Statistical Learning - Written Report

James Hahn

Introduction

Mashable (`https://mashable.com/about/`) is a global media and entertainment platform with 45 million unique visitors per month, 28 million social media followers, and 7.5 million article shares per month. With its plethora of content and access to a variety of news, the articles appeal to a variety of individuals, companies, and industries encompassing entertainment, culture, tech, science, and social good. Naturally, being a news agency, they seek to achieve success in two areas: news popularity and advertisements revenue. Both are connected since the number of article shares dictates news popularity and it increases click rate on articles, allowing more consumers to view the advertisements tied to that article. Thus, article shares dictate news popularity and advertisement revenue (`https://mashable.com/advertise/`). Mashable is not the only news agency with this simple business scheme. Other papers, such as New York Times and Washington Post, and even search engines, such as Google and Bing, strive to increase click rate on their content to drive ad revenue, which allows the company to be successful through hiring top talent, building state-of-the-art facilities, and even advertising themselves. As such, when the Online News Popularity Dataset (`https://archive.ics.uci.edu/ml/datasets/Online+News+Popularity`) was gathered through web scraping metadata about Mashable online news articles and a goal to predict shares, the widespread applicability of the problem across a vast number of companies instantly became apparent.

The dataset itself is thorough, but concise. There are 39,797 unique article samples. In conjunction with these articles, a series of 61 attributes were scraped. Of these 61 attributes, two are non-predictive (article url and time between article publication and dataset acquisition), one is the response (article shares), and the remaining 58 attributes are predictors. With some preliminary analysis of scanning over each column for each sample, there was no missing data found, making the preprocessing task easier. The dataset's structure is best described as heterogenous; it contains both qualitative and quantitative predictors. Specifically, 44 of the predictors are quantitative and the other 14 predictors are qualitative. Included in the qualitative predictors are attributes such as the content classification (binary variables for `lifestyle`, `entertainment`, `business`, `social media`, `tech`, and `world`) and weekday classification (7 binary weekday predictors and 1 binary `is_weekend` predidctor). On the quantitative side, sentiment analysis was performed on the document content, article title, keyword analysis, number of images, and number of specific, individual article metadata

in general. These attributes provide a well-rounded picture of the article's content without sharing article contents and breaking Mashable's content rights restrictions. With these attributes, one may make several conclusions about a given article, such as its positivity, its title's clickbaitness, mixture of media (text and images), article length, and spread of content (videos and hyperlinks).

This dataset is useful for both regression, classification, and statistical inference. The regression task is simple and straightforward since the response, article shares, is continuous. Are there a mixture of article features capable of predicting an article's number of shares? The classification, although somewhat straightfoward, requires modification to the response. This task involves binning the response through use of different splitting techniques (standard deviations above/below the mean and percentiles of article shares) to convert it into a binary or trinary classification task. For the binary task, one may classify articles as popular/non-popular or viral/non-viral. The trinary task deals with ambiguity in popularity since the articles toward the boundary of being viral/non-viral may be impacted through outside factors unaccounted for in this dataset, such as specific words in the article. As such, the three-way task has classes for an article being popular/viral, non-popular/non-viral, and neither. This can be applied in real-world settings through several avenues. First, a news agency strives on producing viral content since it brings in top talent wanting to work for the largest agencies. Second, the more viral content an agency generates, the more the name becomes a household name, such as how the New York Times is today. Third, as discussed earlier, more article shares leads to more click rates and more consumers viewing ads, thus boosting ad revenue. As such, the results discussed may produce advanced statistics to forecast the overall health of a news agency. Finally, with the added statistical inference, before an article is produced, diagnostics of the article can be produced to provide suggestions on avenues to increase the article's potential number of shares. So, this dataset and the regression and classification tasks provide a solid foundation for impacting entire industries and boosting companies' revenue and popularity.

Methods Overview

The methods involved in this project are vast and explorative, investigating the utilized features, size, complexity, and applicability of each model. The first of which includes feature selection. Since the dataset contains many predictors, we are given more freedom to experiment with producing successful models with high accuracy rates. Additionally, the large dataset does not force the constraint that we must limit the number of predictors we utilize in the models. However, with so many predictors, the process of statistical inference is limited. A model utilizing 58 predictors is hard to analyze since some predictors may be more useful than others or some features are heavily connected to other features, introducing redundency into

the model and possible multicollinearity. As such, since statistical inference is important in this project, limiting the number of features to highly valued predictors is important. In terms of feature selection, forward and backward selection will be explored for both regression and classification tasks. Best subset selection is not feasible since there are `58!` possible models. With forward and backward selection, there are only $\sum_{i=1}^{58} i = \frac{58(59)}{2} = 1711$ models we must evaluate. Although PCA and PLS are useful techniques for dimensionality reduction, they are more suitable to situations with fewer dataset samples since they reduce statistical inference. In fact, they do not even reduce the number of original features used in the models since the new features (which are projections) are combinations of the old features. Additionally, as another form of feature selection, lasso regression is utilized for the regression task to find useful features. Finally, to wrap up the feature selection methods, BIC, AIC, adjusted R^2, and C_p are analyzed for the generated models to give more favor to less complex models, thus boosting statistical inference.

Second, the `shares` response is continuous, so the regression task is apparent, but minor adjustments can easily convert it into a categorical response. Therefore, linear and polynomial multiple regression, GAMs, and average-voting kNN were explored for the regression task. In class, we discussed a plethora of classification models, specifically majority-voting kNN, logistic regression, LDA, and QDA. Local regression is not necessary since local regression is a highly flexible method. We have enough data to test it so the variance will not be too too high, but why test local regression when our large dataset is capable of modelling complex relationships with multiple polynomial regression or GAMs with small fear of overfitting? Local regression is simply just a glorified local method with high computational cost. For example, local regression requires fitting a model to a small neighborhood of data points around the data point we are trying to predict. News agencies output anywhere from tens to hundreds of articles per day and they desire accurate results, requiring large neighborhoods of points in the thousands of samples. So, this increases real-world computational costs significantly and makes that path nonviable for several reasons. In terms of classification, logistic regression, LDA, and QDA produce nice models capable of statistical inference since they output probabilities of a sample leaning toward one class or the other. Unfortunately, these models are designed for binary classification by nature, so one vs. rest classification is utilized for three-way classification, comparing the probability of a new data point being in one class compared with the other two classes.

Finally, simple preprocessing techniques can be used. For example, as mentioned earlier, the classification task can be split up several different ways to convert the continuous response into a categorical response. The first is by taking the mean `shares` and classifying all samples one standard deviation above the mean as popular and the rest as non-popular. Two standard deviations can also be used and will be used, but produces class imbalance which may be a concern. Other avenues, dealing with mean's issue of lack of robustness to

outliers, is to split the dataset by percentiles of `shares`. For example, we use two splits at the 75th and 90th percentile, where `viral` articles are defined as being above the 75th percentile and above the 90th percentile. Keep in mind these are two different splits; they are not both used in the same model simulations. This produces a useful picture of article virality since viral articles are generally defined as having many more shares than the average article. Finally, to overcome the potential class imbalance issues, a split was made at the 50th percentile, ensuring half the dataset is in one class and half the dataset is in the other class. These splits are analogous for the three-way classification task, producing splits at the 33rd/66th percentile and 20th/80th percentiles. In addition to these preprocessing steps on the response, several statistical procedures were utilized to reduce variance in model accuracies to ensure high quality results. The first of which is cross-validation, which we use 10-fold cross-validation across all models. Second, although PCA/PLS were mentioned earlier as not being useful, they do have one use case depending on which questions news agencies want to answer. For example, if a news agency only wants to classify an article as popular or non-popular, kNN is useful. If the news agency wants to know the probability of it being popular or non-popular, LDA and QDA are useful. If a news agency wants to knows which features of the article will make it more or less popular, linear/polynomial regression and feature selection procedures are useful. Finally, if the agency only wants to produce an accuracy of results and wants to know how successful the models are, PCA and PLS are useful since they reduce statistical inference, which is not a requirement in this scenario, but they have potential to increase accuracy of the models.

Method Details

The models I personally experimented with were kNN and random forests. K-nearest neighbors is generally used for classification, but kNN can be useful for regression with the use of average-voting. Furthermore, I was responsible for some feature selection by analyzing one key metric, cross-validation accuracy. This metric is in contrast to adjusted R^2, C_p, BIC, and AIC, which are metrics for regression models with RSS as their loss values. Some hints for feature selection were taken from the lasso regression models built by another individual in the group. Additionally, cross-validation was used for parameter tuning, while PCA and PLS were ignored since our dataset is large, achieves significant results in accuracy (seen below), and PCA/PLS are typically used for regression tasks. Finally, to analyze results for the classification test, all five splits discussed above were explored, and type I and II error rates were analyzed for the classification tasks, which was my main concern, for future areas of exploration.

Before useful features are explored, one issue of kNN must be discussed for interpretation of results and

potential avenues of selecting features. K-nearest neighbors utilizes euclidean distance as the distance metric of choice for data points:

$$d_E(x, y) = \sqrt{\sum_{i=1}^{M} (x_i - y_i)^2}$$

In fact, this is the default metric for kNN in R. This works for continuous features, but is not viable for categorical predictors. As mentioned earlier, 44 of the 58 predictors are continuous, so this is not a major issue. In fact, if only 44/58 ($\approx 75.86\%$) of the predictors are utilized, I believe strong, useful models can be built, but some of the categorical predictors seem to be strong indicators of success in terms of article shares since they are associated with the day of the week and category of content in which the article is published. One way to avoid this issue and provide maximum power to our models, gower distance can be used:

$$d_G(x, y) = \frac{\sum_{k=1}^{M} w_k \delta_{xy}^{(k)} d_{xy}^{(k)}}{\sum_{k=1}^{M} w_k \delta_{xy}^{(k)}}$$

Gower distance can be used for both categorical and continuous features. In the above equation, $\delta_{xy}^{(k)} \in \{0, 1\}$, which marks a binary weight in $\{0, 1\}$ where the value is 0 if the feature is assymetric binary and its value for the sample is 0 and it is value 1 otherwise. Also, $d_{xy}^{(k)}$ is the distance between samples x and y along the k^{th} variable and w_k represents the weight of the k^{th} variable.

In fact, in preliminary tests, this change in distance metric significantly boosts accuracy when using all predictors in the model on the binary classification task. With euclidean distance, the accuracy is around 64% using 10-fold cross-validation. With gower distance, the accuracy is around 94% using 10-fold cross-validation using a split at the 50th percentile of shares. On the three-way classification task (splits at 33rd and 66th percentiles), an accuracy of 69% is achieved with gower distance and 40% with euclidean distance. Therefore, for future tests, gower distance will be the metric of choise for the kNN model.

Next, the most important hyperparameter of kNN, k, must be determined. Using 5-fold cross-validation, the value of k was selected from the set $\{1, 3, 5, 21, 51, 101, 501, 1001\}$. I chose a smaller subset of pre-determined values to check since performing 10-fold cross-validation with so many predictors on such a large dataset is infeasible. The results are as follows on the binary task:

```
kNN with k = 1 accuracy: 0.885000

kNN with k = 3 accuracy: 0.933500

kNN with k = 5 accuracy: 0.947500
```

```
kNN with k = 21 accuracy: 0.975000

kNN with k = 51 accuracy: 0.997000

kNN with k = 101 accuracy: 0.999500

kNN with k = 501 accuracy: 1.000000

kNN with k = 1001 accuracy: 1.000000
```

As the value of k increases, the accuracy increases. However, once $k \geq 21$, the accuracy begins to level out. Therefore, with 5-fold CV, we find k = 21 to provide a significant boost in accuracy compared to lower values of k while also being low enough to reduce computational costs, increase possible statistical inference, and have less variance than models with smaller values of k.

There is not much statistical inference that can be performed with kNN since it does not have access to the popular evaluation metrics adjusted R^2, C_p, BIC, and AIC. Additionally, no feature weights are provided similar to linear regression with feature coefficients, so we cannot confidently state one unit of increase in one predictor will lead to a certain increase in shares.

The second model I tested was random forests, which is an ensemble of decision trees. Random forests have two main hyperparameters, which are the number of generated trees and a certain randomness factor that separates them from bagging trees. Bagging simply constructs all trees in the same fashion, choosing from all predictors at each level to dictate the next split. Random forests' randomness factor leads to choosing from a random subset of the remaining predictors at each split decision, thus generating different trees. Because of this randomness, a large number of trees (several hundred) must be generated to ensure all splits are given fair chance. This is different to bagging, which requires only a couple dozen trees. Therefore, 10-fold cross-validation was used to dictate both the number of trees and the randomness factor.

The first test is dictating the number of generated trees. From my tests, only a small number of trees is required to achieve a noticeable error rate. I tested tree sizes in the set $\{2, 4, 6, 20, 50, 100, 500, 1000\}$ and the best error rate was achieved with 20 trees. An MSE of 208,444 was achieved before the error rate begins to even out for the remaining parameter values. In these tests, the randomness factor was fixed at $sqrt(\#predictors) = sqrt(58) \approx 8$.

The next test is dictating the randomness factor of the tree. Values in the set $\{2, 4, 8, 16, 32, 55\}$ were tested; the number of trees was fixed at 20. After using 10-fold cross-validation, a randomness factor of 8 is found to be the best value, which turns out to be our original value. As such, we found random forests to work best on our data with 20 trees and a randomness factor of 8 for the predictor subset selection at each split decision.

Finally, to summarize my findings with these methods, we must validate them on all our splits and regression/classification tasks. As recently mentioned, random forests with 20 trees and a randomness factor of 8 achieves notable results with an MSE of 209,164 on the regression task. On the binary classification task with a split at the 50th percentile, an accuracy of 99% is achieved. On the binary classification task with a split at the 70th percentile, nearly 99% accuracy is achieved. Finally, 99% accuracy is once again achieved with a split at the 90th percentile. On the three-way task with splits at the 33rd and 66th percentiles, 99% accuracy is once again achieved. K-nearest neighbors performs slightly worse but is still an exceptional classifier. On the binary task with splits at the 50th, 70th, and 90th percentiles, accuracies of 95%, 89%, and 91% are achieved. On the three-way task with splits at the 33rd and 66th percentiles, a 78% accuracy is achieved. Finally, on the regression task, kNN results in an MSE of 240,817, slightly worse than random forests.

There was no missing data, so I did not have to worry about that. The most relevant plots for these results are cross-validation error rates for different values of hyperparameters for both models, which are included in the appendix. These methods tie in perfectly with other methods we are utilizing as a group since we are all working on other regression and classification problems. Therefore, they provide more accuracy and availability of models. However, neither model is immensely useful in statistical inference, so they fail to provide support on that front. In terms of feature selection, I found 9 useful predictors with forward subset selection (in order): `kw_avg_avg`, `data_channel_is_tech`, `data_channel_is_socmed`, `kw_max_avg`, `LDA_00`, `weekday_is_thursday`, `weekday_is_wednesday`, `weekday_is_thursday`, and `weekday_is_monday`. Six of these nine predictors are categorical, so using gower distance on kNN proves to be very effective and necessary. All nine predictors formed the basis for tests in the above models.

To conclude, I trust the results of these two models. The forward subset selection was effective and three of the predictors intersected with other useful predictors found by other members in my group: `kw_avg_avg`, `kw_max_avg`, and `LDA_00`. My initial tests with these models was to use features that were highly correlated with the response, but I did not want to introduce any possible multicollinearity, so I decided subset selection was the way to go. Results were impressive, so I think there are no issues on that front. Next, the methods perform extremely well, and one might think they perform suspiciously well. However, I reviewed my code for hours on end, running through any potential bugs, and there were none. This in fact makes sense if we use the regression results to learn about the classification results. The two MSEs for kNN and random forests were around 200k, which means the average prediction was off by about 400 article shares. If we convert this to the binary classification task, we can imagine being off by merely 400 shares when an article's shares can be in the thousands, the classification task is pretty easy. As such, since we found more complex models to produce superior results and the three-way classification task produced a lower accuracy than the binary

classification task, the results are stable and sound.

Summary of Results

First looking at our linear model, using C_p and BIC we found that a 3 variable model would best fit the data. The three variables determined to fit best based on AIC and lowest residual deviance were `kw_avg_avg`, `kw_max_avg`, and `LDA_03`. This linear model of best fit had an error of _____.

We made multiple logistic regression models to fit the various popularity thresholds discussed earlier. Both forward subset selection and LASSO were used to select variables for these different models. At the 95th percentile threshold, the model with the highest predictive accuracy was found using forward subset selection and included 4 variables. The 70th percentile threshold was best fit using LASSO; in comparison to the 95th percentile model, this second logistic regression model was less accurate but had a higher true positive rate. The 50th percentile threshold was the least accurate of the three models with the highest misclassification rate.

We found our best Linear Discriminant Analysis model had the most influential variables of `kw_max_avg`, `rate_positive_words`, `kw_avg_avg` and `LDA_03`. This model had a prediction accuracy of _____.

QDA analysis was performed on the categorical response at the various cutoff points of 50th, 70th, and 95th percentile, as well as one and two standard deviations above the mean. This model was able to generate an accuracy of roughly 75%, but saw a very low true positive rate, especially when the cutoff threshold for popularity was 95th percentile, one standard deviation above the mean, or two standard deviations above the mean.

The best kNN model was found with a k = 21. This model proved effective at predicting popularity outcome with 95% accuracy, 89% accuracy, and 91% accuracy at the 50th, 70th, and 90th percentile thresholds respectively. However, when analyzing 3 categories split at the 33rd and 66th percentiles the kNN model's accuracy dropped to 78%.

Neither regression nor classification trees were effective at predicting the number of shares/popularity of an article. The best fit regression tree was just a single node predicting the average number of shares. At each of the 5 cutoffs for "viral" and "non-viral", the classification tree may have had multiple branches but always predicted not popular; the only exception was the 50th percentile cut of which still had a very low accuracy.

While the regression and classification trees performed poorly, the random forest model as a classification method worked very well. The random forest model generated an accuracy of over 99% for the 50th, 70th,

and 90th percentile cutoffs. This model also performed extremely well when the data was split into thirds, maintaining a 99% accuracy.

Generally, given the large amount of data in our dataset and the large number of predictors, the more flexible models seemed to perform better. When we did various methods of selecting important variables, we saw that `kw_avg_avg`, `kw_max_avg`, `LDA_03`, and `LDA_00` were used across a large amount of the models. The greatest contradiction noticed between the models is the overwhelming success of the random forest model in comparison to the regression and classification trees. As seen above, the random forest model was the most accurate and vastly outperformed the other models considered.

Conclusions and Takeaway

Several key things can be taken away from my investigations and the explorations of other member in my group in terms of model selection for both accuracy and statistical inference, feature selection, and practical use cases of the results.

First, feature selection was extensively explored across several members. As briefly mentioned, four features were repeated across the several feature selection methods explored: `kw_avg_avg`, `kw_max_avg`, `LDA_03`, and `LDA_00`. The first two are features extracted from keyword analysis. They stand for "Avg. keyword (avg. shares)" and "Avg. keyword (max. shares)", respectively, as described by the creators of the dataset. These are assumed to be the max and average number of shares for the most average keyword (not most frequency, not most sparse) keyword in that given document. For example if an article contains the keyword "pineapple" and it appears at the 50th percentile of the keywords in that article in terms of frequency, its average number of shares and maximum number of shares are computed across all articles. As such, if the average keyword in an article has a ton of shares in other articles, this may be a good indicator of the article performing successfully. Analogously, if the best keyword in an article has an absolute ton of shares across other articles, in conjunction with the average keyword performing superior as well, then the article has a good chance at reaching a high number of shares. For example, given an article with the 50th percentile keyword as "pineapple", all other articles containing "pineapple" are tracked to find the maximum number of shares an article has with that keyword and the average number of shares across all articles containing the keyword. With a high success rate, if "pineapple" tends to be present in highly successful articles, then there is a good chance this article will be successful. These chosen features make sense as good predictors for our models. Additionally, `LDA_03` AND `LDA_00` explore "closeness to LDA topic 3" and "closeness to LDA topic 0" respectively, as defined by the dataset creators. These are assumed to be related to article topics `social media` and `lifestyle`

respectively, since they in that order in the dataset attribute representations. In addition, they are believed to be probabilities of belonging to each respective topic, since some statistical analysis shows they are in the range of 0 to 1 and LDA stands for linear discrimnant analysis. These both make sense. After some brief exploration of Mashable's content, I personally only know them for social media, lifestyle, and tech content. Sometimes, the tech content mixes into social media content, such as an article about Burger King creating AI-written ads (`https://mashable.com/article/burger-king-ai-ads-beautiful-disaster/#SFUR74H5Kmq0`). As such, articles related to these topics generally fare better than other articles in other topics. When including the features I specifically found to produce notable accuracies, the tech, social media, and day of publication (Monday, Tuesday, Wednesday, Thursday) are found to be useful. Social media was already found to be a significant predictor with `LDA_03`, so the new additions are the tech category and day of publication. As already mentioned, sometimes tech crosses over into social media, so that produces an explanation as to why that is a significant predictor. Also, the fact that the first four weekdays are significant is interesting because I suspect those are the four days people pay attention to pop media news the most, since Friday, Saturday, and Sunday are spent hanging with friends, taking care of kids, and catching up on activities consumers cannot otherwise complete during the weekday. As such, those four weekdays are pivotal in keeping consumers up-to-date during their busy work week. Interestingly enough, none of our group members found article title or content polarity to be a useful predictor in any model. I initially would have expected those related predictors to be useful since I generally see a lot of controversial news articles shared, such as through the New York Times (i.e. Donald Trump) or ESPN (i.e. Lakers and Lebron).

Next, we examine model selection in terms of both statistical inference and prediction accuracy. From the above results, random forests performs superior across all classification tasks. This is great since it provides a 99% accuracy in terms of whether an article will be viral or not, but basically limits all statistical inference since random forests are based on ensembles of large gatherings of trees. In order to perform statistical inference, a model must be straightforward with key, stable attributes in one prediction model that we can analyze, such as regression coefficients in a linear model. So, in practice, we would use random forests to perform predictions. K-nearest neighbors performs nearly identical in terms of classification accuracy, but allows for a bit more interpretability. For example, kNN with averaging or class prediction can tell you how popular or whether an article will be popular, but with $k = 21$, it provides 21 similar articles (taken from a vast, thorough dataset of 40k samples) that authors can analyze to find similar flaws or similar strengths across the articles. With these flaws and strengths, the article's author can improve or modify an article to increase article shares. However, this may take a bit of time to analyze all 21 articles compared to 5 articles for example ($k = 5$). In terms of regression, random forests had an MSE of 208k, which means each

prediction for an article was off by average of around 400 shares, which is not a lot considering articles receive thousands of shares. As such, this model is very accurate for predicting specific number of shares.

In general, I would say we reached substantial results in terms of accuracy and mixed results in terms of inference. Practical takeaways were already briefly discussed in terms of model selection and feature selection, but to recap, article topic, publication day, and average keyword popularity in an article are the three most valuable attributes in determining a successful news article.

These results do not require a sophisticated statistics background to understand them. In fact, any of these models can be abstracted away into a simple software tool that analyzes an article before it is published, by simplying uploading the PDF or text file. Then, the author receives instant feedback on how well the article will most likely end up performing. I have already briefly discussed the statistical inference side of things; publish early in the week and play less of a focus on major buzzwords, but rather place more emphasis on typical words/keywords utilized in the document to catch a reader's attention. Finally, if possible, even if the article is not specifically social media or lifestyle related, include possible hints or taglines that may lead the reader to believe otherwise, such as the Burger King tech ad that also fit into the social media article. Not all of these conclusions are apparent to the average author, but may marginally improve results for Mashable articles.

Finally, although this dataset is specifically tailored for Mashable, I believe there is room for abstraction to other news agencies. For example, the New York Times has many more article topics and categories, but I believe they can be effectively binned into the 6 categories utilized in this dataset (lifestyle, entertainment, business, social media, tech, world). Additionally, none of these attributes are specifically tailored for Mashable. If I were given this dataset without somebody telling me it was collected from Mashable, I would have guessed it is from a major news agency since the dataset is thorough, large in size, and produces somewhat predictable but nice inferential results.

Appendix

Peer Evaluation

James Hahn: 20%

Eric Vance: 20%

Jordan Abbott: 20%

Saagar Menon: 20%

Suprotik Debnath: 20%