If it doesn't spread - it's dead!

Online News Popularity Regression Task.

Silva Bashllari
Politecnico di Torino
Student number: [redacted]
[redacted]@studenti.polito.it

Alejandro Mesa Gomez
Politecnico di Torino
Student number: [redacted]
[redacted]@studenti.polito.it

***Abstract*—The aim of this report is to provide an approach to building a regression pipeline to predict the shareability of news in a given dataset. In particular, we perform an extensive analysis of all the possible steps to follow in the design of the regression model. Our approach can be best described as a mainly brute force one due to the fact that we explore a wide variety of models and preprocessing techniques. Our best pipeline outperforms the given baseline in the leaderboard of 5986.521, reaching a top 15 score of 5965.901 only 57.6 points away from the top score.**

I. Problem overview

In recent years, great attention has been placed on understanding how news spread in the digital world. In fact, scholars of digital media consider *spreadability* (shareability) as a central dimension of today's media culture, thus the statement: *"If it doesn't spread, it's dead!"*. Spreadability, according to Jekins et al. [2], refers to the potential — both technical and cultural — for audiences to share content for their own purposes. In our particular case study, we will examine the shareability of news articles and the dataset we have contains news only from a particular site, namely "Mashable.com". Thus, we cannot state that the following analysis could be generalized for all the news in the virtual space, but it attempts to scratch the surface of understanding the drivers of shareability. We have been provided with two data sets, namely:

1) the development set - composed of 31 715 records;
2) the evaluation set - composed of 7917 records, almost 25 percent of the development set.

From the variables perspective, the data sets are composed of 50 attributes, a relatively large set. However, the usual rule of thumb is that regression with only one dependent and one independent variable normally requires a minimum of 30 observations and in multiple regression, usually are added at least an additional 10 observations for each additional independent variable added to the equation[3]. We have roughly 634 observations per variable, which exceeds the given guideline. In a very basic exploratory analysis we discovered that there are no duplicates and the missing values are only in the following three variables: number of videos, number of images, and number of keywords in the metadata of the articles. This is expected as not all articles need to necessarily have images, videos, and metadata in them. The ultimate goal of the project is to minimize the RMSE (Root-Mean-Square Error).

II. Proposed approach

A. Preprocessing and Model Selection.

For the purposes of this regression task, we decided to work with a variety of models, aiming to find the one that would most fit the data and the task at hand. The models we decided to employ are:

- ***Multiple Linear Regression.***
- ***Lasso Regression:*** which was initially trained with an arbitrary choice of the alpha parameter 10.
- ***Ridge Regression:*** which was also initially trained with an arbitrary choice of the alpha parameter 10.
- ***Support Vector Machine Regression:*** using the default parameters in Sci-Kit Learn, namely: kernel was the "rbf", degree was 3 and gamma was "scale".
- ***Random Forest Regression:*** using the default values in Sci-Kit Learn, namely: 100 estimators, the criterion was "the squared error" and the maximum depth was none.
- ***Neural Network:*** using the Pytorch library, a fully connected neural network was built which had the following initial architecture: the input layer, a hidden layer of 64 neurons, another hidden layer of 32 neurons, and the output layer. The other parameters were also arbitrarily chosen initially: 30 epochs, a learning rate of 0.001, and Adam as the optimizer.

The rationale for reporting the models first and then the pre-processing steps is for clarity purposes, as some of the pre-processing techniques we chose are mutually exclusive. Thus, attempting to lower the RMSE we experimented with different combinations of the above-mentioned models and the following pre-processing techniques.

1) Nonsensical and null values: We started our data exploration process by visualizing the distribution of each variable. The first thing that we observed, was the existence of 947 elements with 0 values in multiple variables like "n_tokens_content" and others. Logically, it does not make sense that an article has no tokens in it, thus, we decided to delete these rows. Consequently, this brought the distribution of many variables like "n_unique_tokens" closer to that of an almost normal distribution, as it can be seen in Figure 1. Moreover, we replaced the NAN values in the following variables: number of images, number of videos, and number of keywords (in the metadata) with 0 values.

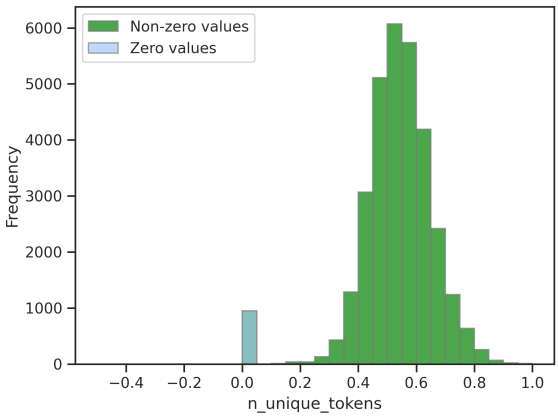

Fig. 1. The distribution of the `n_unique_tokens`.

2) Categorical variables: In our dataset there are 2 categorical variables, namely: weekday and data channel (the topic of the blog). We decided to apply one-hot encoding to these variables which consequently added dimensions to our dataset, but it is absolutely necessary for our regression task.
Yet, we observed that even though the above steps improved the RMSE by some order of magnitude, the value was still too far from the required baseline. After some exploratory research, we understood that the high dimensionality of our dataset could be the culprit. Furthermore, as it can be observed in Figure 2, there are in fact clusters of correlated variables and this is one more reason to try to limit the dimensionality of the dataset, aiming to make the number of correlated variables sparser.

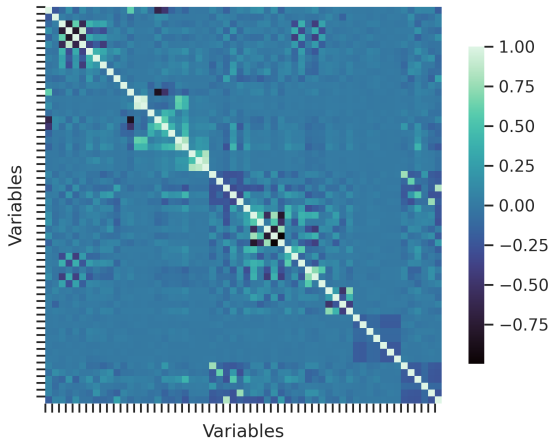

Fig. 2. The correlation matrix of all 50 variables.

3) Dimensionality Reduction: The following two dimensionality reduction techniques were utilized:

a) Fisher Score: one of the most widely used supervised feature selection methods, which is essentially *"a way of measuring the amount of information that an observable random variable X carries about an unknown parameter θ of a distribution that models X."*[1] All the variables were ranked according to the Fisher score and then we selected a subset of them, namely: top 10, top 15 and top 20. With each of these new datasets, we trained the models. The results are reported in Table I.

TABLE I
VALIDATION SCORE FOR TOP 10,15, AND 20 FEATURES.

Fisher	MR	R	L	RF	SVM	NN
10	5981.162	5980.401	5976.004	7423.999	6288.326	3691788.658
15	5972.580	5971.814	**5966.755**	7313.182	6288.215	5994.962
20	6006.457	5992.819	5981.074	6514.800	7019.118	6007.743

As it can be observed in the table, the lowest score of RMSE was achieved by using the Lasso Model (α = 10) and the top 15 Fisher score dataset. Multiple Linear Regression and Ridge Regression were not far behind, but surprisingly performed much better then the more complex models.

b) Principal component analysis(PCA): is another statistical technique for reducing the dimensionality of a given dataset [4] by linearly transforming the data into a new coordinate system. By using the "explained variance", we determined the number of principal components to use. We experimented with the following values: 33%, 60%, and 75%. As in the case of Fisher's score, we then trained all the models with the resulting datasets, and the results are reported in Table II.

TABLE II
VALIDATION SCORE FOR THE EXPLAINED VARIANCE IN PCA

PCA	MR	R	L	RF	SVM	NN
33%	6318.433	6034.486	6033.946	6309.090	6268.836	**6025.416**
60%	6361.281	6048.605	6046.537	6560.118	6270.221	6048.442
75%	6355.818	6043.681	6041.502	6211.679	6274.082	6042.543

The best score we could get with the PCA technique was through the Neural Network and 33% of explained variance. However, as it can be observed in the tables, PCA yielded worse results than the Fisher score across the models. Thus, from now and onward, we will use the datasets created from the Fisher score dimensionality reduction as they generally performed better. We then tried a basic "ensemble" technique by making a simple mean of the combined outputs of the models with the second and third lowest RMSE scores hoping to get the predicted y variable closer to its ground truth. The first combination of Top 15 Fisher Lasso and Top 15 Fisher Ridge gave an RMSE of 5968.564. The second combination of the first two and the Top 15 Multiple Regression produced an RSME of 5969.635.

4) Data Transformation: we tested two different data transformation methods.

a) Standartization: Using the scaler method available from sklearn library, we applied the z-score to all the variables of the Fisher score produced datasets. Subsequently, we trained the models with this transformed dataset and the results can be seen in Table III.

TABLE III
STANDARD SCALING OF TOP 10,15, AND 20 FEATURES.

Fisher	MR	R	L	RF	SVM	NN
10	5977.164	5977.131	5975.554	7687.764	6260.872	5988.131
15	5968.567	5968.471	**5966.361**	8821.276	6264.976	5980.205
20	5999.991	5997.325	5984.813	7097.73	6266.305	5996.667

Again, the lowest RMSE score was achieved by the Lasso model and the Top 15 Fisher score dataset.

b) Log transformation: was another data transformation method that we employed given that it can be very useful in stabilizing the variance of a given dataset and helping with skewed data. In our case, many variables have a skewed distribution and a long tail with a wide variance, such as "LDA_02" which is also our second from the top variable according to Fisher's score. Its distribution can be observed in Figure 3.

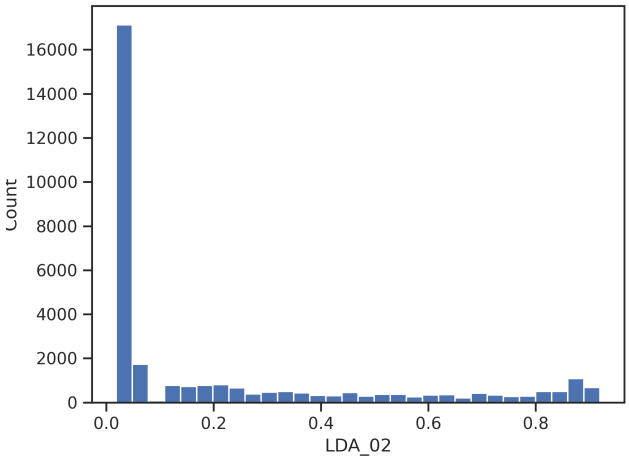

Fig. 3. The distribution of LDA_02 variable.

The results of the models with the log-transformed datasets of the Top 10, Top 15 and Top 20 Fisher variables respectively can be observed in Table IV.

TABLE IV
LOG SCALING OF TOP 10,15, AND 20 FEATURES.

Fisher	MR	R	L	RF	SVM	NN
10	6020.371	6014.905	6008.642	7496.461	6288.361	6045.731
15	6024.154	6018.355	**6008.361**	7636.247	6289.036	6044.172
20	6166.122	6055.999	6031.237	7008.025	6287.918	6055.899

5) Handling Outliers: Although outliers can be handled in a variety of manners, for the purposes of this project, we chose Winsorization. It is generally used because it makes the statistical analyses more robust against outliers. We used two different thresholds for the outliers, namely (0,01 ; 0,99) and (0,05 ; 0,95). The performance of each model is reported in Table V.

Furthermore, the regression algorithms themselves have different hyperparameters to be taken into account, and as mentioned above, we initially chose them in an arbitrary manner. Testing the different variations of the models with different hyperparameters would increase drastically the number of combinations to be considered. Thus, the configuration of the hyperparameters is automated using grid search, explained in Section B below.

TABLE V
WINDSORIZATION OF TOP 10, AND 15 FEATURES.

Fisher	Interval	MR	R	L	RF	SVM	NN
10	(0.05 – 0.95)	6006.426	6005.828	6003.18	7723.865	6278.556	6019.026
	(0.01 – 0.99)	5993.877	5992.956	5988.091	7395.714	6282.441	6013.331
15	(0.05 – 0.95)	6006.379	6004.738	5999.616	8162.312	6278.475	6016.093
	(0.01 – 0.99)	5991.818	5990.91	**5982.938**	7222.848	6282.413	6010.732

B. Hyperparameters tuning

Using the GridSreach algorithm, we tried to tune the hyperparameters of our models hoping to lower the RSME. This is a computationally expensive task and due to resources and time restrictions, we could only try a limited set of combinations. In fact, the SVM regression model was the most computationally expensive one, as it required several days to train and even like that not all the models in the grid converged. Therefore, as the model did not perform well in comparison to any experiment, we decided to focus on the other models. On average, across different models and configurations of preprocessing, the Top 10 Fisher and Top 15 Fisher were the most resilient. Considering the computational limitations we ran GridSearch only on the dataset Top Fisher 10. The following were the best hyper-parameters chosen for each model:

- ***Lasso Regression:*** $\alpha = 3.20$, score: 5970.630.
- ***Ridge Regression:*** $\alpha = 50.51$, score: **5970.018**.
- ***Random Forest Regression:*** score: 6514.800
  - criterion= squared_error
  - max_depth $= 5$
  - n_estimators $= 1000$
  - random_state$= 42$
- ***Neural Network:*** score: 5988.883
  - learning rate $= 0.0013459483238385142$
  - optimizer $=$ Adam
  - number of neurons in layer 1$= 42$
  - number of neurons in layer 2 $= 20$
  - activation function $=$ Leaky RELU
  - max_epochs $= 400$
  - dropout in layer 1 $= 0.5$
  - dropout in layer 2 $=0.5$

From the above-mentioned models, the one that yielded the lowest RSME score was Lasso regression with a score of 5966.361. Furthermore, we tried another "ensemble" technique of the top 2, top 3, and top 4 simple means of our best combinations of models and preprocessing. In general, for all the ensembles, 15 features from the fisher score are chosen.

- Top 2, score: **5965.901**.
- Top 3: 5967.181.

- Top 4: 5971.537.
- The following models were used to calculate the ensemble as an average of the output of the models.
    1) Lasso regression with standardization, $\alpha = 10$.
    2) Lasso regression without standardization, $\alpha = 10$.
    3) Ridge regression with standardization, $\alpha = 50.51$.
    4) Lasso regression without standardization and 10 Fisher Features, $\alpha = 10$.

III. Results

In essence, our approach can be described as a greedy one, given that we have little to no domain information and thus our only "stronghold" could be the exploration of as many methods as possible. After our multiple experiments that combined different models with different preprocessing techniques, it appears that the main substantial difference with regard to RSME was achieved due to the usage of dimensionality reduction algorithms.

IV. Discussion

The proposed approach for this problem and the choice of algorithms are very much centered on achieving the best possible RSME score. Thus, other criteria like the R2 score or more generic concepts like efficiency or interpretability are not taken into account and could be considered for further exploration. In addition, due to limited resources (time and mainly computational capabilities), only a small set of models and preprocessing techniques were taken into account but more can be explored. Last but not least, as mentioned in the Problem Overview section, the dataset contains only the news and references to Mashable.com. Thus, the analysis is valid only for a very particular instance and it does not imply generalization in other datasets. Further analyses would have to be made to properly understand the main features that determine shareability (spreadability) in the online domain.

References

[1] *Fisher information*. 2023. URL: https://en.wikipedia.org/wiki/Fisher_information.

[2] Henry Jenkins, Sam Ford, and Joshua Green. *Spreadable media: Creating value and meaning in a networked culture*. Langara College, 2022.

[3] *Multiple Regression*. URL: https://home.csulb.edu/~msaintg/ppa696/696regmx.htm.

[4] *Principal Component Analysis*. 2023. URL: https://en.wikipedia.org/wiki/Principal_component_analysis.