Data Mining Project: Online News Popularity

[Zhaofeng Shang, Xiang Li, Weixin Ji, Kushal Agrawal, Gokulramanan Soundararajan]

[Rutgers University]

Table of Contents

I. Introduction

It is all known that in the information era, online news has became one of the most crucial channels for people to get latest information around the world. Compared with print industry, online news platform tends to be more convenient, attainable and openness for both the media and individual sides. When browsing the website, readers can easily find out that millions of different news are being updated online with minutes. Because of the diversity of reader's choice, it is essential for news companies to have general idea about people's preference of which news they would like to read [1]. Therefore, the purpose of this paper is to make a prediction of the popularity of online news. We believe that our paper can be helpful for news companies to make strategies for attracting more viewers. By using the Online News Popularity dataset from UCI repository, we intend to make use of a largely and recently collected dataset in our paper.

This paper has the following structure. Section II introduces our dataset including data and sample we pick, the explanation of variables of our dataset as well as the feature selection we use. This section also includes the important way we to conduct in data pre-processing step. Section III gives our implementation of analyzing data by using several exploratory analytical methods respectively (including Classification, Association and Regression). In this section, we also provide supportive data visualization analysis of different methods. Finally, we analyze our results based on our research, generate them in our conclusion and propose our thinking of future work in Section IV. [2]

II. Methodology

1. Data and sample

We use the dataset from UCI Machine learning repository. In this platform, it indicates that detailed data includes date, href details, positive/negative polarity of its over all post, sentimental polarity, title polarity, number of tokens in title, number of keywords, and so on. The dataset were published by Mashable (www.mashable.com) and the acquisition date was on January 8, 2015. This table lists these attributes by category. [3]

Table 2: List of attributes by category.

Feature	Type (#)
Words	
Number of words in the title	number (1)
Number of words in the article	number (1)
Average word length	number (1)
Rate of non-stop words	ratio (1)
Rate of unique words	ratio (1)
Rate of unique non-stop words	ratio (1)
Links	
Number of links	number (1)
Number of Mashable article links	number (1)
Minimum, average and maximum number of shares of Mashable links	number (3)
Digital Media	
Number of images	number (1)
Number of videos	number (1)
Time	
Day of the week	nominal (1)
Published on a weekend?	bool (1)

Feature	Type (#)
Keywords	
Number of keywords	number (1)
Worst keyword (min./avg./max. shares)	number (3)
Average keyword (min./avg./max. shares)	number (3)
Best keyword (min./avg./max. shares)	number (3)
Article category (Mashable data channel)	nominal (1)
Natural Language Processing	
Closeness to top 5 LDA topics	ratio (5)
Title subjectivity	ratio (1)
Article text subjectivity score and its absolute difference to 0.5	ratio (2)
Title sentiment polarity	ratio (1)
Rate of positive and negative words	ratio (2)
Pos. words rate among non-neutral words	ratio (1)
Neg. words rate among non-neutral words	ratio (1)
Polarity of positive words (min./avg./max.)	ratio (3)
Polarity of negative words (min./avg./max.)	ratio (3)
Article text polarity score and its absolute difference to 0.5	ratio (2)

Target	Type (#)
Number of article Mashable shares	number (1)

2. Description of Variables

There are 61 (58 predictive attributes, 2 non-predictive, 1 goal field) numbers of attributes in our dataset. To generate these variables, we have date, href details, positive/negative polarity of its over all post, sentimental polarity, title polarity, number of tokens in title, number of keywords, the number of shares and so on.

3. Data Pre-processing

Data pre-processing is an important step in the data mining process. Real-world data is often incomplete, inconsistent, or lacking in certain behaviors or trends, and is likely to contain many errors. Analyzing data that has not been carefully screened for such errors can produce misleading results. Thus, data preprocessing, a data mining technique that involves transforming raw data into an understandable format, which is a proven method of resolving such issues. This section is dedicated to preprocess raw data for further analysis.

Firstly, there are just 6 attributes: (data_channel_is_lifestyle,data_channel_is_entertainment,data_channel_is_bus,data_channel_is_socmed,data_channel_is_tech,data_channel_is_world) about channel category in the original data set, its partition is too rough. Thus, we add a new attribute (data_channel_is_watercooler) for channels, and subdivide every channel category into more detailed topics, which assists in refining the data set and providing convenience for further analysis.

5	Channel	6	Channel	7	Channel
'Tech'	Tech	world	U.S.	Watercooler	Watercooler
	Mobile		World		Gadgets
	Music		Mobile		U.S.
	Gaming		Gaming		Mobile
	Small Business		Small Business		Media
	Paid Content		Movies		Movies
	Gadgets		Gadgets		Music
	Media		Media		Marketing
	Marketing		Dev & Design		Startups
	Movies		Apps & Software		Advertising
	Startups		Advertising		Gaming
	Advertising		Music		Apps & Software
	How To		Startups		Paid Content
			Paid Content		Small Business
					Dev & Design

1	Channel	2	Channel	3	Channel	4	Channel
Business'	Media	lifestyle	Gadgets	'Entertainment'	Media	Social Media	Social Media
	Business		Marketing		Entertainment		Movies
	Small Business		How To		Music		Marketing
	Paid Content		Apps & Software		Gaming		Media
	Advertising		Dev & Design		Marketing		Music
	Music		Gaming		Startups		How To
	Startups		Lifestyle		Mobile		Paid Content
	Marketing		U.S.		Small Business		Mobile
	Mobile		Mobile		Advertising		Startups
	Gaming		Startups		Movies		Gaming
			Music		Paid Content		
			Paid Content				
			Small Business				
			Conversations				
			Sports				
			Photography				
			Memes				
			Advertising				
			Movies				

Moreover, we preprocess data for the regression. For this purpose, we put data reduction to use, we delete 9 attributes (url, timedelta, week_is_monday, week_is_tuesday,week_is_wednesday, week_is_thursday, week_is_friday, week_is_saturday, week_is_sunday) in total because these attributes are less relevant to predict the online news popularity. After performing preprocessing, there are 396450 instances left.

n_tokens	n_tokens_	n_unique_	n_non_sto	n_non_sto	num_hrefs	num_self_	num_imgs	num_vide
12	219	0.6635945	1	0.8153846	4	2	1	0
9	255	0.6047431	1	0.7919463	3	1	1	0
9	211	0.5751295	1	0.6638655	3	1	1	0
9	531	0.5037879	1	0.6656347	9	0	1	0
13	1072	0.4156456	1	0.5408895	19	19	20	0
10	370	0.5598886	1	0.6981982	2	2	0	0
8	960	0.4181626	1	0.5498339	21	20	20	0
12	989	0.4335736	1	0.5721078	20	20	20	0
11	97	0.6701031	1	0.8367347	2	0	0	0
10	231	0.6363636	1	0.7971014	4	1	1	1

Data Preprocessing for Regression (Partial Table)

Besides that, we also preprocess data for association and classification. For association, we adopt data transformation to transfer the raw data into a readable format for machine. To be more specific, we transfer the data into 1 and 0 firstly, and then use ‘t’ and ‘?’ to represent ‘1’ and ‘0’ respectively. At the same time, we also adopt data reduction, and there are 8751 instances left in the end. In addition, we use data discretization to preprocess data for classification.

n_tokens	n_tokens_	n_unique_	n_non_sto	num_hrefs	num_self_	num_imgs
?	t	?	t	t	?	?
t	?	t	t	?	?	?
t	t	?	t	t	t	t
t	?	t	t	?	?	t
?	t	?	t	?	t	t
?	t	t	t	?	?	?
?	t	?	t	?	?	?
t	t	?	t	t	?	t

Data Preprocessing for Association(Partial Table)

Finally, for realizing the data visualization, we combine seven attributes, including week_is_monday, week_is_tuesday,week_is_wednesday, week_is_thursday, week_is_friday, week_is_saturday, week_is_sunday, into one attribute called weekday, and use channel number to stand for different article topics based on the attribute refinement (shown in the first method). Meanwhile, we find that this articles are all news, so we only can conclude the news popularity. In order to extend the range of this analysis, we can scrap author by data scraping to analyze the popularity of their other kinds of articles other than the news, and then push them to readers.

Date	Year	Month	Day	weekday	isWeekend	Type	Channel	topics	author
20130107	2013	1	7	1	0	3	Media	amazon, Entertai	Lauren Indvik
20130107	2013	1	7	1	0	1	Media	Business, Media,	Seth Fiegerman
20130107	2013	1	7	1	0	1	Business	Apple, apps, App	Seth Fiegerman
20130107	2013	1	7	1	0	3	Entertainr	Space, college f	Annie Bell
20130107	2013	1	7	1	0	5	Tech	apps, Apps and S	Emily Price
20130107	2013	1	7	1	0	5	Tech	CES, Gadgets, Mc	Lance Ulanoff
20130107	2013	1	7	1	0	2	Gadgets	bodymedia, CES,	Andrea Smith
20130107	2013	1	7	1	0	5	Tech	Canon, CES, Gadg	Pete Pachal
20130107	2013	1	7	1	0	5	Tech	Cars, car of the	Stan Schroeder
20130107	2013	1	7	1	0	6	U.S.	Politics, U.S.,	Alex Fitzpatrick
20130107	2013	1	7	1	0	6	World	Apocalypse, Aste	Annie Bell
20130107	2013	1	7	1	0	2	Gadgets	3D printers, 3D	Annie Bell
20130107	2013	1	7	1	0	7	Watercoole	Alt Image Lead,	Eric Larson
20130107	2013	1	7	1	0	7	Watercoole	jokes, Video, Vi	Neha Prakash
20130107	2013	1	7	1	0	7	Watercoole	Downton Abbey, g	Annie Colbert

Data Preprocessing for Visualization (Partial Table)

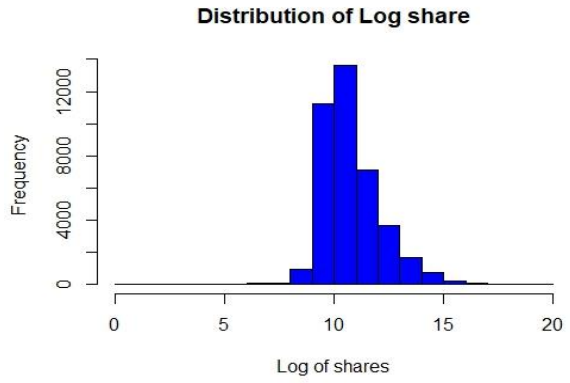
4. Several Exploratory analytical methods

a. Classification analysis

Classification is a data mining function that assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each record present in the data. For example, a classification model could be used to identify loan applicants as low, medium, or high credit risks.

Preprocessing:

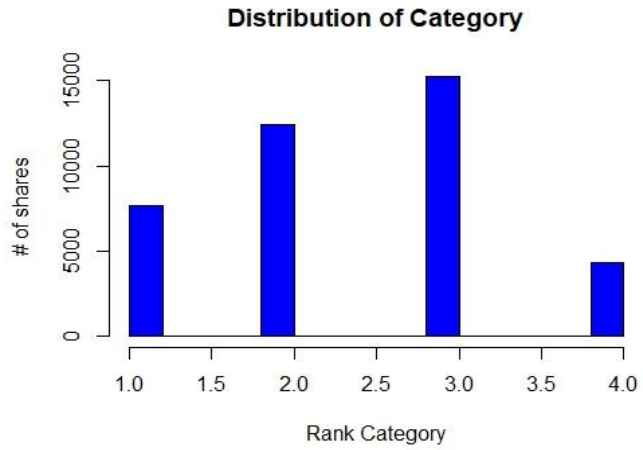
In this project, target label(shares) are continuous attribute and to make a classification model, this attribute had to be transformed. We have used feature engineering to shrink the huge spread of attribute - shares and categorize them into labels 1,2 3 & 4 where 1 refers least popular and 4 being most popular. This could be possible by converting the attribute shares it into $\log_2$ value and then categorize as 1 to 4.



In reference with the above histogram, ranks have been split using the formula:-

$(IF(log\_shares <= 9.75,1,IF(log\_shares <= 10.5,2,IF(log\_shares <= 12.5,3,4)))$

Once category label was set, other dependent attributes (shares, log_shares) have been deleted. Category attribute distribution is as below,



The next step is to analyze the data for missing values & noise and thereby perform data cleansing if required. The dataset didn't have any missing or null values and evidence of the same is given below.

```
In [11]: data.isnull().any() #there is no missing  values

Out[11]: url                             False
         timedelta                       False
         n_tokens_title                  False
         n_tokens_content                False
         n_unique_tokens                 False
         n_non_stop_words                False
         n_non_stop_unique_tokens        False
         num_hrefs                       False
         num_self_hrefs                  False
         num_imgs                        False
         num_videos                      False
         average_token_length            False
         num_keywords                    False
```

However, as the result of exploratory analysis, we found that 1 observation was noise and the same was removed to obtain clean data.

$$onlinedata <- onlinedata[!onlinedata\$n\_unique\_tokens == 701,]$$

Since, the target attribute Classify_2 is of the datatype INT, it was converted into factor from integer.

$$onlinedata\$Classify\_2 <- as.factor(onlinedata\$Classify\_2)$$

The last step in data processing involved performing scaling of attributes with respect to the mean of the attribute using below code. All the remainder attributes except the attribute Classify_2 underwent scaling.

```
for(i in ncol(onlinedata)-1)
{
  onlinedata[,i] <- scale(onlinedata[,i], center = TRUE, scale = TRUE)
}
```

Transforming:

Once, the data is cleaned, data transformation is essential to obtain better classification model as the dataset has enormous number of attributes (60). There is a high possibility for model overfitting in the absence of data transformation and there might be some attributes that do not add/contribute enough value to the target attribute. As a part of transforming the dataset or identifying the important attributes, Principal Component Analysis has been performed.

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation method to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components and gives the importance features by the variance range. On performing PCA, 30 key features were obtained which were retained and other features which were less significant/important were removed. PCA output is shown below,

```
In [58]: ax = drawVectors(T, pca.components_, df.columns.values, plt, scaleFeatures)
         T  = pd.DataFrame(T)

         T.columns = ['component1', 'component2']
         T.plot.scatter(x='component1', y='component2', marker='o', c=labels, alpha=0.75, ax=ax)

         plt.show()
```

```
Features by importance:
 [(7.518670023797414, 'min_negative_polarity'), (7.326961968202904, 'rate_negative_words'), (7.137232718836044, 'global_rate_ne
gative_words'), (7.08660978244684, 'avg_negative_polarity'), (6.625056295119557, 'kw_avg_avg'), (6.1899345198381495, 'kw_avg_ma
x'), (5.957968559440267, 'global_sentiment_polarity'), (5.69264125253344, 'rate_positive_words'), (5.543691960046346, 'kw_max_m
ax'), (5.3660209931570275, 'LDA_03'), (5.267814340146103, 'timedelta'), (5.161567507057758, 'kw_min_min'), (5.139223063257429,
'LDA_04'), (4.608030588614317, 'global_subjectivity'), (4.555312369240785, 'kw_max_avg'), (4.539278541827095, 'data_channel_is_
tech'), (4.151839896231356, 'avg_positive_polarity'), (4.068835555185356, 'kw_min_avg'), (4.054503925207741, 'max_positive_pola
rity'), (3.6987631572372446, 'num_hrefs'), (3.592753234055853, 'global_rate_positive_words'), (3.5704034709819097, 'num_video
s'), (3.481018732278375, 'data_channel_is_entertainment'), (3.301402183938018, 'average_token_length'), (3.18175507071548, 'num
_imgs'), (2.9500956601953336, 'self_reference_max_shares'), (2.888217135821508, 'self_reference_avg_sharess'), (2.8857421888089
005, 'n_tokens_content'), (2.47974530856121, 'title_subjectivity'), (2.4358797678081476, 'kw_min_max'), (2.173210625240051, 'ab
s_title_sentiment_polarity'), (2.1141161644626734, 'LDA_00'), (2.07730789345023, 'n_tokens_title'), (2.071759808164401, 'LDA_0
1'), (2.0635533138490265, 'self_reference_min_shares'), (2.0359719997876335, 'kw_max_min'), (1.8810920719299677, 'data_channel_
is_bus'), (1.7134312111629002, 'max_negative_polarity'), (1.597537570255435, 'data_channel_is_world'), (1.5804274887917575, 'nu
m_keywords'), (1.5153744015672277, 'num_self_hrefs'), (1.4993934358595413, 'LDA_02'), (1.4055078269787262, 'title_sentiment_pol
arity'), (1.2971152096311918, 'min_positive_polarity'), (1.2722503316865867, 'kw_avg_min'), (1.135165696380437, 'data_channel_i
s_lifestyle'), (0.9866143232377169, 'is_weekend'), (0.9116552804623288, 'abs_title_subjectivity'), (0.7666629744089135, 'weekda
```

The attributes with value below 2 obtained from PCA have been rejected and the ones above 2 have been taken into consideration for further analysis. The command using which the attributes were removed can be found below.

onlinedata <- subset(onlinedata, select = -c(url,timedelta,n_tokens_content, self_reference_max_shares, self_reference_avg_sharess, title_subjectivity, n_tokens_title, num_keywords,num_self_hrefs, title_sentiment_polarity, min_positive_polarity, max_negative_polarity, n_non_stop_words, n_non_stop_unique_tokens, n_unique_tokens, LDA_01, LDA_02, LDA_03, LDA_04, LDA_00, shares,is_weekend, weekday_is_sunday, weekday_is_saturday, weekday_is_monday, weekday_is_tuesday, weekday_is_wednesday, weekday_is_thursday, weekday_is_friday, log_share))

Finally, the new dataset, which is ready to build a classification model has the dimension 39643 observations and 33 attributes.

```
> dim(onlinedata)
[1] 39643    33
```

Reason for using C5.0 classification mode:

C4.5 has a better handling for both discrete and continuous attributes which are present in our current dataset. The continuous attributes are handled in C4.5 by creating a threshold and then splitting the list based upon whether the value is above the threshold or is less than or equal to it.

C4.5 algorithm prunes the tree after creation. C4.5 revisits the tree after it is created and attempts to remove branches that do not help by replacing them with leaf nodes.

C5.0 gives additional benefits over C4.5 which are given as below:

- C5.0 is significantly faster than C4.5 (several orders of magnitude)
- C5.0 is more memory efficient than C4.5
- C5.0 fetches equivalent results as C4.5 with considerably smaller decision trees.
- Boosting improves the trees and gives them more accuracy.
- C5.0 allows us to weight different cases and misclassification types.
- A C5.0 option automatically winnows the attributes to remove those that may be unhelpful.

Using Random Forest algorithm may lead to the problem of overfitting as there are large number of attributes present in our data set.

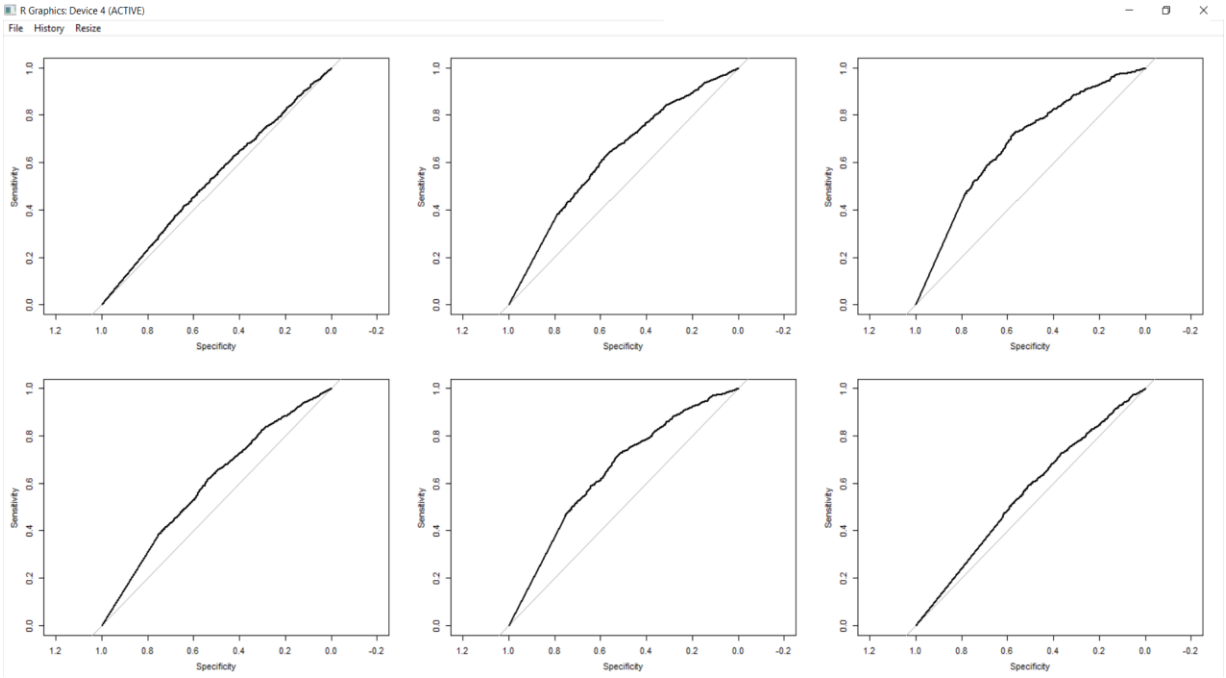
Owing to the above characteristics of C5.0, we believe that using C5.0 for our data set will yield us with most appropriate classification model.

Results of C5.0:

Below command, gives the overall statistics of the model with confusion matrix and performance analyzer. As per the result, based accuracy for each mode is listed below.

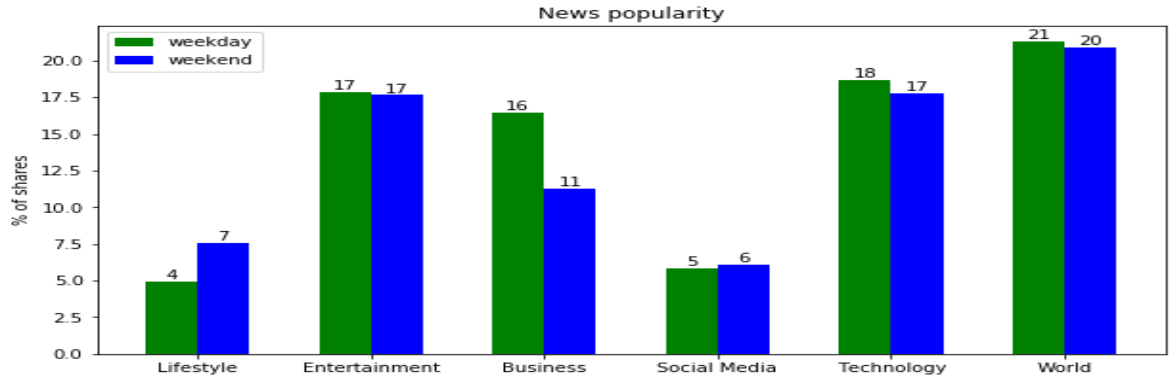
```
> caret::confusionMatrix(onlineC50.pred, onlinedata[samplesets == 2,]$Classify_2)

Confusion Matrix and Statistics

          Reference
Prediction    1    2    3    4
         1  369  377  262   73
         2  615  874  680  151
         3  503 1172 1943  563
         4   29   56  100   64

Overall Statistics

               Accuracy : 0.415
                 95% CI : (0.4041, 0.426)
    No Information Rate : 0.3812
    P-Value [Acc > NIR] : 4.591e-10

                  Kappa : 0.1302
 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4
Sensitivity          0.24340   0.3526   0.6509 0.075206
Specificity          0.88725   0.7298   0.5382 0.973496
Pos Pred Value       0.34135   0.3767   0.4647 0.257028
Neg Pred Value       0.83007   0.7088   0.7145 0.896202
Prevalence           0.19359   0.3166   0.3812 0.108671
Detection Rate       0.04712   0.1116   0.2481 0.008173
Detection Prevalence 0.13804   0.2963   0.5339 0.031797
Balanced Accuracy    0.56533   0.5412   0.5945 0.524351
```

And the performance of the model could be found in the below ROC curve. Since the ROC curve is above the diagonal line the performance of the model is better than random classification.



Research:

We read many types of news everyday based on our interest and news popularity. However, we seldom research on the factors which makes those news popular and causes people to share them more. We may guess that news about entertainment would be more popular and shared during weekends than on weekdays. Likewise, we may guess that news related to business would be shared more on weekdays than on weekend's. In our curiosity to understand these relations between news popularity and data channel (Lifestyle, Business, Social Media, Technology etc.,) we explored our data further and found the following results.



As evident in the above graph, some of our guesses were true while some turned around the opposite way.

Findings:

1. Channels like Entertainment, Social Media, Technology and world have been shared and are equally popular both during weekends and weekdays. So, news publication companies should focus on these channels both during weekends and weekdays equally.
2. However, Lifestyle news are more popular during weekends than during weekdays. Hence, companies can give more importance to Lifestyle news on weekends to bring in more customer's viewership/subscriptions.
3. Likewise, another obvious finding is that Business news are more shared during weekday's than during weekend's.

b. Association analysis

There are 61 different attributes in the online news popularity dataset, they may or may not related to the final shares, or those attributes may have some inter-relationship, so we need use the WEKA to do the association analysis by using the APRIORI method. In this way, we can figure out which attributes are associated with each other. This is meaningful work, because when we know which attributes always come up together, we can group them and minimize our attention to those target groups.

```
Apriori
=======

Minimum support: 0.5 (4375 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 10

Generated sets of large itemsets:

Size of set of large itemsets L(1): 6

Size of set of large itemsets L(2): 9

Size of set of large itemsets L(3): 4

Best rules found:

 1. n_tokens_content=t 6142 ==> n_non_stop_words=t 6142    <conf:(1)> lift:(1) lev:(0) [22] conv:(22.46)
 2. n_tokens_content=t global_subjectivity=t 6132 ==> n_non_stop_words=t 6132    <conf:(1)> lift:(1) lev:(0) [22] conv:(22.43)
 3. global_sentiment_polarity=t 4452 ==> global_subjectivity=t 4448    <conf:(1)> lift:(1) lev:(0) [15] conv:(3.97)
 4. n_non_stop_words=t global_sentiment_polarity=t 4441 ==> global_subjectivity=t 4437    <conf:(1)> lift:(1) lev:(0) [15] conv:(3.96)
 5. n_tokens_content=t 6142 ==> global_subjectivity=t 6132    <conf:(1)> lift:(1) lev:(0) [17] conv:(2.49)
 6. n_tokens_content=t n_non_stop_words=t 6142 ==> global_subjectivity=t 6132    <conf:(1)> lift:(1) lev:(0) [17] conv:(2.49)
 7. n_tokens_content=t 6142 ==> n_non_stop_words=t global_subjectivity=t 6132    <conf:(1)> lift:(1.01) lev:(0) [39] conv:(4.47)
 8. rate_positive_words=t 4658 ==> n_non_stop_words=t 4647    <conf:(1)> lift:(1) lev:(0) [6] conv:(1.42)
 9. global_subjectivity=t rate_positive_words=t 4642 ==> n_non_stop_words=t 4631    <conf:(1)> lift:(1) lev:(0) [5] conv:(1.41)
10. global_sentiment_polarity=t 4452 ==> n_non_stop_words=t 4441    <conf:(1)> lift:(1) lev:(0) [5] conv:(1.36)
11. global_subjectivity=t global_sentiment_polarity=t 4448 ==> n_non_stop_words=t 4437    <conf:(1)> lift:(1) lev:(0) [5] conv:(1.36)
12. n_non_stop_words=t num_keywords=t 4392 ==> global_subjectivity=t 4378    <conf:(1)> lift:(1) lev:(0) [5] conv:(1.31)
13. global_sentiment_polarity=t 4452 ==> n_non_stop_words=t global_subjectivity=t 4437    <conf:(1)> lift:(1) lev:(0) [20] conv:(2.23)
14. num_keywords=t 4411 ==> global_subjectivity=t 4396    <conf:(1)> lift:(1) lev:(0) [4] conv:(1.23)
15. rate_positive_words=t 4658 ==> global_subjectivity=t 4642    <conf:(1)> lift:(1) lev:(0) [4] conv:(1.22)
16. n_non_stop_words=t rate_positive_words=t 4647 ==> global_subjectivity=t 4631    <conf:(1)> lift:(1) lev:(0) [4] conv:(1.22)
17. global_subjectivity=t 8711 ==> n_non_stop_words=t 8680    <conf:(1)> lift:(1) lev:(0) [0] conv:(1)
18. num_keywords=t global_subjectivity=t 4396 ==> n_non_stop_words=t 4378    <conf:(1)> lift:(1) lev:(-0) [-1] conv:(0.85)
19. num_keywords=t 4411 ==> n_non_stop_words=t 4392    <conf:(1)> lift:(1) lev:(-0) [-2] conv:(0.81)
20. n_non_stop_words=t 8718 ==> global_subjectivity=t 8680    <conf:(1)> lift:(1) lev:(0) [0] conv:(1)
```

n_tokens	n_tokens	n_unique	n_non_st	num_hre	num_self	num_img	num_vid	average_	num_key	data_cha	data_cha	data_cha	data_cha	data_cha	data_cha	kw_avg_a	self_refe	weekday	weekday
?	t	?	t	t	?	?	?	?	?	?	t	?	?	?	?	?	?	?	?
t	?	t	t	?	?	?	?	t	?	?	?	?	?	?	t	?	?	?	?
t	t	?	t	t	t	t	?	?	t	?	?	t	?	?	?	?	?	?	?
t	?	t	t	?	?	t	?	?	?	?	?	?	?	?	?	?	?	?	?
?	t	?	t	?	t	t	?	?	?	?	?	?	?	t	?	t	t	?	?
?	t	t	t	?	?	?	t	?	t	?	?	t	?	?	?	t	?	?	t
?	t	?	t	?	?	?	?	?	t	?	?	?	?	t	?	?	?	?	?
t	t	?	t	t	?	t	?	?	?	t	?	?	?	?	?	t	?	?	?
?	?	?	t	t	?	?	?	t	t	?	?	?	?	?	?	?	?	?	?
?	t	?	t	?	?	?	?	?	?	?	t	?	?	?	?	?	?	?	?
t	t	?	t	?	?	?	?	t	t	?	?	?	?	?	t	?	?	?	?
t	?	t	t	?	?	t	?	?	?	?	?	?	?	?	?	?	t	?	?
?	t	?	t	?	?	?	t	?	?	?	?	?	?	?	t	?	?	?	?

To finish the association analysis, the dataset should be changed to fit the requirement of APRIORI, According to each column' value, we separate value into two part, and use the '?' to represent the lower value, the 't' to represent the upper value.

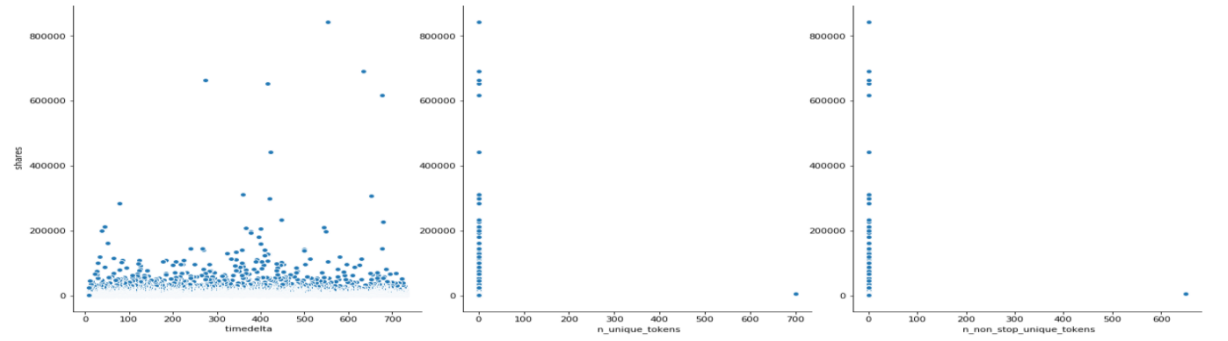
According to the association rules we found in the picture, we got top20 rules by using APRIORI method, we know that 'n_tokens_content' which means Number of words in the content usually come up with 'n_non_stop_words', which means Rate of non-stop words in the content. When a news data has 'n_tokens_content' attribute and 'global_subjectivity', which means Text subjectivity, it has high probability to show up with 'n_non_stop_words', which means Rate of non-stop words in the content.

c. Regression

1. Linear Regression

In the upper part, we analyzed the relationship among the attributes by using the association method, we figured out which attributes have deeper internal relations. However, we don't know exactly how them work and what the specific value it is. Here, we need take the regression analysis to find out the answer.

```
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt

%matplotlib inline
data=pd.read_csv('regression1.csv')
sns.pairplot(data, x_vars=[' timedelta',' n_unique_tokens',' n_non_stop_unique_tokens',], y_vars=' shares', size=7, aspect=0.8)
plt.show()
```



As showed above, there is a relationship between the features and the response using scatterplots. Those are some attributes which have bad linear relationship with the main attribute-shares. The first one is the graph between time-delta and shares. Attribute-‘time-delta’ is Days between the article publication and the dataset acquisition (non-predictive), so that this attribute is meaningless to the attribute-‘shares’. The other two is ‘shares‘ with ‘n_unique_tokens’, which is Rate of unique words in the content, and ‘shares’ with n_non_stop_unique_tokens, which is Rate of unique non-stop words in the content. Because we need take regression analysis to fit every attributes with attribute-‘shares’ into one equation. So we need get rid of some attributes which will do bad influence with the model, in order to reduce the model error.

After finished the data preprocessing, we use the Linear regression model in weka to finish the function.

```
shares =

    55.8073 *  n_tokens_title +
  -593.7402 *  n_non_stop_words +
   960.0803 *  n_non_stop_unique_tokens +
    31.7819 *  num_hrefs +
   -47.5103 *  num_self_hrefs +
    17.8001 *  num_imgs +
  -548.3959 *  average_token_length +
    79.8234 *  num_keywords +
    -0.4242 *  kw_min_avg +
    -0.1929 *  kw_max_avg +
     1.6963 *  kw_avg_avg +
     0.0264 *  self_reference_min_shares +
     0.0052 *  self_reference_max_shares +
    -0.0052 *  self_reference_avg_sharess +
   307.7646 *  is_weekend +
  2732.4499 *  global_subjectivity +
 -7578.1641 *  global_rate_positive_words +
  1203.5637 *  rate_positive_words +
   716.3292 *  rate_negative_words +
 -1769.2139 *  min_positive_polarity +
 -1307.9448 *  avg_negative_polarity +
   669.5334 *  abs_title_subjectivity +
   658.3899 *  abs_title_sentiment_polarity +
 -1968.166
```

According to the model function we made above, we can know the exact values among attributes. In this way, it is beneficial for us to make the better decision.

Here are some analysis:

- Videos doesn’t matter. In the function, there are no attribute called ‘num_videos’, which is put into the dataset. It tells us that the number of video does not affect the number of shares, so that it does not show up in the equation. It indicates that the company can get rid of the video to cut the cost. Meanwhile, there are still lots of attribute did not show up in the equation. Here is one example.
- The words in title do matter. In the model function, the coefficient value before the attribute ‘n_token_title’ is 55.8073, which means that it adds 55 new shares per ‘n_token_title’ increase one. It indicates that increasing the number of words in the title will create new shares.
- ‘average_token_length’ do harm to the shares. This attribute represents the average length of the words in the content. According to the equation above, the coefficient before this attribute is -548, which means that there are 548 shares lost per average length of the words in the content be

added. It indicates that company need reduce the length of words in the news as far as possible. From this analysis result, readers may not like too many long words showed in a single news. Because it may make the news harder to understand.

In this part, we use scikit-learn library in python to do the validation, in order to verify the accuracy of the model. Here shows two ways:

Holdout method

Firstly, using holdout method. This is the simplest kind of cross validation. The data set is separated into two sets, called the training set and the testing set. We put all the attributes except 'shares' into the X list, the attribute 'shares' is put into the Y list. The function is fitted by using the training set only. Then the function approximator is asked to predict the output values for the data in the testing set. The advantage of this method is that it is usually takes no longer to compute. However, its evaluation can have a high variance. The evaluation may depend heavily on which data points end up in the training set and which end up in the test set, and thus the evaluation may be significantly different depending on how the division is made.

We need divide the datasets into two parts, one train set, one test set.

```
import matplotlib.pyplot as plt
%matplotlib inline
import numpy as np
import pandas as pd
from sklearn import datasets, linear_model
```

```
from sklearn.cross_validation import train_test_split
X_train, X_test, Y_train,Y_test=train_test_split(X,Y,random_state=1)
print (X_train.shape)
print (Y_train.shape)
print (X_test.shape)
print (Y_test.shape)
```

```
(29733, 29)
(29733, 1)
(9911, 29)
(9911, 1)
```

After separating, use the X_train and Y_train data to finish the regression model, let the model do the prediction based on the X_test data. After that, use Y_predict and Y_test data to calculate the MSE and RMSE based on the metrics from the sklearn package.

```
from sklearn.linear_model import LinearRegression
lin = LinearRegression()
lin.fit(X_train, Y_train)
Y_predict=lin.predict(X_test)
from sklearn import metrics
#calculate the MSE(mean squared error)
MSE=metrics.mean_squared_error(Y_predict,Y_test)
print("MSE : ",MSE)
#calculate the RMSE(root mean squared error)
RMSE=np.sqrt(metrics.mean_squared_error(Y_test,Y_predict))
print("RMSE : ",RMSE)
```

```
MSE :  104439885.957
RMSE :  10219.5834532
```

K-fold cross validation

Another way is K-fold cross validation, which is one way to improve over the holdout method. Here, we let K become 10, then the data set is divided into *10* subsets, and the holdout method is repeated 10 times. Each time, one of the *10* subsets is used as the test set and the other *9* subsets are put together to form a training set. Then the average error across all *10* trials is computed. The advantage of this method is that it matters less how the data gets divided. Every data point gets to be in a test set exactly once, and gets to be in a training set *9* times. The variance of the resulting estimate is reduced as *K* is increased. The disadvantage of this method is that the training algorithm has to be repeated into *k* times, which means it takes *k* times as much computation to make an evaluation.

```
X=data[[' n_tokens_title',' n_tokens_content',' num_hrefs',' num_self_hrefs',' num_imgs',' average_token_length',' num_keywords',
Y=data[[' shares']]
from sklearn.model_selection import cross_val_predict
predicted = cross_val_predict(lin,X,Y,cv=10)
#caculate the MSE(mean squared error)
MSE=metrics.mean_squared_error(Y,predicted)
print("MSE : ",MSE)
#caculate the RMSE(root mean squared error)
RMSE=np.sqrt(metrics.mean_squared_error(Y,predicted))
print("RMSE : ",RMSE)
```

```
MSE :  134802907.494
RMSE :  11610.4654297
```

2. Principal Components, Ridge and LASSO Regression

Our goal is to find the model that can predict the popularity of fresh news using its features. First, to have roughly view of the data, we perform simple linear regression on all the variable. The result is shown below.

```
##
## Call:
## lm(formula = shares ~ ., data = Dat)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -27963  -2277  -1204    -71 837227
##
## Coefficients: (2 not defined because of singularities)
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    1.954e+05  6.129e+06   0.032 0.974570
## timedelta                      1.678e+00  3.924e-01   4.275 1.91e-05 ***
## n_tokens_title                 1.125e+02  2.915e+01   3.858 0.000114 ***
## n_tokens_content               5.896e-01  2.235e-01   2.638 0.008332 **
## n_unique_tokens                3.354e+03  1.924e+03   1.744 0.081228 .
## n_non_stop_words              -1.583e+03  5.910e+03  -0.268 0.788858
## n_non_stop_unique_tokens      -1.375e+03  1.630e+03  -0.844 0.398852
## num_hrefs                      2.619e+01  6.705e+00   3.906 9.38e-05 ***
## num_self_hrefs                -6.159e+01  1.784e+01  -3.453 0.000556 ***
## num_imgs                       1.148e+01  8.941e+00   1.284 0.199124
## num_videos                     4.083e+00  1.575e+01   0.259 0.795447
## average_token_length          -5.439e+02  2.430e+02  -2.238 0.025219 *
## num_keywords                   5.507e+01  3.715e+01   1.482 0.138246
## data_channel_is_lifestyle     -9.580e+02  3.952e+02  -2.424 0.015336 *
## data_channel_is_entertainment -1.076e+03  2.563e+02  -4.198 2.70e-05 ***
## data_channel_is_bus           -7.752e+02  3.827e+02  -2.026 0.042790 *
## data_channel_is_socmed        -5.249e+02  3.727e+02  -1.408 0.159039
## data_channel_is_tech          -4.774e+02  3.717e+02  -1.284 0.199026
## data_channel_is_world         -3.136e+02  3.784e+02  -0.829 0.407209
## kw_min_min                     1.592e+00  1.629e+00   0.977 0.328425
## kw_max_min                     1.079e-01  5.035e-02   2.144 0.032034 *
## kw_avg_min                    -4.935e-01  3.097e-01  -1.594 0.110971
## kw_min_max                    -2.487e-03  1.177e-03  -2.112 0.034674 *
## kw_max_max                    -2.459e-05  5.898e-04  -0.042 0.966748
## kw_avg_max                     4.521e-05  8.481e-04   0.053 0.957490
## kw_min_avg                    -3.641e-01  7.565e-02  -4.813 1.49e-06 ***
## kw_max_avg                    -2.061e-01  2.530e-02  -8.143 3.95e-16 ***
## kw_avg_avg                     1.685e+00  1.439e-01  11.707  < 2e-16 ***
## self_reference_min_shares      2.659e-02  7.524e-03   3.534 0.000409 ***
## self_reference_max_shares      5.879e-03  4.083e-03   1.440 0.149849
## self_reference_avg_sharess    -6.423e-03  1.044e-02  -0.615 0.538339
## weekday_is_monday              2.655e+02  2.631e+02   1.009 0.312888
## weekday_is_tuesday            -2.805e+02  2.592e+02  -1.082 0.279135
## weekday_is_wednesday          -1.198e+02  2.591e+02  -0.462 0.643813
## weekday_is_thursday           -2.918e+02  2.597e+02  -1.124 0.261055
## weekday_is_friday             -2.520e+02  2.689e+02  -0.937 0.348792
## weekday_is_saturday            3.804e+02  3.205e+02   1.187 0.235301
## weekday_is_sunday                     NA         NA      NA       NA
## is_weekend                            NA         NA      NA       NA
## LDA_00                        -1.969e+05  6.129e+06  -0.032 0.974376
## LDA_01                        -1.977e+05  6.129e+06  -0.032 0.974272
## LDA_02                        -1.981e+05  6.129e+06  -0.032 0.974213
## LDA_03                        -1.973e+05  6.129e+06  -0.032 0.974321
## LDA_04                        -1.973e+05  6.129e+06  -0.032 0.974322
## global_subjectivity            2.497e+03  8.504e+02   2.936 0.003322 **
## global_sentiment_polarity      8.146e+02  1.668e+03   0.489 0.625174
## global_rate_positive_words    -1.392e+04  7.165e+03  -1.943 0.052036 .
## global_rate_negative_words     1.041e+02  1.368e+04   0.008 0.993927
## rate_positive_words            2.024e+03  5.775e+03   0.350 0.726025
## rate_negative_words            2.114e+03  5.821e+03   0.363 0.716520
## avg_positive_polarity         -1.685e+03  1.366e+03  -1.233 0.217460
## min_positive_polarity         -1.898e+03  1.144e+03  -1.659 0.097057 .
## max_positive_polarity          3.113e+02  4.311e+02   0.722 0.470213
## avg_negative_polarity         -1.707e+03  1.258e+03  -1.356 0.175038
## min_negative_polarity          8.207e+01  4.590e+02   0.179 0.858083
## max_negative_polarity         -1.787e+02  1.046e+03  -0.171 0.864428
## title_subjectivity            -9.160e+01  2.741e+02  -0.334 0.738249
## title_sentiment_polarity       2.041e+02  2.504e+02   0.815 0.414962
## abs_title_subjectivity         6.557e+02  3.640e+02   1.801 0.071634 .
## abs_title_sentiment_polarity   6.199e+02  3.957e+02   1.567 0.117183
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11500 on 39586 degrees of freedom
## Multiple R-squared:  0.02355,    Adjusted R-squared:  0.02214
## F-statistic: 16.75 on 57 and 39586 DF,  p-value: < 2.2e-16
```

From the result, we can see that both R^2 and Adjusted R^2, two important parameters measuring for goodness-of-fit, are around 0.02, which indicate that the full models can explain extremely small part of data set.

Moreover, we notice that the coefficients of weekday_is_Sunday and weekday_is_Saturday is NA. These two variables are dummy variables which used 1 to indicate that weekday is Sunday or Saturday. There are 2737 ones in variable weekday_is_Sunday and 2453 ones in variable weekday_is Saturday. Comparing to 39644 observations, the information provided by these two variables is not enough, which can cause the singularity of matrix xTx. The singularity of matrix xTx can cause the missing of some coefficients. To solve this question, we tried principal components regression, ridge regression and lasso regression.

When performing these techniques, we first divide the entire data set into train set and test set. The train set consists of 75% of the whole data and the rest forms the test set.

PCA regression

In order to predict online news popularity, we need to build a model to achieve this goal. At first, we use all of variables to perform the simple linear regression. Based on this result, we can see that the values of R^2 and adjusted R^2 are about 0.02, which means that this model is a bad model because it explains less about data set. Besides that, it shows that the coefficient of weekday_is_Sunday and weekday_is_Saturdy are all NA, which is due to that there are few 1 in these two variables. There are 2737 ones in variable weekday_is_Sunday and 2453 ones in variable weekday_is Saturday. Comparing to 39644 observations, the information provided by these two variables is not enough, which can cause the singularity of matrix xTx. The singularity of matrix xTx can cause the missing of some coefficients. Thus, we try to use principal components analysis (PCA), which is viewed as special case of multivariate reduced-rank regression, to solve this problem.

Principal components analysis, which have a role in discovering important features of the data by reducing dimensionality and decreasing computational cost during variable selection. When using PCA, we assume that these variables are independent, which is an important assumption.

First, seven principal components can explain most variance. The results is as follows. Thus, we use seven four components to perform regression.

```
## Importance of components:
##                             PC1       PC2       PC3       PC4       PC5
## Standard deviation     2.316e+05 1.073e+05 4.985e+04 4.748e+04 1.862e+04
## Proportion of Variance 7.628e-01 1.638e-01 3.533e-02 3.205e-02 4.930e-03
## Cumulative Proportion  7.628e-01 9.266e-01 9.620e-01 9.940e-01 9.990e-01
##                             PC6       PC7       PC8       PC9  PC10  PC11
## Standard deviation     6.582e+03 4.494e+03 2.842e+03 1.088e+03 496.1 453.2
## Proportion of Variance 6.200e-04 2.900e-04 1.100e-04 2.000e-05   0.0   0.0
## Cumulative Proportion  9.996e-01 9.999e-01 1.000e+00 1.000e+00   1.0   1.0
##                          PC12  PC13 PC14  PC15  PC16  PC17  PC18  PC19
## Standard deviation     192.5 156.2 35.5 10.21 7.331 7.067 3.912 3.342
## Proportion of Variance   0.0   0.0  0.0  0.00 0.000 0.000 0.000 0.000
## Cumulative Proportion    1.0   1.0  1.0  1.00 1.000 1.000 1.000 1.000
##                          PC20  PC21  PC22  PC23   PC24   PC25   PC26   PC27
## Standard deviation     2.039 1.625 0.813 0.502 0.4702 0.4385 0.4323 0.4302
## Proportion of Variance 0.000 0.000 0.000 0.000 0.0000 0.0000 0.0000 0.0000
## Cumulative Proportion  1.000 1.000 1.000 1.000 1.0000 1.0000 1.0000 1.0000
##                            PC28   PC29   PC30   PC31   PC32   PC33   PC34
## Standard deviation     0.4278 0.4139 0.3914 0.3856 0.2903 0.2689 0.2555
## Proportion of Variance 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
## Cumulative Proportion  1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
##                            PC35   PC36   PC37  PC38   PC39   PC40   PC41
## Standard deviation     0.2449 0.2373 0.2335 0.214 0.1641 0.1607 0.1559
## Proportion of Variance 0.0000 0.0000 0.0000 0.000 0.0000 0.0000 0.0000
## Cumulative Proportion  1.0000 1.0000 1.0000 1.000 1.0000 1.0000 1.0000
##                          PC42  PC43   PC44   PC45    PC46    PC47    PC48
## Standard deviation     0.151 0.138 0.1277 0.1026 0.08898 0.07567 0.06863
## Proportion of Variance 0.000 0.000 0.0000 0.0000 0.00000 0.00000 0.00000
## Cumulative Proportion  1.000 1.000 1.0000 1.0000 1.00000 1.00000 1.00000
##                            PC49    PC50    PC51    PC52    PC53    PC54
## Standard deviation     0.06665 0.04806 0.03905 0.03559 0.02914 0.02167
## Proportion of Variance 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
```

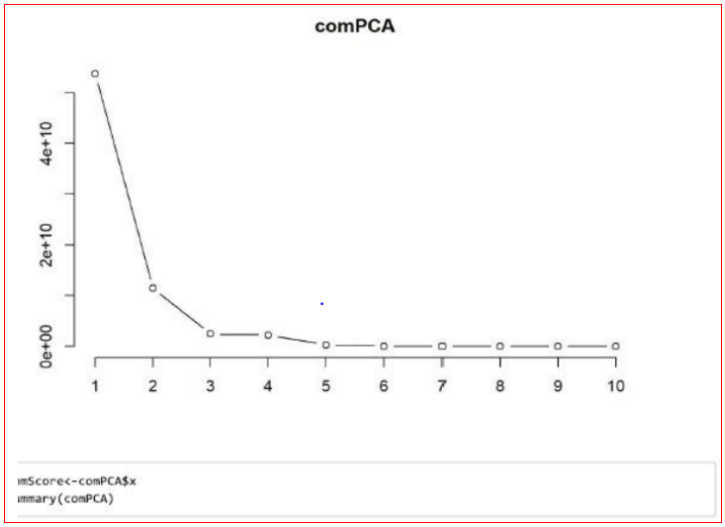
Second, we use the train set to validate the model, while using test set to compare the performances of different models. Using the PCA regression, the mean square error of the model on the training set is 123200737. The result is as follow. At this point, the result which is not good since R^2 is only 0.007, is not better than simple linear regression.

```
##
## Call:
## lm(formula = V8 ~ ., data = train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -40724  -2284  -1654   -480 838749
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.358e+03  6.805e+01  49.350  < 2e-16 ***
## PC1          7.744e-04  2.929e-04   2.644  0.00819 **
## PC2         -4.804e-03  6.332e-04  -7.588 3.34e-14 ***
## PC3         -8.746e-03  1.369e-03  -6.390 1.68e-10 ***
## PC4          7.969e-03  1.448e-03   5.503 3.77e-08 ***
## PC5         -6.456e-03  3.710e-03  -1.740  0.08183 .
## PC6         -8.859e-02  9.836e-03  -9.007  < 2e-16 ***
## PC7          1.352e-02  1.517e-02   0.891  0.37288
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11730 on 29725 degrees of freedom
## Multiple R-squared:  0.00731,    Adjusted R-squared:  0.007076
## F-statistic: 31.27 on 7 and 29725 DF,  p-value: < 2.2e-16
```

```
pca_pre<-predict(pca_fit,test)
MSE<-mean((pca_pre-test[,8])^2)
MSE # Used to compare different models
```

```
## [1] 123200737
```

This bad results is due to that the information provided by the two variables are not enough compared with the 39644 observations, and collinearity existing among variables (shown as follows). Hens, we suppose that if we can use principal components to run lasso and ridge.



Ridge regression

When using ridge regression, we use 10-fold cross validation on train set to determine the best lamda. After picking up the optimal lamda, the mean square error of the model on test data set is 121764701, which is better than that from principal components regression. The result is as followed:

```
## 60 x 1 sparse Matrix of class "dgCMatrix"
##                                          1
## (Intercept)                   -1.793861e+03
## timedelta                      1.802823e+00
## n_tokens_title                 1.110597e+02
## n_tokens_content               4.317714e-01
## n_unique_tokens                6.584633e+00
## n_non_stop_words              -3.515134e+00
## n_non_stop_unique_tokens       5.720222e+00
## num_hrefs                      2.660563e+01
## num_self_hrefs                -6.335614e+01
## num_imgs                       7.238677e+00
## num_videos                     1.157819e+01
## average_token_length          -1.589680e+02
## num_keywords                   8.539575e+01
## data_channel_is_lifestyle     -1.090749e+03
## data_channel_is_entertainment -1.237649e+03
## data_channel_is_bus           -1.164383e+03
## data_channel_is_socmed        -5.030139e+02
## data_channel_is_tech          -4.754283e+02
## data_channel_is_world         -5.887924e+02
## kw_min_min                     1.603253e-01
## kw_max_min                     2.971068e-03
## kw_avg_min                     1.912590e-01
## kw_min_max                    -2.865725e-03
## kw_max_max                    -3.793539e-04
## kw_avg_max                     1.279919e-03
## kw_min_avg                    -1.458753e-01
## kw_max_avg                    -1.201892e-01
## kw_avg_avg                     1.169409e+00
## self_reference_min_shares      1.046404e-02
## self_reference_max_shares      5.439722e-03
## self_reference_avg_sharess    -1.336894e-03
## weekday_is_monday              3.051687e+02
## weekday_is_tuesday            -1.469474e+02
## weekday_is_wednesday           6.803334e+01
## weekday_is_thursday           -2.740554e+02
## weekday_is_friday             -8.383964e+01
## weekday_is_saturday            4.162125e+02
## weekday_is_sunday             -4.634198e+01
## is_weekend                     1.864841e+02
## LDA_00                         6.704297e+02
## LDA_01                        -1.523147e+02
## LDA_02                        -7.475333e+02
## LDA_03                         3.785326e+02
## LDA_04                        -1.157827e+02
## global_subjectivity            2.732672e+03
```

```
## global_sentiment_polarity      5.867844e+02
## global_rate_positive_words    -1.124933e+04
## global_rate_negative_words    -1.431345e+02
## rate_positive_words           -4.264183e+02
## rate_negative_words            2.460463e+02
## avg_positive_polarity         -7.882993e+02
## min_positive_polarity         -1.547223e+03
## max_positive_polarity          1.885535e+02
## avg_negative_polarity         -9.822432e+02
## min_negative_polarity          3.539597e+02
## max_negative_polarity         -4.301404e+02
## title_subjectivity             4.680582e+01
## title_sentiment_polarity       1.068170e+02
## abs_title_subjectivity         6.013256e+02
## abs_title_sentiment_polarity   7.890740e+02
```

From the result, we can see that the magnitudes of the coefficients from ridge regression are like that from simple linear regression. Moreover, there are no NA coefficients. To achieve best models, we need to use stepwise methods to perform variable selections. However, using stepwise methods may not solve the problem of multicollinearity and have high computation cost. Therefore, LASSO can be applied.

LASSO regression

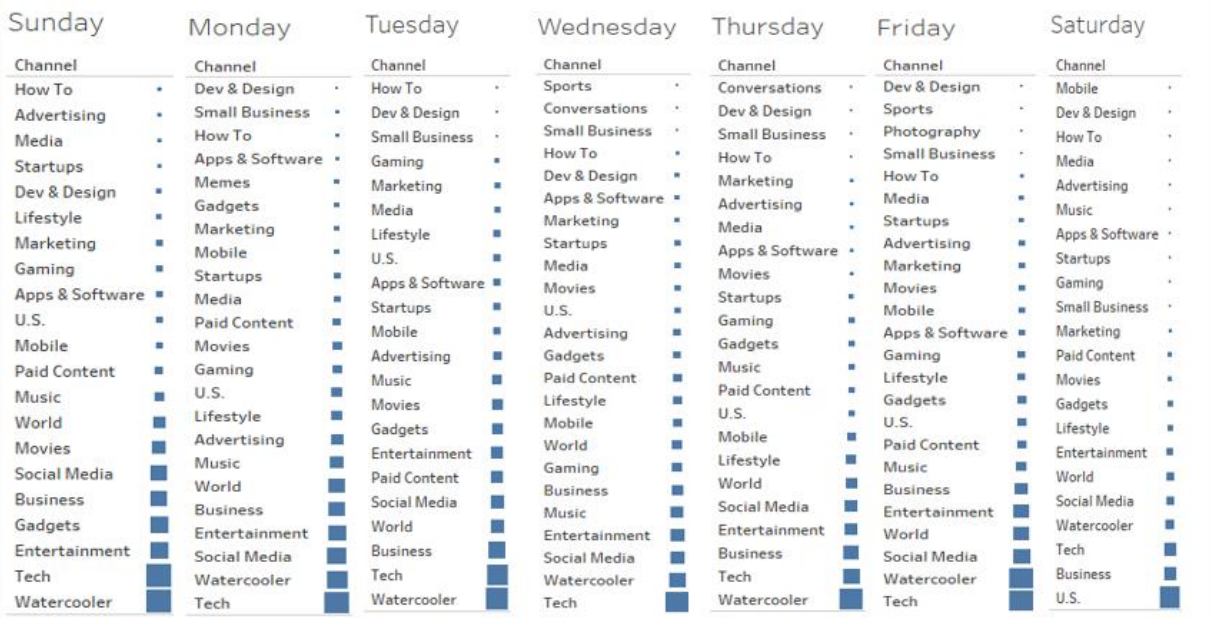
Similarly, we used 10-fold cross validation to determine the optimal lamda for LASSO regression. The result is as followed:

```
## 60 x 1 sparse Matrix of class "dgCMatrix"
##                                       1
## (Intercept)                  -2.328615e+03
## timedelta                     1.796363e+00
## n_tokens_title                1.063043e+02
## n_tokens_content              3.517763e-01
## n_unique_tokens               3.069667e+00
## n_non_stop_words              .
## n_non_stop_unique_tokens      .
## num_hrefs                     2.595584e+01
## num_self_hrefs               -5.603344e+01
## num_imgs                      5.839297e+00
## num_videos                    9.453074e+00
## average_token_length         -1.450603e+02
## num_keywords                  7.032808e+01
## data_channel_is_lifestyle    -7.007533e+02
## data_channel_is_entertainment -9.873508e+02
## data_channel_is_bus          -6.061569e+02
## data_channel_is_socmed       -9.425572e+01
## data_channel_is_tech         -4.779642e+01
## data_channel_is_world        -1.826990e+02
## kw_min_min                    .
## kw_max_min                    1.492264e-02
```

```
## kw_avg_min                    7.462019e-02
## kw_min_max                   -2.508783e-03
## kw_max_max                   -3.367600e-04
## kw_avg_max                    6.143643e-04
## kw_min_avg                   -1.992937e-01
## kw_max_avg                   -1.485516e-01
## kw_avg_avg                    1.383979e+00
## self_reference_min_shares     9.161530e-03
## self_reference_max_shares     4.666450e-03
## self_reference_avg_sharess    .
## weekday_is_monday             3.209097e+02
## weekday_is_tuesday           -7.072933e+01
## weekday_is_wednesday          8.301573e+01
## weekday_is_thursday          -1.998029e+02
## weekday_is_friday             .
## weekday_is_saturday           4.282896e+02
## weekday_is_sunday             .
## is_weekend                    1.575175e+02
## LDA_00                        4.319081e+02
## LDA_01                        .
## LDA_02                       -6.734806e+02
## LDA_03                        5.972577e+02
## LDA_04                       -1.443519e+02
## global_subjectivity           2.386597e+03
## global_sentiment_polarity     .
## global_rate_positive_words   -9.549831e+03
## global_rate_negative_words    .
## rate_positive_words          -2.657673e+02
## rate_negative_words           .
## avg_positive_polarity        -1.203068e+00
## min_positive_polarity        -1.510210e+03
## max_positive_polarity         .
## avg_negative_polarity        -2.668826e+02
## min_negative_polarity         3.137694e+00
## max_negative_polarity        -5.550329e+02
## title_subjectivity            .
## title_sentiment_polarity      2.917449e+01
## abs_title_subjectivity        4.443253e+02
## abs_title_sentiment_polarity  7.768818e+02
```

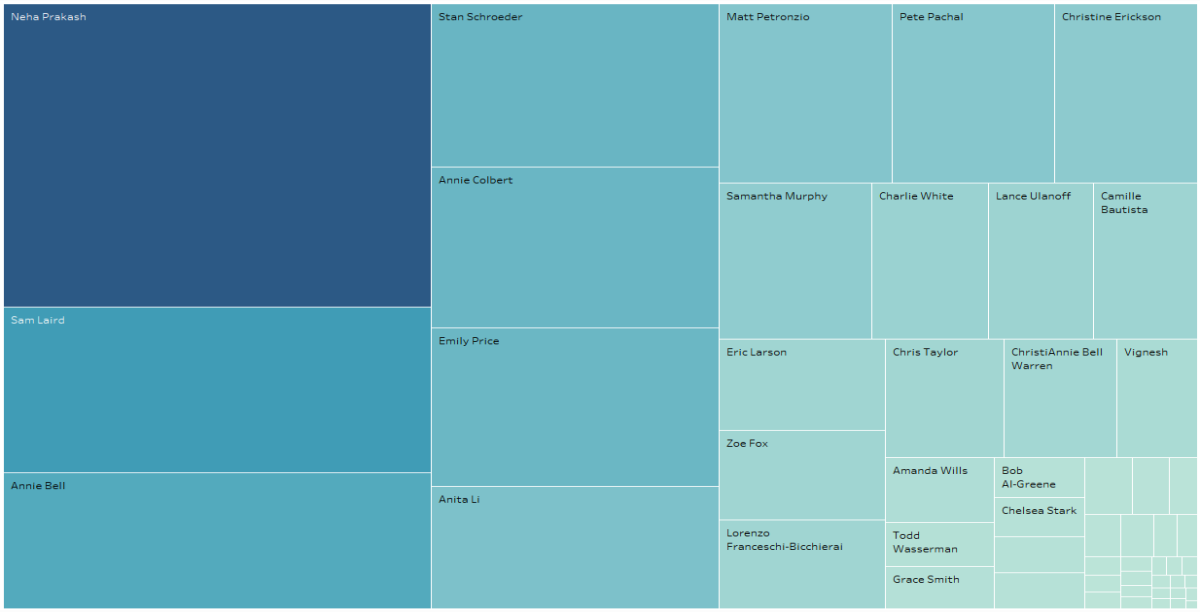
First, the mean square error of the prediction is 121612240, which is smaller than that from ridge regression. Thus, so far, lasso performs the best prediction. From the result, we can see LASSO performs variable selections and disregard 12 variables.

III. Data Visualization Analysis



This graph describes every topic's popularity sorted by the date, we can see that Tech is the most popular topic in Monday, Wednesday, Friday, and the Watercooler is popular in Sunday, Tuesday, Thursday. the lowest 2 topics is conversations and Dev&Design topics.

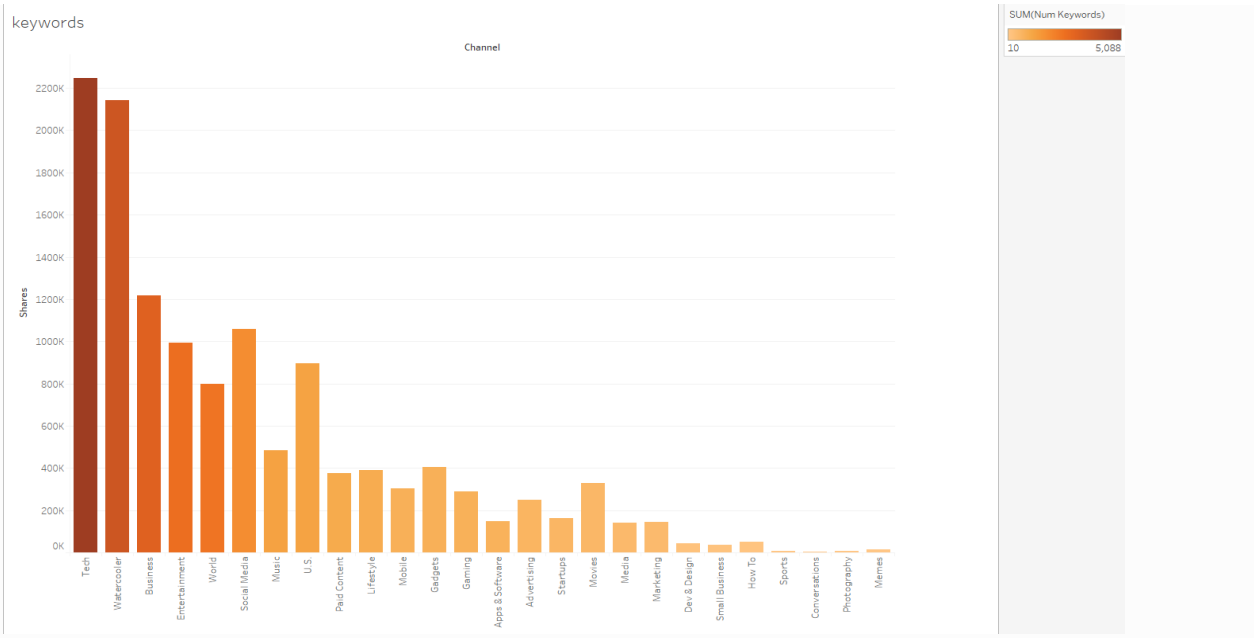
Then, we find that the top 2 topics in a week expect Saturday is Tech and WaterCooler, so we want to find why they are popular in most of days. Because those are all news, so we can analyze the author according to these news to find out the reason.



After pushing all those authors into the treemaps graph, we can find out that author Neha Prakash, Sam Laird and Annie Bell. In this way, the company could publish more those authors' article to increase the news' popularity.

Then we find that the lowest 2 topics is conversations and Dev&Design, so we compare then with top 2 topics to find the reason.

According to the graph above, we find that the top2 news topic have more number of links and keywords in the news. Because more links will give readers more choice, readers can read more related article following their own hearts. Putting more keywords rather than meaningful words will save readers time, let them know the news in a quick way, which may appeal to them a lot. In this way, the companies need pay more attention on those attributes.



Following the analysis above, the keywords are very important factor related to the main attribute ‘shares’. Here, we sort all the topics in the dataset by the number of keywords, in order to find out which attributes are more dependent on this property. To increase the number of shares, companies need add more keywords into the Tech, watercooler, and Business topics’ articles.

IV. Conclusion

Over the course of this paper we have attempted to find the model that can predict the popularity of fresh news using its features. We began the paper by discussing the reasoning behind the various determinants that we chose to examine. These were the amount of key words, number of linked embedded, the number of images, reference articles with high popularity. Our response variable was the amount of readers’ shares. We continue to do sample selection and data prep-processing .

Firstly, this paper analyses data mining model; classification. We have designed a model using C5.0 algorithm with 80:20 random sampling split for training and testing respectively. We handled feature-engineering methodology to convert the continuous to category target label, which was challenging and interesting. Even though, model’s overall accuracy is less (41 %), other performance measures show that the model is performing better. Model is performing above the diagonal line in ROC curve. However, in future studies, proper pruning will improve the model’s performance even better.

And about research question findings, News publishing firm can consider these findings in relation with the publications and thereby improve their business and profits effectively.

Then, this paper continues by analyzing several methods of data mining, regression. We employ three ways of regression, including linear regression, PCA regression, and Ridge and LASSO Regression in order to find a more precise model for predicting online news popularity. The results indicate that our best model is coming from Ridge and Lasso Regression since the mean square error of the prediction is 121612240.

Our PCA regression performance is not good, which mean square error is 123200737 due to less information provided and variable collinearity. From linear regression, we can get conclusion that we can increase the amount of key words, number of linked embedded, number of images, reference articles with high popularity, a more attractive title with relatively more words and publish more famous authors' articles, such as Neha Prakash, Sam Laird and Annie Bell, while decrease the average length of words in order to attract more readers.

We recognize that there are some limitations to our study. We didn't analyze every single possible variable that may have an effect on online news popularity,.We chose factors we thought would have the largest impact but our list is not exhaustive. Secondly we are limited to the data mining techniques that we used in the study. The 3 regression methods that we employed seemed to be the most logical methods to get predicted model for online news popularity. Again there are other methods that may have preformed better that we did not explore.

Based on our results we recommend that future studies focus on refining the model by including more independent variables, extending the time interval, currently we only collected data for 2 years. We also suggest that future studies explore what factors influence news with particular topic.

V. References

Anon, (2017). [online] Available at: https://www.linkedin.com/pulse/online-news-popularity-trend-analysis-krunal-khatri/.

Archive.ics.uci.edu. (2017). UCI Machine Learning Repository: Online News Popularity Data Set. [online] Available at: http://archive.ics.uci.edu/ml/datasets/Online+News+Popularity# .

He Ren and Quan Yang, 'Predicting and Evaluating the Popularity of Online News', Standford University Machine Learning Report, 2015.

K. Fernandes, P. Vinagre and P. Cortez. A Proactive In-telligent Decision Support System for Predicting the Pop-ularity of Online News. Proceedings of the 17th EPIA 2015- Portuguese Conference on Artificial Intelligence, September, Coimbra, Portugal.

https://en.wikipedia.org/wiki/C4.5_algorithm