Predicting Online News Sharing

Claudio Fantasia
Politecnico di Torino
Student id: s319911
s319911@studenti.polito.it

Micol Rosini
Politecnico di Torino
Student id: s302935
s302935@studenti.polito.it

***Abstract*—This report presents the development of a regression model for predicting online news popularity. The main objective is to improve the accuracy of the model by employing advanced data preprocessing techniques and feature selection. By utilizing high-performance algorithms, the model can handle large datasets effectively and ensure accurate predictions. The comprehensive analysis of multiple features provides valuable insights into the factors influencing online news popularity, enhancing the prediction of online news popularity.**

I. Problem Overview

In today's digital age, news consumption and sharing have become integral to people's daily lives and sources of entertainment. In this project, the goal is to predict the number of shares for each described news article. The dataset provided is obtained from the Mashable website and it is arranged as follow:

- A **development** set composed of 31715 records of news. Each sample has 50 different features, including the target of the regression which is `shares`.
- A **evaluation** set composed of 7919 records of news. This set does not contain the feature `shares`.

However, with a dataset comprising 50 features, the process of feature selection becomes crucial in this regression task. The goal of feature selection is to identify the most informative and relevant features that significantly contribute to the accurate prediction of online news popularity. There are features, like the `id`, `url`, and `timedelta`, that have no predictive power. Other features contain missing values, such as `num_imgs`, `num_videos`, and `num_keywords`. The dataset includes two categorical features that require specific considerations in the regression analysis: `data_channel` and `weekday`. Furthermore, there are features containing negative values that represent attributes like sentiment score and subjectivity expressed in the news. Moreover, the dataset does not contain any duplicate records, this characteristic simplifies the feature selection process, as redundant data points can skew the model's performance.
The full feature set is mainly categorized as in Table I [1].
To evaluate the performance of the regression pipeline, the Root-Mean-Square Error (RMSE) will be used as the evaluation metric.

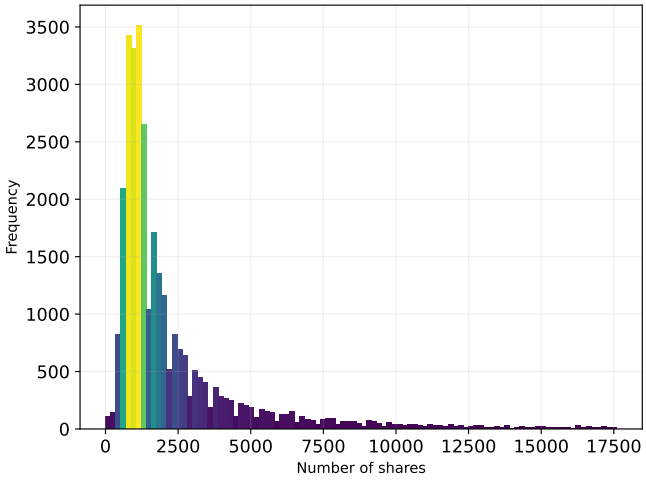

Fig. 1: Histogram of the feature `shares` without outliers (i.e. number of shares bigger than 17500).

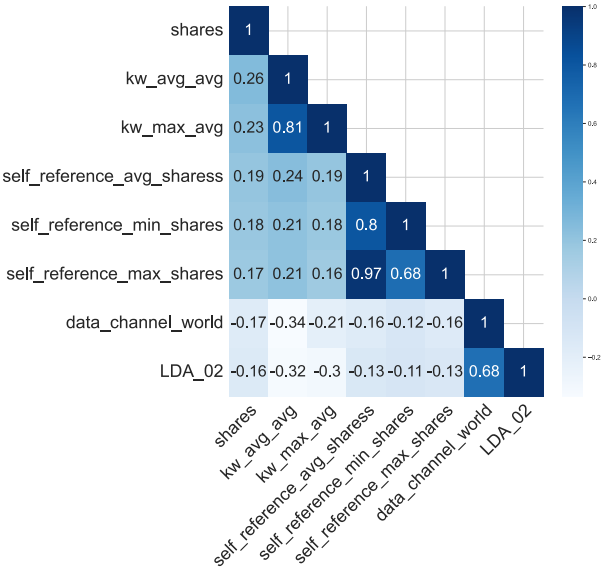

Fig. 2: Heat map depicting the correlation between the **7** features that exhibit the highest correlation with the `shares` variable using *Spearman* correlation.

Aspects	Features
Metadata	Id/Url; Days elapsed since article publication
Word	Number of words of the title/content; Average word length; Rate of unique/non-stop words of contents
Links	Number of links; Number of links to other articles in Mashable
Digital Media	Number of images/videos
Data channel	Type of data channel
Publication Time	Day of the week/weekend
Keyworkds	Number of keywords; Worst/best/average keywords (#shares);
References	Min/Max/Avg shares of referenced articles
NLP	Closeness to five LDA topics; Title/Text polarity/subjectivity; Rate and polarity of positive/negative words; Absolute subjectivity/polarity level
Target	Number of shares

TABLE I: All Available Features within their category.

II. Proposed approach

A. Preprocessing

One of the typical problems concerning regression tasks is the identification and correct elimination of outliers. Indeed, this has been the most difficult issue to deal with. We had to make a careful selection of the different features from which to remove outliers and which could not be removed due to the too large number of data we would be removing.

The feature `shares` in the dataset exhibits numerous outliers, which pose challenges for the model's ability to converge and achieve optimal predictions. To address this issue, a robust approach was employed to identify and remove outliers: news articles with a number of shares falling outside the range of

$$[Q1 - 8 \times IQR,\ Q3 + 8 \times IQR]$$

were classified as outliers and subsequently eliminated from the dataset. Here, $Q1$ and $Q3$ represents respectively the first and the third quartile and IQR denotes the interquartile range. We chose the factor 8 because it was the one giving the best performance. This data preprocessing step aims to mitigate the impact of extreme values in the `shares` that could arise due to some unusual human behaviour or other factors feature.

Fig.1 shows the distribution of the feature `shares` without the outliers. We would like to mention that we tried to cap our outliers at a certain upper and lower threshold instead of removing them altogether, but this did not lead to the desired results, creating unusual data distributions

We also removed the non predictive variables like `id`, `url`, and `timedelta`, since are more related to identification and metadata rather than content characteristics. Categorical feature like `data_channel` and `weekday` are encoded using one-hot encoding. In addition, looking at the distribution of our features via histograms and boxplots, we observed several outliers that appeared in the features `average_token_length` and `n_tokens_content`: a significant number of values were set to zero and, given that it is highly unlikely for news articles to have a content with zero tokens or an average token length of zero, we considered these values as potential noise or mistakes in the data, so we opted to remove these news articles from our dataset, which ultimately improved the performance of our predictions..

To address missing values in the `num_imgs` and `num_videos` features, we performed data imputation by filling the NaN values with zeros. Therefore, by filling the `NaN` values with zeros, we ensure consistency in the dataset and prevent the omission of potentially relevant information. Instead, for treating the NaN values in the feature `num_keywords` we grouped the dataset by data channel and set the missing values to the mean for each data channel without second thoughts, because it is very odd for a news to have zero number of keyword. And this was also confirmed by the fact that there are no zero values in the other records present in $num\_keywords$.

To address skewed distributions and non-bell-shaped patterns observed in certain features, we applied a log transformation. This technique helps normalize the data and reduce the influence of extreme values.

Fig.3 shows the effect of this transformation in some features.

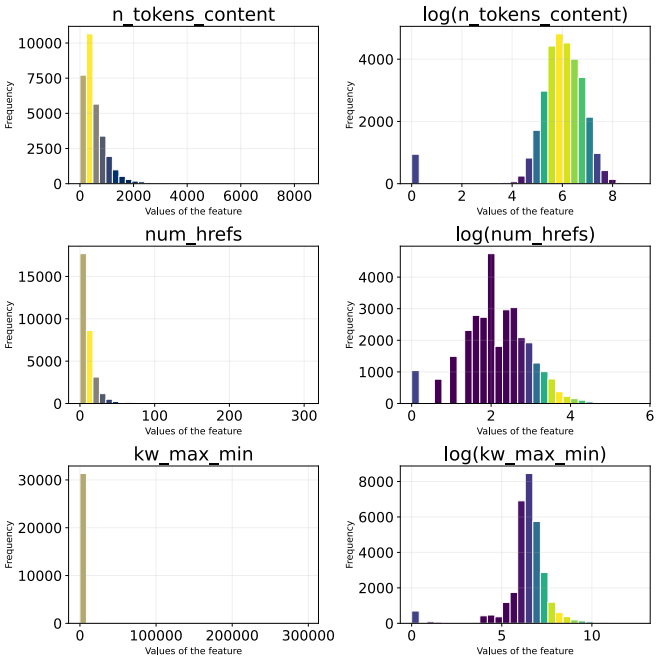

Fig. 3: Histograms of feature with and without log transformation. In the figure are depicted the following features: `n_tokens_conten`, `n_hrefs`, `kw_max_min`.

After conducting an in-depth analysis of the features that exhibit the strongest correlations with the number of shares,

employing various correlation analysis techniques such as Pearson, Spearman, Kendall, and feature importance measured from a random forest regressor, along with the Fisher criterion [1], we attempted to train the model exclusively on this subset of features obtaining unsatisfactory resulting predictions.
To address this issue, we shifted our focus to **examining the correlation between features themselves**. We recognized that the inclusion of strongly correlated features (i.e., with a correlation coefficient exceeding 0.7) in the regression model can lead to detrimental effects. Therefore, a decision was made to remove some of the features within each highly correlated pair. To determine which feature to keep among the correlated features, we eliminated the features less correlated to our target value using *Spearman* correlation, which is particularly suitable for capturing monotonic relationships. The feature that exhibited the highest correlation with the target variable, `shares`, was retained, while the other feature was removed from the model.
Fig.2 reports the 7 features most correlated with shares using *Spearman* analysis.
This selection process ensures that the retained features maintain the strongest association with the target variable, improving the model's predictive performance.
Fig.4 illustrates the stepwise process employed for feature selection in this analysis.

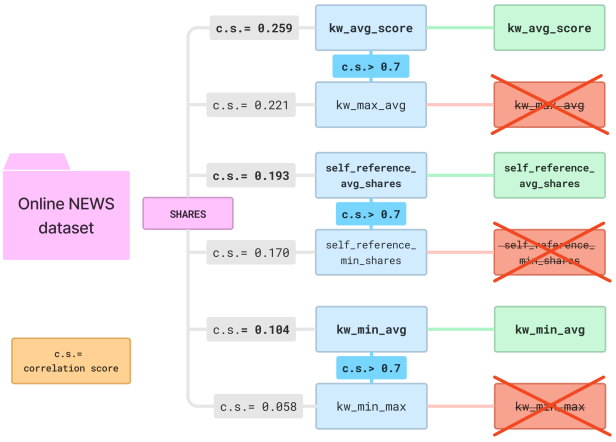

Fig. 4: The stepwise process employed for feature selection based on correlation analysis: if between two features there is high correlation (> 0.7), the one with lower correlation score with the label is dropped.

Finally, we drop also the outliers of features that are strongly correlated with shares like `kw_avg_avg`, `self_reference_avg_sharess` and `kw_avg_min` to mitigate the impact of noisy data and the potential presence of extreme values, which could introduce significant variance into the distributions of these features.
In an effort to further improve the predictive performance of our model and to reduce correlation between our features, we explored additional techniques such as Principal Component Analysis (PCA). We applied PCA [2] to a subset of components that exhibited higher correlations between eachother and that belong to the same semantic group (i.e. $token$,$polarity$,kw_data) with the goal of obtaining new features that capture as much variance as possible within the data and substituting them to the original ones.
However, despite these attempts, we did not observe any significant improvement in the model's prediction accuracy. It is possible that the nature of the dataset and the specific relationships between features and the target variable may not be well-suited for the application of PCA or feature removal based on correlation.

To conclude, after our preprocessing steps we ended up with 28189 data points left, so with the 89% of our initial dataset. Keeping 46 features.

B. Model selection

The models that we have considered for this classification task are:

- **Linear Regression:** the classical and fast regression model that works well with linear problem
- **Support Vector Regression:** we chose this model because it is very powerful at addressing non linear regression like the one that we are facing. Using a proper kernel function(e.g. Gaussian or Polynomials Kernels) we can find an optimal solution. [3]
- **Random Forest Regressor:** selected because it is known to have good performances with high dimensionality data [4] with different scales. This algorithm use ensemble methods that combine multiple decision trees where each tree is trained indipendently on different subsets of the training data and the final prediction is obtained by averaging the predictions of individual trees.
- **Gradient Boosting Regressor:** This model is an ensemble method that use weak learners as Random Trees like the previous model. But they have different training process. Gradient Boosting Regressor builds decision trees sequentially, each one attempting to correct the errors of the previous ones. [5]

Although our algorithms are particularly sensitive to the scale of the data, we decided to apply logarithmic transformation to some features , but not to normalize using z-score or other normalisation techniques the entire dataset, because by normalizing the data we had far worse performance, especially using Random Forest Regressor.

As shown in Table II, we conducted experiments without removing any features or outliers from the dataset obtaining very poor performances since the presence of outliers introduced noise and increased the variance in the data, making it more challenging for the model to learn meaningful patterns and, similarly, including all the features, regardless of their relevance or correlation with the target variable, hindered the

Techniques	Regressor model	RMSE
Original dataset	Linear Regression	9413
	Random Forest Regressor	10258
	SVR(kernel = 'poly')	9653
	SVR(kernel = 'rbf')	9651
	Gradient Boosting Regressor	10275
Top 10 - Spearman with shares	Linear Regressor	2301
	Random Forest Regressor	2362
	SVR(kernel = 'poly')	2509
	SVR(kernel = 'rbf')	2508
	Gradient Boosting Regressor	2294
PCA	Linear Regressor	2301
	Random Forest Regressor	2332
	SVR(kernel = 'poly')	2523
	SVR(kernel = 'rbf')	2528
	Gradient Boosting Regressor	2290
F.S. 2-correlated variables	Linear Regressor	2335
	Random Forest Regressor	2301
	SVR(kernel = 'poly')	2522
	SVR(kernel = 'rgb')	2522
	Gradient Boosting Regressor	2322

TABLE II: Results of the model selection phase.

Model	Parameters	Value
SVR	rbf	0.1
	poly	1
		10
Random Forest	n_estimators	100, 200, 300
	max_depth	None, 10, 35
	min_samples_split	2, 5, 10
Gradient Boosting	n_estimators	100, 200, 300
	max_depth	3, 4, 5
	learning_rate	0.01, 0.05, 0.1

TABLE III: Hyperparameters configuration considered.

Regressor model	Parameters	RMSE
Gradient Boosting Regressor	n_estimators 200 max_depth 3 learning_rate 0.05	2289
Random Forest Regressor	n_estimators 300 max_depth 35 min_sample_split 2	2283

TABLE IV: Results of the models with the best hyperparamters configuration.

model's ability to accurately predict the online news popularity. For this reason, we conducted several experiments to evaluate different feature selection techniques on the model's performance. We first employed Spearman correlation analysis to identify the top 10 features that exhibited the highest correlation with the target variable. However, the results obtained from this approach did not yield optimal performance. Next, we attempted to apply Principal Component Analysis (PCA) on subsets of features that appeared to be correlated. The goal was to substitute these subsets with their principal components, which capture the most significant variance. Unfortunately, this approach did not lead to improved results. Finally, finding high correlation values between some pairs of variables and dropping the feature that exhibited the lower correlation with the target variable proved to be the most effective technique in improving the model's predictive capabilities.

C. Hyperparameters tuning

From the model selection phase, the best dataset turned out to be the two-most-correlated feature selection, therefore the models have been working with it. We decided to tune SVR, Random Forest Regressor and Gradient Boosting regressor since all these models obtains good results. To tune our models we used a Grid Search with cross-validation (cv $= 5$) over the parameters reported in Table III.

III. Results

In this section are reported the main outcomes of the hyperparameters tuning phase. Table IV showcases the best configuration and the RMSE results for the most successful models.

Among all the models we explored and the different feature selection techniques we applied, we found that the Random Forest Regressor, combined with feature selection, proper logarithmic transformation and outlier removal, proved to be the optimal configuration for our task, as it surpassed the baseline performance and provided reliable predictions of online news popularity.

Based on slightly improved results in the training set, we decided to submit the evaluations performed by Random Forest Regressor with parameters: **`n_estimators`** $= 300$, **`max_depth`** $= 35$ and **`min_samples_split`** $= 2$.

IV. Discussion

The high dimensionality of the dataset necessitates a comprehensive and rigorous data preprocessing process. While selecting the appropriate model is crucial, the most significant improvements often stem from the steps taken during data preprocessing. These steps involve handling missing values, addressing categorical features, applying some normalization techniques to numerical data, and applying feature selection or deleting outliers.

There is a large difference in RMSE score obtained for the training and test sets. This happens because by removing outliers from the target value and other important features, we are changing the distribution from which our train data are sampled.

We implemented a technique that yielded good results, however, it's important to acknowledge that further analysis and exploration of feature performances could potentially lead to even greater improvements. Conducting a deeper and longer investigation into the behavior and relevance of individual features, assessing their correlations and interactions, and considering domain-specific knowledge can reveal additional opportunities for refinement. Moreover, the integration of a deep neural network could further enhance the task by leveraging its capacity to learn complex representations and capture intricate patterns and relationships in the data.

REFERENCES

[1] Q. Y. He Ren, "Predicting and evaluating the popularity of online news,"

[2] K. L. Yuchao Zhang, "Predicting and evaluating the online news popularity based on random forest,"

[3] P. R. Tomasz Trzcinski, "Predicting popularity of online videos using support vector regression,"

[4] Y. J. Feras Namous, Ali Rodan, "Online news popularity prediction,"

[5] M. J. A. P. Taufeeq Uddin, "Predicting the popularity of online news from content metadata,"

1

[1]All the materials employed for this paper are available at the project repository: https://github.com/micolrosini/Data-Science-Lab