StumbleUpon Case Report

Dave Arrigg, Malia Hariz, Shuang Xu Li, Amrita Nair

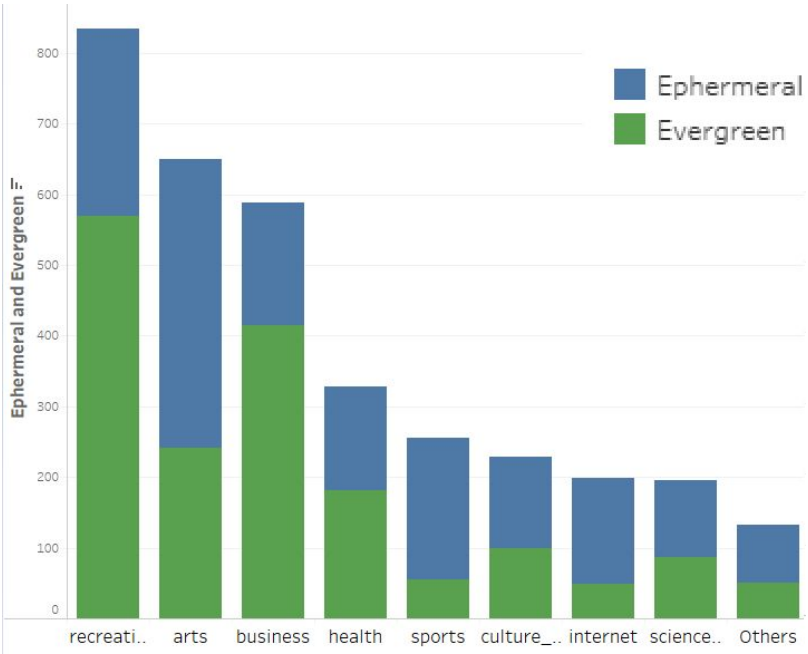
Executive Summary:

As consultants working for StumbleUpon, our main objective is to help improve their backend algorithm to sort out which pages will remain interesting in the long term (“evergreen” pages) and which will quickly become either outdated or irrelevant (“ephemeral” pages). Throughout the case, we conducted several iterations of data exploration, data cleansing, model creation, and performance evaluations. The StumbleUpon data given to us has attributes of the different website attributes like the alchemy category, length of the article, spelling error ratio, compression ratio, etc.

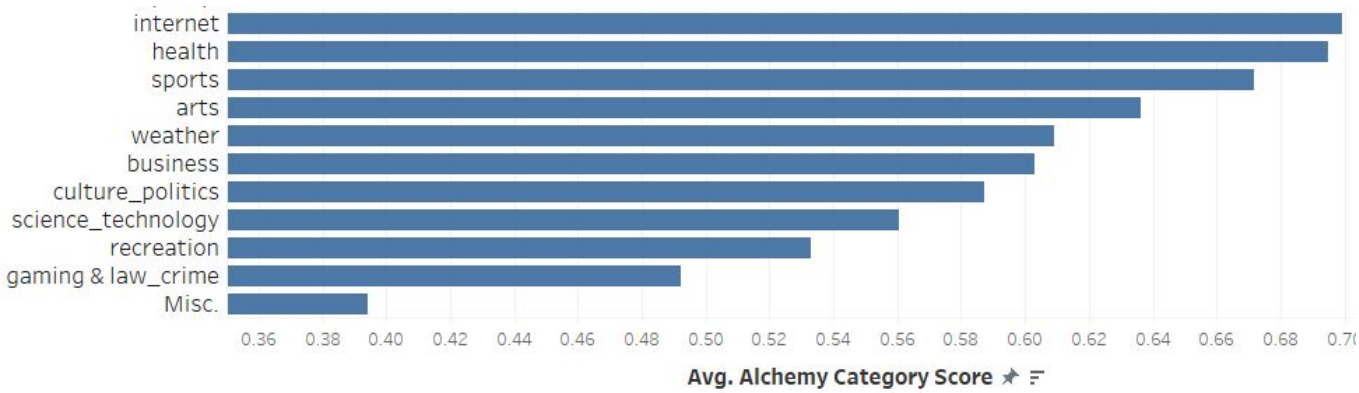
Using this data, we created several different machine learning models and evaluated the performance of each to determine which method or combination of methods is the best approach. After much consideration, we decided to use a random forest model based on the accuracy of its predictions of whether a page is ephemeral or evergreen. To improve the customer experience of their users, StumbleUpon will deploy our random forest model to their servers so that they can accurately filter in the evergreen pages on their website.

Data Exploration Summary:

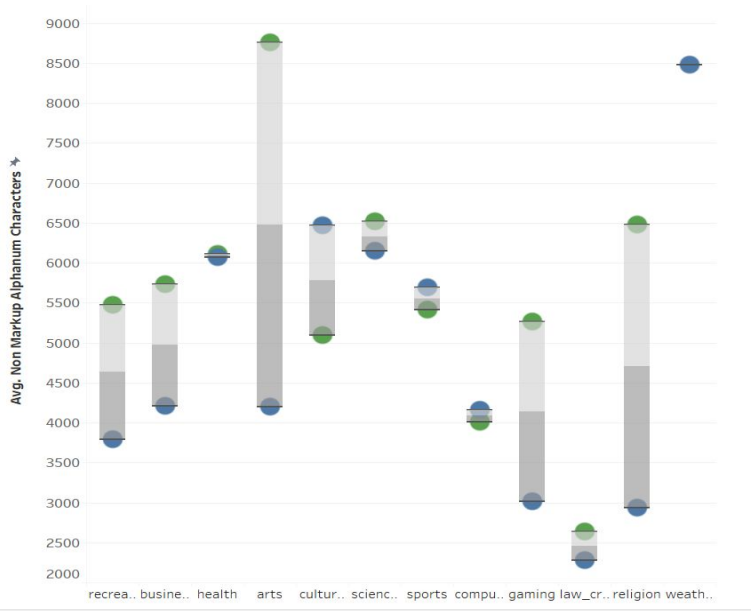
The attribute *label* signifies whether a page is an evergreen (if its value is 1) or ephemeral (if its value is 0). The machine learning model that we developed categorizes pages with respect to the target variable *label*. We explored the data to see how each page attribute affects the *label* and our significant findings are shown below:



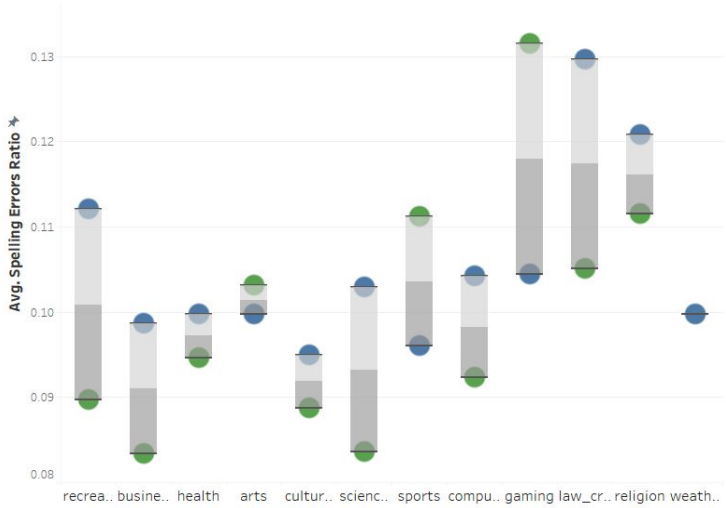
Alchemy Category: When evaluating evergreen content by industry, we noticed that recreation, business, and arts related articles are more likely to remain relevant. This makes sense because the other categories are more likely to be event-related articles that will lose traction with time (eg: Superbowl, religious event, new technology innovations, etc). The question is: How accurately did Alchemy categorize the pages?



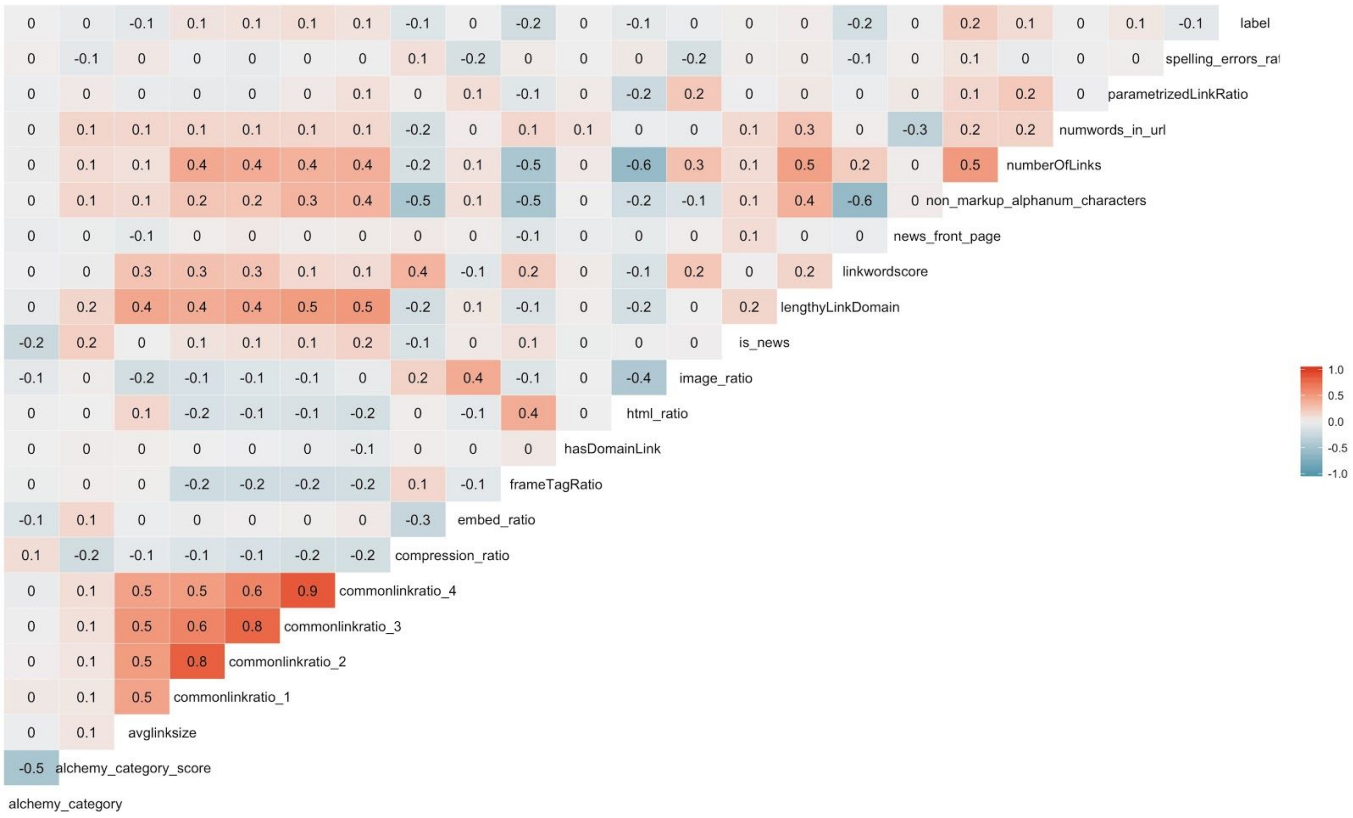
Alchemy Category Score: We have more confidence in predictions for some *alchemy categories* than others.



Length of Article:
In this boxplot, green here signifies average alphanumeric characters for evergreen pages and blue signifies the same for ephemeral pages. On average, we see that evergreen pages have more *non-markup alphanumeric characters*. This could be because the web crawlers direct the flow of traffic to lengthy pages.



Spelling Error Ratio:
For most categories *spelling error ratio* affects whether the page is evergreen or not. According to the data dictionary, a spelling error is considered as those words that are not available in Wikipedia or are unfamiliar names, technical jargon or slang words.



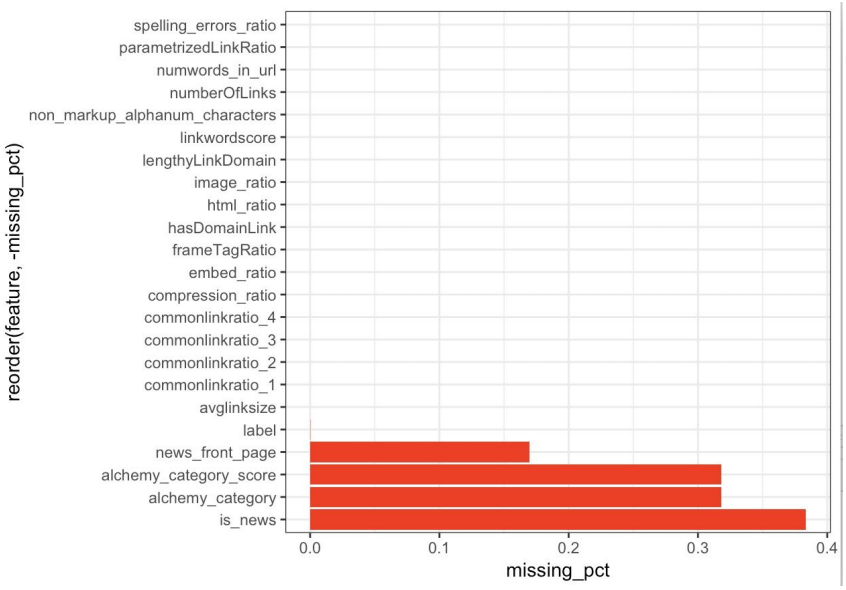
The correlation heat map shows the mutual relationship between each attribute of the data provided. Red signifies that there is a strong positive correlation between the 2 attributes, whereas blue signifies a negative correlation. From the map above, our target *label* does not have a strong correlation with any attribute. We find there is a high correlation among the four *common link ratio* related columns. We will retain them when building our model because we find that including all of the attributes will result in a more robust model.

Data Pre-Processing and Cleaning Strategy:

At an initial glance, the dataset provided by StumbleUpon has 5000 observations and 27 variables, and it fell into two categories: 3 text and 24 numeric data fields. The *boilerplate* text could have been useful for our prediction if we took the time to perform text analysis, but we decide to focus on numeric features first in the current phase. Thus, we excluded those columns containing text data, such as *URL, URLID, boilerplate*.

The rest of the dataset can be categorized into two subsets: one relates to the amount of embedded media (e.g., *compression ratio, embedded ratio*), and the other is about the content of embedded media

(e.g., *is news, alchemy category*, and *hasDomainLink*). The value in frameBased is constant 0, so we exclude this column in the dataset as well.



We then looked for special characters for each attribute. We found that there are missing values that occurred for four attributes in the form of question marks “?”. We replaced all of them with null values and visualized the percentage of missing values for each column as shown here.

For the target *label*, there is only one observation that had '1?’ special value that we decided to regard it as 1. There were 1589 records that had missing values in the *alchemy category* and *alchemy category score* columns. We found that *alchemy category* has a value named as unknown with a corresponding score 0.4. We replaced all null values in *alchemy category* as unknown and all missing value in *alchemy category score* with 0.4.

For *is_news* column, the missing value accounts for 38% of total rows. The remaining observations have an *is_news* value of 1. If we used the imputation method, the missing values would be replaced with 1. This is unreasonable because not all pages are news articles. We therefore assume that this is an input error and replace all of the missing value with 0. The meaning of this column has changed for us - if it has the value ‘1’, we are sure it is a news article; if it has the value 0, we are unsure whether it is a news article.

For *news_front_page* column, there are around 17% missing values in it. Approximately 80% of rows are shown as 1, and the remaining is shown as 0. We decide to use the imputation method to fill in these missing value by applying the Random Forest algorithm.

To improve the performance of all our models, we created a series of dummy variables for categorical features such as *alchemy_category, hasDomainLink, is_news, lengthyLinkDomain, news_front_page*. Take *alchemy_category* column as an example: we created a new column named *alchemy_category_health* for its factor value health, and this new coulumn’s value would be 1 for its corresponding rows in the original *alchemy_category* column. Otherwise, it would be 0. This process will bring better performance for some algorithms.

Model Development:

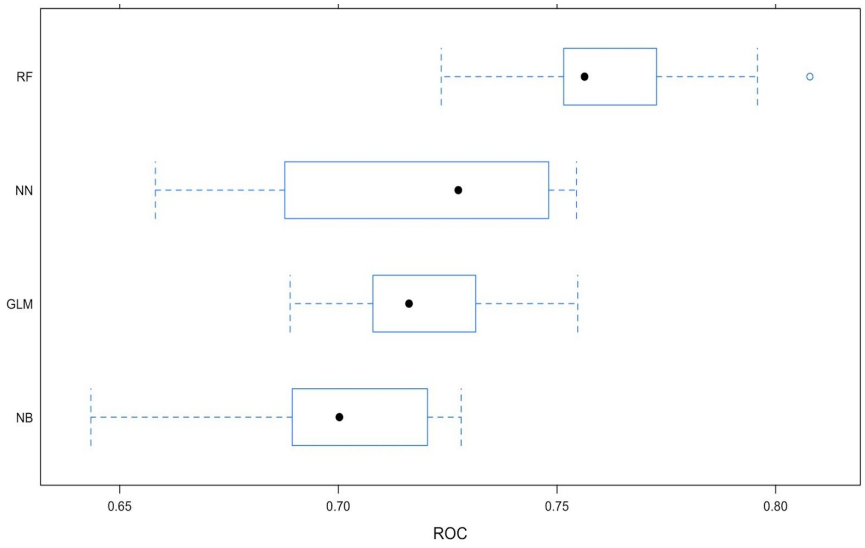
We created predictions with four different machine learning algorithms: Random Forest, Neural Network, Logistic Regression and Naive Bayes. For each model, we preprocessed the data with scaling and used ten cross-validations to improve the predictive power when the model is given new data. We understand that adding features past a certain point can be detrimental depending on the algorithm we use:

- Tree algorithms like Random Forest and Decision Trees are pretty stable in performance regardless of the number of correlated and/or noisy attributes. (Source: Analysis of RF Model)
- Regression based algorithms and Neural nets on the other hand, are extremely sensitive, because they are forced to infer relationships based on every feature in the model.

Based on the above findings, we used the entire data set as the input for Random Forest model.

Model Performance and Interpretation:

We use ROC-AUC to evaluate the performance of our classification models. ROC shows us how many correct positive classifications can be gained for allowing more and more false positives. In this case, it means how many correct predictions of evergreen pages can be achieved for allowing more and more wrong evergreen predictions. AUC is the area under the curve, and it summarizes the model's performance in a single number.



The box plot above measures the AUC value (indicated on the horizontal axis as ROC) of the ROC curve of each model. All predicted values are obtained by default with a threshold of 0.5. We see that Neural Network and Naive Bayes models have a significant variability depending on the processed sample. The Random Forest model achieves the highest performance score with relatively lower variability.

The comparison of misclassification rates is as below. Precision (aka Positive Predictive Value) refers to the proportion of correctly predicted evergreen pages out of all the webpages that were predicted as evergreen. Recall (aka Sensitivity) refers to the proportion of correctly predicted evergreen pages out of all the actual evergreen webpages. F1 score is the weighted average of Precision and Recall. The ideal

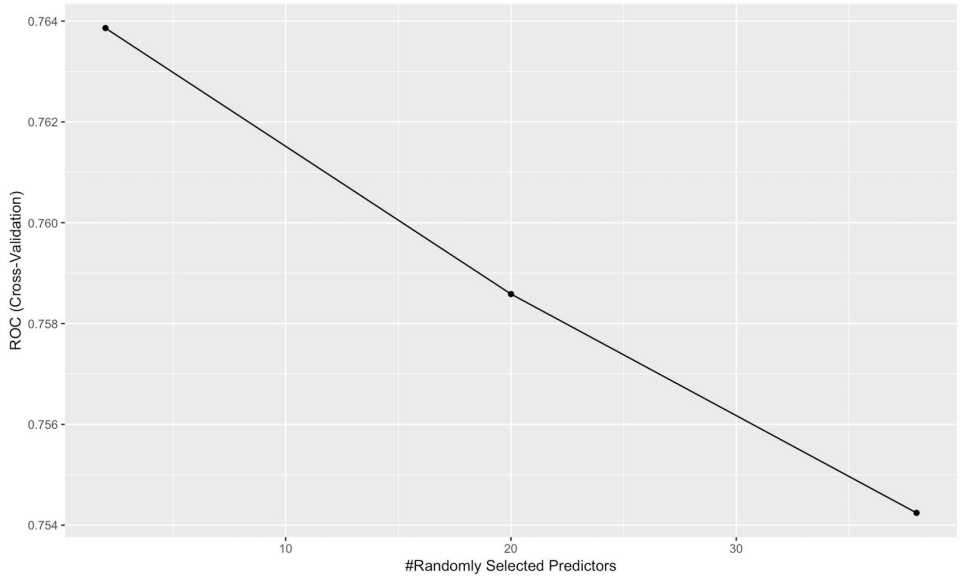
F1 score value is 1. From the table below we can see that the best model in term of F1 is the Random Forest.

	RF	NN	NB	GLM
Sensitivity	0.7392996	0.6358268	0.99212598	0.6850394
Specificity	0.6502058	0.6646341	0.01626016	0.6260163
Pos Pred Value	0.6909091	0.6618852	0.51012146	0.6541353
Neg Pred Value	0.7022222	0.6386719	0.66666667	0.6581197
Precision	0.6909091	0.6618852	0.51012146	0.6541353
Recall	0.7392996	0.6358268	0.99212598	0.6850394
F1	0.7142857	0.6485944	0.67379679	0.6692308
Prevalence	0.5140000	0.5080000	0.50800000	0.5080000
Detection Rate	0.3800000	0.3230000	0.50400000	0.3480000
Detection Prevalence	0.5500000	0.4880000	0.98800000	0.5320000
Balanced Accuracy	0.6947527	0.6502305	0.50419307	0.6555278

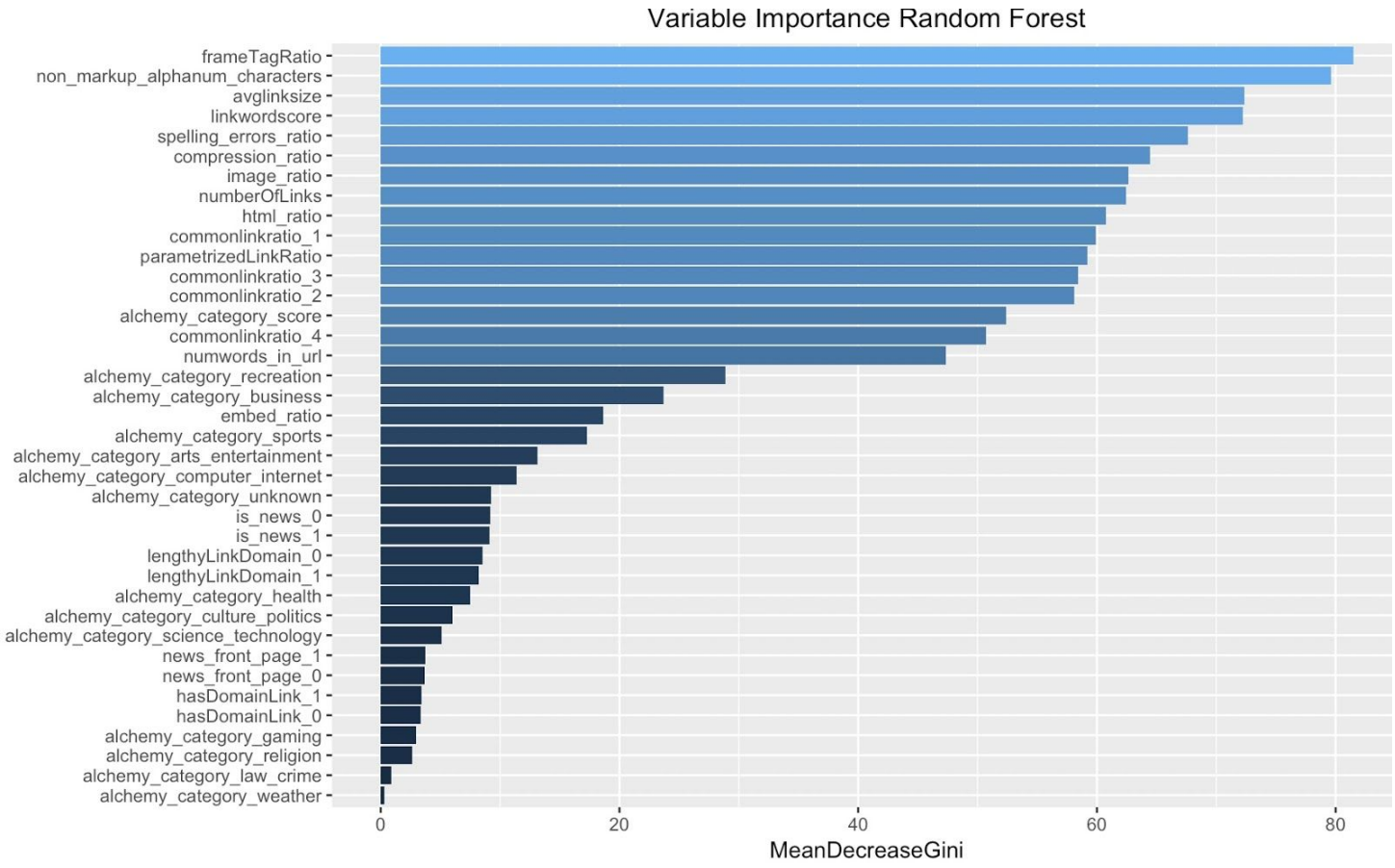
```
> output_report
                 metric best_model     value
1           Sensitivity         NB 0.9921260
2           Specificity         NN 0.6646341
3        Pos Pred Value         RF 0.6909091
4        Neg Pred Value         RF 0.7022222
5             Precision         RF 0.6909091
6                Recall         NB 0.9921260
7                    F1         RF 0.7142857
8            Prevalence         RF 0.5140000
9        Detection Rate         NB 0.5040000
10 Detection Prevalence         NB 0.9880000
11    Balanced Accuracy         RF 0.6947527
```

Model Evaluation:

We have decided to use the Random Forest model as it has the best performance. On investigating further, we found that as the number of randomly selected variables increases, the AUC score (indicated as vertical axis ROC) decreases. The model reaches the best performance when the number is 2 or 3.



The Random Forest model also ranked the variable importance in the modeling process. The top 2 is frame *FrameTagRatio* which is the ratio of iframe markups over a total number of markups, and the *non_markup_aphanum_characters*, which is the page's number of alphanumeric characters. The feature ranking also indicates that if the web content is about recreation, business and sports, it will have a higher probability of being evergreen.



Conclusion:

We will recommend Random Forest model to StumbleUpon because it has the highest accuracy compared to all the other models we investigated in terms of correctly classifying the pages as evergreen and ephemeral. We would recommend this model also because it is less prone to overfitting thanks to random feature selection and cross validations using subsets of the training dataset.

Based on our findings in the first phase, we find that the quantitative attributes of embedded media (eg: ratio of images to content, ratio of external links to content, etc) and the metrics of the page's content (eg: attributes related to length of page, speed of page, etc) have only moderate predictive power, probably because they appear in virtually all the web pages.

Our model predictive power would have been better if we had accurately categorized data (*alchemy_category, is_news* and *news_frontpage*). Considering that we predicted and estimated these attributes, we predicted the target *label* using predicted attributes - this would create noise in the data. Also, we are not sure about the time frame of an evergreen page. Would an evergreen page be evergreen indefinitely or does it have an "expiration date"? If we had such data, we would be able to rank the evergreen pages so that StumbleUpon would give preference to highly ranked pages.

In our next phase, we will dive into content features (ie. *Boilerplate* text), and explore text analysis methods such as TF-IDF ('Term Frequency-Inverse Document Frequency') to understand the text's relationship with webpage longevity.

Appendix:

Alternative Techniques Investigated:

Reducing Complexity of Models using Principal Component Analysis and Dropping Variables:

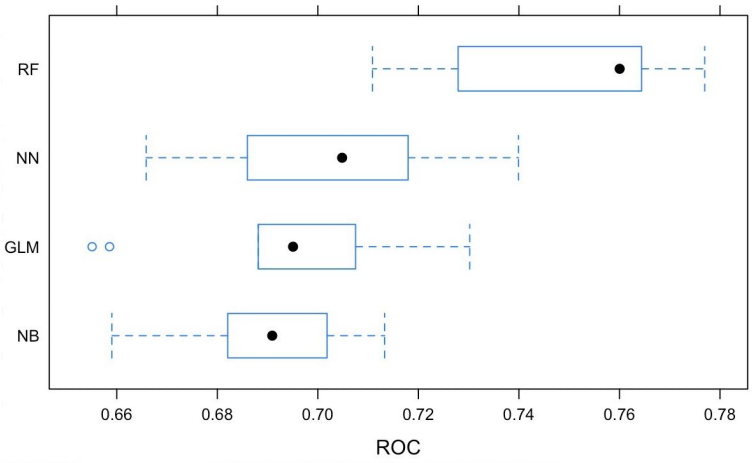
Since there were many redundant variables in the dataset, we performed Principal Component Analysis to 3 feature subsets:

Feature Subset	Attributes within subset
Speed of page loading	*frameTagRatio, compression_ratio*
Quality of content	*non_markup_alphanum_characters, spelling_errors_ratio, image_ratio, numberOfLinks*
External links	*Avglinksize, html_ratio, commonLinkRatio_1, commonLinkRatio_2, commonLinkRatio_3, commonLinkRatio_4, compression_ratio*

The rest of the attributes in the dataset that weren't covered above either had very poor correlation with the target *label*, ranked poorly in the feature ranking (image shown in Variable Importance bar graph) or were redundant (eg: *linkwordscore* is captured in both *non_markup_alphanum_char* and *avglinksize*).

For the External links feature subset, we were able to perform PCA that resulted in 2 components that captured 80% of the variance - we then named these components *linkquality1* and *linkquality2,* and added them to the dataset. We deleted the attributes that made up the External links subset. For the other two feature subsets, we were unable to capture enough variance with the minimum number of components and hence, left them as they were.

For the models Neural nets, Logistic regression and Naive Bayes algorithms, we used the attributes that we think would be the key drivers in determining whether the page is evergreen or not. We used attributes that have relatively higher correlation to the target *label* and that rank high in importance. We also dropped the all the attributes related to external links and used *linkquality1* and *linkquality2* instead.

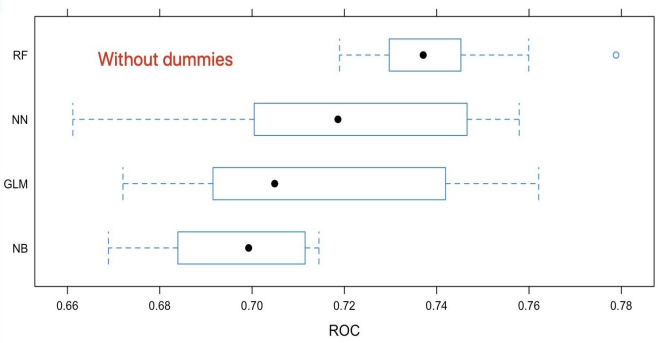


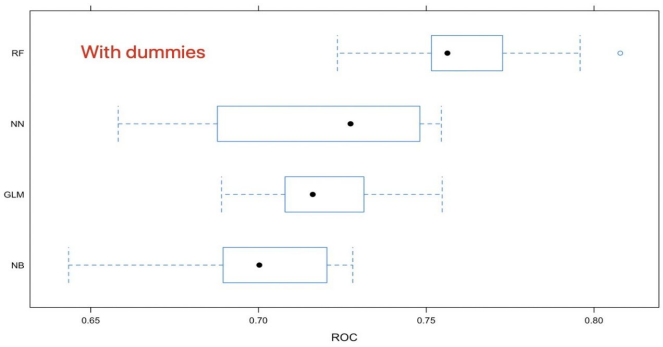
We noticed the following performance changes:

For Neural Nets, Logistic Regression and Naive Bayes models, there was a significant decrease in variability of AUC score, but the average score decreased.
For Random Forest model, there was a significant increase in the variability

Comparison of our models with and without using dummies:

We compared our model's performance with and without converting our categorical variables to dummy variables. Our model's performance improved with dummies (notice the change in scale and variance) and hence we decided to use it.





Resources Referred:

- Analysis of RF Model
- Text Mining with R
- Correlation Plot
- Predicting Web Page Longevity Through Relevancy Features

Breakdown of Web page Attributes:

FieldName	Type	Description
alchemy_category_score	Object	Alchemy category score (per the publicly available Alchemy API found at www.alchemyapi.com)
avglinksize	double	Average number of words in each link
commonLinkRatio_1	double	# of links sharing at least 1 word with 1 other links / # of links
commonLinkRatio_2	double	# of links sharing at least 1 word with 2 other links / # of links
commonLinkRatio_3	double	# of links sharing at least 1 word with 3 other links / # of links
commonLinkRatio_4	double	# of links sharing at least 1 word with 4 other links / # of links
compression_ratio	double	Compression achieved on this page via gzip (measure of redundancy)
embed_ratio	double	Count of number of <embed> usage
frameTagRatio	double	Ratio of iframe markups over total number of markups
html_ratio	double	Ratio of tags vs text in the page
image_ratio	double	Ratio of <img> tags vs text in the page
linkwordscore	double	Percentage of words on the page that are in hyperlink's text
numwords_in_url	double	Number of words in url
parametrizedLinkRatio	double	A link is parametrized if it's url contains parameters or has an attached onClick event
spelling_errors_ratio	double	Ratio of words not found in wiki (considered to be a spelling mistake)
urlid	integer	StumbleUpon's unique identifier for each url
non_markup_alphanum_characters	integer	Page's text's number of alphanumeric characters
numberOfLinks	integer	Number of <a> markups
frameBased	integer (0 or 1)	A page is frame-based (1) if it has no body markup but have a frameset markup
hasDomainLink	integer (0 or 1)	True (1) if it contains an <a> with an url with domain
is_news	object	True (1) if StumbleUpon's news classifier determines that this webpage is news
lengthyLinkDomain	integer (0 or 1)	True (1) if at least 3 <a> 's text contains more than 30 alphanumeric characters
news_front_page	object	True (1) if StumbleUpon's news classifier determines that this webpage is front-page news
label	***Object***	***User-determined label. Either evergreen (1) or non-evergreen (0); available for train.tsv only***
boilerplate	object	Boilerplate text
url	object	Url of the webpage to be classified
alchemy_category	object	Alchemy category (per the publicly available Alchemy API found at www.alchemyapi.com)