Online News Popularity Prediction
– CS 760 Project Report

Team members: *Luwan Zhang, Song Wang*

Abstract

Online news is becoming a major daily tool for younger generations to connect to the world. Predicting such popularity is also a booming research topic that attracts many machine learning experts around the globe. In this project, we use a unique online news dataset to predict the popularity of each online article. In particular, we examine the performance of several modern state-of-the-art machine learning algorithms. K-Nearest-Neighbor works out the best under the regression setting, while Random Forest wins under the classification setting. Furthermore, we identify some important features that play a significant role in contributing to the prediction accuracy.

1 Introduction

Within the expansion of the Internet, there has been a growing interest in online news, which allows an easy and fast spread of information around the globe. Thus, predicting the popularity of online news is becoming a recent research trend. Popularity can be often measured by considering the number of interactions in the web and social networks, to name a few, number of shares, likes, comments, number of clicks etc. Predicting such popularity is valuable for bloggers, content providers, advertisers,and even politicians or activists, so as to better understand readers' demands and cater to public opinions.

In this project, our primary interest is to predict the popularity of an article before its initial release based on an online news popularity dataset obtained from UCI Machine Learning Repository(dataset access) thanks to the donation from Dr. Kelwin Fernandes. Specifically, our tasks have three main facets:

(1) Predict the number of shares– a regression task

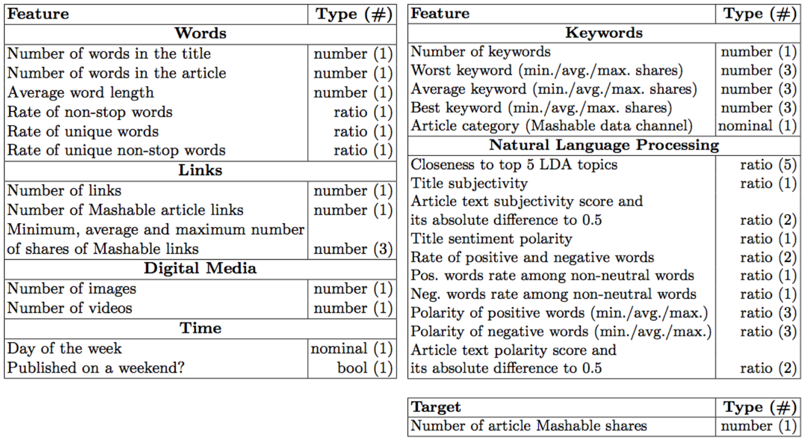
(2) Predict whether an article will be popular/appealing or not after its publication– a binary classification task

(3) Identify significant factors/features that can increase the probability of an article being popular– a feature selection task

2 Data Description

The dataset contains information about 39797 articles published by Mashable (www.mashable.com) in a two-year period, from January 7 2013 to January 7 2015. The response of interest is the number of shares for each article, which indicates the level of its overall popularity across different communities. The data also comes with 61 features, either numerically or categorically. Some of these features should be expected as important factors contributing to the popularity of an article.

A more detailed explanation on these features can be referred to Figure **??**. All features have been classified into 3 types: number–an integer in this context, ratio– a float number within the range of $[0, 1]$, boolean variable, and nominal variable transformed with the usual *1-of-C* encoding. For any further details, one can refer to [**?**].



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 features by category.

3 Regression task

In this section, we aim to predict the number of shares for each article. Given the response is integer-based, to allow for the validity of most existing predicting models, we transform the response variable into its logarithm, after which the normality assumption looks to be more appropriate. In addition, all numerical features have been standardized, in order to prevent the original scale obscuring the importance of themselves. We started with a linear model, then moved to explore a non-linear model, because the R^2 is too low to support the sufficiency of a linear model. To avoid overfitting, we also randomly hold out a subset of 9644 instances as the test set. Table **??** shows the comparison in terms of MSE performance between a linear model and several non-linear models. For each method in Table **??**, the k-nearest-neighbor method is using kd-tree algorithm, and the neural-network is based on 5 epochs, 3 hidden layers with 200, 100, 50 units sequentially, and two activation functions Tanh and Rectifier respectively. Surprisingly, the neural-network did not perform significantly better as expected. On the other hand, kNN(k=10) outperforms the others.

Model	Training set MSE	Test set MSE
Linear regression	0.76	0.75
kNN(k=5)	0.32	0.32
kNN(k=10)	0.25	0.31
Neural-network(Tanh)	0.62	0.77
Neural-network(Rectifier)	0.66	0.74

Table 1: MSE performance comparison to predict the number of shares.

4 Binary classification task

From an editor point of view, it often makes more sense to develop a strategy to decide whether an article would be popular after publication rather than simply get an accurate estimate of total number of shares. This special concern motivates us to make the original numerical response variable coated with a binary representation, in which 1 means the article is popular and 0 means not popular. In particular, we use the median value as the threshold. Our goal is to predict the popularity status for each article.

4.1 Evaluating prediction accuracy

In order to investigate the performance of different algorithms in a comprehensive manner, we adopt multiple criterion shown in Table **??**. For consistency, we still stick to the test-set randomly held out in the last section. Among the 7 methods listed in Table **??**, Random Forest with 1000 trees outperforms the others. The plain Logistic regression and its Lasso version almost have the same behavior, which indicates in this case the penalty term is not necessary. It is also worth noting that kNN does not perform as well as in the regression setting. We've tried $k = 3, 5, 10$ respectively, unfortunately their performances show little difference. Figure **??** plots the ROC curve for each method.

Table 2: Popularity prediction performance comparison on the hold-out test set using median as the threshold

Model	accuracy	precision	recall	F_1 score	AUC
Logistic regression	0.65	0.65	0.63	0.64	0.70
GLM_Lasso	0.65	0.65	0.63	0.64	0.70
Random Forest	0.67	0.67	0.67	0.67	0.73
Tree	0.65	0.65	0.64	0.64	0.68
AdaBoost	0.66	0.65	0.64	0.65	0.71
kNN(k=5)	0.61	0.61	0.58	0.60	0.64
Neural-netowrk(Rectifier)	0.65	0.65	0.63	0.64	0.71

4.2 Effect of training size

To further investigate the effect of training size on the learning accuracy, we run each algorithm listed in Table **??** using the training set with sample size = 1000, 2000, 5000, 10000, 20000,and 30000 respectively. The test-set accuracy can be referred to Figure **??**. From the plot we can see, the increase momentum tends to get mild as the training size approaches to 10000. More specifically, most classifiers get significantly improved in terms of learning accuracy when the sample size reaches at 10000. However, different algorithms seem to meet the bottleneck at different rates. For this type of data, Logistic regression and ensemble methods could be able to attain a reasonably good performance with a relatively low sample size, while kNN and a single decision tree method require more samples in order to get a comparable performance.

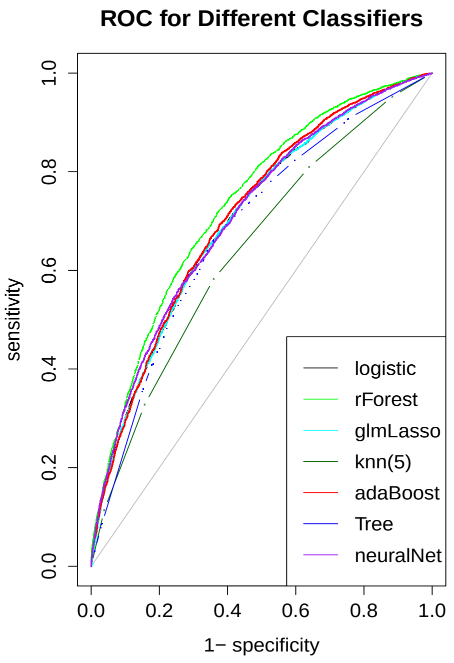


Figure 2: ROC curve for each method based on Table ??

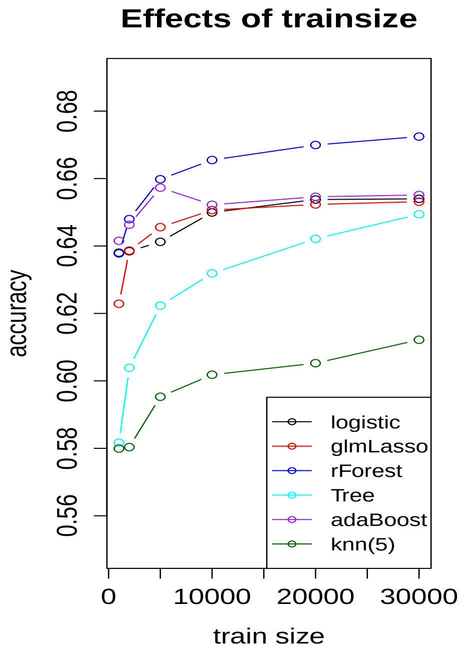


Figure 3: Effect of training size on the test-set accuracy

5 Variable importance ranking

With the influx of available features, it is both economically beneficial and computationally efficient to select most relevant features into the pipeline for a further learning task. The definition of relevance may vary as the shift of one's primary interest. In this section, we are particularly interested in features that can contribute more to improve prediction accuracy. Table **??** shows the ranking result based on the classification task using Random Forest with 1000 trees and 5-fold cross validation.

6 Discussions and future work

In this project, our main goal is to predict the popularity of online news using a public UCI data set. To achieve a desirable prediction accuracy, we've investigated several state-of-the-art machine learning algorithms. We have also studied the effect of training size on the learning accuracy for each examined algorithm. Besides, we also tried several approaches to combine the predic-

Rank	Feature name	importance percentage
1	Keyword-related	0.19
2	Article polarity	0.18
3	Closeness to LDA topics	0.11
4	Title-related	0.10
5	Self-reference	0.08
6	Article sentiment	0.06
7	Article channel	0.05
8	Weekend/weekdays	0.03
8	Embeded videos/images	0.03
10	# of words in content	0.02

Table 3: Variable importance ranking based on the classification task using Random Forest with 1000 trees and 5-fold cross validation.

tion results given by the classifiers we've already built, but unfortunately ended with no significant improvement.

However, since our experiments were so far only executed in the old-fashioned batch mode, our next step is to take the advantage of the active learning along our learning process. We believe this exploration would provide us with more useful insights to build a good classifier when only a small amount of labelled data is available. We are also very interested in researching more clever and efficient ways to combine multiple classifiers so as to get a significantly better classifier other than directly adopting the traditional majority vote concept.

References

[1] K. Fernandes, P. Vinagre and P. Cortez (2015). A Proactive Intelligent Decision Support System for Predicting the Popularity of Online News. *Proceedings of the 17th EPIA- Portuguese Conference on Artificial Intelligence, September, Coimbra, Portugal.*

[2] Blei, D.M., Ng, A.Y., Jordan, M.I(2003). Latent dirichlet allocation. *Journal of Machine Learning Research* 3, 9931022.

[3] Bandari, Roja, Sitaram Asur, and Bernardo A. Huberman(2012). The Pulse of News in Social Media: Forecasting Popularity. *ICWSM.*