2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES) | 978-1-6654-3521-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICSES52305.2021.9633817

Application and evaluation of Machine Learning for news article popularity prediction

Sejal Bhatia

Department of Computer Engineering, Dr. D. Y. Patil Institute of Technology, Pune, India
E-mail : sejal.sbhatia1@gmail.com

Abstract :- The internet is increasingly becoming the primary source of news worldwide. Social networking sites have further enabled instantaneous spread of such articles by often allowing single-click user sharing. Majority of the organizations publishing such articles drive revenue through advertisements which is ultimately dependent on the popularity of the article. This popularity is mainly defined in terms of views and shares. One of the emerging applications of Machine Learning is to help organizations predict which articles are most likely to become popular and thus allow them to improve targeted advertising campaigns in order to optimize revenue. This paper proposes and evaluates Machine Learning based approaches alongside Rolling, Growing and a Hybrid training window techniques in order to predict the popularity of news articles..

***Index Terms*—article popularity, news popularity, rolling win- dow, growing window, machine learning, k-nearest neighbor, decision tree, xgboost, cross-validation, random forest**

I. INTRODUCTION

Over the last couple of decades, online presence has become a key factor in the success of any news organization. As hundreds of thousands of new users gain access to the Internet every day, it has become the most effective way to reach your audience for any media platform. No other medium even comes near the ability of the internet when we consider how many users can be reached every second. The very nature of the internet demands open access and thus creates a fierce and competitive market for user's attention. Most, if not all, the revenue associated with this market is driven by advertisement. This creates an additional incentive for organizations to do everything in their power to ensure that their content reaches as many users as possible.

This is where the explosive growth of social media platforms such as Facebook, Twitter, YouTube, etc. injects itself in the middle. More and more individuals [1] all around the world admit to getting their news via one of the above mentioned social media platform. One of the main features of any social media platform is the ability it grants users to share articles. This ability turns users into relays for content developers, allowing them to reach users who they may not have been able to reach on their own. Number of times an article gets shared has become a key metric in determining the popularity it was able to gain. Moreover, with the growing realization among users with respect to fake or untrustworthy news articles online, popularity

or shares on an article are also becoming factors that users use to judge the legitimacy of the articles. Given that popular articles are seen as more reliable and thus become more popular, and in turn drive more revenue from advertisements, it provides even more of a reason for organizations to ensure that they propagate articles that are more likely to be popular.

One of the major challenges in the application of Machine Learning to predict the popularity of news articles is presented by the temporal nature of the data. A well-trained and static Machine Learning model cannot be used for this application as the features of an article, that determine whether it eventually becomes popular, are constantly and rapidly evolving. This drives the need of models that can continue to learn from the new information and adapt to stay relevant. This is commonly achieved by applying a rolling window approach to train models that makes use of new data points while ridding itself of the oldest data points. In this paper, we evaluate a variety of Machine Learning algorithms alongside conducting experiments to determine the Optimal Rolling Window Size (ORWS). Moreover, we go one step further to evaluate and compare the rolling window approach to a growing window approach where we don't remove old data points. Finally, we propose a hybrid window approach where we attempt to alleviate the cold-start problem by combining a rolling window with a growing window approach.

All the experiments performed in this paper make use of a news article popularity dataset collected from the publishing site Mashable. This dataset contains information about 39,797 articles collected over a period of two years. We evaluate the window based approaches alongside cross-validation on the dataset to ensure a thorough evaluation of the ability of the targeted Machine Learning algorithms (which included K- Nearest Neighbors, XGBoost, Decision Tree, Random Forest) to predict whether an article will be popular or not based on metadata attributes such as number of words, images or videos, day of the week on which the article was published, polarity of the article, number of other articles linked within the article etc.

The paper from here on is organized as, Section II provides an overview of the relevant work we identified and improved upon, Section III provides a brief introduction to the Machine Learning algorithms and Section IV elaborates on the proposed approaches. In section V, we note our observations based on the empirical results and section VI provides a conclusion of our work. Section VII provides a glance at possible future work.

II. Related Work

Kelwin Fernandes, et al. [2] proposed a system for Intelligent Decision Support (IDSS). They tried to predict the popularity of an article before being published. They improved a few of the article features using optimization techniques in order to maximize popularity for a given article. They performed a rolling window evaluation on 39000 news articles that were collected over a 2-year period from the Mashable website, and tested five classification models under distinct metrics. They achieved an overall AUC (Area Under the Curve) of 73% using Random Forest (RF) classifier. Further, He Ren and Quan Yang in [4] made some changes in the work done by Kelwin Fernandes, et. al. [2] and got an accuracy of 69% by using Random Forest and considering top 20 features.They used techniques such as Fisher Criterion and Mutual Information to maximize accuracy of feature selection, based on which the algorithms were evaluated.

Ioannis Arapakis, et al. [3] used 13319 online news articles collected from Yahoo! News website. They targeted the cold-start online news popularity prediction problem. They collected page views, tweet counts and observed values at 1 hour, 1 day, & 1 week after an article is published to determine popularity. They achieved 79.7% accuracy using SVM algorithm. They observed that inclusion of early-stage popularity measurements as features helps improve model performance [4].

Alexandru Tatar, et al. [5] used the data from two news sites based out of Europe. They considered user comments as a factor of news popularity. They used linear regression and constant scaling models for prediction. They report that a linear model on a logarithmic scale is the most efficient method to rank online news articles.

Ananto Setyo Wicaksono et al. [6] considered online news article data from Mashable. They proposed that in order to achieve a higher accuracy of prediction using Machine Learning models, hyper-parameter tuning needs to be applied which can be time-consuming if we choose grid search. Hence, they proposed genetic algorithm to reduce computation time. They achieved a reduction in computational time by 425.06% with Support Vector Machine, 17% with Random forest, 651.06% with Adaptive Boosting, and 396.72% with K - Nearest Neighbor.

Aasim Khan, et al. [7] also used data from many online news articles from Mashable news service and considered articles above 3395 shares as popular. They report that the top three models were based on Gradient Boosting, Logistic Regression and Random Forest algorithm with Gradient Boost outperforming others with a mean accuracy of 79.7%.

Dhanashree Deshpande [8] used AdaBoost, LPBoost and Random Forest with feature reduction techniques like Linear Discriminant Analysis. They evaluated these techniques on the Mashable news dataset and identified AdaBoost as the optimal model which achieved 73% F-measure and 69% accuracy.

III. Algorithms

We evaluated the performance of the following four Supervised Machine Learning algorithms on predicting activity on each of these datasets:

A. Decision Tree Classifier

Decision Tree method generates a tree structure with leaf nodes, internal nodes and a root node [9]. Internal nodes, branches and leaf nodes represent the features of the dataset, decision rules, and the outcome or class respectively. At each node, training data is split according to the number of classes. The classifier picks the feature that best splits the data based on splitting criteria (e.g. information gain, gini index, etc.).

B. K-Nearest Neighbor Classifier

K-nearest neighbors [10] is a lazy algorithm that stores all available records and classifies new records based on a similarity measure by calculating distance between records. KNN classifier assigns a new data point to a class that is nearest (i.e. by distance value) to the majority of its K neighbors. Various metrics can be used for measuring similarity, with the most popular being Manhattan and Euclidean distance. Once the distance between training data points and the new data point is calculated, the nearest K neighbors determine the class of the new data point based on a majority vote amongst themselves.

C. Random Forest Classifier

Random Forests (RF) [11] is an ensemble classification algorithm and uses decision trees as base classifiers. Random Forest builds multiple trees via randomization technique. The major advantage of Random Forest over Decision Trees is improved accuracy and fewer chances of over-fitting since it uses an aggregate of several Decision trees. The ultimate prediction is determined by a majority vote amongst the Decision Trees in the forest.

D. XGBoost Classifier

Gradient Boosting and XGBoost are both iterative methods that use gradient descent to slowly improve weak tree-based learners [12]. Generally, Gradient Descent algorithm is used in Gradient Boosting to build a new tree that corrects the errors made by previous trees, and hence the tree construction is slow. XGBoost optimizes the Gradient Boosting architecture through distributed implementation, and hence improves its performance and speed.

IV. Approach

In this paper we use a news article popularity dataset provided by UCI Machine Learning Repository. This dataset contains statistical attributes about 39,797 articles collected in a 2-year time frame by Mashable. The dataset contains 61 attributes, of which 58 were predictive, 2 were non-predictive and 1 goal field. Non-predictive attributes, *url* and *timedelta*, were not included while training the model. The goal field identified in this dataset is identified as the frequency of an article being shared. We have followed the same approach as the original authors who collected the dataset to transform the prediction target into a binary class as follows:

$$IsPopular? = \begin{cases} 0, & \text{if } numberOfShares < 1400 \\ 1, & otherwise \end{cases} \tag{1}$$

In order to thoroughly understand and evaluate the dataset, we started by reproducing the results presented in [2] using a rolling window approach. Additionally, we used cross validation over 10 folds to evaluate different machine learning algorithms and tune their parameters to achieve optimum performance. Finally, we propose and evaluate two novel approaches to ensure the most effective and efficient use of the window based approach. In this section, we take a detailed look at these to understand the approaches and the reasoning behind them.

A. Rolling Window

For this dataset, cross-validation alone is not sufficient in order to evaluate the prediction capability of the algorithms under test. The news articles are collected over a period of time, denoted by the *timedelta* attribute which is defined by the difference between the dataset acquisition date and the date of article publication [2]. This calls for an evaluation of rolling and growing window approaches where we judge the performance of the algorithms in predicting the shares for newly published articles, given the amount of data we have already collected at any particular instant of time T.

In rolling window evaluation, as explained in [12], the model is trained using W consecutive samples (which identifies the training window). Then, the next L consecutive samples (testing window) are considered for testing the model, and AUC (Area Under the Curve) is recorded. For the next iteration, we drop L oldest samples from W and add L new samples in W to fit the model, finally evaluating the model on the next set of L samples, and so on. Hence, the window size remains the same and we travel from oldest W samples to latest W samples to train the model. Finally, for a given window size W we calculate the average AUC across all iterations.

To start the experiments, we considered a rolling window size (W) of 5000 samples for training and a window (L) of 1000 samples for testing, and recorded the performance of each algorithm. Next, we increased the rolling window size by 2500 samples, i.e., the new rolling window size (W) became 7500, while testing window size (L) remained the same, i.e., 1000. Similarly, we kept on increasing the rolling window size (W) to 10000, 12500, 15000, 17500 and 20000 while keeping testing window size (L) at 1000.

B. Growing Window

A growing window, unlike rolling window, does not drop older samples but keeps on adding L samples to the training window W and consider the next L samples for testing. Hence, instead of dropping and losing older samples the window size keeps on growing. We started with 5000 samples in the training window, and tested with the next 1000 records. We then kept adding 1000 samples in the training window at each iteration, selecting the next 1000 for testing, and continued this until we reached 39000 training samples.

C. Hybrid Approach

Recommender systems are popular for struggling with the Cold Start problem and researchers are actively trying to find new and optimized ways to address it [13]. News popularity prediction is not immune from the Cold Start problem, specifically the new item problem. When a news publishing platform is launched, it takes days, or sometimes months for the platform to have enough data about articles and their popularity, delaying our ability to start predicting popularity of articles yet to be published.

When we compared the *timedelta* between the first news article in the dataset and the 10000th article, we noticed that they were collected about 6 months apart. This means that, with the rolling window technique presented in [2], we would not be able to make any predictions until we had at least 6 months of data from the website. Our paper tries to minimize this requirement, by attempting to start making predictions with as little as 5000 records (approximately 3 months of data).

In this paper, we present a Hybrid approach where we considered the optimal window size by comparing results from growing window and rolling window approach. We call this the Optimal Rolling Window Size (ORWS). We then continued to grow our training window until we reached ORWS. Once we reached ORWS, we treated the range of data similar to the original rolling window approach as described in the subsection above.

V. Experimental Results

To start, we evaluated the Machine Learning algorithms using 10-fold cross-validation. We also used this as an opportunity to tune the parameters of these algorithms using Grid Search in order to optimize their performance. Results of these experiments are presented in Fig. 1, which shows that both Random Forest and XGBoost were highly effective in predicting popularity, while the maximum AUC of 0.74 was achieved by XGBoost. Another key observation here is that Decision Tree, while it is relatively the most light-weight algorithm, was still able to achieve comparable performance.

We performed rolling window based evaluation on the dataset using rolling window sizes (W) ranging from 5000 to 20000 with increments of 2500 as below. The results of these experiments are shown in Fig. 2

$$W \in \{5000, 7500, 10000, 12500, 15000, 17500, 20000\}$$

It can be observed from Fig. 2 that a rolling window size of $W = 20000$ is the Optimal Rolling Window Size, with many algorithms showing an increase in performance as the rolling window size increases. Even though we have not included them here, we performed experiments with rolling window sizes greater than 20000. Those experiments yielded negligible lift for the model while exaggerating the need for additional data and consequently making the cold start problem worse. Another interesting observation we can make from Fig. 2 is that the performance of Random Forest across different window sizes was the most consistent while XGBoost was able to improve as more data was provided to it.

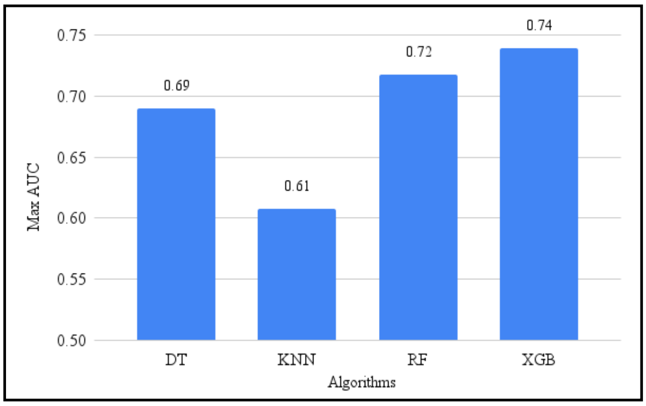

Fig. 1. AUC with best hyper-parameters using Cross-validation

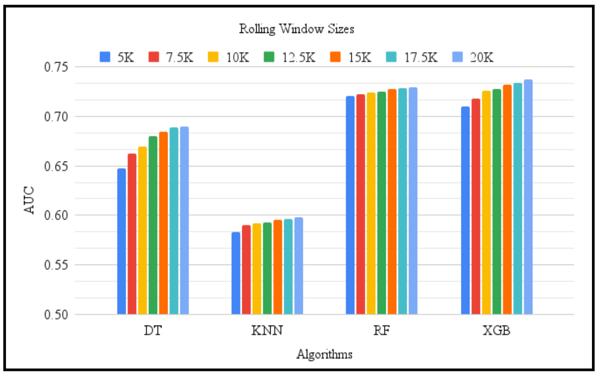

Fig. 2. AUC with different rolling window sizes

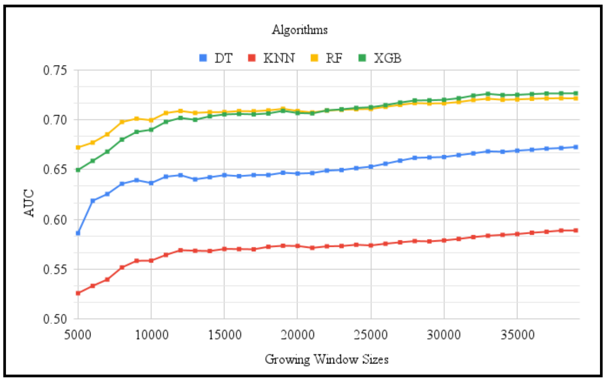

Fig. 3. AUC with different growing window sizes

Fig. 3 shows the variation in AUC for each algorithm as the growing window size increased. Comparing this to the results from Fig. 2 reinforces our observations regarding both Random Forest and XGBoost algorithms. Similar to Fig. 2, Random Forest demonstrates its robustness and XGBoost its improvement with growing window as shown in Fig. 3.

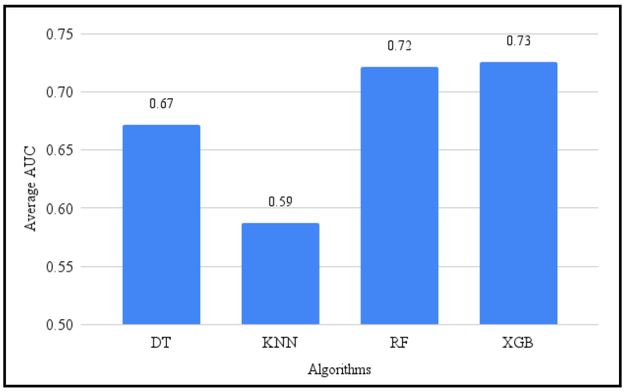

Fig. 4. AUC with hybrid approach

Fig. 4 presents the average AUC achieved with the selected algorithms using the Hybrid Window Approach. Even though the AUC did not improve compared to the cross-validation approach in Fig. 1, the hybrid approach provides the flexibility to start getting predictions on the dataset as early as 3 months from website launch as opposed to the 6 month waiting period of the rolling window approach presented by [2].

VI. Conclusion

This paper empirically provides several key takeaways with respect to the application of Machine Learning in order to predict article popularity. Experiments using Cross-fold validation clearly show the dominance of XGBoost algorithm when it comes to generic prediction. On the other hand, experiments with different rolling window sizes make Random Forest stand out with its ability to provide high AUC even with a small amount of data. The proposed Hybrid Window Approach also yields comparable results to the rolling window approach while simultaneously alleviating the cold start problem. This paper also shows that the availability and quantity of data should drive decisions while building solutions to predict news article popularity.

VII. Future Work

There are several avenues that can be explored beyond the work demonstrated by this paper. These include but are not limited to: experiments to evaluate how the threshold that determines popularity would impact performance of prediction, using a combination of Natural Language Processing and classification to evaluate how content of the articles drives popularity as opposed to just the metadata, treating this as a regression problem by keeping the target as Number of Shares, etc.

References

[1] I. Pentina, M. Tarafdar, “From “information” to “knowing”: Exploring the role of social media in contemporary news consumption,” Computers in Human Behavior, vol. 35, pp. 211-223, 2014.

[2] K. Fernandes, P. Vinagre, P. Cortez, “A Proactive Intelligent Decision Support System for Predicting the Popularity of Online News,” EPIA 2015, pp. 535–546, 2015.

[3] Ioannis Arapakis, B. Barla Cambazoglu, Mounia Lalmas, “On the Feasibility of Predicting News Popularity at Cold Start,” Springer, pp. 290-299, 2014.

[4] H. Ren, Q. Yang, “Predicting and Evaluating the Popularity of Online News,” Machine Learning Project Work Report, 2015, pp. 1-5.

[5] A. Tatar, P. Antoniadis, Marcelo Dias De Amorim, S. Fdida, “From popularity prediction to ranking online news,” Social Network Analysis and Mining 4.1, 2014, pp. 174.

[6] A. S. Wicaksono, A. A. Supianto, “Hyper Parameter Optimization using Genetic Algorithm on Machine Learning Methods for Online News Popularity Prediction,” International Journal of Advanced Computer Science and Applications, Vol. 9, No. 12, 2018.

[7] A. Khan, G. Worah, M. Kothari, Y. H Jadhav, A. V. Nimkar,” News Popularity Prediction with Ensemble Methods of Classification,” 9th ICCCNT 2018 July 10-12, 2018, IISC, Bengaluru.

[8] D. Deshpande, “Prediction & Evaluation of Online News Popularity us- ing Machine Intelligence,” 2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA).

[9] Y. Y. Song, Y. Lu, “Decision tree methods: applications for classifica- tion and prediction,” Shanghai Arch Psychiatry, vol. 27(2), 2015, pp. 130–135.

[10] T. Denoeux, “A k-nearest neighbor classification rule based on Dempster-Shafer theory,” IEEE Transactions on Systems, Man, and Cybernetics, 1995, vol. 25(5), pp. 804-813.

[11] L. Breiman, “Random Forests,” Machine Learning, vol. 45, pp. 5–32, 2001.

[12] T. Chen, C, Guestrin, “XGBoost: A scalable tree boosting system,” Proceedings of the 22nd ACM sigkdd international conference on knowledge discovery and data mining, Aug. 2016, pp. 785–794.

[13] J. Bobadilla, F. Ortega, A. Hernando, J. Bernal. “Acollaborativefiltering approach to mitigate the new user cold start problem,” Knowledge-Based Systems, Feb. 2012, vol. 26, pp. 225–238.