# A Weighted Ensemble Learning Approach for Forecasting Online Article Popularity

Lorenzo Spanio
*Politecnico di Torino*
Student id: s327504
s327504@studenti.polito.it

***Abstract*—This study forecasts online news article popularity using a weighted ensemble of Random Forest, LightGBM and Ridge. The framework captures complex data patterns, achieving highly competitive results.**

## I. Problem overview

This study addresses the challenge of predicting the virality of online news articles, a task of significant interest to media organizations, advertisers, and communication scientists. The primary objective is to develop a regression model to forecast an article's popularity, as measured by its total number of social media shares. The analysis is based on a dataset partitioned into a development set of 31,715 labeled instances, used for model training and validation, and an evaluation set of 7,917 instances for which the share count was withheld. Each instance represents a unique online article and is described by a feature space of 50 attributes. These features can be broadly classified into several categories:

- Content Metrics: quantitative text descriptors, such as word counts for the title and content, the rate of unique words.
- Topical Features: indicators of subject matter, including the article's publication channel (e.g., 'Tech', 'Business') and its affinity to five pre-computed Latent Dirichlet Allocation (LDA) topics.
- Sentiment and Subjectivity: Metrics derived from sentiment analysis, such as polarity and subjectivity scores for both the title and content.
- Inter-Article Reference Features: statistics of the number of articles referenced within the current article.
- Keyword Statistics: features reflecting the historical popularity of an article's keywords, quantified by their minimum, maximum, and average share counts in previous publications.
- Media and Link Counts: simple counts of embedded images, videos, and hyperlinks.

Preliminary data exploration reveals two principal challenges. First, the target variable, shares, exhibits a heavily right-skewed distribution, where a vast majority of articles receive a modest number of shares while a small fraction achieves extreme virality. To mitigate the impact of this skew on model training, a logarithmic transformation was applied, resulting in the more symmetric distribution shown in Figure 1. This characteristic can disproportionately influence model training and makes the evaluation metric, Root-Mean-Square Error (RMSE), highly sensitive to prediction errors on these popular articles.

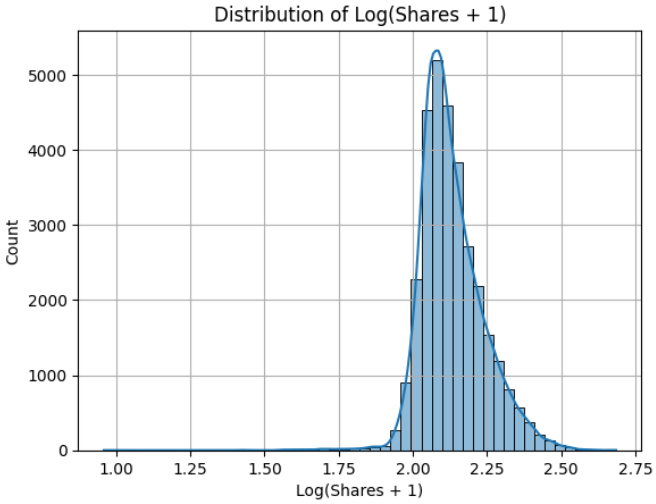

**Fig. 1** Distribution of the log1p-transformed target variable, log(shares + 1). This mitigates the extreme right-skew of the original share counts, resulting in a more symmetric, approximately normal distribution suitable for model training.

## II. Proposed approach

The analytical approach was designed to address two principal characteristics of the dataset: the heavily right-skewed distribution of the target variable and the heterogeneous composition of the feature space. The former was addressed by applying a log(shares + 1) transformation, selected over the standard logarithm to accommodate articles with zero shares. Hereafter, all references to the 'logarithmic space' or 'log transformation' imply this log1p operation. The latter, consisting of continuous, discrete, and categorical data types, required a sophisticated preprocessing pipeline.

The proposed solution is a multi-stage regression framework centered on a weighted average ensemble model. The complete workflow is partitioned into three key phases: preprocessing, model selection and hyperparameters tuning.

### A. Preprocessing

To enhance the predictive power of the models, the initial preprocessing step involved enriching the original feature set. A series of new attributes were engineered to extract more granular and context-aware signals from the data:

- Interaction Features: interaction terms were created to capture their influence when together, e.g combining the data channel with the day of the week or multiplying sentiment scores by article length.
- Ratio-Based Features: to normalize for article length and content, I computed ratios like images per word or images per video and many more.
- Temporal Features: the publication date was extracted from the article's URL, enabling the creation of features such as year, month, day of the week, and week of the year.
- Cyclical Features: to better capture the cyclical nature of time, the temporal features were further transformed using sine and cosine functions. For instance, a simple numerical representation of months would incorrectly suggests that December is distant from January. This mapping allows the model to correctly understand that the end of a cycle is immediately followed by the beginning of the next.
- Slug-Based Features: the article's URL slug was processed to extract features like its length, whether it contained numbers, and the presence of keywords like ”video”. The slug itself was retained for text-based feature extraction.
- Binning: several key numerical features with skewed distributions were discretized into categorical bins. This was achieved using both quantile-based binning and fixed-width binning for features where specific thresholds were more meaningful.
- Polynomial features: to capture non-linear effects, second-degree polynomial features were generated from a targeted subset of predictors. This subset consisted of the seven most influential features, determined by the feature importance scores from a preliminary LightGBM model.

Data exploration revealed a significant inconsistency between the provided weekday column and the publication date extracted from the article's URL, likely caused by authors publishing across different timezones. To ensure data consistency, the URL was treated as the ground truth to engineer a corrected day-of-the-week feature, and the original, erroneous column was discarded to prevent model noise. This engineering process yielded two design matrices: a 359-column matrix with polynomial features for the Ridge model, and a 324-column matrix for the tree-based models. Polynomial terms were omitted for the latter, as tree ensembles inherently capture non-linear interactions. This high dimensionality is appropriate for the large dataset; tree models are robust to numerous features, while Ridge regression's regularization manages multicollinearity and prevents overfitting in this wide feature space.

Two distinct preprocessing pipelines were constructed to accommodate the different requirements of our models. For tree-based models, depending on what type of features they go through different transformations:

- Numerical Features: missing values are imputed using the median. Scaling is omitted as tree-based models are insensitive to feature scale.
- Categorical Features: missing values are imputed with the most frequent value, followed by one-hot encoding.
- Text Features: the text from the URL slug is vectorized using TF-IDF to create a set of 100 features representing the important unigrams and bigrams, so that the model can capture the relationship of two words such as ”New York”, not from just single words. This was done only using data in the training text.

For the linear model: this pipeline included all the steps from the tree-based preprocessor but added two crucial ones:

- Polynomial Features: to allow the linear model to capture non-linear interactions, a select subset of the most important numerical features underwent polynomial feature expansion, of degree, before being scaled.
- Scaling: all numerical features were scaled using standard scaler after imputation. This is essential for regularization-based linear models whose performance depends on features being on a comparable scale.

### B. Model Selection

The model selection strategy was centered on creating a diverse and robust ensemble. Given the tabular nature of the data and the complexity of the prediction task, tree-based algorithms were identified as a strong starting point. However, instead of relying on a single algorithm, two distinct yet powerful tree-based models were chosen to form the core of the ensemble: Random Forest and LightGBM. The decision to include both was strategic, as they address the bias-variance tradeoff from complementary angles:

- Random Forest operates on the principle of bagging, constructing many deep decision trees independently on data subsets and averaging their predictions. This process is highly effective at reducing variance, making the model robust and less prone to overfitting. It works by combining multiple complex, high-variance models into a single low-variance predictor.
- LightGBM is a gradient boosting model that builds trees sequentially. Each new, shallow tree (a weak learner) is trained to correct the errors of the preceding ones. This sequential, error-correcting process reduces bias, allowing the model to capture intricate patterns that bagging methods might overlook.

By employing both, the ensemble leverages two fundamentally different approaches to learning. The models are likely to make different types of errors, which can be partially canceled out when their predictions are combined. To further enhance this diversity, a Ridge model was included. While Random Forest and LightGBM are powerful non-linear models, the inclusion of a regularized linear model like Ridge introduces a completely different modeling paradigm. It can effectively capture simple linear trends that the more complex tree models might overlook, providing unique, non-redundant information and helping to reduce the overall prediction error, as demonstrated in [1].

### *C. Hyperparameters tuning*

To maximize the performance of Random Forest and LightGBM, I employed Optuna. A Bayesian optimization approach, specifically the Tree-structured Parzen Estimator (TPE) sampler, was chosen for its efficiency over random or grid search, as it leverages the results of past trials to intelligently select new hyperparameter combinations. The evaluation of each trial was conducted using 3-fold stratified cross-validation. To achieve a stable and reliable validation score, the stratification was performed based on the target variable's distribution. Specifically, the log-transformed target variable was first binned into 10 deciles, creating a discrete categorical representation. This binned array was then used as the stratification key for the cross-validation splitter, ensuring that the distribution of article popularities in each of the three validation folds was representative of the overall dataset. Moreover, to prevent overfitting and to find the optimal number of boosting rounds an early stopping mechanism was implemented for LightGBM. The training was halted if the validation RMSE did not improve for 100 consecutive rounds. The optimal hyperparameters for the tree-based models, determined through the tuning process, are presented in Table III for LightGBM and Table IV for Random Forest.

| **Parameter** | **Type** | **Interval** | **Dist.** |
|---|---|---|---|
| `learning_rate` | Float | [0.01, 0.08] | Log |
| `num_leaves` | Integer | [20, 300] | Unif. |
| `max_depth` | Integer | [5, 15] | Unif. |
| `min_child_samples` | Integer | [10, 100] | Unif. |
| `colsample_bytree` | Float | [0.5, 1.0] | Unif. |
| `subsample` | Float | [0.5, 1.0] | Unif. |
| `reg_alpha` | Float | [0.01, 20.0] | Log |
| `reg_lambda` | Float | [0.01, 20.0] | Log |

**Table I** : Hyperparameter search space for the LightGBM model. Using L2 as loss and RMSE as metric. The standard Gbdt is the boosting algorithm chosen. Logarithmic distrution for some hyperparameters to focus the search more intensely on the smaller values in the range and spreads the search out more as the values get larger.

| **Parameter** | **Type** | **Interval** |
|---|---|---|
| `n_estimators` | Integer | [100, 700] (step 50) |
| `max_depth` | Integer | [2, 30] |
| `min_samples_split` | Integer | [2, 20] |
| `min_samples_leaf` | Integer | [2, 20] |
| `max_features` | Categ. | [’max_features’, ’sqrt’, 0.5, 0.8] |

**Table II** The hyperparameter search space for Random Forest.

| **Parameter** | **Optimal values** |
|---|---|
| `learning_rate` | 0.0119 |
| `num_leaves` | 119 |
| `max_depth` | 12 |
| `min_child_samples` | 91 |
| `colsample_bytree` | 0.9004 |
| `subsample` | 0.8713 |
| `reg_alpha` | 0.3351 |
| `reg_lambda` | 12.2956 |

**Table III** Optimal hyperparameter values for LightGBM.

| **Parameter** | **Optimal values** |
|---|---|
| `n_estimators` | 700 |
| `max_depth` | 26 |
| `min_samples_split` | 9 |
| `min_samples_leaf` | 4 |
| `max_features` | ’sqrt’ |

**Table IV** Optimal hyperparameter values for Random Forest.

The model training protocol employed a 5-fold stratified cross-validation scheme. Stratification was performed on binned quantiles of the log-transformed target variable to ensure that each fold contained a representative distribution of share counts, as seen in the previous section. For the linear model component, Ridge regression was implemented using the RidgeCV method for efficient hyperparameter optimization. The regularization strength, alpha, was determined by an internal cross-validated search across a logarithmic grid of 100 values ranging from $10^{-3}$ to $10^{3}$.

The performance metric used to evaluate models is the Root Mean Squared Error (RMSE). However, due to the severe right-skew of the target variable, all selected models were trained to predict the logarithm of shares. Consequently, the direct optimization and validation metric was the RMSE on the log-transformed values, i.e., the Root Mean Squared Logarithmic Error (RMSLE). Model performance was assessed via out-of-fold (OOF) predictions.

To generate the final predictions, I used a weighted-average ensemble of all models:

$$\hat{y}_i \;=\; \sum_{m=1}^{M} w_m \, \hat{y}_i^{(m)}, \quad w_m \geq 0, \;\; \sum_{m=1}^{M} w_m = 1.$$

The weights were chosen by minimizing RMSLE on OOF predictions from the development set, so they were learned on data held out from each model's training. This constrained optimization problem was solved using Sequential Least Squares Programming (SLSQP). The learned weights were then applied to the evaluation-set predictions to produce the final submission.

Note that naively exponentiating log-space predictions to return to the original scale is theoretically unsound: such a retransformation targets the conditional median rather than the conditional mean and thus induces a systematic underestimation bias, a consequence of Jensen's Inequality for the convex exponential function. To correct this back-transformation bias, I then applied the non-parametric Duan smearing estimator [2]. Let the log-space residuals be

$$e_i \;=\; y_i^{(\log)} - \hat{y}_i^{(\log)}, \qquad i = 1,\dots,n,$$

and define the smearing factor

$$S \;=\; \frac{1}{n}\sum_{i=1}^{n}\exp(e_i).$$

Predictions on the original scale are then obtained as

$$\hat{y}_i \;=\; \exp\big(\hat{y}_i^{(\log)}\big)\,S \;-\; 1.$$

Here, $y_i^{(\log)}$ are the ground truth on the log1p scale, $\hat{y}_i^{(\log)}$ are the out-of-fold log1p predictions, $y_i$ are the ground truth on the original scale, and $\hat{y}_i$ the corresponding back-transformed prediction.

Two things should be noted here:

- The Duan Smearing correction was not applied when calculating the RMSE for model comparison. The goal of this metric was to assess the relative performance between models, not to generate the final, optimally-scaled predictions.
- The standard Duan Smearing method does not involve a final subtraction since it assumes the logarithm of the target variable. However, since log1p transformation was used instead of the standard logarithm the subtraction is needed.

## III. Results

The optimized ensemble weights in Table V clearly highlight LightGBM as the primary contributor to the final prediction. Conversely, the low weight assigned to Random Forest indicates that its predictions were largely redundant, providing little new information beyond what LightGBM already captured. Notably, the Ridge model retained a small positive weight, indicating it provided unique, non-redundant information that helped reduce the overall prediction error.

The proposed ensemble achieved a highly competitive RMSE on the evaluation set, ranking among the top entries on the public leaderboard. This strong performance stems from key methodological decisions: logarithmic target transformation, extensive feature engineering, stratified cross-validation, and the weighted ensemble itself. While the ensemble performed best, the individual base models also demonstrated strong predictive capabilities on their own, as detailed in Table VI.

| **Weight_Random_Forest** | **Weight_LightGBM** | **Weight_Ridge** |
|---|---|---|
| 0.1556 | 0.8009 | 0.0434 |

**Table V** Optimal weights for the final weighted average ensemble. Weights were determined by minimizing the out-of-fold RMSLE on the development set.

| **Model** | **RMSLE** | **RMSE** | **RMSE evaluation set** |
|---|---|---|---|
| Baseline | 1.1381 | 11899.27 | 6092.877 |
| LightGBM | 0.8410 | 11,900.42 | 5915.584 |
| Random Forest | 0.8469 | 11,916.23 | 5927.018 |
| Ridge | 0.8564 | 11,917.68 | 5923.340 |
| Weighted average of the models | 0.8406 | 11903.50 | 5912.627 |

**Table VI** RMSLE and RMSE on the out-of-fold. The third column contains the RMSE values the models obtain on the public evaluation set. The baseline is predicting the mean of shares.

## IV. Discussion

The ensemble's success confirms that a hybrid approach is essential for predicting article virality. By outperforming any single component, the model proves that popularity is governed by a mix of linear and non-linear dynamics that can only be captured by combining diverse algorithms.

The primary limitation of this study is the absence of the article's full text, the model relies on pre-computed sentiment and topic scores (LDA), which may not capture the full nuance of the content. The use of modern transformer-based embeddings could significantly improve the capability of understanding the content of the title and of the articles themselves. Furthermore, the model has no information about the article's author or external events that could heavily influence its virality.

Future improvements could focus on increasing ensemble diversity. For instance, substituting Random Forest with more distinct algorithms such as XGBoost, CatBoost, or even a Support Vector Regressor could introduce new learning strategies and improve the model's overall robustness. Additionally, a systematic feature selection process could remove noisy features to improve model performance and interpretability, especially for the Ridge. Model robustness could also be enhanced with a more rigorous validation framework. The current method performs data-dependent preprocessing (e.g., quantile binning) on the full dataset before cross-validation, which risks leaking information from validation folds and inflating performance metrics. Although encapsulating preprocessing within each cross-validation fold is methodologically superior, the current single-pass approach was chosen for its computational efficiency.

## References

[1] S. Kumar, M. Srivastava, and V. Prakash, "Advanced hybrid prediction model: optimizing lightgbm, xgboost, lasso regression, and random forest with bayesian optimization," vol. 102, no. 9, 2024.

[2] N. Duan, "Smearing estimate: A nonparametric retransformation method," vol. 78, no. 383, 1983.