1 Statement

1.1 Dataset Selection

The Online News Popularity dataset (Fernandes et al., 2015a) was chosen due to its relevance in understanding modern digital content consumption. Sourced from the UCI Machine Learning Repository, this real-world dataset contains 39,644 articles. It also provides a rich feature set of 58 predictive attributes, ranging from basic metadata (e.g., word count, publication day) to complex linguistic features (e.g., sentiment polarity, subjectivity). This complexity makes it an ideal candidate for applying and comparing different classical machine learning techniques. Furthermore, as will be demonstrated in the EDA, it presents realistic data challenges, such as multicollinearity and features on disparate scales, which must be addressed in preprocessing.

1.2 Purpose and Goal

The primary purpose of this analysis is to develop and evaluate a binary classification model capable of accurately predicting whether an online news article will become "popular" or "unpopular." Following the methodology of the dataset's original authors (Fernandes et al., 2015b), popularity is defined using a fixed share threshold:

- "Unpopular" (Class 0): An article that received ≤ 1,400 shares.
- "Popular" (Class 1): An article that received > 1,400 shares.

This threshold of 1,400 shares was adopted as it almost represented the median value in the original study. Our own analysis of the dataset confirms this is an effective threshold, as it yields a well-balanced class distribution (20,082 "unpopular" instances vs. 19,562 "popular" instances), which is ideal for training robust classification models.

There are several specific goals: 1) To perform an Exploratory Data Analysis (EDA) to identify and plan for data quality issues, including target variable skew, feature scaling, and multicollinearity; 2) To implement and train four classical machine learning models: Logistic Regression, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Random Forest; 3) To evaluate and compare the performance of these models using appropriate classification metrics, including Accuracy, Precision, Recall, and F1-Score; 4) To interpret the results and identify which article characteristics are the most significant predictors of popularity.

2 Exploratory Data Analysis (EDA)

Before applying machine learning models, an Exploratory Data Analysis (EDA) was conducted. The purpose of this EDA was threefold: 1) To analyze the distribution of the target variable shares, 2) To investigate the features for issues requiring preprocessing, and 3) To identify preliminary relationships between features and the target.

2.1 Target Variable Analysis

The original target variable, shares, is a continuous integer representing the number of shares an article received.

The distribution of the shares variable, as Figure 1 demonstrates, shows extreme right-skew. This renders the "shares" variable problematic for standard regression models, as they would be biased by the extreme outliers. This observation strongly supports the decision of a binary classification task. By using the 1,400-share threshold, we create a well-balanced target variable, is_popular, which is suitable for the intended classification models.

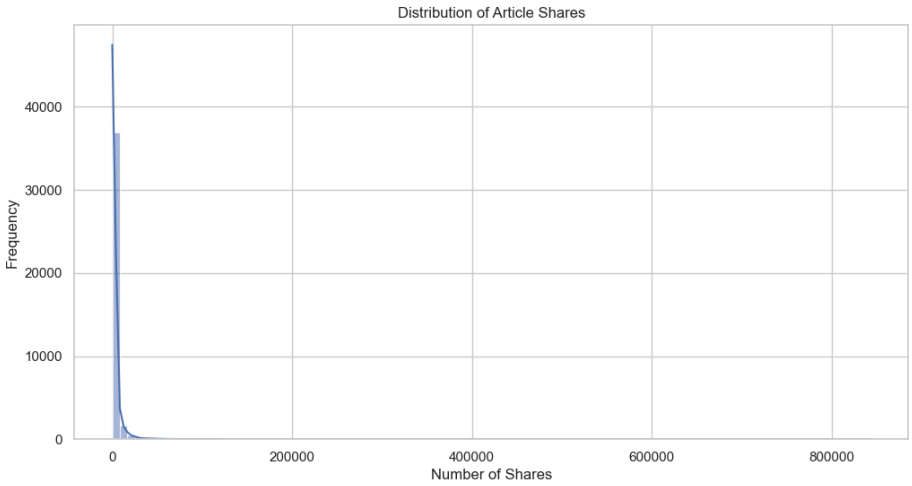

Figure.1 *Distribution of Article Shares*

2.2 Feature Analysis: Scaling and Multicollinearity

The 58 predictive features were analyzed for two key properties that influence model performance: feature scaling and multicollinearity.

2.2.1 Feature Scaling

The features, as Table 1 demonstrates, exist on wildly different scales. n_tokens_content has a range of over 8,000, while title_subjectivity has a range of 1. This is a critical issue for distance-based and regularized models. Therefore, feature scaling (e.g., StandardScaler) is a mandatory preprocessing step for Logistic Regression, SVM, and KNN.

Statistic	n_tokens_content	title_subjectivity	global_sentiment_polarity
Mean	546.515	0.282	0.119
Std. Dev.	471.108	0.324	0.097

Table.1 *Scale Comparison for Sample Features*

2.2.2 Multicollinearity

A correlation heatmap was generated for a subset of 12 features to investigate collinearity.

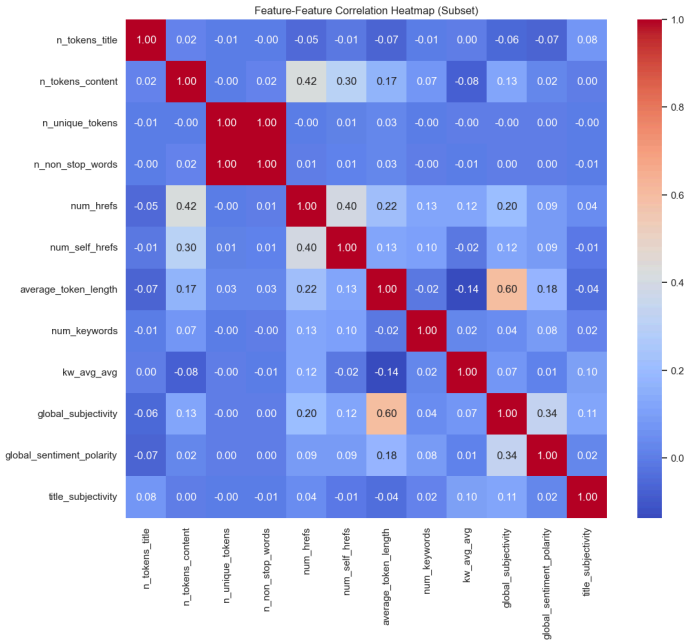

Figure.2: *Correlation heatmap for a subset of 12 features.*

The heatmap (Figure 2) revealed a perfect 1.00 correlation between the features n_unique_tokens and n_non_stop_words. This perfect multicollinearity indicates that these features are redundant. Including both would violate the assumptions of linear models and can lead to unstable coefficients. To resolve this, n_non_stop_words was slated for removal from the feature set.

2.3 Preliminary Feature-Target Relationship Analysis

To gain an initial understanding of which features might be predictive, a correlation analysis was run between all features and the log_shares variable.

Top 3 Positive Correlations		Top 3 Negative Correlations	
Features	Correlation	Features	Correlation
kw_avg_avg	0.222	LDA_02	-0.165
LDA_03	0.126	data_channel_is_world	-0.152
is_weekend	0.114	data_channel_is_entertainment	-0.083

Table.2 *Top 5 Positive and Negative Correlates with Popularity*

The results show that the strongest predictors are related to the article's content and topic. kw_avg_avg (the average popularity of an article's keywords) is the strongest positive predictor.

Conversely, articles in certain topic channels (LDA_02, data_channel_is_world) are strongly associated with less popularity.

3. Modeling

Based on the findings from the Exploratory Data Analysis, the dataset was prepared for modeling. First, the n_non_stop_words feature was dropped to resolve the issue of perfect multicollinearity. The complete dataset was then partitioned into a training set, containing 80% of the instances (31,715 articles), and a test set, containing the remaining 20% (7,929 articles). To address the variance in feature scales and prevent data leakage, a sklearn.pipeline.Pipeline was constructed for each model, which automatically applied StandardScaler to the training data before fitting and to the test data before prediction.

3.1 Modeling Selections

Four classical classification models were selected for training and evaluation. These included Logistic Regression, a linear model configured with max_iter=1000 to ensure solver convergence; K-Nearest Neighbors (KNN), a non-linear, instance-based model using the default of 5 neighbors; Random Forest, a non-linear ensemble model using the default of 100 trees and a random_state for reproducibility; and a Support Vector Machine (SVM), a non-linear, kernel-based model using the default 'rbf' (Radial Basis Function) kernel.

3.2 Model Evaluation

To ensure a comprehensive comparison, a suite of standard classification metrics was used:

- Accuracy: The percentage of total correct predictions. This is a reliable primary metric given the well-balanced nature of the dataset.
- Precision: The ability of the model to avoid false positives (e.g., "Of all articles predicted 'popular', how many actually were?").
- Recall: The ability of the model to find all positive instances (e.g., "Of all *actual* 'popular' articles, how many did the model find?").
- F1-Score: The harmonic mean of Precision and Recall, providing a single score that balances both concerns.

4 Discussion & Interpretation

4.1 Model Comparison & Evaluation

The empirical results in Table 3 reveal a consistent performance ranking across all evaluated models. The Random Forest classifier delivers the strongest outcomes, achieving the highest Accuracy (0.6624), Precision (0.6620), Recall (0.6621), and F1-Score (0.6621). This superior performance reflects the model's ability to capture non-linear relationships and interactions across features, an advantage that simpler linear models such as Logistic Regression cannot

fully exploit. Importantly, Random Forest also maintains a reasonable computational cost, completing training and inference in 1.89 seconds, which is efficient for an ensemble method.

In comparison, the Support Vector Machine (SVM) demonstrates moderate predictive performance (≈0.654 across all metrics) but incurs a substantial computational burden, requiring 35.95 seconds—roughly two orders of magnitude slower than the other models. While SVM offers competitive accuracy, its runtime cost makes it less practical for large-scale or time-sensitive applications.

Logistic Regression performs slightly below SVM (≈0.651 across metrics) but compensates with extremely low execution time (0.13 seconds), making it an efficient baseline model. Meanwhile, K-Nearest Neighbors (KNN) is the weakest performer, producing the lowest classification metrics (≈0.607) despite its modest runtime (0.23 seconds).

A notable pattern across all models is the near-identical values of Accuracy, Precision, Recall, and F1-Score. This convergence indicates a well-balanced dataset with no significant class imbalance. Under these conditions, Accuracy serves as a reliable summary metric, and the closely aligned F1-Scores confirm that none of the models exhibit substantial trade-offs between Precision and Recall.

Model	Accuracy	Precision	Recall	F1-Scores	Times(s)
Random Forest	0.6624	0.6620	0.6621	0.6621	1.89
Support Vector Machine (SVM)	0.6542	0.6541	0.6542	0.6541	35.95
Logistic Regression	0.6515	0.6514	0.6515	0.6512	0.13
K-Nearest Neighbors (KNN)	0.6071	0.6069	0.6071	0.6063	0.23

Table.3 *Model Performance Comparison*

4.2 Interpretation of Key Findings

To understand why the Random Forest model was successful, an analysis of its feature importances was conducted.

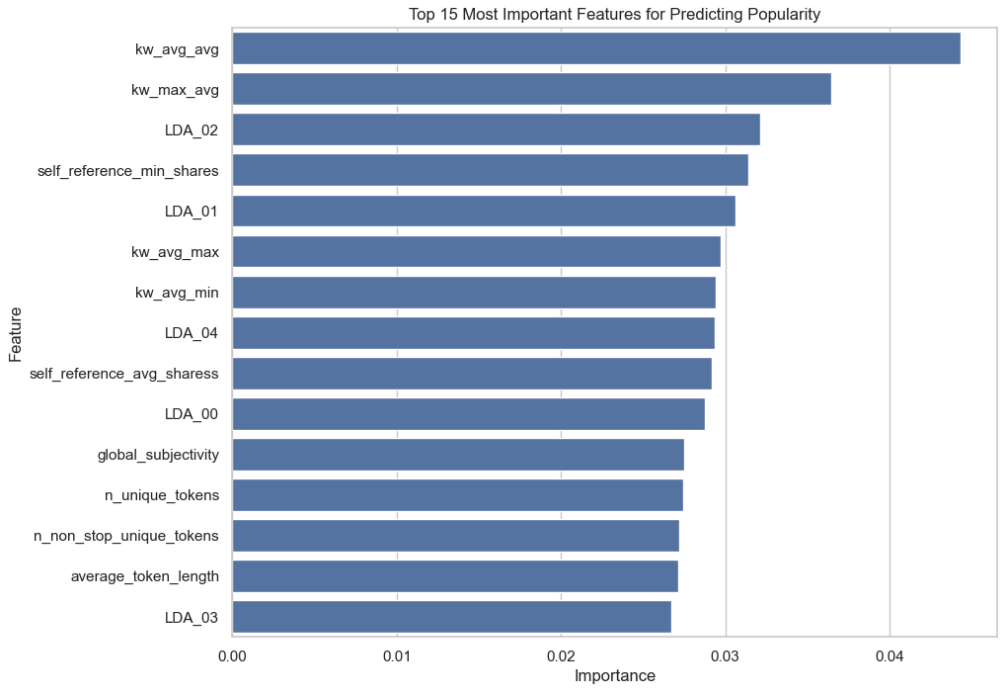

Figure 3: *Feature importances from the Random Forest model, showing the top 15 predictors.*

The model's decisions are overwhelmingly driven by what an article is about. The kw_avg_avg (average popularity of the article's keywords) was the most important feature. This indicates a strong "rich get richer" dynamic where content about already-popular topics is more likely to be shared. This is further supported by five of the top ten predictors being LDA_ topic features (e.g., LDA_02, LDA_01, LDA_04), confirming that an article's topic is a powerful heuristic for its shareability.

The model also learned that an article's context within its own publication matters. Features like self_reference_min_shares and self_reference_avg_sharess were highly ranked, suggesting that linking to other popular (or avoiding unpopular) internal content is a key predictive signal.

Notably, features related to the writing style itself (e.g., global_subjectivity, average_token_length) were of secondary importance. While predictive, they ranked clearly below the semantic and contextual features, suggesting the *substance* of the content is a stronger driver than its *style*.

References

Fernandes, K., Vinagre, P., Cortez, P., & Sernadela, P. (2015a). Online News Popularity [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C5NS3V

Fernandes, K., Vinagre, P., & Cortez, P. (2015b). A proactive intelligent decision support system for predicting the popularity of online news. In F. Pereira, P. Machado, E. Costa, & A. Cardoso (Eds.), *Progress in artificial intelligence: EPIA 2015* (Lecture Notes in Computer Science, Vol. 9273, pp. 535–546). Springer. https://doi.org/10.1007/978-3-319-23485-4_53