Predicting the popularity of articles posted on Mashable

Vansh Chandwaney | October 31, 2022

Executive summary

The objective of this report is to analyse different models used to predict the total shares of an article posted by Mashable. This paper will first contextualise the aim of the research and explain the current business scenario. Then, the dataset available is explored and some changes made to this for analysis are highlighted. Subsequently, the models used – Random Forest and Logistic Regression – are explained. They are also analysed in detail and compared with each other. Finally, Random Forest is recommended to be used for this case, as it most accurately identifies articles that have true scope in generating a high number of shares.

Contextualising the Research

At Mashable, there is currently a need to automate the process of selecting the articles written by contributors that get posted to the website. Keeping a revenue-maximising objective in mind, articles that have the potential to get a higher number of shares should be prioritized. In the current scenario, Mashable earns an estimated $0.75 in revenue per share when an article is shared 1000 times or less, and $2 for every share after that. In this report, two prediction models will be discussed that will largely eliminate the need to select contributions manually, by selecting articles that will likely be shared more.

Description of Data

The dataset used in this analysis has details of 39645 articles posted on Mashable. Broadly, this dataset contained information about keywords used, data channels, the day of posting, hyperlinks, the language used, overall sentiment and the total shares for each article.

There were 60 possible metrics in the dataset that could have been used to predict shares. However, using too many variables for predictive analysis could lead to a model that is overly complex and prone to overfitting, which is when a model performs well when trained, but can't handle new, unseen data. Thus, there was a need to select the most important metrics, or features, in the dataset. For the predictive models, two different sets of variables were used.

- In the first set, I removed features by hand based on my knowledge of digital media, after which there were 46 predictor variables left. The features that were removed included minimum and maximum shares for the best, worst and average keywords and minimum and maximum polarities of the positive and negative keywords. For the purpose of this analysis, taking average values for these instead of minimums and maximums is sufficient.
- In the second set, I used XGBoost, which is an algorithm that provides an importance score for each feature based on its contribution to predicting the target variable. I selected variables that contributed 5% or greater and ended up with 9.

Analysis of findings

Initially, I attempted to predict shares using multiple linear regression, which is a modelling technique to identify relationships between multiple variables and a target variable (shares, in this case). However, a strong correlation with shares could not be found even after trying out several combinations of variables. Owing to the intricate nature of digital media algorithms, it becomes challenging to reliably forecast article shares even when data on such diverse metrics are available.

For this reason, I used two classification models for this analysis - Logistic Regression and Random Forest. If the articles in the dataset had total shares over 1000, they would be classified as ‘popular,’ and ‘unpopular’ if the shares were 1000 or less. The threshold of success was 100, as each article share over this amount generated significantly greater revenue.

Logistic regression simply estimates the probability of an event occurring (in this case, shares being over 1000) based on the predictor variables. This model was used as it is less sensitive to outliers, or unusually high or low values than other variables, compared to some other models. Random Forest uses multiple decision trees and merges them to obtain an accurate prediction. I used Random Forest as it is effectively able to capture non-linear relationships between predictor variables and the target variable.

My primary objective with these two models was to obtain a good **sensitivity** value. Sensitivity is a model’s ability to correctly predict a ‘success’, or popular article in this case. In this context,

maximizing sensitivity is important as we want to ensure that popular articles can be accurately identified. The other 2 metrics assessed were **specificity** and **accuracy.** Specificity is the model's ability to correctly identify a failure, and accuracy is the fraction of the model's correct predictions.

Model <chr>	Accuracy <dbl>	Sensitivity <dbl>	Specificity <dbl>
Naive Bayes	0.3381	0.9581	0.0624
Logistic Reg	0.7062	0.2627	0.9034
Random Forest	0.7182	0.9214	0.2627

Figure 1

Model <chr>	Accuracy <dbl>	Sensitivity <dbl>	Specificity <dbl>
Naive Bayes	0.5523	0.6431	0.51070
Logistic Reg	0.6862	0.0383	0.98286
Random Forest	0.6972	0.9049	0.24650

Figure 2

Figure 1 presents results from the model developed using 46 predictor variables, and Figure 2 presents results from the model using 9 variables. These values were obtained by running the model on the test dataset, which had distinct observations from the dataset used to train the model. This is to ensure the trained model fares well with new data. I have benchmarked the two models with a Naive Bayes model, which is also popular for classification tasks. In both comparisons, the Random Forest and the Logistic Regression models outperform Naive Bayes in terms of accuracy. I will focus on the model with 9 variables, as this simpler model will be less prone to overfitting.

As seen in Figure 2, Logistic Regression has a poor sensitivity score and is only able to correctly classify true positives in 3.83% of cases. Thus, despite it having an excellent specificity value, it is not recommended to be used. Random Forest outperforms the other 2 models in terms of accuracy, and more importantly, sensitivity. In context, a sensitivity of 0.9049 means that **an estimated 90.49% of the articles that are genuinely popular are correctly classified as popular by the model.** Thus, the model likely won't miss out on many popular articles, and Mashable will be able to benefit from the higher amount of revenue generated for each additional share**.**

Conclusion and recommendations

The objective of this analysis was to automate the process of selecting and rejecting articles based on their expected popularity. Upon testing different models, the Random Forest model was found to be an effective one for this task, owing to its ability to correctly classify popular articles. Thus, I recommend it to be used to select articles.