Online News Popularity

Rajat Chaudhary[a,*], **Yuvaraj G**[b,**], **Srinivas Shavukapu Kattegummula**[c,***] **and Sonu Goyal**[d,****]

Codebase is available at this GitHub repository

Abstract. The rapid growth of digital media has made predicting online news popularity a critical challenge for media companies and content creators. Popularity is influenced by diverse factors such as article keywords, sentiment, publication timing and social engagement signals, making manual estimation highly unreliable. Accurate prediction of article virality is essential for optimizing content strategies, improving audience engagement and forecasting advertising revenue. It also enables organizations to allocate resources efficiently by focusing on high impact content and reducing effort on low performing articles.
This project aims to develop a machine learning model capable of predicting article popularity measured by shares or classified as viral/non-viral and to identify the key drivers behind virality.

1 Problem Definition

- **Objective :** Enable media companies and content creators to predict the popularity of online news articles
- **How Will the Solution Be Used ?** The solution will be integrated into a real time prediction pipeline, allowing editors and content strategists to assess the potential virality of articles before or shortly after publication
- **Current Solutions/Workarounds :** Currently, popularity estimation relies on manual judgment, trending topic monitoring, and historical performance analysis
- **Problem Framing :** This is a supervised learning problem where the target variable is either the number of shares (regression) or a multiclass classification classification (Trending, Viral, Performer and Niche).
- **Performance Measurement :** RMSE, MAE, MSE on predicted shares for regression. Accuracy, Precison and F1-score for classification
- **Comparable Problems :** Similar approaches are used in social media trend prediction, ad click through rate forecasting etc
- **Assumptions :** Historical data on article features and shares is available and representative.

2 Data Collection and Preparation

- **Data Source :** The dataset is obtained from the UCI Machine Learning Repository (Online News Popularity) . It contains articles published on Mashable over a specific time period, along with metadata, content based features and social engagement metrics.

* Corresponding Author. Email: rajat.iisc.ac.in
** Corresponding Author. Email: yuvarajgopi@iisc.ac.in
*** Corresponding Author. Email: srinivassk@iisc.ac.in
**** Corresponding Author. Email: sonugoyal@iisc.ac.in

Table 1. Description of Data

Attribute	Details
Size	Approximately 39,000 observations
Features	58
Feature Type	Integer and Real
Dataset Characteristics	Multivariate
Format	Tabular data

- **Data Preprocessing :**
  - Divided data into **training (80%) and test (20%) set**
  - **Removed leading/trailing whitespaces** in column names for consistency.
  - **Dropped irrelevant columns (url, shares, timedelta)** after transformation.
  - Applied logarithmic transformation to the shares column to reduce skewness and stabilize variance.
  - Separated numerical and categorical features
  - Converted categorical variables into binary format using One-Hot Encoding for model compatibility.
  - Standardized numerical features to ensure uniform scale across features.

3 Exploratory Data Analysis (EDA)

- **Target variable (Shares)** : **Shares column is highly skewed** with a long tail, indicating that a small number of articles receive very high shares.
- **Quality : No missing values** were found in the dataset.
- **Analysis :**
  - **Univariate analysis** revealed that most numerical features had distributions concentrated at low values while categorical features showed article spread across channels and weekdays.
  - **Bivariate analysis** indicated weak linear relationships between individual features and popularity with box plots suggesting slight influence from data channel and weekend status.
  - **Multivariate analysis** using correlation matrices and pair plots confirmed generally weak correlations. These findings suggest that news popularity is driven by complex, non-linear interactions rather than single features
- **Outlier** :
  - Rate columns (n_unique, n_non_stop*) contained values > 1, which is invalid since these represent ratios. These were corrected by replacing **outliers with the mean**.

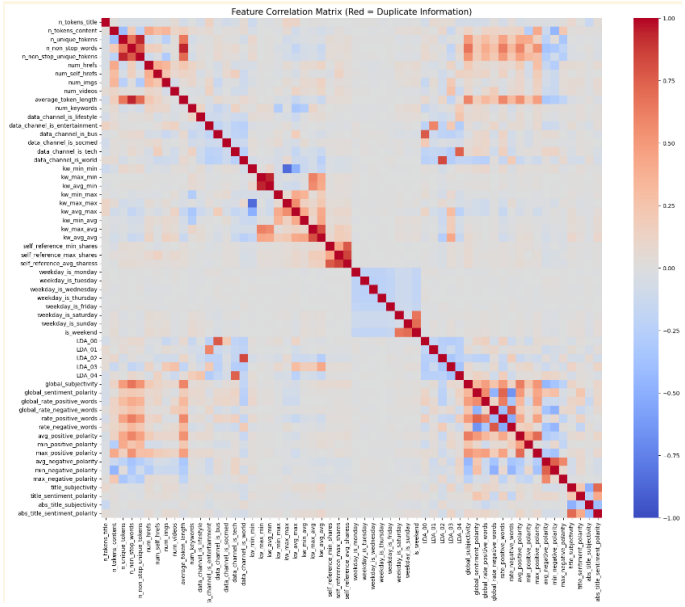

Figure 1. Feature Correlation Matrix (Red = Duplicate Information)

- Keyword share columns (kw_*) had negative values, which were replaced with the **mean values**
- Sentiment polarity values were within the expected range (-1 to +1), requiring no cleaning.

- **Insights :**
  - News popularity is influenced by multiple factors rather than any single feature.
  - **Linear models may not capture these relationships effectively**, non-linear models and feature interactions should be explored.
  - Features like data channel, weekend status and keyword-based metrics may play a significant role in prediction.

4 Model Selection and Evaluation

- **Regression**
  Problem was framed as a regression task to predict the logarithmic transformation of article shares (shares_log). Multiple models were evaluated to identify the best-performing algorithm:
  - Random Forest Regressor
  - Gradient Boosting Regressor
  - Lasso Regression
  - Support Vector Regressor (SVR)
  - K-Nearest Neighbors Regressor (KNN)
  - Multi-layer Perceptron Regressor (MLP)

Table 2. Model Performance (on log scale)

Model	MAE	MSE	RMSE
Random Forest Regressor	0.6839	0.8190	0.9050
Gradient Boosting Regressor	**0.6529**	**0.7474**	**0.8645**
Lasso Regression	0.7077	0.8573	0.9259

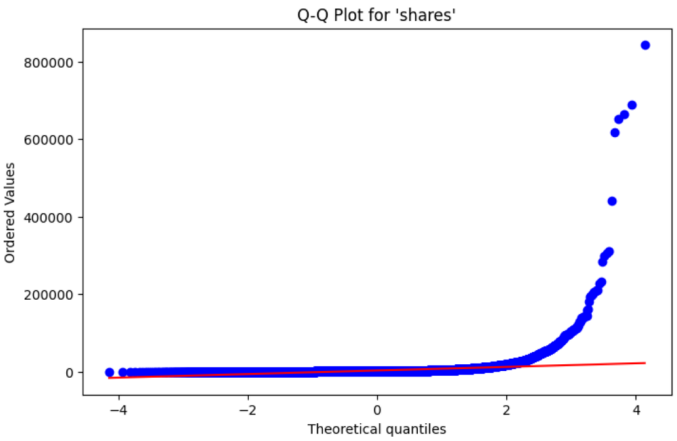

Figure 2. Q-Q plot for Shares (target variable)

Table 3. Other Models

Model	CV Score (Neg MSE)	MAE (Original Scale)
SVR	-0.8138	2333.99
KNN	-0.8525	2439.32
MLP	-0.7950	2362.38

Gradient Boosting Regressor outperformed other models and was selected for hyperparameter tuning.

- **Regression vs. Classification**
  While regression models were explored to predict the exact number of shares, the results showed relatively high error margins (MAE 2,300 shares). This is expected because news popularity is inherently complex and influenced by factors beyond numeric share counts. **In real world scenarios, what matters most is whether an article becomes viral rather than its precise share count**. For example, in some regions, an article with relatively low shares may still be considered viral due to cultural or platform specific dynamics.

  Therefore, we shifted our focus from regression to a classification setting, framing the problem as predicting whether an article will be viral or not viral. This approach aligns better with business objectives

- **Classification**
  - To align with decisions that care about virality rather than exact share counts, we framed popularity as a **multi-class classification** problem using dynamic thresholds computed from the training set only

Table 4. Popularity Levels Defined by Training Set Percentiles

Class	Shares Range (based on training percentiles)
Niche	$0 \leq S < P_{50}$
Performer	$P_{50} \leq S < P_{95}$
Trending	$P_{95} \leq S < P_{99}$
Viral	$S \geq P_{99}$

Table 5. Classifier Performance Summary (All Models)

Classifier	Accuracy	Category	Total Actual	Correct Hits	Hit Rate (%)	Precision	F1-Score
XGBoost (Dynamic Tiers)	**58.57%**	Niche (< 50%)	4075	2633	64.61%	67.63%	0.6609
		Performer (50–95%)	3466	1945	56.12%	58.13%	0.5711
		Trending (95–99%)	318	62	19.50%	9.75%	0.1300
		Viral (> 99%)	70	4	5.71%	7.41%	0.0645
LightGBM (Balanced)	**62.27%**	Niche (< 50%)	4075	2798	68.66%	65.88%	0.6724
		Performer (50–95%)	3466	2133	61.54%	58.57%	0.6002
		Trending (95–99%)	318	6	1.89%	15.38%	0.0336
		Viral (> 99%)	70	0	0.00%	0.00%	0.0000
Random Forest (Balanced)	**63.12%**	Niche (< 50%)	4075	2739	67.21%	68.03%	0.6762
		Performer (50–95%)	3466	2264	65.32%	58.14%	0.6152
		Trending (95–99%)	318	2	0.63%	28.57%	0.0123
		Viral (> 99%)	70	0	0.00%	0.00%	0.0000
AdaBoost (Balanced Tree)	**42.92%**	Niche (< 50%)	4075	2353	57.74%	64.75%	0.6105
		Performer (50–95%)	3466	918	26.49%	55.43%	0.3585
		Trending (95–99%)	318	94	29.56%	6.47%	0.1062
		Viral (> 99%)	70	38	54.29%	3.20%	0.0605
Naive Bayes (Transformed)	**57.76%**	Niche (< 50%)	4075	3327	81.64%	58.66%	0.6827
		Performer (50–95%)	3466	1223	35.29%	61.71%	0.4490
		Trending (95–99%)	318	25	7.86%	11.79%	0.0943
		Viral (> 99%)	70	5	7.14%	7.94%	0.0752

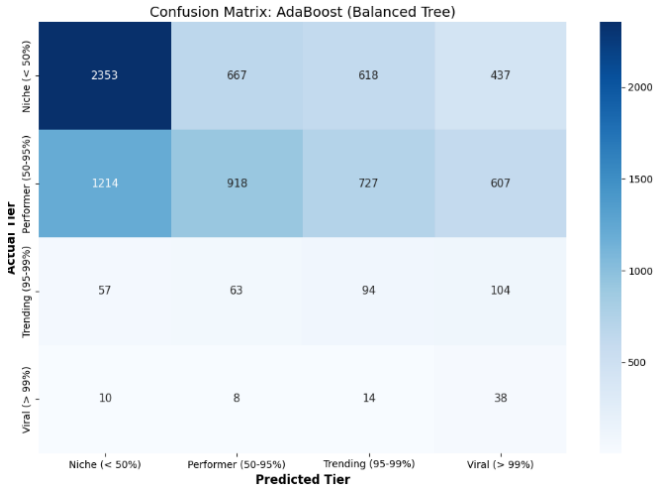

Figure 3. AdaBoost (Balanced Tree)

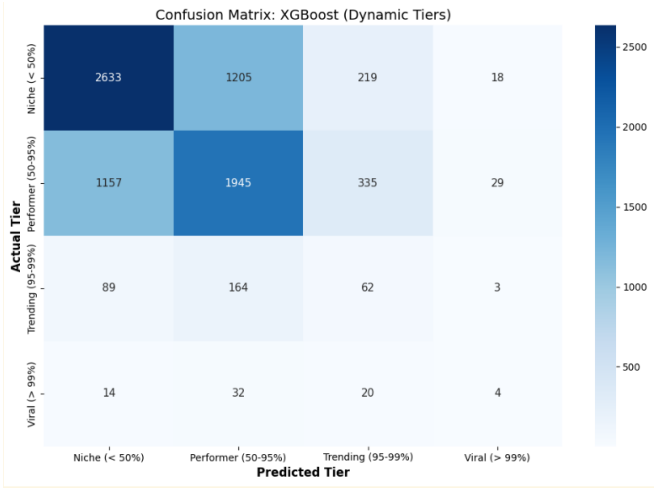

Figure 4. XGBoost (Dynamic Tiers)

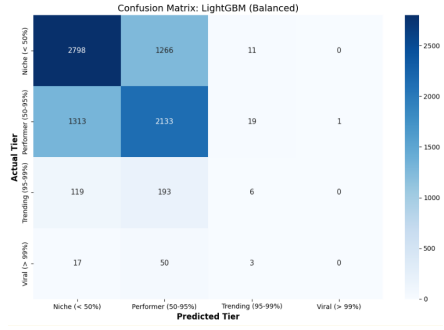

Figure 5. LightGBM (Balanced)

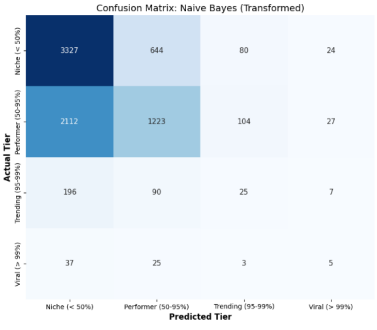

Figure 6. Naive Bayes (Transformed)

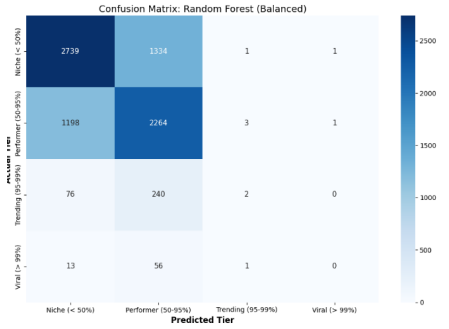

Figure 7. Random Forest (Balanced)

- Before training, we computed popularity tiers using training set percentiles **(median, top 5%, top 1%)** and applied these thresholds to both train and test sets.
- LightGBM and Random Forest slightly improved overall accuracy **62–63%** but did not significantly improve minority class detection.
- All models performed reasonably well **F1-Score 0.57–0.68** for majority classes
- **Severe imbalance led to poor F1-scores** for Trending and Viral across most models.
- AdaBoost showed higher recall for **Viral 54%** but extremely low **precision 3%**, indicating many false positives.
- Naive Bayes achieved the highest recall for Niche **81.64%** but struggled with minorities.

- Trending (top 5%) and Viral (top 1%) remain challenging, with F1-scores often below **0.10**
- **Hyperparameter Tuning with Optuna :** We optimized classifier performance using Optuna. Optuna employs Bayesian optimization to efficiently search the parameter space
- **Macro Level Metrics** Since our goal is to predict viral popularity, overall accuracy is misleading due to extreme class imbalance. We focused on macro-averaged metrics, which treat all classes equally

$$\text{Macro Precision} = \frac{1}{K}\sum_{i=1}^{K}\text{Precision}_i$$

$$\text{Macro Recall} = \frac{1}{K}\sum_{i=1}^{K}\text{Recall}_i$$

$$\text{Macro F1} = \frac{1}{K}\sum_{i=1}^{K}\text{F1}_i$$

This approach ensures Trending and Viral classes influence the score as much as Niche and Performer.

- **Class Weights** Class imbalance was addressed by assigning higher weights to minority classes

$$w_i = \frac{N}{K \cdot n_i}$$

- XGBoost achieved Weighted AUC = **0.7010**, Macro AUC = **0.7025** and Accuracy = **58.57%**

Table 6. Best Accuracy

Model	Accuracy	Observations
Random Forest	63.12%	Highest overall accuracy
LightGBM	62.27%	-
XGBoost	58.57%	AUC: Macro **0.7025** Weighted **0.7010**

5 Conclusion

Predicting online news popularity is a complex task influenced by multiple factors such as content attributes, timing and social signals. To address the skewed distribution of shares, we reframed the problem as a multi class classification using dynamic percentile thresholds, enabling better differentiation between Niche, Performer, Trending and Viral tiers. Among the models evaluated, XGBoost emerged as the most balanced performer, achieving Macro **AUC = 0.7025, Weighted AUC = 0.7010**, and an overall accuracy of **58.57%**. While Random Forest achieved the highest accuracy **63.12%**, **its performance on macro level metrics was less robust compared to XGBoost**, which is critical for handling class imbalance. LightGBM also performed competitively with 62.27% accuracy

XGBoost provided superior macro-level scoring, making it the final model of choice for this task

6 Limitation

- The dataset is limited to **Mashable articles from a specific time frame**, which may not generalize well to other platforms or current trends.
- Features are primarily metadata and basic content attributes. **Deeper semantic features (e.g. sentiment analysis, topic modeling)** were not fully explored.
- **Popularity patterns change rapidly** due to evolving user behavior and external events, which static models cannot fully capture.

7 Future scope of work

- Use advanced text embeddings (e.g, BERT, GPT-based models) for richer semantic understanding of article content
- Apply time-series or sequence models (e.g LSTM, Transformer-based) to capture evolving popularity trends.

8 Contribution

- **Rajat Chaudhary (rajatc@iisc.ac.in)** I contributed by :
  - Identifying the problem and dataset
  - Prepared a comprehensive report documenting key observations, methodologies and insights gained throughout the project
  - Performing Exploratory Data Analysis to uncover patterns and distribution insights
  - Training both regression and classification models and finally evaluating them
  - Project organization and Git cleanup etc.
- **Yuvaraj G (Yuvarajgopi@iisc.ac.in)** Architect the problem design ,Model selection, training the model with proper hyperparameters, Created a structured code ,Report writing
- **Srinivas Shavukapu Kattegummula (srinivassk@iisc.ac.in)** Data preprocessing , exploratory Data Analysis, feature engineering and model development.
- **Sonu Goyal (sonugoyal@iisc.ac.in)**
  - Defining the business problem and understanding its objectives
  - A comprehensive EDA was conducted to uncover patterns, trends, and relationships within the data.
  - Based on the insights from EDA, appropriate machine learning algorithms were selected and implemented.
  - A structured presentation was created to communicate the entire workflow and findings.

9 Citation

The dataset was sourced from the UCI Machine Learning Repository

10 References

A Proactive Intelligent Decision Support System for Predicting the Popularity of Online News