# Reproducing “A Proactive Intelligent Decision Support System for Predicting the Popularity of Online News”

Irene Burri

January 13, 2026

## 1 Introduction

The dataset used in this study was obtained from the UCI Machine Learning Repository[1]. The corresponding paper selected for replication is entitled “A Proactive Intelligent Decision Support System for Predicting the Popularity of Online News”[2].

The paper focuses on predicting the popularity of online news articles using machine learning and optimization techniques. The project develops a predictive framework that evaluates five classification models (Random Forest, AdaBoost, Support Vector Machine and Naive Bayes) and incorporates an optimization component aimed at maximizing article engagement.

## 2 Problem description

Online content platforms produce a large number of articles every day, making it difficult to predict in advance which content will become popular. This challenge is particularly relevant for large digital media platforms such as Mashable, which receive thousands of article submissions but are able to publish only a limited fraction of them.

Being able to predict the popularity of online news articles is a relevant problem, as it can support several aspects of the editorial process. In particular, popularity prediction can assist in identifying topics and formats with higher impact before publication, improving the allocation of promotional resources, guiding content creation toward themes that are more likely to attract user engagement, and providing proactive data driven support to editorial decision making.

The problem addressed in the original paper and replicated in this project, is tackled through a two approach: a predicting modeling part to classify article popularity (section 3.2), a feature optimization part that searches for feature adjustments capable of increasing expected popularity (section 3.3).

The original paper is relevant because it introduces a proactive Intelligent Decision Support System that integrates prediction and optimization within an Adaptive Business

[1] Dataset link: `https://archive.ics.uci.edu/dataset/332/online+news+popularity`

[2] Paper link: `https://www.semanticscholar.org/paper/A-Proactive-Intelligent-Decision-Support-System-i`
`ad7f3da7a5d6a1e18cc5a176f18f52687b912fea`

Intelligence framework. It focuses on prepublication features, evaluates five classifiers on a large Mashable dataset using rolling windows and applies a stochastic hill-climbing method to optimize article attributes that can realistically be edited by authors.

# 3 Methodology

This project replicates the predictive and optimization framework proposed in the original paper by reconstructing its main components: the dataset preparation, the prediction module, and the optimization module. The methodology follows the Adaptive Business Intelligence (ABI) paradigm, in which prediction and optimization are combined to support proactive decision making.

## 3.1 Data description

The dataset used in this study contains 39644 news articles published on Mashable from 2013 to 2015, each described by 58 attributes. These features include article properties (e.g title, content length, keyword count), media elements (e.g. images/videos), temporal information (e.g. day, hour) and content metrics (e.g.polarity, subjectivity). The target variable corresponds to the number of shares an article received on social media.

Following the original paper, the prediction task is a binary classification problem. Articles with fewer than $D_1 = 1400$ shares are labeled as unpopular (Class 0, 51.2%), while those above this threshold are labeled as popular (Class 1, 48.8%).

## 3.2 Prediction module

The prediction module estimates the likelihood that a candidate article will become popular. Five machine learning models are evaluated, following the configuration described in the original paper: Random Forest, Adaptive Boosting, Support Vector Machine, Naive Bayes and K-Nearest Neighbors. All models are trained and tested using a rolling window evaluation scheme designed to preserve temporal order and prevent information leakage.

Each iteration uses a window of $W = 10000$ articles for training and the subsequent $L = 1000$ articles for testing, resulting in 29 iterations and a total of 29000 test samples. Within each training window, a 70/30 split is applied to perform grid search over the model hyperparameters. The best configuration is then retrained on the full training window before evaluation on the corresponding test window.

The hyperparameter grids replicate those of the original study: Random Forest and AdaBoost with number of estimators in $\{10, 20, 50, 100, 200, 400\}$, SVM with RBF kernel and $C \in [2^0, \dots, 2^6]$, KNN with $k \in \{1, 3, 5, 10, 20\}$ and Naive Bayes with no tunable parameters (see table 1).

The primary evaluation metric is AUC, while accuracy, precision, recall and F1-score serve as secondary metrics.

All features available prior to publication are included as inputs. The models output a predicted probability of popularity and class labels are assigned using the standard decision threshold of $D_2 = 0.5$.

Table 1: Hyperparameter search grids

| **Model** | **Grid** |
|---|---|
| Random Forest | n_estimators: [10, 20, 50, 100, 200, 400] |
| AdaBoost | n_estimators: [10, 20, 50, 100, 200, 400] |
| SVM | C: $[2^0, 2^1, \ldots, 2^6]$ |
| KNN | n_neighbors: [1, 3, 5, 10, 20] |
| Naive Bayes | no hyperparameters |

## 3.3 Optimization module

The optimization module aims to identify feasible modifications to an article's attributes that can increase its predicted popularity. It considers editable only attributes that can realistically be adjusted by authors or editors, such as the number of images, title polarity and the presence of specific keywords, while immutable metadata is excluded.

Starting from the original feature vector of an article, the module applies a stochastic hill climbing local search strategy. At each iteration, small random perturbations (±1) are applied to the editable features and the resulting configuration is evaluated using the trained popularity prediction model. If the modified feature vector has an higher predicted probability of popularity, it is accepted as the new current solution according to a predefined acceptance probability $P$. This process is repeated iteratively until no further improvements are observed or a maximum number of iterations is reached.

The optimization procedure is applied to 1,000 articles classified as unpopular from the final test set. For each article, stochastic hill climbing is performed over successive iterations, exploring the feature space through random perturbations while selectively accepting improvements. To assess the robustness of the search strategy, acceptance probabilities $P \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ are tested, ranging from purely greedy search to fully stochastic exploration.

Two alternative feature subsets are evaluated: a configuration excluding features related to keyword and a configuration including keywords among the editable attributes. Optimization progress is monitored at iterations $\{0, 1, 2, 4, 8, 10, 20, 40, 60, 80, 100\}$ to analyze convergence behavior over time. Performance is evaluated using two complementary metrics: Mean Gain measures the average improvement in predicted popularity and Conversion Rate quantifies the proportion of unpopular articles optimized above decision threshold $D_2 = 0.5$.

A key methodological difference with respect to the original paper is about the treatment of keyword related attributes during the optimization phase. In the original study, each article is associated with an explicit set of keywords and the optimization procedure operates directly on this set by adding or removing individual keywords. Each modification triggers a recomputation of multiple derived features, including statistics related to the minimum, average and maximum number of shares associated with the selected keywords and features derived from referenced Mashable articles. This approach allows keyword perturbations to have a broad impact on the feature representation used by the prediction model.

In this replication study, the explicit list of keywords associated with each article is not available. So it is not possible to directly manipulate the keyword set or to recompute keyword dependent statistics. For this reason the perturbations on keyword related features is limited to the `num_keywords` attribute, which captures only the total number of keywords assigned to an article. All other keyword derived features are therefore kept fixed during the optimization process. This implies that the optimization procedure including keyword related features is simpler than in the original paper and consequently leads to smaller improvements in predicted popularity.

In the original paper the authors predefine a fixed subset of features to optimize that are modifiable before publication (see table 2). This selection is informed by the feature importance analysis of the Random Forest model, which highlights the attributes with the strongest influence on popularity prediction.

Table 2: Optimizable attributes considered in the original paper

| **Feature** | **Perturbation Strategy (Paper)** | **Replicated** |
|---|---|---|
| Number of words in the title ($n$) | $n' \in \{n-1, n+1\},\ n' \geq 0,\ n' \neq n$ | Yes |
| Number of words in the content ($n$) | $n' \in \{n-1, n+1\},\ n' \geq 0,\ n' \neq n$ | Yes |
| Number of images ($n$) | $n' \in \{n-1, n+1\},\ n' \geq 0,\ n' \neq n$ | Yes |
| Number of videos ($n$) | $n' \in \{n-1, n+1\},\ n' \geq 0,\ n' \neq n$ | Yes |
| Day of the week ($w$) | $w' \in [0, 7),\ w' \neq w$ | Yes |
| Number of keywords | $n \to n \pm k$ | Yes |
| Keyword derived statistics (min/avg/max shares) | Recomputed after keyword modification | No |

# 4 Results

## 4.1 Prediction results

Model performance is evaluated on all test sets produced by the rolling window approach, for a total of 29,000 predictions.

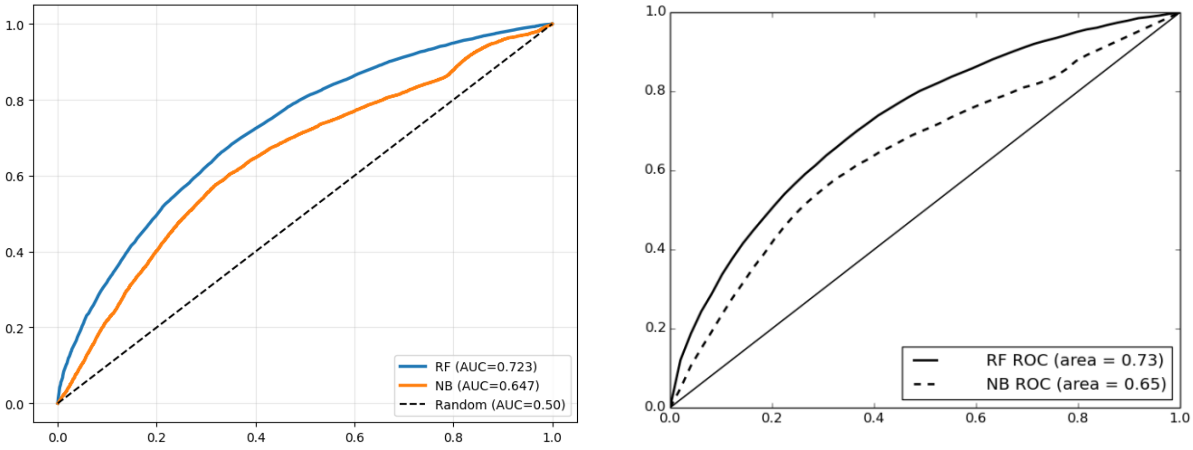
Random Forest achieved 0.723 AUC, matching paper expectations (AUC = 0.73). The 0.65-0.73 range demonstrates meaningful discrimination above random (0.50). Precision exceeds recall for all models, indicating conservative popularity predictions. KNN and Naive Bayes record the weakest results, particularly in terms of recall and F1-score, indicating difficulties in correctly identifying popular articles.

Overall, the ranking and absolute values of the performance metrics are highly consistent with the original paper expectation, as shown in table 3. The consistency across models confirms the robustness and reproducibility of the original findings.

These results validate the effectiveness of these methods, particularly Random Forests, for proactive online news popularity prediction under a realistic temporal evaluation setting.

Table 3: Results comparison

| **Model** | **Paper Results** | | | | | **This Study** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Accuracy | Precision | Recall | F1 | AUC | Accuracy | Precision | Recall | F1 | AUC |
| Random Forest (RF) | **0.67** | 0.67 | **0.71** | **0.69** | **0.73** | **0.664** | 0.655 | **0.638** | **0.646** | **0.723** |
| AdaBoost | 0.66 | **0.68** | 0.67 | 0.67 | 0.72 | 0.660 | **0.656** | 0.616 | 0.636 | 0.698 |
| SVM | 0.66 | 0.67 | 0.68 | 0.68 | 0.71 | 0.656 | 0.650 | 0.618 | 0.633 | 0.709 |
| KNN | 0.62 | 0.66 | 0.55 | 0.60 | 0.67 | 0.626 | 0.640 | 0.508 | 0.566 | 0.666 |
| Naive Bayes | 0.62 | **0.68** | 0.49 | 0.57 | 0.65 | 0.602 | 0.653 | 0.369 | 0.472 | 0.647 |



(a) ROC curves obtained in this study.

(b) ROC curves reported in the original paper.

Figure 1: Comparison between ROC curves obtained in this study and those reported in the original paper.

The feature importance analysis obtained from the Random Forest model trained on the full dataset is consistent with the results reported in the original paper. Keyword related features and LDA based topic features dominate the ranking, with keyword average statistics and topic proportions appearing among the top 15 most important variables. This confirms that keywords and latent topics are strong predictors, as observed in the original study. Although the exact importance values differ, the overall ranking supports the conclusion that keyword related features account for a substantial share of the model's explanatory power.

## 4.2 Optimization results

Table 4 summarizes the Mean Gain (MG) and Conversion Rate (CR) achieved under different acceptance probabilities $P$, for the feature subsets excluding and including keyword related attributes. Including keyword related features improves both optimization metrics. Compared to the configuration without keywords, the setup with keywords achieves higher Mean Gain values (up to 0.0748 versus 0.0692) and a higher Conversion Rate (up to 30.11% versus 26.15%). These results confirm that keyword related information positively contributes to the optimization process.

Across both feature subsets, purely random search ($P = 1.0$) yields the weakest performance, while low to intermediate values of $P$ provide the best trade-off between exploration and exploitation. In contrast with the original paper, where the best optimization results were reported for higher stochasticity levels ($P \approx 0.8$), the optimal configurations in this replication are observed for lower values of $P$, specifically $P \in \{0.0, 0.2\}$.

Table 4: Optimization performance with and without keyword related features.

| $P$ | **Without Keywords** | | **With Keywords** | |
|---|---|---|---|---|
| | **MG** | **CR** | **MG** | **CR** |
| 0.0 | 0.0663 | 25.52% | **0.0748** | 28.68% |
| 0.2 | **0.0692** | **26.15%** | 0.0743 | **30.11%** |
| 0.4 | 0.0679 | 25.36% | 0.0720 | 26.62% |
| 0.6 | 0.0654 | 25.04% | 0.0705 | 27.89% |
| 0.8 | 0.0634 | 23.61% | 0.0674 | 27.10% |
| 1.0 | 0.0613 | 22.03% | 0.0646 | 25.04% |

Figure 2 illustrates the convergence behavior of the Mean Gain across optimization iterations. In all configurations, the optimization process exhibits a rapid initial improvement, followed by a gradual saturation after approximately 80-100 iterations. This convergence pattern is consistent with that reported in the original study and indicates diminishing returns as the number of iterations increases.

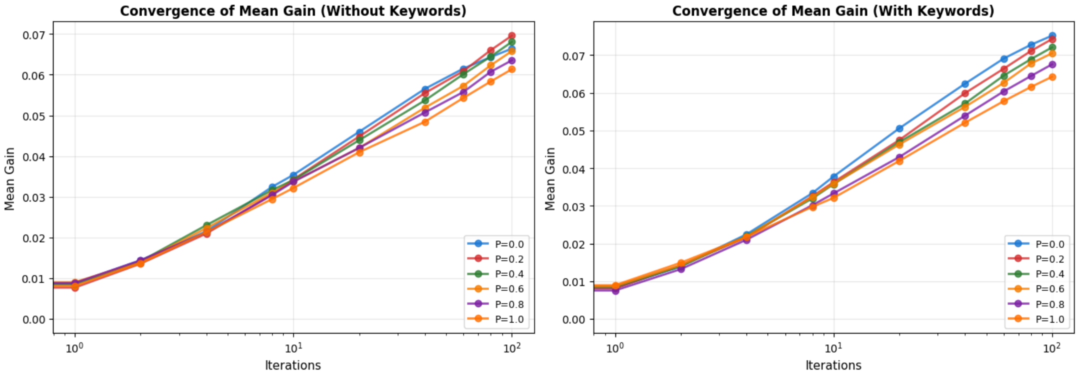


Figure 2: Convergence of MG

Figure 3 provides a direct comparison of optimization performance results from this study and the original paper. In both cases, intermediate values of $P$ outperform purely greedy and purely random strategies. However, the achieved improvements differs, while the original study reports substantially higher Mean Gain values when keyword based edits are included, the present replication reaches lower absolute gains.

These quantitative differences can be attributed to the simplified handling of keyword information in the replication. Unlike the original work, which performs add/remove operations on the actual keyword set and recomputes multiple keyword based statistics, this study limits keyword related perturbations to the number of keywords only. Despite this limitation, the qualitative behavior of the optimization process is preserved.

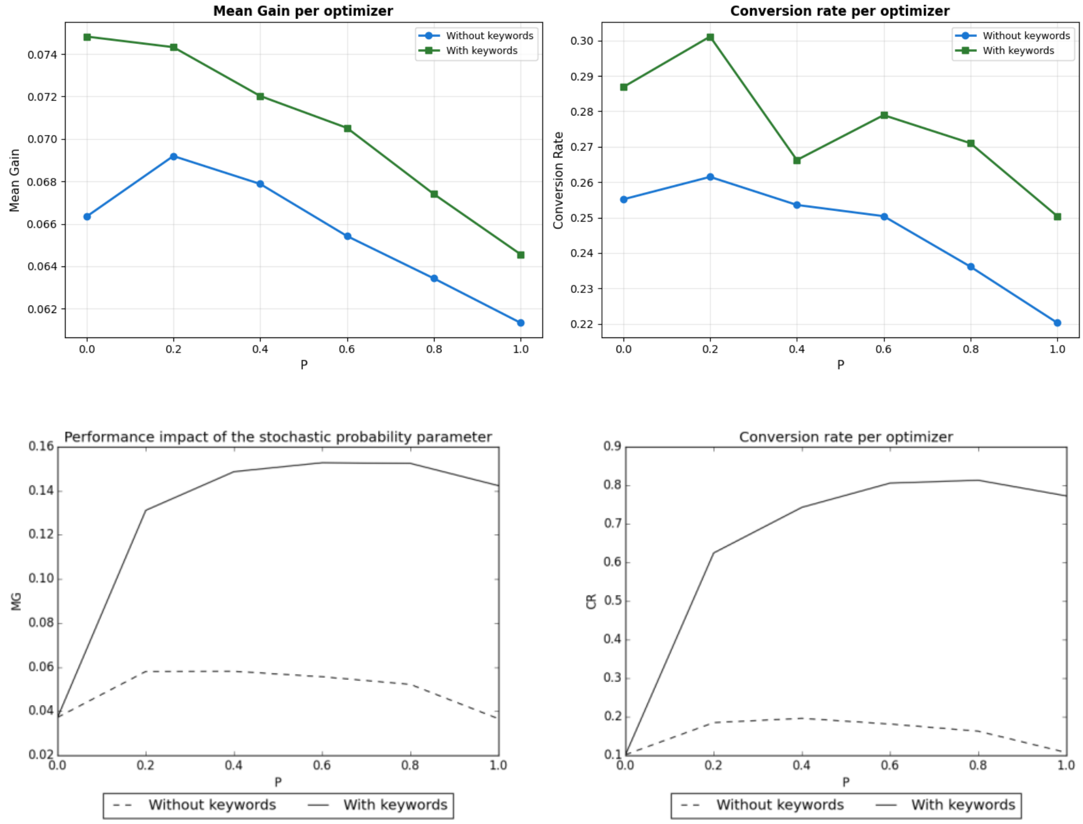


Figure 3: Mean Gain and Conversion Rate versus the acceptance probability $P$ for this study (top) and the original paper (bottom).

# 5 Conclusion

This report successfully replicates the predictive and optimization framework proposed in the original paper on online news popularity prediction. The prediction results are highly consistent with the original findings, with Random Forest achieving comparable performance under a realistic rolling window evaluation. Feature importance analysis confirms that keyword related attributes and LDA based topic features play a central role in popularity prediction.

The optimization module reproduces the main qualitative behaviors observed in the original study, including rapid initial improvements and convergence after a limited number of iterations. While the inclusion of keyword related features improves optimization outcomes, the achieved gains are smaller due to limitations in keyword manipulation compared to the original implementation. Overall, the results validate the robustness of the proposed decision support framework and highlight the importance of rich, editable content features for effective optimization.