Predicting News Articles Popularity

Maham Maham, Ghassane Ben El Aattar
Politecnico di Torino
Student ids: s314696, s290047
s314696@studenti.polito.it, s290047@studenti.polito.it

***Abstract*—In this report, we show a possible approach to the shares prediction of an article regression problem. The proposed approach uses a Random Forest Regressor model after removing outliers from the data, performing feature selection, encoding categorical features with One-Hot Encoding and scaling the data using a Standard Scaler. The results have been evaluated using the Root Mean Square Error, and it is observed that the random forest regressor model works better than the ridge.**

I. Problem overview

This project aims to build a regression pipeline to predict the number of shares for online articles.

Predicting the number of shares of any news article determines the popularity of that article, and this information can be obtained for news-related online channels, magazines, and advertisers. If the number of shares is accurately predicted, the information can prove to be helpful for these media agencies to work more efficiently and generate more revenue.

There are two different parts of the dataset:

- A development set: containing 31,715 records with a 'shares' column for the training and validation of the model.
- An evaluation set: containing 7917 records without 'shares' column.

The dataset contains both numerical and categorical features. The distribution of the dataset is given in the table.

TABLE I
Features Information

Type	Total Columns
Numerical	47
Categorical	3

While majority of the features contain numerical data, the three columns with categorical data are 'url', 'data_channels' , and 'weekday'.

We ran a few techniques under exploratory data analysis to understand our data better. Our dataset had outliers, which is something that is specifically addressed in the preprocessing step.

We can see in Figure 1 the distribution of the attribute 'shares', which shows the presence of extreme values in the dataset.

Also, two of the features in the dataset were non-predictive, 'id' and 'url', although further analysis can be made on the latter.

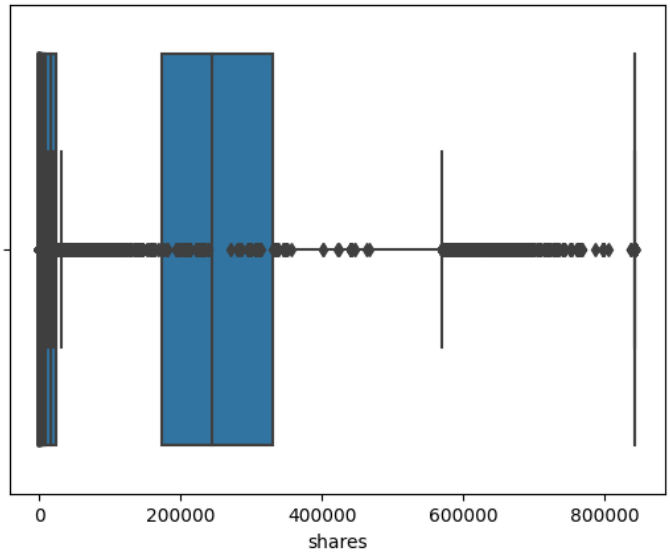

Fig. 1. Outliers in the dataset

II. Proposed approach

A. Preprocessing

The first preprocessing step was to perform feature selection.

We decided to remove the 'id' and 'url' feature. The 'url' feature contains the link of the article, which also includes a suffix of linked words which characterize the topic of it.

We attempted to split this suffix into unique tokens and build a TF-IDF matrix with new features, but this implied a growth of the number of features to several thousands, which would have made building a regression model too computationally expensive, given the large number of records in the dataset.

We also removed 5 other features from the dataset, which were those with the lowest computed importance by the Random Forest Regression. Since regression is not robust to outliers, we removed them using the Inter Quartile Range (IQR) method. We could identify the outliers by calculating the IQR values for each numeric column; hence, those values were removed.

The range of values to remove with respect to the IQR can make quite a difference in the analysis. For example, choosing a range such as +/- 1.5*IQR removes 85% of the data. This is why we attempted different ranges of values in this step, as will be shown later.

After removing the outliers, we identified our numerical and categorical columns to further preprocess the data. As shown in Table 1, only 3 of the 50 features are categorical. We then checked the missing values in both the development and evaluation dataset.

TABLE II
DEVELOPMENT AND EVALUATION SET

	Development set	Evaluation set
num_images	6375	1552
Num_videos	6331	1594
num_keywords	6318	1608

For the missing numerical values, we computed the mean of the values and filled in the missing values with those.

We can see from Table II the attributes with missing values and the number of them in both datasets. Although those numbers are not negligible, the number of missing values for a single feature reaches at most 20%, which we thought would not justify removing those features completely.

To streamline our preprocessing steps and to ensure consistency, we merged the development and evaluation datasets into a single dataset.

For the easy computation of the categorical data, we applied One-Hot Encoding. Our dataset's only categorical columns were 'data_channels' and 'weekday'. There are six categories of 'data_channels' and seven categories of 'weekday'. Therefore, now the total features are 62.

This transformation did not create an excessive amount of features, while retaining some potentially useful information for our model. This justified our choice to handle the categorical features in this way.

Before actually running any regression algorithm, we split the combined dataset back to the development and evaluation sets. Then, we split the development set again into the training set and validation set, by choosing the default 80/20 split (training/validation).

Finally, we decided to perform Standard Scaling on the data. This was doing by fitting the scaler with the training set and transforming all sets with it.

Although scaling is not strictly needed in Tree based algorithms such as Random Forest, it's potentially useful in Ridge regression algorithms.

B. Model selection

To perform our regression task, we worked with these two standard algorithms:

- Random Forest: This algorithm has the advantage of computing the feature importance, which has been useful for feature selection purposes.
- Ridge: As more than 20% of the features in the dataset after performing feature selection and One-Hot Encoding were obtained through the latter, we assumed there would be several highly negatively correlated features. When a regression model has a high correlation between predictor variables, multicollinearity can arise, which is well handled by Ridge [1].

C. Hyperparameters tuning

Both algorithms have different sets of hyperparameters that could be tuned. For Random Forest, we evaluated the values of the following two parameters:

- n_estimators: The number of trees in the forest.
- max_depth: The maximum depth of a tree.

To verify results with different values for each hyperparameter, we performed a grid search with K-Fold cross validation for validating, with k = 5. For Ridge, the only hyperparameter we tried tuning is α, which is a constant that multiplies the l^2 term, controlling regularization strength [2]. We also used K-fold cross validation for validation with k = 5 for the Ridge model as well.

III. RESULTS

Values w.r.t. IQR	Random forest param.	Ridge param.
+/- 1.5*IQR	n° estimators = [50, 62, 100]	α = logspace(-4, 4, 20)
+/- 5*IQR	max depth = [None, 10, 20]	
+/- 7*IQR		

TABLE III
DIFFERENT PARAMETERS USED FOR THE MODELS

We evaluated all our results using the Root Mean Square Error (RMSE). Apart from the regression hyperparameters, we also decided to test different ranges of values of numerical data with respect to the interquartile range.

The ranges +/- 1.5*IQR and +/- 5*IQR removed most of the data entries, while also performing worse than the +/- 7*IQR range of values.

When it comes to the regression hyperparameters for the Random Forest algorithm, the best result obtained was with n° estimators = 62 and no maximum depth, the latter being the default parameter in sklearn [2].

For the Ridge regression model, the best parameter of α was α = 206. In our case, Random Forest performed significatively better than Ridge, always yielding a smaller error no matter the hyperparameters used.

As mentioned before, the Random Forest Regressor can compute the feature importance in the model, which helped us in performing further feature selection. By removing the 5 features with the least computed importance we were able to further improve the error, which was greater than 6020 before this feature selection step. The lowest RMSE obtained overall on the public evaluation set was 5988.357.

IV. DISCUSSION

This study proposes an approach to predict the number of shares of online news articles. The regression models were trained on 31,715 data instances and evaluated on 7917.

The effectiveness of the results is declared when the RMSE values surpass a certain baseline. The approach proposed in this study can be considered an initial effort to solve this problem. Many other improvements can be made to make better decisions based on this problem. For example,

• Convolutional neural networks can be used for prediction.

• More sophisticated techniques for treating categorical data can be introduced for this problem.

• Advanced feature selection algorithms can be applied as well.

It's also possible that with high computational power we could have extracted all potential information from the feature 'url', although an improvement in performance is not guaranteed.

Our experiments' outcomes are encouraging; however, these outcomes can be improved with more experimentation and model comparison.

REFERENCES

[1] A. Bager, Y. Roman, M. Algedih, and B. Mohammed, "Addressing multicollinearity in regression models: a ridge regression application," 2017.

[2] SciKitLearn, "Documentation: sklearn.linear model.ridge.,"