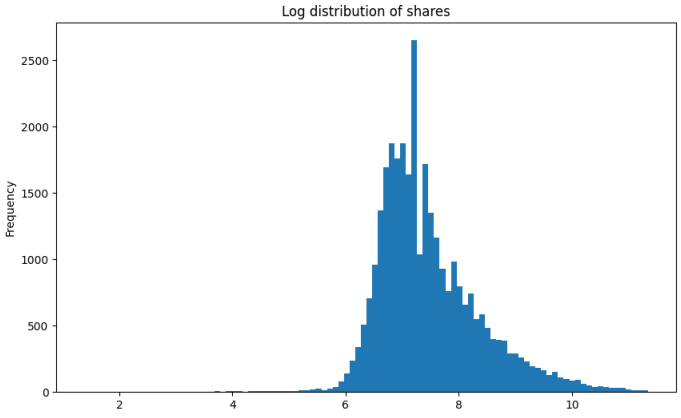

Fig. 5. Log distribution of target variable

Some models were trained by using log normalization on the skewed features present in dataset but all attempts produced models that gave RMSE bigger then 6000 which was below baseline thus this approach was not further investigated.

B. Classification of outliers and regression on each class

The different behaviour of outliers led to the idea of developing a new model composed by a classifier and a regression task for each of the label predicted, as shown in Fig. 6. The development of the classification model had to tackle the unbalanced dataset produced by the labeling of outliers. Outliers are by definition less in quantity than normal instances. Techniques of under sampling and over sampling have been tested. The regression on only normal labels performed much better producing RMSE scores of around 3000 but the same wasn't true for the outlier regressor that gave RMSE scores of around 20000. An explanation to this is the highly unpredictibility of some news that are viral and the lack of features that could depict the phenomena of virality. Although theoretically it seemed to be a good idea, practically this approach did not gave good results obtaining RMSE scores bigger then 30000.

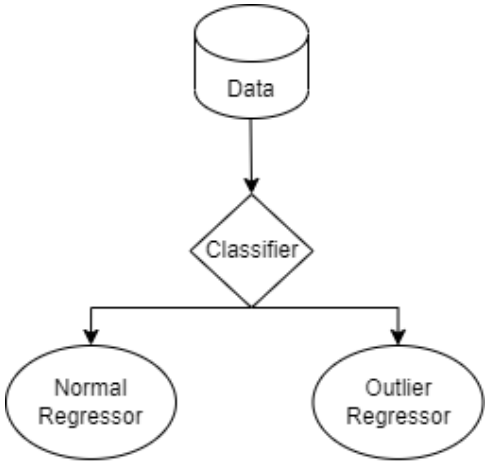

Fig. 6. Log distribution of target variable

References

[1] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.