



# PREDICTING ONLINE NEWS POPULARITY

- Jin, Muwen
  - Li, Qiuying
  - Wang, Nanjun
  - Yan, Kai
  - Yin, Jingwen
  - Zhu, Yuxin
- 

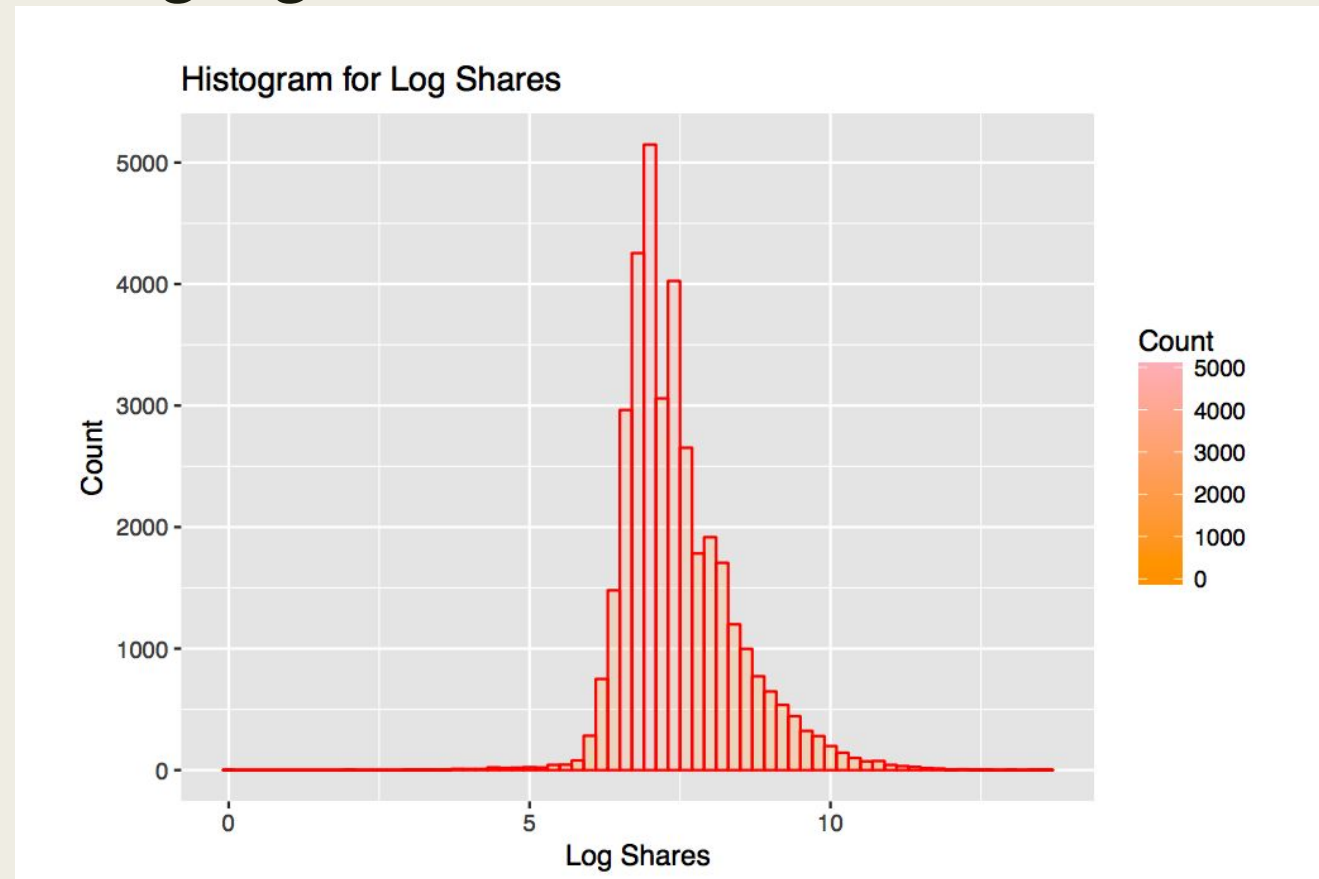
# Content

- Data & Features
- Exploratory Data Analysis
- Features Selection
- Machine Learning Models Comparison

Aspects	Features	
Words	<ul style="list-style-type: none"> <li>• Number of words in the title</li> <li>• Number of words in the article</li> <li>• Average words length</li> </ul>	<ul style="list-style-type: none"> <li>• Rate of non-stop words</li> <li>• Rate of unique words</li> <li>• Rate of unique non-stop words</li> </ul>
Links	<ul style="list-style-type: none"> <li>• Number of links</li> <li>• Number of Mashable article links</li> </ul>	<ul style="list-style-type: none"> <li>• Minimum, average and maximum number of shares of Mashable links</li> </ul>
Digital Media	<ul style="list-style-type: none"> <li>• Number of images</li> <li>• Number of videos</li> </ul>	
Publication Time	<ul style="list-style-type: none"> <li>• Day of the week</li> <li>• Published on a weekend</li> </ul>	
Key Words	<ul style="list-style-type: none"> <li>• Number of key words</li> <li>• Worst keyword</li> <li>• Average keyword</li> </ul>	<ul style="list-style-type: none"> <li>• Best keywords</li> <li>• Article category</li> </ul>
NLP	<ul style="list-style-type: none"> <li>• Closeness to top 5 LDA topics</li> <li>• Title subjectivity</li> <li>• Article text subjectivity score and its absolute difference to 0.5</li> <li>• Title sentiment polarity</li> <li>• Rate of positive and negative words</li> </ul>	<ul style="list-style-type: none"> <li>• Rate of positive and negative words</li> <li>• Neg. words rate among non-neutral words</li> <li>• Polarity of positive words (min./avg./max.)</li> <li>• Polarity of negative words (min./avg./max.)</li> <li>• Article text polarity score and its absolute difference to 0.5</li> </ul>
Response	Number of article shares	

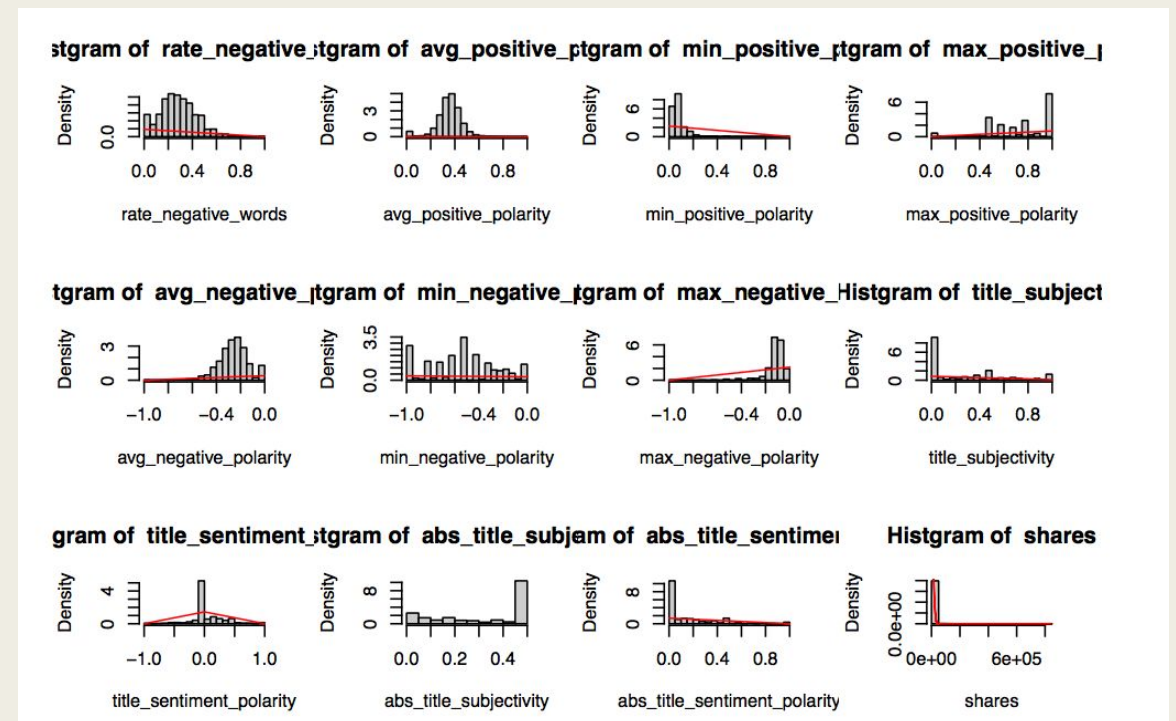
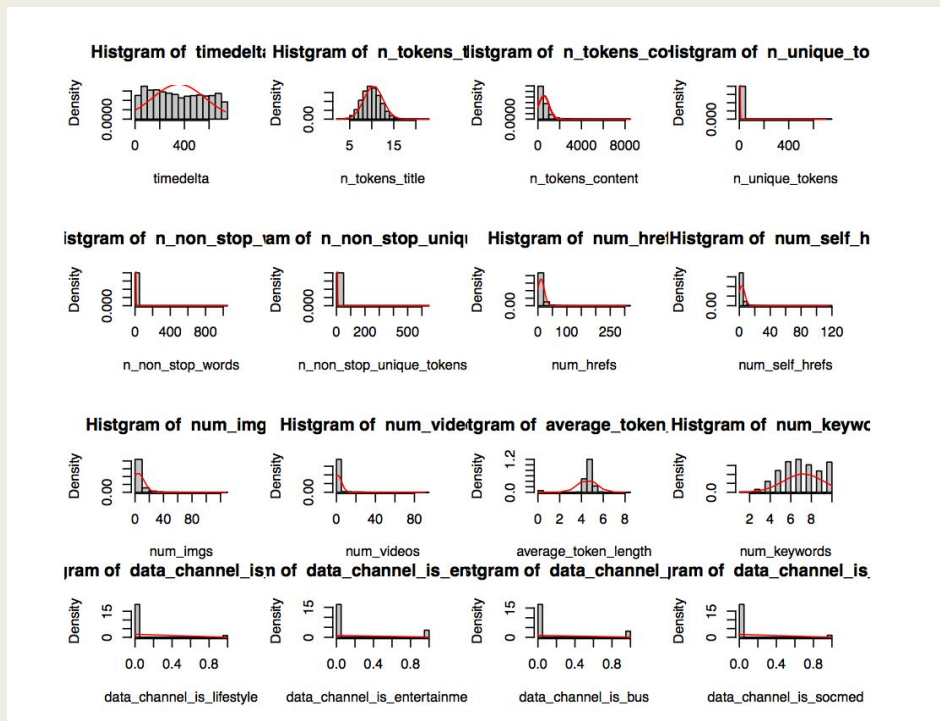
# EDA continues

2. Data Transformation: Transform heavily skewed response variable Y using log transformation.



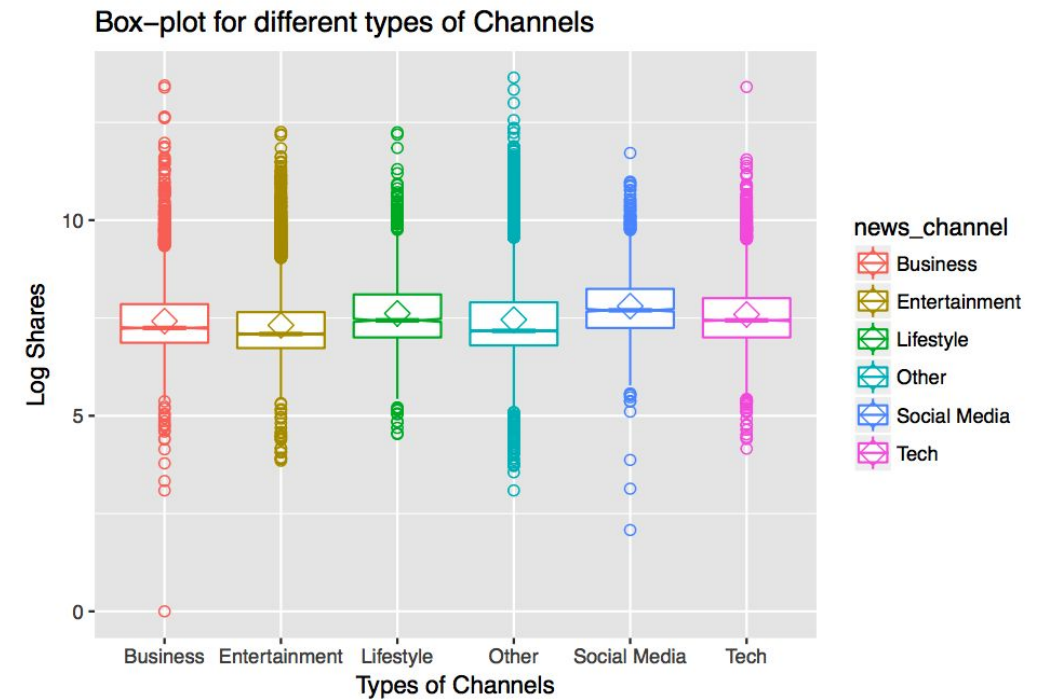
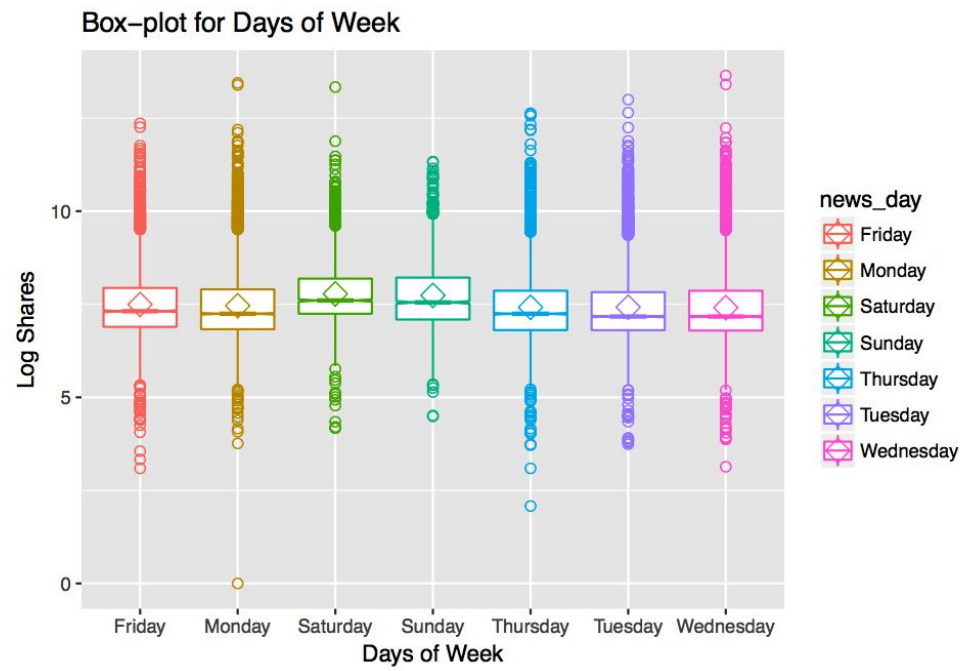
# Exploratory Data Analysis

## 1. Data Distribution : Plot the histograms to check distributions.



# EDA continues

## 3. Check interesting variables.



# EDA continues

## 4. Other Modifications:

- a. 19th, 21st, 23rd, 25th variables contain negative values that cannot be explained by information available, so they will be removed.
- b. Outliers in variable “n\_unique\_tokens”, “n\_non\_stop\_words”, and “n\_non\_stop\_unique\_tokens”, may due to typing error as they are the ratios and should have a value between 0 and 1. We remove those observations.
- c. The number of total variables left is 49.

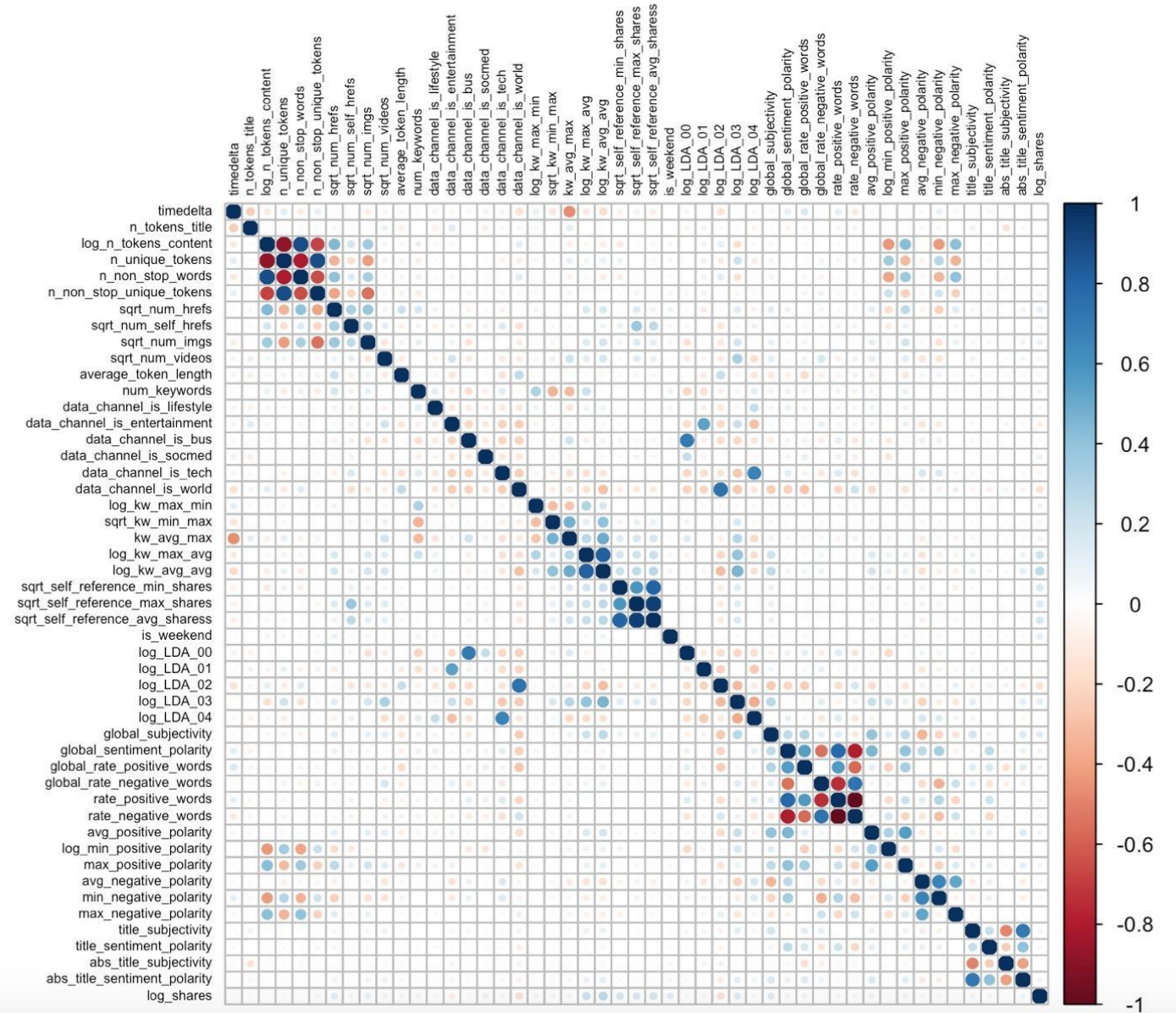
# Features Selection

- **Remove redundant variables**

- We consider pair-wise correlations among predictor variables  
( cutoff  $> 0.5$  )



# Correlation

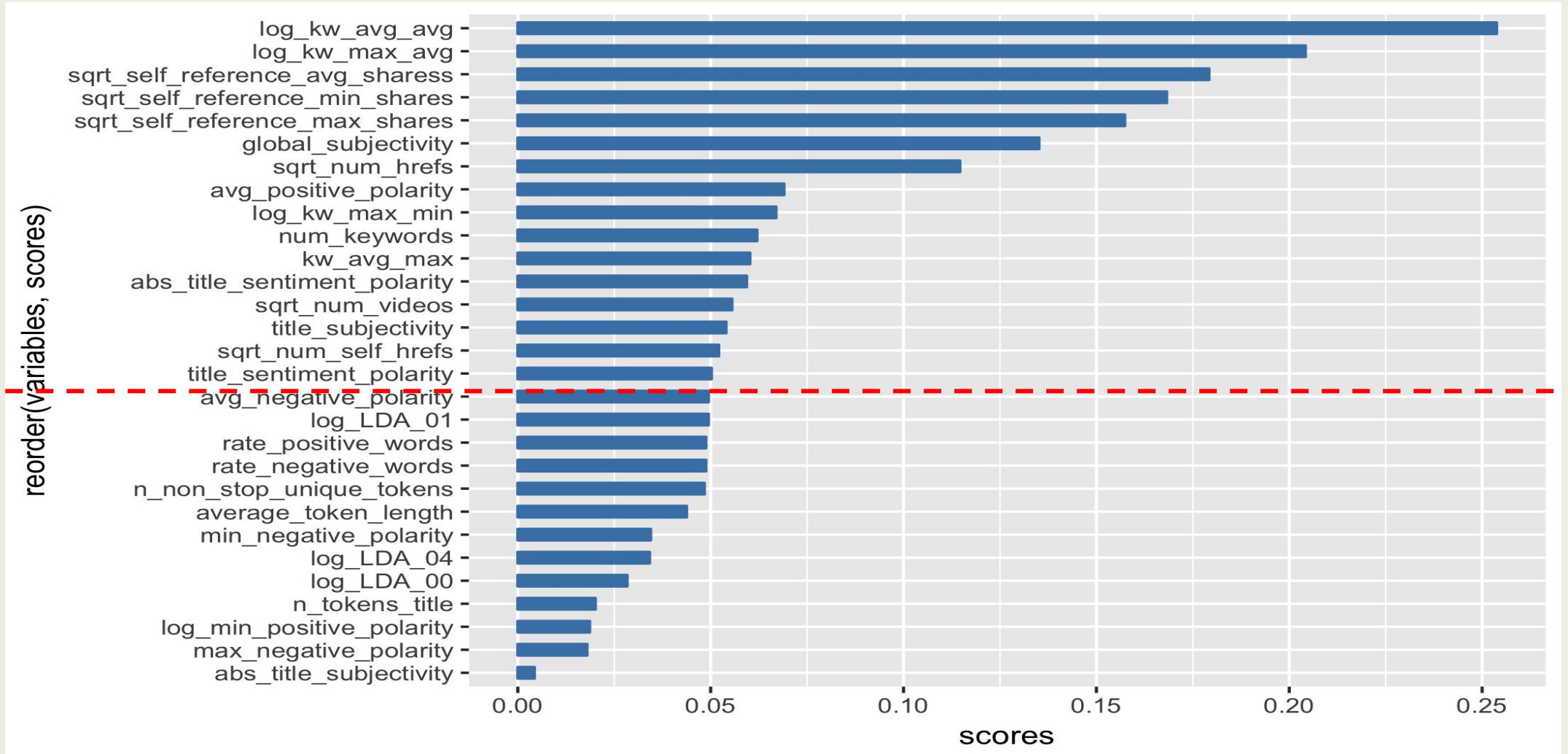


# Features Selection

## ■ Rank variables by importance

- We use Pearson's correlation as the indicator to measure the linear relationship between continuous variables and the response variable. Then rank them from the highest correlated to the lowest correlated.

# Importance rank



# Selected Variables

Aspects	Features
Keywords	<ul style="list-style-type: none"><li>• kw_avg_avg: Avg. keyword (avg. shares)</li><li>• kw_max_avg: Avg. keyword (max. shares)</li><li>• kw_avg_max: Best keyword (avg. shares)</li><li>• num_keywords: Number of keywords in the metadata</li><li>• data_channel_is_lifestyle: Is data channel 'Lifestyle'?</li><li>• data_channel_is_entertainment: Is data channel 'Entertainment'?</li><li>• data_channel_is_bus: Is data channel 'Business'?</li><li>• data_channel_is_socmed: Is data channel 'Social Media'?</li></ul>
Links	<ul style="list-style-type: none"><li>• self_reference_avg_sharess: Avg. shares of referenced articles in Mashable</li><li>• self_reference_min_shares: Min. shares of referenced articles in Mashable</li><li>• self_reference_max_shares: Max. shares of referenced articles in Mashable</li><li>• num_hrefs: Number of links</li><li>• num_self_hrefs: Number of links to other articles published by Mashable</li></ul>
NLP	<ul style="list-style-type: none"><li>• global_subjectivity: Text subjectivity</li><li>• avg_positive_polarity: Avg. polarity of positive words</li><li>• abs_title_sentiment_polarity: Absolute polarity level</li><li>• title_subjectivity: Title subjectivity</li></ul>
Digital Media	<ul style="list-style-type: none"><li>• num_videos: Number of videos</li><li>• abs_title_sentiment_polarity: Absolute polarity level</li><li>• title_sentiment_polarity: Title polarity</li></ul>

# Machine Learning Models Comparison

- Linear Regression
- Support Vector Machine
- Regression Tree
- Random Forest
- Gradient Boosting



# Linear Regression

**MSE = 0.7841**

```
Residuals:
    Min       1Q   Median       3Q      Max
-8.0883 -0.5541 -0.1601  0.4073  5.6924

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.521e+00  1.505e-01  10.109 < 2e-16 ***
log_kw_avg_avg  8.571e-01  3.364e-02  25.482 < 2e-16 ***
log_kw_max_avg -1.456e-01  2.296e-02  -6.343 2.29e-10 ***
sqrt_self_reference_avg_shares 1.760e-03  6.231e-04   2.825 0.004734 **
sqrt_self_reference_min_shares 1.010e-03  3.230e-04   3.126 0.001775 **
sqrt_self_reference_max_shares -4.517e-04  3.545e-04  -1.274 0.202580
global_subjectivity  5.527e-01  6.902e-02   8.007 1.22e-15 ***
sqrt_num_hrefs  3.801e-02  4.339e-03   8.759 < 2e-16 ***
avg_positive_polarity -8.964e-02  7.006e-02  -1.279 0.200738
log_kw_max_min -9.312e-03  6.565e-03  -1.418 0.156109
num_keywords  1.800e-02  3.311e-03   5.435 5.53e-08 ***
kw_avg_max -7.069e-07  5.847e-08 -12.090 < 2e-16 ***
abs_title_sentiment_polarity  1.572e-02  3.587e-02   0.438 0.661133
sqrt_num_videos  2.448e-02  5.739e-03   4.265 2.01e-05 ***
title_subjectivity  3.482e-02  2.355e-02   1.478 0.139289
sqrt_num_self_hrefs -4.080e-03  7.043e-03  -0.579 0.562419
title_sentiment_polarity  7.983e-02  2.197e-02   3.634 0.000279 ***
data_channel_is_lifestyle1 -2.853e-02  2.402e-02  -1.188 0.234880
data_channel_is_entertainment1 -2.197e-01  1.486e-02 -14.780 < 2e-16 ***
data_channel_is_bus1  4.306e-02  1.576e-02   2.733 0.006285 **
data_channel_is_socmed1  2.108e-01  2.348e-02   8.980 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8694 on 27184 degrees of freedom
Multiple R-squared:  0.1111,    Adjusted R-squared:  0.1104
F-statistic: 169.9 on 20 and 27184 DF,  p-value: < 2.2e-16
```

# Support Vector Machine

- The value of **epsilon** defines a margin of tolerance where no penalty is given to errors.
- **Cost** is a parameter allows one to trade off training error vs. model complexity.
- **Sigma** determines the width of Gaussian distribution.

Kernel	Parameters	Expression
Linear	$\epsilon = 0.1,$ $Cost = 1$	$K(u, v) = u^T v$
Gaussian	$\epsilon = 0.1$ $Cost = 0.5,$ $\sigma = 0.04$	$K(u, v) = \exp(\frac{- u - v ^2}{2\sigma^2})$

# Support Vector Machine

Algorithms	MSE	Training Time
SVM Linear	0.82	550s
SVM Gaussian	0.79	255s



# Regression Tree- rpart (recursive partitioning)

A rpart model with a continuous response (an anova model).

Each node shows

- the predicted value
- the percentage of observations in the node.

12 splits, 53 nodes

# rpart - CP(complexity parameters)

```
> summary(c_fit)
Call:
rpart(formula = log(shares) ~ ., data = cat, method = "anova",
      control = rpart.control(minsplit = 10, minbucket = 4, cp = 0.002))
n= 31715
```

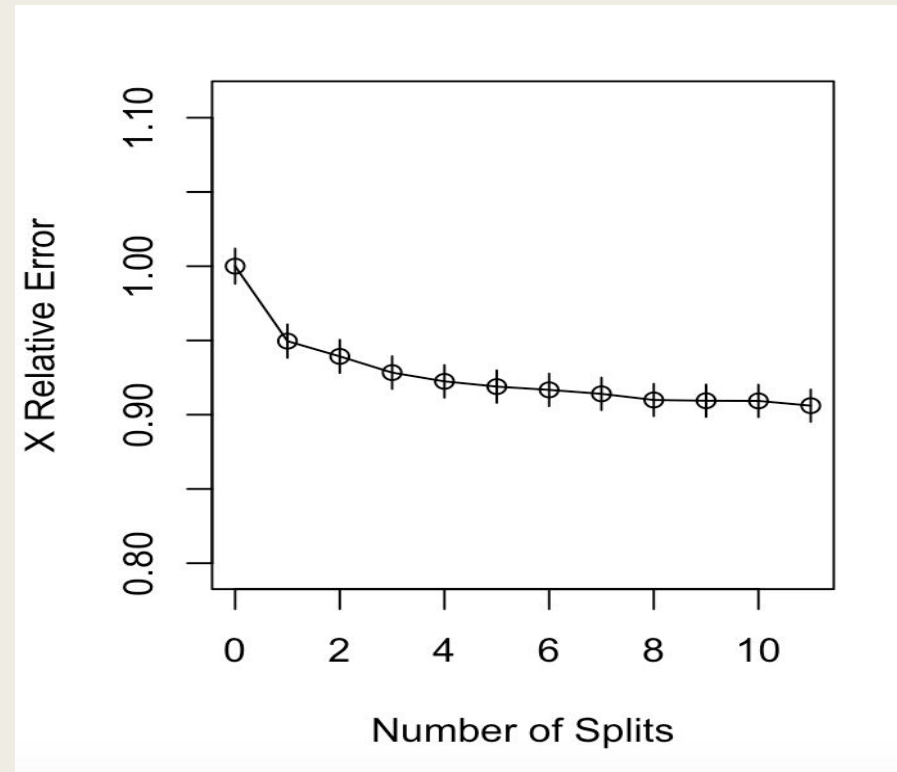
	CP	nsplit	rel error	xerror	xstd
1	0.052938572	0	1.0000000	1.0000200	0.01183431
2	0.011582417	1	0.9470614	0.9495463	0.01126572
3	0.011169415	2	0.9354790	0.9392656	0.01113900
4	0.006287195	3	0.9243096	0.9283532	0.01101877
5	0.004176868	4	0.9180224	0.9224415	0.01096581
6	0.003725943	5	0.9138455	0.9189170	0.01092625
7	0.003407788	6	0.9101196	0.9167224	0.01095825
8	0.003091044	7	0.9067118	0.9140441	0.01093393
9	0.002376130	8	0.9036208	0.9099215	0.01088119
10	0.002374690	9	0.9012446	0.9093941	0.01089503
11	0.002123137	10	0.8988699	0.9092912	0.01089433
12	0.002000000	11	0.8967468	0.9060781	0.01087814

1. cp is the amount by which splitting that node improved the relative error.

2. The relative error is  $(1 - R^2)$ , similar to linear regression.

3. The xerror is related to the PRESS statistic.

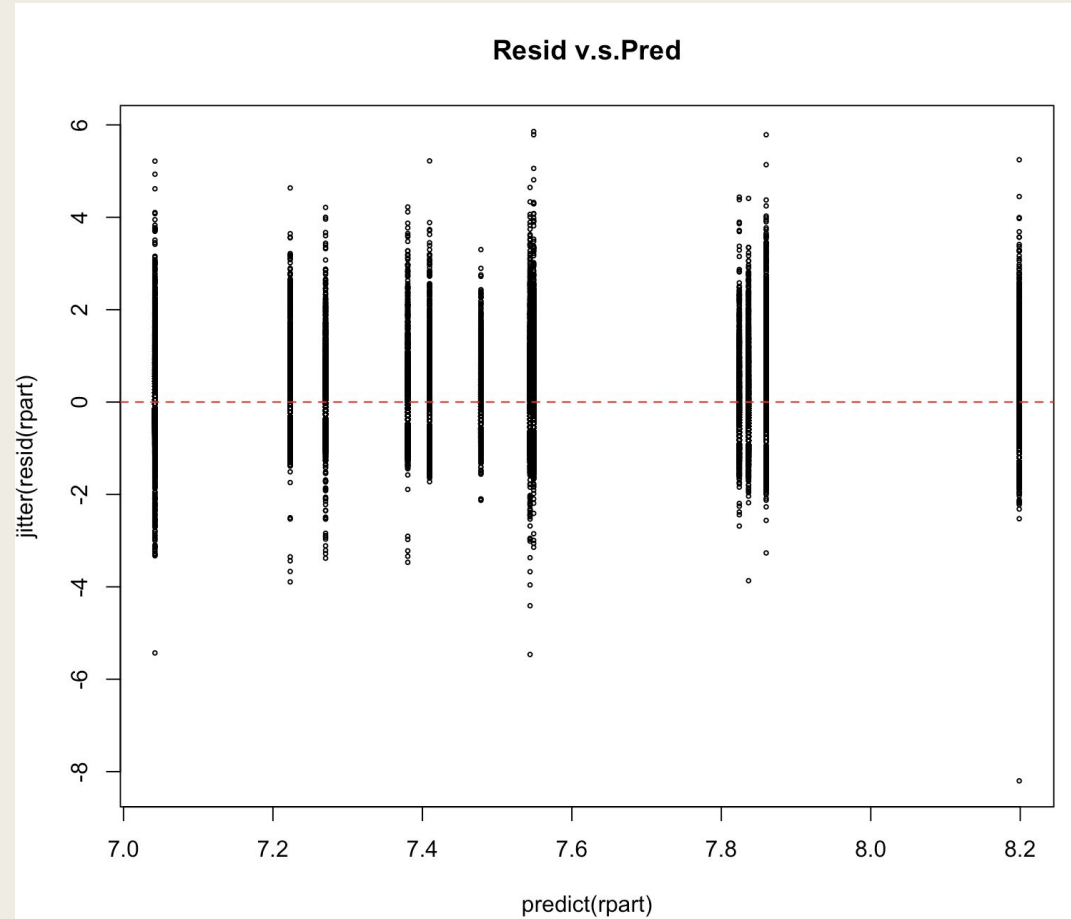
# Regression Tree- rpart



1. The first split and second appears to improve the fit the most.

2. The figure on the right shows the tree should be pruned to include only 1 or 2 splits

# rpart- prediction accuracy



**MSE = 0.8108**

This plot shows the residuals of predicted shares v.s. the predicted shares based on the nodes/leaves.

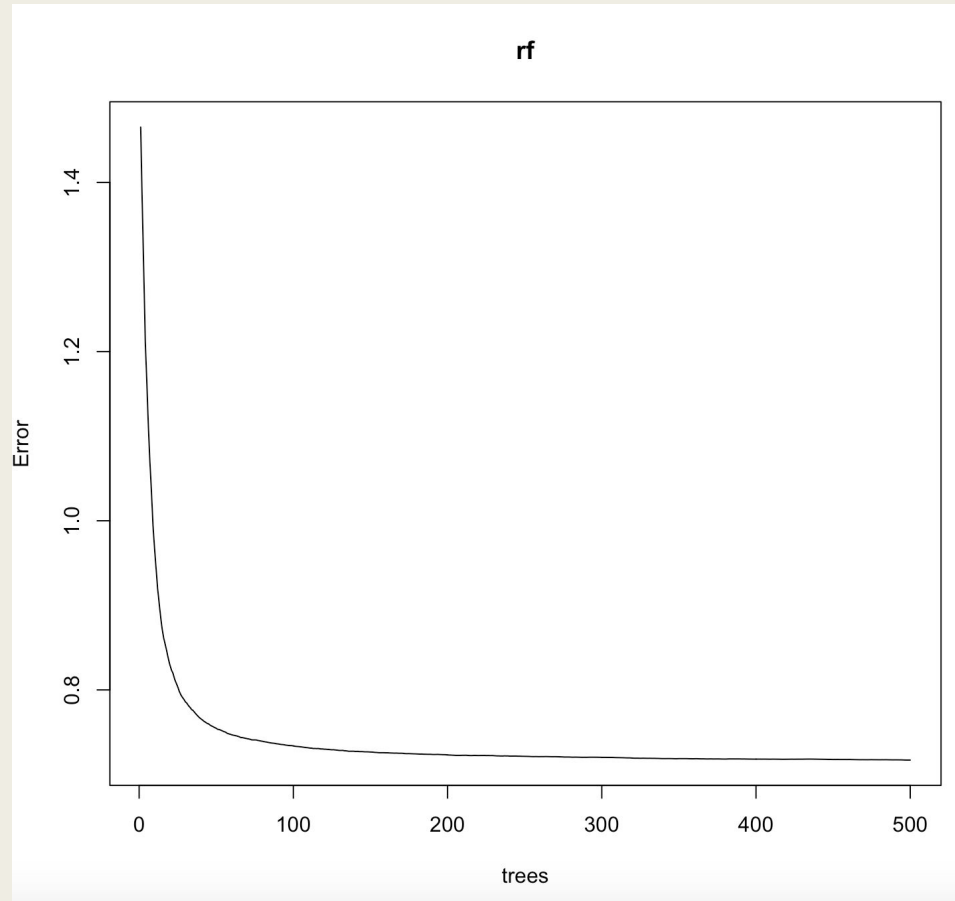
# Why we need a random Forest?

“Given its performance, random forest and variable selection using random forest should probably become part of the standard tool-box of methods for the analysis of microarray data.”

----- Ramón Díaz-Uriarte

(2010)

# RandomForest



Improve on rpart with respect to:

- Accuracy: Random Forests  
test error = 0.7569 smaller than rpart, which is 0.8108
- Stability : If we change the data a little, random forest is relatively stable because it is a combination of many trees.

# Gradient Boosting

- Machine learning technique for regression and classification problems
- In the form of an ensemble of weak prediction models, typically decision trees
- Optimization of an arbitrary differentiable loss function

n. trees	interaction.depth	shrinkage	n.minobsinnode
150	3	0.1	10

<b>MSE</b>	<b>0.7582</b>
------------	---------------

# Comparison

Algorithms	MSE	Training Time (Without Cross Validation)
Linear Regression	0.7841	<1min
Support Vector Machine (Linear)	0.8200	10min
Support Vector Machine (Gaussian)	0.7856	4min
Random Forest	0.7569	3min
Regression Tree	0.8108	<1min
Gradient Boosting	0.7582	3min