Data Science 316 A1 Project

Sebastien Meniere, Jaunré Lastname

1. Introduction and Problem Statement

Online news popularity prediction is a well-developed research field, with the aim to model the underlying mechanisms influencing how digital media is distributed through online social networks and, to make predictions about news article popularity prior to publication.

Existing literature measures popularity through likes, shares, and views. Reliably predicting the popularity of a news article before it is published is of great importance to the news agencies / providers. Allowing for data driven article optimization for improved popularity, competitive advantages over other publishers, higher user engagement with articles, all of which could lead to greater industry success.

Furthermore, an acute understanding of what news will be popular is valuable to many other sectors, including, consumer markets, political affair, marketing, and entertainment. This is because popular / trending news can have a strong influence on the public’s opinions, interests, and decision making.

The difficulty in creating a robust predictive model in this context, is that there are many unknown, and unmeasured variables in the physical world that will influence which articles become popular. These could be, current political affairs, fashion trends, consumer fads. These factors are difficult to incorporate into training data sets, as they are time / period specific, and the information captured might not generalize to future events. Some of these confounding variables are not knowable before publication.

The news affects how people act, and the way in which people act makes the news. This rapid, and constant feedback loop is increasingly difficult to model as people consume more media.

In this project we will be investigating and evaluating the predictive classification models presented in the research of Fernande et. Al. We will also attempt to improve upon the model’s classification performance, and attempt to construct a high preforming and highly interpretable model.

ADD REFFS HERE 🡨🡪

2. Data Description

We acquired the data set from the UCI machine learning repository. The authors collected 39644 articles published by the reputable news organisation Mashable, over a two year time period. Each article was processed as to summarise some important characteristic. In total there are 61 features, that stem from 6 categories.

|  |  |
| --- | --- |
| Category | Feature examples |
| WORD | Number of words in article / title, Average word length, Rate of unique / non-stop words |
| Links | Number of links, Number of Mashable article links |
| Digital Media | Number of images / images |
| Time | Day of the week, Published on weekend |
| Keywords | Number of keywords, Article category |
| NLP | Title subjectivity, Article text subjectivity, Title sentiment polarity |
| Target | Number of shares |

All of the variables fall into two types, number and ratio. Where the ratio variables are generally continuous between 0 and 1. And number type variables are either floats or integers. There are no missing values in the data set. The target variable is an integer type, ranging from zero to eight hundred thousand. This means that it will have to be partitioned into bins such that we can use classification models for prediction. We will discuss the partitioning threshold and its implications later.

We must also point out that there are some variables that will be omitted from model training process as they do contain any useful information, for example to url to the article page is non-informative.

We did observe several outliers, these were observations where the number of shares was far larger than any reasonable amount of standard deviations from the mean.

3. The Current Approach

The paper investigates 5 different classification models, which were all trained using the same procedure. The list of models is: Random Forest (RF), Naïve Bayes (NB), Adaptive Boosting (AdaBoost), SVM with a radial basis function (RBF) kernel, and K-Nearest Neighbors (KNN). The authors did not explicitly explain why each of the above models were chosen, as opposed to others, however, given the context and aim of the paper these models were selected as they represent the cutting edge / state of the art in classification.

Moreover, the aim was to identify which model would perform the best for the given task, and this selection of models is a good representation of the classification space.

In order for classification models to be applied into the dataset, the continuous numeric target variables, shares, had to be transformed into a categorical variable. The authors selected the threshold value to be 1400, meaning values greater than or larger than this value were classified as popular and values smaller were classified as unpopular.

Many of these models require a hyperparameter, which is a parameter that must be chosen before the model can be trained. In the case of K-Nearest Neighbors classifier the hyperparameter is the number of neighbors that will be considered for each data point. For AdaBoost and Random Forest, the hyperparameter is the number of trees. The authors fine-tuned / optimized these parameters using a grid search. Which is the process of iterating over a set of predefined hyperparameter values, training the model with each element of the set, and then testing the performance of that model. The result of this is finding the value of the hyperparameter that gives the best performance. Without this step, the authors might have arbitrarily chosen the values of the hyperparameters which could have led to suboptimal, poor preforming models.

The authors also demonstrate good practices for model training by using disjoint testing and training data sets. They decided to employ a commonly used split of 70% training data and 30% testing data. In which the entire data set is randomly partitioned into two sets, with the above ratios. The validation set can be used to understand the model’s performance on unseen data, which will be the case when then model is deployed into the real world. Additionally, comparing the model’s predictive performance on the training data versus the testing data can reveal whether the model tends

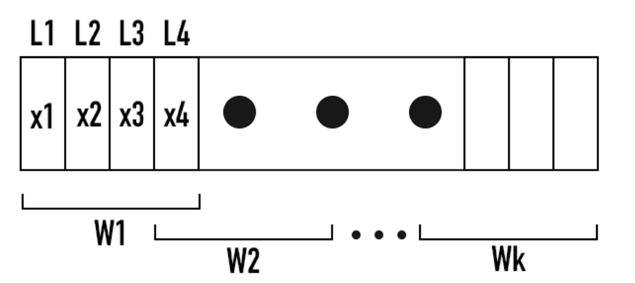
to overfit the training data, implying a poor choice of hyperparameters or too much flexibility in the model.

*Test-Train Data Split*



For robust evaluation of the trained models, the authors used a rolling window analysis. This approach uses a training window of W consecutive datapoints, which is used to fit the model. Then L predictions are made, then the training window is updated by replacing the L oldest samples with L more recent ones. A new model is fit using the new window, and the process is repeated until the end of the data set is reached. A window size of W = 10000 and a prediction size of L = 1000 was chosen. Motivation for these specific values was not discussed.

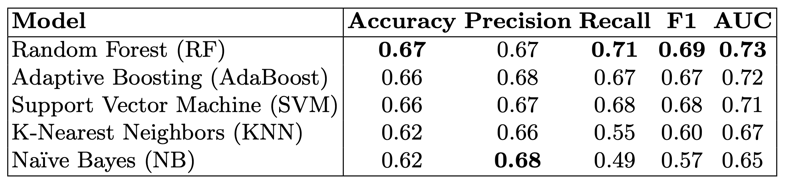
*Rolling Windows Analysis*



We note that the authors did not perform any feature selection, and all starting features from the original data set were used in the training of each model.

After the models were trained several performance metrics were calculated for each of models to evaluate their performance. The AUC, F1, precision, accuracy and recall. The results obtained are listed in the below table.

*Model Evaluation*



We note that the metrics: accuracy, precision, recall, and F1 were all calculated with a probability threshold value of 0.5. Meaning that observations were classified as belonging to class 0 if the predicted probability was below 0.5. Whereas the AUC calculation is irrespective of this threshold value, as is the area under the ROC curve which iterates over the interval [0, 1].

The authors award RF as the highest overall performing model.

4. Potential Problems

Overall, we believe the author’s methodology to be very strong. Starting from the beginning by transforming the target into a balanced categorical variable, using testing a validation sets to mitigate overfitting and evaluate model performance, tuning model hyperparameters through grid searching, using rolling windows for robust analysis, and finally calculating meaning metrics to measure each model’s performance.

Problems:

1. The value selected to be the threshold, did result in a balanced binary classification space. But the fact that there are only two classes limits the model predictive capabilities. Since the range of the target variable is very wide, simply partitioning at this value greatly reduces the information that can be extracted from the prediction. The task might be better suited for regression models. Furthermore, the binary nature of the new target excludes some other powerful models, like Linear discriminant analysis and multiclass logistic regression. Further exploration into different bin sizes could be used.
2. The authors also did not use any dimensionality reduction techniques, nor did they use any feature selection techniques. This is not always a problem but, seeing as the data set has nearly 60 features, there could easily exist some redundant features that could be adding unnecessary noise to the final models.
3. Rolling windows analysis might have been overkill. It has to do with time series which the data set is, but they took the union of all the predictions, and only trained models on small subset windows. This might have resulted in many poor models. Rather we should explore the results of these models without the rolling windows. However since is time series related and since this is news, and the target is popularity, we know that whats popular now will change, and so this will change the attributes that were popular. So we should experiment with rolling windows with larger windows, and smaller windows. Also there’s no trend with time, so the rolling windows analysis might be useless, k-fold cross validation could be a better solution. If there was a trend
4. Grid search values, preforming a grid search is a tedious and computationally intensive practice. For this reason, it’s understandable that the authors only selected 6 values in their grid search sets. But we believe that they could have done a better job at fine tuning the hyperparameters. Maybe using grid sets with more values, and then doubling down and zooming in to really find the best values.
5. Eliminating outliers. The authors did not remove any outliers from the data. This could be a pretty significant problem as there exist a substantial number of points that exist far beyond any reasonable number of standard deviations from the mean. The presence of these outliers could be disproportionately skewing the data.
6. Grid search RF and AdaBoost, tuning for max depth.
7. Preforming pca before random forest for potencially better results
8. Tuning more of the hyperparameters in the svm.

Really we should boil it down to, better preprocessing of the data, this includes

* Removing outliers
* Potentially using more bins
* Potentially transforming the data with pca
* For models that don’t do automoatic feature selection we should do feature selection
* Fine tune ALL hyperparameters for ALL models, not just some hyperparameters (bigger grid search)
* Get standard error estimates for the predictions
* Use a different method than rolling window analysis
* Do dim reduction for KNN

The goal is really to improve on each of the models. And then to show whats the best. Because the above steps could boost the performance of all each of the models separately.

Also I think we should scrap KNN and replace it with bagging or smthing, because KNN is kinda ass, unless we do dim reduction.