Analysis on Online News Popularity Dataset

Alaz Yilmaz - 14848171
Emre Genc - 15897281
Ekin Dayi - 14940692
Economics and Business School, University of Amsterdam, Amsterdam, Netherlands

***Abstract*—The popularity of online news items depends on several factors, including content, publication date etc. This research makes use of the Online News Popularity dataset, which was created by Fernandes et al. (2015) and comprises 39,644 articles with characteristics: words, links, digital media, publication time, keywords, NLP(Natural Language Processing). Classifying articles according to popularity, identifying important determinants affecting article performance, and revealing patterns through clustering are among the main goals.**

I. Research Aim

The objective of this project is to utilise the Online News Popularity dataset [1] to examine the factors that contribute to the popularity of online news stories. The methodology employed combines supervised and unsupervised machine learning techniques to categorise articles based on their popularity and to identify patterns in factors such as sentiment, word usage and temporal characteristics. Furthermore, the study investigates the significance of these characteristics and seeks to enhance model performance, thereby offering practical insights.

II. Introduction

Nowadays, online news have become crucial in distributing information across the globe. The popularity of articles is determined by a variety of factors. This paper examines the elements featured in the Online News Popularity Dataset that influence the popularity of online news articles. By analysing the elements in the dataset, the research hopes to find patterns and give practical insights for optimising content in the competitive online news scene.

The process starts with exploratory data analysis (EDA), which examines the question of whether feature selection should be applied followed by outlier detection, train/test data split, data scaling for feature standardization, and data visualization to spot trends.

For the main model implementation and analysis part, first we focused on classification; a range of supervised learning models are implemented: XGBoost, Logistic Regression, Support Vector Machines (SVM), and Naive Bayes Classifier to predict article popularity. Then, we performed unsupervised learning techniques such as K-means clustering, Gaussian Mixture Models (GMM) and Agglomerative Clustering to discover underlying structures and patterns in the data. As the final stage, in order to optimise overall performance, ensemble techniques are employed, whereby multiple models are combined in order to capitalise on their respective strengths and enhance the predictive accuracy of the resulting model.

III. Data Preprocessing

The Online News Popularity Dataset contains 39,644 articles, each with 61 variables including url, content elements (e.g., word count, title length), metadata (e.g., publication day, content channel), and the target variable shares. Preprocessing consisted of multiple processes, but we will start with loading in the dataset and analysing the data. We found no NA values in the dataset and decided to drop the url column as it was not needed.

Feature
Words
Number of words in the title
Number of words in the article
Average word length
Rate of non-stop words
Rate of unique words
Rate of unique non-stop words
Links
Number of links
Number of Mashable article links
Minimum, average and maximum number of shares of Mashable links
Digital Media
Number of images
Number of videos
Time
Day of the week
Published on a weekend?

Feature
Keywords
Number of keywords
Worst keyword (min/avg/max shares)
Average keyword (min/avg/max shares)
Best keyword (min/avg/max shares)
Article category (Mashable data channel)
Natural Language Processing
Closeness to top 5 LDA topics
Title subjectivity
Article text subjectivity score and its absolute difference to 0.5
Title sentiment polarity
Rate of positive and negative words
Positive words rate, non-neutral words
Negative words rate, non-neutral words
Polarity of positive words (min/avg/max)
Polarity of negative words (min/avg/max)
Article text polarity score and its absolute difference to 0.5

TABLE I: Grouped features by category

The table presents a thorough summary of the various features that were used in the analysis, grouped into discrete groups. In order to investigate patterns and relationships in the data, these features were used as part of an unsupervised learning strategy.

By looking at the data types, we saw that all features were numerical. Looking at the distribution of shares, it has a very skewed distribution, necessitating careful research.

We decided to use 1400 shares as a threshold value to determine whether the article is popular or not. This value is given in the documents in the dataset and is also the median value of the shares in the dataset. Articles above or equal to 1400 shares will be considered popular, and articles below 1400 shares will be considered unpopular.

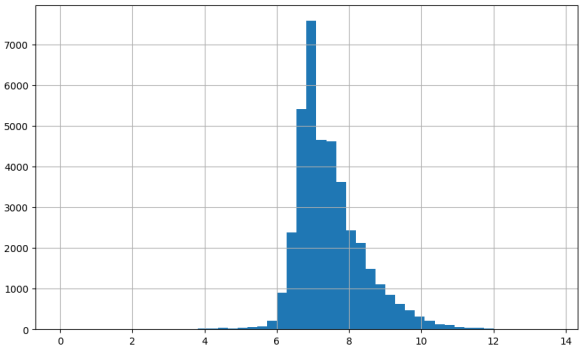

Fig. 1: Histogram of Shares.

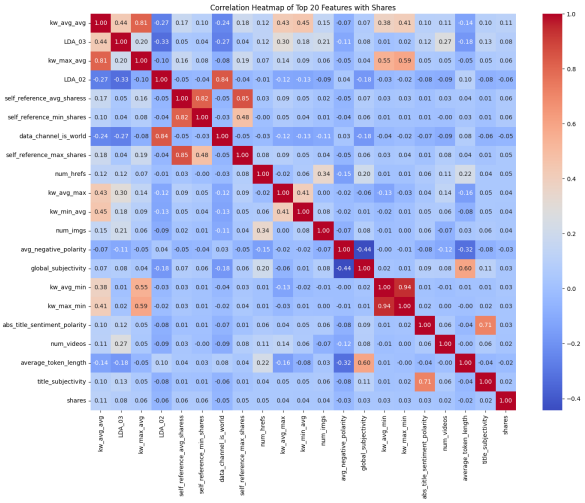

Fig. 2: Correlation Heatmap of Top 20 Features with Shares

A. *Correlation Heatmap Analysis*

Figure 2 displays the correlation heatmap for the top 20 features with the highest correlation to the target variable, `shares`. The heatmap provides insights into the relationships between these features and highlights patterns that can guide feature selection and interpretation. Below are the key observations:

- The correlations between the features and the target variable are generally low to moderate, with the highest correlation around 0.45. This indicates that no single feature has a very strong linear relationship with the target variable.
- Most feature pairs exhibit low covariances, as evidenced by the dominance of blue and light-colored cells in the heatmap. This suggests that the features are largely independent and capture distinct aspects of the data.
- Features related to keyword metrics show moderate positive correlations among themselves, reflecting their shared relevance to keyword importance.
- Self-reference metrics also demonstrate moderate positive correlations with each other, as expected given their conceptual similarity.
- Sentiment polarity and subjectivity features show weak correlations with most other features and with the target variable. This suggests that these features contribute unique, non-redundant information.
- Temporal features exhibit very weak correlations with the target variable and other features, indicating limited relevance in predicting shares.
- The overall structure of the heatmap suggests that the features capture diverse aspects of the articles, with minimal redundancy. This reinforces the value of using a wide range of features in predictive models.

B. *Feature Distributions Analysis*

Figure 3 illustrates the distributions of the top 20 features with the highest correlation with the target variable, `shares`. These features represent various aspects of the articles, such as keyword metrics, self-references, subjectivity, sentiment polarity, and multimedia content. To summarize:

- Keyword metrics: These features exhibit heavily right-skewed distributions, indicating that most articles have relatively low values for these metrics, with a few having significantly higher keyword relevance.
- Self-reference metrics: These distributions are also highly skewed, with the majority of articles having low self-referenced shares and only a small number of articles receiving a high number of shares through self-referencing.
- Latent Dirichlet Allocation (LDA) features: The distributions are concentrated around smaller values, reflecting the probability distributions over topics for articles.
- Subjectivity features: These features exhibit bell-shaped distributions, with most articles falling within moderate levels of subjectivity, indicating a balanced mix of objective and subjective content.
- Sentiment polarity features: These features are concentrated near zero, indicating that most articles have a neutral sentiment, with fewer articles exhibiting strongly positive or negative sentiments.
- Multimedia content features: The distributions are right-skewed, showing that most articles contain minimal multimedia content, with only a small proportion including higher numbers of images or videos.
- Token and content length metrics: The distributions indicate that most articles have a moderate token length and hyperlink count, with a few articles being outliers with significantly higher values.

Overall, the features with the strongest correlation to the target variable `shares` display diverse distribution

patterns, ranging from right-skewed to approximately normal, reflecting the varied characteristics of the articles in the dataset.

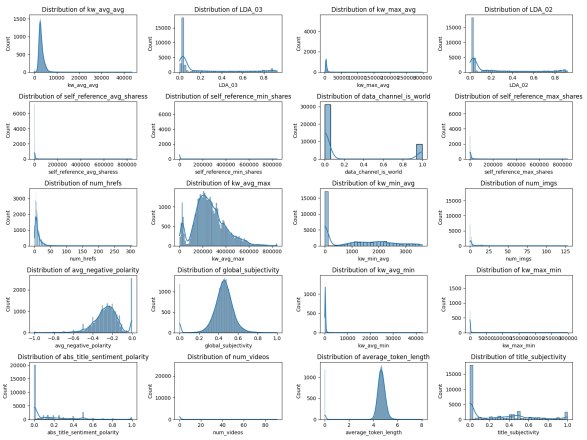

Fig. 3: Distributions of the Top 20 Features with the highest correlation to Shares

C. Feature Selection

By minimizing overfitting and lowering computational cost, feature selection is a step to boosting model performance. The KBest algorithm, a statistical technique for choosing the best features according to their significance to the target variable, was utilised for this investigation. For feature selection we used KBest algorithm to select the best 20 features according to their F scores (ANOVA F-Test) which is calculated by:

$$F = \frac{\text{variance between classes}}{\text{variance within classes}} \tag{1}$$

This feature selection technique is applied on Logistic Regression and XGBoost models and following results are obtained:

Model	Class	Precision	Recall	F1-Score	Support
Logistic Regression	0	0.64	0.60	0.62	3604
	1	0.66	0.69	0.67	3929
	Accuracy		**0.65**		7533
XGBoost	0	0.66	0.60	0.63	3604
	1	0.66	0.71	0.68	3929
	Accuracy		**0.66**		7533

TABLE II: Logistic Regression and XGBoost Scores with feature selection

In order to examine how the feature selection performed, same models are applied without doing the feature selection and the corresponding scores are recorded.

Model	Class	Precision	Recall	F1-Score	Support
Logistic Regression	0	0.65	0.62	0.64	3604
	1	0.67	0.70	0.68	3929
	Accuracy		**0.66**		7533
XGBoost	0	0.69	0.64	0.66	3604
	1	0.69	0.73	0.71	3929
	Accuracy		**0.69**		7533

TABLE III: Logistic Regression and XGBoost Scores without feature selection

As can be seen, the scores of the models built without feature selection were partially higher than the models with feature selection, so the rest of the research was conducted without feature selection.

D. Outlier Detection and Elimination

Outliers can have a major impact on model performance, particularly when dealing with skewed data. We used the Isolation Forest method, which is a tree-based unsupervised learning technique. This approach finds outliers based on their relative isolation in the feature space, making it suitable for processing high-dimensional data.

The Isolation Forest algorithm calculates an anomaly score for each data point based on the number of splits required to isolate it in a random binary tree. Outliers, which are more isolated than normal data points, tend to have shorter path lengths and higher anomaly scores. The anomaly score for a data point x is defined as:

$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}} \tag{2}$$

where:

- $s(x, n)$: Anomaly score for a data point x, where n is the number of samples.
- $E(h(x))$: The expected path length of x in a tree.
- $c(n)$: The average path length of a random binary tree, approximated as:

$$c(n) = 2H(n-1) - \frac{2(n-1)}{n} \tag{3}$$

where $H(i)$ is the i-th harmonic number.

Therefore, data points with higher anomaly scores are classified as outliers.

The Isolation Forest algorithm was applied to our dataset with a contamination rate of 0.05, meaning we assumed 5% of the data points to be outliers. Outliers were removed by identifying data points classified as outliers and retaining only the remaining data points. This preprocessing step improved the robustness of our predictive models by eliminating noise and reducing the influence of anomalous data points.

E. Train/Test Split

To accurately evaluate model performance, the dataset was separated into training and testing subsets with a 80-20 ratio. The training set was used to create and refine models, while the test set was used to assess their generalization to new data and to calculate metrics.

F. Data Standardization

Since variables on bigger scales can dominate models and distort findings, feature scaling is essential when working with numerical qualities of different magnitudes. Therefore, we used standard scaling to standardize all features. All features now have a standard deviation of one and a mean of zero. Standardization enhances model performance and convergence rates.

The formula for standard scaling is as follows:

$$x' = \frac{x - \mu}{\sigma} \tag{4}$$

where:

- x': The scaled value of the feature.
- x: The original value of the feature.
- μ: The mean of the feature across the training dataset.
- σ: The standard deviation of the feature across the training dataset.

IV. Supervised Models

A. Logistic Regression

Logistic Regression was employed to predict whether an article would be popular, leveraging its capability to model the probability of a binary outcome based on input features. The model optimizes the logistic function:

$$P(y = 1 \mid X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n)}} \tag{5}$$

where:

- $P(y = 1 \mid X)$: The probability of the positive class given the input features X.
- β_0: The intercept term.
- β_i: The coefficients corresponding to each input feature X_i, which determine their contribution to the predicted probability.
- X_i: The i-th feature of the input data.
- n: The total number of features.

To enhance performance, hyperparameter tuning was conducted using Grid Search with 5-fold cross-validation, optimizing the regularization strength (C), regularization type (ℓ_1 or ℓ_2), and solver method (`liblinear`). The best configuration was selected based on cross-validated accuracy.

B. XGBoost

XGBoost (Extreme Gradient Boosting) was utilized as a classification model to predict article popularity due to its efficiency and performance in handling structured data. The model optimizes the objective function by minimizing the following loss:

$$\mathcal{L}(\Theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{6}$$

where:

- $l(y_i, \hat{y}_i)$: The loss function, which measures the difference between the true label y_i and the predicted label $\hat{y}_i$ (e.g., log-loss for binary classification).
- $\Omega(f_k)$: The regularization term that penalizes model complexity to prevent overfitting.
- Θ: The set of parameters for the XGBoost model.
- f_k: The k-th decision tree in the ensemble.
- n: The number of training examples.
- K: The total number of trees in the model.

Hyperparameter tuning was performed using Grid Search with 5-fold cross-validation to optimize parameters such as the number of estimators ($n\_estimators$), learning rate ($learning\_rate$), maximum tree depth (max_depth), minimum child weight (min_child_weight), and subsampling ratio ($subsample$). The best combination of hyperparameters was selected based on cross-validated accuracy.

C. Support Vector Machines

Support Vector Machines (SVM) were employed to predict whether an article would be popular, utilizing their ability to find an optimal hyperplane for binary classification. The SVM model aims to maximize the margin between the two classes, defined as:

$$\mathcal{L} = \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{n} \max(0, 1 - y_i(w \cdot x_i + b)) \tag{7}$$

where:

- $\mathcal{L}$: The loss function to minimize.
- w: The weight vector that defines the orientation of the hyperplane.
- b: The bias term that shifts the hyperplane.
- C: The regularization parameter that controls the trade-off between maximizing the margin and minimizing classification errors.
- x_i: The feature vector of the i-th training sample.
- y_i: The label of the i-th training sample ($y_i \in \{-1, 1\}$).
- $\|w\|^2$: The squared norm of the weight vector, representing the margin width.
- $\max(0, 1 - y_i(w \cdot x_i + b))$: The hinge loss for misclassified or marginally classified samples.

In this implementation, the SVM classifier was configured with:

- Kernel ($kernel$): A linear kernel was selected to separate the data using a linear hyperplane.
- Random state ($random\_state$): Set to ensure reproducibility of results.

The model was trained on the scaled training data and evaluated on the test set. Metrics such as precision, recall, F1-score, and accuracy were computed to assess the performance of the classifier. This approach ensured the SVM effectively identified patterns in the feature space to predict article popularity.

D. Naive Bayes Classifier

The Naive Bayes classifier was employed to predict whether an article would be popular, leveraging its simplicity and probabilistic approach. The Gaussian Naive Bayes model assumes that the features follow a Gaussian (normal) distribution. The probability of a class given the input features is calculated using Bayes' Theorem:

$$P(y \mid X) = \frac{P(X \mid y)P(y)}{P(X)} \tag{8}$$

where:

- $P(y \mid X)$: The posterior probability of the class y given the input features X.
- $P(X \mid y)$: The likelihood of the input features given the class y, modeled as a Gaussian distribution:

$$P(X_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(X_i - \mu_y)^2}{2\sigma_y^2}\right) \tag{9}$$

- $P(y)$: The prior probability of the class y.
- $P(X)$: The evidence, which is the probability of the input features X, ensuring the posterior probabilities sum to 1.
- μ_y: The mean of the feature for class y.
- σ_y^2: The variance of the feature for class y.
- X_i: The i-th feature of the input data.

Hyperparameter tuning was performed using Grid Search with 5-fold cross-validation to optimize the following parameter:

- Variance smoothing ($var\_smoothing$): A small value added to the variance of each feature to prevent numerical instability during calculations. The grid search explored a range of values logarithmically spaced between 10^0 and 10^{-9}.

The best combination of hyperparameters was selected based on cross-validated accuracy.

V. Unsupervised Learning

Unsupervised learning methods were applied to explore patterns within the dataset by clustering articles based on various feature subsets. Each subset represented specific characteristics of the articles, and clustering was combined with Principal Component Analysis (PCA) for dimensionality reduction. The following clustering techniques were implemented:

A. K-Means Clustering

The K-Means algorithm partitions data into k clusters by minimizing the within-cluster variance. The objective function for K-Means is:

$$\mathcal{L} = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2 \tag{10}$$

where:

- $\mathcal{L}$: The total within-cluster variance.
- x: A data point in cluster C_i.
- μ_i: The centroid of cluster C_i.
- k: The number of clusters.

B. Gaussian Mixture Models (GMM)

GMM models the data as a mixture of Gaussian distributions, assigning probabilities for each point belonging to a cluster. The likelihood for GMM is:

$$P(X) = \sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(X \mid \mu_k, \Sigma_k) \tag{11}$$

where:

- $P(X)$: The likelihood of the data.
- π_k: The weight (prior probability) of the k-th Gaussian component.
- $\mathcal{N}(X \mid \mu_k, \Sigma_k)$: The Gaussian distribution with mean μ_k and covariance Σ_k.
- K: The number of components.

C. Agglomerative Clustering

Agglomerative Clustering is a hierarchical approach that iteratively merges data points or clusters based on similarity. The linkage criterion determines how distances between clusters are calculated:

$$D(A, B) = \min_{a \in A, b \in B} \|a - b\| \tag{12}$$

where A and B are clusters and $\|a - b\|$ is the Euclidean distance.

The dataset was divided into subsets to focus on specific article characteristics:

- X_words: Features related to word usage (e.g., token counts, lexical diversity).
- X_links: Features related to hyperlinks (e.g., number of links, self-references).
- X_digital: Features related to digital content (e.g., images, videos).
- X_kw: Keyword-related features (e.g., keyword frequency and averages).
- X_time: Temporal features (e.g., publication weekday, weekend indicators).
- X_nlp: Features derived from NLP (e.g., sentiment polarity, subjectivity).

Silhouette scores were calculated for each clustering method to evaluate the quality of the clusters. The silhouette score is defined as:

$$S = \frac{b - a}{\max(a, b)} \tag{13}$$

where:

- S: The silhouette score for a data point.
- a: The average intra-cluster distance.
- b: The average inter-cluster distance to the nearest neighboring cluster.

Conclusion & Summary of Unsupervised learning Models

This unsupervised learning analysis revealed distinct groupings within the dataset. Subsets like X_words and X_nlp demonstrated strong clustering performance, suggesting their importance in differentiating

articles. These insights can inform feature engineering and future analysis of article popularity.

These methods provided complementary insights:

- K-means efficiently partitioned the data into distinct groups.
- GMM revealed overlapping and elliptical clusters.
- Agglomerative clustering identified hierarchical relationships within the data.

VI. Ensemble Learning

An ensemble learning approach was implemented using Weighted Majority Voting to combine predictions from multiple classifiers, leveraging their collective strengths to improve predictive performance. The ensemble integrates Logistic Regression, XGBoost, Naive Bayes, and Support Vector Machines (SVM). Each classifier's contribution to the final prediction was weighted based on its performance. The prediction for each data point was computed as:

$$\hat{y} = \text{round}\left(\frac{\sum_{i=1}^{n} w_i \cdot \hat{y}_i}{\sum_{i=1}^{n} w_i}\right) \tag{14}$$

where:

- $\hat{y}$: The final ensemble prediction.
- n: The total number of classifiers in the ensemble.
- w_i: The weight assigned to the i-th classifier, reflecting its relative importance.
- $\hat{y}_i$: The predicted label from the i-th classifier.

In the implementation:

- Classifiers included:
  - Logistic Regression (optimized with Grid Search).
  - XGBoost (optimized with Grid Search).
  - Naive Bayes (optimized with Grid Search).
  - Support Vector Machines (with a linear kernel).
- Weights were manually assigned ($[0.8, 1.0, 0.6, 0.9]$) based on the individual classifier's performance metrics.
- Predictions were aggregated using a weighted average of the classifiers' outputs.
- The final prediction was obtained by rounding the weighted average to the nearest integer.

The ensemble was trained on the scaled training data and evaluated on the test set using metrics such as precision, recall, F1-score, and accuracy. By combining classifiers with Weighted Majority Voting, the ensemble leveraged the complementary strengths of different models, achieving robust and accurate predictions for the task of predicting article popularity.

VII. Results and Discussion

Supervised Learning Performance

To evaluate the predictive performance of different supervised learning methods, we compared the classification metrics of Logistic Regression, XGBoost, Naive Bayes, and Support Vector Machines (SVM). The models were assessed on the test set using precision, recall, F1-score, and accuracy. Below are the results for each model:

A. Logistic Regression

Logistic Regression achieved an overall accuracy of 66%. While the precision and recall for both classes were balanced, the model performed slightly better for class 1 (popular articles) with a recall of 70%. The detailed classification report is shown in Table IV.

Model	Class	Precision	Recall	F1-Score	Support
Logistic Regression	0	0.66	0.62	0.64	3604
	1	0.67	0.70	0.68	3929
Accuracy		**0.66**		7533	
Macro Avg	0.66	0.66	0.66	7533	
Weighted Avg	0.66	0.66	0.66	7533	

TABLE IV: Logistic Regression Classification Report

B. XGBoost

XGBoost performed the best among the evaluated models, achieving an overall accuracy of 69%. It demonstrated balanced precision and recall for both classes, with a higher recall of 73% for class 1. The detailed classification report is provided in Table V.

Model	Class	Precision	Recall	F1-Score	Support
XGBoost	0	0.69	0.64	0.66	3604
	1	0.69	0.73	0.71	3929
Accuracy		**0.69**		7533	
Macro Avg	0.69	0.68	0.68	7533	
Weighted Avg	0.69	0.69	0.69	7533	

TABLE V: XGBoost Classification Report

C. Naive Bayes

Naive Bayes achieved an overall accuracy of 61%. The model performed well for class 0 (non-popular articles) with a recall of 78%, but struggled with class 1, achieving a recall of only 47%. The detailed classification report is shown in Table VI.

Model	Class	Precision	Recall	F1-Score	Support
Naive Bayes	0	0.57	0.78	0.66	3604
	1	0.69	0.47	0.56	3929
Accuracy		**0.61**		7533	
Macro Avg	0.63	0.62	0.61	7533	
Weighted Avg	0.64	0.61	0.61	7533	

TABLE VI: Naive Bayes Classification Report

D. Support Vector Machines (SVM)

The SVM classifier achieved an overall accuracy of 66%, similar to Logistic Regression. It demonstrated balanced performance, with slightly better recall for class 1 at 69%. The detailed classification report is presented in Table VII.

Model	Class	Precision	Recall	F1-Score	Support
Support Vector Machines	0	0.65	0.62	0.63	3604
	1	0.66	0.69	0.68	3929
Accuracy		**0.66**		7533	
Macro Avg	0.65	0.65	0.65	7533	
Weighted Avg	0.66	0.66	0.66	7533	

TABLE VII: Support Vector Machines Classification Report

Ensemble Classifier Performance

The ensemble classifier was implemented using a weighted majority voting approach to combine predictions from multiple supervised learning models. This method leverages the strengths of individual classifiers to enhance overall predictive performance. The final prediction for each data point was determined by weighting the predictions of the base models and selecting the class with the highest weighted average.

The ensemble achieved an accuracy of 66% on the test set, with balanced precision and recall across both classes. Class 1 (popular articles) exhibited slightly better recall (70%) and F1-score (0.69), indicating the ensemble's ability to correctly identify popular articles more effectively. The detailed classification report is provided in Table VIII.

Model	Class	Precision	Recall	F1-Score	Support
Ensemble Classifier	0	0.66	0.62	0.64	3604
	1	0.67	0.70	0.69	3929
Accuracy		**0.66**		7533	
Macro Avg	0.66	0.66	0.66	7533	
Weighted Avg	0.66	0.66	0.66	7533	

TABLE VIII: Ensemble Classifier Classification Report

Unsupervised Learning Performance

To evaluate the performance of the unsupervised clustering methods, we visualized the resulting clusters for each feature subset after applying K-Means, Gaussian Mixture Models (GMM), and Agglomerative Clustering. Principal Component Analysis (PCA) was used to reduce the dimensions of the data to two for visualization purposes.

1. K-Means Clustering

Figure 4 shows the clustering results for K-Means.

- X_words: The clusters are well-separated along the first dimension, indicating distinct patterns in word usage.
- X_links: The clustering captures variations in hyperlink-related features but shows overlapping regions, suggesting less distinct separation.
- X_digital: The clusters are more diagonal, reflecting relationships between digital content features such as images and videos.
- X_kw: Keyword-related features exhibit some overlap, with better separation along the second dimension.
- X_time: Temporal features show sparse and inconsistent clustering, likely due to limited variability in this subset.
- X_nlp: The clusters are distinctly separated, suggesting that NLP-derived features like sentiment polarity and subjectivity contribute strongly to clustering.

2. Gaussian Mixture Models (GMM)

Figure 5 presents the clustering results for GMM.

- X_words, X_nlp: Similar to K-Means, these subsets show strong cluster separation, with GMM offering a smoother transition between clusters due to its probabilistic nature.
- X_links, X_digital, X_kw: The clustering is slightly more spread out, capturing overlaps between regions, which aligns with the nature of GMM.
- X_time: The sparse clustering observed with K-Means persists here, indicating limited clustering potential for temporal features.

3. Agglomerative Clustering

The clustering results for Agglomerative Clustering are shown in Figure 6.

- X_words, X_nlp: Similar patterns to K-Means and GMM, with clear separation of clusters.
- X_links, X_digital, X_kw: These subsets show reasonable clustering, though with a tendency to form larger clusters compared to K-Means and GMM.
- X_time: Temporal features again fail to produce meaningful clusters, indicating limited variability or influence in this subset.

The clustering analysis revealed the following key insights:

- NLP-derived features (X_nlp) and word usage features (X_words) consistently formed the most distinct clusters, highlighting their importance in distinguishing articles.
- Hyperlink-related (X_links) and digital content features (X_digital) provided moderate separation, while keyword features (X_kw) were less distinct.
- Temporal features (X_time) showed minimal clustering potential, likely due to their categorical or sparse nature.
- GMM performed well in capturing overlapping clusters, while K-Means and Agglomerative Clustering provided sharper separation in subsets with distinct patterns.

VIII. Conclusion

This paper implemented a series of machine learning algorithms to conduct a comprehensive research on online news popularity dataset to investigate the factors that contribute to the popularity of online news articles. The process commenced with the definition of popularity utilising a threshold of 1,400 shares, followed by comprehensive preprocessing, which included outlier detection through Isolation Forest, data

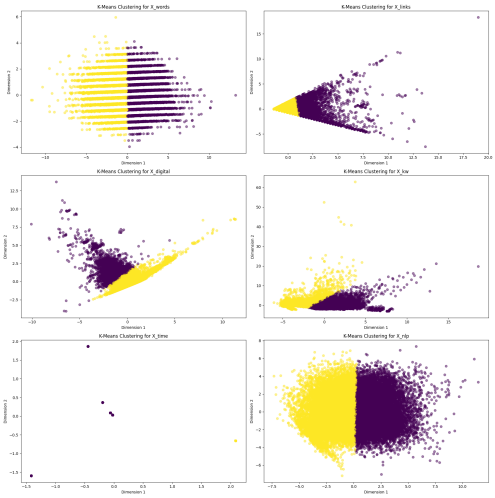

Fig. 4: K-Means Clustering Results for Different Feature Subsets

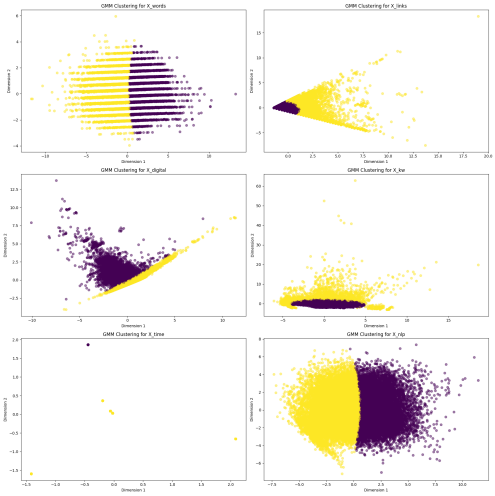

Fig. 5: Gaussian Mixture Models (GMM) Clustering Results for Different Feature Subsets

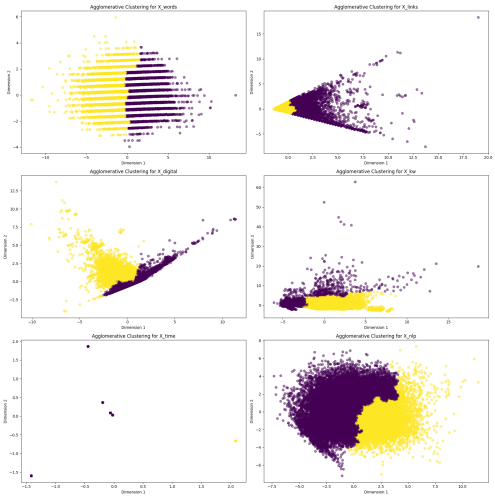

Fig. 6: Agglomerative Clustering Results for Different Feature Subsets

standardisation, and the partitioning of data into training and testing subsets. For the modeling stage, first supervised learning models were implemented for the purpose of classifying articles. The XGBoost model demonstrated the highest level of accuracy at 69%, with the Logistic Regression, SVM, and Naive Bayes models exhibiting comparable performance. Furthermore, ensemble learning technique, which is based on supervised models enhanced the prediction accuracy through weighted majority voting, which considered the strengths of the base models. Beyond these supervised models, K-Means, Gaussian Mixture Models, and Agglomerative Clustering techniques are employed in terms of unsupervised models. Effective clustering has been shown for sentiment and word use characteristics acquired from natural language processing (NLP). It has been discovered that there is little chance of clustering temporal and keyword-related variables. These results have been confirmed by a further decrease of dimensionality using PCA. The results of the research demonstrated the importance of content-level characteristics, specifically word structure and sentiment polarity, in predicting and interpreting article popularity. These results provide practical advice for enhancing digital content marketing tactics. By incorporating more data or experimenting with sophisticated neural network topologies for more gains, future study might expand on this.

REFERENCES

[1] Fernandes, K., Vinagre, P., and Cortez, P. (2015). A proactive intelligent decision support system for predicting the popularity of online news. In Portuguese Conference on Artificial Intelligence, pages 535–546. Springer.