Predicting Online News Popularity
A Comparative Study of Classification Methods

Gregory Hunkins
University of Rochester
ghunkins@u.rochester.edu

1. Introduction

The role of online news in the spread of information in modern society is growing at an explosive rate. In the past year, the percentage of Americans that report that online news is a vital source of news has risen 5 points in the past year from 38% to 43% of the market, continuing a decade-long trend [3]. This trend, importantly, is seen across all age groups and demographics. With a linear continuation of this rate of growth, online news will overtake television, currently 50% of the news market, as the primary news source for Americans in mid-2018 [3]. As such, a vital, open question is what determines the success or failure of an online news article.

This work explores this question by doing comparative analysis between seven popular machine learning methods on a binary classification problem of *popular* or *not popular* on the existing University of California, Irvine (UCI) dataset of article meta-data extracted from 38,000 Mashable articles [2]. A grid-search on a pipeline consisting of preprocessing techniques and model parameters with cross-validation verification revealed a competitive best model of a Neural Network with an accuracy of 67.36%.

These results suggest that this is still an open question, and that the limits of current model-based methods has been reached. As such, a fundamental shift from model tuning to investigating further feature extraction from the articles raw data is proposed as the next step forward.

2. Related Work

Fernandes et. al. first explored this question on the UCI dataset using data science methods by extracting meta-data from an online news source. Their final approach using a Random Forest classifier achieves the state-of-the-art baseline accuracy of 67%.

Many other works have attempted to solve this problem using a wide array of methodologies. Most recently, a state-of-the-art accuracy of 69% was achieved last year by Ren et. al. using a more finely tuned Random Forest classifier [6].

An additionally baseline for this work is considered from Kaggle, a popular online machine learning competition website. The website has hosted three separate private competitions on this dataset. Spanning an aggregated 45 entries, the best model resulted in an accuracy of 64.70%.

3. Dataset

The UCI Online News Popularity dataset [2] is used for this task of predicting viral news based on article meta-data. It contains 61 attributes of which 58 are predictive. An exhaustive overview of the attributes is beyond the scope of this paper, but may be found at the data source [2]. The target attribute of *shares* is binarized at a threshold of 1400 for the *popular* or *not popular* predictive task. In Figure 1 shows the correlation matrix obtained using Pearson's standard correlation coefficient [1]. From here we observe a few very weakly correlated attributes, but mostly see that the data is linearly uncorrelated. For the target *shares* variable, the maximum negative correlation of -0.059 comes from *lda_02* with a maximum positive correlation of 0.110 from *kw_avg_avg*. As such, we see that linear methods may not be successful, but may be used as a baseline.

4. Methodology

Formally, the goal of this task is to solve the binary classification task of predicting *popular* or *not popular* on input data X by approximating function F for such a mapping using popular machine learning techniques.

$$F(X) = \begin{cases} \textit{popular} & \text{if shares} \geq 1400 \\ \textit{not popular} & \text{otherwise} \end{cases} \tag{1}$$

Our methodology to approximate F can be broken into four distinct parts: (1) data cleaning, (2) normalization, (3) feature reduction, and (4) model. The final three sections were aggregated into a Scikit-learn [5] pipeline and a three-fold grid search was applied to determine the best pipeline configuration and hyper-parameters for each investigated model. The primary metric is the cross-validation accuracy on the validation data.

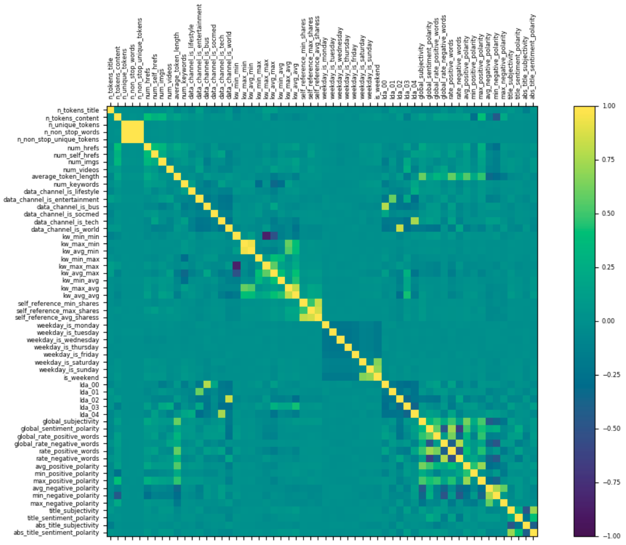

Figure 1: Correlation Matrix for the full UCI Online News Popularity dataset.

4.1. Data Cleaning

The data provided by UCI proved to be extraordinarily clean. No missing values were noted. The non-predictive columns of *url* and *timedelta* were removed from the dataset prior to any modeling. The column of *shares* was separated as well and used as the target variable. In all models except linear regression, this target value was binarized using the criteria of Equation 1.

4.2. Scaling

The scaling options were as follows: standard scaling, robust scaling, or none. Standard scaling normalized the data to a mean of 0 with a standard deviation of 1. Robust scaling, meanwhile, does the same but performs a Interquartile Range scaling to further account for outliers.

4.3. Normalization & Feature Reduction

The normalization and feature reduction options were as follows: Principle Components Analaysis (PCA), Recursive Feature Elimination (RFE), and none.

4.4. Model

Model parameters were specific to each model and are covered in the Experiments section.

4.5. Open-Source

Following the current trend of modern research, the full code used in this analysis is available under the MIT license. This is to increase transparency and enable quick iteration by other researchers and students. It can accessed on Github via the following link: https://github.com/ghunkins/Predicting-Viral-News.

5. Experiments

In the parameter grid-search tables below, the bolded values denote the value used in the best performing cross-validated model. The results for these best models are summarized in Table 8.

5.1. Linear Regression

Table 1 summarizes the configurations of the Pipeline tested for Linear Regression. As such, a total of 30 parameter combinations were tested.

In order to use a Linear Regression as a model for the binary classification task, the model was fit to the original non-binary target share value. A custom accuracy function that binarized the prediction allowed the model to be trained according to the binary classification problem.

Table 1: Linear Regression Cross Validation Configuration

Stage	Value
Scaling	**None**, Standard Scaling, Robust Scaling
Normalization	**None**, PCA(20), PCA(40), PCA(20, whiten), PCA(40, whiten)
Model Configuration	Normalization: [**True**, False]

5.2. Logistic Regression

Table 2 summarizes the configurations of the Pipeline tested for Logistic Regression. As such, a total of 420 parameter combinations were tested.

Table 2: Logistic Regression Cross Validation Configuration

Stage	Value
Scaling	None, **Standard Scaling**, Robust Scaling
Normalization	None, PCA(20), PCA(40), PCA(20, whiten), PCA(40, whiten), **RFE()**, RFE(20)
Model Configuration	C: [10^{-4}, 10^{-3}, $\mathbf{10^{-2}}$, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}, 10^{4}] Penalty: [**l1**, l2]

5.3. Decision Tree

Table 3 summarizes the configurations of the Pipeline tested. As such, a total of 168 combinations were tested.

Table 3: Decision Tree Cross Validation Configuration

Stage	Value
Scaling	None, **Standard Scaling**, Robust Scaling
Normalization	**None**, PCA(20), PCA(40), PCA(20, whiten), PCA(40, whiten), RFE(), RFE(20)
Model Configuration	Criterion: [gini, **entropy**] Max Depth: [∞, 3, **5**, 7, 10]

5.4. Support Vector Machine

Table 4 summarizes the configurations of the Pipeline tested. As such, a total of 8 combinations were tested. A larger grid search was attempted. However, due to time and computational constraints, the grid search was limited to the parameters below.

Table 4: Support Vector Machine Cross Validation Configuration

Stage	Value
Scaling	**Standard Scaling**, Robust Scaling
Normalization	**None**, PCA(20), PCA(40), PCA(20, whiten), PCA(40, whiten)
Model Configuration	Kernel: [**rbf**]

5.5. Random Forest

Table 5 summarizes the configurations of the Pipeline tested. As such, a total of 24 combinations were tested.

Table 5: Random Forest Cross Validation Configuration

Stage	Value
Scaling	**None**, Standard Scaling
Normalization	**None**, PCA(40)
Model Configuration	Estimators: [**10**] Criterion: [gini, **entropy**] Max Depth: [∞, 5, **10**]

5.6. Bagging with Decision Tree

Table 6 summarizes the configurations of the Pipeline tested. As such, a total of 8 combinations were tested. More

Table 6: Bagging Cross Validation Configuration

Stage	Value
Scaling	**None**, Standard Scaling
Normalization	**None**, PCA(40)
Model Configuration	Classifier: [DecisionTree(Max Depth = 5)] Estimators: [10, **20**]

5.7. Neural Network

Table 7 summarizes the configurations of the Pipeline tested. As such, a total of 768 combinations were tested.

Table 7: Neural Network Cross Validation Configuration

Stage	Value
Scaling	**Standard Scaling**
Normalization	**None**
Model Configuration	Optimizer: [**Adam**, RMSProp, SGD] Loss: [**Binary Crossentropy**, MSE] Activation: [**ReLU**, tanh] Hidden Layers: [0, **1**, 2]** Hidden Width: [**58**, 32, 16]** Batch Size: [**64**, 512] Dropout: [0.0%, **20.0%**]

**A select subset of all combinations of the hidden layers and hidden widths were tested.

6. Results

Table 8 summarizes the results of the best models selected via the grid searches across each model type. As such, it can be seen that the Neural Network learned the most generalized mapping from meta-data to *popular* or *not popular* with a competitive accuracy of 67.36%. Linear Regression offers the quickest fitting of the data, while the Random Forest classifier offers the fastest testing of data. Additionally, the Random Forest classifier shows a high accuracy of 98.65% on the training data while not being able to translate that accuracy to the validation data.

The baseline of Kaggle's highest reported accuracy of 64.70% is easily beaten, showcasing the power of grid search and the models tested. However, the state-of-the-art accuracy of 69% from Ren et. al. [6] was not achieved. This is likely due to the parameters chosen for their Random Forest method were not computationally feasible with the given resources in the specified time period.

While small incremental increases in accuracy may be possible with a greater grid search and more models, a fundamentally different approach must be adopted to increase accuracy significantly. As such, text-based feature learning using a Long-Short Term Memory network [4] coupled

Table 8: Results of Best Model Per Model Class

Method	Validation Accuracy (%)	Train Accuracy (%)	Mean Fit Time (s)	Mean Test Time (s)
Linear Regression	56.32	56.63	**0.261**	0.123
Logistic Regression	64.42	65.36	3.156	0.099
Decision Tree	62.69	64.98	0.824	0.017
Support Vector Machine	64.30	71.06	106.519	26.214
Random Forest	60.22	**98.65**	1.604	**0.067**
Bagging	63.55	66.17	3.661	0.120
Neural Network	**67.36**	73.02	N/A	N/A

with meta-data features is suggested as the next logical step in solving this highly difficult problem. As such, a network can be informed by both high level meta-data features and low-level learned features that may more holistically capture an article's writing style. Together, these features may be able to increase accuracy beyond the capabilities of the tuned models already investigated here.

7. Conclusion

In conclusion, a successful comparison of seven machine learning models was undertaken for the task of predicting online article popularity. With the metric of validation accuracy, the Neural Network proved to be the best model available with the experiments performed, giving a cross-validated accuracy of 67.36%. This gave the same maximum accuracy as the original paper from Fernandes et. al. [3], but falls short from the state-of-the-art 69% accuracy from Ren et. al. [6]. A suggested next step is testing an expansion of the given meta-data using LSTM-encoded features from the raw article data for further feature extraction. With further experiments, we can continue to learn the inner mechanisms of viral online news popularity and thereby gain valuable knowledge and insight into this task of ever-growing importance.

References

[1] J. Benesty, J. Chen, Y. Huang, and I. Cohen. Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer, 2009.

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

[3] J. Gottfried and E. Shearer. Americans online news use is closing in on tv news use. *Fact Tank: News in the Numbers*, 2017.

[4] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[6] H. Ren and Q. Yang. Predicting and evaluating the popularity of online news. *Google Scholar*, 2017.