Evaluating the Popularity of Online News Through Machine Learning

Sang Jae Park
park4s@bu.edu

Jiazhou Liu
ljzhou@bu.edu

Paula Hernadez
pandreah@bu.edu

Abstract

With up-to-date technology, people can access online news easily. Most people prefer online news over printed newspapers due to availability and convenience. This aspect gives the chance to analyze news propagation over the web. In this research, we use different learning models such as Neural Network, Decision-Tree Classifier, and Logistic Regression to make predictions about the popularity of online news articles. The Online News Popularity Data Set we used consists of various features extracted from Mashable's online news articles. After training and testing procedures, the Neural Network proves to have the best performance on predictions, as it achieves 0.66 prediction accuracy and 0.67 F-1 score. Besides building prediction models, another goal of our research is to extract features that are highly correlated with popularity of articles. Thus, we split our research into two stages. In the first stage, our models learn from all of the features, while in the second stage they only learn from a subset of them. In order to extract the ideal subset of features, we apply recursive feature elimination with cross validation.

1 Introduction

In the age of Internet, reliance on online news has grown greatly; therefore, it is meaningful for us to analyze what makes a popular article. On the other hand, nowadays many fake news are produced with certain models that boost the popularity of articles, so that our work would also be helpful to distinguish those fake news from faithful ones. In this research, we explore Decision-Tree Classifier, Logistic Regression and Neural Network as models to predict the popularity of online news articles.

2 Related Work

The experiments performed in this project are a result of an effort to improve on the work done by Fernandes et al.[1] In their work, Fernandes et al. make use of five learning techniques, Random Forest, Adaptive Boosting, Support Vector Machine, K-Nearest Neighbor, and Naïve Bayes to predict popularity of online news articles. They curated the $39,797$ instances in the dataset for their research by collecting data from articles originally posted online through the digital media website Mashable.com over a two year period.[2] In order to use the number of shares, one of the features of the data, as a measurement for popularity, the researchers established a threshold to determine the popularity of an article: an article receiving $1,400$ shares or more is considered as popular while an article receiving fewer than $1,400$ is considered as unpopular; it turns this problem into a binary classification task.

[1] Fernandes, Kelwin, et al. "A Proactive Intelligent Decision Support System for Predicting the Popularity of Online News." Progress in Artificial Intelligence Lecture Notes in Computer Science, 2015, pp. 535–546., doi:10.1007/978-3-319-23485-4_53.

[2] Time period: Jan 7, 2013 to Jan 7, 2015

Preprint. Work in progress.

Using a rolling window consisting of $10,000$ training instances and $1,000$ testing instances per iteration, Fernandez et al. find Random Forest to be the most successful model. Fernandes et al. also provide a list of the 16 most important features for the Random Forest prediction.[3]

3 Data set and Data Preprocessing

The Online News Popularity Data Set used in our research was obtained from the UCI Machine Learning Repository. It is also the dataset used by Fernandes et al. in their research, which allows us to compare our methodology and results to theirs.

The dataset consists of $39,797$ instances containing 61 attributes each: 58 predictive features, 2 non-predictive features, and one target. No missing values were found.

Feature	Type (#)
Words	
Number of words in the title	number (1)
Number of words in the article	number (1)
Average word length	number (1)
Rate of non-stop words	ratio (1)
Rate of unique words	ratio (1)
Rate of unique non-stop words	ratio (1)
Links	
Number of links	number (1)
Number of Mashable article links	number (1)
Minimum, average and maximum number of shares of Mashable links	number (3)
Digital Media	
Number of images	number (1)
Number of videos	number (1)
Time	
Day of the week	nominal (1)
Published on a weekend?	bool (1)

Feature	Type (#)
Keywords	
Number of keywords	number (1)
Worst keyword (min./avg./max. shares)	number (3)
Average keyword (min./avg./max. shares)	number (3)
Best keyword (min./avg./max. shares)	number (3)
Article category (Mashable data channel)	nominal (1)
Natural Language Processing	
Closeness to top 5 LDA topics	ratio (5)
Title subjectivity	ratio (1)
Article text subjectivity score and its absolute difference to 0.5	ratio (2)
Title sentiment polarity	ratio (1)
Rate of positive and negative words	ratio (2)
Pos. words rate among non-neutral words	ratio (1)
Neg. words rate among non-neutral words	ratio (1)
Polarity of positive words (min./avg./max.)	ratio (3)
Polarity of negative words (min./avg./max.)	ratio (3)
Article text polarity score and its absolute difference to 0.5	ratio (2)

Target	Type (#)
Number of article Mashable shares	number (1)

Figure 1: List of predictive and goal attributes of dataset.[4]

In order to prepare the data for the learning models, we normalized the input vectors to unit norms. In addition, we discarded instances that have number of shares over two standard deviations from the mean of shares. Through this preocess, we reduced the size of our data set to 39,130 instances.

In consistency with Fernandes et al., we assign a popularity binary classification label to each article, considering any article that obtained 1,400 or more shares to be popular and, conversely, any article that obtained less than 1,400 shares to be unpopular.

From the 39,130 instances we used an rando sample of 35,000 instances. This subset was split into a training set of 30,000 instances and a testing set of 5,000 instances.

4 Aproach

4.1 Decision-Tree Classifier

Decision-Tree Classifier is a non-parametric supervised learning method used for classification and regression. It has advantages in terms of visualization and interpretation, and it handles both numerical and categorical values. Meanwhile, it also has disadvantages compared to other approaches. For instance, a decision tree can become over complex and overfit the training data. In our case, the decision tree generates poor performance in terms of accuracy. However, we are still able to include it in our research as a baseline for other learning models.

[3] After we performed the Feature Selection step in our project, 8 out of the 13 features we identified as most influential for popularity prediction coincided with Fernandes' et al. list of 16 important features for the Random Forest model.

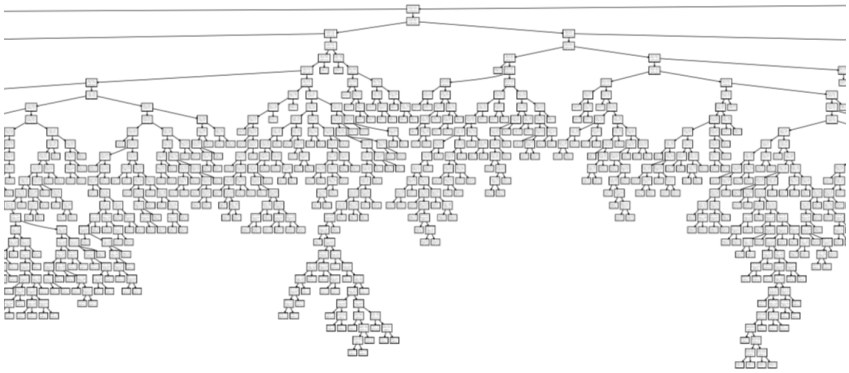

Figure 2: Decision Tree

To evaluate the quality of each split, the criterion we choose is information gain (2) measured by entropy (1) decrease.

$$Entropy(S) \equiv \sum_{i=1}^{c} - p_i \; log_2 \; p_i \tag{1}$$

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \tag{2}$$

4.2 Logistic Regression

Logistic Regression is a model that measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function (3). The outputs of such model represents the probability, that the dependent variable has label equals to 1, given input features w and weights θ. In our case, we measure the cost by finding mean squared error (4). To train our model for better performance, we apply Gradient Descent algorithm and adjust the weights accordingly.

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1+e^T x} \tag{3}$$

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 \tag{4}$$

4.3 Neural Network

Neural Network is a computing system inspired by the biological neural networks in 20th century. Nowadays, it becomes the most widely used learning model for classification and regression tasks. In our research, we build our model using Keras API on top of Tensorflow library. The network consists of 5 hidden layers which are fully connected. In regard to activation function, we use Relu function for the hidden layers, and sigmoid function for the output layer. During training stage, we define the loss function as mean squared error, and we choose Stochastic Gradient Descent (SGD) as optimizer. In order to prevent overfitting, we also add L2 regularization to the network. The formula for updating weights in SGD is provided below (5).

$$w_{t+1} = w_t - \gamma_t \nabla_w Q(z_t, w_t) \tag{5}$$

5 Feature Selection

A feature selection step, in which 13 features were chosen out of the original 58 features, was taken to improve the predictive potential of our models. Table 1 shows those features, ordering them according to their correlations with the output feature. In our research, we decide to apply RFECV techniques implemented in scikit-learn library. Recursive feature elimination (RFE) is a feature selection method that fits a model and removes the weakest feature until the specified number of features is reached; it always requires a definite number of remaining features. Nevertheless, in many cases, that number is unknown in advance. To find the optimal number of features, Cross-validation (CV) is often used with RFE ,in order to score different feature subsets and select the best scoring collection of features.

Table 1: Correlation

	Features	PCC
0	Number of words in title	0.016958
1	Average word length	0.012142
2	Average negative polarity	0.008783
3	Top 1 relevan topic	0.000806
4	Top 3 relevant topic	-0.003793
5	Top 2 relevant topic	-0.005183
6	Entertainment channel	-0.010183
7	Top5 relevant topic	-0.013253
8	Tech channel	-0.016622
9	Rate negative words	-0.017006
10	Day published is weekend	-0.022007
11	Avg positive polarity	-0.032029
12	Number of unique words	-0.059163

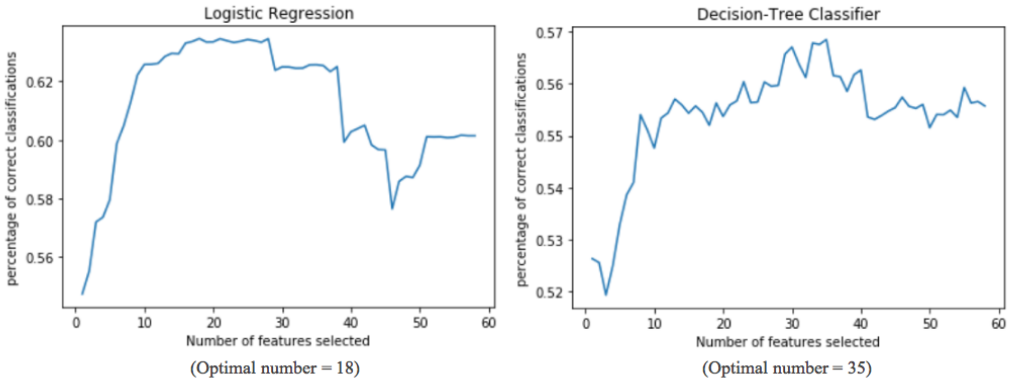

Figure 3: RFECV for Logistic Regression and Decision-Tree Classifier

6 Experiments and Results

Initially, all three models were trained using all 58 predictive features as their input. The Neural Network was trained through 100 epochs with batch size of 32 to obtain an accuracy measure of 66% and an F-1 score of 0.67. The ROC graph can be seen in Figure (4) as well as the AUC metric. Table 2 shows that the Neural Network outperformed both the Decision-Tree Classifier and the Logistic Regression.[5]

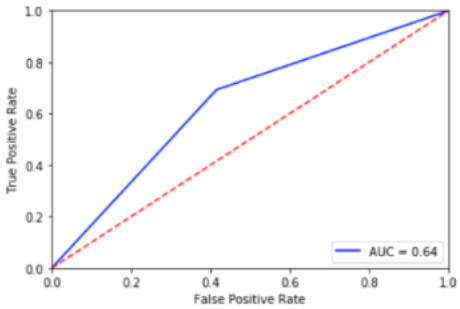

Figure 4: ROC curve of Neural Network (accuracy = 0.66)

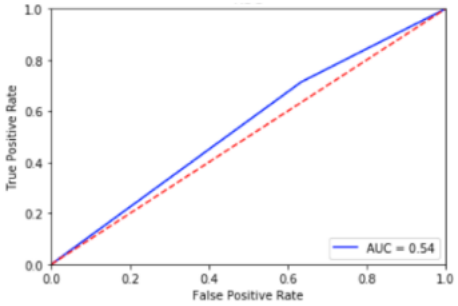

Figure 6: ROC curve of Decision Tree (accuracy = 0.53)

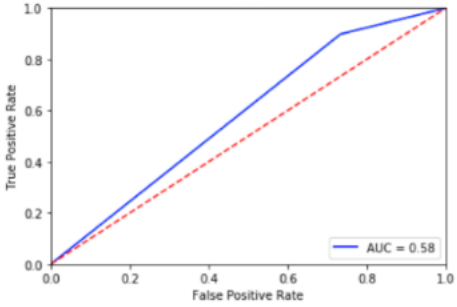

Figure 5: ROC curve of Logistic Regression (accuracy = 0.56)

Figure 4: ROC curves for all models

Table 2: 58 Features

Model	Accuracy	Precision	Recall	F-1 Score
Random Forest	0.67	0.67	0.71	0.69
Adaptive Boosting	0.66	0.68	0.67	0.67
SVM	0.66	0.67	0.68	0.68
K-NN	0.62	0.66	0.55	0.60
Naive Bayes	0.62	0.68	0.49	0.57
Neural Netwrok	0.66	0.68	0.67	0.67
Decision-Tree	0.53	0.55	0.55	0.55
Logistic Regression	0.56	0.55	0.90	0.68

After Feature Selection took place and a subset of the features were selected, all the models were run again with only 13 features as their inputs. In this case, there was an improvement in the prediction accuracy of both the Decision-Tree Classifier and the Logistic Regression, as their accuracy levels reached 55% and 63% respectively. On the other hand, the accuracy of the Neural Network decreased to 62%. More performance metrics for the 13 feature trials can be found in Table 3.

[5]The first five rows of Table 2 display the results obtained by Fernandes et al. for comparison

Table 3: 13 Features

Model	Accuracy	Precision	Recall	F-1 Score
Neural Network	0.62	0.64	0.66	0.66
Decision-Tree	0.55	0.56	0.71	0.63
Logistic Regression	0.63	0.62	0.77	0.69

We intuit that feature selection does not yield better results than the use of the 58 original variables in the Neural Network due to the fact that Neural Network handles the case itself, by giving those unrelated inputs low weights. Meanwhile, we risk omitting features that are given nonzero weights and compromise the performance of the model.

7 Conclusion

Through this research we obtained prediction models that are comparable with those of previous researchers in terms of performance. Among three models, the Neural Network has best prediction accuracy, which is about 66%. On the other hand, we realized that not all the features are equally important, and feature selection can significantly improve the efficiency and accuracy of our models. For future work, if we are able to access more data, such as the most frequently used keywords in popular articles, we could potentially build an intelligent system that gives suggestions to the authors, and help them write articles which are appealing to the general audience.

References

[1] Bottou, Léon. "Large-scale machine learning with stochastic gradient descent," in Proceedings of COMPSTAT'2010, pp. 177–186, Springer, 2010.

[2] Fernandes, Kelwin, et al. "A Proactive Intelligent Decision Support System for Predicting the Popularity of Online News." Progress in Artificial Intelligence Lecture Notes in Computer Science, 2015, pp. 535–546., doi:10.1007/978-3-319-23485-4_53.

[3] Girosi, F., Jones, M., and Poggio, T. (1995). Regularization Theory and Neural Networks Architectures.Neural Computation, 7(2):219–269.

[4] Guyon I., Elisseeff A. (2006) An Introduction to Feature Extraction. In: Guyon I., Nikravesh M., Gunn S., Zadeh L.A. (eds) Feature Extraction. Studies in Fuzziness and Soft Computing, vol 207. Springer, Berlin, Heidelberg

[5] Kleinbaum, David G., et al. Logistic Regression: a Self-Learning Text. Springer, 2010.

[6] Mitchell, T. "Chapter 3: Decision Tree Learning." Machine Learning, McGraw Hill, 1997

[7] Wackerly, Dennis D., et al. Mathematical Statistics with Applications. 7th ed., Brooks/Cole, 2012.