Predicting the Popularity of Mashable.com News Articles

Josh Wilder, Chi-Hua Wu, Vicky Pang, Akshay Pokharkar, Anant Khandelwal

University of Connecticut

Predictive Modeling - OPIM 5604

Professor Jennifer Eigo

04/26/2019

Table of Contents

Executive Summary

In order for online news companies like Mashable to succeed, they need to determine patterns and trends that contribute to the popularity of their models. Our goal was to create and develop a model predicting which Mashable articles were widely shared on social networks based on several features of online news. Based on the results, we determined which variables would contribute toward future content creations having a wider reach on social media through organic sharing.

Our business insights and recommendations for Mashable are based on our logistic regression model, which was implemented on a dataset built using stratified undersampling. This model provides multiple insights and recommendations that will help Mashable improve their business. These insights includes the impact of image insertions, article categorization, keyword strength, and article release day. We then conclude with direction and specific actions Mashable could take to improve their articles' virality.

Problem Statement

Mashable is a global, multi-platform media and entertainment company, and they post articles of multiple genres online from which they earn revenue from advertisers. In order for companies like Mashable to succeed, they must be aware of the trends within their successful articles. Without learning these trends, a company like Mashable may fail against competitors who also uses data-driven strategy. Therefore, it is imperative to understand and predict what article characteristics are most appealing to readers.

Our solution to this problem involves predicting which articles are widely shared on social media. Shares on social media is a key metric for article virality, and the specific business insight we are looking for is which factors lead to higher article virality. This is very significant for business because it results in more article views without requiring paid marketing.

Dataset Introduction

The dataset is called "Online News Popularity Data Set" and it can be accessed from UCI Machine Learning Repository (https://archive.ics.uci.edu). Each row represents a news article and was collected from January 7, 2013 to January 7, 2015. There are 39,797 rows and 61 columns as shown in the Appendix Table 1. The original dataset contained 37 attributes of articles, and several natural language processing features were extracted by previous researchers (Fernandes, Vinagre & Cortez, 2015). In this study, 17 selected predictors are used to build models.

Methodology

The methodology of this study is followed by SEMMA. SEMMA stands for Sample, Explore, Modify, Models, and Assess (Shmueli, Bruce, Stephens & Patel, 2017, p. 18). During this process, our data visualizations were created by both Tableau and JMP software, and the models were built via JMP software.

Sample

In the Sample section, the dataset was partitioned into 50% training, 30% validation, and 20% testing set split. The partition was created by JMP (with the fixed random seed of 1234). The training set was used for building models, the validation set was used for accessing the model to avoid overfitting problems, and the test set was used for selecting the best model. However, the results of data visualization showed that this dataset is imbalanced, and we later used undersampling to resolve this problem.

Explore

During data exploration, we discovered anticipated and unanticipated relationships between variables. We began our data exploration focusing on our target variable, “shares.” We then continued to use visualizations on all of our data to detect outliers, missing values, abnormal features, and highly correlated predictors.

1. The Distribution of Shares

The distribution of shares resembles a Johnson Su distribution, as shown in Appendix **Figure 1**. This distribution was very skewed. Although the majority of the articles have between 1,000 and 3,000 shares, there are thousands of articles with over 10,000 shares, and the highest shared articles had hundreds of thousands of shares. The mean number of shares were over double the median, which is a sign that the distribution was very skewed to the right. Please see Appendix **Figure 2**. We determined that this distribution was a problem, because we would expect models like linear regression to be very sensitive, but wildly inaccurate with very highly shared articles. On the other hand, very highly shared articles were the most important in terms of business results, so keeping them in the data is desirable.

2. The Correlations Between Variables.

As the color maps of correlation shows in Appendix **Figure 3**, there were several variables that had high correlations. This meant that they provided similar or overlapping information that could be overly repetitive during the prediction of a target variable. These variables often were similar in meaning, for example, “self_reference_max_shares”, “self_reference_min_shares”, and “self_reference_avg_shares,” whose correlations were near 1.

3. Missing Values and Outliers.

As shown in Appendix **Figure 4** (the portion of scatterplot Matrix), there was a row with abnormally high values for multiple attributes. This included values above 1 for variables that are meant to rate from 0 to 1. In the next section, we detected and excluded them in order to build a robust model to avoid the situation that the extreme values would highly affect the model.

There are 1181 rows which contained a large portion of zero values in "n_token_content" to "average_token_length", "self_reference" related variables, and "global_subjectivity" to "max_negative_polarity." These zeroes suggest, for example, that these articles have no words. Viewing a handful of these articles allowed us to confirm our suspicion that these articles actually did have words, and there was a flaw in data collection. Therefore, we viewed these rows as ones containing missing values.

Modify

Using exploratory analysis and visualizations, we found high values that were considered potential outliers. This section would include statistical methods used to detect outliers. This section would also include data preprocessing, undersampling, and variable selection.

1. Excluding Missing Data.

Although no data was "missing" as represented by a dot in JMP, this dataset did contain missing data. As shown in Appendix **Figure 5**, we found rows that had zeros in multiple columns. Our challenge was differentiating between zeroes that genuinely represented zeros and zeroes that were just missing data. After checking a couple of articles to make sure that there weren't wordless video articles, we went ahead and deleted all rows, which had zero in the column. We continued to do this analysis for multiple columns, and some of which included missing data as -1's instead of zeros (over 2,000 rows were excluded). One downfall of our strategy was that we deleted rows based on missing data in columns that we later deleted. However, we did our best to un-exclude rows that could be used after excluding the problem columns, but this procedure became tedious to do each time. The outcome wouldn't have had enough impact on the results anyway.

2. Excluding Outliers.

Since the popularity of news varies, we applied JMP's interquartile range outlier detection technique to examine our data, setting tail quantile equal to 0.1 and Q equal to 10. The results are shown in Appendix **Figure 6**. It showed that there were three variables that contained wrong data, "N_tokens_content", "non_stop_words", and "n_non_stop_unique_tokens." These are variables that should be rates between 0 and 1, and values outside of this range are invalid data. Our analysis determined that these invalid values are outliers and we excluded them.

3. Principal Component Analysis.

As the result showed in the color map, the correlation of keyword-related variables concluded that they both had positive and negative relationships in between. We then applied the principal component analysis to reduce the number of variables, and then we decided to choose three components to represent six similar variables. The results can be found in Appendix **Figure 7**, and it showed that three components can stand for 84.546% of information.

4. Combined Dummy Variables into One Categorical Variable.

Due to the dataset containing several dummy variables in published articles, this would cause problems to have n classes in a model instead of n-1. According to what we've learned in our lectures, the JMP software is easier to handle categorical variable rather than several dummy variables. As seen in Appendix **Figure 8**, we used the formula in JMP to create the categorical variable.

5. Creating an Alternate Decision Variable.

After observing the skewness of the distribution for shares, we considered altering our response column. We also considered a slight reframing of our business goal, would it make more sense to predict the number of shares that each article gets or should we predict a group of articles that would account for a disproportionality high number of shares? Attempting to determine the feasibility of the second idea, we created a column with a binary response where a 1 is an article which had at least 2900 shares. This came out to be the top 24% most shared articles, and they accounted for 71.68% of the total shares of all Mashable articles on social media. Based on this data exploration, we decided it would make sense (in terms of our business problem) to focus on this group of important articles. We created "IsAtLeast2900Shares" and used it as our target variable. The formula is as shown in Appendix **Figure 9**.

6. Variable Selection.

The results of Natural Language Processing usually depends on the researcher's interpretation and the programming methods. Since the variables contained a large portions of natural language processing, this procedure decided to select a few of them into the model. Based on the results in data exploration and our best business judgement, we selected universal variables for building models, and the models were not limited to all of those variables. As shown in **Table 2** in Appendix, it contained all the variables we selected for building models.

7. Undersampling.

The unbalanced proportion in shares over 2900 and less caused the model to predict the class with a large proportion. We undersampled to create a 50/50 balanced proportion of 1's and 0's in the response column (in the training and validation data). We used this dataset to build and tune most of our models. The proportion of the data can be found in Appendix **Figure 10**.

Models

Moving forward, we created our models, which were able to predict categorical variables. This procedure would apply Logistic Regression, Decision Tree, Bootstrap Forest, Boosted Tree, K-Nearest Neighbor, Neural Network, and Discriminant Analysis to predict the desirable outcomes. During this procedure, we've tried several combinations of predictors to build models, and based on each results, we will select the best model.

1. Logistic Regression.

After our original variable selection, we reduced the numbers of predicting variables by implementing stepwise selection. Regardless of whether we used forward, backward, or mixed selection, the same 17 of the 22 considered variables were always chosen, counting each category of the categorical variable "data channel" as separate variables. We conducted stepwise selection by using Max Validation R-Square as the stopping rule except when possible, but based on p-values. The results can be seen in Appendix **Figure 11**.

Since we were dealing with a "rare event" problem, we decided to use a stratified undersampling dataset to see if it would improve our logistic regression model. We built this second model using the same stepwise selection process. As the results shown in Appendix

Figure 12, the model out performed our original logistic regression (model comparison is explained in ASSESS section).

Although we acknowledged that this was insufficient evidence to prove that stratified undersampling would improve all of our models, we decided to build the rest of our models on the dataset we built using stratified undersampling, because of the improvement we saw here. A second reason for this choice was that it would eliminate the need for using alternate cutoffs for each model.

2. Decision Tree.

We tried a decision tree on the original dataset, but it found less than ten true positives. This section would detail the decision tree implemented on the stratified dataset, and both models used "IsAtLeast2900Shares" as the target variable. During splitting and pruning, we found that the optimal number of splits were 12, achieving the best possible validation R-square, of 0.1037. Screenshots of these models can be found in the Appendix **Figure 13**.

3. Bootstrap Forest.

As single trees may not have good predictive ability, we combined results from multiple trees to improve the performance. The multi-tree approach that we used was the Bootstrap Forest technique, which helped us make the most out of the available data. This method had a lesser chance of overfitting and was not sensitive to outliers. We proceeded with 50 to 100 to 1000 of trees, and later found that 50 trees showed that the accuracy was lower. For 100 trees, the accuracy was higher and any trees larger than 100 almost yielded the same results. As shown in Appendix **Figure 14**, the results we obtained from 100 trees were as follows: RMSE = 0.4612 and Accuracy = 65.47%

4. Boosted Tree.

The advantage of using the boosted tree technique was that it supported loss function. Each subsequent tree was designed to correct errors of ones that came prior, in this case it helped boost the performance where it made mistakes. By keeping the number of trees as 100, the RMSE came out to 0.4694 and the Accuracy came out to 66.46%. The results is shown in Appendix **Figure 15**.

5. K-Nearest Neighbor (KNN).

The most challenging aspect of KNN models were software crashes. When we used all of the dataset (39,644 total row numbers) to build a model with data partition rule 50-30-20, the JMP software crashed for two group members who has attempted to build a KNN model on two different computers. A third group member tried KNN on the undersampled dataset with 23,617 total row numbers and succeeded. We believed that the reduced number of rows made computation easier. We used the same variables that we used in the regression model after the stepwise selection. Tuning to minimize validation set misclassification rate, we decided to use K equals 81, considering all Ks are up to 100. The accuracy of this model was 69.307%, as shown in Appendix **Figure 16**. Unfortunately, the KNN models had limited interpretation. Aside, suggesting "make articles similar to ones that have succeeded in the past," there was little to recommend for business without actually using the model.

6. Neural Network.

During this procedure, we used the same variables that we used in the regression model after the stepwise selection. We tested a multitude of different neural network structures in terms of the number and types of nodes in each hidden layer. Sparing the details, we ultimately found that a two hidden layer model with 3 tanh nodes, 3 linear nodes, and 4 gaussian nodes in both layers worked best. We implemented this model using 12 tours in order to improve accuracy. The difficulty of model interpretation limits the amount of business insights that could be drawn from this model. The results can be found in Appendix **Figure 17**.

7. Linear Discriminant Analysis.

We used stepwise selection in order to achieve the lowest validation misclassification rate. The resulting model only used four predictors: the first two Keyword Strength principal components, self_reference_avg_shares, and num_hrefs. As shown in Appendix **Figure 18**, the validation misclassification rate was 31.95%. This model was limited because it could only be used for continuous predictors and it does not determine probabilities. Therefore, alternate cutoff points could not be considered.

Assess

The first two models that we compared were both logistic regressions that also used the stepwise selection to select predictor variables. The first model was built on the original dataset (with excluded data), and the second model was built on the undersampled dataset. At first, we thought to compare the models using RMSE, Accuracy, or Accuracy of 1's, but realized that these were all problematic. The first model used a cutoff of .50, which excelled in terms of these test metrics, but only actually found 149 true positives, while the second model found over a thousand. We then realized that we could compare the models using test AUC (area underneath the test ROC curve) in order to quickly compare these models considering all possible cutoffs. We knew that selecting a specific alternate cut off wasn't necessary for business recommendations.

The second model, which was the logistic regression model was built on the undersampled dataset, had high test AUC. Therefore, we selected this as the best model, and decided to continue testing models built on the undersampled data. Due to the structure of the undersampled data, models built upon it did not have an issue with finding too few true positives. We compared all of these models using RMSE and Accuracy, both of which ranked our models the same way. We found that our KNN model was the best model statistically, but decided it would not help us make good business insights. Due to interpretive ability, we chose the logistic regression model as our best model for making business insights and recommendations. Please see model comparisons in Appendix **Table 3**.

Results

We will make business recommendations for mashable based on our logistic regression model built on the stratified undersampling dataset which used stepwise selection. During stepwise selection, one variable that was eliminated was number of images. For this reason, we recommend Mashable to not worry about including many images in their articles.

Based on variable significance and the formula of our model, we recommend making less world, business, and entertainment articles (**Appendix Figure 19**). We recommend making more articles categorized as social media, tech, lifestyle, and others. We make this recommendation noting that we are doing so assuming the goal of maximizing shares on social media, but recognize that although making more social media articles contribute to this goal, it may not translate to more total views or profit.

Our next recommendation is to focus on keyword strength. Based on the significance of multiple keyword strength predictor variables in our model, we believe that having popular keyword is crucial to having a popular article. In terms of business practice, I would recommend having at least employee whose sole job is to improve keyword strength. We believe that an expert in this specific task would do much better than writers or editors, and would have a significant impact on article virality on social media.

Our last recommendation is to publish more articles on the weekend. Although it is important to engage audiences as news occurs in real-time and before competitors, for articles that are less time-sensitive, they should be published on the weekend in order to reach a larger audience particularly through social media shares.

Conclusion

In conclusion, in order to improve social media virality, we suggest Mashable focus most on publishing the right types of articles, with the right keywords, at the right time. Specifically, efforts like publishing articles on the weekend would cost Mashable little to nothing and would greatly increase social media visibility, while efforts like including many images in articles cost Mashable time and money and achieve very little. With our insights and recommendations, we believe that the average popularity of Mashable articles on social media would increase considerably.

References

Taylor, C. (2016, October 26). How Internet Affects the Newspaper Business. Retrieved from: https://smallbusiness.chron.com/

Bryant, M. (2016, 3 Mar).*"20 Years Ago Today, the World Wide Web Was Born",* TNW Insider. Retrieved from: https://thenextweb.com/insider/

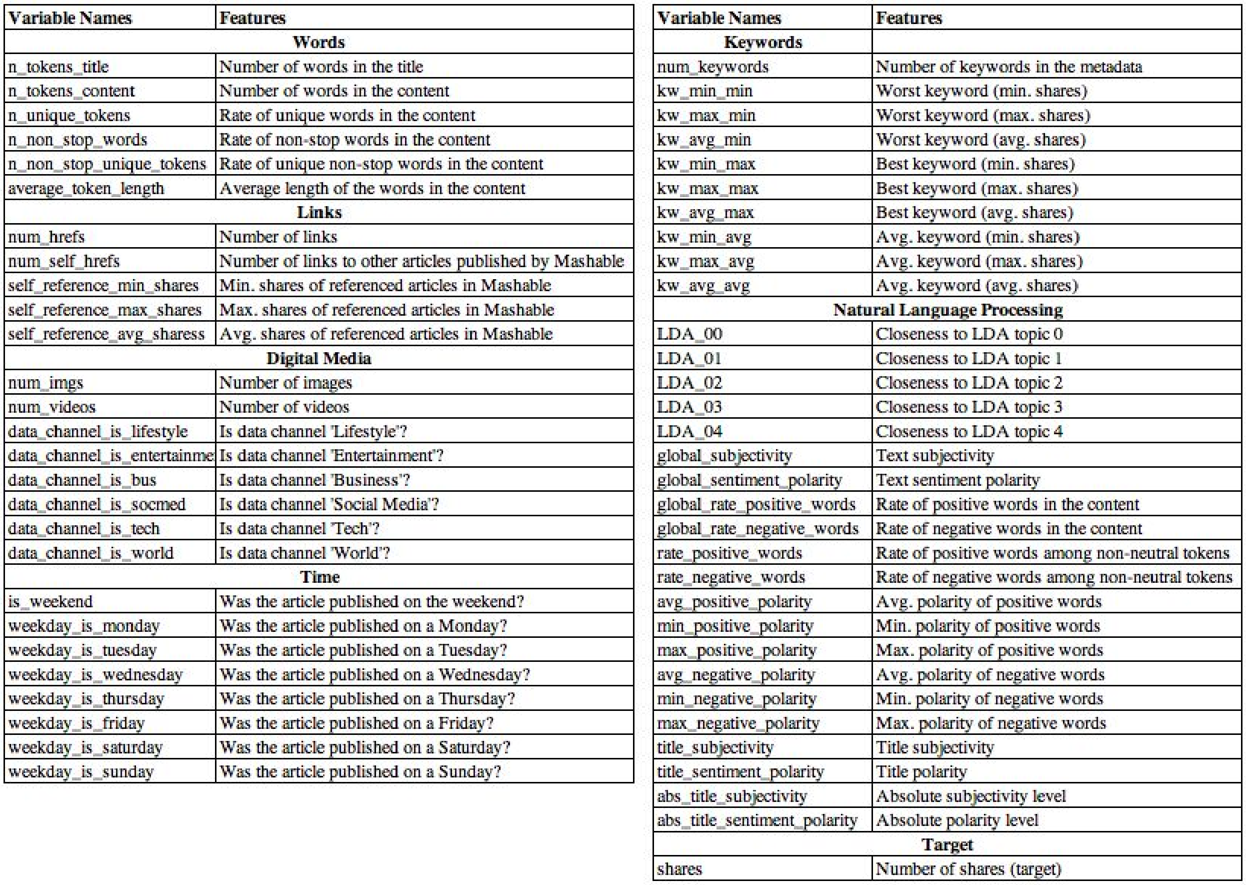
Laird, S., & Laird, S. (2012, April 18). *How Social Media Is Taking Over the News Industry* [INFOGRAPHIC]. Retrieved from: https://mashable.com/

Shmueli, G., Bruce, P. C., Stephens, M. L., & Patel, N. R. (2017). *Data Mining for Business Analytics: Concepts, Techniques, and Applications with JMP Pro.*John Wiley & Sons.

Fernandes, K., Vinagre, P., & Cortez, P. (2015, September). *A proactive intelligent decision support system for predicting the popularity of online news*. In Portuguese Conference on Artificial Intelligence (pp. 535-546). Springer, Cham.

Appendix - Tables

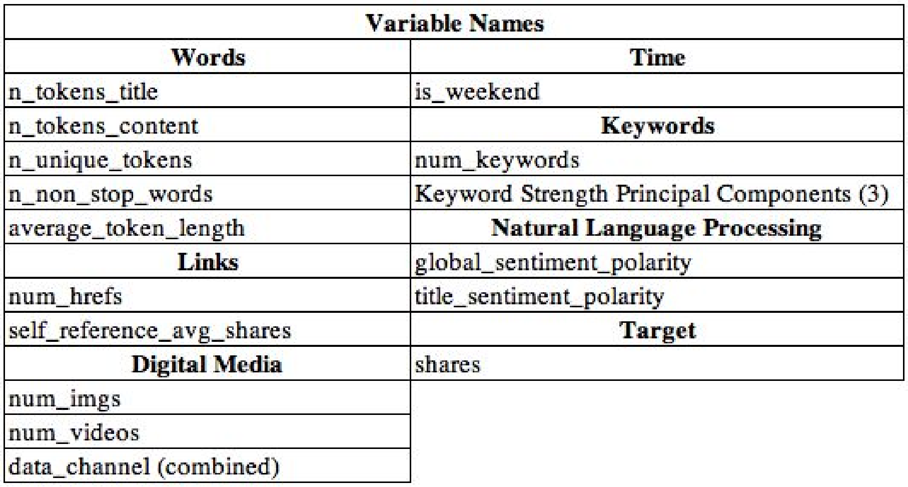
Table 1. List of attributes by category.



Variable Names	Features
Words	
n_tokens_title	Number of words in the title
n_tokens_content	Number of words in the content
n_unique_tokens	Rate of unique words in the content
n_non_stop_words	Rate of non-stop words in the content
n_non_stop_unique_tokens	Rate of unique non-stop words in the content
average_token_length	Average length of the words in the content
Links	
num_hrefs	Number of links
num_self_hrefs	Number of links to other articles published by Mashable
self_reference_min_shares	Min. shares of referenced articles in Mashable
self_reference_max_shares	Max. shares of referenced articles in Mashable
self_reference_avg_sharess	Avg. shares of referenced articles in Mashable
Digital Media	
num_imgs	Number of images
num_videos	Number of videos
data_channel_is_lifestyle	Is data channel 'Lifestyle'?
data_channel_is_entertainme	Is data channel 'Entertainment'?
data_channel_is_bus	Is data channel 'Business'?
data_channel_is_socmed	Is data channel 'Social Media'?
data_channel_is_tech	Is data channel 'Tech'?
data_channel_is_world	Is data channel 'World'?
Time	
is_weekend	Was the article published on the weekend?
weekday_is_monday	Was the article published on a Monday?
weekday_is_tuesday	Was the article published on a Tuesday?
weekday_is_wednesday	Was the article published on a Wednesday?
weekday_is_thursday	Was the article published on a Thursday?
weekday_is_friday	Was the article published on a Friday?
weekday_is_saturday	Was the article published on a Saturday?
weekday_is_sunday	Was the article published on a Sunday?

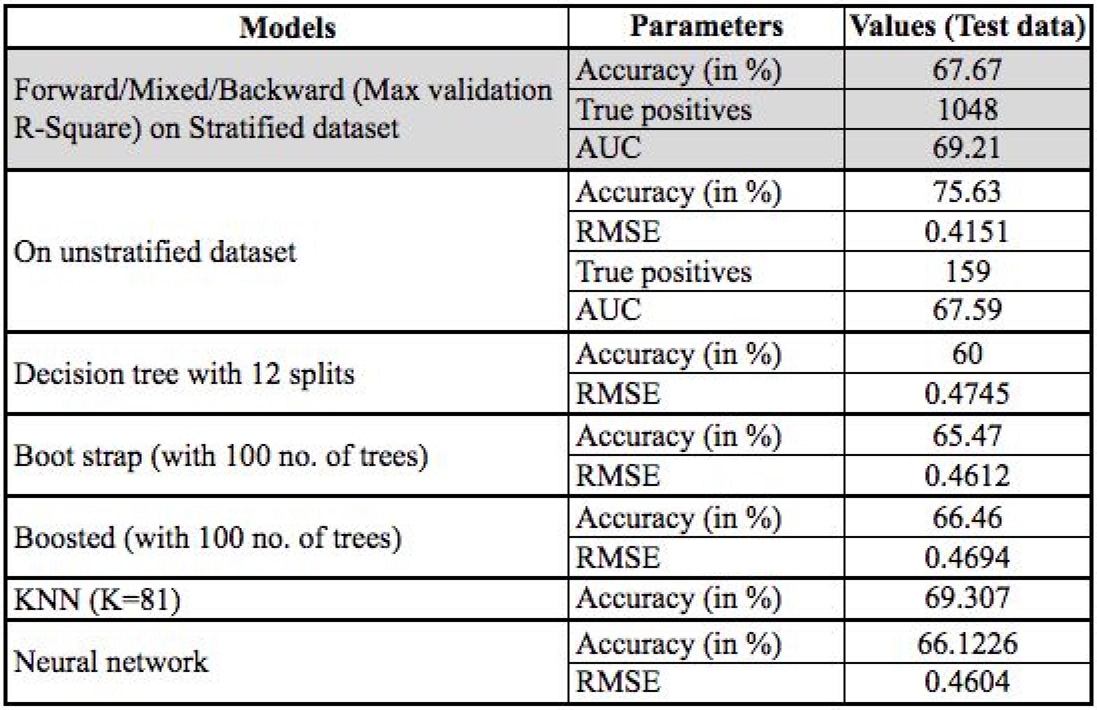
Variable Names	Features
Keywords	
num_keywords	Number of keywords in the metadata
kw_min_min	Worst keyword (min. shares)
kw_max_min	Worst keyword (max. shares)
kw_avg_min	Worst keyword (avg. shares)
kw_min_max	Best keyword (min. shares)
kw_max_max	Best keyword (max. shares)
kw_avg_max	Best keyword (avg. shares)
kw_min_avg	Avg. keyword (min. shares)
kw_max_avg	Avg. keyword (max. shares)
kw_avg_avg	Avg. keyword (avg. shares)
Natural Language Processing	
LDA_00	Closeness to LDA topic 0
LDA_01	Closeness to LDA topic 1
LDA_02	Closeness to LDA topic 2
LDA_03	Closeness to LDA topic 3
LDA_04	Closeness to LDA topic 4
global_subjectivity	Text subjectivity
global_sentiment_polarity	Text sentiment polarity
global_rate_positive_words	Rate of positive words in the content
global_rate_negative_words	Rate of negative words in the content
rate_positive_words	Rate of positive words among non-neutral tokens
rate_negative_words	Rate of negative words among non-neutral tokens
avg_positive_polarity	Avg. polarity of positive words
min_positive_polarity	Min. polarity of positive words
max_positive_polarity	Max. polarity of positive words
avg_negative_polarity	Avg. polarity of negative words
min_negative_polarity	Min. polarity of negative words
max_negative_polarity	Max. polarity of negative words
title_subjectivity	Title subjectivity
title_sentiment_polarity	Title polarity
abs_title_subjectivity	Absolute subjectivity level
abs_title_sentiment_polarity	Absolute polarity level
Target	
shares	Number of shares (target)

Table 2. Variable Selection List



Variable Names	
Words	**Time**
n_tokens_title	is_weekend
n_tokens_content	**Keywords**
n_unique_tokens	num_keywords
n_non_stop_words	Keyword Strength Principal Components (3)
average_token_length	**Natural Language Processing**
Links	global_sentiment_polarity
num_hrefs	title_sentiment_polarity
self_reference_avg_shares	**Target**
Digital Media	shares
num_imgs	
num_videos	
data_channel (combined)	

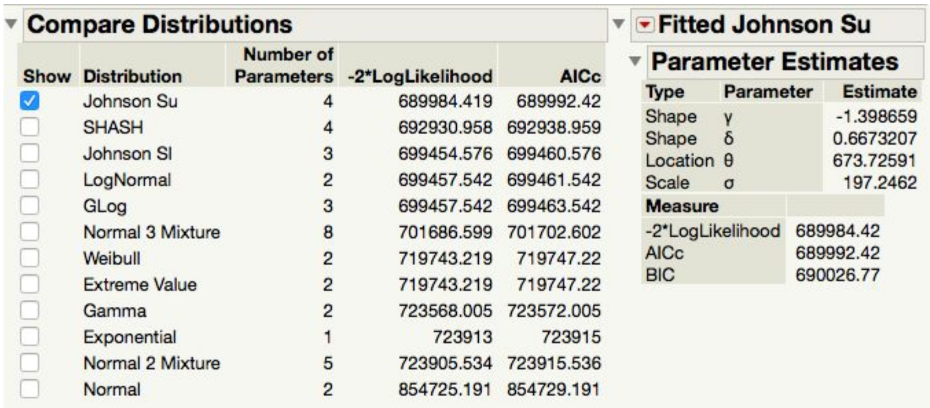
Table 3. The Comparison of models



Models	Parameters	Values (Test data)
Forward/Mixed/Backward (Max validation R-Square) on Stratified dataset	Accuracy (in %)	67.67
	True positives	1048
	AUC	69.21
On unstratified dataset	Accuracy (in %)	75.63
	RMSE	0.4151
	True positives	159
	AUC	67.59
Decision tree with 12 splits	Accuracy (in %)	60
	RMSE	0.4745
Boot strap (with 100 no. of trees)	Accuracy (in %)	65.47
	RMSE	0.4612
Boosted (with 100 no. of trees)	Accuracy (in %)	66.46
	RMSE	0.4694
KNN (K=81)	Accuracy (in %)	69.307
Neural network	Accuracy (in %)	66.1226
	RMSE	0.4604

Appendix - Figures

Figure 1. The Result for the Distribution for the number of shares



Compare Distributions

Show	Distribution	Number of Parameters	-2*LogLikelihood	AICc
☑	Johnson Su	4	689984.419	689992.42
□	SHASH	4	692930.958	692938.959
□	Johnson Sl	3	699454.576	699460.576
□	LogNormal	2	699457.542	699461.542
□	GLog	3	699457.542	699463.542
□	Normal 3 Mixture	8	701686.599	701702.602
□	Weibull	2	719743.219	719747.22
□	Extreme Value	2	719743.219	719747.22
□	Gamma	2	723568.005	723572.005
□	Exponential	1	723913	723915
□	Normal 2 Mixture	5	723905.534	723915.536
□	Normal	2	854725.191	854729.191

Fitted Johnson Su

Parameter Estimates

Type	Parameter	Estimate
Shape	γ	-1.398659
Shape	δ	0.6673207
Location	θ	673.72591
Scale	σ	197.2462

Measure	
-2*LogLikelihood	689984.42
AICc	689992.42
BIC	690026.77

Figure 2. The distribution of the count of shares when shares under 20K.

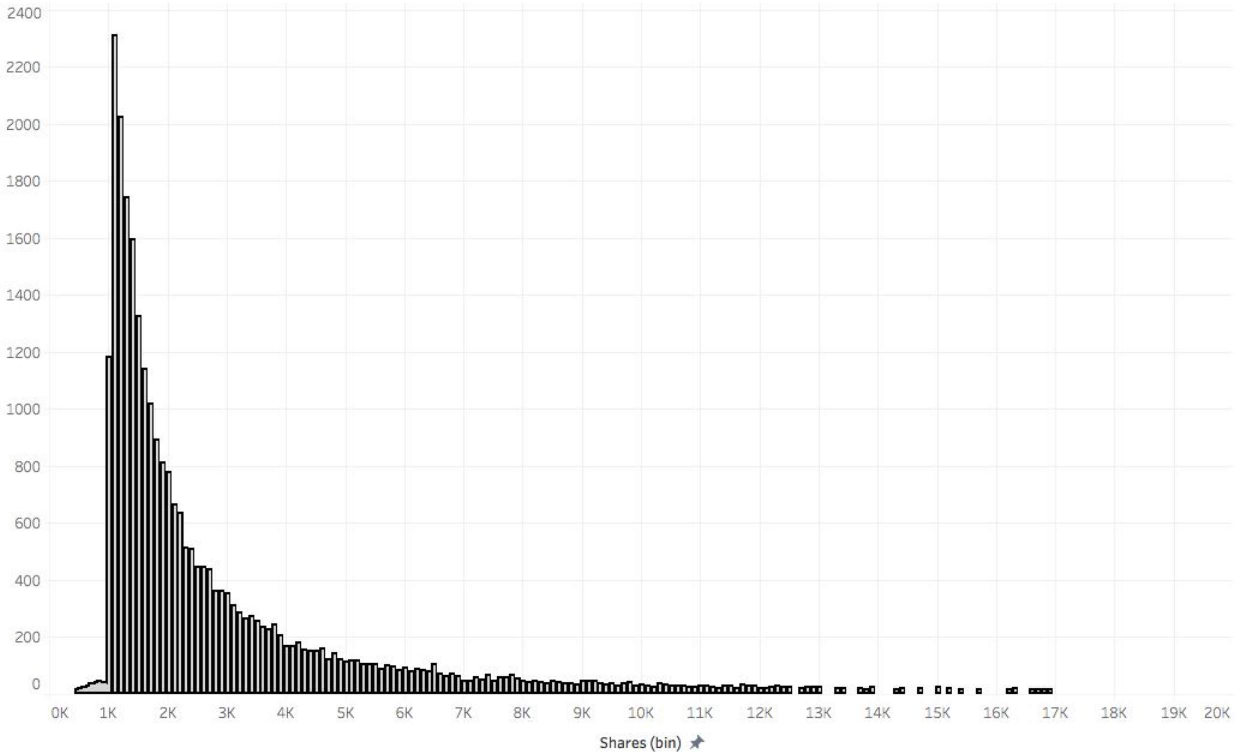


Figure 3. The color Map on Correlations

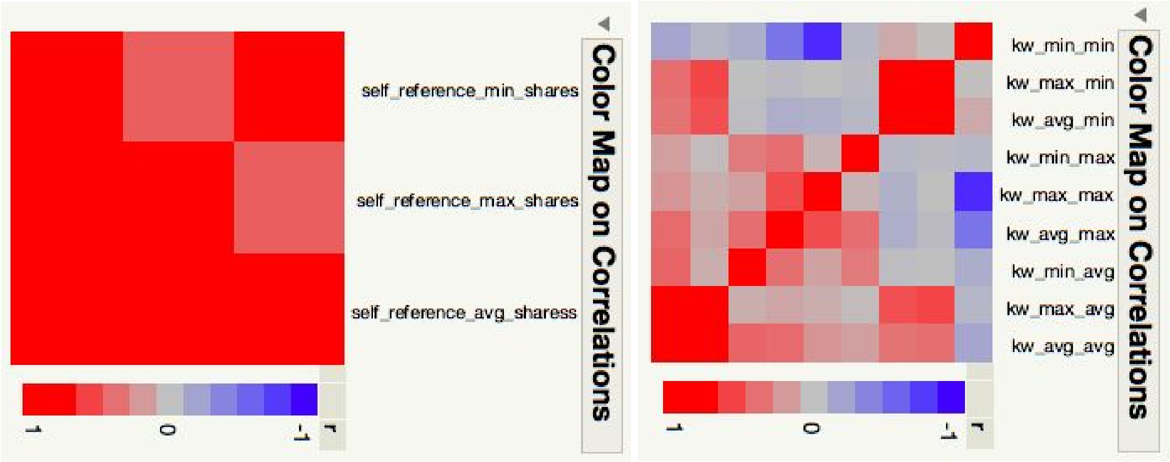


Figure 4. Scatterplot Matrix (a portion of the variables)

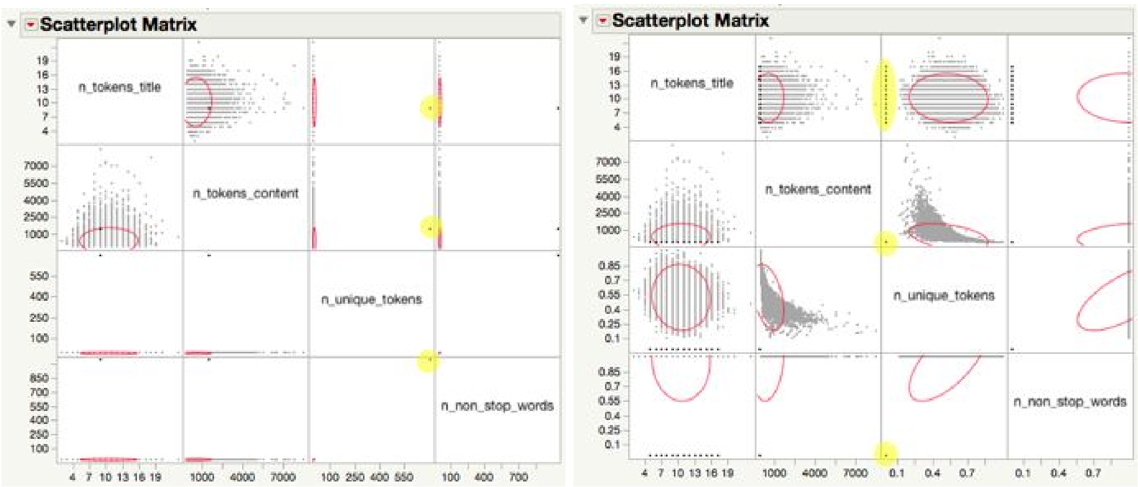
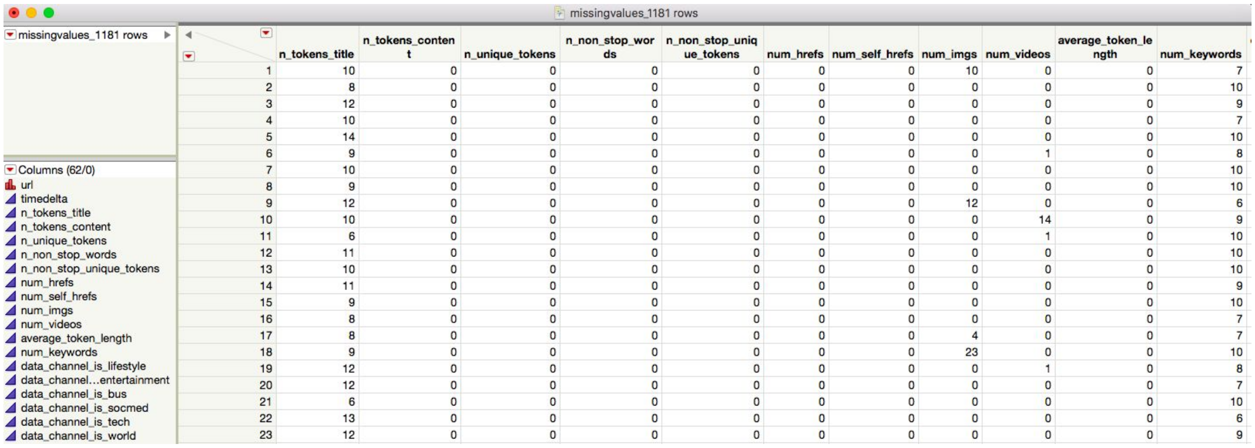


Figure 5. An Example of Rows with Missing Data Shown as Zeroes.



missingvalues_1181 rows

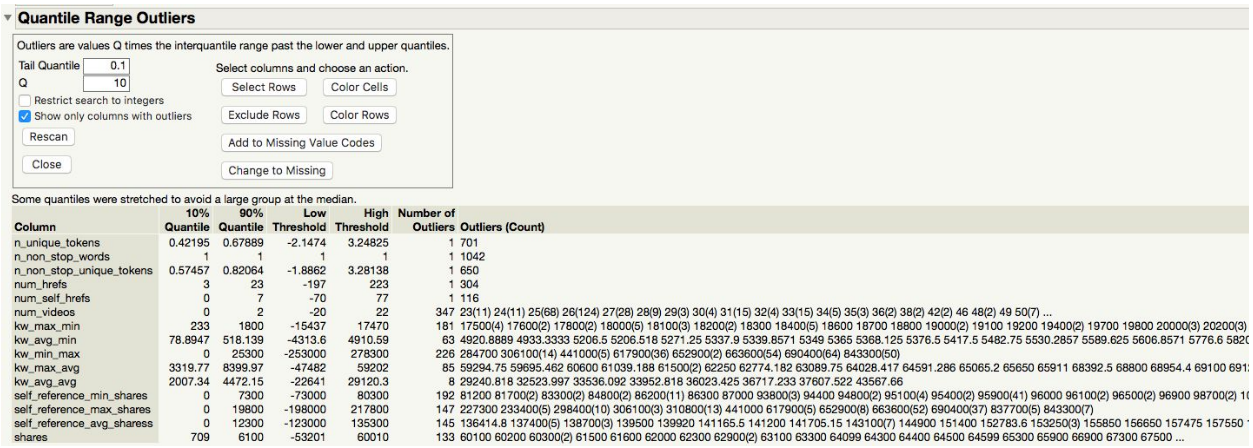
missingvalues_1181 rows

Columns (62/0)

- url
- timedelta
- n_tokens_title
- n_tokens_content
- n_unique_tokens
- n_non_stop_words
- n_non_stop_unique_tokens
- num_hrefs
- num_self_hrefs
- num_imgs
- num_videos
- average_token_length
- num_keywords
- data_channel_is_lifestyle
- data_channel...entertainment
- data_channel_is_bus
- data_channel_is_socmed
- data_channel_is_tech
- data_channel_is_world

	n_tokens_title	n_tokens_content	n_unique_tokens	n_non_stop_words	n_non_stop_unique_tokens	num_hrefs	num_self_hrefs	num_imgs	num_videos	average_token_length	num_keywords
1	10	0	0	0	0	0	0	10	0	0	7
2	8	0	0	0	0	0	0	0	0	0	10
3	12	0	0	0	0	0	0	0	0	0	9
4	10	0	0	0	0	0	0	0	0	0	7
5	14	0	0	0	0	0	0	0	0	0	10
6	9	0	0	0	0	0	0	0	1	0	8
7	10	0	0	0	0	0	0	0	0	0	10
8	9	0	0	0	0	0	0	0	0	0	10
9	12	0	0	0	0	0	0	12	0	0	6
10	10	0	0	0	0	0	0	0	14	0	9
11	6	0	0	0	0	0	0	0	1	0	10
12	11	0	0	0	0	0	0	0	0	0	10
13	10	0	0	0	0	0	0	0	0	0	10
14	11	0	0	0	0	0	0	0	0	0	9
15	9	0	0	0	0	0	0	0	0	0	10
16	8	0	0	0	0	0	0	0	0	0	7
17	8	0	0	0	0	0	0	4	0	0	7
18	9	0	0	0	0	0	0	23	0	0	10
19	12	0	0	0	0	0	0	0	1	0	8
20	12	0	0	0	0	0	0	0	0	0	7
21	6	0	0	0	0	0	0	0	0	0	10
22	13	0	0	0	0	0	0	0	0	0	6
23	12	0	0	0	0	0	0	0	0	0	9

Figure 6. The results of using Quantile Range Outliers.



Quantile Range Outliers

Outliers are values Q times the interquantile range past the lower and upper quantiles.

Tail Quantile 0.1

Q 10

☐ Restrict search to integers

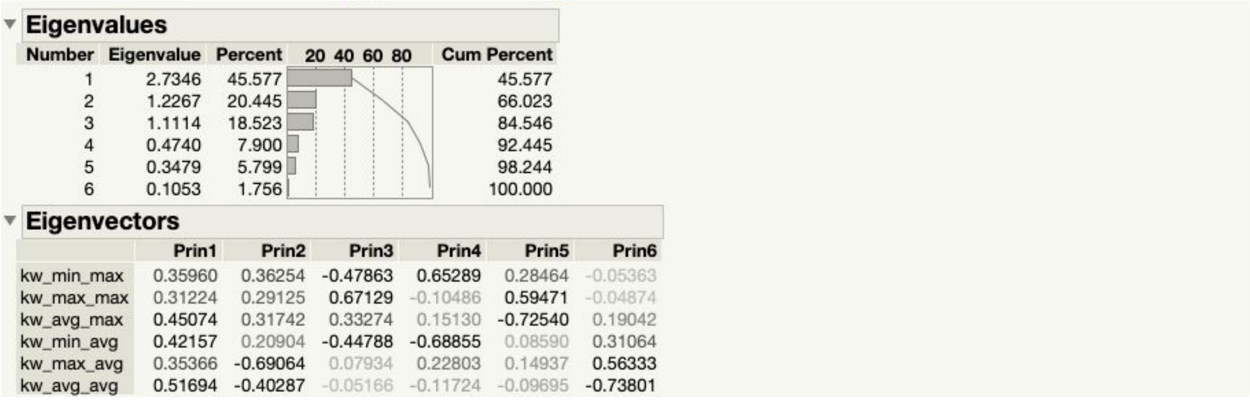
☑ Show only columns with outliers

Select columns and choose an action.

Some quantiles were stretched to avoid a large group at the median.

Column	10% Quantile	90% Quantile	Low Threshold	High Threshold	Number of Outliers	Outliers (Count)
n_unique_tokens	0.42195	0.67889	-2.1474	3.24825	1	701
n_non_stop_words	1	1	1	1	1	1042
n_non_stop_unique_tokens	0.57457	0.82064	-1.8862	3.28138	1	650
num_hrefs	3	23	-197	223	1	304
num_self_hrefs	0	7	-70	77	1	116
num_videos	0	2	-20	22	347	23(11) 24(11) 25(68) 26(124) 27(28) 28(9) 29(3) 30(4) 31(15) 32(4) 33(15) 34(5) 35(3) 36(2) 38(2) 42(2) 46 48(2) 49 50(7) ...
kw_max_min	233	1800	-15437	17470	181	17500(4) 17600(2) 17800(2) 18000(5) 18100(3) 18200(2) 18300 18400(5) 18600 18700 18800 19000(2) 19100 19200 19400(2) 19700 19800 20000(3) 20200(3)
kw_avg_min	78.8947	518.139	-4313.6	4910.59	63	4920.8889 4933.3333 5206.5 5206.518 5271.25 5337.9 5339.8571 5349 5365 5368.125 5376.5 5417.5 5482.75 5530.2857 5589.625 5606.8571 5776.6 582
kw_min_max	0	25300	-253000	278300	226	284700 306100(14) 441000(5) 617900(36) 652900(2) 663600(54) 690400(64) 843300(50)
kw_max_avg	3319.77	8399.97	-47482	59202	85	59294.75 59695.462 60600 61039.188 61500(2) 62250 62774.182 63089.75 64028.417 64591.286 65065.2 65650 65911 68392.5 68800 68954.4 69100 691
kw_avg_avg	2007.34	4472.15	-22641	29120.3	8	29240.818 32523.997 33536.092 33952.818 36023.425 36717.233 37607.522 43567.66
self_reference_min_shares	0	7300	-73000	80300	192	81200 81700(2) 83300(2) 84800(2) 86200(11) 86300 87000 93800(3) 94400 94800(2) 95100(4) 95400(2) 95900(41) 96000 96100(2) 96500(2) 96900 98700(2) 1
self_reference_max_shares	0	19800	-198000	217800	147	227300 233400(5) 298400(10) 306100(3) 310800(13) 441000 617900(5) 652900(8) 663600(52) 690400(37) 837700(5) 843300(7)
self_reference_avg_sharess	0	12300	-123000	135300	145	136414.8 137400(5) 138700(3) 139500 139920 141165.5 141200 141705.15 143100(7) 144900 151400 152783.6 153250(3) 155850 156650 157475 157550
shares	709	6100	-53201	60010	133	60100 60200 60300(2) 61500 61600 62000 62300 62900(2) 63100 63300 64099 64300 64400 64500 64599 65300 65900 66900 67300 67500 ...

Figure 7. The Result of Principal Components Analysis



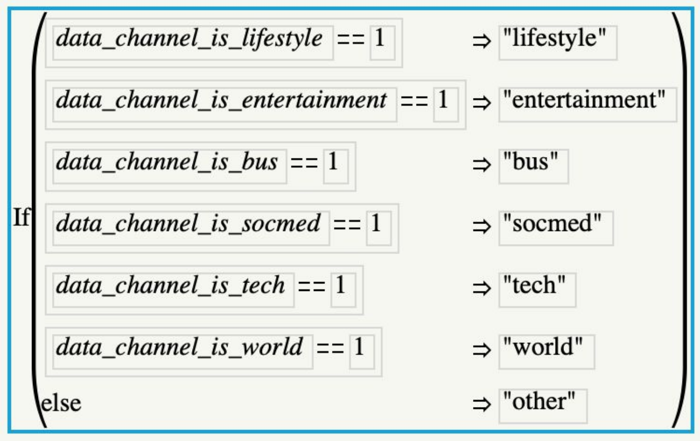
Eigenvalues

Number	Eigenvalue	Percent	Cum Percent
1	2.7346	45.577	45.577
2	1.2267	20.445	66.023
3	1.1114	18.523	84.546
4	0.4740	7.900	92.445
5	0.3479	5.799	98.244
6	0.1053	1.756	100.000

Eigenvectors

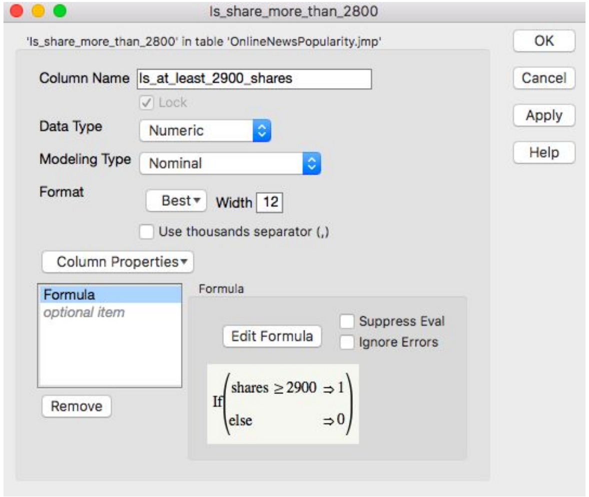
	Prin1	Prin2	Prin3	Prin4	Prin5	Prin6
kw_min_max	0.35960	0.36254	-0.47863	0.65289	0.28464	-0.05363
kw_max_max	0.31224	0.29125	0.67129	-0.10486	0.59471	-0.04874
kw_avg_max	0.45074	0.31742	0.33274	0.15130	-0.72540	0.19042
kw_min_avg	0.42157	0.20904	-0.44788	-0.68855	0.08590	0.31064
kw_max_avg	0.35366	-0.69064	0.07934	0.22803	0.14937	0.56333
kw_avg_avg	0.51694	-0.40287	-0.05166	-0.11724	-0.09695	-0.73801

Figure 8. The Formula to Combine Dummy Variables into One Categorical Variable



$$\text{If}\begin{pmatrix} data\_channel\_is\_lifestyle == 1 & \Rightarrow \text{"lifestyle"} \\ data\_channel\_is\_entertainment == 1 & \Rightarrow \text{"entertainment"} \\ data\_channel\_is\_bus == 1 & \Rightarrow \text{"bus"} \\ data\_channel\_is\_socmed == 1 & \Rightarrow \text{"socmed"} \\ data\_channel\_is\_tech == 1 & \Rightarrow \text{"tech"} \\ data\_channel\_is\_world == 1 & \Rightarrow \text{"world"} \\ \text{else} & \Rightarrow \text{"other"} \end{pmatrix}$$

Figure 9. The Formula to Create an Alternative Decision Variable



Is_share_more_than_2800

'Is_share_more_than_2800' in table 'OnlineNewsPopularity.jmp'

Column Name: Is_at_least_2900_shares

☑ Lock

Data Type: Numeric

Modeling Type: Nominal

Format: Best Width 12

□ Use thousands separator (,)

Column Properties

Formula
optional item

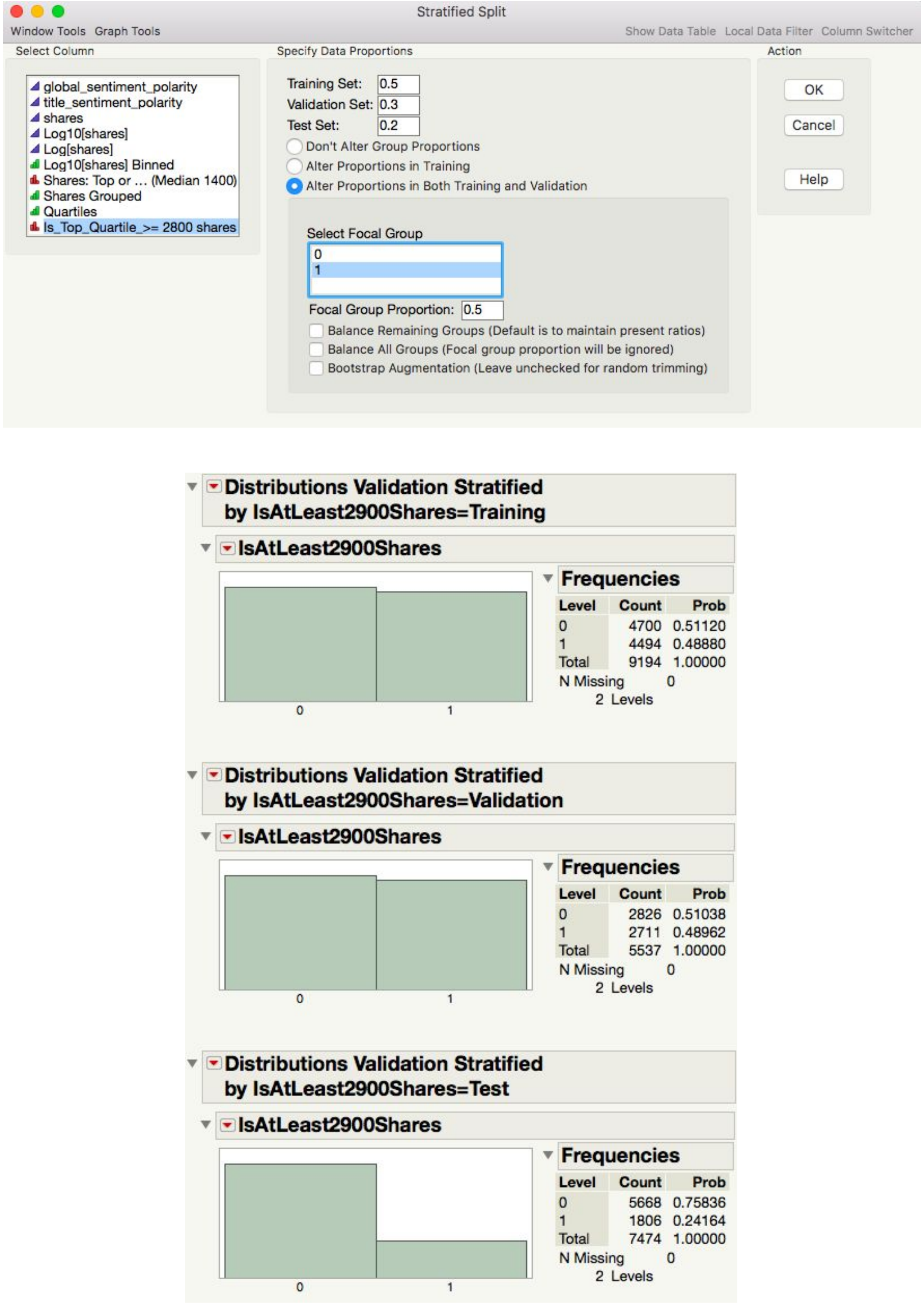
Formula

□ Suppress Eval

□ Ignore Errors

$$\text{If}\begin{pmatrix} \text{shares} \geq 2900 & \Rightarrow 1 \\ \text{else} & \Rightarrow 0 \end{pmatrix}$$

Figure 10. The Results of Undersampling



Select Column

- global_sentiment_polarity
- title_sentiment_polarity
- shares
- Log10[shares]
- Log[shares]
- Log10[shares] Binned
- Shares: Top or … (Median 1400)
- Shares Grouped
- Quartiles
- Is_Top_Quartile_>= 2800 shares

Training Set: 0.5
Validation Set: 0.3
Test Set: 0.2

○ Don't Alter Group Proportions
○ Alter Proportions in Training
● Alter Proportions in Both Training and Validation

Select Focal Group
0
1

Focal Group Proportion: 0.5

□ Balance Remaining Groups (Default is to maintain present ratios)
□ Balance All Groups (Focal group proportion will be ignored)
□ Bootstrap Augmentation (Leave unchecked for random trimming)

Distributions Validation Stratified by IsAtLeast2900Shares=Training

IsAtLeast2900Shares

Frequencies

Level	Count	Prob
0	4700	0.51120
1	4494	0.48880
Total	9194	1.00000

N Missing 0
2 Levels

Distributions Validation Stratified by IsAtLeast2900Shares=Validation

IsAtLeast2900Shares

Frequencies

Level	Count	Prob
0	2826	0.51038
1	2711	0.48962
Total	5537	1.00000

N Missing 0
2 Levels

Distributions Validation Stratified by IsAtLeast2900Shares=Test

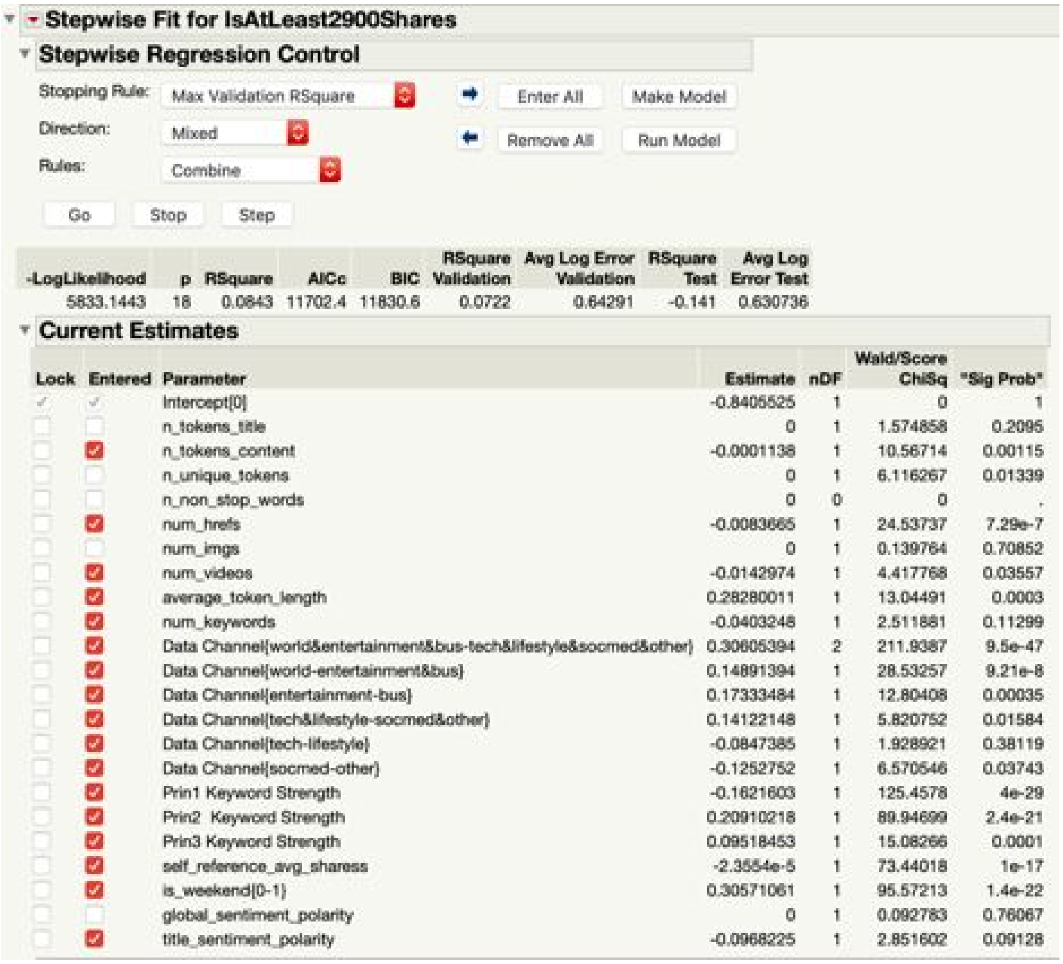
IsAtLeast2900Shares

Frequencies

Level	Count	Prob
0	5668	0.75836
1	1806	0.24164
Total	7474	1.00000

N Missing 0
2 Levels

Figure 11.The Results of Stepwise



Stepwise Fit for IsAtLeast2900Shares

Stepwise Regression Control

Stopping Rule: Max Validation RSquare

Direction: Mixed

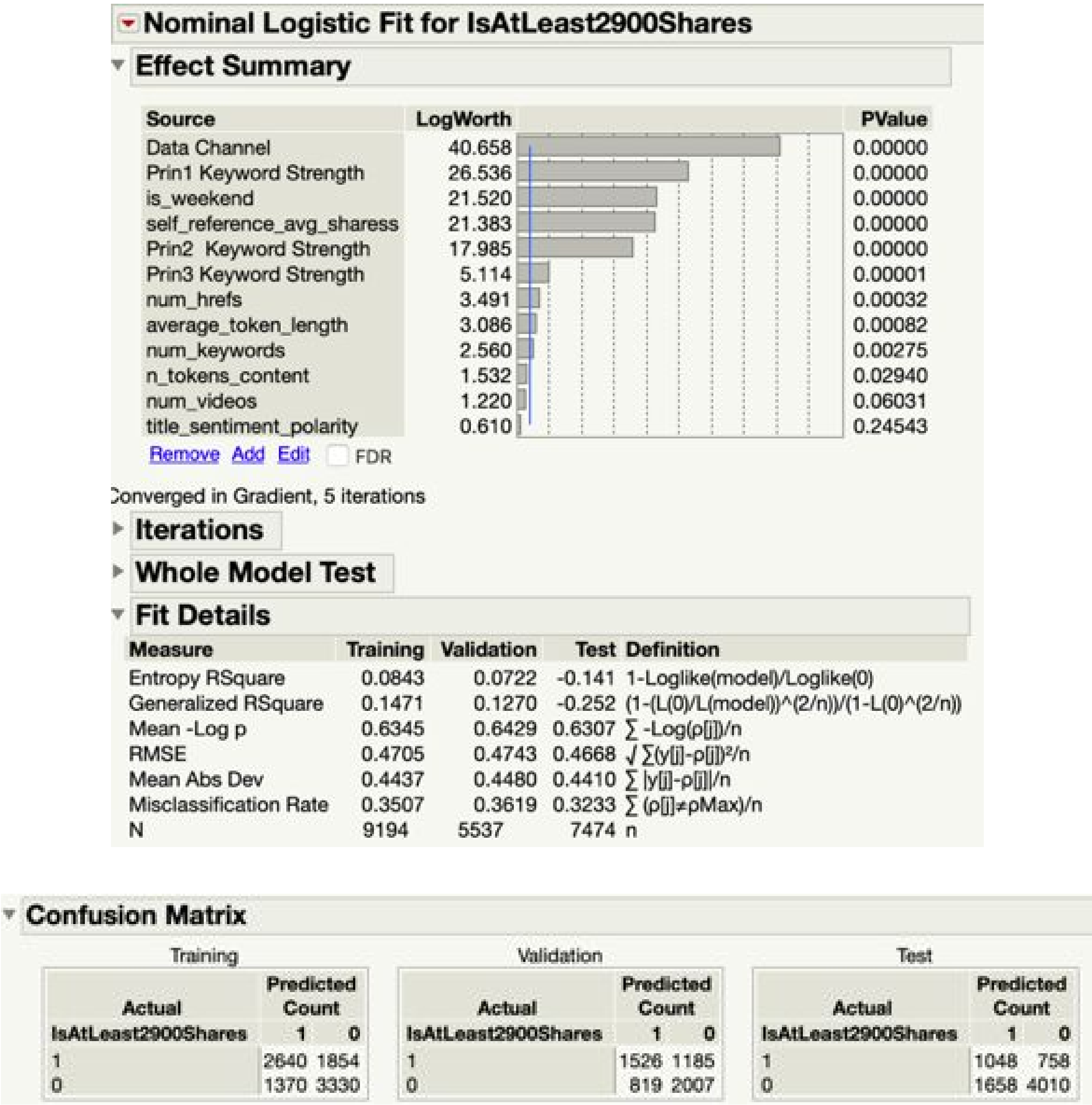
Rules: Combine

-LogLikelihood	p	RSquare	AICc	BIC	RSquare Validation	Avg Log Error Validation	RSquare Test	Avg Log Error Test
5833.1443	18	0.0843	11702.4	11830.6	0.0722	0.64291	-0.141	0.630736

Current Estimates

Lock	Entered	Parameter	Estimate	nDF	Wald/Score ChiSq	"Sig Prob"
✓	✓	Intercept[0]	-0.8405525	1	0	1
□	□	n_tokens_title	0	1	1.574858	0.2095
□	☑	n_tokens_content	-0.0001138	1	10.56714	0.00115
□	□	n_unique_tokens	0	1	6.116267	0.01339
□	□	n_non_stop_words	0	0	0	.
□	☑	num_hrefs	-0.0083665	1	24.53737	7.29e-7
□	□	num_imgs	0	1	0.139764	0.70852
□	☑	num_videos	-0.0142974	1	4.417768	0.03557
□	☑	average_token_length	0.28280011	1	13.04491	0.0003
□	☑	num_keywords	-0.0403248	1	2.511881	0.11299
□	☑	Data Channel{world&entertainment&bus-tech&lifestyle&socmed&other}	0.30605394	2	211.9387	9.5e-47
□	☑	Data Channel{world-entertainment&bus}	0.14891394	1	28.53257	9.21e-8
□	☑	Data Channel{entertainment-bus}	0.17333484	1	12.80408	0.00035
□	☑	Data Channel{tech&lifestyle-socmed&other}	0.14122148	1	5.820752	0.01584
□	☑	Data Channel{tech-lifestyle}	-0.0847385	1	1.928921	0.38119
□	☑	Data Channel{socmed-other}	-0.1252752	1	6.570646	0.03743
□	☑	Prin1 Keyword Strength	-0.1621603	1	125.4578	4e-29
□	☑	Prin2 Keyword Strength	0.20910218	1	89.94699	2.4e-21
□	☑	Prin3 Keyword Strength	0.09518453	1	15.08266	0.0001
□	☑	self_reference_avg_sharess	-2.3554e-5	1	73.44018	1e-17
□	☑	is_weekend[0-1]	0.30571061	1	95.57213	1.4e-22
□	□	global_sentiment_polarity	0	1	0.092783	0.76067
□	☑	title_sentiment_polarity	-0.0968225	1	2.851602	0.09128

Figure 12. The Results of Logistic Regression



Nominal Logistic Fit for IsAtLeast2900Shares

Effect Summary

Source	LogWorth	PValue
Data Channel	40.658	0.00000
Prin1 Keyword Strength	26.536	0.00000
is_weekend	21.520	0.00000
self_reference_avg_sharess	21.383	0.00000
Prin2 Keyword Strength	17.985	0.00000
Prin3 Keyword Strength	5.114	0.00001
num_hrefs	3.491	0.00032
average_token_length	3.086	0.00082
num_keywords	2.560	0.00275
n_tokens_content	1.532	0.02940
num_videos	1.220	0.06031
title_sentiment_polarity	0.610	0.24543

Converged in Gradient, 5 iterations

Iterations

Whole Model Test

Fit Details

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.0843	0.0722	-0.141	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.1471	0.1270	-0.252	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.6345	0.6429	0.6307	∑ -Log(ρ[j])/n
RMSE	0.4705	0.4743	0.4668	√ ∑(y[j]-ρ[j])²/n
Mean Abs Dev	0.4437	0.4480	0.4410	∑ |y[j]-ρ[j]|/n
Misclassification Rate	0.3507	0.3619	0.3233	∑ (ρ[j]≠ρMax)/n
N	9194	5537	7474	n

Confusion Matrix

Training

Actual IsAtLeast2900Shares	Predicted Count 1	 0
1	2640	1854
0	1370	3330

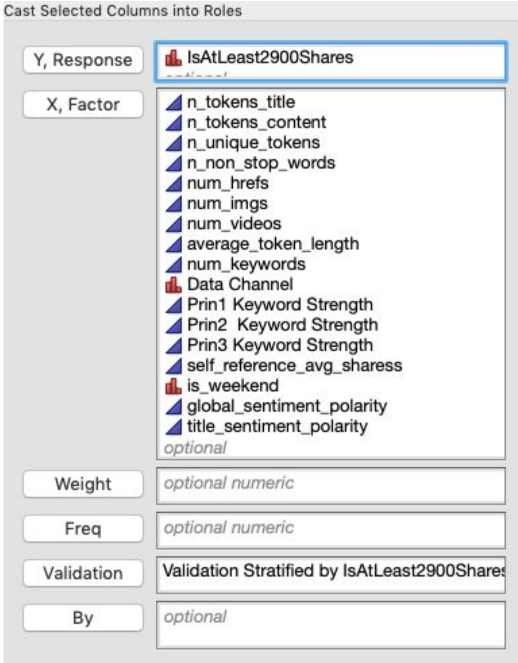
Validation

Actual IsAtLeast2900Shares	Predicted Count 1	 0
1	1526	1185
0	819	2007

Test

Actual IsAtLeast2900Shares	Predicted Count 1	 0
1	1048	758
0	1658	4010

Figure 13. The Results of Decision Tree



Cast Selected Columns into Roles

Y, Response: IsAtLeast2900Shares

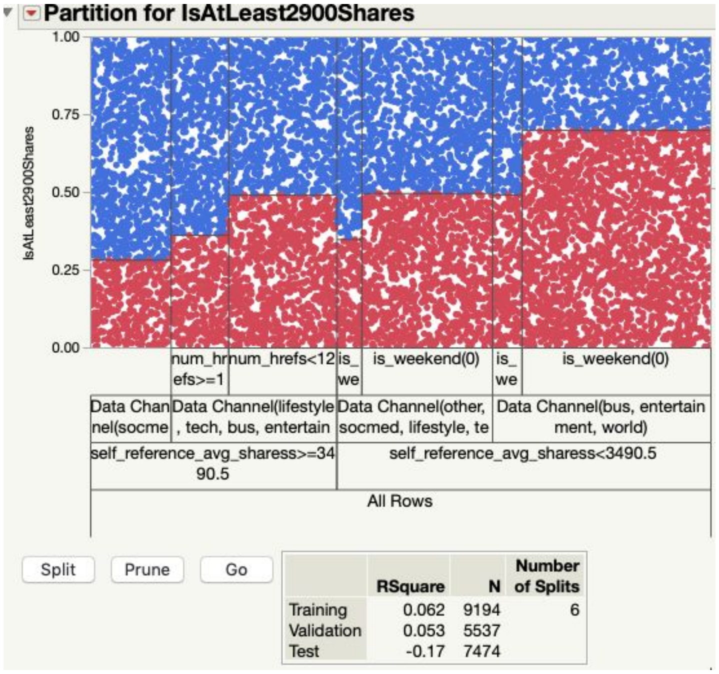
X, Factor:
n_tokens_title
n_tokens_content
n_unique_tokens
n_non_stop_words
num_hrefs
num_imgs
num_videos
average_token_length
num_keywords
Data Channel
Prin1 Keyword Strength
Prin2 Keyword Strength
Prin3 Keyword Strength
self_reference_avg_sharess
is_weekend
global_sentiment_polarity
title_sentiment_polarity
optional

Weight: optional numeric

Freq: optional numeric

Validation: Validation Stratified by IsAtLeast2900Shares

By: optional

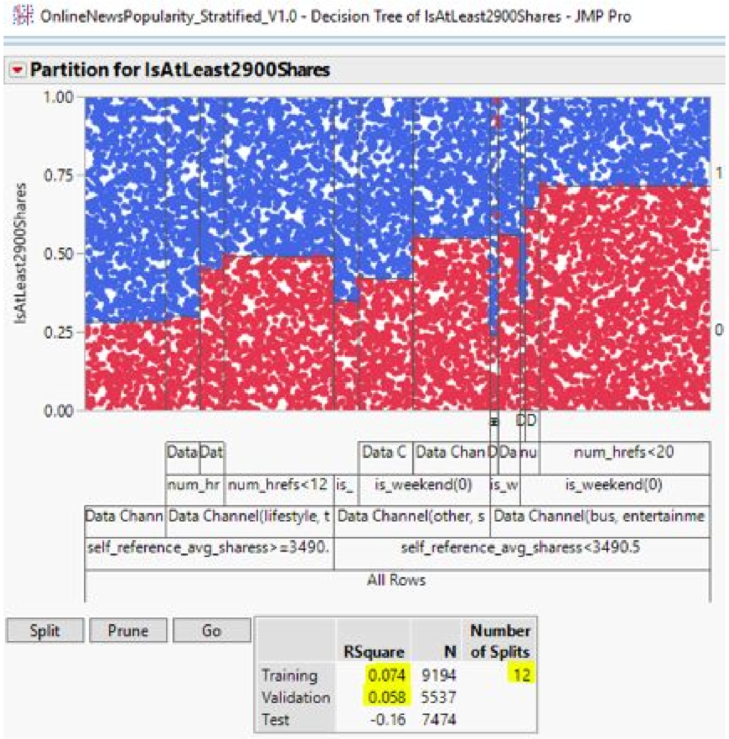


Partition for IsAtLeast2900Shares

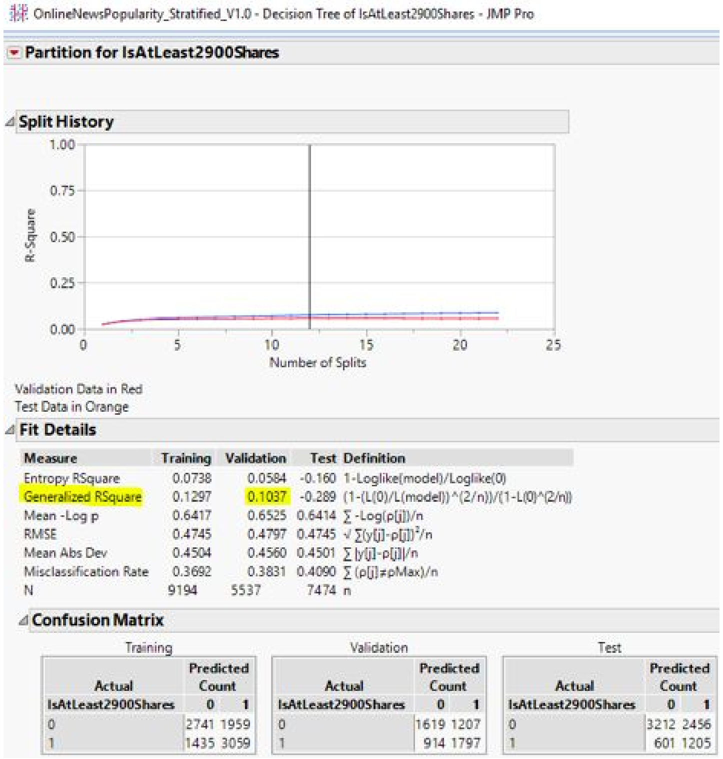
IsAtLeast2900Shares
1.00
0.75
0.50
0.25
0.00
num_hrefs>=1
num_hrefs<12
is_we
is_weekend(0)
is_we
is_weekend(0)
Data Channel(socme
Data Channel(lifestyle, tech, bus, entertain
Data Channel(other, socmed, lifestyle, te
Data Channel(bus, entertainment, world)
self_reference_avg_sharess>=3490.5
self_reference_avg_sharess<3490.5
All Rows
Split
Prune
Go

	RSquare	N	Number of Splits
Training	0.062	9194	6
Validation	0.053	5537	
Test	-0.17	7474	

Figure 13. The Results of Decision Tree (continued)


OnlineNewsPopularity_Stratified_V1.0 - Decision Tree of IsAtLeast2900Shares - JMP Pro
Partition for IsAtLeast2900Shares
IsAtLeast2900Shares
1.00
0.75
0.50
0.25
0.00
1
0
DD
Data
Dat
Data C
Data Chan
D
Da
nu
num_hrefs<20
num_hr
num_hrefs<12
is_
is_weekend(0)
is_w
is_weekend(0)
Data Chann
Data Channel(lifestyle, t
Data Channel(other, s
Data Channel(bus, entertainme
self_reference_avg_sharess>=3490.
self_reference_avg_sharess<3490.5
All Rows
Split
Prune
Go

	RSquare	N	Number of Splits
Training	0.074	9194	12
Validation	0.058	5537	
Test	-0.16	7474	


OnlineNewsPopularity_Stratified_V1.0 - Decision Tree of IsAtLeast2900Shares - JMP Pro
Partition for IsAtLeast2900Shares
Split History
R-Square
1.00
0.75
0.50
0.25
0.00
0
5
10
15
20
25
Number of Splits

Validation Data in Red
Test Data in Orange

Fit Details

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.0738	0.0584	-0.160	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.1297	0.1037	-0.289	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.6417	0.6525	0.6414	∑ -Log(ρ[j])/n
RMSE	0.4745	0.4797	0.4745	√ ∑(y[j]-ρ[j])²/n
Mean Abs Dev	0.4504	0.4560	0.4501	∑ |y[j]-ρ[j]|/n
Misclassification Rate	0.3692	0.3831	0.4090	∑ (ρ[j]≠ρMax)/n
N	9194	5537	7474	n

Confusion Matrix

Training

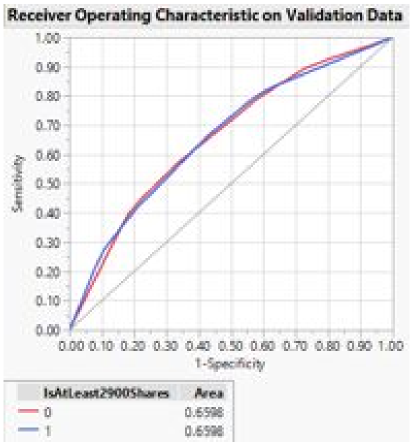
Actual	Predicted Count	
IsAtLeast2900Shares	0	1
0	2741	1959
1	1435	3059

Validation

Actual	Predicted Count	
IsAtLeast2900Shares	0	1
0	1619	1207
1	914	1797

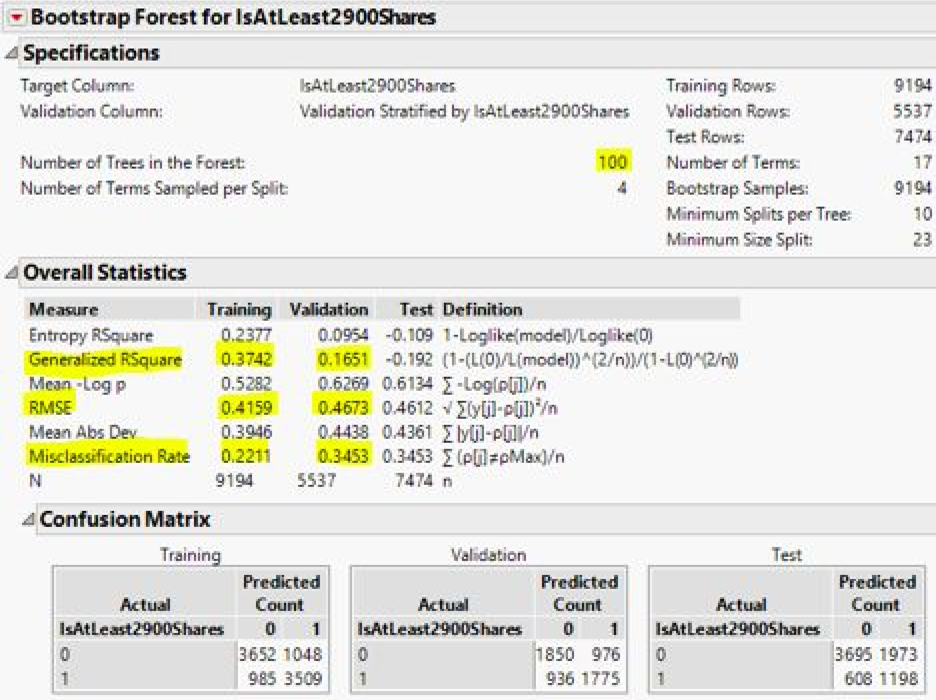
Test

Actual	Predicted Count	
IsAtLeast2900Shares	0	1
0	3212	2456
1	601	1205



IsAtLeast2900Shares	Area
0	0.6598
1	0.6598

Figure 14. The Results of Bootstrap Forest



Bootstrap Forest for IsAtLeast2900Shares

Specifications

Target Column:	IsAtLeast2900Shares	Training Rows:	9194
Validation Column:	Validation Stratified by IsAtLeast2900Shares	Validation Rows:	5537
		Test Rows:	7474
Number of Trees in the Forest:	100	Number of Terms:	17
Number of Terms Sampled per Split:	4	Bootstrap Samples:	9194
		Minimum Splits per Tree:	10
		Minimum Size Split:	23

Overall Statistics

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.2377	0.0954	-0.109	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.3742	0.1651	-0.192	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.5282	0.6269	0.6134	∑ -Log(ρ[j])/n
RMSE	0.4159	0.4673	0.4612	$\sqrt{\sum(y[j]-\rho[j])^2/n}$
Mean Abs Dev	0.3946	0.4438	0.4361	∑ |y[j]-ρ[j]|/n
Misclassification Rate	0.2211	0.3453	0.3453	∑ (ρ[j]≠ρMax)/n
N	9194	5537	7474	n

Confusion Matrix

Training

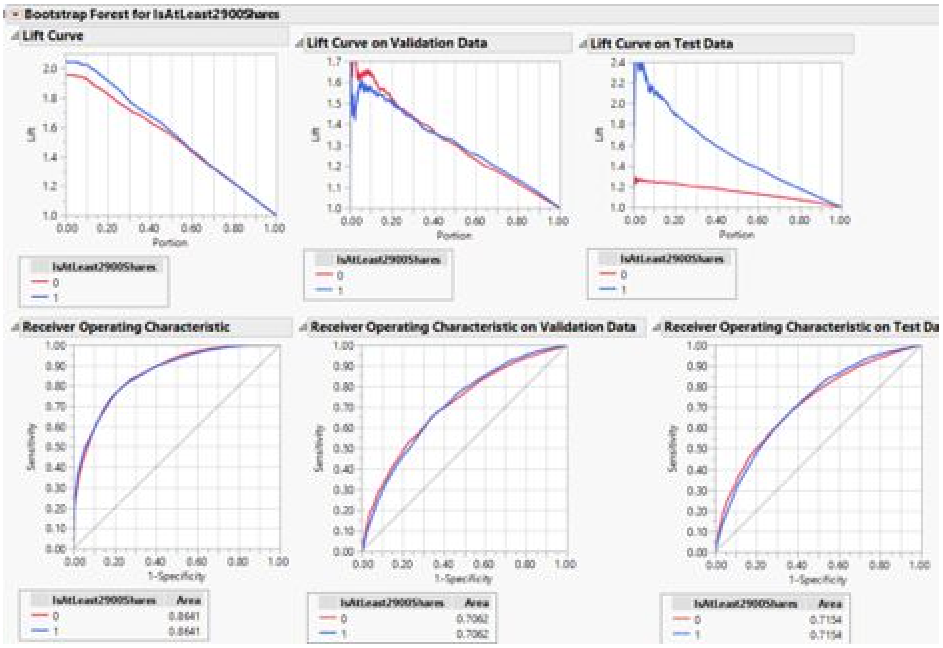
Actual	Predicted Count	
IsAtLeast2900Shares	0	1
0	3652	1048
1	985	3509

Validation

Actual	Predicted Count	
IsAtLeast2900Shares	0	1
0	1850	976
1	936	1775

Test

Actual	Predicted Count	
IsAtLeast2900Shares	0	1
0	3695	1973
1	608	1198



Receiver Operating Characteristic

IsAtLeast2900Shares	Area
0	0.8641
1	0.8641

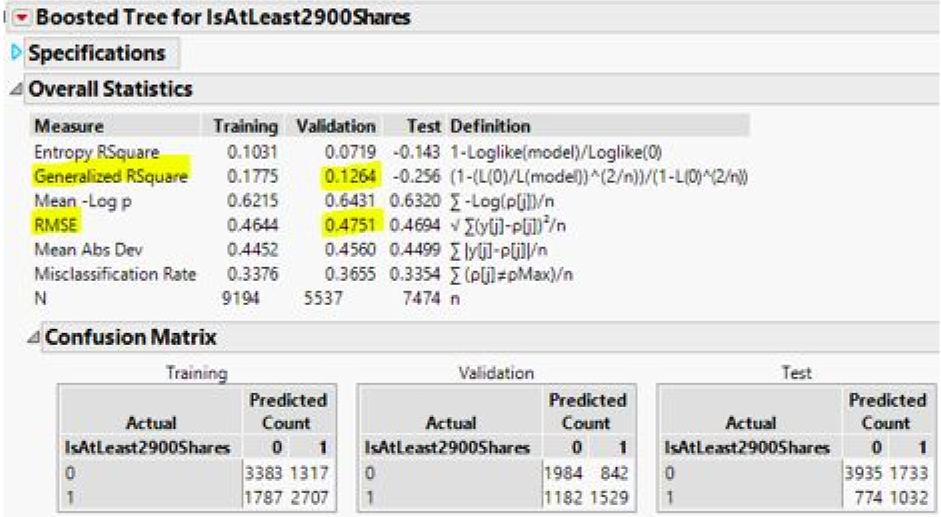
Receiver Operating Characteristic on Validation Data

IsAtLeast2900Shares	Area
0	0.7062
1	0.7062

Receiver Operating Characteristic on Test Da

IsAtLeast2900Shares	Area
0	0.7154
1	0.7154

Figure 15. The Results of Boosted Tree



Boosted Tree for IsAtLeast2900Shares

Specifications

Overall Statistics

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.1031	0.0719	-0.143	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.1775	0.1264	-0.256	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.6215	0.6431	0.6320	∑ -Log(ρ[j])/n
RMSE	0.4644	0.4751	0.4694	√ ∑(y[j]-ρ[j])²/n
Mean Abs Dev	0.4452	0.4560	0.4499	∑ |y[j]-ρ[j]|/n
Misclassification Rate	0.3376	0.3655	0.3354	∑ (ρ[j]≠ρMax)/n
N	9194	5537	7474	n

Confusion Matrix

Training

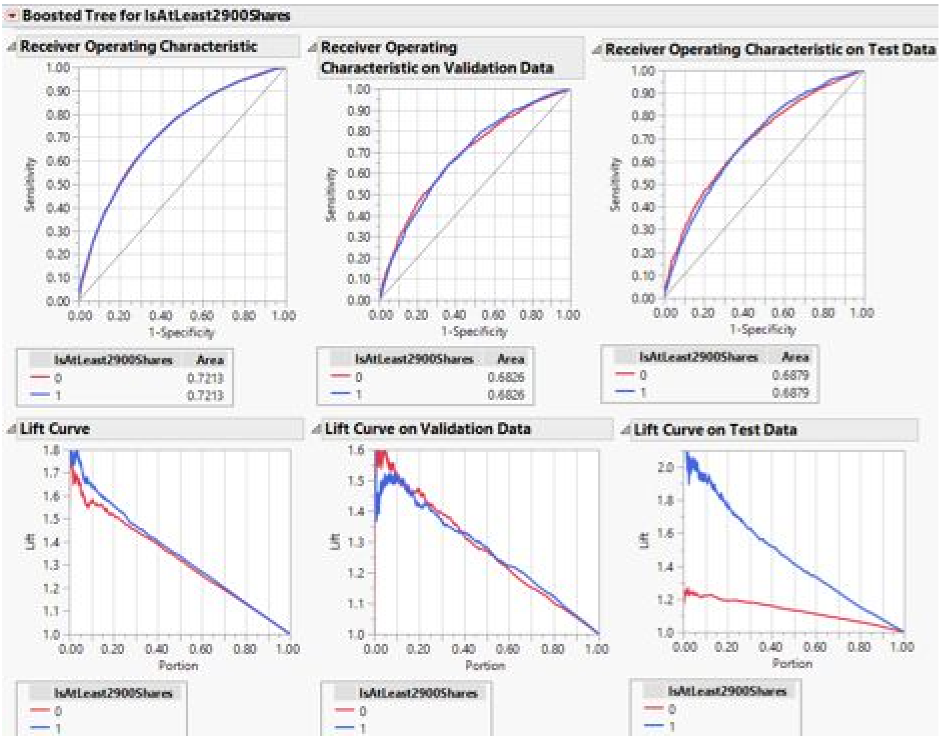
Actual	Predicted Count	
IsAtLeast2900Shares	0	1
0	3383	1317
1	1787	2707

Validation

Actual	Predicted Count	
IsAtLeast2900Shares	0	1
0	1984	842
1	1182	1529

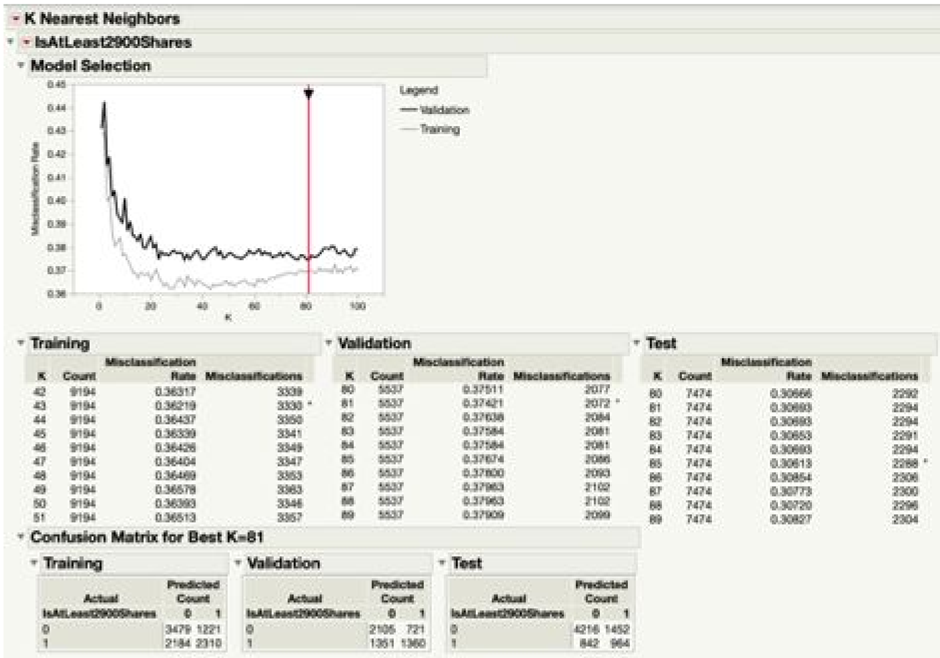
Test

Actual	Predicted Count	
IsAtLeast2900Shares	0	1
0	3935	1733
1	774	1032



Boosted Tree for IsAtLeast2900Shares

Figure 16. The Results of K-Nearest Neighbor



K Nearest Neighbors

IsAtLeast2900Shares

Model Selection

Training

K	Count	Misclassification Rate	Misclassifications
42	9194	0.36317	3339
43	9194	0.36219	3330 *
44	9194	0.36437	3350
45	9194	0.36339	3341
46	9194	0.36426	3349
47	9194	0.36404	3347
48	9194	0.36469	3353
49	9194	0.36578	3363
50	9194	0.36393	3346
51	9194	0.36513	3357

Validation

K	Count	Misclassification Rate	Misclassifications
80	5537	0.37511	2077
81	5537	0.37421	2072 *
82	5537	0.37638	2084
83	5537	0.37584	2081
84	5537	0.37584	2081
85	5537	0.37674	2086
86	5537	0.37800	2093
87	5537	0.37963	2102
88	5537	0.37963	2102
89	5537	0.37909	2099

Test

K	Count	Misclassification Rate	Misclassifications
80	7474	0.30666	2292
81	7474	0.30693	2294
82	7474	0.30693	2294
83	7474	0.30653	2291
84	7474	0.30693	2294
85	7474	0.30613	2288 *
86	7474	0.30854	2306
87	7474	0.30773	2300
88	7474	0.30720	2296
89	7474	0.30827	2304

Confusion Matrix for Best K=81

Training

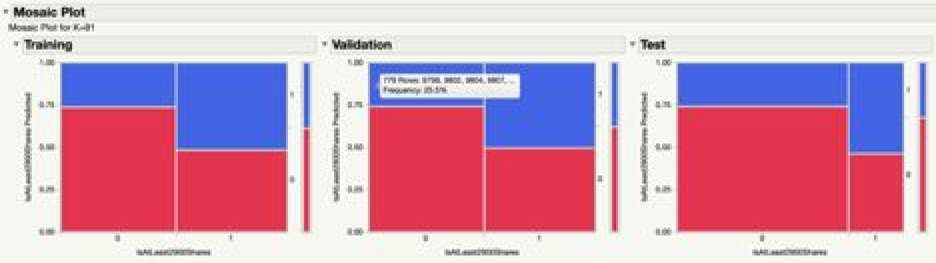
Actual IsAtLeast2900Shares	Predicted Count 0	1
0	3479	1221
1	2184	2310

Validation

Actual IsAtLeast2900Shares	Predicted Count 0	1
0	2105	721
1	1351	1360

Test

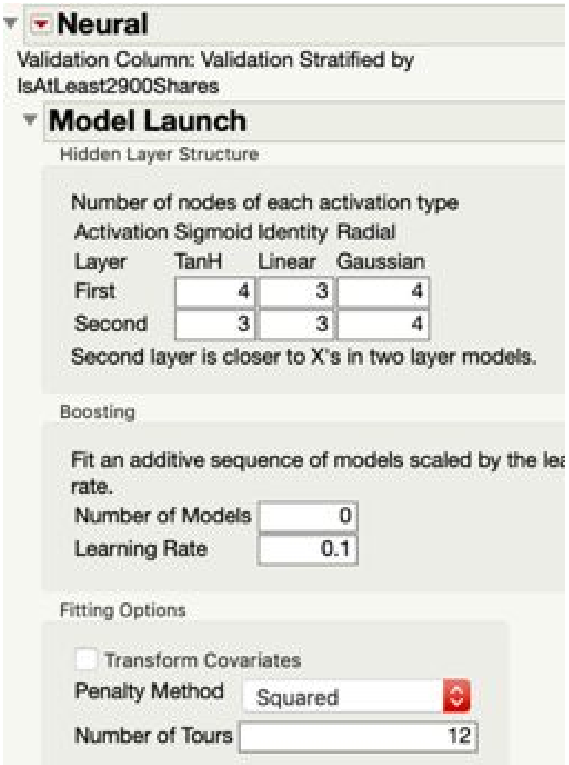
Actual IsAtLeast2900Shares	Predicted Count 0	1
0	4216	1452
1	842	964



Mosaic Plot

Mosaic Plot for K=81

Figure 17. The Results of Neural Network



Neural

Validation Column: Validation Stratified by IsAtLeast2900Shares

Model Launch

Hidden Layer Structure

Number of nodes of each activation type

Activation	Sigmoid	Identity	Radial
Layer	TanH	Linear	Gaussian
First	4	3	4
Second	3	3	4

Second layer is closer to X's in two layer models.

Boosting

Fit an additive sequence of models scaled by the lea rate.

Number of Models 0

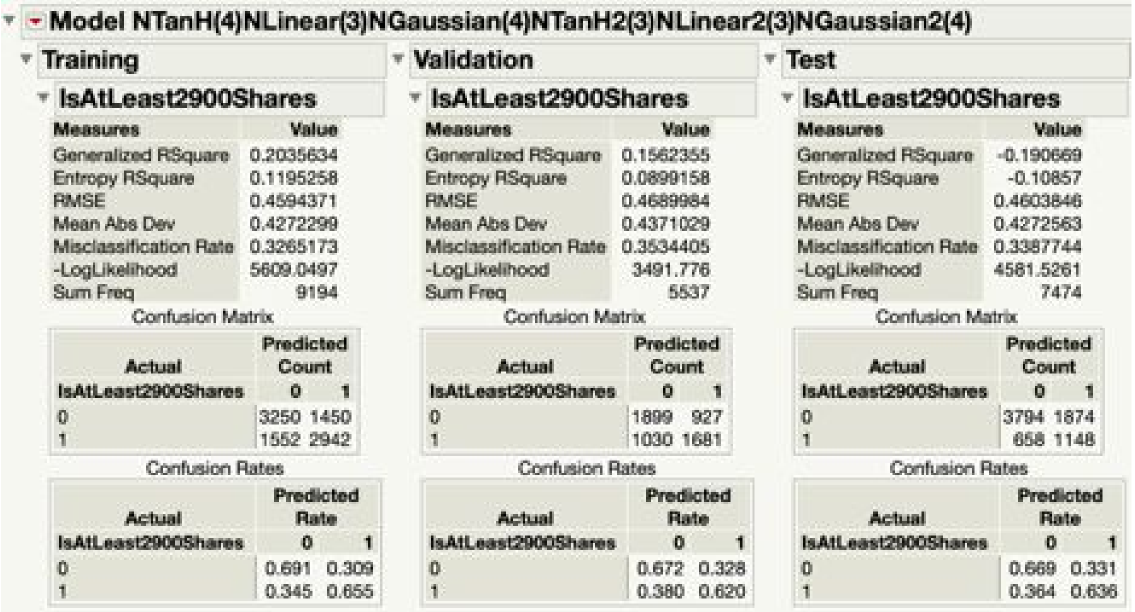
Learning Rate 0.1

Fitting Options

□ Transform Covariates

Penalty Method Squared

Number of Tours 12



Model NTanH(4)NLinear(3)NGaussian(4)NTanH2(3)NLinear2(3)NGaussian2(4)

Training

IsAtLeast2900Shares

Measures	Value
Generalized RSquare	0.2035634
Entropy RSquare	0.1195258
RMSE	0.4594371
Mean Abs Dev	0.4272299
Misclassification Rate	0.3265173
-LogLikelihood	5609.0497
Sum Freq	9194

Confusion Matrix

Actual	Predicted Count	
IsAtLeast2900Shares	0	1
0	3250	1450
1	1552	2942

Confusion Rates

Actual	Predicted Rate	
IsAtLeast2900Shares	0	1
0	0.691	0.309
1	0.345	0.655

Validation

IsAtLeast2900Shares

Measures	Value
Generalized RSquare	0.1562355
Entropy RSquare	0.0899158
RMSE	0.4689984
Mean Abs Dev	0.4371029
Misclassification Rate	0.3534405
-LogLikelihood	3491.776
Sum Freq	5537

Confusion Matrix

Actual	Predicted Count	
IsAtLeast2900Shares	0	1
0	1899	927
1	1030	1681

Confusion Rates

Actual	Predicted Rate	
IsAtLeast2900Shares	0	1
0	0.672	0.328
1	0.380	0.620

Test

IsAtLeast2900Shares

Measures	Value
Generalized RSquare	-0.190669
Entropy RSquare	-0.10857
RMSE	0.4603846
Mean Abs Dev	0.4272563
Misclassification Rate	0.3387744
-LogLikelihood	4581.5261
Sum Freq	7474

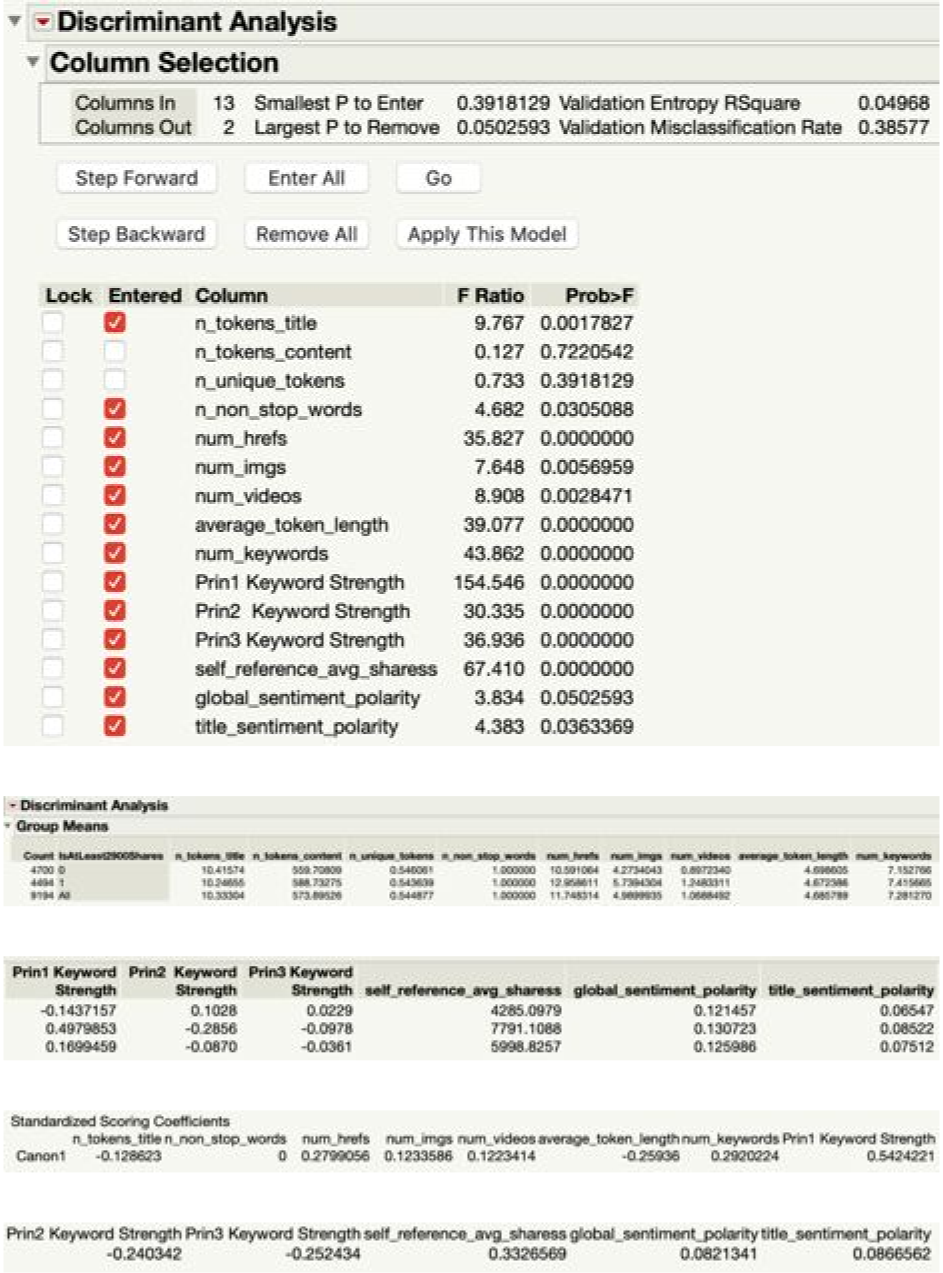
Confusion Matrix

Actual	Predicted Count	
IsAtLeast2900Shares	0	1
0	3794	1874
1	658	1148

Confusion Rates

Actual	Predicted Rate	
IsAtLeast2900Shares	0	1
0	0.669	0.331
1	0.364	0.636

Figure 18. The Results of Linear Discriminant Analysis



Discriminant Analysis

Column Selection

Columns In	13	Smallest P to Enter	0.3918129	Validation Entropy RSquare	0.04968
Columns Out	2	Largest P to Remove	0.0502593	Validation Misclassification Rate	0.38577

Lock	Entered	Column	F Ratio	Prob>F
□	☑	n_tokens_title	9.767	0.0017827
□	□	n_tokens_content	0.127	0.7220542
□	□	n_unique_tokens	0.733	0.3918129
□	☑	n_non_stop_words	4.682	0.0305088
□	☑	num_hrefs	35.827	0.0000000
□	☑	num_imgs	7.648	0.0056959
□	☑	num_videos	8.908	0.0028471
□	☑	average_token_length	39.077	0.0000000
□	☑	num_keywords	43.862	0.0000000
□	☑	Prin1 Keyword Strength	154.546	0.0000000
□	☑	Prin2 Keyword Strength	30.335	0.0000000
□	☑	Prin3 Keyword Strength	36.936	0.0000000
□	☑	self_reference_avg_sharess	67.410	0.0000000
□	☑	global_sentiment_polarity	3.834	0.0502593
□	☑	title_sentiment_polarity	4.383	0.0363369

Discriminant Analysis

Group Means

Count	IsAtLeast2900Shares	n_tokens_title	n_tokens_content	n_unique_tokens	n_non_stop_words	num_hrefs	num_imgs	num_videos	average_token_length	num_keywords
4700	0	10.41574	559.70809	0.546061	1.000000	10.591064	4.2734043	0.8972340	4.698605	7.152766
4494	1	10.24655	588.73275	0.543639	1.000000	12.958611	5.7394304	1.2483311	4.672386	7.415665
9194	All	10.33304	573.89526	0.544877	1.000000	11.748314	4.9899935	1.0688492	4.685789	7.281270

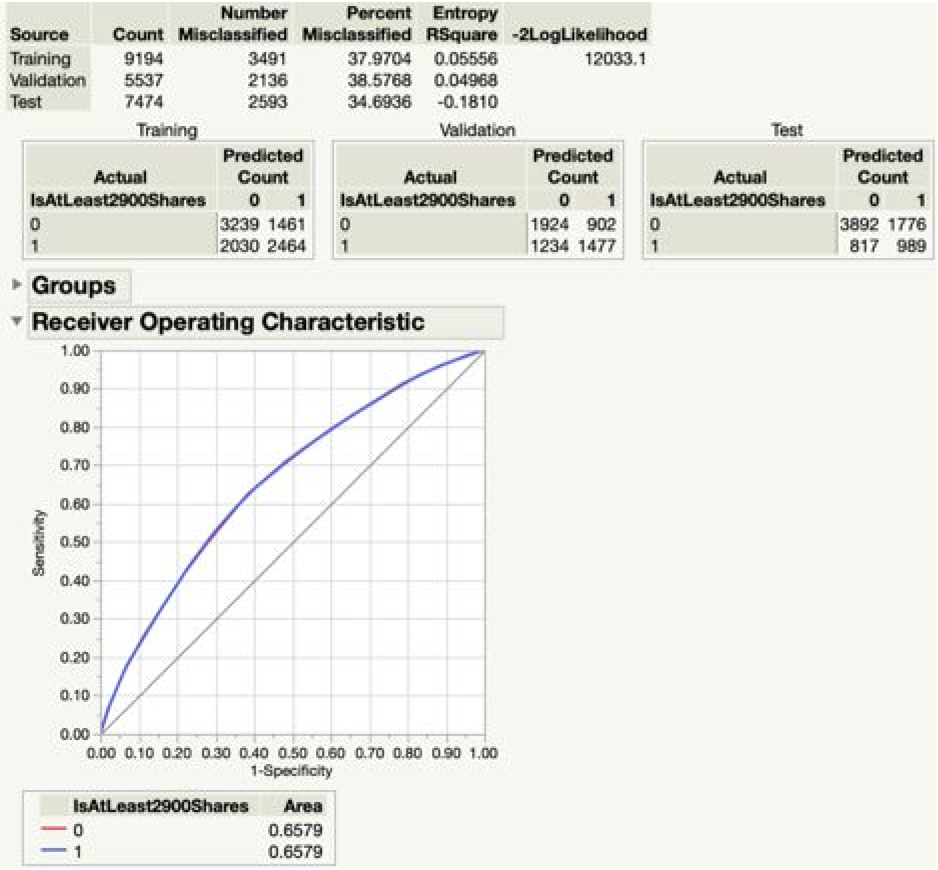
Prin1 Keyword Strength	Prin2 Keyword Strength	Prin3 Keyword Strength	self_reference_avg_sharess	global_sentiment_polarity	title_sentiment_polarity
-0.1437157	0.1028	0.0229	4285.0979	0.121457	0.06547
0.4979853	-0.2856	-0.0978	7791.1088	0.130723	0.08522
0.1699459	-0.0870	-0.0361	5998.8257	0.125986	0.07512

Standardized Scoring Coefficients

	n_tokens_title	n_non_stop_words	num_hrefs	num_imgs	num_videos	average_token_length	num_keywords	Prin1 Keyword Strength
Canon1	-0.128623	0	0.2799056	0.1233586	0.1223414	-0.25936	0.2920224	0.5424221

Prin2 Keyword Strength	Prin3 Keyword Strength	self_reference_avg_sharess	global_sentiment_polarity	title_sentiment_polarity
-0.240342	-0.252434	0.3326569	0.0821341	0.0866562

Figure 18. The Results of Linear Discriminant Analysis (continued)



Source	Count	Number Misclassified	Percent Misclassified	Entropy RSquare	-2LogLikelihood
Training	9194	3491	37.9704	0.05556	12033.1
Validation	5537	2136	38.5768	0.04968	
Test	7474	2593	34.6936	-0.1810	

Training

Actual IsAtLeast2900Shares	Predicted Count 0	1
0	3239	1461
1	2030	2464

Validation

Actual IsAtLeast2900Shares	Predicted Count 0	1
0	1924	902
1	1234	1477

Test

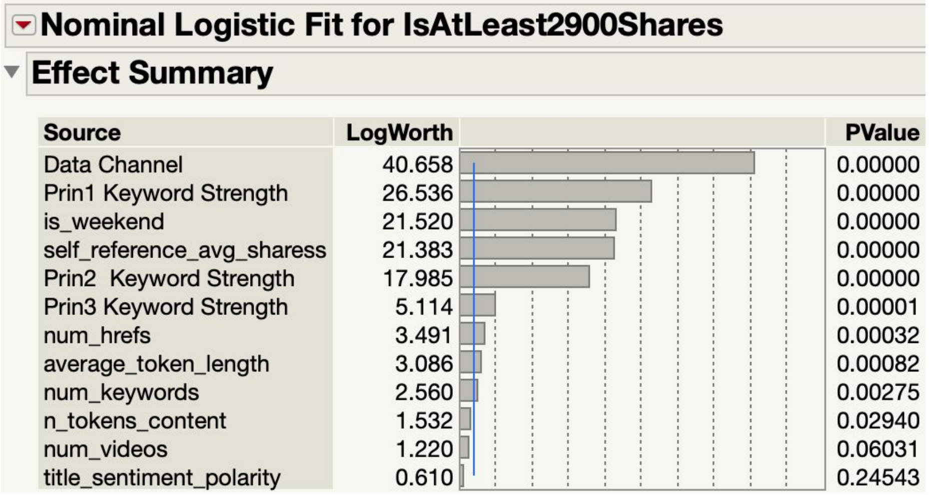
Actual IsAtLeast2900Shares	Predicted Count 0	1
0	3892	1776
1	817	989

▸ Groups

▾ Receiver Operating Characteristic

IsAtLeast2900Shares	Area
🟥 0	0.6579
🟦 1	0.6579

Figure 19. Best Model Interpretation

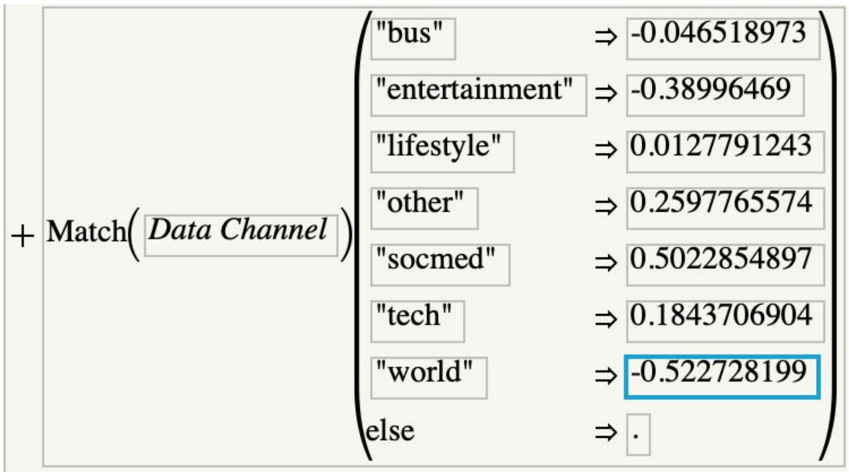


Nominal Logistic Fit for IsAtLeast2900Shares

Effect Summary

Source	LogWorth	PValue
Data Channel	40.658	0.00000
Prin1 Keyword Strength	26.536	0.00000
is_weekend	21.520	0.00000
self_reference_avg_sharess	21.383	0.00000
Prin2 Keyword Strength	17.985	0.00000
Prin3 Keyword Strength	5.114	0.00001
num_hrefs	3.491	0.00032
average_token_length	3.086	0.00082
num_keywords	2.560	0.00275
n_tokens_content	1.532	0.02940
num_videos	1.220	0.06031
title_sentiment_polarity	0.610	0.24543

Statistical Significance



```
+ Match( Data Channel )  "bus"           ⇒ -0.046518973
                         "entertainment" ⇒ -0.38996469
                         "lifestyle"     ⇒ 0.0127791243
                         "other"         ⇒ 0.2597765574
                         "socmed"        ⇒ 0.5022854897
                         "tech"          ⇒ 0.1843706904
                         "world"         ⇒ -0.522728199
                         else            ⇒ .
```

Coefficients for specific Data Channels