Online News Popularity Prediction - Project Code

G6

4/11/2017

Contents

1 Introduction **2**

2 Data preparation **2**

2.1 Loading libraries 2

2.2 Loading dataset 2

2.3 Data Analysis 4

2.4 Data Distribution 17

2.5 Data Sampling 19

2.6 Initial Regression Analysis 20

2.7 Data Cleanup 43

3 Data Modelling **45**

3.1 Logistic Regression - All 52 predictors 45

3.2 Logistic Regression - Top 30 predictors 50

3.3 Regularization & Cross Validation 55

3.4 SVM - All 52 predictors 65

3.5 SVM - Top 30 predictors 68

3.6 SVM - Cross-Validation 71

3.7 Random Forest - All 52 Predictors 73

3.8 Random Forest - Top 30 Predictors 76

3.9 Random Forest - Channel as Categorical 79

3.10 Random Forest - Cross-Validation 82

3.11 Naive Bayes - All 52 Predictors 85

3.12 Naive Bayes - Top 30 Predictors 92

3.13 Naive Bayes - Cross Validation 97

4 Artificially unbalancing the data **101**

5 Results **116**

1 Introduction

For this analysis, we have used R programming language. Each section has R code embedded under it which was used for analyzing that component.

Please refer here for description of the different columns.

2 Data preparation

2.1 Loading libraries

All the libraries required for the analysis are loaded here (at the start).

```
#######################################################################
# Load Libraries
#######################################################################
library(bestglm)
library(glmnet)
library(ROCR)
library(caret)
library(e1071)
library(kernlab)
library(woe)
library(randomForest)
```

2.2 Loading dataset

Here, we have downloaded the data locally and are loading the data directly from a local folder. We are also checking some statistics about the data and top few rows.

```
#######################################################################
# Preparation
#######################################################################
## Load data
set.seed(123)
setwd[redacted]
news.data <- read.csv("OnlineNewsPopularity.csv", sep = ",", header = TRUE)
#news.data <- read.csv(url("https://archive.ics.uci.edu/ml/machine-learning-databases/00332/OnlineNewsP

## Inspect data
nrow(news.data)
```

Masked user directory for anonimity

```
## [1] 39644
```

```
names(news.data) # column names
```

```
##  [1] "url"                          "timedelta"
##  [3] "n_tokens_title"               "n_tokens_content"
##  [5] "n_unique_tokens"              "n_non_stop_words"
##  [7] "n_non_stop_unique_tokens"     "num_hrefs"
```

```
##  [9] "num_self_hrefs"                "num_imgs"
## [11] "num_videos"                    "average_token_length"
## [13] "num_keywords"                  "data_channel_is_lifestyle"
## [15] "data_channel_is_entertainment" "data_channel_is_bus"
## [17] "data_channel_is_socmed"        "data_channel_is_tech"
## [19] "data_channel_is_world"         "kw_min_min"
## [21] "kw_max_min"                    "kw_avg_min"
## [23] "kw_min_max"                    "kw_max_max"
## [25] "kw_avg_max"                    "kw_min_avg"
## [27] "kw_max_avg"                    "kw_avg_avg"
## [29] "self_reference_min_shares"     "self_reference_max_shares"
## [31] "self_reference_avg_sharess"    "weekday_is_monday"
## [33] "weekday_is_tuesday"            "weekday_is_wednesday"
## [35] "weekday_is_thursday"           "weekday_is_friday"
## [37] "weekday_is_saturday"           "weekday_is_sunday"
## [39] "is_weekend"                    "LDA_00"
## [41] "LDA_01"                        "LDA_02"
## [43] "LDA_03"                        "LDA_04"
## [45] "global_subjectivity"           "global_sentiment_polarity"
## [47] "global_rate_positive_words"    "global_rate_negative_words"
## [49] "rate_positive_words"           "rate_negative_words"
## [51] "avg_positive_polarity"         "min_positive_polarity"
## [53] "max_positive_polarity"         "avg_negative_polarity"
## [55] "min_negative_polarity"         "max_negative_polarity"
## [57] "title_subjectivity"            "title_sentiment_polarity"
## [59] "abs_title_subjectivity"        "abs_title_sentiment_polarity"
## [61] "shares"
```

```r
head(news.data, 2)
```

```
##                                                        url timedelta
## 1 http://mashable.com/2013/01/07/amazon-instant-video-browser/       731
## 2  http://mashable.com/2013/01/07/ap-samsung-sponsored-tweets/       731
##   n_tokens_title n_tokens_content n_unique_tokens n_non_stop_words
## 1             12              219       0.6635945                1
## 2              9              255       0.6047431                1
##   n_non_stop_unique_tokens num_hrefs num_self_hrefs num_imgs num_videos
## 1                0.8153846         4              2        1          0
## 2                0.7919463         3              1        1          0
##   average_token_length num_keywords data_channel_is_lifestyle
## 1             4.680365            5                          0
## 2             4.913725            4                          0
##   data_channel_is_entertainment data_channel_is_bus data_channel_is_socmed
## 1                             1                   0                      0
## 2                             0                   1                      0
##   data_channel_is_tech data_channel_is_world kw_min_min kw_max_min
## 1                    0                     0          0          0
## 2                    0                     0          0          0
##   kw_avg_min kw_min_max kw_max_max kw_avg_max kw_min_avg kw_max_avg
## 1          0          0          0          0          0          0
## 2          0          0          0          0          0          0
##   kw_avg_avg self_reference_min_shares self_reference_max_shares
## 1          0                       496                       496
## 2          0                         0                         0
```

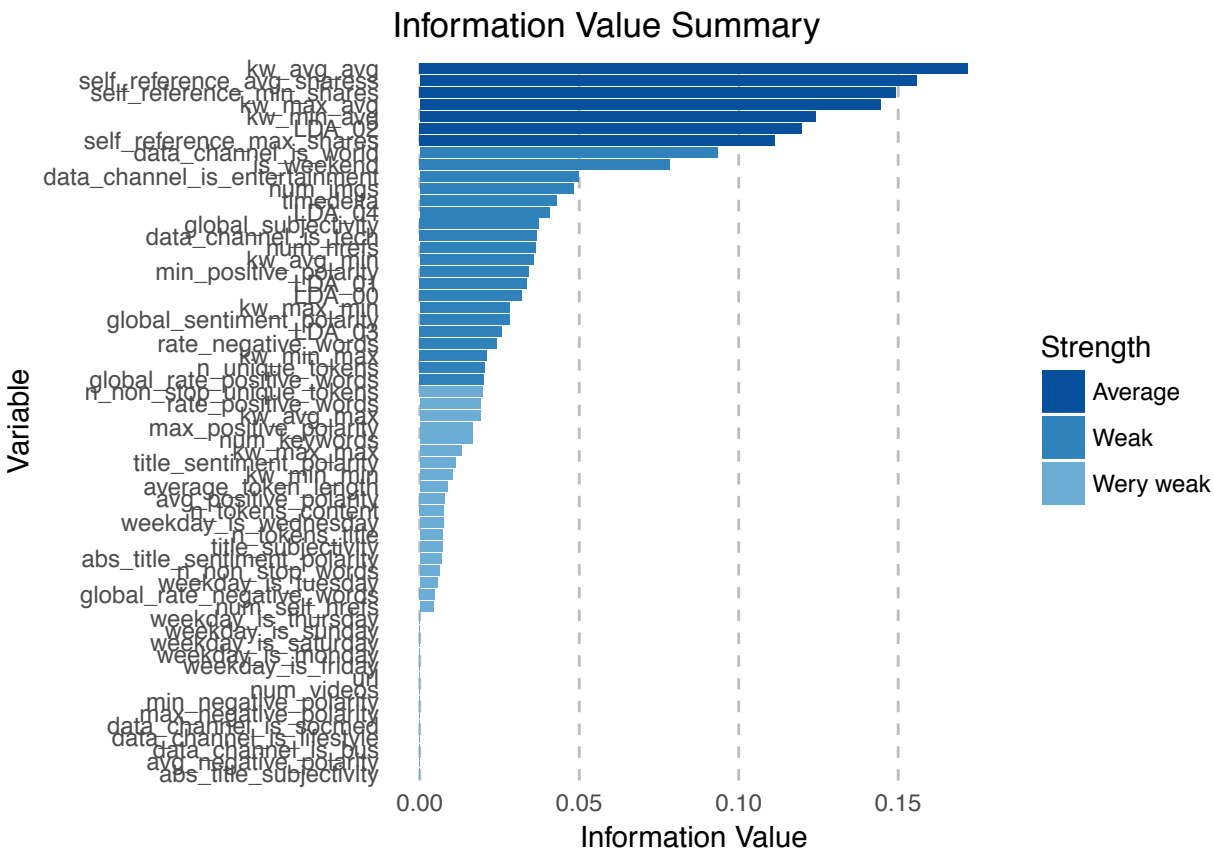
```
##   self_reference_avg_sharess weekday_is_monday weekday_is_tuesday
## 1                        496                 1                  0
## 2                          0                 1                  0
##   weekday_is_wednesday weekday_is_thursday weekday_is_friday
## 1                    0                   0                 0
## 2                    0                   0                 0
##   weekday_is_saturday weekday_is_sunday is_weekend    LDA_00     LDA_01
## 1                   0                 0          0 0.5003312 0.37827893
## 2                   0                 0          0 0.7997557 0.05004668
##       LDA_02     LDA_03     LDA_04 global_subjectivity
## 1 0.04000468 0.04126265 0.04012254           0.5216171
## 2 0.05009625 0.05010067 0.05000071           0.3412458
##   global_sentiment_polarity global_rate_positive_words
## 1                0.09256198                 0.04566210
## 2                0.14894781                 0.04313725
##   global_rate_negative_words rate_positive_words rate_negative_words
## 1                 0.01369863           0.7692308           0.2307692
## 2                 0.01568627           0.7333333           0.2666667
##   avg_positive_polarity min_positive_polarity max_positive_polarity
## 1             0.3786364            0.10000000                   0.7
## 2             0.2869146            0.03333333                   0.7
##   avg_negative_polarity min_negative_polarity max_negative_polarity
## 1              -0.35000                -0.600                  -0.2
## 2              -0.11875                -0.125                  -0.1
##   title_subjectivity title_sentiment_polarity abs_title_subjectivity
## 1                0.5                  -0.1875                    0.0
## 2                0.0                   0.0000                    0.5
##   abs_title_sentiment_polarity shares
## 1                       0.1875    593
## 2                       0.0000    711
```

2.3 Data Analysis

We have used different methods for initial data analysis as shown below. Apart from ths, we also used Microsoft Excel for manually inspecting the data for irregularities.

```
#######################################################################
# Data Analysis
#######################################################################
## Information Value
data <- news.data
data$shares <- ifelse(data$shares > 1400, 1, 0)
rownames(data) = NULL
iv_df <- iv.mult(data, y="shares", summary=TRUE, verbose=TRUE)
```

```
iv.plot.summary(iv_df)
```




```
## Remove URL column (Non-numeric and 0 IV)
news.data <- news.data[, -1]

## One Variable Stats
summary(news.data)
```

```
##    timedelta     n_tokens_title n_tokens_content n_unique_tokens
##  Min.   :  8.0   Min.   : 2.0   Min.   :   0.0   Min.   :  0.0000
##  1st Qu.:164.0   1st Qu.: 9.0   1st Qu.: 246.0   1st Qu.:  0.4709
##  Median :339.0   Median :10.0   Median : 409.0   Median :  0.5392
##  Mean   :354.5   Mean   :10.4   Mean   : 546.5   Mean   :  0.5482
##  3rd Qu.:542.0   3rd Qu.:12.0   3rd Qu.: 716.0   3rd Qu.:  0.6087
##  Max.   :731.0   Max.   :23.0   Max.   :8474.0   Max.   :701.0000
##  n_non_stop_words    n_non_stop_unique_tokens   num_hrefs
##  Min.   :   0.0000   Min.   :  0.0000         Min.   :  0.00
##  1st Qu.:   1.0000   1st Qu.:  0.6257         1st Qu.:  4.00
##  Median :   1.0000   Median :  0.6905         Median :  8.00
##  Mean   :   0.9965   Mean   :  0.6892         Mean   : 10.88
##  3rd Qu.:   1.0000   3rd Qu.:  0.7546         3rd Qu.: 14.00
##  Max.   :1042.0000   Max.   :650.0000         Max.   :304.00
##  num_self_hrefs       num_imgs         num_videos    average_token_length
##  Min.   :  0.000   Min.   :  0.000   Min.   : 0.00   Min.   :0.000
##  1st Qu.:  1.000   1st Qu.:  1.000   1st Qu.: 0.00   1st Qu.:4.478
##  Median :  3.000   Median :  1.000   Median : 0.00   Median :4.664
##  Mean   :  3.294   Mean   :  4.544   Mean   : 1.25   Mean   :4.548
##  3rd Qu.:  4.000   3rd Qu.:  4.000   3rd Qu.: 1.00   3rd Qu.:4.855
```

```
##  Max.   :116.000   Max.   :128.000   Max.   :91.00   Max.   :8.042
##   num_keywords    data_channel_is_lifestyle data_channel_is_entertainment
##  Min.   : 1.000   Min.   :0.00000           Min.   :0.000
##  1st Qu.: 6.000   1st Qu.:0.00000           1st Qu.:0.000
##  Median : 7.000   Median :0.00000           Median :0.000
##  Mean   : 7.224   Mean   :0.05295           Mean   :0.178
##  3rd Qu.: 9.000   3rd Qu.:0.00000           3rd Qu.:0.000
##  Max.   :10.000   Max.   :1.00000           Max.   :1.000
##  data_channel_is_bus data_channel_is_socmed data_channel_is_tech
##  Min.   :0.0000      Min.   :0.0000         Min.   :0.0000
##  1st Qu.:0.0000      1st Qu.:0.0000         1st Qu.:0.0000
##  Median :0.0000      Median :0.0000         Median :0.0000
##  Mean   :0.1579      Mean   :0.0586         Mean   :0.1853
##  3rd Qu.:0.0000      3rd Qu.:0.0000         3rd Qu.:0.0000
##  Max.   :1.0000      Max.   :1.0000         Max.   :1.0000
##  data_channel_is_world   kw_min_min       kw_max_min       kw_avg_min
##  Min.   :0.0000        Min.   : -1.00   Min.   :     0   Min.   :   -1.0
##  1st Qu.:0.0000        1st Qu.: -1.00   1st Qu.:   445   1st Qu.:  141.8
##  Median :0.0000        Median : -1.00   Median :   660   Median :  235.5
##  Mean   :0.2126        Mean   : 26.11   Mean   :  1154   Mean   :  312.4
##  3rd Qu.:0.0000        3rd Qu.:  4.00   3rd Qu.:  1000   3rd Qu.:  357.0
##  Max.   :1.0000        Max.   :377.00   Max.   :298400   Max.   :42827.9
##    kw_min_max       kw_max_max       kw_avg_max       kw_min_avg
##  Min.   :     0   Min.   :     0   Min.   :     0   Min.   :  -1
##  1st Qu.:     0   1st Qu.:843300   1st Qu.:172847   1st Qu.:   0
##  Median :  1400   Median :843300   Median :244572   Median :1024
##  Mean   : 13612   Mean   :752324   Mean   :259282   Mean   :1117
##  3rd Qu.:  7900   3rd Qu.:843300   3rd Qu.:330980   3rd Qu.:2057
##  Max.   :843300   Max.   :843300   Max.   :843300   Max.   :3613
##    kw_max_avg       kw_avg_avg    self_reference_min_shares
##  Min.   :     0   Min.   :    0   Min.   :     0
##  1st Qu.:  3562   1st Qu.: 2382   1st Qu.:   639
##  Median :  4356   Median : 2870   Median :  1200
##  Mean   :  5657   Mean   : 3136   Mean   :  3999
##  3rd Qu.:  6020   3rd Qu.: 3600   3rd Qu.:  2600
##  Max.   :298400   Max.   :43568   Max.   :843300
##  self_reference_max_shares self_reference_avg_sharess weekday_is_monday
##  Min.   :     0            Min.   :     0.0           Min.   :0.000
##  1st Qu.:  1100            1st Qu.:   981.2           1st Qu.:0.000
##  Median :  2800            Median :  2200.0           Median :0.000
##  Mean   : 10329            Mean   :  6401.7           Mean   :0.168
##  3rd Qu.:  8000            3rd Qu.:  5200.0           3rd Qu.:0.000
##  Max.   :843300            Max.   :843300.0           Max.   :1.000
##  weekday_is_tuesday weekday_is_wednesday weekday_is_thursday
##  Min.   :0.0000     Min.   :0.0000       Min.   :0.0000
##  1st Qu.:0.0000     1st Qu.:0.0000       1st Qu.:0.0000
##  Median :0.0000     Median :0.0000       Median :0.0000
##  Mean   :0.1864     Mean   :0.1875       Mean   :0.1833
##  3rd Qu.:0.0000     3rd Qu.:0.0000       3rd Qu.:0.0000
##  Max.   :1.0000     Max.   :1.0000       Max.   :1.0000
##  weekday_is_friday weekday_is_saturday weekday_is_sunday   is_weekend
##  Min.   :0.0000    Min.   :0.00000     Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.0000    1st Qu.:0.00000     1st Qu.:0.00000   1st Qu.:0.0000
##  Median :0.0000    Median :0.00000     Median :0.00000   Median :0.0000
```

```
##  Mean   :0.1438    Mean   :0.06188     Mean   :0.06904   Mean   :0.1309
##  3rd Qu.:0.0000    3rd Qu.:0.00000     3rd Qu.:0.00000   3rd Qu.:0.0000
##  Max.   :1.0000    Max.   :1.00000     Max.   :1.00000   Max.   :1.0000
##      LDA_00            LDA_01            LDA_02            LDA_03
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.02505   1st Qu.:0.02501   1st Qu.:0.02857   1st Qu.:0.02857
##  Median :0.03339   Median :0.03334   Median :0.04000   Median :0.04000
##  Mean   :0.18460   Mean   :0.14126   Mean   :0.21632   Mean   :0.22377
##  3rd Qu.:0.24096   3rd Qu.:0.15083   3rd Qu.:0.33422   3rd Qu.:0.37576
##  Max.   :0.92699   Max.   :0.92595   Max.   :0.92000   Max.   :0.92653
##      LDA_04        global_subjectivity global_sentiment_polarity
##  Min.   :0.00000   Min.   :0.0000      Min.   :-0.39375
##  1st Qu.:0.02857   1st Qu.:0.3962      1st Qu.: 0.05776
##  Median :0.04073   Median :0.4535      Median : 0.11912
##  Mean   :0.23403   Mean   :0.4434      Mean   : 0.11931
##  3rd Qu.:0.39999   3rd Qu.:0.5083      3rd Qu.: 0.17783
##  Max.   :0.92719   Max.   :1.0000      Max.   : 0.72784
##  global_rate_positive_words global_rate_negative_words rate_positive_words
##  Min.   :0.00000            Min.   :0.000000           Min.   :0.0000
##  1st Qu.:0.02838            1st Qu.:0.009615           1st Qu.:0.6000
##  Median :0.03902            Median :0.015337           Median :0.7105
##  Mean   :0.03962            Mean   :0.016612           Mean   :0.6822
##  3rd Qu.:0.05028            3rd Qu.:0.021739           3rd Qu.:0.8000
##  Max.   :0.15549            Max.   :0.184932           Max.   :1.0000
##  rate_negative_words avg_positive_polarity min_positive_polarity
##  Min.   :0.0000      Min.   :0.0000        Min.   :0.00000
##  1st Qu.:0.1852      1st Qu.:0.3062        1st Qu.:0.05000
##  Median :0.2800      Median :0.3588        Median :0.10000
##  Mean   :0.2879      Mean   :0.3538        Mean   :0.09545
##  3rd Qu.:0.3846      3rd Qu.:0.4114        3rd Qu.:0.10000
##  Max.   :1.0000      Max.   :1.0000        Max.   :1.00000
##  max_positive_polarity avg_negative_polarity min_negative_polarity
##  Min.   :0.0000        Min.   :-1.0000       Min.   :-1.0000
##  1st Qu.:0.6000        1st Qu.:-0.3284       1st Qu.:-0.7000
##  Median :0.8000        Median :-0.2533       Median :-0.5000
##  Mean   :0.7567        Mean   :-0.2595       Mean   :-0.5219
##  3rd Qu.:1.0000        3rd Qu.:-0.1869       3rd Qu.:-0.3000
##  Max.   :1.0000        Max.   : 0.0000       Max.   : 0.0000
##  max_negative_polarity title_subjectivity title_sentiment_polarity
##  Min.   :-1.0000       Min.   :0.0000     Min.   :-1.00000
##  1st Qu.:-0.1250       1st Qu.:0.0000     1st Qu.: 0.00000
##  Median :-0.1000       Median :0.1500     Median : 0.00000
##  Mean   :-0.1075       Mean   :0.2824     Mean   : 0.07143
##  3rd Qu.:-0.0500       3rd Qu.:0.5000     3rd Qu.: 0.15000
##  Max.   : 0.0000       Max.   :1.0000     Max.   : 1.00000
##  abs_title_subjectivity abs_title_sentiment_polarity     shares
##  Min.   :0.0000         Min.   :0.0000               Min.   :     1
##  1st Qu.:0.1667         1st Qu.:0.0000               1st Qu.:   946
##  Median :0.5000         Median :0.0000               Median :  1400
##  Mean   :0.3418         Mean   :0.1561               Mean   :  3395
##  3rd Qu.:0.5000         3rd Qu.:0.2500               3rd Qu.:  2800
##  Max.   :0.5000         Max.   :1.0000               Max.   :843300
```

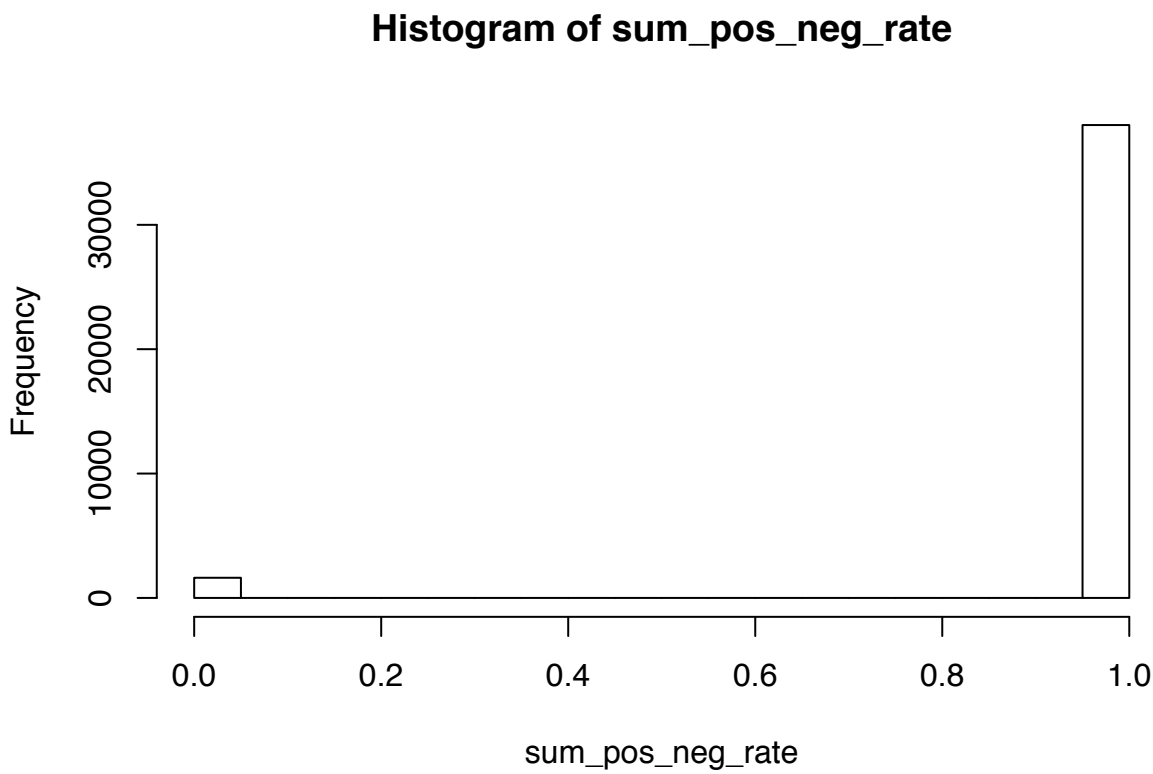
```r
## Missing Values
any(is.na(news.data))
```

```
## [1] FALSE
```

```r
## Check for missing values encoded as 0
sum_pos_neg_rate <- trunc(news.data$rate_positive_words + news.data$rate_negative_words,
              digits = 0)
unique(sum_pos_neg_rate)
```

```
## [1] 1 0
```

```r
hist(sum_pos_neg_rate)
```




```r
any(0 %in% news.data$average_token_length)
```

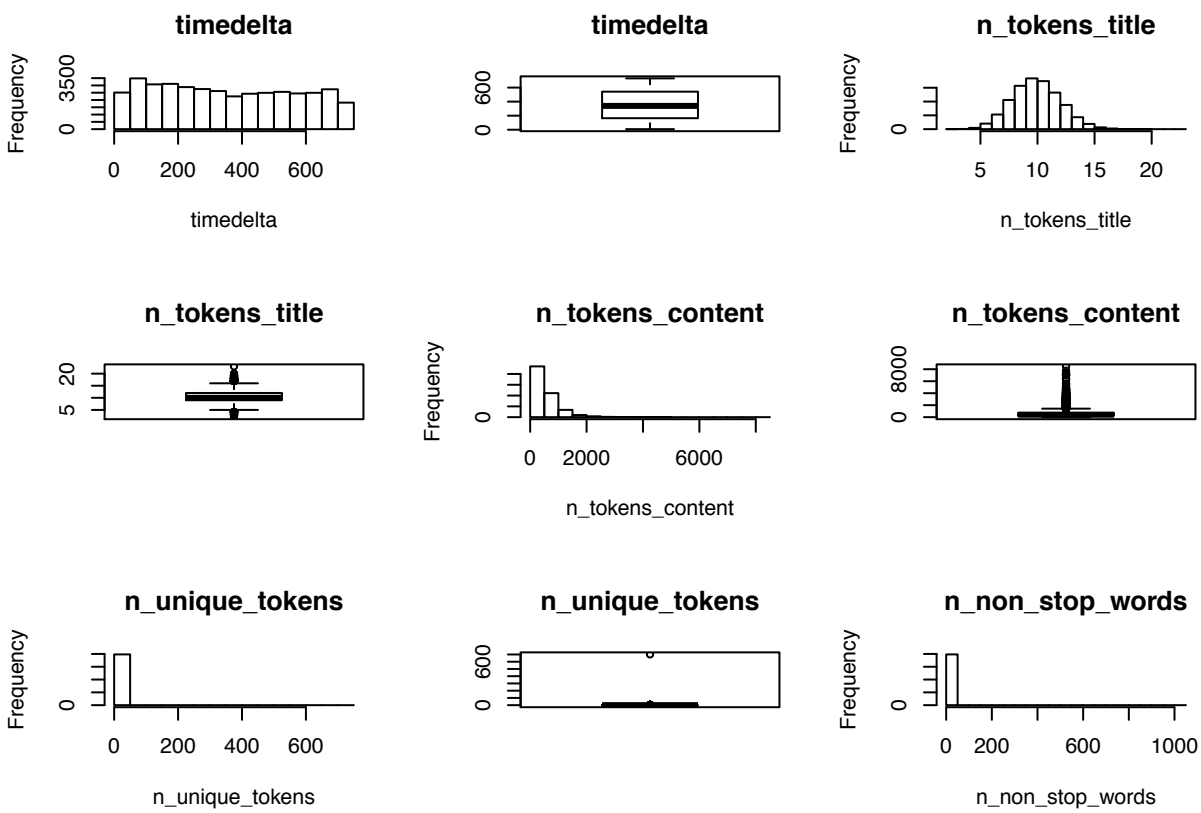
```
## [1] TRUE
```

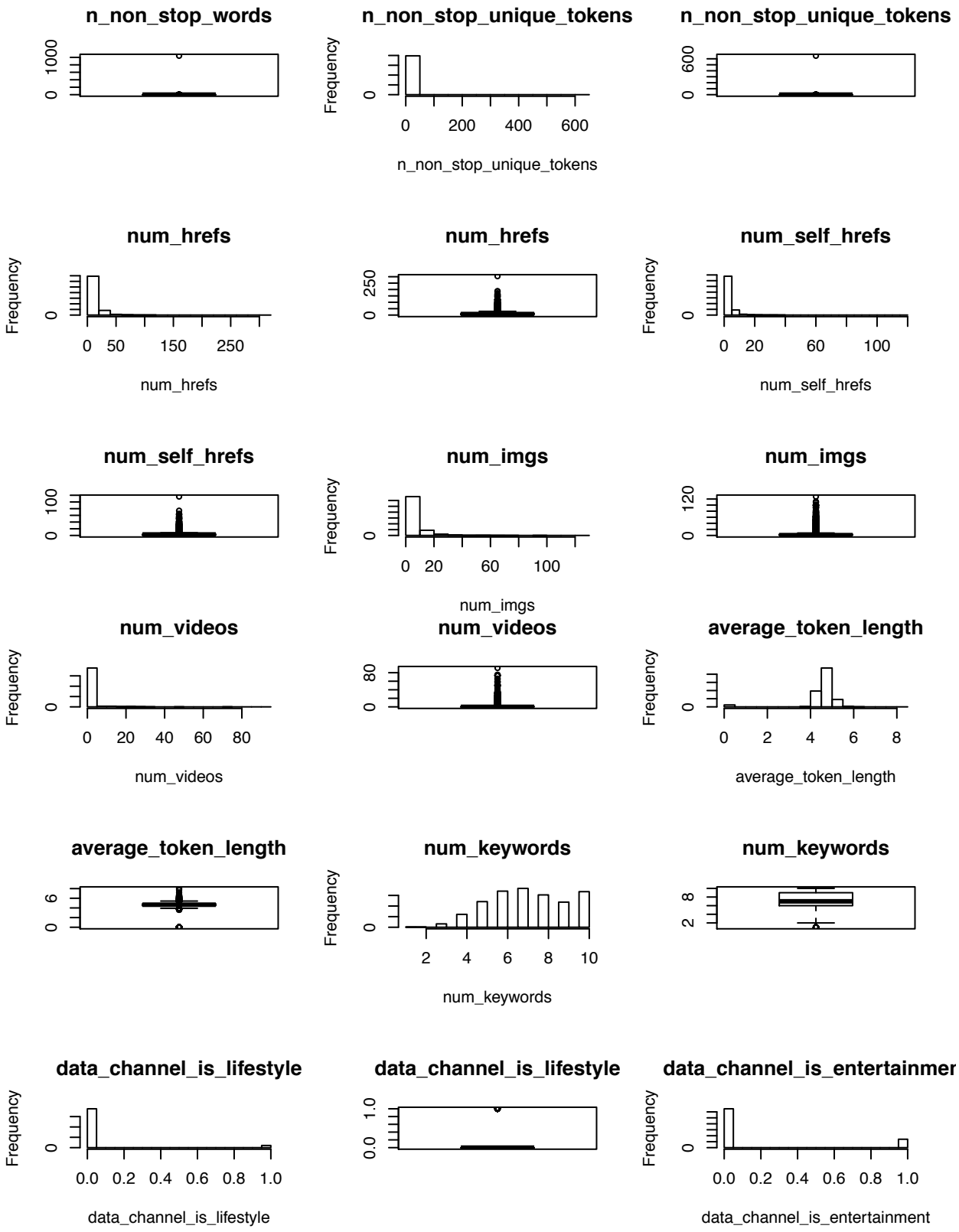
```r
## Two Variable Stats - Correlation
#pairs(news.data)
#round(cor(news.data), digits = 2)
#library(corrplot)
#corrplot(cor(x = news.data[, -60], y = news.data$shares), order="hclust")
#library(corrgram)
#corrgram(news.data)
sort(cor(news.data, y = news.data$shares)[, 1])
```

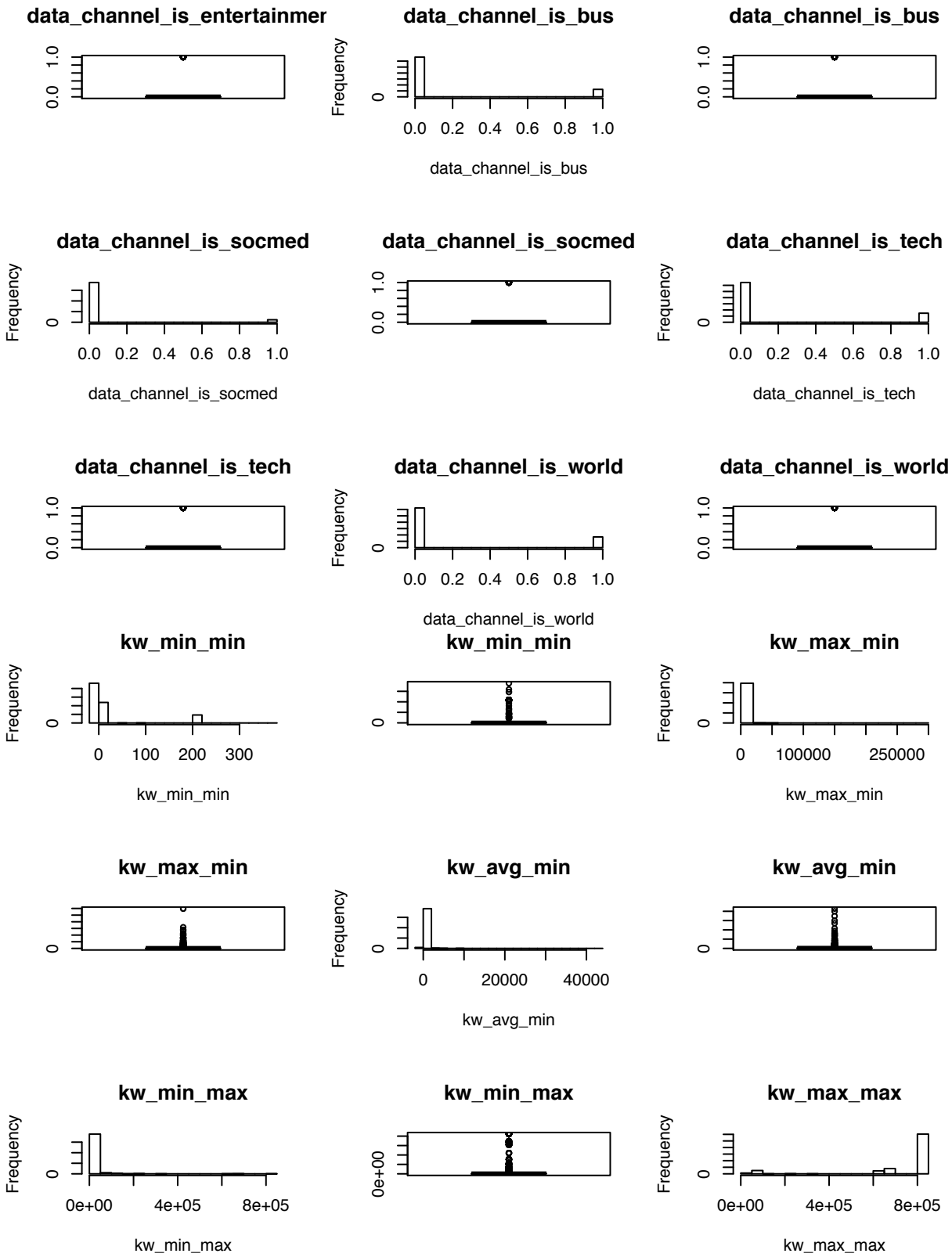
```
##                       LDA_02         data_channel_is_world
##                -5.916273e-02                 -4.949731e-02
##         avg_negative_polarity          average_token_length
##                -3.202891e-02                 -2.200730e-02
##         max_negative_polarity         min_negative_polarity
##                -1.930021e-02                 -1.929738e-02
## data_channel_is_entertainment                       LDA_04
##                -1.700620e-02                 -1.662194e-02
##          data_channel_is_tech           rate_positive_words
##                -1.325287e-02                 -1.324130e-02
##           data_channel_is_bus                       LDA_01
##                -1.237617e-02                 -1.018287e-02
##           weekday_is_thursday            weekday_is_tuesday
##                -8.833244e-03                 -7.940652e-03
##           rate_negative_words             weekday_is_friday
##                -5.183138e-03                 -3.884350e-03
##          weekday_is_wednesday                       LDA_00
##                -3.800672e-03                 -3.793063e-03
##                num_self_hrefs                   kw_min_min
##                -1.900403e-03                 -1.050988e-03
##         min_positive_polarity      n_non_stop_unique_tokens
##                -4.009526e-05                  1.141719e-04
##              n_non_stop_words    global_rate_positive_words
##                 4.429416e-04                  5.432341e-04
##               n_unique_tokens        abs_title_subjectivity
##                 8.063457e-04                  1.480998e-03
##              n_tokens_content                   kw_min_max
##                 2.458984e-03                  3.901432e-03
##     global_sentiment_polarity        data_channel_is_socmed
##                 4.162929e-03                  5.021216e-03
##     data_channel_is_lifestyle    global_rate_negative_words
##                 5.831267e-03                  6.615173e-03
##                   kw_max_max             weekday_is_sunday
##                 7.862569e-03                  8.229539e-03
##                    timedelta                n_tokens_title
##                 8.662288e-03                  8.783119e-03
##             weekday_is_monday         max_positive_polarity
##                 9.726435e-03                  1.006779e-02
##         avg_positive_polarity      title_sentiment_polarity
##                 1.214220e-02                  1.277187e-02
##           weekday_is_saturday                   is_weekend
##                 1.508225e-02                  1.695819e-02
##                 num_keywords            title_subjectivity
##                 2.181823e-02                  2.196668e-02
##                   num_videos  abs_title_sentiment_polarity
##                 2.393607e-02                  2.713523e-02
##                   kw_max_min                   kw_avg_min
##                 3.011394e-02                  3.040555e-02
##           global_subjectivity                      num_imgs
##                 3.160406e-02                  3.938760e-02
##                   kw_min_avg                   kw_avg_max
##                 3.955069e-02                  4.468584e-02
##                    num_hrefs     self_reference_max_shares
##                 4.540401e-02                  4.711522e-02
```

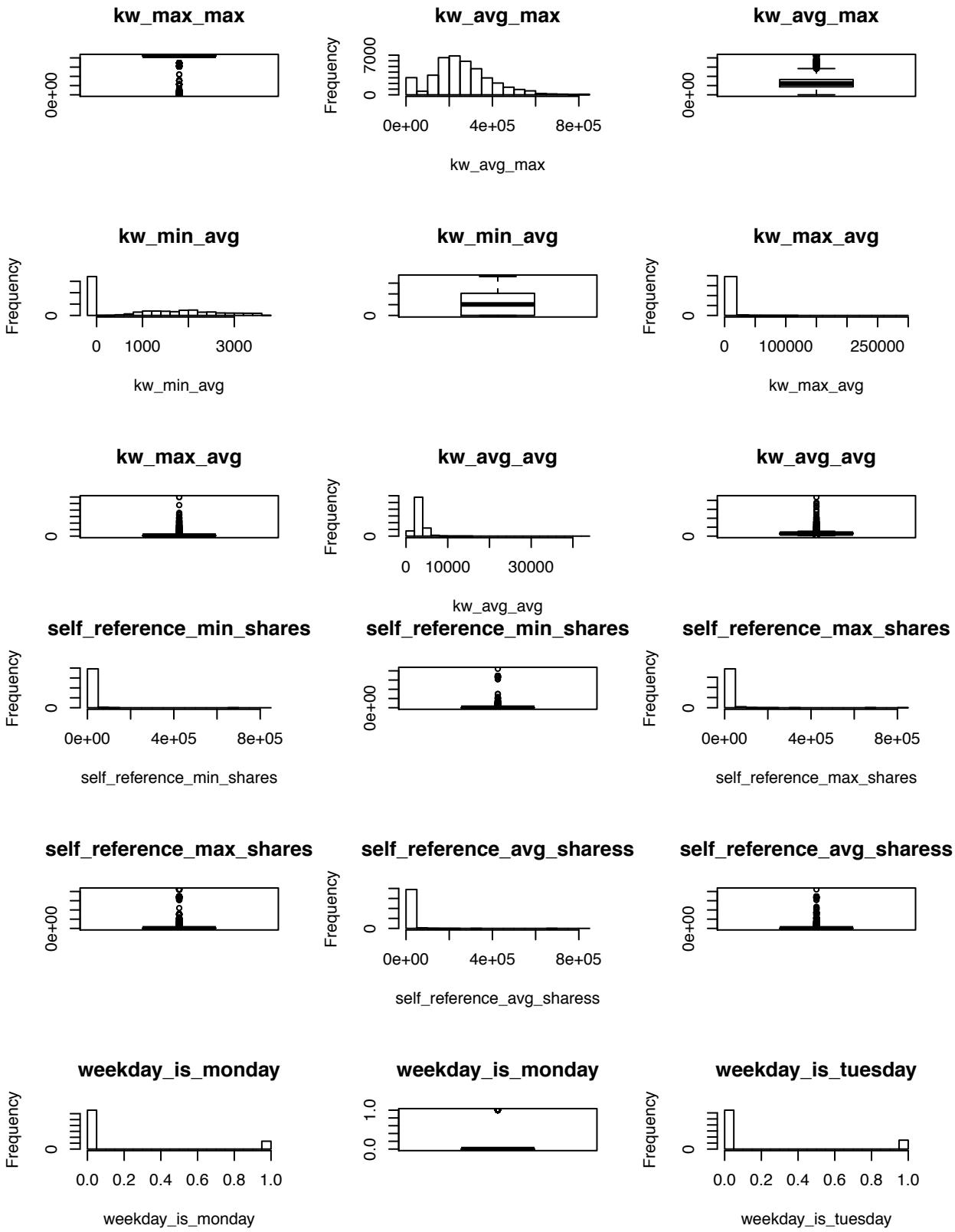
```
##     self_reference_min_shares    self_reference_avg_sharess
##                  5.595758e-02                  5.778890e-02
##                    kw_max_avg                       LDA_03
##                  6.430586e-02                  8.377149e-02
##                    kw_avg_avg                       shares
##                  1.104129e-01                  1.000000e+00
```

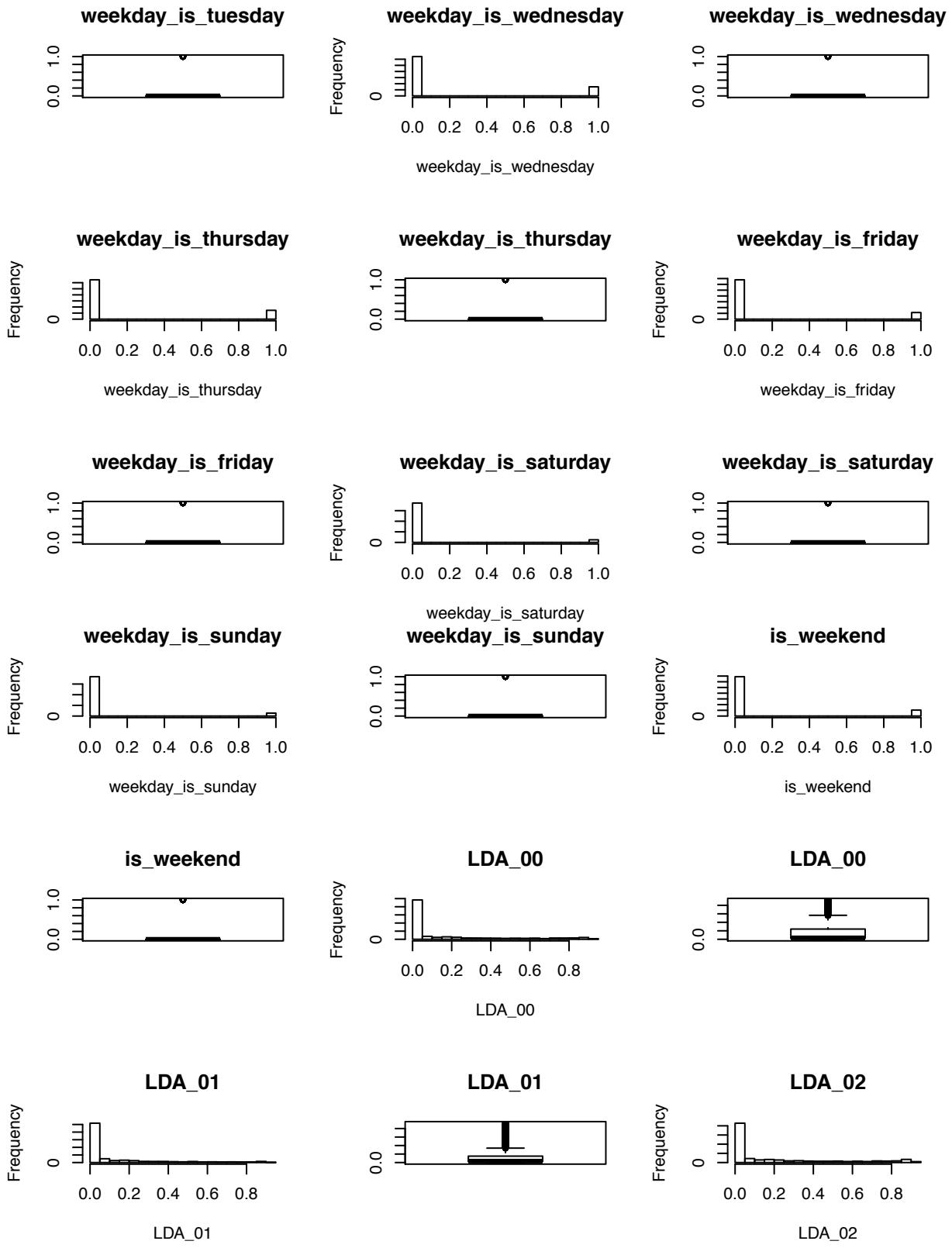
```
## Outliers
#par(mfrow=c(1,2))
#hist(news.data[, 4], main=names(news.data[4]), xlab=names(news.data[4]))
#boxplot(news.data[, 4], main=names(news.data[4]), type="l")
par(mfrow=c(3,3))
for (i in 1:length(news.data)) {
  hist(news.data[, i], main=names(news.data[i]), xlab=names(news.data[i]))
  boxplot(news.data[,i], main=names(news.data[i]), type="l")
}
```

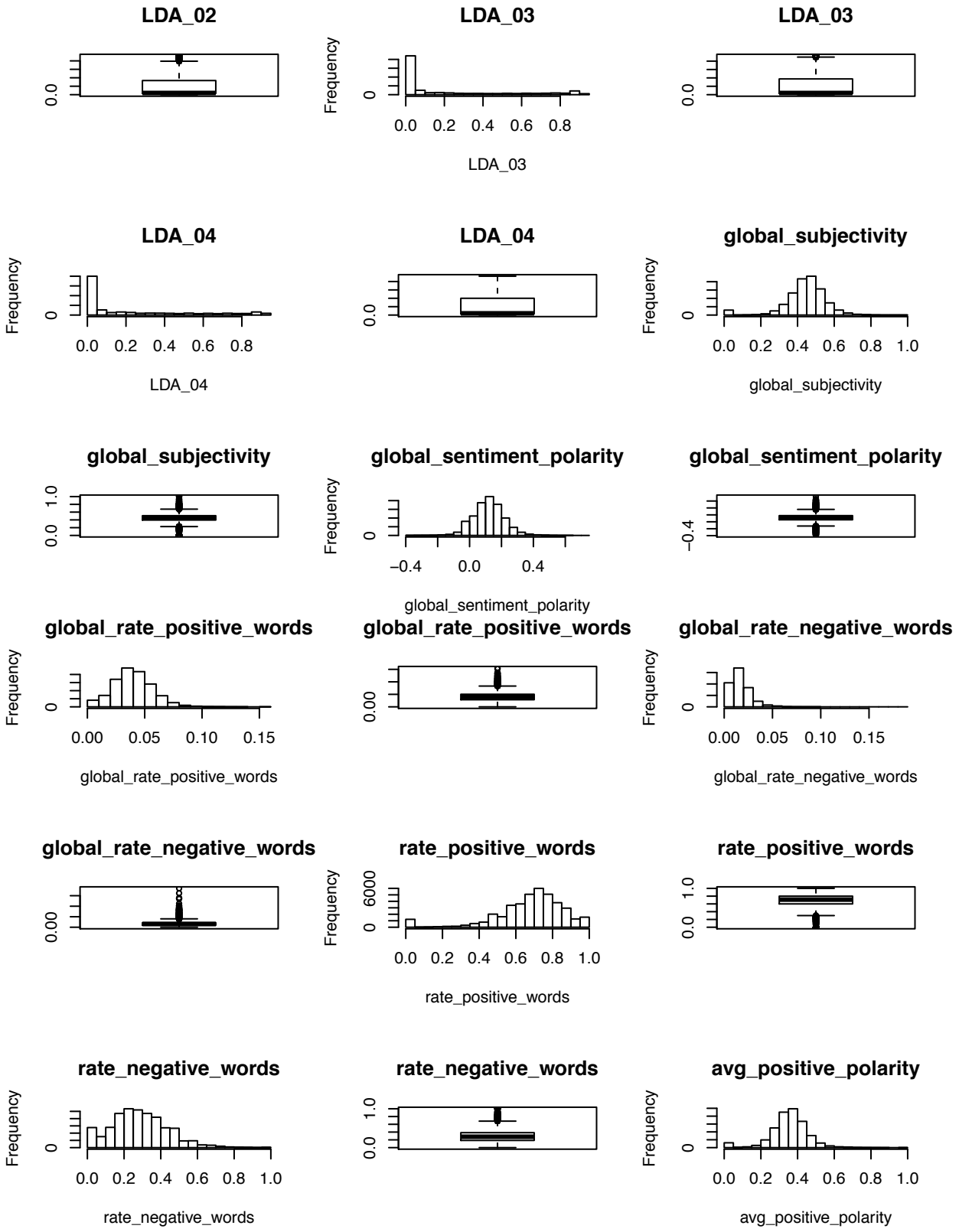


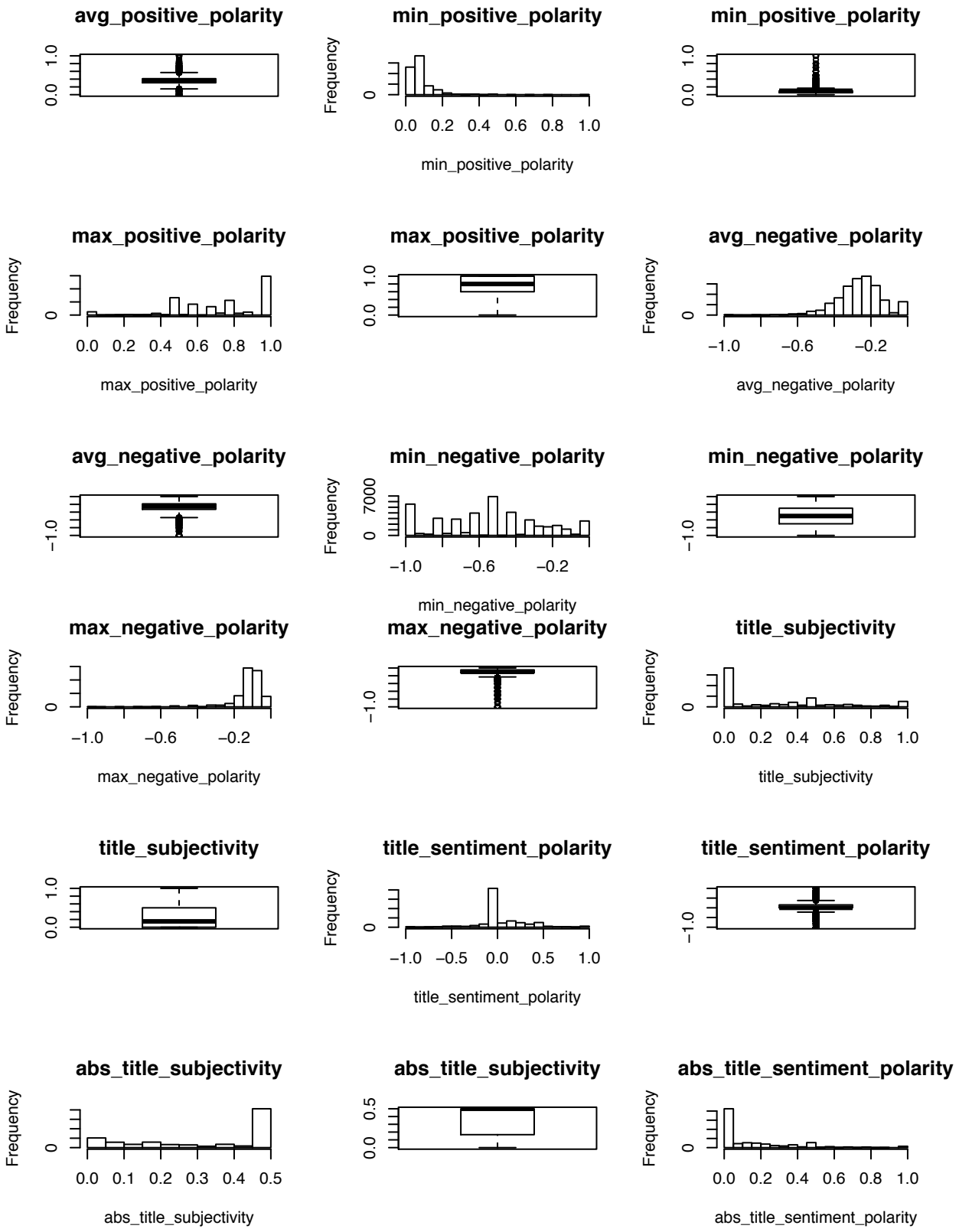

n_non_stop_words
n_non_stop_unique_tokens
n_non_stop_unique_tokens
Frequency
n_non_stop_unique_tokens
num_hrefs
num_hrefs
num_self_hrefs
Frequency
num_hrefs
num_self_hrefs
num_self_hrefs
num_imgs
num_imgs
Frequency
num_imgs
num_videos
num_videos
average_token_length
Frequency
num_videos
average_token_length
average_token_length
num_keywords
num_keywords
Frequency
num_keywords
data_channel_is_lifestyle
data_channel_is_lifestyle
data_channel_is_entertainmer
Frequency
data_channel_is_lifestyle
data_channel_is_entertainment


data_channel_is_entertainmer
data_channel_is_bus
Frequency
0.0 0.2 0.4 0.6 0.8 1.0
data_channel_is_bus
data_channel_is_bus
data_channel_is_socmed
Frequency
0.0 0.2 0.4 0.6 0.8 1.0
data_channel_is_socmed
data_channel_is_socmed
data_channel_is_tech
Frequency
0.0 0.2 0.4 0.6 0.8 1.0
data_channel_is_tech
data_channel_is_tech
data_channel_is_world
Frequency
0.0 0.2 0.4 0.6 0.8 1.0
data_channel_is_world
data_channel_is_world
kw_min_min
Frequency
0 100 200 300
kw_min_min
kw_min_min
kw_max_min
Frequency
0 100000 250000
kw_max_min
kw_max_min
kw_avg_min
Frequency
0 20000 40000
kw_avg_min
kw_avg_min
kw_min_max
Frequency
0e+00 4e+05 8e+05
kw_min_max
kw_min_max
kw_max_max
Frequency
0e+00 4e+05 8e+05
kw_max_max


kw_max_max
0e+00
kw_avg_max
Frequency
0
7000
0e+00
4e+05
8e+05
kw_avg_max
kw_avg_max
0e+00
kw_min_avg
Frequency
0
0
1000
3000
kw_min_avg
kw_min_avg
0
kw_max_avg
Frequency
0
0
100000
250000
kw_max_avg
kw_max_avg
0
kw_avg_avg
Frequency
0
0
10000
30000
kw_avg_avg
kw_avg_avg
0
self_reference_min_shares
Frequency
0
0e+00
4e+05
8e+05
self_reference_min_shares
self_reference_min_shares
0e+00
self_reference_max_shares
Frequency
0
0e+00
4e+05
8e+05
self_reference_max_shares
self_reference_max_shares
0e+00
self_reference_avg_sharess
Frequency
0
0e+00
4e+05
8e+05
self_reference_avg_sharess
self_reference_avg_sharess
0e+00
weekday_is_monday
Frequency
0
0.0
0.2
0.4
0.6
0.8
1.0
weekday_is_monday
weekday_is_monday
0.0
1.0
weekday_is_tuesday
Frequency
0
0.0
0.2
0.4
0.6
0.8
1.0
weekday_is_tuesday


weekday_is_tuesday
weekday_is_wednesday
weekday_is_wednesday
weekday_is_thursday
weekday_is_thursday
weekday_is_friday
weekday_is_friday
weekday_is_saturday
weekday_is_saturday
weekday_is_sunday
weekday_is_sunday
is_weekend
is_weekend
LDA_00
LDA_00
LDA_01
LDA_01
LDA_02
Frequency
0.0 0.2 0.4 0.6 0.8 1.0
0.0 0.2 0.4 0.6 0.8
0 1.0

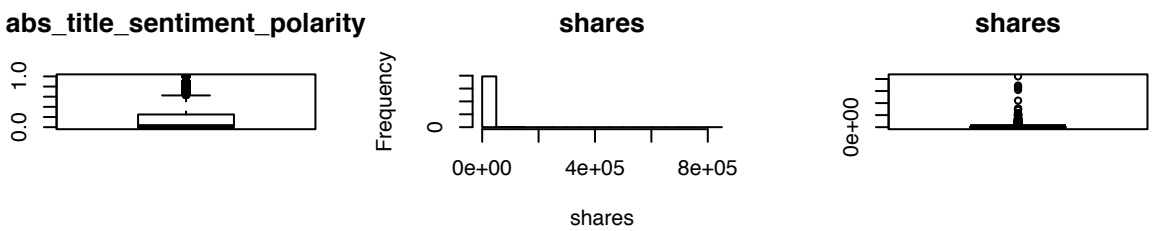

LDA_02
LDA_03
LDA_03
LDA_03
LDA_04
LDA_04
global_subjectivity
global_subjectivity
global_subjectivity
global_sentiment_polarity
global_sentiment_polarity
global_rate_positive_words
global_rate_positive_words
global_rate_negative_words
global_rate_negative_words
global_rate_negative_words
rate_positive_words
rate_positive_words
rate_negative_words
rate_negative_words
avg_positive_polarity
Frequency




```
## Outlier in number of unique tokens
summary(news.data$n_unique_tokens)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##   0.0000   0.4709   0.5392   0.5482   0.6087 701.0000
```

```
# Make a copy of news data with outliers and missing values removed
news.data.copy <- news.data
```

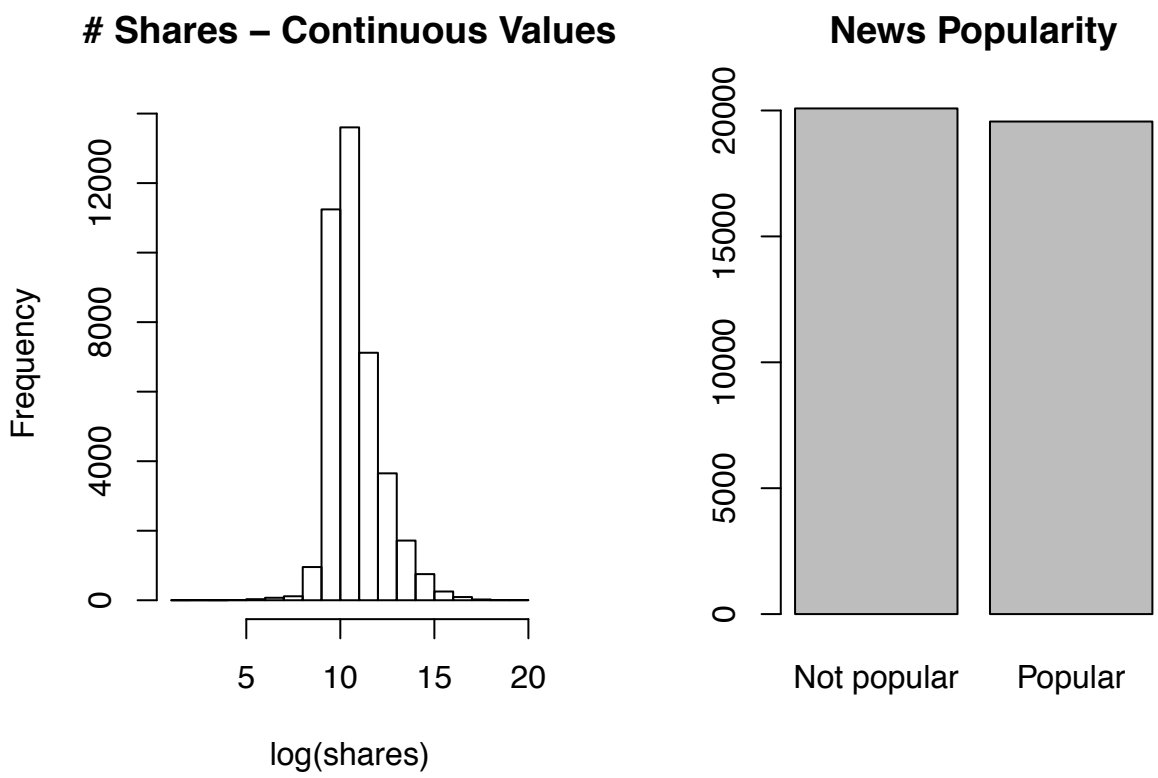


2.4 Data Distribution

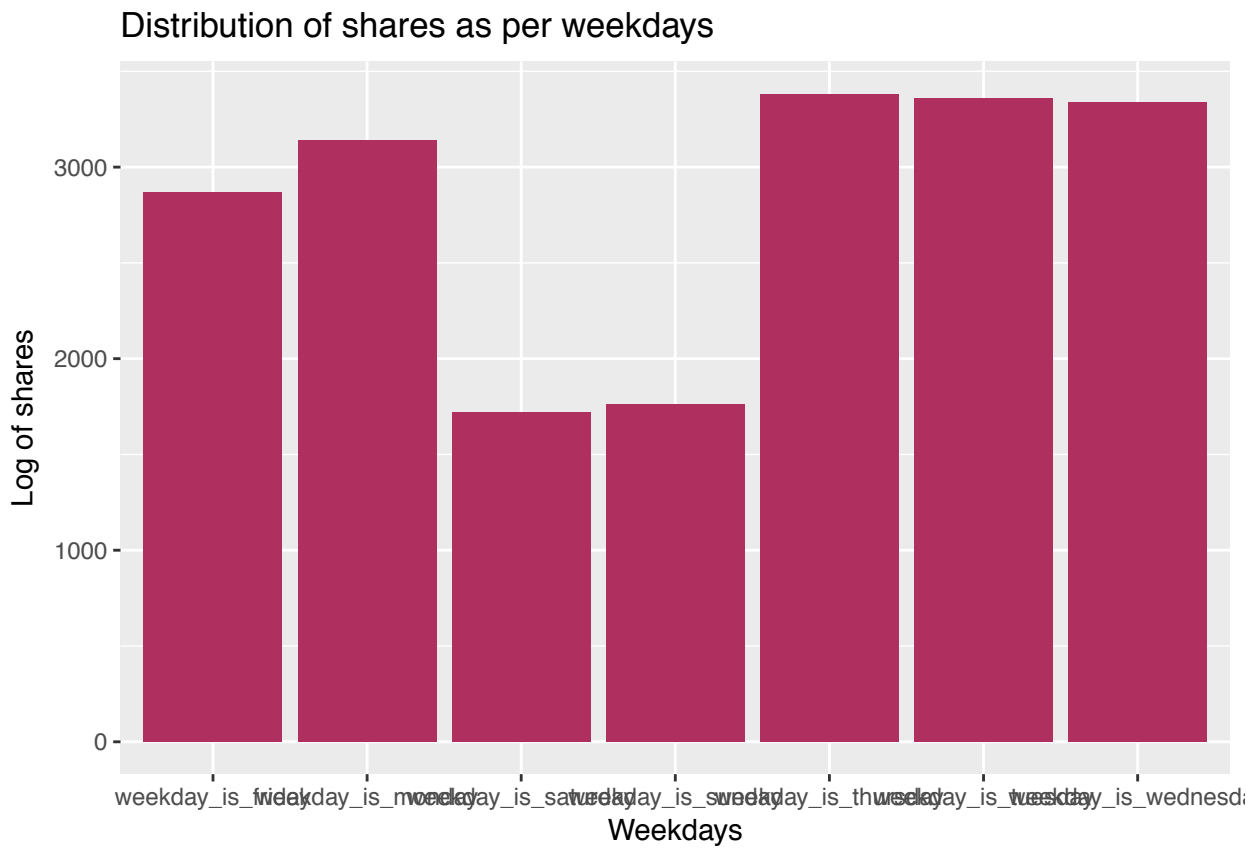
```
#######################################################################
# Data Distribution
#######################################################################
#- hist(news.data$shares, main="Shares", xlab = "Shares")
par(mfrow=c(1,2))
hist(log2(news.data[, 60] + 1), main="# Shares - Continuous Values",
     xlab="log(shares)")
## Convert continuous response variable to binary (Using 3rd Qu.)
summary(news.data$shares)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1     946    1400    3395    2800  843300
```

```
news.data$shares <- ifelse(news.data$shares > 1400, 1, 0)
barplot(table(news.data$shares), main = "News Popularity",
   names.arg = c("Not popular", "Popular"))
```




```
news.data.cat <- cbind(news.data[, c(1:30)],
                       weekday = apply(news.data[, c(31:37)], 1,
                                       function(x) names(x)[as.logical(as.numeric(
                                         as.character(x)))]),
                       news.data[, c(39:60)])
p1 <- ggplot(data=news.data.cat, aes(as.factor(as.character(weekday)),
                                     log2(shares + 1)))
p1 + geom_bar(stat="identity", fill="maroon") + xlab("Weekdays") + ylab("Log of shares") +
  ggtitle("Distribution of shares as per weekdays")
```



2.5 Data Sampling

As we have sufficient data, we split the data into three sets in the ratio of 50:25:25

```
#######################################################################
# Split the Dataset Into Training, Test and Validation
#######################################################################
sample.index <- sample(1:3, size=nrow(news.data), replace=TRUE,
                       prob=c(0.5,0.25,0.25))
news.train <- news.data[sample.index == 1, ]
news.validation <- news.data[sample.index == 2, ]
news.test <- news.data[sample.index == 3, ]

dim(news.train)
```

```
## [1] 19805    60
```

```
dim(news.test)
```

```
## [1] 9931   60
```

```
dim(news.validation)
```

```
## [1] 9908   60
```

2.6 Initial Regression Analysis

Here, we performed logistic regression on raw data without removing any predictor or outlier to see the impact of data cleanup.

```
#######################################################################
# Logistic Regression - All predictors
#######################################################################
## Model fitting
fit.lm <- glm(shares ~ ., data = news.train, family=binomial)
fit.lm
```

```
##
## Call:  glm(formula = shares ~ ., family = binomial, data = news.train)
##
## Coefficients:
##                   (Intercept)                      timedelta
##                     1.174e+04                     -5.434e-05
##                n_tokens_title                n_tokens_content
##                     6.756e-03                      4.214e-05
##               n_unique_tokens                n_non_stop_words
##                     3.971e-01                     -1.085e+01
##      n_non_stop_unique_tokens                      num_hrefs
##                    -1.079e+00                      7.016e-03
##                num_self_hrefs                       num_imgs
##                    -2.220e-02                      1.468e-03
##                    num_videos            average_token_length
##                     3.100e-03                     -1.142e-01
##                  num_keywords       data_channel_is_lifestyle
##                     4.287e-02                     -1.084e-01
## data_channel_is_entertainment             data_channel_is_bus
##                    -1.642e-01                     -1.115e-01
##        data_channel_is_socmed            data_channel_is_tech
##                     8.355e-01                      5.247e-01
##         data_channel_is_world                      kw_min_min
##                     7.566e-02                      1.482e-03
##                    kw_max_min                      kw_avg_min
##                    -1.168e-05                      1.159e-04
##                    kw_min_max                      kw_max_max
##                    -6.137e-07                     -3.890e-07
##                    kw_avg_max                      kw_min_avg
##                    -5.989e-07                     -9.275e-05
##                    kw_max_avg                      kw_avg_avg
##                    -9.759e-05                      7.354e-04
##     self_reference_min_shares       self_reference_max_shares
##                     7.631e-06                      2.448e-06
##    self_reference_avg_sharess               weekday_is_monday
##                    -1.435e-06                     -6.953e-01
##            weekday_is_tuesday            weekday_is_wednesday
##                    -8.045e-01                     -8.240e-01
##           weekday_is_thursday               weekday_is_friday
##                    -7.726e-01                     -5.379e-01
##           weekday_is_saturday               weekday_is_sunday
##                     2.816e-01                             NA
```

```
##                    is_weekend                        LDA_00
##                            NA                    -1.174e+04
##                        LDA_01                        LDA_02
##                    -1.174e+04                    -1.174e+04
##                        LDA_03                        LDA_04
##                    -1.174e+04                    -1.174e+04
##           global_subjectivity      global_sentiment_polarity
##                     1.109e+00                    -5.925e-01
##    global_rate_positive_words     global_rate_negative_words
##                    -1.310e+00                     1.145e+00
##           rate_positive_words            rate_negative_words
##                     1.156e+01                     1.107e+01
##         avg_positive_polarity          min_positive_polarity
##                    -1.855e-01                    -5.785e-01
##         max_positive_polarity          avg_negative_polarity
##                     1.227e-01                     8.247e-02
##         min_negative_polarity          max_negative_polarity
##                    -4.518e-02                     2.727e-01
##            title_subjectivity       title_sentiment_polarity
##                     1.152e-01                     1.615e-01
##        abs_title_subjectivity   abs_title_sentiment_polarity
##                     3.067e-01                     9.444e-02
##
## Degrees of Freedom: 19804 Total (i.e. Null);  19747 Residual
## Null Deviance:       27460
## Residual Deviance: 24880     AIC: 24990
```

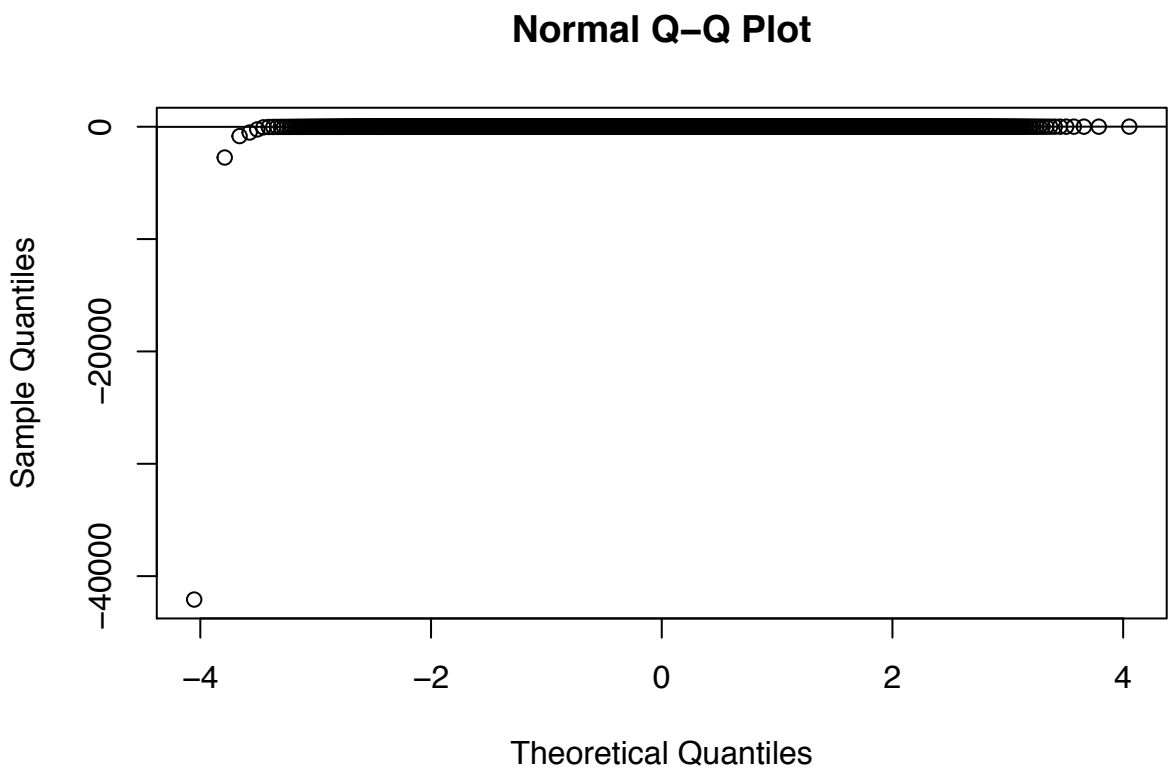
```r
summary(fit.lm)
```

```
##
## Call:
## glm(formula = shares ~ ., family = binomial, data = news.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.6146  -1.0371  -0.5741   1.0670   2.1213
##
## Coefficients: (2 not defined because of singularities)
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    1.174e+04  1.410e+05   0.083 0.933637
## timedelta                     -5.434e-05  1.044e-04  -0.521 0.602711
## n_tokens_title                 6.756e-03  7.669e-03   0.881 0.378393
## n_tokens_content               4.214e-05  6.037e-05   0.698 0.485192
## n_unique_tokens                3.971e-01  5.126e-01   0.775 0.438579
## n_non_stop_words              -1.085e+01  1.353e+02  -0.080 0.936087
## n_non_stop_unique_tokens      -1.079e+00  4.327e-01  -2.493 0.012668 *
## num_hrefs                      7.016e-03  1.841e-03   3.812 0.000138 ***
## num_self_hrefs                -2.220e-02  4.701e-03  -4.722 2.33e-06 ***
## num_imgs                       1.468e-03  2.394e-03   0.613 0.539758
## num_videos                     3.100e-03  4.276e-03   0.725 0.468386
## average_token_length          -1.142e-01  6.410e-02  -1.781 0.074888 .
## num_keywords                   4.287e-02  9.834e-03   4.360 1.30e-05 ***
## data_channel_is_lifestyle     -1.084e-01  1.040e-01  -1.042 0.297363
## data_channel_is_entertainment -1.642e-01  6.722e-02  -2.443 0.014549 *
```

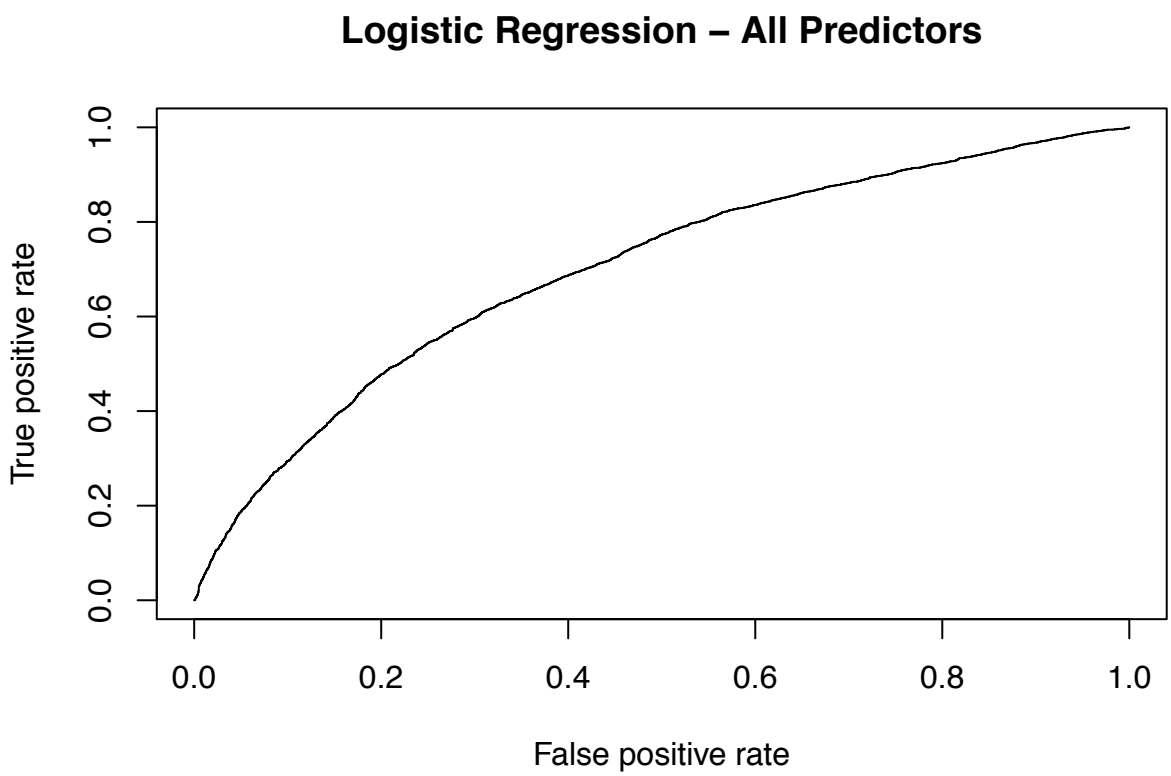
```
## data_channel_is_bus           -1.115e-01  1.012e-01  -1.102 0.270664
## data_channel_is_socmed         8.355e-01  1.010e-01   8.273  < 2e-16 ***
## data_channel_is_tech           5.247e-01  9.739e-02   5.388 7.14e-08 ***
## data_channel_is_world          7.566e-02  9.918e-02   0.763 0.445520
## kw_min_min                     1.482e-03  4.330e-04   3.424 0.000617 ***
## kw_max_min                    -1.168e-05  2.167e-05  -0.539 0.589925
## kw_avg_min                     1.159e-04  1.419e-04   0.817 0.413934
## kw_min_max                    -6.137e-07  3.106e-07  -1.976 0.048167 *
## kw_max_max                    -3.890e-07  1.564e-07  -2.487 0.012878 *
## kw_avg_max                    -5.989e-07  2.268e-07  -2.640 0.008285 **
## kw_min_avg                    -9.275e-05  2.056e-05  -4.510 6.47e-06 ***
## kw_max_avg                    -9.759e-05  7.453e-06 -13.094  < 2e-16 ***
## kw_avg_avg                     7.354e-04  4.154e-05  17.702  < 2e-16 ***
## self_reference_min_shares      7.631e-06  2.754e-06   2.771 0.005593 **
## self_reference_max_shares      2.448e-06  1.444e-06   1.695 0.090043 .
## self_reference_avg_sharess    -1.435e-06  3.407e-06  -0.421 0.673565
## weekday_is_monday             -6.953e-01  7.053e-02  -9.859  < 2e-16 ***
## weekday_is_tuesday            -8.045e-01  6.962e-02 -11.556  < 2e-16 ***
## weekday_is_wednesday          -8.240e-01  6.973e-02 -11.817  < 2e-16 ***
## weekday_is_thursday           -7.726e-01  6.974e-02 -11.078  < 2e-16 ***
## weekday_is_friday             -5.379e-01  7.180e-02  -7.491 6.83e-14 ***
## weekday_is_saturday            2.816e-01  8.914e-02   3.159 0.001582 **
## weekday_is_sunday                     NA         NA      NA       NA
## is_weekend                            NA         NA      NA       NA
## LDA_00                        -1.174e+04  1.410e+05  -0.083 0.933634
## LDA_01                        -1.174e+04  1.410e+05  -0.083 0.933627
## LDA_02                        -1.174e+04  1.410e+05  -0.083 0.933628
## LDA_03                        -1.174e+04  1.410e+05  -0.083 0.933629
## LDA_04                        -1.174e+04  1.410e+05  -0.083 0.933632
## global_subjectivity            1.109e+00  2.258e-01   4.914 8.91e-07 ***
## global_sentiment_polarity     -5.925e-01  4.447e-01  -1.332 0.182734
## global_rate_positive_words    -1.310e+00  1.878e+00  -0.698 0.485342
## global_rate_negative_words     1.145e+00  3.620e+00   0.316 0.751666
## rate_positive_words            1.156e+01  1.353e+02   0.085 0.931945
## rate_negative_words            1.107e+01  1.353e+02   0.082 0.934795
## avg_positive_polarity         -1.855e-01  3.627e-01  -0.512 0.608953
## min_positive_polarity         -5.785e-01  3.041e-01  -1.902 0.057111 .
## max_positive_polarity          1.227e-01  1.133e-01   1.083 0.278993
## avg_negative_polarity          8.247e-02  3.340e-01   0.247 0.805005
## min_negative_polarity         -4.518e-02  1.215e-01  -0.372 0.710085
## max_negative_polarity          2.727e-01  2.779e-01   0.981 0.326351
## title_subjectivity             1.152e-01  7.256e-02   1.588 0.112355
## title_sentiment_polarity       1.615e-01  6.684e-02   2.417 0.015666 *
## abs_title_subjectivity         3.067e-01  9.582e-02   3.201 0.001370 **
## abs_title_sentiment_polarity   9.444e-02  1.051e-01   0.899 0.368912
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 27455  on 19804  degrees of freedom
## Residual deviance: 24878  on 19747  degrees of freedom
## AIC: 24994
##
```

```
## Number of Fisher Scoring iterations: 10
```

```r
## Q-Q plots
qqnorm(fit.lm$residuals)
qqline(fit.lm$residuals)
```




```r
## ROC Curves
scores <- predict(fit.lm, newdata= news.validation, type = "response")
prob <- prediction(scores, labels=news.validation$shares)
perf <- performance(prob, "tpr", "fpr")
plot(perf, colorize=F, main="Logistic Regression - All Predictors")
```




```r
## AUC
unlist(attributes(performance(prob, "auc"))$y.values)
```

```
## [1] 0.6975675
```

```r
## Confusion Matrix
pred.fit.lm <- ifelse(scores>0.5, 1, 0)
confusionMatrix(pred.fit.lm, news.validation$shares, positive = "1")
```

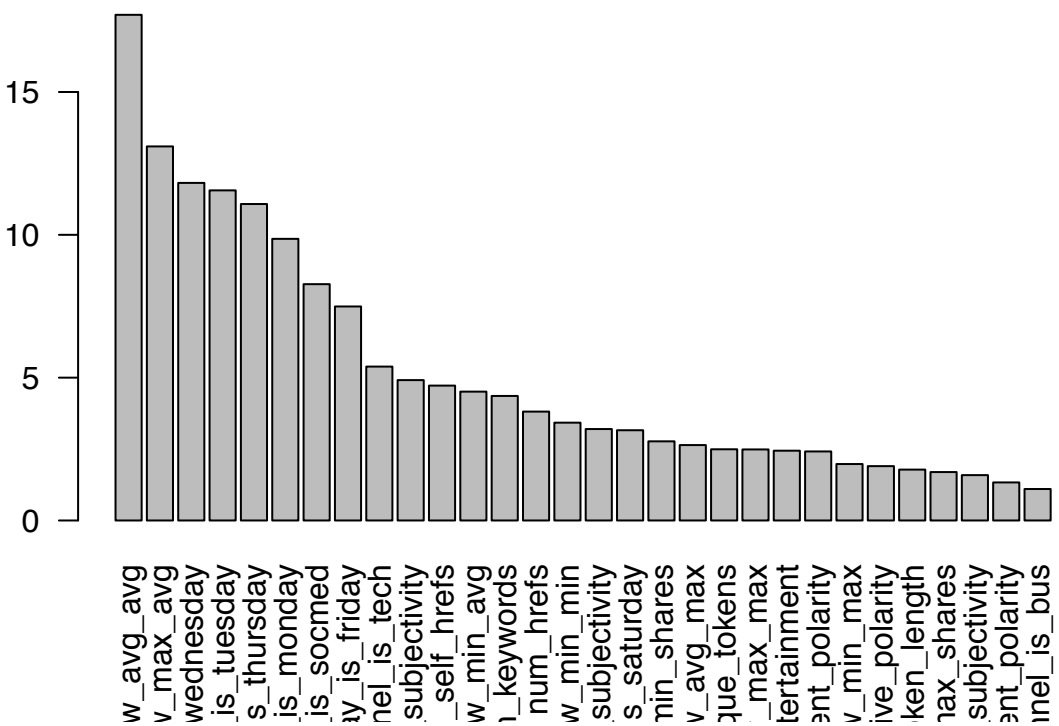
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 3384 1801
##          1 1672 3051
##
##                Accuracy : 0.6495
##                  95% CI : (0.64, 0.6589)
##     No Information Rate : 0.5103
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.2983
##  Mcnemar's Test P-Value : 0.02986
##
##             Sensitivity : 0.6288
##             Specificity : 0.6693
##          Pos Pred Value : 0.6460
##          Neg Pred Value : 0.6527
```

```
##              Prevalence : 0.4897
##          Detection Rate : 0.3079
##    Detection Prevalence : 0.4767
##       Balanced Accuracy : 0.6491
##
##        'Positive' Class : 1
##
```

```
## Prediction Error on Validation Set
mse.fit.lm = mean((news.validation$shares-pred.fit.lm)^2)
mse.fit.lm
```

```
## [1] 0.3505248
```

```
## Variable Importance
fit.lm.imp <- varImp(fit.lm)
fit.lm.imp$Names <- row.names(fit.lm.imp)
imp.values <- fit.lm.imp[order(fit.lm.imp$Overall, decreasing=TRUE)[1:30],1]
imp.names <- fit.lm.imp[order(fit.lm.imp$Overall, decreasing=TRUE)[1:30],2]
barplot(imp.values, names = imp.names, las = 2)
```




```
#######################################################################
# Logistic Regression - Using Weekdays Only
#######################################################################
## Data Prep
news.train.weekdays <- news.train[, -c(37, 38)]
news.validation.weekdays <- news.validation[, -c(37, 38)]

## Model fitting
fit.lm.weekdays <- glm(shares ~ ., data = news.train.weekdays,
```

```r
                        family=binomial)
fit.lm.weekdays
```

```
##
## Call:  glm(formula = shares ~ ., family = binomial, data = news.train.weekdays)
##
## Coefficients:
##                   (Intercept)                      timedelta
##                     1.174e+04                     -5.434e-05
##                n_tokens_title                n_tokens_content
##                     6.756e-03                      4.214e-05
##               n_unique_tokens                n_non_stop_words
##                     3.971e-01                     -1.085e+01
##      n_non_stop_unique_tokens                      num_hrefs
##                    -1.079e+00                      7.016e-03
##                num_self_hrefs                       num_imgs
##                    -2.220e-02                      1.468e-03
##                    num_videos            average_token_length
##                     3.100e-03                     -1.142e-01
##                  num_keywords       data_channel_is_lifestyle
##                     4.287e-02                     -1.084e-01
## data_channel_is_entertainment             data_channel_is_bus
##                    -1.642e-01                     -1.115e-01
##        data_channel_is_socmed            data_channel_is_tech
##                     8.355e-01                      5.247e-01
##         data_channel_is_world                     kw_min_min
##                     7.566e-02                      1.482e-03
##                    kw_max_min                     kw_avg_min
##                    -1.168e-05                      1.159e-04
##                    kw_min_max                     kw_max_max
##                    -6.137e-07                     -3.890e-07
##                    kw_avg_max                     kw_min_avg
##                    -5.989e-07                     -9.275e-05
##                    kw_max_avg                     kw_avg_avg
##                    -9.759e-05                      7.354e-04
##     self_reference_min_shares       self_reference_max_shares
##                     7.631e-06                      2.448e-06
##    self_reference_avg_sharess               weekday_is_monday
##                    -1.435e-06                     -6.953e-01
##            weekday_is_tuesday            weekday_is_wednesday
##                    -8.045e-01                     -8.240e-01
##           weekday_is_thursday               weekday_is_friday
##                    -7.726e-01                     -5.379e-01
##           weekday_is_saturday                         LDA_00
##                     2.816e-01                     -1.174e+04
##                        LDA_01                         LDA_02
##                    -1.174e+04                     -1.174e+04
##                        LDA_03                         LDA_04
##                    -1.174e+04                     -1.174e+04
##           global_subjectivity       global_sentiment_polarity
##                     1.109e+00                     -5.925e-01
##    global_rate_positive_words      global_rate_negative_words
##                    -1.310e+00                      1.145e+00
```

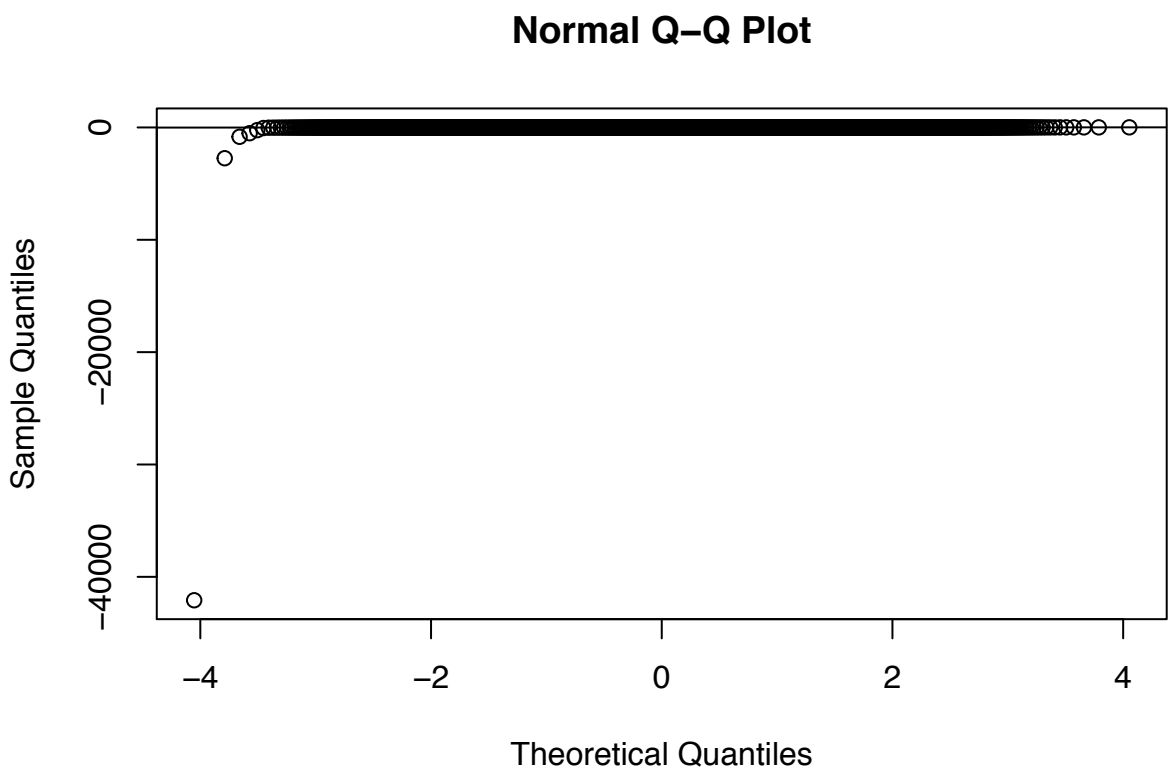
```
##           rate_positive_words            rate_negative_words
##                     1.156e+01                      1.107e+01
##         avg_positive_polarity          min_positive_polarity
##                    -1.855e-01                     -5.785e-01
##         max_positive_polarity          avg_negative_polarity
##                     1.227e-01                      8.247e-02
##         min_negative_polarity          max_negative_polarity
##                    -4.518e-02                      2.727e-01
##            title_subjectivity       title_sentiment_polarity
##                     1.152e-01                      1.615e-01
##        abs_title_subjectivity   abs_title_sentiment_polarity
##                     3.067e-01                      9.444e-02
##
## Degrees of Freedom: 19804 Total (i.e. Null);  19747 Residual
## Null Deviance:       27460
## Residual Deviance: 24880     AIC: 24990
```

```r
summary(fit.lm.weekdays)
```

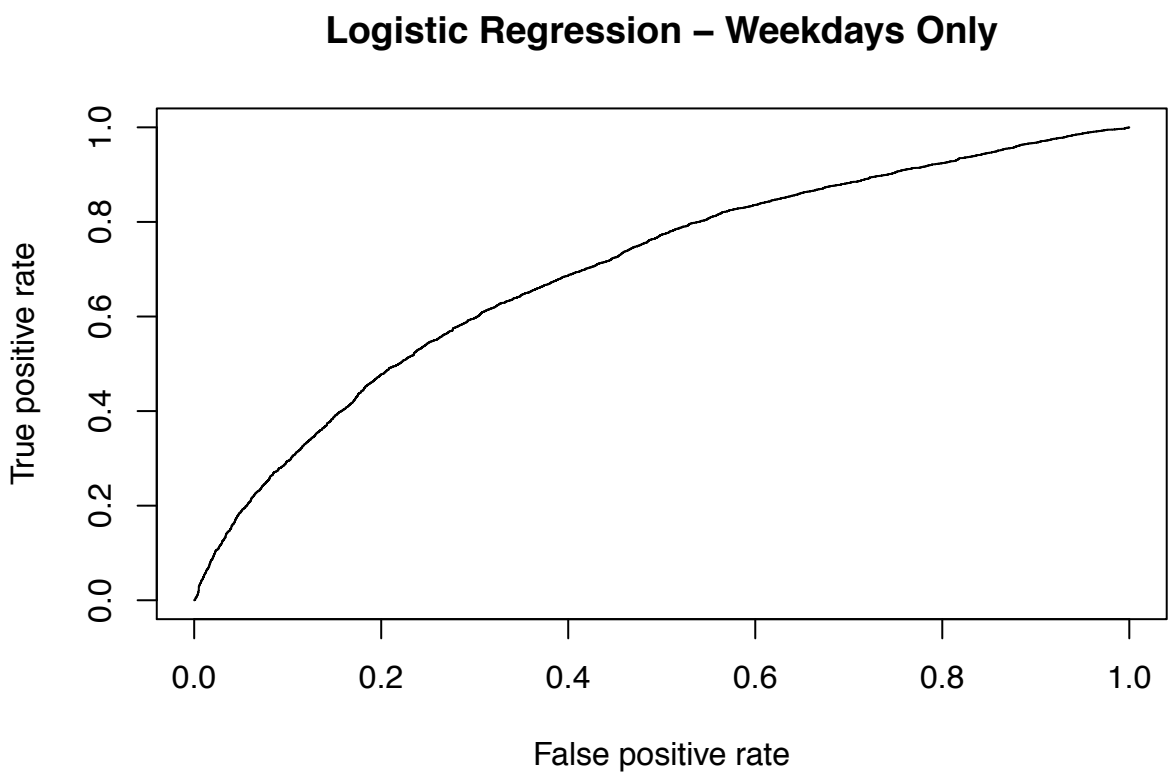
```
##
## Call:
## glm(formula = shares ~ ., family = binomial, data = news.train.weekdays)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.6146  -1.0371  -0.5741   1.0670   2.1213
##
## Coefficients:
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    1.174e+04  1.410e+05   0.083 0.933637
## timedelta                     -5.434e-05  1.044e-04  -0.521 0.602711
## n_tokens_title                 6.756e-03  7.669e-03   0.881 0.378393
## n_tokens_content               4.214e-05  6.037e-05   0.698 0.485192
## n_unique_tokens                3.971e-01  5.126e-01   0.775 0.438579
## n_non_stop_words              -1.085e+01  1.353e+02  -0.080 0.936087
## n_non_stop_unique_tokens      -1.079e+00  4.327e-01  -2.493 0.012668 *
## num_hrefs                      7.016e-03  1.841e-03   3.812 0.000138 ***
## num_self_hrefs                -2.220e-02  4.701e-03  -4.722 2.33e-06 ***
## num_imgs                       1.468e-03  2.394e-03   0.613 0.539758
## num_videos                     3.100e-03  4.276e-03   0.725 0.468386
## average_token_length          -1.142e-01  6.410e-02  -1.781 0.074888 .
## num_keywords                   4.287e-02  9.834e-03   4.360 1.30e-05 ***
## data_channel_is_lifestyle     -1.084e-01  1.040e-01  -1.042 0.297363
## data_channel_is_entertainment -1.642e-01  6.722e-02  -2.443 0.014549 *
## data_channel_is_bus           -1.115e-01  1.012e-01  -1.102 0.270664
## data_channel_is_socmed         8.355e-01  1.010e-01   8.273  < 2e-16 ***
## data_channel_is_tech           5.247e-01  9.739e-02   5.388 7.14e-08 ***
## data_channel_is_world          7.566e-02  9.918e-02   0.763 0.445520
## kw_min_min                     1.482e-03  4.330e-04   3.424 0.000617 ***
## kw_max_min                    -1.168e-05  2.167e-05  -0.539 0.589925
## kw_avg_min                     1.159e-04  1.419e-04   0.817 0.413934
## kw_min_max                    -6.137e-07  3.106e-07  -1.976 0.048167 *
## kw_max_max                    -3.890e-07  1.564e-07  -2.487 0.012878 *
## kw_avg_max                    -5.989e-07  2.268e-07  -2.640 0.008285 **
```

```
## kw_min_avg                    -9.275e-05  2.056e-05  -4.510 6.47e-06 ***
## kw_max_avg                    -9.759e-05  7.453e-06 -13.094  < 2e-16 ***
## kw_avg_avg                     7.354e-04  4.154e-05  17.702  < 2e-16 ***
## self_reference_min_shares      7.631e-06  2.754e-06   2.771 0.005593 **
## self_reference_max_shares      2.448e-06  1.444e-06   1.695 0.090043 .
## self_reference_avg_sharess    -1.435e-06  3.407e-06  -0.421 0.673565
## weekday_is_monday             -6.953e-01  7.053e-02  -9.859  < 2e-16 ***
## weekday_is_tuesday            -8.045e-01  6.962e-02 -11.556  < 2e-16 ***
## weekday_is_wednesday          -8.240e-01  6.973e-02 -11.817  < 2e-16 ***
## weekday_is_thursday           -7.726e-01  6.974e-02 -11.078  < 2e-16 ***
## weekday_is_friday             -5.379e-01  7.180e-02  -7.491 6.83e-14 ***
## weekday_is_saturday            2.816e-01  8.914e-02   3.159 0.001582 **
## LDA_00                        -1.174e+04  1.410e+05  -0.083 0.933634
## LDA_01                        -1.174e+04  1.410e+05  -0.083 0.933627
## LDA_02                        -1.174e+04  1.410e+05  -0.083 0.933628
## LDA_03                        -1.174e+04  1.410e+05  -0.083 0.933629
## LDA_04                        -1.174e+04  1.410e+05  -0.083 0.933632
## global_subjectivity            1.109e+00  2.258e-01   4.914 8.91e-07 ***
## global_sentiment_polarity     -5.925e-01  4.447e-01  -1.332 0.182734
## global_rate_positive_words    -1.310e+00  1.878e+00  -0.698 0.485342
## global_rate_negative_words     1.145e+00  3.620e+00   0.316 0.751666
## rate_positive_words            1.156e+01  1.353e+02   0.085 0.931945
## rate_negative_words            1.107e+01  1.353e+02   0.082 0.934795
## avg_positive_polarity         -1.855e-01  3.627e-01  -0.512 0.608953
## min_positive_polarity         -5.785e-01  3.041e-01  -1.902 0.057111 .
## max_positive_polarity          1.227e-01  1.133e-01   1.083 0.278993
## avg_negative_polarity          8.247e-02  3.340e-01   0.247 0.805005
## min_negative_polarity         -4.518e-02  1.215e-01  -0.372 0.710085
## max_negative_polarity          2.727e-01  2.779e-01   0.981 0.326351
## title_subjectivity             1.152e-01  7.256e-02   1.588 0.112355
## title_sentiment_polarity       1.615e-01  6.684e-02   2.417 0.015666 *
## abs_title_subjectivity         3.067e-01  9.582e-02   3.201 0.001370 **
## abs_title_sentiment_polarity   9.444e-02  1.051e-01   0.899 0.368912
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 27455  on 19804  degrees of freedom
## Residual deviance: 24878  on 19747  degrees of freedom
## AIC: 24994
##
## Number of Fisher Scoring iterations: 10
```

```r
## Q-Q plot
qqnorm(fit.lm.weekdays$residuals)
qqline(fit.lm.weekdays$residuals)
```




```
## ROC Curves
scores <- predict(fit.lm.weekdays, newdata=news.validation.weekdays,
                  type = "response")
prob <- prediction(scores, labels=news.validation.weekdays$shares)
perf <- performance(prob, "tpr", "fpr")
plot(perf, colorize=F, main="Logistic Regression - Weekdays Only")
```




```
## AUC
unlist(attributes(performance(prob, "auc"))$y.values)
```

```
## [1] 0.6975675
```

```
## Confusion Matrix
pred.fit.lm.weekdays <- ifelse(scores>0.5, 1, 0)
confusionMatrix(pred.fit.lm.weekdays, news.validation.weekdays$shares,
                positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 3384 1801
##          1 1672 3051
##
##                Accuracy : 0.6495
##                  95% CI : (0.64, 0.6589)
##     No Information Rate : 0.5103
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.2983
##  Mcnemar's Test P-Value : 0.02986
##
##             Sensitivity : 0.6288
##             Specificity : 0.6693
##          Pos Pred Value : 0.6460
```

```
##          Neg Pred Value : 0.6527
##              Prevalence : 0.4897
##          Detection Rate : 0.3079
##    Detection Prevalence : 0.4767
##       Balanced Accuracy : 0.6491
##
##        'Positive' Class : 1
##
```

```
## Prediction Error on Validation Set
mse.fit.lm.weekdays = mean((news.validation.weekdays$shares -
                            pred.fit.lm.weekdays)^2)
mse.fit.lm.weekdays
```

```
## [1] 0.3505248
```

```
#######################################################################
# Logistic Regression - Using Is_Weekend Only
#######################################################################
## Data Prep
news.train.weekend <- news.train[, -c(31:37)]
news.validation.weekend <- news.validation[, -c(31:37)]

## Model fitting
fit.lm.weekend <- glm(shares ~ ., data = news.train.weekend,
                      family=binomial)
fit.lm.weekend
```

```
##
## Call:  glm(formula = shares ~ ., family = binomial, data = news.train.weekend)
##
## Coefficients:
##                   (Intercept)                      timedelta
##                     1.168e+04                     -5.054e-05
##                n_tokens_title               n_tokens_content
##                     6.514e-03                      3.957e-05
##               n_unique_tokens               n_non_stop_words
##                     3.449e-01                     -1.079e+01
##      n_non_stop_unique_tokens                      num_hrefs
##                    -1.025e+00                      7.234e-03
##                num_self_hrefs                       num_imgs
##                    -2.219e-02                      1.394e-03
##                    num_videos           average_token_length
##                     3.008e-03                     -1.186e-01
##                  num_keywords      data_channel_is_lifestyle
##                     4.274e-02                     -1.013e-01
## data_channel_is_entertainment            data_channel_is_bus
##                    -1.630e-01                     -1.084e-01
##        data_channel_is_socmed           data_channel_is_tech
##                     8.377e-01                      5.321e-01
##         data_channel_is_world                     kw_min_min
##                     7.672e-02                      1.490e-03
##                    kw_max_min                     kw_avg_min
```

```
##                    -1.134e-05                     1.128e-04
##                    kw_min_max                    kw_max_max
##                    -6.060e-07                    -3.787e-07
##                    kw_avg_max                    kw_min_avg
##                    -6.024e-07                    -9.377e-05
##                    kw_max_avg                    kw_avg_avg
##                    -9.771e-05                     7.359e-04
##     self_reference_min_shares      self_reference_max_shares
##                     7.666e-06                     2.512e-06
##    self_reference_avg_sharess                    is_weekend
##                    -1.564e-06                     8.645e-01
##                        LDA_00                        LDA_01
##                    -1.168e+04                    -1.168e+04
##                        LDA_02                        LDA_03
##                    -1.168e+04                    -1.168e+04
##                        LDA_04            global_subjectivity
##                    -1.168e+04                     1.120e+00
##     global_sentiment_polarity     global_rate_positive_words
##                    -5.966e-01                    -1.485e+00
##    global_rate_negative_words            rate_positive_words
##                     1.538e+00                     1.151e+01
##           rate_negative_words          avg_positive_polarity
##                     1.101e+01                    -1.753e-01
##         min_positive_polarity          max_positive_polarity
##                    -5.796e-01                     1.204e-01
##         avg_negative_polarity          min_negative_polarity
##                     8.544e-02                    -4.333e-02
##         max_negative_polarity             title_subjectivity
##                     2.593e-01                     1.204e-01
##      title_sentiment_polarity         abs_title_subjectivity
##                     1.678e-01                     3.053e-01
##  abs_title_sentiment_polarity
##                     8.290e-02
##
## Degrees of Freedom: 19804 Total (i.e. Null);  19752 Residual
## Null Deviance:       27460
## Residual Deviance: 24930     AIC: 25030
```

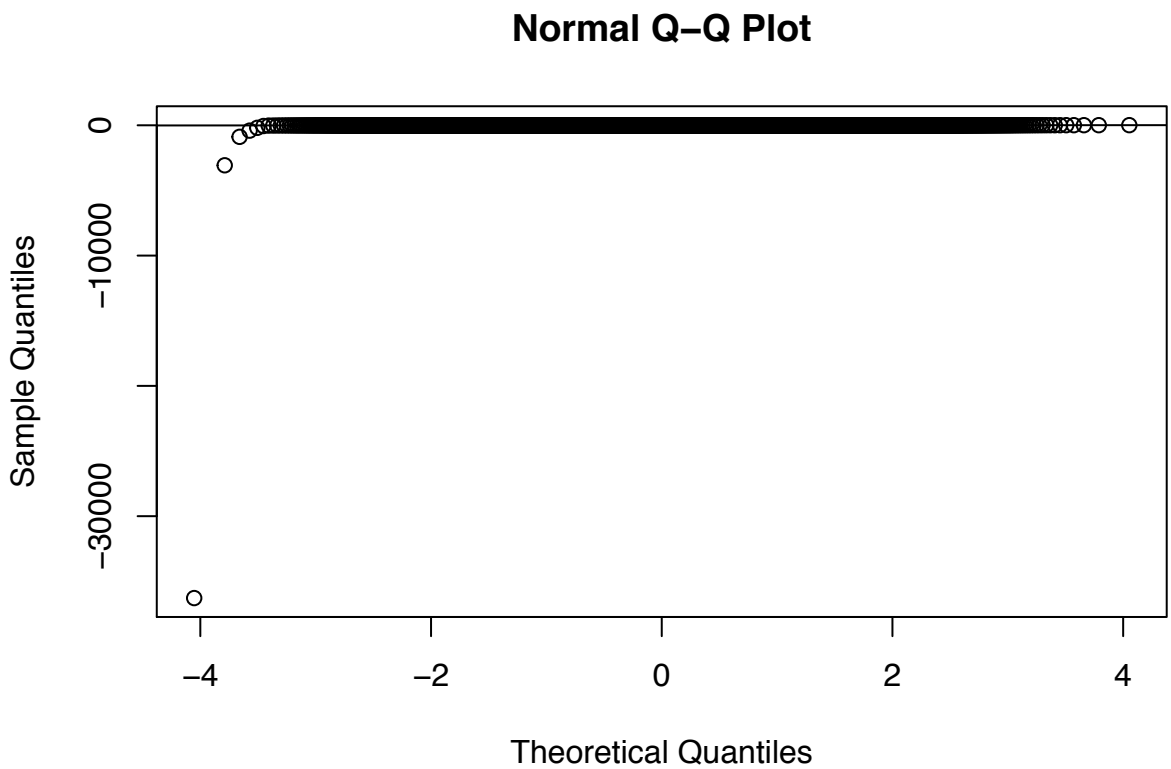
```r
summary(fit.lm.weekend)
```

```
##
## Call:
## glm(formula = shares ~ ., family = binomial, data = news.train.weekend)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.5824  -1.0383  -0.5806   1.0677   2.1420
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   1.168e+04  1.379e+05   0.085 0.932517
## timedelta                    -5.054e-05  1.042e-04  -0.485 0.627728
## n_tokens_title                6.514e-03  7.659e-03   0.851 0.395046
## n_tokens_content              3.957e-05  6.029e-05   0.656 0.511624
```

```
## n_unique_tokens                3.449e-01  5.120e-01   0.674 0.500531
## n_non_stop_words              -1.079e+01  1.324e+02  -0.082 0.935025
## n_non_stop_unique_tokens      -1.025e+00  4.323e-01  -2.371 0.017763 *
## num_hrefs                      7.234e-03  1.840e-03   3.931 8.46e-05 ***
## num_self_hrefs                -2.219e-02  4.692e-03  -4.730 2.25e-06 ***
## num_imgs                       1.394e-03  2.393e-03   0.583 0.560005
## num_videos                     3.008e-03  4.264e-03   0.706 0.480435
## average_token_length          -1.186e-01  6.401e-02  -1.853 0.063882 .
## num_keywords                   4.274e-02  9.820e-03   4.352 1.35e-05 ***
## data_channel_is_lifestyle     -1.013e-01  1.039e-01  -0.976 0.329211
## data_channel_is_entertainment -1.630e-01  6.714e-02  -2.428 0.015194 *
## data_channel_is_bus           -1.084e-01  1.011e-01  -1.072 0.283702
## data_channel_is_socmed         8.377e-01  1.009e-01   8.304  < 2e-16 ***
## data_channel_is_tech           5.321e-01  9.725e-02   5.471 4.46e-08 ***
## data_channel_is_world          7.672e-02  9.906e-02   0.774 0.438650
## kw_min_min                     1.490e-03  4.328e-04   3.442 0.000577 ***
## kw_max_min                    -1.134e-05  2.169e-05  -0.523 0.600993
## kw_avg_min                     1.128e-04  1.421e-04   0.794 0.427390
## kw_min_max                    -6.060e-07  3.102e-07  -1.954 0.050756 .
## kw_max_max                    -3.787e-07  1.563e-07  -2.422 0.015438 *
## kw_avg_max                    -6.024e-07  2.264e-07  -2.661 0.007797 **
## kw_min_avg                    -9.377e-05  2.054e-05  -4.566 4.98e-06 ***
## kw_max_avg                    -9.771e-05  7.431e-06 -13.150  < 2e-16 ***
## kw_avg_avg                     7.359e-04  4.147e-05  17.743  < 2e-16 ***
## self_reference_min_shares      7.666e-06  2.760e-06   2.778 0.005473 **
## self_reference_max_shares      2.512e-06  1.449e-06   1.734 0.082877 .
## self_reference_avg_sharess    -1.564e-06  3.418e-06  -0.458 0.647268
## is_weekend                     8.645e-01  4.754e-02  18.186  < 2e-16 ***
## LDA_00                        -1.168e+04  1.379e+05  -0.085 0.932510
## LDA_01                        -1.168e+04  1.379e+05  -0.085 0.932503
## LDA_02                        -1.168e+04  1.379e+05  -0.085 0.932504
## LDA_03                        -1.168e+04  1.379e+05  -0.085 0.932504
## LDA_04                        -1.168e+04  1.379e+05  -0.085 0.932507
## global_subjectivity            1.120e+00  2.255e-01   4.967 6.80e-07 ***
## global_sentiment_polarity     -5.966e-01  4.442e-01  -1.343 0.179208
## global_rate_positive_words    -1.485e+00  1.875e+00  -0.792 0.428418
## global_rate_negative_words     1.538e+00  3.612e+00   0.426 0.670233
## rate_positive_words            1.151e+01  1.324e+02   0.087 0.930730
## rate_negative_words            1.101e+01  1.324e+02   0.083 0.933731
## avg_positive_polarity         -1.753e-01  3.622e-01  -0.484 0.628357
## min_positive_polarity         -5.796e-01  3.037e-01  -1.908 0.056371 .
## max_positive_polarity          1.204e-01  1.132e-01   1.064 0.287416
## avg_negative_polarity          8.544e-02  3.335e-01   0.256 0.797831
## min_negative_polarity         -4.333e-02  1.214e-01  -0.357 0.721096
## max_negative_polarity          2.593e-01  2.775e-01   0.934 0.350105
## title_subjectivity             1.204e-01  7.244e-02   1.663 0.096388 .
## title_sentiment_polarity       1.678e-01  6.674e-02   2.515 0.011916 *
## abs_title_subjectivity         3.053e-01  9.568e-02   3.191 0.001418 **
## abs_title_sentiment_polarity   8.290e-02  1.049e-01   0.790 0.429500
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 27455  on 19804  degrees of freedom
## Residual deviance: 24926  on 19752  degrees of freedom
## AIC: 25032
##
## Number of Fisher Scoring iterations: 10
```

```
## Q-Q plot
qqnorm(fit.lm.weekend$residuals)
qqline(fit.lm.weekend$residuals)
```




```
## ROC Curves
scores <- predict(fit.lm.weekend, newdata=news.validation.weekend,
                  type = "response")
prob <- prediction(scores, labels=news.validation.weekend$shares)
perf <- performance(prob, "tpr", "fpr")
plot(perf, colorize=F, main="Logistic Regression - Is_Weekend Only")
```

```r
## AUC
unlist(attributes(performance(prob, "auc"))$y.values)
```

```
## [1] 0.6956649
```

```r
## Confusion Matrix
pred.fit.lm.weekend <- ifelse(scores>0.45, 1, 0)
confusionMatrix(pred.fit.lm.weekend, news.validation.weekend$shares, positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 2869 1377
##          1 2187 3475
##
##                Accuracy : 0.6403
##                  95% CI : (0.6308, 0.6497)
##     No Information Rate : 0.5103
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.2827
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.7162
##             Specificity : 0.5674
##          Pos Pred Value : 0.6137
##          Neg Pred Value : 0.6757
```

```
##              Prevalence : 0.4897
##          Detection Rate : 0.3507
##    Detection Prevalence : 0.5715
##       Balanced Accuracy : 0.6418
##
##        'Positive' Class : 1
##
```

```
## Prediction Error on Validation Set
mse.fit.lm.weekdays = mean((news.validation$shares-pred.fit.lm.weekdays)^2)
mse.fit.lm.weekdays
```

```
## [1] 0.3505248
```

```
#######################################################################
# Logistic Regression - Categorical Variable
#######################################################################
## Data Prep
names(news.train)
```

```
##  [1] "timedelta"                     "n_tokens_title"
##  [3] "n_tokens_content"              "n_unique_tokens"
##  [5] "n_non_stop_words"              "n_non_stop_unique_tokens"
##  [7] "num_hrefs"                     "num_self_hrefs"
##  [9] "num_imgs"                      "num_videos"
## [11] "average_token_length"          "num_keywords"
## [13] "data_channel_is_lifestyle"     "data_channel_is_entertainment"
## [15] "data_channel_is_bus"           "data_channel_is_socmed"
## [17] "data_channel_is_tech"          "data_channel_is_world"
## [19] "kw_min_min"                    "kw_max_min"
## [21] "kw_avg_min"                    "kw_min_max"
## [23] "kw_max_max"                    "kw_avg_max"
## [25] "kw_min_avg"                    "kw_max_avg"
## [27] "kw_avg_avg"                    "self_reference_min_shares"
## [29] "self_reference_max_shares"     "self_reference_avg_sharess"
## [31] "weekday_is_monday"             "weekday_is_tuesday"
## [33] "weekday_is_wednesday"          "weekday_is_thursday"
## [35] "weekday_is_friday"             "weekday_is_saturday"
## [37] "weekday_is_sunday"             "is_weekend"
## [39] "LDA_00"                        "LDA_01"
## [41] "LDA_02"                        "LDA_03"
## [43] "LDA_04"                        "global_subjectivity"
## [45] "global_sentiment_polarity"     "global_rate_positive_words"
## [47] "global_rate_negative_words"    "rate_positive_words"
## [49] "rate_negative_words"           "avg_positive_polarity"
## [51] "min_positive_polarity"         "max_positive_polarity"
## [53] "avg_negative_polarity"         "min_negative_polarity"
## [55] "max_negative_polarity"         "title_subjectivity"
## [57] "title_sentiment_polarity"      "abs_title_subjectivity"
## [59] "abs_title_sentiment_polarity"  "shares"
```

```r
news.train.cat <- cbind(news.train[, c(1:30)],
                        weekday = apply(news.train[, c(31:37)], 1,
                                        function(x) names(x)[as.logical(as.numeric(
                                          as.character(x)))]),
                        news.train[, c(39:60)])

dim(news.train.cat)
```

```
## [1] 19805    53
```

```r
names(news.train.cat)
```

```
##  [1] "timedelta"                     "n_tokens_title"
##  [3] "n_tokens_content"              "n_unique_tokens"
##  [5] "n_non_stop_words"              "n_non_stop_unique_tokens"
##  [7] "num_hrefs"                     "num_self_hrefs"
##  [9] "num_imgs"                      "num_videos"
## [11] "average_token_length"          "num_keywords"
## [13] "data_channel_is_lifestyle"     "data_channel_is_entertainment"
## [15] "data_channel_is_bus"           "data_channel_is_socmed"
## [17] "data_channel_is_tech"          "data_channel_is_world"
## [19] "kw_min_min"                    "kw_max_min"
## [21] "kw_avg_min"                    "kw_min_max"
## [23] "kw_max_max"                    "kw_avg_max"
## [25] "kw_min_avg"                    "kw_max_avg"
## [27] "kw_avg_avg"                    "self_reference_min_shares"
## [29] "self_reference_max_shares"     "self_reference_avg_sharess"
## [31] "weekday"                       "LDA_00"
## [33] "LDA_01"                        "LDA_02"
## [35] "LDA_03"                        "LDA_04"
## [37] "global_subjectivity"           "global_sentiment_polarity"
## [39] "global_rate_positive_words"    "global_rate_negative_words"
## [41] "rate_positive_words"           "rate_negative_words"
## [43] "avg_positive_polarity"         "min_positive_polarity"
## [45] "max_positive_polarity"         "avg_negative_polarity"
## [47] "min_negative_polarity"         "max_negative_polarity"
## [49] "title_subjectivity"            "title_sentiment_polarity"
## [51] "abs_title_subjectivity"        "abs_title_sentiment_polarity"
## [53] "shares"
```

```r
news.validation.cat <- cbind(news.validation[, c(1:30)],
                        weekday = apply(news.validation[, c(31:37)], 1,
                                        function(x) names(x)[as.logical(as.numeric(
                                          as.character(x)))]),
                        news.validation[, c(39:60)])

## Model fitting

fit.lm.cat <- glm(shares ~ ., data = news.train.cat,
                      family=binomial)
fit.lm.cat
```

```
##
```

```
## Call:  glm(formula = shares ~ ., family = binomial, data = news.train.cat)
##
## Coefficients:
##                   (Intercept)                      timedelta
##                     1.174e+04                     -5.434e-05
##                n_tokens_title               n_tokens_content
##                     6.756e-03                      4.214e-05
##               n_unique_tokens               n_non_stop_words
##                     3.971e-01                     -1.085e+01
##      n_non_stop_unique_tokens                      num_hrefs
##                    -1.079e+00                      7.016e-03
##                num_self_hrefs                       num_imgs
##                    -2.220e-02                      1.468e-03
##                    num_videos           average_token_length
##                     3.100e-03                     -1.142e-01
##                  num_keywords      data_channel_is_lifestyle
##                     4.287e-02                     -1.084e-01
## data_channel_is_entertainment            data_channel_is_bus
##                    -1.642e-01                     -1.115e-01
##        data_channel_is_socmed           data_channel_is_tech
##                     8.355e-01                      5.247e-01
##         data_channel_is_world                     kw_min_min
##                     7.566e-02                      1.482e-03
##                    kw_max_min                     kw_avg_min
##                    -1.168e-05                      1.159e-04
##                    kw_min_max                     kw_max_max
##                    -6.137e-07                     -3.890e-07
##                    kw_avg_max                     kw_min_avg
##                    -5.989e-07                     -9.275e-05
##                    kw_max_avg                     kw_avg_avg
##                    -9.759e-05                      7.354e-04
##     self_reference_min_shares      self_reference_max_shares
##                     7.631e-06                      2.448e-06
##    self_reference_avg_sharess       weekdayweekday_is_monday
##                    -1.435e-06                     -1.574e-01
##    weekdayweekday_is_saturday       weekdayweekday_is_sunday
##                     8.195e-01                      5.379e-01
##    weekdayweekday_is_thursday      weekdayweekday_is_tuesday
##                    -2.347e-01                     -2.666e-01
##   weekdayweekday_is_wednesday                         LDA_00
##                    -2.861e-01                     -1.174e+04
##                        LDA_01                         LDA_02
##                    -1.174e+04                     -1.174e+04
##                        LDA_03                         LDA_04
##                    -1.174e+04                     -1.174e+04
##           global_subjectivity      global_sentiment_polarity
##                     1.109e+00                     -5.925e-01
##    global_rate_positive_words     global_rate_negative_words
##                    -1.310e+00                      1.145e+00
##           rate_positive_words            rate_negative_words
##                     1.156e+01                      1.107e+01
##         avg_positive_polarity          min_positive_polarity
##                    -1.855e-01                     -5.785e-01
##         max_positive_polarity          avg_negative_polarity
```

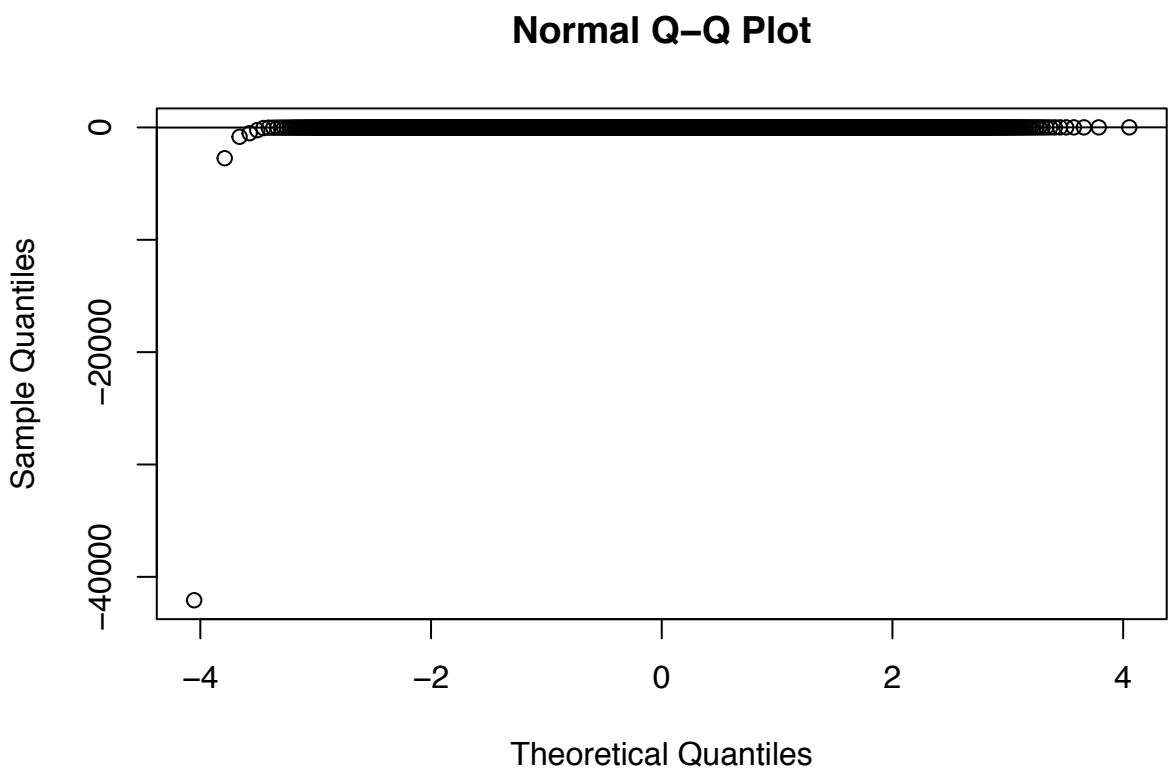
```
##                     1.227e-01                      8.247e-02
##         min_negative_polarity          max_negative_polarity
##                    -4.518e-02                      2.727e-01
##            title_subjectivity       title_sentiment_polarity
##                     1.152e-01                      1.615e-01
##        abs_title_subjectivity   abs_title_sentiment_polarity
##                     3.067e-01                      9.444e-02
##
## Degrees of Freedom: 19804 Total (i.e. Null);  19747 Residual
## Null Deviance:       27460
## Residual Deviance: 24880     AIC: 24990
```

```r
summary(fit.lm.cat)
```

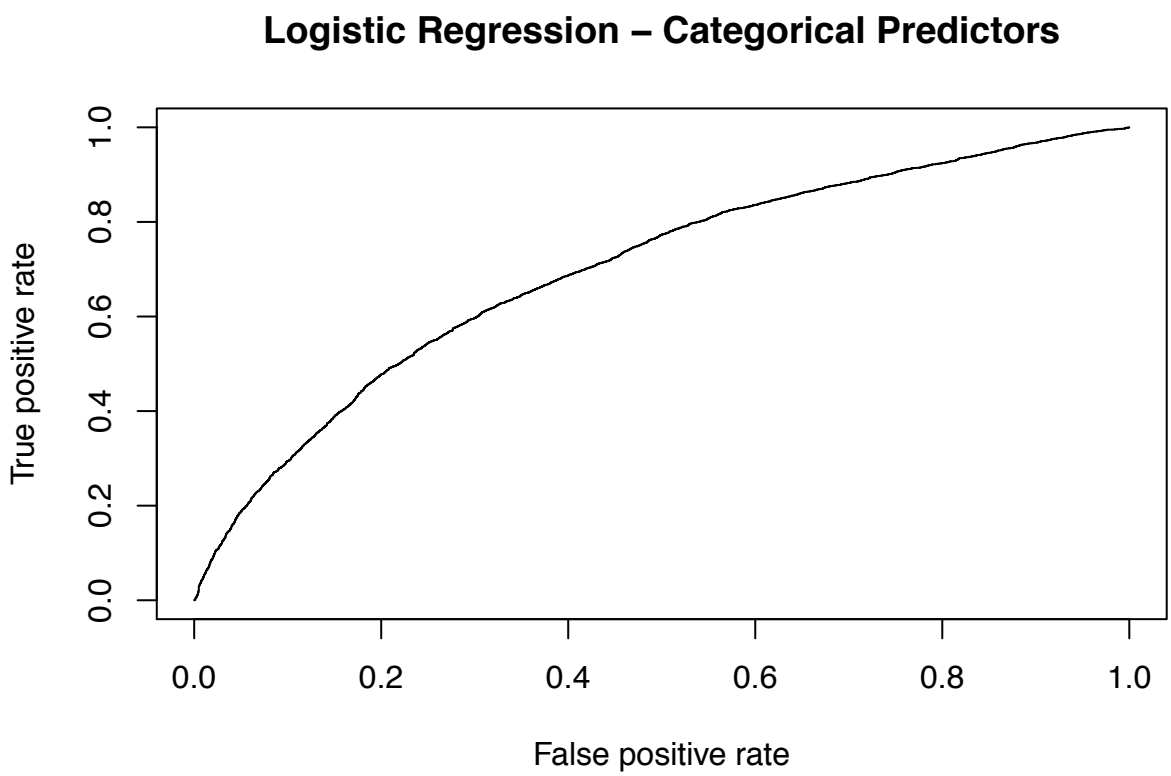
```
##
## Call:
## glm(formula = shares ~ ., family = binomial, data = news.train.cat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.6146  -1.0371  -0.5741   1.0670   2.1213
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   1.174e+04  1.410e+05   0.083 0.933640
## timedelta                    -5.434e-05  1.044e-04  -0.521 0.602711
## n_tokens_title                6.756e-03  7.669e-03   0.881 0.378393
## n_tokens_content              4.214e-05  6.037e-05   0.698 0.485192
## n_unique_tokens               3.971e-01  5.126e-01   0.775 0.438579
## n_non_stop_words             -1.085e+01  1.353e+02  -0.080 0.936087
## n_non_stop_unique_tokens     -1.079e+00  4.327e-01  -2.493 0.012668 *
## num_hrefs                     7.016e-03  1.841e-03   3.812 0.000138 ***
## num_self_hrefs               -2.220e-02  4.701e-03  -4.722 2.33e-06 ***
## num_imgs                      1.468e-03  2.394e-03   0.613 0.539758
## num_videos                    3.100e-03  4.276e-03   0.725 0.468386
## average_token_length         -1.142e-01  6.410e-02  -1.781 0.074888 .
## num_keywords                  4.287e-02  9.834e-03   4.360 1.30e-05 ***
## data_channel_is_lifestyle    -1.084e-01  1.040e-01  -1.042 0.297363
## data_channel_is_entertainment -1.642e-01  6.722e-02  -2.443 0.014549 *
## data_channel_is_bus          -1.115e-01  1.012e-01  -1.102 0.270664
## data_channel_is_socmed        8.355e-01  1.010e-01   8.273  < 2e-16 ***
## data_channel_is_tech          5.247e-01  9.739e-02   5.388 7.14e-08 ***
## data_channel_is_world         7.566e-02  9.918e-02   0.763 0.445520
## kw_min_min                    1.482e-03  4.330e-04   3.424 0.000617 ***
## kw_max_min                   -1.168e-05  2.167e-05  -0.539 0.589925
## kw_avg_min                    1.159e-04  1.419e-04   0.817 0.413934
## kw_min_max                   -6.137e-07  3.106e-07  -1.976 0.048167 *
## kw_max_max                   -3.890e-07  1.564e-07  -2.487 0.012878 *
## kw_avg_max                   -5.989e-07  2.268e-07  -2.640 0.008285 **
## kw_min_avg                   -9.275e-05  2.056e-05  -4.510 6.47e-06 ***
## kw_max_avg                   -9.759e-05  7.453e-06 -13.094  < 2e-16 ***
## kw_avg_avg                    7.354e-04  4.154e-05  17.702  < 2e-16 ***
## self_reference_min_shares     7.631e-06  2.754e-06   2.771 0.005593 **
## self_reference_max_shares     2.448e-06  1.444e-06   1.695 0.090043 .
```

```
## self_reference_avg_sharess    -1.435e-06  3.407e-06  -0.421 0.673565
## weekdayweekday_is_monday      -1.574e-01  5.387e-02  -2.922 0.003475 **
## weekdayweekday_is_saturday     8.195e-01  7.706e-02  10.634  < 2e-16 ***
## weekdayweekday_is_sunday       5.379e-01  7.180e-02   7.491 6.83e-14 ***
## weekdayweekday_is_thursday    -2.347e-01  5.286e-02  -4.441 8.95e-06 ***
## weekdayweekday_is_tuesday     -2.666e-01  5.263e-02  -5.067 4.05e-07 ***
## weekdayweekday_is_wednesday   -2.861e-01  5.268e-02  -5.432 5.59e-08 ***
## LDA_00                        -1.174e+04  1.410e+05  -0.083 0.933634
## LDA_01                        -1.174e+04  1.410e+05  -0.083 0.933628
## LDA_02                        -1.174e+04  1.410e+05  -0.083 0.933628
## LDA_03                        -1.174e+04  1.410e+05  -0.083 0.933629
## LDA_04                        -1.174e+04  1.410e+05  -0.083 0.933632
## global_subjectivity            1.109e+00  2.258e-01   4.914 8.91e-07 ***
## global_sentiment_polarity     -5.925e-01  4.447e-01  -1.332 0.182734
## global_rate_positive_words    -1.310e+00  1.878e+00  -0.698 0.485342
## global_rate_negative_words     1.145e+00  3.620e+00   0.316 0.751666
## rate_positive_words            1.156e+01  1.353e+02   0.085 0.931945
## rate_negative_words            1.107e+01  1.353e+02   0.082 0.934795
## avg_positive_polarity         -1.855e-01  3.627e-01  -0.512 0.608953
## min_positive_polarity         -5.785e-01  3.041e-01  -1.902 0.057111 .
## max_positive_polarity          1.227e-01  1.133e-01   1.083 0.278993
## avg_negative_polarity          8.247e-02  3.340e-01   0.247 0.805005
## min_negative_polarity         -4.518e-02  1.215e-01  -0.372 0.710085
## max_negative_polarity          2.727e-01  2.779e-01   0.981 0.326351
## title_subjectivity             1.152e-01  7.256e-02   1.588 0.112355
## title_sentiment_polarity       1.615e-01  6.684e-02   2.417 0.015666 *
## abs_title_subjectivity         3.067e-01  9.582e-02   3.201 0.001370 **
## abs_title_sentiment_polarity   9.444e-02  1.051e-01   0.899 0.368912
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 27455  on 19804  degrees of freedom
## Residual deviance: 24878  on 19747  degrees of freedom
## AIC: 24994
##
## Number of Fisher Scoring iterations: 10
```

```r
## Q-Q plot
qqnorm(fit.lm.cat$residuals)
qqline(fit.lm.cat$residuals)
```




```
## ROC Curves
scores <- predict(fit.lm.cat, newdata=news.validation.cat,
                  type = "response")
prob <- prediction(scores, labels=news.validation.cat$shares)
perf <- performance(prob, "tpr", "fpr")
plot(perf, colorize=F, main="Logistic Regression - Categorical Predictors")
```




```r
## AUC
unlist(attributes(performance(prob, "auc"))$y.values)
```

```
## [1] 0.6975675
```

```r
## Confusion Matrix
pred.fit.lm.cat <- ifelse(scores>0.45, 1, 0)
confusionMatrix(pred.fit.lm.cat, news.validation.cat$shares, positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 2867 1397
##          1 2189 3455
##
##                Accuracy : 0.6381
##                  95% CI : (0.6285, 0.6475)
##     No Information Rate : 0.5103
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.2782
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.7121
##             Specificity : 0.5670
##          Pos Pred Value : 0.6122
##          Neg Pred Value : 0.6724
```

```
##              Prevalence : 0.4897
##          Detection Rate : 0.3487
##    Detection Prevalence : 0.5696
##       Balanced Accuracy : 0.6396
##
##        'Positive' Class : 1
##
```

```
## Prediction Error on Validation Set
mse.fit.lm.cat = mean((news.validation.cat$shares-pred.fit.lm.cat)^2)
mse.fit.lm.cat
```

```
## [1] 0.3619298
```

2.7 Data Cleanup

Based on the data and initial regression analysis, we perform data clean up as follows:

```
#######################################################################
# Data Cleanup
#######################################################################
## Remove outlier based on n_unique_tokens, n_non_stop_words,
## n_non_stop_unique_tokens values
news.data <- news.data[news.data[, 4] <= 1,]
news.data <- news.data[news.data[, 5] <= 1,]
news.data <- news.data[news.data[, 6] <= 1,]

## Remove missing values for rate_positive_words and rate_negative_words
news.data <- news.data[news.data$rate_positive_words +
                         news.data$rate_negative_words != 0, ]

## Remove missing values for average_token_length (Encoded as 0)
news.data <- news.data[news.data[, 11] != 0, ]

## Remove other redundant columns
news.data <- news.data[, -c(31:37)]
dim(news.data)
```

```
## [1] 38458    53
```

```
## Resampling
sample.index <- sample(1:3, size=nrow(news.data), replace=TRUE,
                         prob=c(0.5,0.25,0.25))
news.train <- news.data[sample.index == 1, ]
news.validation <- news.data[sample.index == 2, ]
news.test <- news.data[sample.index == 3, ]

dim(news.train)
```

```
## [1] 19208    53
```

```r
dim(news.test)
```

```
## [1] 9543   53
```

```r
dim(news.validation)
```

```
## [1] 9707   53
```

```r
## Similarly, cleaning up news.data.copy
news.data.copy <- news.data.copy[news.data.copy[, 4] <= 1,]
news.data.copy <- news.data.copy[news.data.copy[, 5] <= 1,]
news.data.copy <- news.data.copy[news.data.copy[, 6] <= 1,]
news.data.copy <- news.data.copy[news.data.copy$rate_positive_words +
                         news.data.copy$rate_negative_words != 0, ]
news.data.copy <- news.data.copy[news.data.copy[, 11] != 0, ]
news.data.copy <- news.data.copy[, -c(31:37)]
dim(news.data.copy)
```

```
## [1] 38458    53
```

3 Data Modelling

```
## Creating formula for Top 30 predictors
f <- "kw_avg_avg + self_reference_avg_sharess + self_reference_min_shares + kw_max_avg + kw_min_avg + LI
```

3.1 Logistic Regression - All 52 predictors

```
#######################################################################
# Logistic Regression - All 52 predictors
#######################################################################
## Model fitting
fit.lm.all <- glm(shares ~ ., data = news.train, family=binomial)
fit.lm.all
```

```
##
## Call:  glm(formula = shares ~ ., family = binomial, data = news.train)
##
## Coefficients:
##                   (Intercept)                      timedelta
##                     1.085e+07                      7.810e-06
##                n_tokens_title                n_tokens_content
##                    -6.616e-03                      1.374e-04
##               n_unique_tokens                n_non_stop_words
##                    -8.414e-01                     -1.085e+07
##      n_non_stop_unique_tokens                      num_hrefs
##                     2.460e-02                      8.720e-03
##                num_self_hrefs                       num_imgs
##                    -1.638e-02                      1.202e-03
##                    num_videos            average_token_length
##                     3.937e-04                     -5.618e-02
##                  num_keywords       data_channel_is_lifestyle
##                     3.707e-02                     -1.702e-01
## data_channel_is_entertainment             data_channel_is_bus
##                    -3.439e-01                     -2.356e-01
##        data_channel_is_socmed            data_channel_is_tech
##                     7.118e-01                      5.302e-01
##         data_channel_is_world                      kw_min_min
##                    -9.133e-02                      2.002e-03
##                    kw_max_min                      kw_avg_min
##                     2.678e-05                     -1.936e-04
##                    kw_min_max                      kw_max_max
##                    -7.426e-07                     -2.977e-07
##                    kw_avg_max                      kw_min_avg
##                    -6.787e-07                     -9.164e-05
##                    kw_max_avg                      kw_avg_avg
##                    -9.211e-05                      7.510e-04
##     self_reference_min_shares       self_reference_max_shares
##                     8.774e-06                      4.057e-07
##    self_reference_avg_sharess                      is_weekend
##                     2.134e-06                      8.400e-01
```

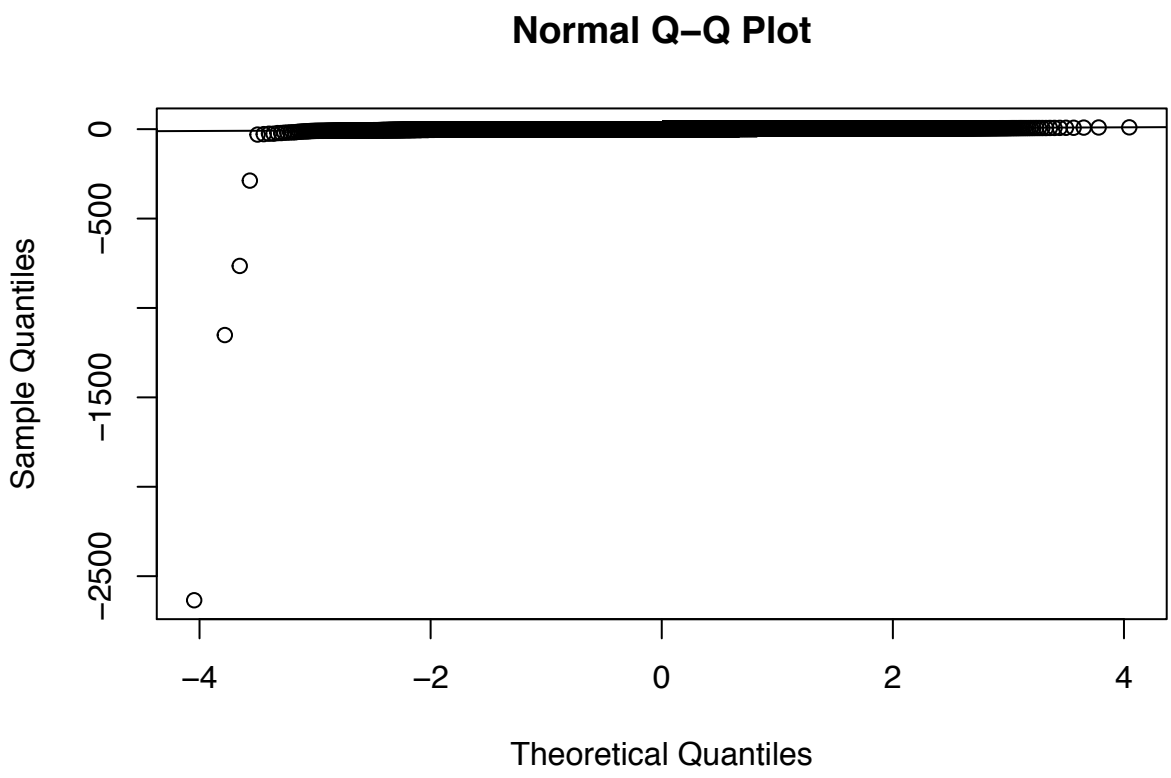
```
##                        LDA_00                        LDA_01
##                     7.456e-01                    -4.475e-01
##                        LDA_02                        LDA_03
##                    -4.856e-01                    -4.035e-01
##                        LDA_04            global_subjectivity
##                            NA                     9.007e-01
##     global_sentiment_polarity     global_rate_positive_words
##                     1.964e-01                    -4.346e+00
##    global_rate_negative_words            rate_positive_words
##                     6.052e+00                     4.516e-01
##           rate_negative_words          avg_positive_polarity
##                            NA                    -3.947e-01
##         min_positive_polarity          max_positive_polarity
##                    -4.098e-01                    -6.850e-02
##         avg_negative_polarity          min_negative_polarity
##                    -2.582e-02                    -1.249e-02
##         max_negative_polarity             title_subjectivity
##                     3.100e-02                     1.528e-01
##      title_sentiment_polarity         abs_title_subjectivity
##                     2.119e-01                     2.416e-01
##  abs_title_sentiment_polarity
##                    -3.109e-02
##
## Degrees of Freedom: 19207 Total (i.e. Null);  19157 Residual
## Null Deviance:       26620
## Residual Deviance: 23970     AIC: 24070
```

```r
summary(fit.lm.all)
```

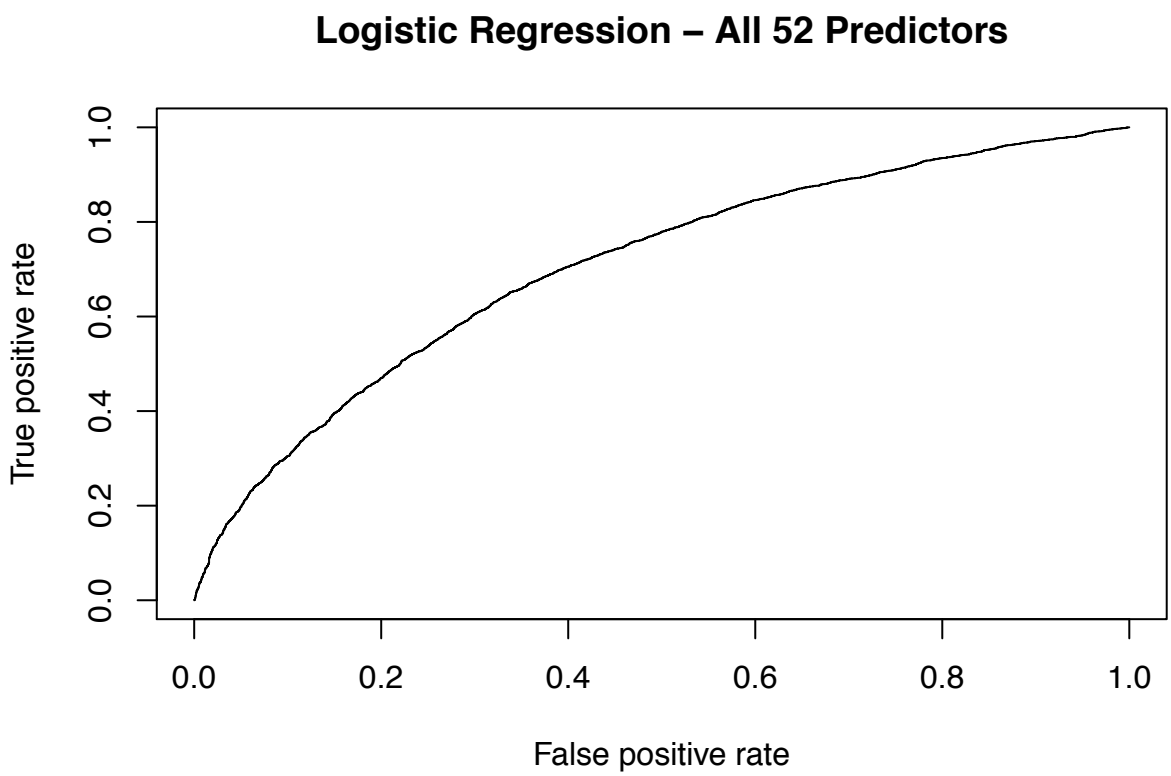
```
##
## Call:
## glm(formula = shares ~ ., family = binomial, data = news.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.9690  -1.0271  -0.6367   1.0634   2.1522
##
## Coefficients: (2 not defined because of singularities)
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    1.085e+07  7.559e+06   1.435 0.151231
## timedelta                      7.810e-06  1.052e-04   0.074 0.940816
## n_tokens_title                -6.616e-03  7.821e-03  -0.846 0.397622
## n_tokens_content               1.374e-04  6.513e-05   2.110 0.034836 *
## n_unique_tokens               -8.414e-01  6.155e-01  -1.367 0.171612
## n_non_stop_words              -1.085e+07  7.559e+06  -1.435 0.151231
## n_non_stop_unique_tokens       2.460e-02  4.503e-01   0.055 0.956430
## num_hrefs                      8.720e-03  1.901e-03   4.587 4.50e-06 ***
## num_self_hrefs                -1.638e-02  4.945e-03  -3.312 0.000926 ***
## num_imgs                       1.202e-03  2.507e-03   0.480 0.631417
## num_videos                     3.937e-04  4.110e-03   0.096 0.923688
## average_token_length          -5.618e-02  6.686e-02  -0.840 0.400787
## num_keywords                   3.707e-02  1.005e-02   3.689 0.000226 ***
## data_channel_is_lifestyle     -1.702e-01  1.061e-01  -1.604 0.108688
## data_channel_is_entertainment -3.439e-01  6.985e-02  -4.924 8.48e-07 ***
```

```
## data_channel_is_bus           -2.356e-01  1.031e-01  -2.284 0.022350 *
## data_channel_is_socmed         7.118e-01  1.034e-01   6.885 5.76e-12 ***
## data_channel_is_tech           5.302e-01  9.960e-02   5.324 1.02e-07 ***
## data_channel_is_world         -9.133e-02  1.017e-01  -0.898 0.369079
## kw_min_min                     2.002e-03  4.330e-04   4.623 3.78e-06 ***
## kw_max_min                     2.678e-05  1.475e-05   1.815 0.069521 .
## kw_avg_min                    -1.936e-04  9.267e-05  -2.090 0.036655 *
## kw_min_max                    -7.426e-07  3.212e-07  -2.312 0.020765 *
## kw_max_max                    -2.977e-07  1.574e-07  -1.892 0.058534 .
## kw_avg_max                    -6.787e-07  2.275e-07  -2.983 0.002850 **
## kw_min_avg                    -9.164e-05  2.099e-05  -4.366 1.26e-05 ***
## kw_max_avg                    -9.211e-05  6.936e-06 -13.280  < 2e-16 ***
## kw_avg_avg                     7.510e-04  4.168e-05  18.020  < 2e-16 ***
## self_reference_min_shares      8.774e-06  3.008e-06   2.917 0.003531 **
## self_reference_max_shares      4.057e-07  1.442e-06   0.281 0.778418
## self_reference_avg_sharess     2.134e-06  3.611e-06   0.591 0.554637
## is_weekend                     8.400e-01  4.876e-02  17.230  < 2e-16 ***
## LDA_00                         7.456e-01  1.214e-01   6.140 8.26e-10 ***
## LDA_01                        -4.475e-01  1.353e-01  -3.308 0.000938 ***
## LDA_02                        -4.856e-01  1.222e-01  -3.974 7.06e-05 ***
## LDA_03                        -4.035e-01  1.288e-01  -3.133 0.001733 **
## LDA_04                                NA         NA      NA       NA
## global_subjectivity            9.007e-01  2.271e-01   3.966 7.29e-05 ***
## global_sentiment_polarity      1.964e-01  4.399e-01   0.446 0.655319
## global_rate_positive_words    -4.346e+00  1.901e+00  -2.286 0.022242 *
## global_rate_negative_words     6.052e+00  3.683e+00   1.643 0.100353
## rate_positive_words            4.516e-01  3.011e-01   1.500 0.133642
## rate_negative_words                   NA         NA      NA       NA
## avg_positive_polarity         -3.947e-01  3.604e-01  -1.095 0.273445
## min_positive_polarity         -4.098e-01  3.093e-01  -1.325 0.185189
## max_positive_polarity         -6.850e-02  1.157e-01  -0.592 0.553861
## avg_negative_polarity         -2.582e-02  3.353e-01  -0.077 0.938616
## min_negative_polarity         -1.249e-02  1.233e-01  -0.101 0.919282
## max_negative_polarity          3.100e-02  2.771e-01   0.112 0.910919
## title_subjectivity             1.528e-01  7.392e-02   2.066 0.038789 *
## title_sentiment_polarity       2.119e-01  6.770e-02   3.130 0.001747 **
## abs_title_subjectivity         2.416e-01  9.847e-02   2.454 0.014143 *
## abs_title_sentiment_polarity  -3.109e-02  1.059e-01  -0.294 0.769071
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 26624  on 19207  degrees of freedom
## Residual deviance: 23970  on 19157  degrees of freedom
## AIC: 24072
##
## Number of Fisher Scoring iterations: 5
```

```r
## Q-Q plots
qqnorm(fit.lm.all$residuals)
qqline(fit.lm.all$residuals)
```




```
## ROC Curves
scores <- predict(fit.lm.all, newdata= news.validation, type = "response")
prob <- prediction(scores, labels=news.validation$shares)
perf <- performance(prob, "tpr", "fpr")
plot(perf, colorize=F, main="Logistic Regression - All 52 Predictors")
```




```
## AUC
auc.fit.lm.all <- unlist(attributes(performance(prob, "auc"))$y.values)
auc.fit.lm.all
```

```
## [1] 0.7049882
```

```
## Confusion Matrix
pred.fit.lm.all <- ifelse(scores>0.5, 1, 0)
confusionMatrix(pred.fit.lm.all, news.validation$shares, positive = "1")
```

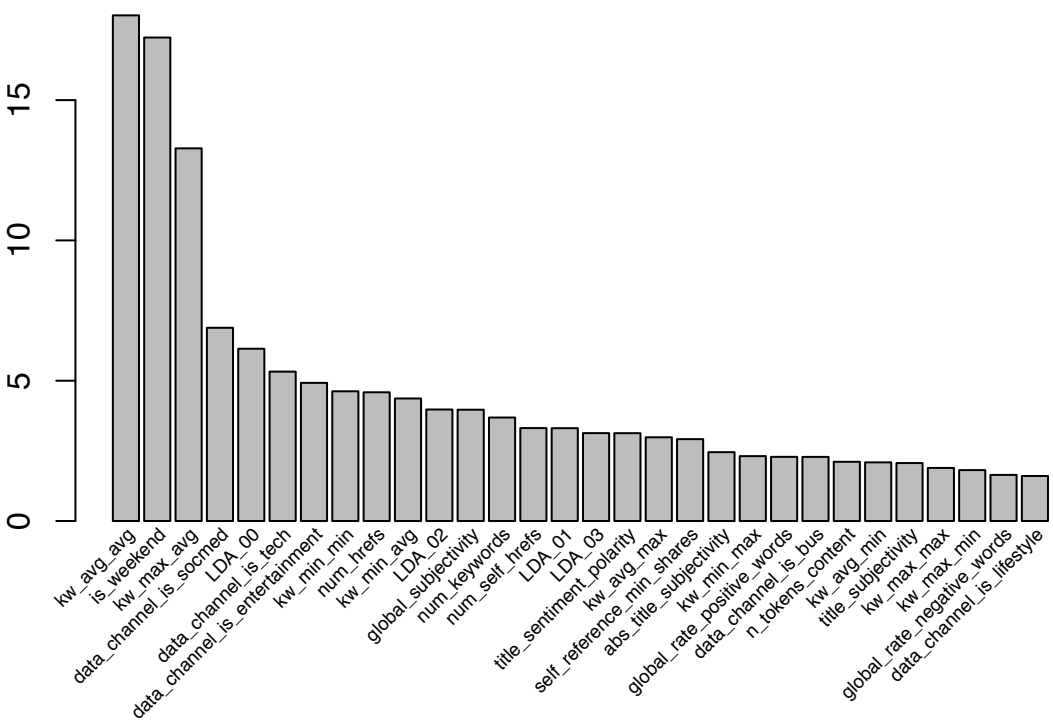
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 3353 1732
##          1 1613 3009
##
##                Accuracy : 0.6554
##                  95% CI : (0.6459, 0.6649)
##     No Information Rate : 0.5116
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.31
##  Mcnemar's Test P-Value : 0.04133
##
##             Sensitivity : 0.6347
##             Specificity : 0.6752
##          Pos Pred Value : 0.6510
```

```
##          Neg Pred Value : 0.6594
##              Prevalence : 0.4884
##          Detection Rate : 0.3100
##    Detection Prevalence : 0.4762
##       Balanced Accuracy : 0.6549
##
##        'Positive' Class : 1
##
```

```
## Prediction Error on Validation Set
mse.fit.lm.all <- mean((news.validation$shares-pred.fit.lm.all)^2)
mse.fit.lm.all
```

```
## [1] 0.3445967
```

```
## Variable Importance
par(mfrow=c(1,1))
fit.lm.all.imp <- varImp(fit.lm.all)
fit.lm.all.imp$Names <- row.names(fit.lm.all.imp)
imp.values <- fit.lm.all.imp[order(fit.lm.all.imp$Overall, decreasing=TRUE)[1:30],1]
imp.names <- fit.lm.all.imp[order(fit.lm.all.imp$Overall, decreasing=TRUE)[1:30],2]
#barplot(imp.values, names = imp.names, las = 2)
imp.values.plot <- barplot(imp.values, axes = FALSE, axisnames = FALSE)
text(imp.values.plot, par("usr")[3], labels = imp.names, srt = 45, adj = c(1.0,1.0), xpd = TRUE, cex=.6
axis(2)
```



3.2 Logistic Regression - Top 30 predictors

```r
#######################################################################
# Logistic Regression - Top 30 predictors
#######################################################################
## Model fitting
top_formula <- as.formula(paste("shares ~ ", f))
fit.lm.top <- glm(top_formula, data = news.train, family=binomial)
fit.lm.top
```

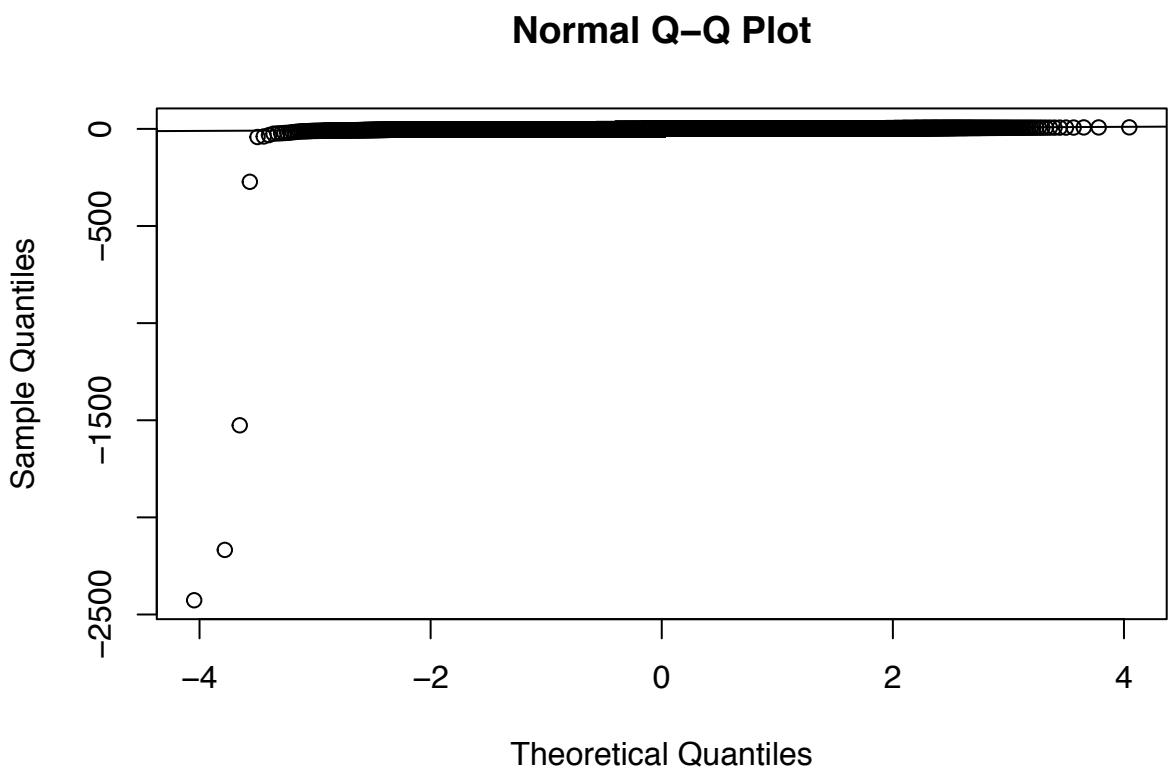
```
##
## Call:  glm(formula = top_formula, family = binomial, data = news.train)
##
## Coefficients:
##                   (Intercept)                     kw_avg_avg
##                    -1.543e+00                      7.768e-04
##    self_reference_avg_sharess      self_reference_min_shares
##                     3.427e-06                      8.323e-06
##                    kw_max_avg                     kw_min_avg
##                    -9.546e-05                     -9.537e-05
##                        LDA_02      self_reference_max_shares
##                    -8.971e-02                      3.543e-08
##         data_channel_is_world                     is_weekend
##                    -2.301e-01                      8.307e-01
## data_channel_is_entertainment                       num_imgs
##                    -4.775e-01                      7.344e-04
##                     timedelta                         LDA_04
##                     2.514e-04                      1.414e-01
##           global_subjectivity           data_channel_is_tech
##                     1.003e+00                      5.115e-01
##                     num_hrefs                     kw_avg_min
##                     6.993e-03                     -1.043e-04
##         min_positive_polarity                         LDA_01
##                    -5.516e-01                     -1.597e-01
##                        LDA_00                     kw_max_min
##                     8.292e-01                      1.360e-05
##     global_sentiment_polarity                         LDA_03
##                    -2.285e-01                             NA
##           rate_negative_words                     kw_min_max
##                    -2.100e-01                     -1.652e-08
##               n_unique_tokens     global_rate_positive_words
##                    -8.706e-01                     -1.132e+00
##      n_non_stop_unique_tokens            rate_positive_words
##                     2.407e-02                             NA
##                    kw_avg_max
##                    -1.922e-06
##
## Degrees of Freedom: 19207 Total (i.e. Null);  19179 Residual
## Null Deviance:       26620
## Residual Deviance: 24220     AIC: 24270
```

```r
summary(fit.lm.top)
```

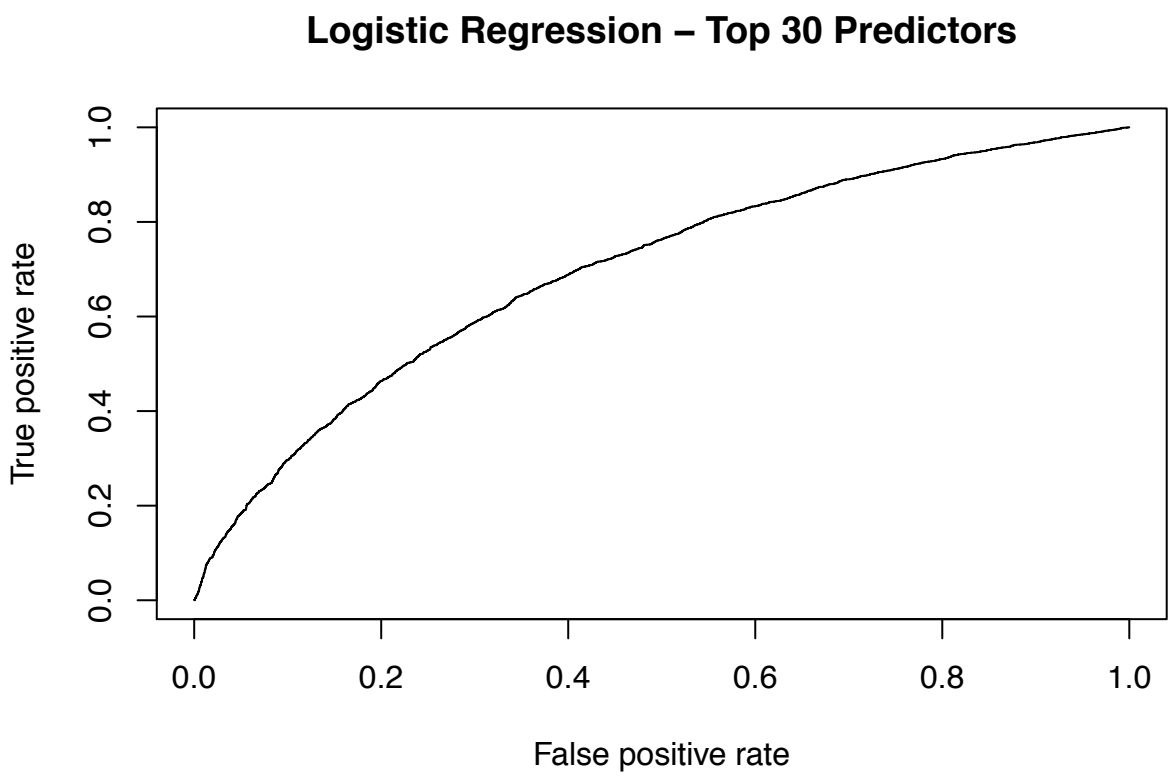
```
##
```

```
## Call:
## glm(formula = top_formula, family = binomial, data = news.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.9482  -1.0502  -0.6546   1.0767   2.0307
##
## Coefficients: (2 not defined because of singularities)
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   -1.543e+00  2.266e-01  -6.810 9.78e-12 ***
## kw_avg_avg                     7.768e-04  3.934e-05  19.746  < 2e-16 ***
## self_reference_avg_sharess     3.427e-06  3.543e-06   0.967  0.33339
## self_reference_min_shares      8.323e-06  2.999e-06   2.775  0.00552 **
## kw_max_avg                    -9.546e-05  6.793e-06 -14.053  < 2e-16 ***
## kw_min_avg                    -9.537e-05  1.985e-05  -4.805 1.55e-06 ***
## LDA_02                        -8.971e-02  1.198e-01  -0.749  0.45413
## self_reference_max_shares      3.543e-08  1.377e-06   0.026  0.97947
## data_channel_is_world         -2.301e-01  7.471e-02  -3.080  0.00207 **
## is_weekend                     8.307e-01  4.840e-02  17.162  < 2e-16 ***
## data_channel_is_entertainment -4.775e-01  5.949e-02  -8.026 1.00e-15 ***
## num_imgs                       7.344e-04  2.348e-03   0.313  0.75449
## timedelta                      2.514e-04  9.454e-05   2.659  0.00784 **
## LDA_04                         1.414e-01  1.013e-01   1.396  0.16264
## global_subjectivity            1.003e+00  2.027e-01   4.947 7.53e-07 ***
## data_channel_is_tech           5.115e-01  6.201e-02   8.248  < 2e-16 ***
## num_hrefs                      6.993e-03  1.647e-03   4.247 2.17e-05 ***
## kw_avg_min                    -1.043e-04  1.053e-04  -0.991  0.32177
## min_positive_polarity         -5.516e-01  2.617e-01  -2.108  0.03505 *
## LDA_01                        -1.597e-01  9.700e-02  -1.646  0.09977 .
## LDA_00                         8.292e-01  9.091e-02   9.121  < 2e-16 ***
## kw_max_min                     1.360e-05  1.708e-05   0.797  0.42572
## global_sentiment_polarity     -2.285e-01  2.847e-01  -0.803  0.42217
## LDA_03                                NA         NA      NA       NA
## rate_negative_words           -2.100e-01  1.767e-01  -1.189  0.23457
## kw_min_max                    -1.652e-08  3.147e-07  -0.052  0.95814
## n_unique_tokens               -8.706e-01  3.743e-01  -2.326  0.02004 *
## global_rate_positive_words    -1.132e+00  1.265e+00  -0.895  0.37086
## n_non_stop_unique_tokens       2.407e-02  3.787e-01   0.064  0.94931
## rate_positive_words                   NA         NA      NA       NA
## kw_avg_max                    -1.922e-06  1.847e-07 -10.404  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 26624  on 19207  degrees of freedom
## Residual deviance: 24215  on 19179  degrees of freedom
## AIC: 24273
##
## Number of Fisher Scoring iterations: 5
```

```r
## Q-Q plots
qqnorm(fit.lm.top$residuals)
qqline(fit.lm.top$residuals)
```




```
## ROC Curves
scores <- predict(fit.lm.top, newdata=news.validation, type = "response")
prob <- prediction(scores, labels=news.validation$shares)
perf <- performance(prob, "tpr", "fpr")
plot(perf, colorize=F, main="Logistic Regression - Top 30 Predictors")
```




```r
## AUC
auc.fit.lm.top <- unlist(attributes(performance(prob, "auc"))$y.values)
auc.fit.lm.top
```

```
## [1] 0.6960673
```

```r
## Confusion Matrix
pred.fit.lm.top <- ifelse(scores>0.5, 1, 0)
confusionMatrix(pred.fit.lm.top, news.validation$shares, positive = "1")
```

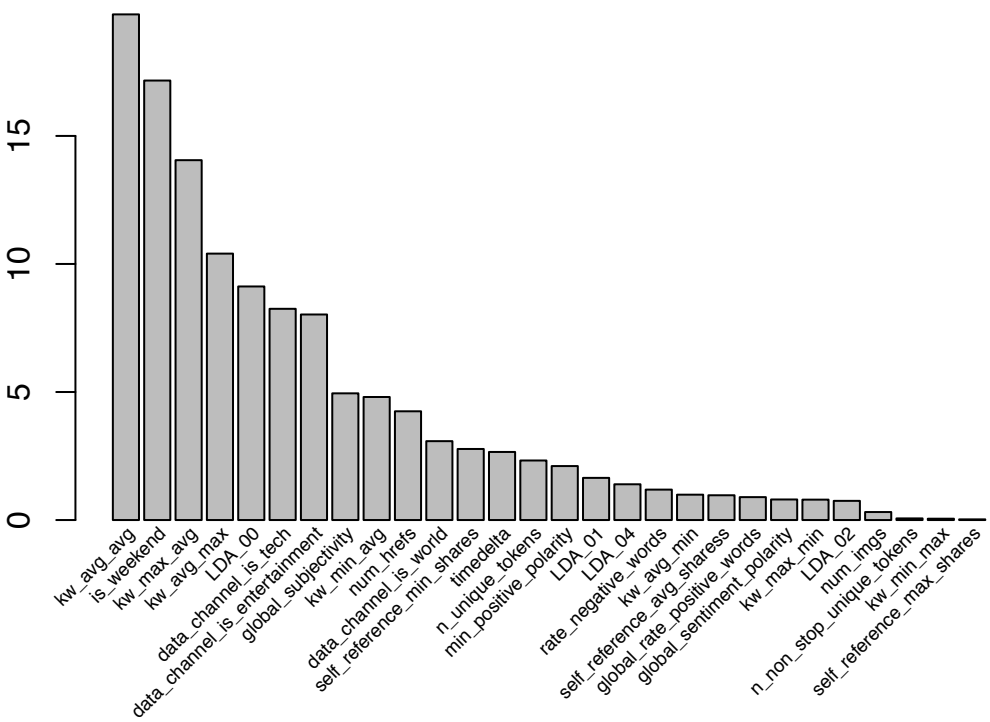
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 3299 1782
##          1 1667 2959
##
##                Accuracy : 0.6447
##                  95% CI : (0.6351, 0.6542)
##     No Information Rate : 0.5116
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.2886
##  Mcnemar's Test P-Value : 0.05224
##
##             Sensitivity : 0.6241
##             Specificity : 0.6643
##          Pos Pred Value : 0.6396
```

```
##          Neg Pred Value : 0.6493
##              Prevalence : 0.4884
##          Detection Rate : 0.3048
##    Detection Prevalence : 0.4766
##       Balanced Accuracy : 0.6442
##
##        'Positive' Class : 1
##
```

```
## Prediction Error on Validation Set
mse.fit.lm.top <- mean((news.validation$shares-pred.fit.lm.top)^2)
mse.fit.lm.top
```

```
## [1] 0.3553106
```

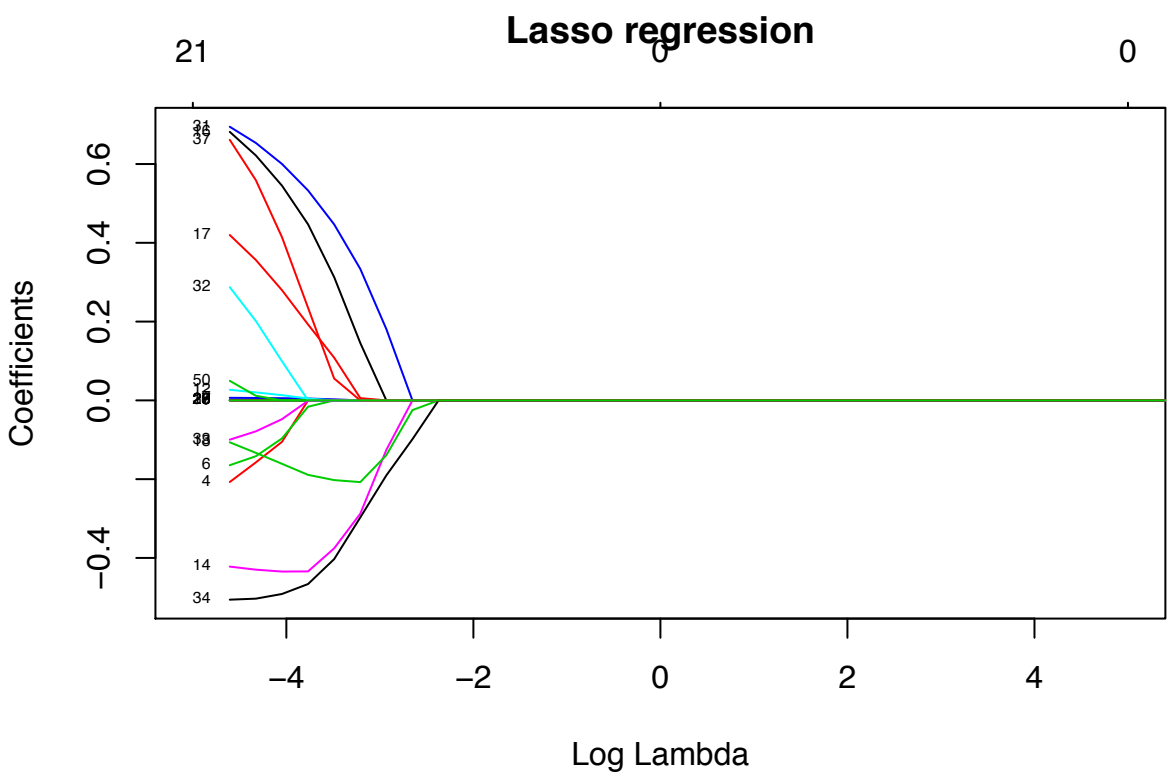
```
## Variable Importance
fit.lm.top.imp <- varImp(fit.lm.top)
fit.lm.top.imp$Names <- row.names(fit.lm.top.imp)
imp.values <- fit.lm.top.imp[order(fit.lm.top.imp$Overall, decreasing=TRUE)[1:30],1]
imp.names <- fit.lm.top.imp[order(fit.lm.top.imp$Overall, decreasing=TRUE)[1:30],2]
#barplot(imp.values, names = imp.names, las = 2)
imp.values.plot <- barplot(imp.values, axes = FALSE, axisnames = FALSE)
text(imp.values.plot, par("usr")[3], labels = imp.names, srt = 45, adj = c(1.0,1.0), xpd = TRUE, cex=.6
axis(2)
```



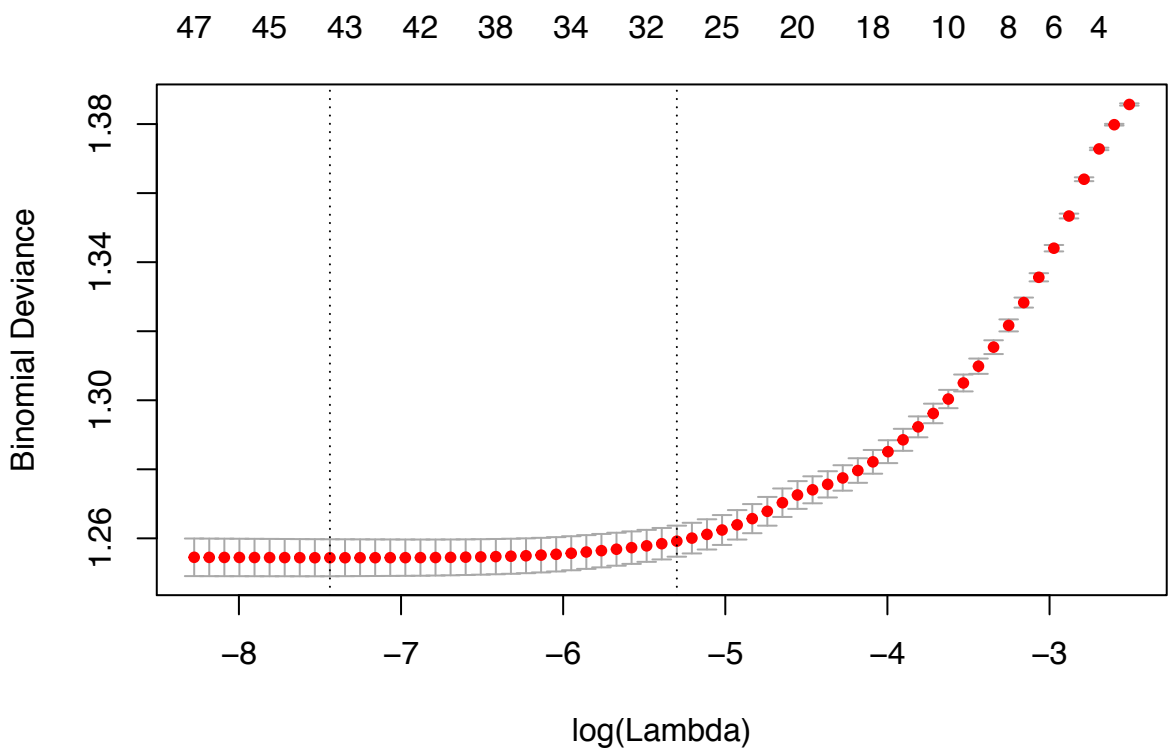
3.3 Regularization & Cross Validation

```
#######################################################################
# Lasso Regularization (CV)
#######################################################################
## Lasso Regularization
x.train = model.matrix(shares ~ ., news.train)[, -1]
y.train = news.train$shares
x.validation = model.matrix(shares ~ ., news.validation)[, -1]
y.validation = news.validation$shares

grid = 10 ^ seq(10, -2, length = 100)
fit.lasso <- glmnet(x.train, y.train, alpha = 1, lambda = grid,
                    family = "binomial")
plot(fit.lasso, main = "Lasso regression", label = TRUE,
     xvar = "lambda", xlim = c(-5, 5))
```




```
## Lasso Cross Validation
cv.lasso <- cv.glmnet(x.train, y.train, alpha = 1,
                      family = "binomial")
plot(cv.lasso)
```




```
bestlam.cv.lasso <- cv.lasso$lambda.min
bestlam.cv.lasso
```

```
## [1] 0.0005880631
```

```
log(bestlam.cv.lasso)
```

```
## [1] -7.438676
```

```
## Refit Lasso
refit.lasso <- glmnet(x.train, y.train, alpha=1, lambda =    bestlam.cv.lasso, family = "binomial")
refit.lasso
```

```
##
## Call:  glmnet(x = x.train, y = y.train, family = "binomial", alpha = 1,      lambda = bestlam.cv.las
##
##      Df    %Dev    Lambda
## [1,] 43 0.09939 0.0005881
```

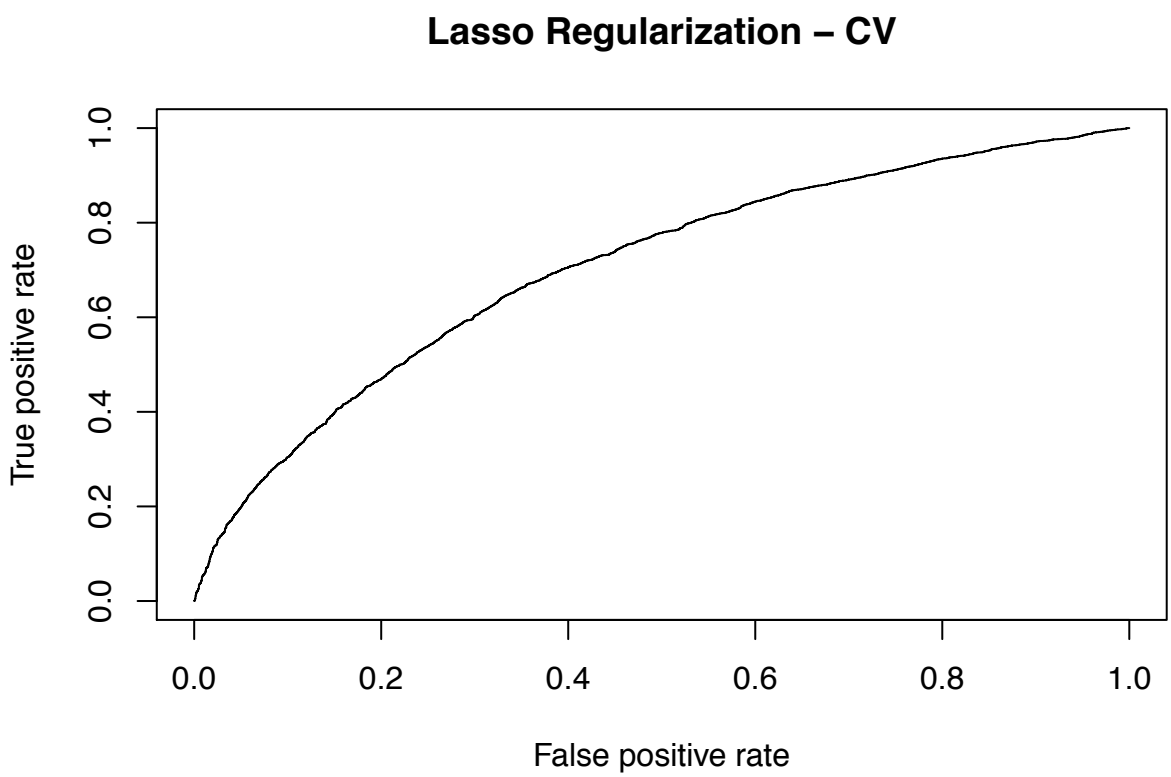
```
predict(refit.lasso, s = bestlam.cv.lasso, type = "coefficients")
```

```
## 53 x 1 sparse Matrix of class "dgCMatrix"
##                                          1
## (Intercept)                     7.179760e+06
## timedelta                       .
## n_tokens_title                 -5.974045e-03
## n_tokens_content                1.344055e-04
```

```
## n_unique_tokens               -6.590248e-01
## n_non_stop_words              -7.179762e+06
## n_non_stop_unique_tokens      -4.187563e-02
## num_hrefs                      8.375660e-03
## num_self_hrefs                -1.465164e-02
## num_imgs                       1.155154e-03
## num_videos                     .
## average_token_length          -4.981525e-02
## num_keywords                   3.926093e-02
## data_channel_is_lifestyle     -1.086634e-01
## data_channel_is_entertainment -3.396034e-01
## data_channel_is_bus           -1.919092e-01
## data_channel_is_socmed         7.344560e-01
## data_channel_is_tech           5.576524e-01
## data_channel_is_world         -6.524333e-02
## kw_min_min                     1.945710e-03
## kw_max_min                     .
## kw_avg_min                    -2.540582e-05
## kw_min_max                    -7.408960e-07
## kw_max_max                    -2.666595e-07
## kw_avg_max                    -5.363999e-07
## kw_min_avg                    -7.196807e-05
## kw_max_avg                    -8.268596e-05
## kw_avg_avg                     6.946667e-04
## self_reference_min_shares      8.267503e-06
## self_reference_max_shares      2.255161e-07
## self_reference_avg_sharess     2.367979e-06
## is_weekend                     8.257840e-01
## LDA_00                         1.040947e+00
## LDA_01                        -5.627209e-02
## LDA_02                        -1.500079e-01
## LDA_03                         .
## LDA_04                         3.254439e-01
## global_subjectivity            9.100514e-01
## global_sentiment_polarity      .
## global_rate_positive_words    -2.234479e+00
## global_rate_negative_words     2.054267e+00
## rate_positive_words            1.874300e-01
## rate_negative_words           -4.011371e-12
## avg_positive_polarity         -2.818491e-01
## min_positive_polarity         -3.597115e-01
## max_positive_polarity         -4.685337e-02
## avg_negative_polarity          .
## min_negative_polarity          .
## max_negative_polarity          .
## title_subjectivity             1.231620e-01
## title_sentiment_polarity       1.925281e-01
## abs_title_subjectivity         2.107707e-01
## abs_title_sentiment_polarity   .
```

```
## Q-Q plot
#qqnorm(refit.lasso$residuals)
#qqline(refit.lasso$residuals)
```

```r
## ROC Curves
scores <- predict(refit.lasso, newx=x.validation, type = "response")
prob <- prediction(scores, labels=news.validation$shares)
perf <- performance(prob, "tpr", "fpr")
plot(perf, colorize=F, main="Lasso Regularization - CV")
```




```r
## AUC
auc.refit.lasso <- unlist(attributes(performance(prob, "auc"))$y.values)
auc.refit.lasso
```

```
## [1] 0.7050061
```

```r
## Confusion Matrix
pred.refit.lasso <- ifelse(scores>0.5, 1, 0)
confusionMatrix(pred.refit.lasso, news.validation$shares, positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 3352 1737
##          1 1614 3004
##
##                Accuracy : 0.6548
##                  95% CI : (0.6452, 0.6642)
##     No Information Rate : 0.5116
##     P-Value [Acc > NIR] : < 2e-16
```

```
##
##                   Kappa : 0.3088
##  Mcnemar's Test P-Value : 0.03507
##
##             Sensitivity : 0.6336
##             Specificity : 0.6750
##          Pos Pred Value : 0.6505
##          Neg Pred Value : 0.6587
##              Prevalence : 0.4884
##          Detection Rate : 0.3095
##    Detection Prevalence : 0.4757
##       Balanced Accuracy : 0.6543
##
##        'Positive' Class : 1
##
```

```r
## Prediction Error on Validation Set
mse.refit.lasso <- mean((news.validation$shares-pred.refit.lasso)^2)
mse.refit.lasso
```

```
## [1] 0.3452148
```

```r
## Variable Importance
#refit.lasso.imp <- varImp(refit.lasso, lambda = bestlam.cv.lasso)
#refit.lasso.imp$Names <- row.names(refit.lasso.imp)
#imp.values <- refit.lasso.imp[order(refit.lasso.imp$Overall, decreasing=TRUE)[1:30],1]
#imp.names <- refit.lasso.imp[order(refit.lasso.imp$Overall, decreasing=TRUE)[1:30],2]
#barplot(imp.values, names = imp.names, las = 2)

#######################################################################
# Logistic Regression - Cross Validation
#######################################################################
## Model fitting
ctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 10,
                     savePredictions = TRUE)
fit.lm.cv <- train(shares ~ .,  data=news.train, method="glm", family="binomial",
                 trControl = ctrl, tuneLength = 5)
fit.lm.cv
```

```
## Generalized Linear Model
##
## 19208 samples
##    52 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 17288, 17287, 17287, 17287, 17288, 17287, ...
## Resampling results:
##
##   RMSE       Rsquared
##   0.4666881  0.1290228
##
##
```

```r
fit.lm.cv$finalModel
```

```
##
## Call:  NULL
##
## Coefficients:
##                   (Intercept)                      timedelta
##                     1.085e+07                      7.810e-06
##               n_tokens_title               n_tokens_content
##                    -6.616e-03                      1.374e-04
##              n_unique_tokens               n_non_stop_words
##                    -8.414e-01                     -1.085e+07
##      n_non_stop_unique_tokens                      num_hrefs
##                     2.460e-02                      8.720e-03
##               num_self_hrefs                       num_imgs
##                    -1.638e-02                      1.202e-03
##                    num_videos           average_token_length
##                     3.937e-04                     -5.618e-02
##                  num_keywords      data_channel_is_lifestyle
##                     3.707e-02                     -1.702e-01
## data_channel_is_entertainment            data_channel_is_bus
##                    -3.439e-01                     -2.356e-01
##        data_channel_is_socmed           data_channel_is_tech
##                     7.118e-01                      5.302e-01
##         data_channel_is_world                     kw_min_min
##                    -9.133e-02                      2.002e-03
##                    kw_max_min                     kw_avg_min
##                     2.678e-05                     -1.936e-04
##                    kw_min_max                     kw_max_max
##                    -7.426e-07                     -2.977e-07
##                    kw_avg_max                     kw_min_avg
##                    -6.787e-07                     -9.164e-05
##                    kw_max_avg                     kw_avg_avg
##                    -9.211e-05                      7.510e-04
##     self_reference_min_shares      self_reference_max_shares
##                     8.774e-06                      4.057e-07
##    self_reference_avg_sharess                     is_weekend
##                     2.134e-06                      8.400e-01
##                        LDA_00                         LDA_01
##                     7.456e-01                     -4.475e-01
##                        LDA_02                         LDA_03
##                    -4.856e-01                     -4.035e-01
##                        LDA_04            global_subjectivity
##                            NA                      9.007e-01
##     global_sentiment_polarity     global_rate_positive_words
##                     1.964e-01                     -4.346e+00
##    global_rate_negative_words            rate_positive_words
##                     6.052e+00                      4.516e-01
##           rate_negative_words          avg_positive_polarity
##                            NA                     -3.947e-01
##         min_positive_polarity          max_positive_polarity
##                    -4.098e-01                     -6.850e-02
##         avg_negative_polarity          min_negative_polarity
```

```
##                    -2.582e-02                    -1.249e-02
##         max_negative_polarity            title_subjectivity
##                     3.100e-02                     1.528e-01
##      title_sentiment_polarity         abs_title_subjectivity
##                     2.119e-01                     2.416e-01
##  abs_title_sentiment_polarity
##                    -3.109e-02
##
## Degrees of Freedom: 19207 Total (i.e. Null);  19157 Residual
## Null Deviance:       26620
## Residual Deviance: 23970     AIC: 24070
```

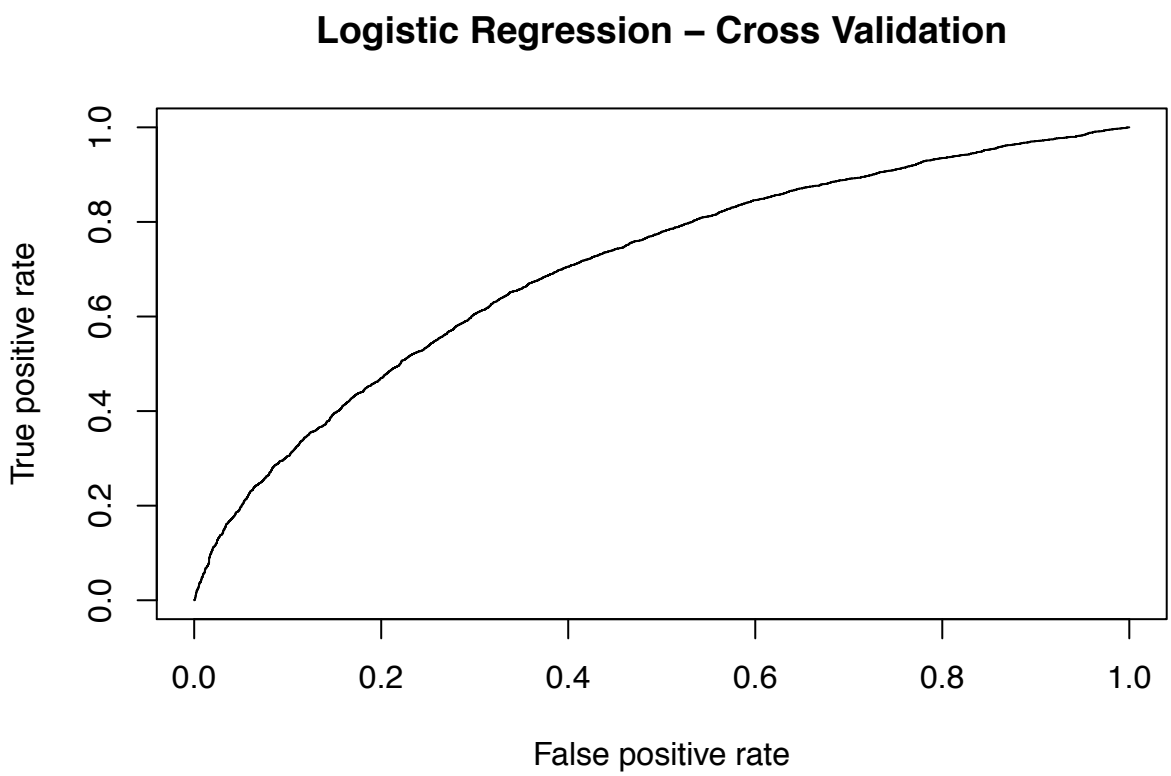
```r
summary(fit.lm.cv)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.9690  -1.0271  -0.6367   1.0634   2.1522
##
## Coefficients: (2 not defined because of singularities)
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    1.085e+07  7.559e+06   1.435 0.151231
## timedelta                      7.810e-06  1.052e-04   0.074 0.940816
## n_tokens_title                -6.616e-03  7.821e-03  -0.846 0.397622
## n_tokens_content               1.374e-04  6.513e-05   2.110 0.034836 *
## n_unique_tokens               -8.414e-01  6.155e-01  -1.367 0.171612
## n_non_stop_words              -1.085e+07  7.559e+06  -1.435 0.151231
## n_non_stop_unique_tokens       2.460e-02  4.503e-01   0.055 0.956430
## num_hrefs                      8.720e-03  1.901e-03   4.587 4.50e-06 ***
## num_self_hrefs                -1.638e-02  4.945e-03  -3.312 0.000926 ***
## num_imgs                       1.202e-03  2.507e-03   0.480 0.631417
## num_videos                     3.937e-04  4.110e-03   0.096 0.923688
## average_token_length          -5.618e-02  6.686e-02  -0.840 0.400787
## num_keywords                   3.707e-02  1.005e-02   3.689 0.000226 ***
## data_channel_is_lifestyle     -1.702e-01  1.061e-01  -1.604 0.108688
## data_channel_is_entertainment -3.439e-01  6.985e-02  -4.924 8.48e-07 ***
## data_channel_is_bus           -2.356e-01  1.031e-01  -2.284 0.022350 *
## data_channel_is_socmed         7.118e-01  1.034e-01   6.885 5.76e-12 ***
## data_channel_is_tech           5.302e-01  9.960e-02   5.324 1.02e-07 ***
## data_channel_is_world         -9.133e-02  1.017e-01  -0.898 0.369079
## kw_min_min                     2.002e-03  4.330e-04   4.623 3.78e-06 ***
## kw_max_min                     2.678e-05  1.475e-05   1.815 0.069521 .
## kw_avg_min                    -1.936e-04  9.267e-05  -2.090 0.036655 *
## kw_min_max                    -7.426e-07  3.212e-07  -2.312 0.020765 *
## kw_max_max                    -2.977e-07  1.574e-07  -1.892 0.058534 .
## kw_avg_max                    -6.787e-07  2.275e-07  -2.983 0.002850 **
## kw_min_avg                    -9.164e-05  2.099e-05  -4.366 1.26e-05 ***
## kw_max_avg                    -9.211e-05  6.936e-06 -13.280  < 2e-16 ***
## kw_avg_avg                     7.510e-04  4.168e-05  18.020  < 2e-16 ***
## self_reference_min_shares      8.774e-06  3.008e-06   2.917 0.003531 **
## self_reference_max_shares      4.057e-07  1.442e-06   0.281 0.778418
```

```
## self_reference_avg_sharess     2.134e-06  3.611e-06   0.591 0.554637
## is_weekend                     8.400e-01  4.876e-02  17.230  < 2e-16 ***
## LDA_00                         7.456e-01  1.214e-01   6.140 8.26e-10 ***
## LDA_01                        -4.475e-01  1.353e-01  -3.308 0.000938 ***
## LDA_02                        -4.856e-01  1.222e-01  -3.974 7.06e-05 ***
## LDA_03                        -4.035e-01  1.288e-01  -3.133 0.001733 **
## LDA_04                                NA         NA      NA       NA
## global_subjectivity            9.007e-01  2.271e-01   3.966 7.29e-05 ***
## global_sentiment_polarity      1.964e-01  4.399e-01   0.446 0.655319
## global_rate_positive_words    -4.346e+00  1.901e+00  -2.286 0.022242 *
## global_rate_negative_words     6.052e+00  3.683e+00   1.643 0.100353
## rate_positive_words            4.516e-01  3.011e-01   1.500 0.133642
## rate_negative_words                   NA         NA      NA       NA
## avg_positive_polarity         -3.947e-01  3.604e-01  -1.095 0.273445
## min_positive_polarity         -4.098e-01  3.093e-01  -1.325 0.185189
## max_positive_polarity         -6.850e-02  1.157e-01  -0.592 0.553861
## avg_negative_polarity         -2.582e-02  3.353e-01  -0.077 0.938616
## min_negative_polarity         -1.249e-02  1.233e-01  -0.101 0.919282
## max_negative_polarity          3.100e-02  2.771e-01   0.112 0.910919
## title_subjectivity             1.528e-01  7.392e-02   2.066 0.038789 *
## title_sentiment_polarity       2.119e-01  6.770e-02   3.130 0.001747 **
## abs_title_subjectivity         2.416e-01  9.847e-02   2.454 0.014143 *
## abs_title_sentiment_polarity  -3.109e-02  1.059e-01  -0.294 0.769071
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 26624  on 19207  degrees of freedom
## Residual deviance: 23970  on 19157  degrees of freedom
## AIC: 24072
##
## Number of Fisher Scoring iterations: 5
```

```r
## Q-Q plots
#qqnorm(fit.lm.cv$residuals)
#qqline(fit.lm.cv$residuals)

## ROC Curves
scores <- predict(fit.lm.cv$finalModel, newdata= news.validation, type = "response")
prob <- prediction(scores, labels=news.validation$shares)
perf <- performance(prob, "tpr", "fpr")
plot(perf, colorize=F, main="Logistic Regression - Cross Validation")
```




```
## AUC
auc.fit.lm.cv <- unlist(attributes(performance(prob, "auc"))$y.values)
auc.fit.lm.cv
```

```
## [1] 0.7049882
```

```
## Confusion Matrix
pred.fit.lm.cv <- ifelse(scores>0.5, 1, 0)
confusionMatrix(pred.fit.lm.cv, news.validation$shares, positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 3353 1732
##          1 1613 3009
##
##                Accuracy : 0.6554
##                  95% CI : (0.6459, 0.6649)
##     No Information Rate : 0.5116
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.31
##  Mcnemar's Test P-Value : 0.04133
##
##             Sensitivity : 0.6347
##             Specificity : 0.6752
##          Pos Pred Value : 0.6510
```

```
##          Neg Pred Value : 0.6594
##              Prevalence : 0.4884
##          Detection Rate : 0.3100
##    Detection Prevalence : 0.4762
##       Balanced Accuracy : 0.6549
##
##        'Positive' Class : 1
##
```

```
## Prediction Error on Validation Set
mse.fit.lm.cv = mean((news.validation$shares-pred.fit.lm.cv)^2)
mse.fit.lm.cv
```

```
## [1] 0.3445967
```

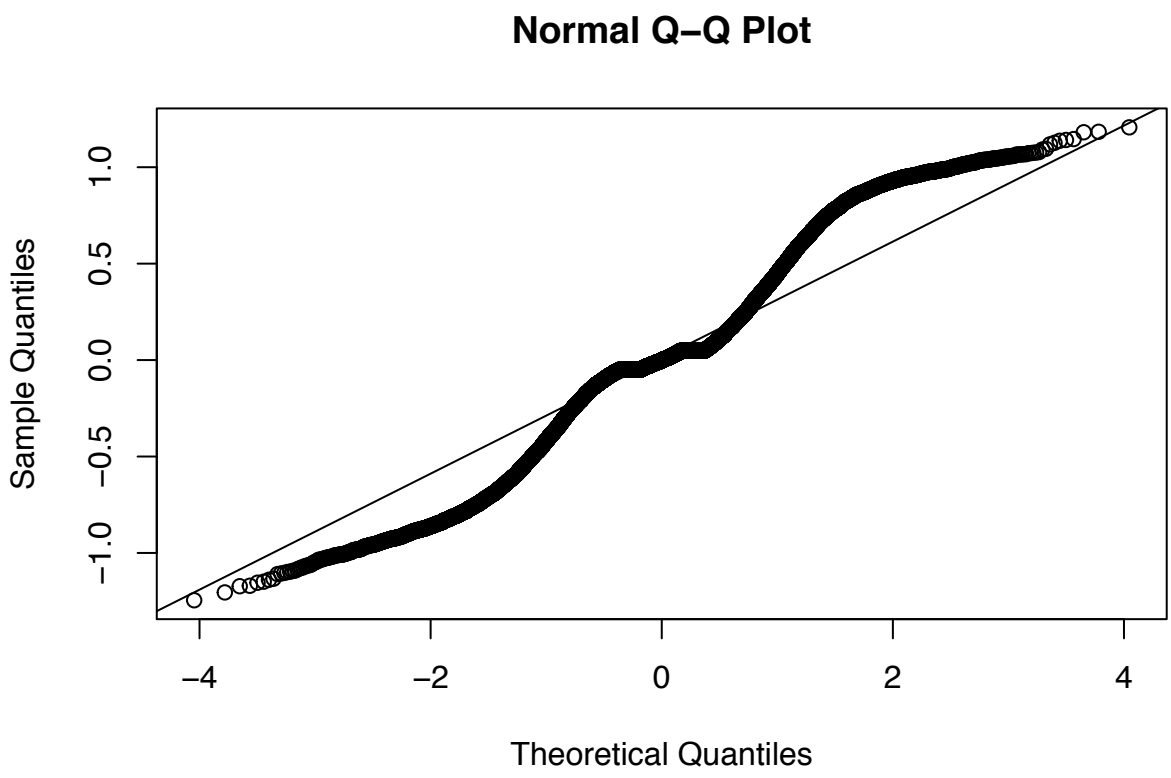
3.4 SVM - All 52 predictors

```
#######################################################################
# Support Vector Machines - All 52 predictors
#######################################################################
## Model fitting
fit.svm.all <- svm(shares ~ ., data = news.train, scale = TRUE)
summary(fit.svm.all)
```

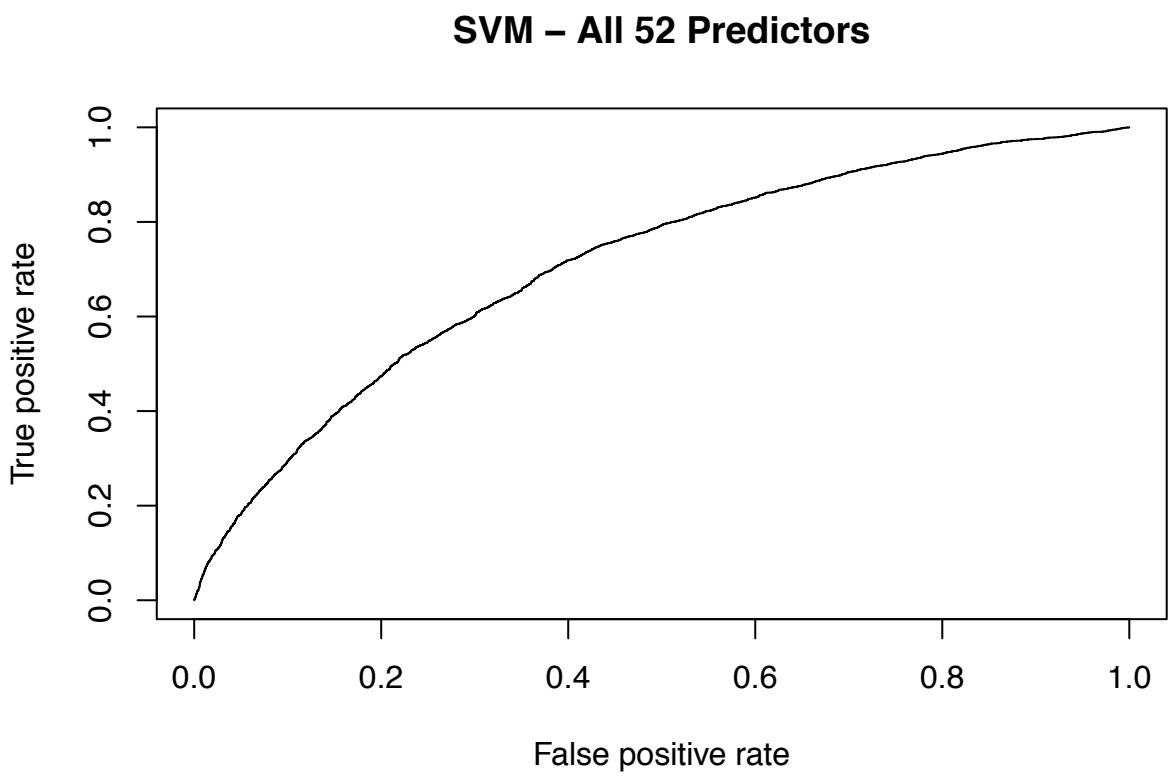
```
##
## Call:
## svm(formula = shares ~ ., data = news.train, scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  radial
##        cost:  1
##       gamma:  0.01923077
##     epsilon:  0.1
##
##
## Number of Support Vectors:  16431
```

```
#plot(fit.svm.all)

## Q-Q plots
qqnorm(fit.svm.all$residuals)
qqline(fit.svm.all$residuals)
```




```
## ROC Curves
scores <- predict(fit.svm.all, newdata=news.validation, type = "prob")
prob <- prediction(scores, labels=news.validation$shares)
perf <- performance(prob, "tpr", "fpr")
plot(perf, colorize=F, main="SVM - All 52 Predictors")
```




```
## AUC
auc.fit.svm.all <- unlist(attributes(performance(prob, "auc"))$y.values)
auc.fit.svm.all
```

```
## [1] 0.7100042
```

```
## Confusion Matrix
pred.fit.svm.all <- ifelse(scores>0.5, 1, 0)
confusionMatrix(pred.fit.svm.all, news.validation$shares, positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 3298 1703
##          1 1668 3038
##
##                Accuracy : 0.6527
##                  95% CI : (0.6432, 0.6622)
##     No Information Rate : 0.5116
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.305
##  Mcnemar's Test P-Value : 0.5581
##
##             Sensitivity : 0.6408
##             Specificity : 0.6641
##          Pos Pred Value : 0.6456
```

```
##          Neg Pred Value : 0.6595
##              Prevalence : 0.4884
##          Detection Rate : 0.3130
##    Detection Prevalence : 0.4848
##       Balanced Accuracy : 0.6525
##
##        'Positive' Class : 1
##
```

```
## Prediction Error on Validation Set
mse.fit.svm.all <- mean((news.validation$shares-pred.fit.svm.all)^2)
mse.fit.svm.all
```

```
## [1] 0.3472752
```

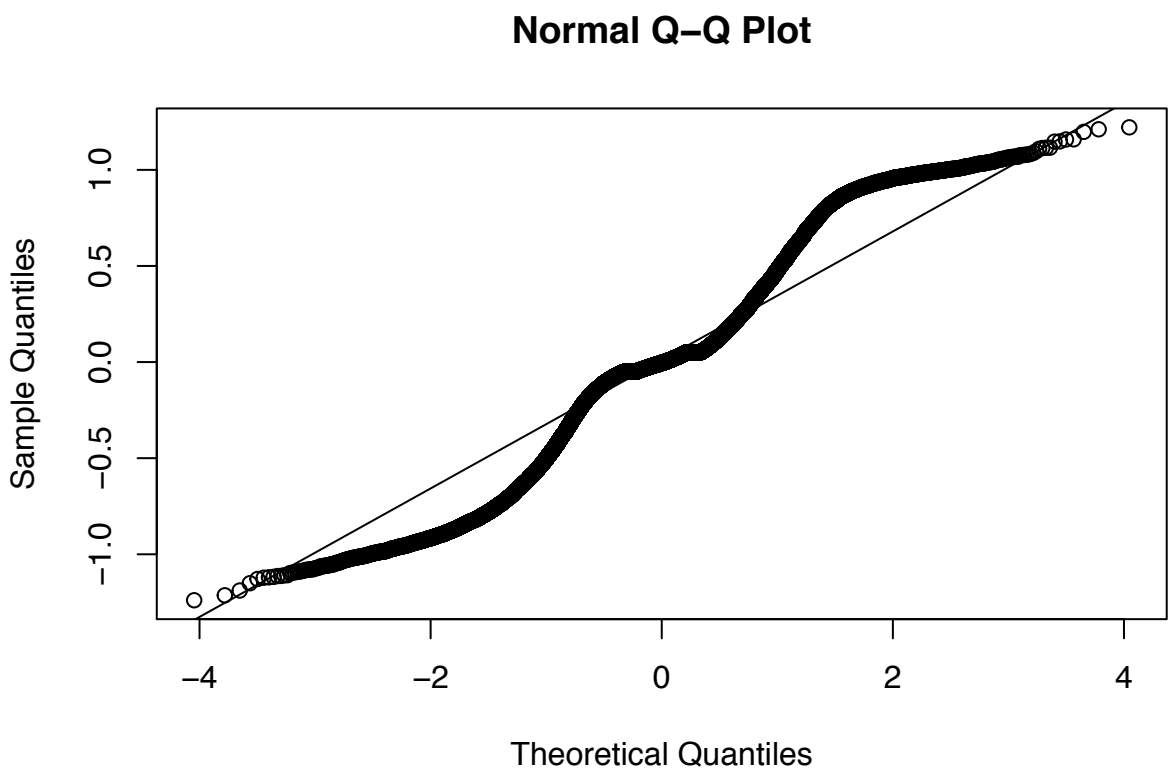
```
## Variable Importance
#fit.svm.all.imp <- varImp(fit.svm.all)
#fit.svm.all.imp$Names <- row.names(fit.svm.all.imp)
#imp.values <- fit.svm.all.imp[order(fit.svm.all.imp$Overall, decreasing=TRUE)[1:30],1]
#imp.names <- fit.svm.all.imp[order(fit.svm.all.imp$Overall, decreasing=TRUE)[1:30],2]
#barplot(imp.values, names = imp.names, las = 2)
```

3.5 SVM - Top 30 predictors

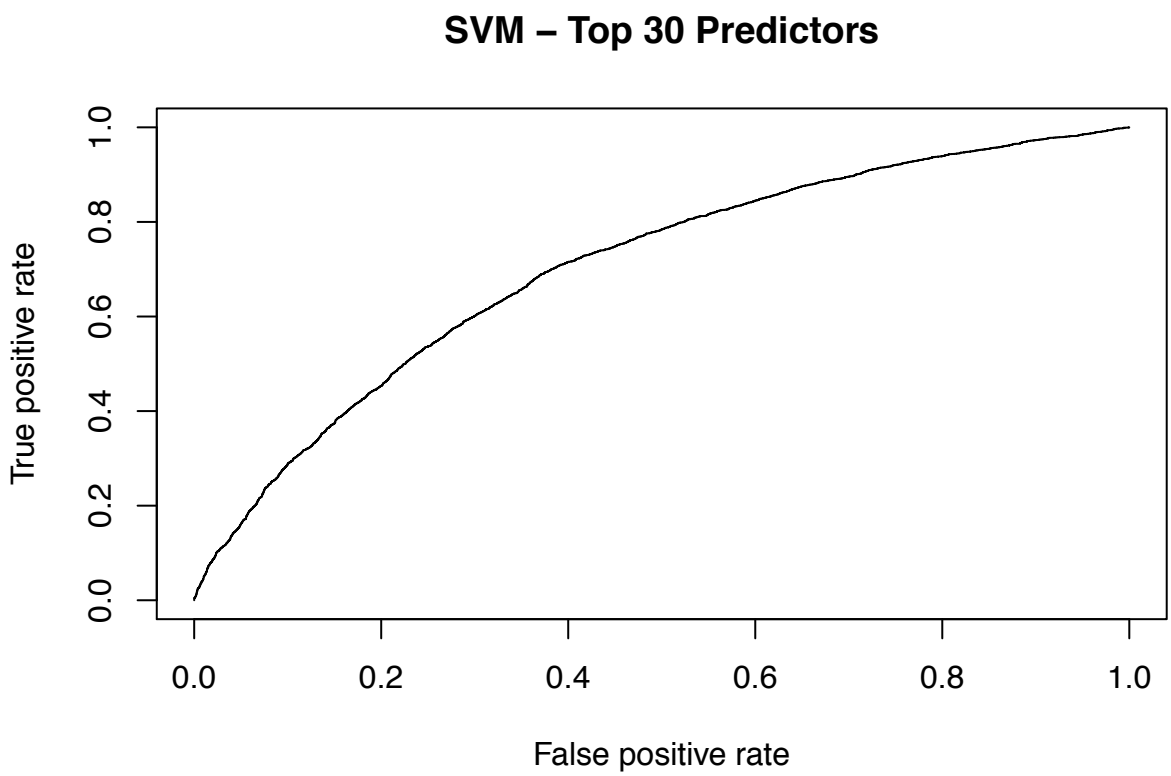
```
#######################################################################
# Support Vector Machines - Top 30 predictors
#######################################################################
## Model fitting
top_formula <- as.formula(paste("shares ~ ", f))
fit.svm.top <- svm(top_formula, data = news.train, scale = TRUE)
summary(fit.svm.top)
```

```
##
## Call:
## svm(formula = top_formula, data = news.train, scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  radial
##        cost:  1
##       gamma:  0.03333333
##     epsilon:  0.1
##
##
## Number of Support Vectors:  15876
```

```
## Q-Q plots
qqnorm(fit.svm.top$residuals)
qqline(fit.svm.top$residuals)
```




```
## ROC Curves
scores <- predict(fit.svm.top, newdata=news.validation, type = "prob")
prob <- prediction(scores, labels=news.validation$shares)
perf <- performance(prob, "tpr", "fpr")
plot(perf, colorize=F, main="SVM - Top 30 Predictors")
```




```
## AUC
auc.fit.svm.top <- unlist(attributes(performance(prob, "auc"))$y.values)
auc.fit.svm.top
```

```
## [1] 0.7026149
```

```
## Confusion Matrix
pred.fit.svm.top <- ifelse(scores>0.5, 1, 0)
confusionMatrix(pred.fit.svm.top, news.validation$shares, positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 3241 1635
##          1 1725 3106
##
##                Accuracy : 0.6539
##                  95% CI : (0.6443, 0.6633)
##     No Information Rate : 0.5116
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.3076
##  Mcnemar's Test P-Value : 0.1247
##
##             Sensitivity : 0.6551
##             Specificity : 0.6526
##          Pos Pred Value : 0.6429
```

```
##          Neg Pred Value : 0.6647
##              Prevalence : 0.4884
##          Detection Rate : 0.3200
##    Detection Prevalence : 0.4977
##       Balanced Accuracy : 0.6539
##
##        'Positive' Class : 1
##
```

```
## Prediction Error on Validation Set
mse.fit.svm.top <- mean((news.validation$shares-pred.fit.svm.top)^2)
mse.fit.svm.top
```

```
## [1] 0.346142
```

```
## Variable Importance
#fit.svm.top.imp <- varImp(fit.svm.top)
#fit.svm.top.imp$Names <- row.names(fit.svm.top.imp)
#imp.values <- fit.svm.top.imp[order(fit.svm.top.imp$Overall, decreasing=TRUE)[1:30],1]
#imp.names <- fit.svm.top.imp[order(fit.svm.top.imp$Overall, decreasing=TRUE)[1:30],2]
#barplot(imp.values, names = imp.names, las = 2)
```

3.6 SVM - Cross-Validation

```
#######################################################################
# SVM - Cross Validation
#######################################################################
## Model fitting
control <- trainControl(method="repeatedcv", number = 5, savePredictions = TRUE)
fit.svm.cv <- train(news.train[, -53], as.factor(news.train$shares), method="lssvmRadial", trControl=co
fit.svm.cv.final <- fit.svm.cv$finalModel
fit.svm.cv.final
```

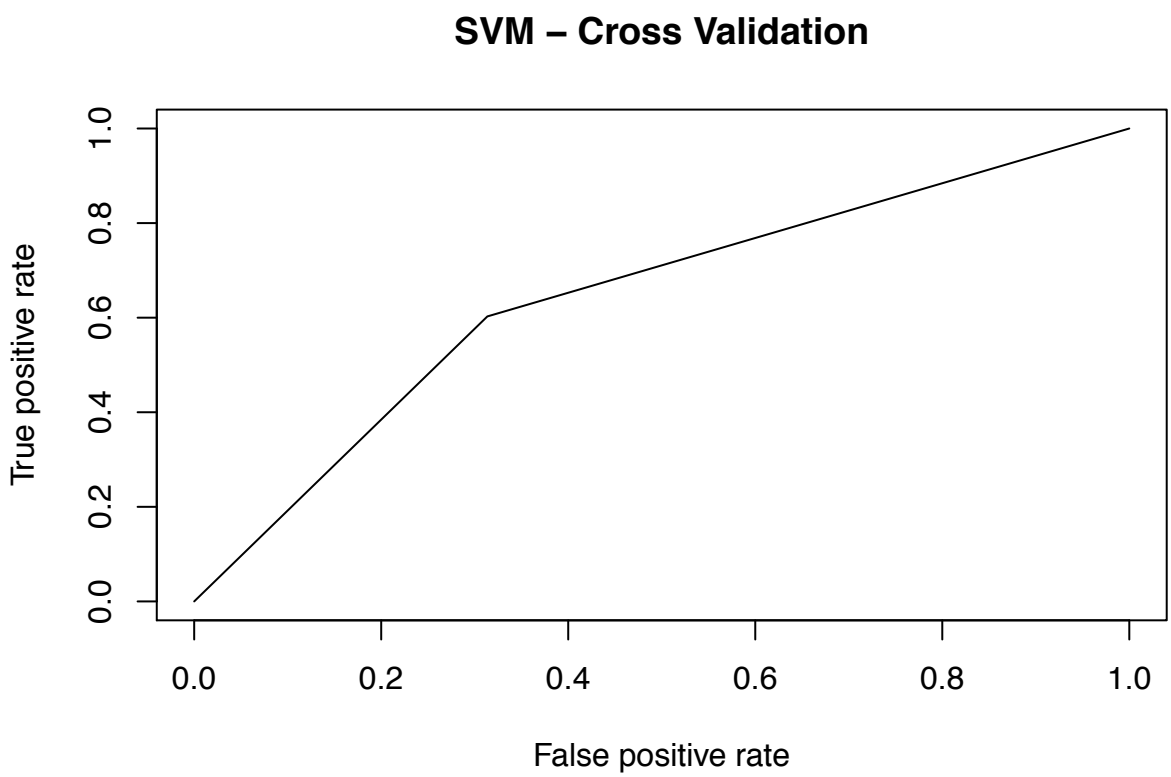
```
## Least Squares Support Vector Machine object of class "lssvm"
##
## problem type : classification
##  parameter : tau = 0.5
##
## Gaussian Radial Basis kernel function.
##  Hyperparameter : sigma =  0.00660491182045035
##
## Number of data points used for training : 32
## Training error : 0.348605
```

```
summary(fit.svm.cv.final)
```

```
## Length  Class   Mode
##      1  lssvm     S4
```

```r
## Q-Q plots
#qqnorm(fit.svm.cv.final$residuals)
#qqline(fit.svm.cv.final$residuals)

## ROC Curves
scores <- predict(fit.svm.cv.final, newdata=news.validation[, 1:52])
#scores <- caret::predict.train(fit.svm.cv.final, news.validation)
prob <- prediction(as.numeric(scores), labels=news.validation$shares)
perf <- performance(prob, "tpr", "fpr")
plot(perf, colorize=F, main="SVM - Cross Validation")
```




```r
## AUC
auc.fit.svm.cv <- unlist(attributes(performance(prob, "auc"))$y.values)
auc.fit.svm.cv
```

```
## [1] 0.6445417
```

```r
## Confusion Matrix
pred.fit.svm.cv <- ifelse(as.numeric(scores)>1, 1, 0)
confusionMatrix(as.numeric(pred.fit.svm.cv), news.validation$shares, positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 3409 1884
##          1 1557 2857
```

```
##
##                Accuracy : 0.6455
##                  95% CI : (0.6359, 0.655)
##     No Information Rate : 0.5116
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.2895
##  Mcnemar's Test P-Value : 2.738e-08
##
##             Sensitivity : 0.6026
##             Specificity : 0.6865
##          Pos Pred Value : 0.6473
##          Neg Pred Value : 0.6441
##              Prevalence : 0.4884
##          Detection Rate : 0.2943
##    Detection Prevalence : 0.4547
##       Balanced Accuracy : 0.6445
##
##        'Positive' Class : 1
##
```

```
## Prediction Error on Validation Set
mse.fit.svm.cv <- mean((news.validation$shares-pred.fit.svm.cv)^2)
mse.fit.svm.cv
```

```
## [1] 0.3544865
```

3.7 Random Forest - All 52 Predictors

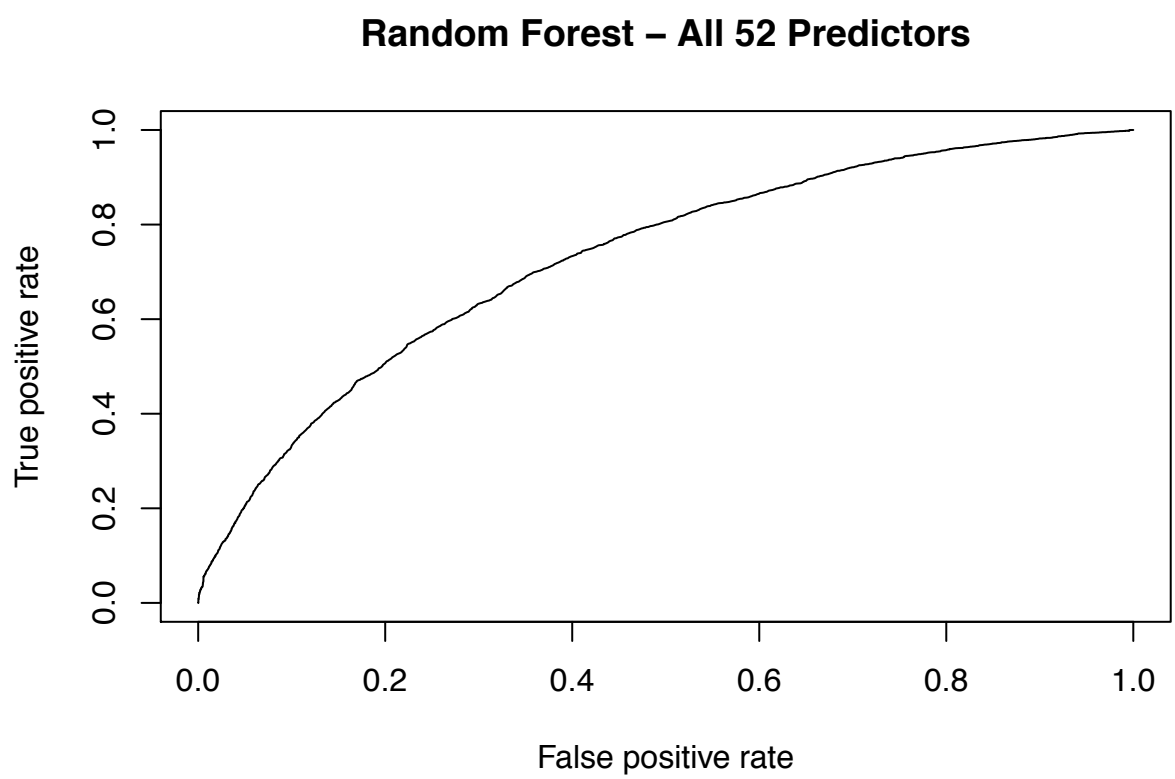
```
#######################################################################
# Random Forest - All 52 Predictors
#######################################################################
## Model fitting
fit.rf.1800 <- randomForest(x = news.train[,1:52], y = as.factor(news.train[,53]), data = news.train, s
fit.rf.1800
```

```
##
## Call:
##  randomForest(x = news.train[, 1:52], y = as.factor(news.train[,      53]), ntree = 1800, mtry = 8, 
##                Type of random forest: classification
##                      Number of trees: 1800
## No. of variables tried at each split: 8
##
##         OOB estimate of  error rate: 33.11%
## Confusion matrix:
##      0    1 class.error
## 0 6544 3206   0.3288205
## 1 3154 6304   0.3334743
```

```r
summary(fit.rf.1800)
```

```
##                 Length Class  Mode
## call                8  -none- call
## type                1  -none- character
## predicted       19208  factor numeric
## err.rate         5400  -none- numeric
## confusion           6  -none- numeric
## votes           38416  matrix numeric
## oob.times       19208  -none- numeric
## classes             2  -none- character
## importance        208  -none- numeric
## importanceSD      156  -none- numeric
## localImportance     0  -none- NULL
## proximity           0  -none- NULL
## ntree               1  -none- numeric
## mtry                1  -none- numeric
## forest             14  -none- list
## y               19208  factor numeric
## test                0  -none- NULL
## inbag               0  -none- NULL
```

```r
## ROC Curves
scores <- predict(fit.rf.1800, newdata= news.validation, type = "prob")[, 2]
prob <- prediction(scores, labels=news.validation$shares)
perf <- performance(prob, "tpr", "fpr")
plot(perf, colorize=F, main="Random Forest - All 52 Predictors")
```



```r
## AUC
auc.fit.rf.1800 <- unlist(attributes(performance(prob, "auc"))$y.values)
auc.fit.rf.1800
```

```
## [1] 0.7286781
```

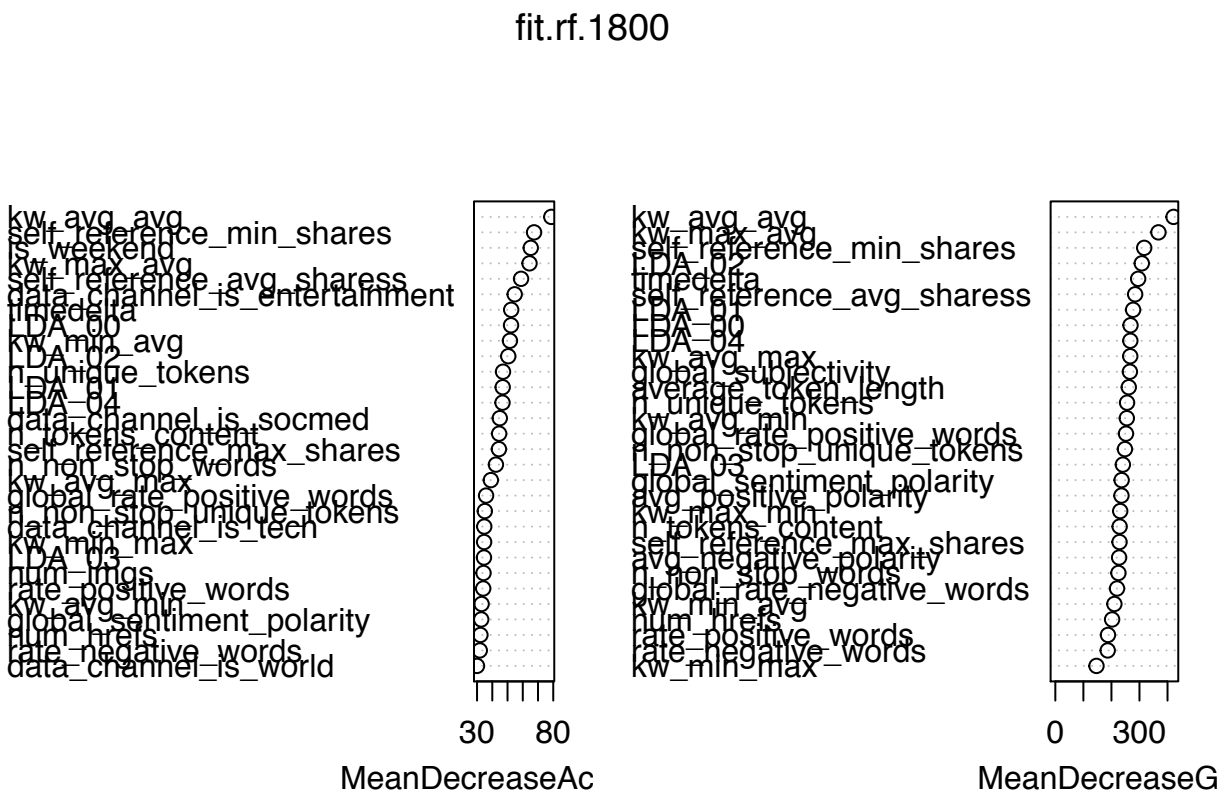
```r
## Confusion Matrix
pred.fit.rf.1800 <- ifelse(scores>0.5, 1, 0)
confusionMatrix(pred.fit.rf.1800, news.validation$shares, positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 3332 1589
##          1 1634 3152
##
##                Accuracy : 0.668
##                  95% CI : (0.6585, 0.6773)
##     No Information Rate : 0.5116
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.3357
##  Mcnemar's Test P-Value : 0.4383
##
##             Sensitivity : 0.6648
##             Specificity : 0.6710
##          Pos Pred Value : 0.6586
##          Neg Pred Value : 0.6771
##              Prevalence : 0.4884
##          Detection Rate : 0.3247
##    Detection Prevalence : 0.4930
##       Balanced Accuracy : 0.6679
##
##        'Positive' Class : 1
##
```

```r
## Prediction Error on Validation Set
mse.fit.rf.1800 <- mean((news.validation$shares-pred.fit.rf.1800)^2)
mse.fit.rf.1800
```

```
## [1] 0.3320284
```

```r
## Variable Importance
varImpPlot(fit.rf.1800, sort=TRUE, n.var = 30)
```



3.8 Random Forest - Top 30 Predictors

```
#######################################################################
# Random Forest - Top 30 Predictors
#######################################################################
## Model fitting
top_formula <- as.formula(paste("as.factor(shares) ~ ", f))
fit.rf.top <- randomForest(top_formula, data = news.train, scale = TRUE, ntree =1800, mtry = 8, importa
fit.rf.top
```

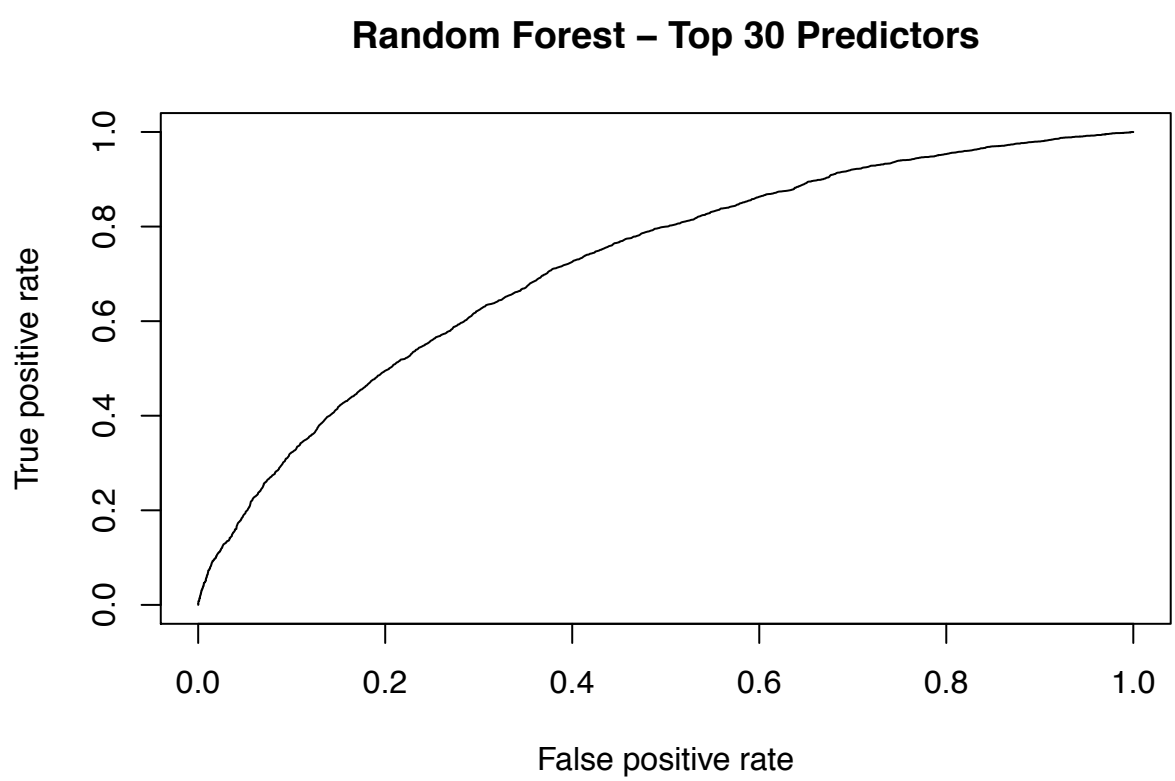
```
##
## Call:
##  randomForest(formula = top_formula, data = news.train, scale = TRUE,      ntree = 1800, mtry = 8, i
##                Type of random forest: classification
##                      Number of trees: 1800
## No. of variables tried at each split: 8
##
##         OOB estimate of  error rate: 33.53%
## Confusion matrix:
##      0    1 class.error
## 0 6481 3269   0.3352821
## 1 3172 6286   0.3353775
```

```
summary(fit.rf.top)
```

```
##                  Length Class  Mode
```

```
## call                7  -none- call
## type                1  -none- character
## predicted       19208  factor numeric
## err.rate         5400  -none- numeric
## confusion           6  -none- numeric
## votes           38416  matrix numeric
## oob.times       19208  -none- numeric
## classes             2  -none- character
## importance        120  -none- numeric
## importanceSD       90  -none- numeric
## localImportance     0  -none- NULL
## proximity           0  -none- NULL
## ntree               1  -none- numeric
## mtry                1  -none- numeric
## forest             14  -none- list
## y               19208  factor numeric
## test                0  -none- NULL
## inbag               0  -none- NULL
## terms               3  terms  call
```

```
## ROC Curves
scores <- predict(fit.rf.top, newdata= news.validation, type = "prob")[, 2]
prob <- prediction(scores, labels=news.validation$shares)
perf <- performance(prob, "tpr", "fpr")
plot(perf, colorize=F, main="Random Forest - Top 30 Predictors")
```




```
## AUC
auc.fit.rf.top <- unlist(attributes(performance(prob, "auc"))$y.values)
auc.fit.rf.top
```

```
## [1] 0.7217758
```

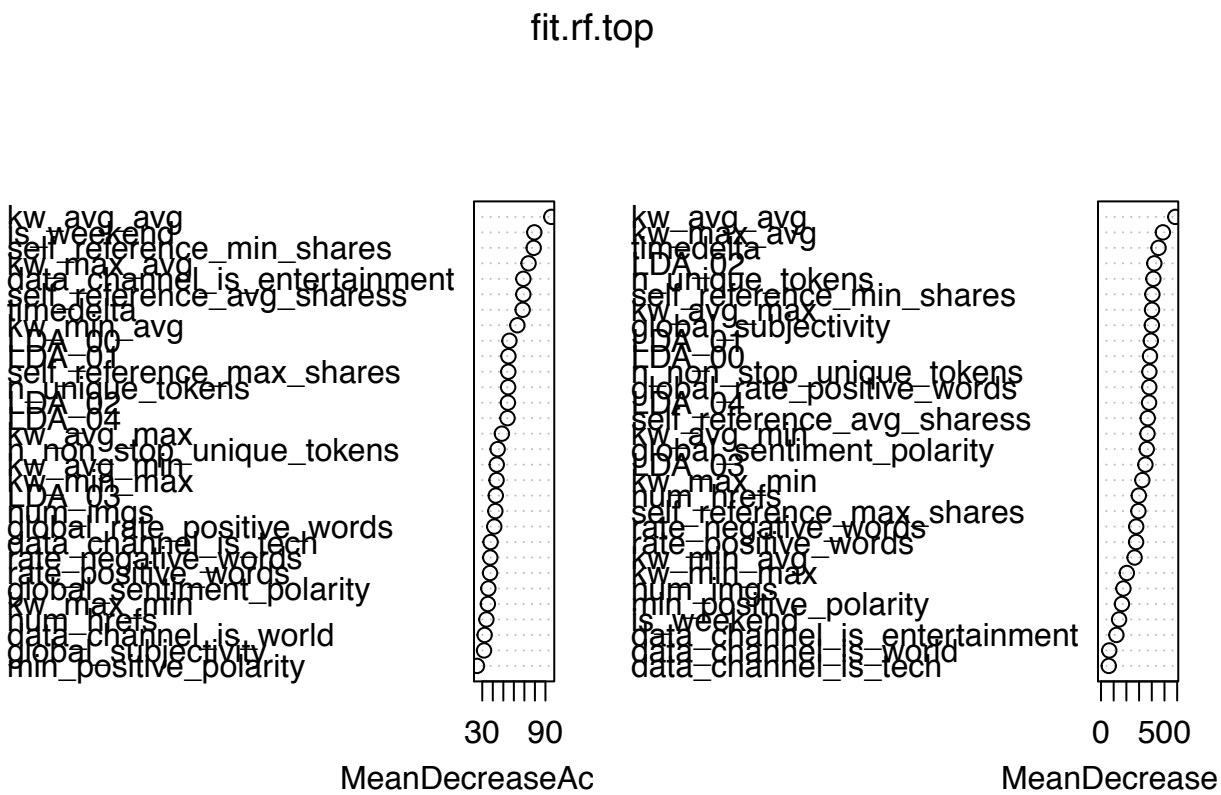
```r
## Confusion Matrix
pred.fit.rf.top <- ifelse(scores>0.5, 1, 0)
confusionMatrix(pred.fit.rf.top, news.validation$shares, positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 3287 1613
##          1 1679 3128
##
##                Accuracy : 0.6609
##                  95% CI : (0.6513, 0.6703)
##     No Information Rate : 0.5116
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.3216
##  Mcnemar's Test P-Value : 0.2573
##
##             Sensitivity : 0.6598
##             Specificity : 0.6619
##          Pos Pred Value : 0.6507
##          Neg Pred Value : 0.6708
##              Prevalence : 0.4884
##          Detection Rate : 0.3222
##    Detection Prevalence : 0.4952
##       Balanced Accuracy : 0.6608
##
##        'Positive' Class : 1
##
```

```r
## Prediction Error on Validation Set
mse.fit.rf.top <- mean((news.validation$shares-pred.fit.rf.top)^2)
mse.fit.rf.top
```

```
## [1] 0.3391367
```

```r
## Variable Importance
varImpPlot(fit.rf.top, sort=TRUE, n.var = 30)
```



3.9 Random Forest - Channel as Categorical

```
#######################################################################
# Random Forest - Channel as Categorical
#######################################################################
## Pre-processing
news.data.cat <- news.data
news.data.cat$data_channel_is_other <- ifelse(news.data$data_channel_is_bus == 0 & news.data$data_channe

sample.index <- sample(1:3, size=nrow(news.data.cat), replace=TRUE, prob=c(0.5,0.25,0.25))
news.train.cat <- news.data.cat[sample.index == 1, ]
news.validation.cat <- news.data.cat[sample.index == 2, ]
news.test.cat <- news.data.cat[sample.index == 3, ]

## Model Fitting
fit.rf.cat <- randomForest(x = news.train.cat[, c(1:52, 54)], y = as.factor(news.train.cat[,53]), data =
fit.rf.cat
```

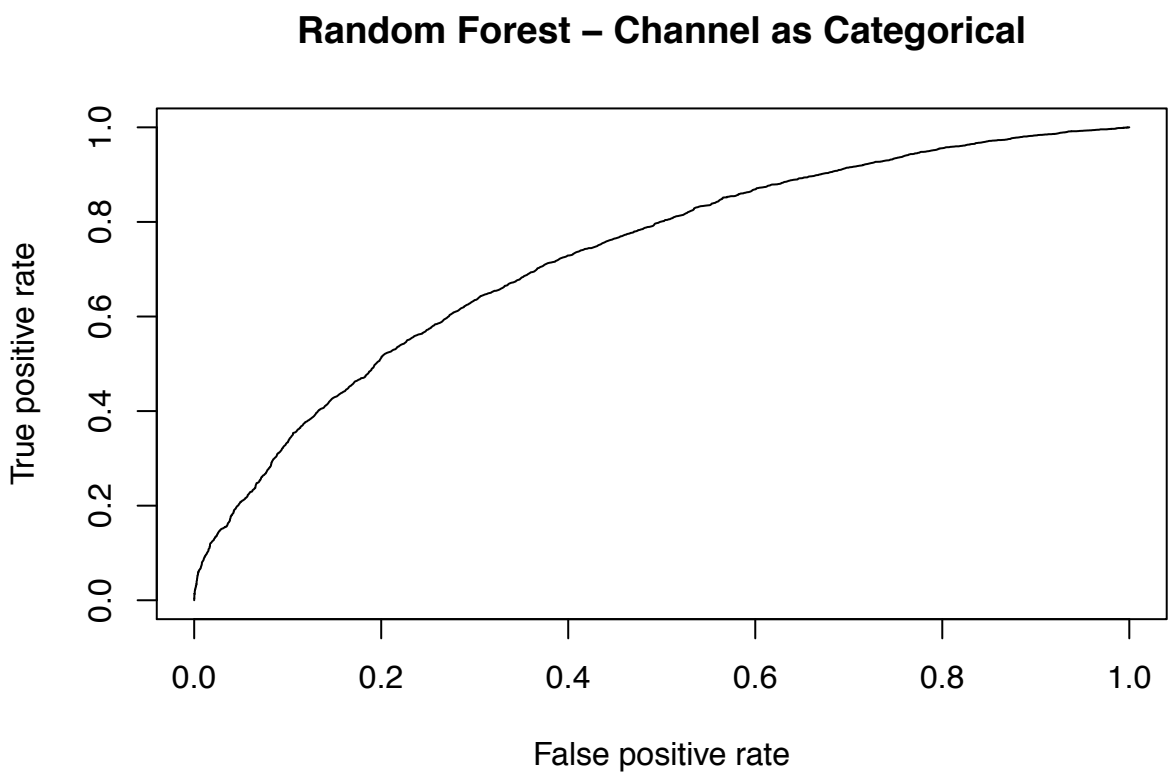
```
##
## Call:
##  randomForest(x = news.train.cat[, c(1:52, 54)], y = as.factor(news.train.cat[,      53]), ntree = 18
##                Type of random forest: classification
##                      Number of trees: 1800
## No. of variables tried at each split: 8
##
##         OOB estimate of  error rate: 33.26%
```

```
## Confusion matrix:
##      0    1 class.error
## 0 6489 3210   0.3309620
## 1 3131 6233   0.3343657
```

```r
summary(fit.rf.cat)
```

```
##                 Length Class  Mode
## call                8  -none- call
## type                1  -none- character
## predicted       19063  factor numeric
## err.rate         5400  -none- numeric
## confusion           6  -none- numeric
## votes           38126  matrix numeric
## oob.times       19063  -none- numeric
## classes             2  -none- character
## importance        212  -none- numeric
## importanceSD      159  -none- numeric
## localImportance     0  -none- NULL
## proximity           0  -none- NULL
## ntree               1  -none- numeric
## mtry                1  -none- numeric
## forest             14  -none- list
## y               19063  factor numeric
## test                0  -none- NULL
## inbag               0  -none- NULL
```

```r
## ROC Curves
scores <- predict(fit.rf.cat, newdata= news.validation.cat, type = "prob")[, 2]
prob <- prediction(scores, labels=news.validation.cat$shares)
perf <- performance(prob, "tpr", "fpr")
plot(perf, colorize=F, main="Random Forest - Channel as Categorical")
```




```r
## AUC
auc.fit.rf.cat <- unlist(attributes(performance(prob, "auc"))$y.values)
auc.fit.rf.cat
```

```
## [1] 0.7271816
```

```r
## Confusion Matrix
pred.fit.rf.cat <- ifelse(scores>0.5, 1, 0)
confusionMatrix(pred.fit.rf.cat, news.validation.cat$shares, positive = "1")
```

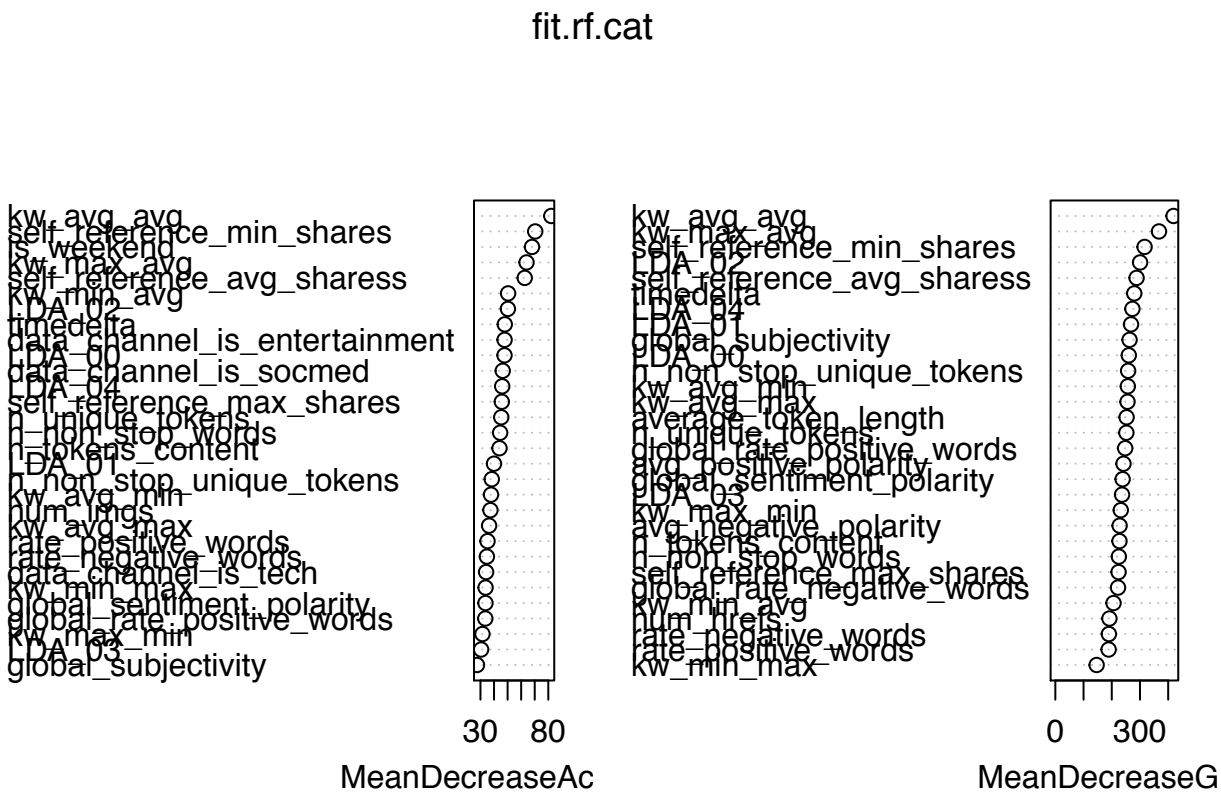
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 3319 1599
##          1 1649 3163
##
##                Accuracy : 0.6662
##                  95% CI : (0.6567, 0.6756)
##     No Information Rate : 0.5106
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.3322
##  Mcnemar's Test P-Value : 0.3899
##
##             Sensitivity : 0.6642
##             Specificity : 0.6681
##          Pos Pred Value : 0.6573
```

```
##          Neg Pred Value : 0.6749
##              Prevalence : 0.4894
##          Detection Rate : 0.3251
##    Detection Prevalence : 0.4946
##       Balanced Accuracy : 0.6661
##
##        'Positive' Class : 1
##
```

```
## Prediction Error on Validation Set
mse.fit.rf.cat <- mean((news.validation.cat$shares-pred.fit.rf.cat)^2)
mse.fit.rf.cat
```

```
## [1] 0.3338129
```

```
## Variable Importance
varImpPlot(fit.rf.cat, sort=TRUE, n.var = 30)
```



3.10 Random Forest - Cross-Validation

```
#######################################################################
# Random Forest - Cross Validation
#######################################################################
# Model fitting
control <- trainControl(method = "repeatedcv", number = 5, savePredictions=TRUE)
```

```r
fit.rf.cv <- train(as.factor(shares) ~ ., data = news.train, method = "rf", tuneLength = 5, trControl =
fit.rf.cv.final <- fit.rf.cv$finalModel
fit.rf.cv.final
```

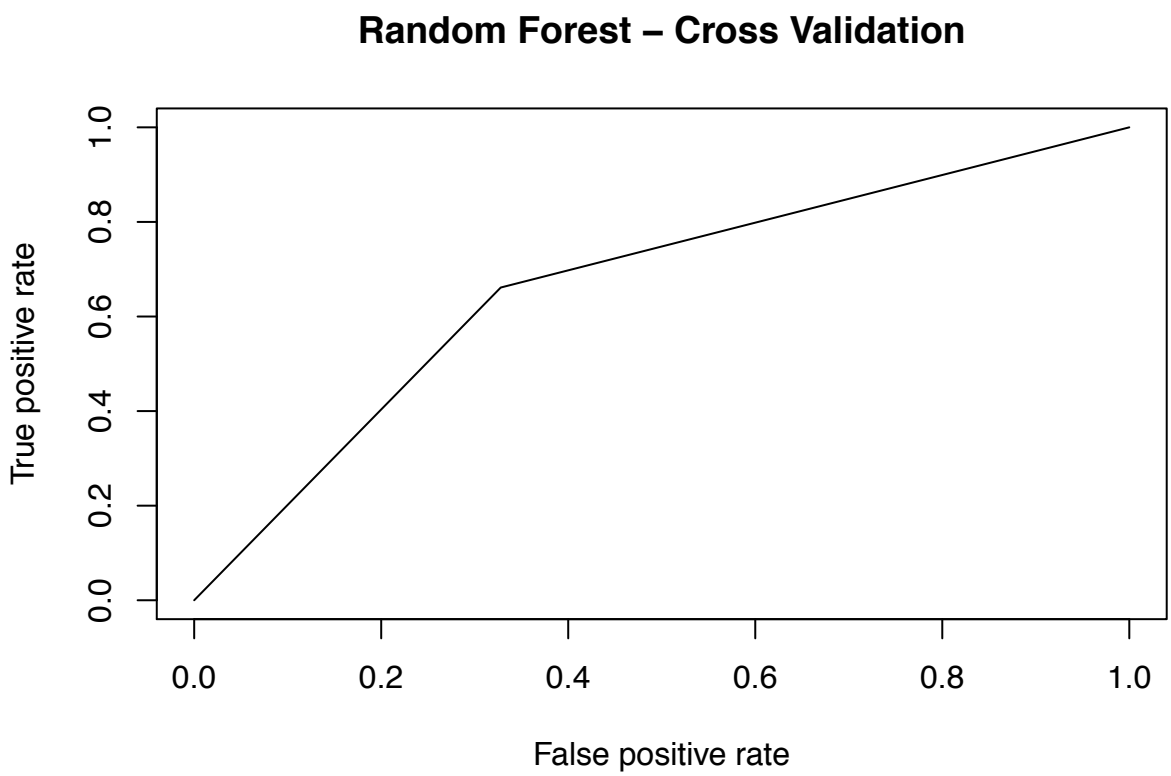
```
##
## Call:
##  randomForest(x = x, y = y, mtry = param$mtry)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 33.55%
## Confusion matrix:
##      0    1 class.error
## 0 6489 3261   0.3344615
## 1 3184 6274   0.3366462
```

```r
summary(fit.rf.cv.final)
```

```
##                 Length Class      Mode
## call                4  -none-     call
## type                1  -none-     character
## predicted       19208  factor     numeric
## err.rate         1500  -none-     numeric
## confusion           6  -none-     numeric
## votes           38416  matrix     numeric
## oob.times       19208  -none-     numeric
## classes             2  -none-     character
## importance         52  -none-     numeric
## importanceSD        0  -none-     NULL
## localImportance     0  -none-     NULL
## proximity           0  -none-     NULL
## ntree               1  -none-     numeric
## mtry                1  -none-     numeric
## forest             14  -none-     list
## y               19208  factor     numeric
## test                0  -none-     NULL
## inbag               0  -none-     NULL
## xNames             52  -none-     character
## problemType         1  -none-     character
## tuneValue           1  data.frame list
## obsLevels           2  -none-     character
```

```r
## Q-Q plots
#qqnorm(fit.rf.cv.final$residuals)
#qqline(fit.rf.cv.final$residuals)

## ROC Curves
scores <- predict(fit.rf.cv.final, news.validation)
prob <- prediction(as.numeric(scores), labels=news.validation$shares)
perf <- performance(prob, "tpr", "fpr")
plot(perf, colorize=F, main="Random Forest - Cross Validation")
```




```
## AUC
auc.fit.rf.cv <- unlist(attributes(performance(prob, "auc"))$y.values)
auc.fit.rf.cv
```

```
## [1] 0.6667118
```

```
## Confusion Matrix
pred.fit.rf.cv <- ifelse(as.numeric(scores)>1, 1, 0)
confusionMatrix(as.numeric(pred.fit.rf.cv), news.validation$shares, positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 3338 1606
##          1 1628 3135
##
##                Accuracy : 0.6668
##                  95% CI : (0.6574, 0.6762)
##     No Information Rate : 0.5116
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.3334
##  Mcnemar's Test P-Value : 0.7119
##
##             Sensitivity : 0.6613
##             Specificity : 0.6722
##          Pos Pred Value : 0.6582
```

```
##          Neg Pred Value : 0.6752
##              Prevalence : 0.4884
##          Detection Rate : 0.3230
##    Detection Prevalence : 0.4907
##       Balanced Accuracy : 0.6667
##
##        'Positive' Class : 1
##
```

```
## Prediction Error on Validation Set
mse.fit.rf.cv <- mean((news.validation$shares-pred.fit.rf.cv)^2)
mse.fit.rf.cv
```

```
## [1] 0.3331616
```

```
## Variable Importance
#fit.rf.cv.final.imp <- varImp(fit.rf.cv.final)
#fit.rf.cv.final.imp$Names <- row.names(fit.rf.cv.final.imp)
#imp.values <- fit.rf.cv.final.imp[order(fit.rf.cv.final.imp$Overall, decreasing=TRUE)[1:30],1]
#imp.names <- fit.rf.cv.final.imp[order(fit.rf.cv.final.imp$Overall, decreasing=TRUE)[1:30],2]
#barplot(imp.values, names = imp.names, las = 2)
#imp.values.plot <- barplot(imp.values, axes = FALSE, axisnames = FALSE)
#text(imp.values.plot, par("usr")[3], labels = imp.names, srt = 45, adj = c(1.0,1.0), xpd = TRUE, cex=.
#axis(2)
#varImpPlot(fit.rf.cv, sort=TRUE, n.var = 30)
```

3.11 Naive Bayes - All 52 Predictors

```
#######################################################################
# Naive Bayes - All 52 Predictors
#######################################################################
## Model fitting
fit.nb.all <- naiveBayes(as.factor(shares) ~ ., data = news.train,
                    laplace = 0.01)
fit.nb.all
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##        0        1
## 0.507601 0.492399
##
## Conditional probabilities:
##    timedelta
## Y       [,1]     [,2]
##   0 351.8253 212.3286
```

```
##   1 372.6382 210.9193
##
##    n_tokens_title
## Y       [,1]     [,2]
##   0 10.48369 2.099312
##   1 10.25693 2.114560
##
##    n_tokens_content
## Y       [,1]     [,2]
##   0 541.1452 417.1528
##   1 580.1450 501.2322
##
##    n_unique_tokens
## Y        [,1]       [,2]
##   0 0.5509231 0.09825618
##   1 0.5429947 0.10496439
##
##    n_non_stop_words
## Y   [,1]         [,2]
##   0    1 3.975407e-09
##   1    1 4.043431e-09
##
##    n_non_stop_unique_tokens
## Y        [,1]       [,2]
##   0 0.6989485 0.09705316
##   1 0.6878740 0.10284656
##
##    num_hrefs
## Y       [,1]      [,2]
##   0 10.14390  9.801597
##   1 12.31254 12.461688
##
##    num_self_hrefs
## Y       [,1]     [,2]
##   0 3.245026 3.566516
##   1 3.573800 4.153360
##
##    num_imgs
## Y       [,1]     [,2]
##   0 4.015385 7.859911
##   1 5.128040 8.636760
##
##    num_videos
## Y       [,1]     [,2]
##   0 1.240205 4.237263
##   1 1.321844 4.406901
##
##    average_token_length
## Y       [,1]      [,2]
##   0 4.696328 0.2805929
##   1 4.678095 0.2841189
##
##    num_keywords
## Y       [,1]     [,2]
```

```
##   0 7.064000 1.918486
##   1 7.344576 1.880428
##
##    data_channel_is_lifestyle
## Y         [,1]      [,2]
##   0 0.04789744 0.2135602
##   1 0.06269824 0.2424322
##
##    data_channel_is_entertainment
## Y        [,1]      [,2]
##   0 0.2207179 0.4147519
##   1 0.1308945 0.3373027
##
##    data_channel_is_bus
## Y        [,1]      [,2]
##   0 0.1633846 0.3697352
##   1 0.1664200 0.3724769
##
##    data_channel_is_socmed
## Y         [,1]      [,2]
##   0 0.03343590 0.1797811
##   1 0.08627617 0.2807863
##
##    data_channel_is_tech
## Y        [,1]      [,2]
##   0 0.1512821 0.3583420
##   1 0.2307042 0.4213057
##
##    data_channel_is_world
## Y        [,1]      [,2]
##   0 0.2717949 0.4449075
##   1 0.1412561 0.3483039
##
##    kw_min_min
## Y       [,1]     [,2]
##   0 23.45426 66.30431
##   1 31.12381 75.18357
##
##    kw_max_min
## Y       [,1]     [,2]
##   0 1055.735 4080.226
##   1 1282.170 4865.931
##
##    kw_avg_min
## Y       [,1]     [,2]
##   0 291.4152 628.9478
##   1 342.4846 763.8474
##
##    kw_min_max
## Y       [,1]     [,2]
##   0 12662.25 59460.38
##   1 13006.42 52133.18
##
##    kw_max_max
```

```
## Y       [,1]     [,2]
##   0 757933.9 210045.7
##   1 741959.2 222962.8
##
##    kw_avg_max
## Y       [,1]     [,2]
##   0 254086.1 128065.7
##   1 257064.3 134977.8
##
##    kw_min_avg
## Y        [,1]     [,2]
##   0  996.6734 1036.156
##   1 1203.3945 1201.867
##
##    kw_max_avg
## Y       [,1]     [,2]
##   0 5199.327 6138.151
##   1 6140.077 7474.803
##
##    kw_avg_avg
## Y       [,1]     [,2]
##   0 2903.353 1195.640
##   1 3332.595 1438.295
##
##    self_reference_min_shares
## Y       [,1]     [,2]
##   0 2897.951 12682.52
##   1 5297.637 22843.08
##
##    self_reference_max_shares
## Y        [,1]     [,2]
##   0  7723.222 28577.69
##   1 12857.940 46606.65
##
##    self_reference_avg_sharess
## Y       [,1]     [,2]
##   0 4762.339 16814.64
##   1 8182.664 27888.39
##
##    is_weekend
## Y         [,1]      [,2]
##   0 0.08307692 0.2760126
##   1 0.17657010 0.3813246
##
##    LDA_00
## Y        [,1]      [,2]
##   0 0.1686540 0.2487415
##   1 0.2117662 0.2826035
##
##    LDA_01
## Y        [,1]      [,2]
##   0 0.1578189 0.2351664
##   1 0.1219013 0.1958560
##
```

```
##    LDA_02
## Y       [,1]      [,2]
##   0 0.259582 0.3096707
##   1 0.168008 0.2394007
##
##    LDA_03
## Y        [,1]      [,2]
##   0 0.2006147 0.2810526
##   1 0.2345495 0.2992509
##
##    LDA_04
## Y        [,1]      [,2]
##   0 0.2133305 0.2751248
##   1 0.2637750 0.3054033
##
##    global_subjectivity
## Y        [,1]       [,2]
##   0 0.4482801 0.08728411
##   1 0.4668890 0.08680308
##
##    global_sentiment_polarity
## Y        [,1]       [,2]
##   0 0.1169798 0.09635949
##   1 0.1306176 0.09534242
##
##    global_rate_positive_words
## Y         [,1]       [,2]
##   0 0.03984364 0.01646563
##   1 0.04207235 0.01589851
##
##    global_rate_negative_words
## Y         [,1]       [,2]
##   0 0.01727619 0.01067166
##   1 0.01695952 0.01048360
##
##    rate_positive_words
## Y        [,1]      [,2]
##   0 0.6949947 0.1543457
##   1 0.7128630 0.1462852
##
##    rate_negative_words
## Y        [,1]      [,2]
##   0 0.3050053 0.1543457
##   1 0.2871370 0.1462852
##
##    avg_positive_polarity
## Y        [,1]       [,2]
##   0 0.3616516 0.08495166
##   1 0.3693109 0.08507541
##
##    min_positive_polarity
## Y         [,1]       [,2]
##   0 0.09998028 0.06912497
##   1 0.09760649 0.07177736
```

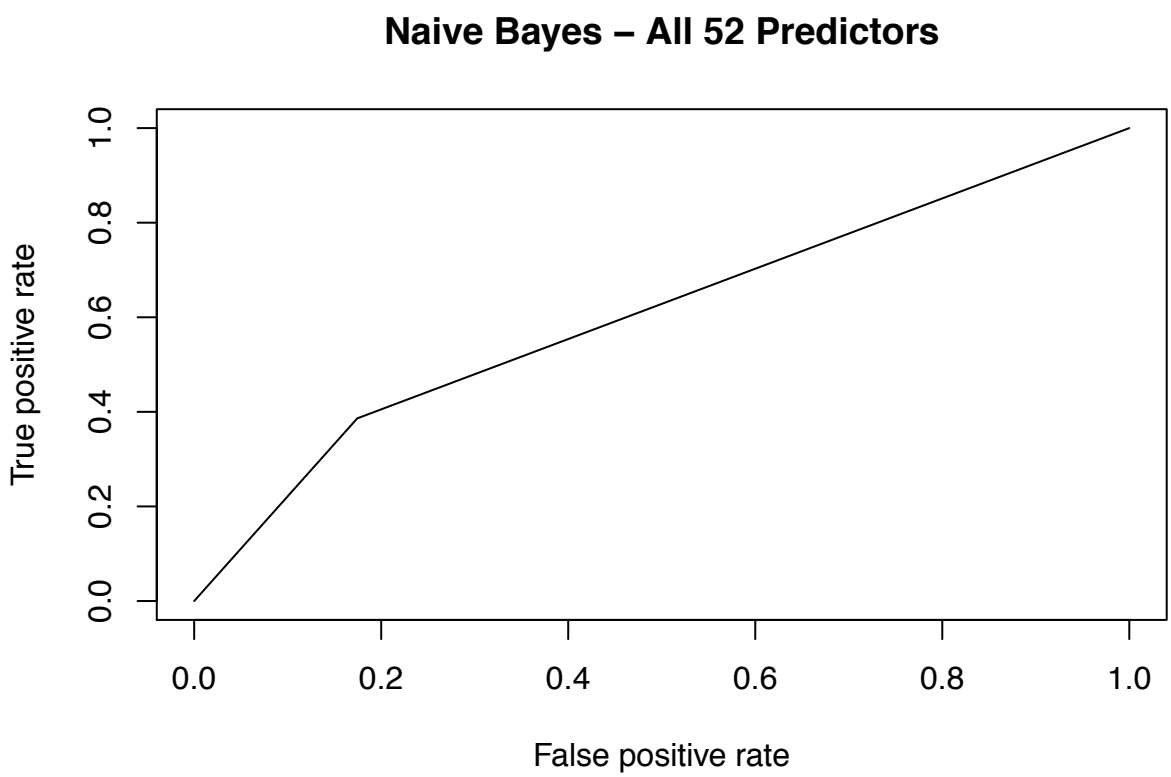
```
##
##    max_positive_polarity
## Y        [,1]      [,2]
##   0 0.7691510 0.2121827
##   1 0.7920525 0.2112550
##
##    avg_negative_polarity
## Y         [,1]      [,2]
##   0 -0.2664905 0.1213368
##   1 -0.2693636 0.1224530
##
##    min_negative_polarity
## Y         [,1]      [,2]
##   0 -0.5352559 0.2796472
##   1 -0.5410727 0.2793548
##
##    max_negative_polarity
## Y        [,1]       [,2]
##   0 -0.110717 0.09438168
##   1 -0.111959 0.09840840
##
##    title_subjectivity
## Y        [,1]      [,2]
##   0 0.2699761 0.3147301
##   1 0.2906008 0.3298548
##
##    title_sentiment_polarity
## Y         [,1]      [,2]
##   0 0.05714087 0.2549165
##   1 0.08472946 0.2759676
##
##    abs_title_subjectivity
## Y        [,1]      [,2]
##   0 0.3408352 0.1891848
##   1 0.3422861 0.1883739
##
##    abs_title_sentiment_polarity
## Y        [,1]      [,2]
##   0 0.1469672 0.2159775
##   1 0.1648975 0.2369470
```

```r
summary(fit.nb.all)
```

```
##         Length Class  Mode
## apriori  2     table  numeric
## tables  52     -none- list
## levels   2     -none- character
## call     4     -none- call
```

```r
## Q-Q plots
#qqnorm(fit.nb.all$residuals)
#qqline(fit.nb.all$residuals)
```

```r
## ROC Curves
scores <- predict(fit.nb.all, news.validation)
prob <- prediction(as.numeric(scores), labels=news.validation$shares)
perf <- performance(prob, "tpr", "fpr")
plot(perf, colorize=F, main="Naive Bayes - All 52 Predictors")
```




```r
## AUC
auc.fit.nb.all <- unlist(attributes(performance(prob, "auc"))$y.values)
auc.fit.nb.all
```

```
## [1] 0.6059098
```

```r
## Confusion Matrix
confusionMatrix(scores, news.validation$shares, positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 4100 2910
##          1  866 1831
##
##                Accuracy : 0.611
##                  95% CI : (0.6012, 0.6207)
##     No Information Rate : 0.5116
##     P-Value [Acc > NIR] : < 2.2e-16
##
```

```
##                   Kappa : 0.2139
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.3862
##             Specificity : 0.8256
##          Pos Pred Value : 0.6789
##          Neg Pred Value : 0.5849
##              Prevalence : 0.4884
##          Detection Rate : 0.1886
##    Detection Prevalence : 0.2778
##       Balanced Accuracy : 0.6059
##
##        'Positive' Class : 1
##
```

```
## Prediction Error on Validation Set
mse.fit.nb.all = mean((news.validation$shares-as.numeric(scores))^2)
mse.fit.nb.all
```

```
## [1] 0.9678582
```

```
## Variable Importance
#fit.nb.all.imp <- varImp(fit.nb.all)
#fit.nb.all.imp$Names <- row.names(fit.nb.all.imp)
#imp.values <- fit.nb.all.imp[order(fit.nb.all.imp$Overall, decreasing=TRUE)[1:30],1]
#imp.names <- fit.nb.all.imp[order(fit.nb.all.imp$Overall, decreasing=TRUE)[1:30],2]
#barplot(imp.values, names = imp.names, las = 2)
#imp.values.plot <- barplot(imp.values, axes = FALSE, axisnames = FALSE)
#text(imp.values.plot, par("usr")[3], labels = imp.names, srt = 45, adj = c(1.0,1.0), xpd = TRUE, cex=.
#axis(2)
```

3.12 Naive Bayes - Top 30 Predictors

```
#######################################################################
# Naive Bayes - Top 30 Predictors
#######################################################################
## Model fitting
top_formula <- as.formula(paste("as.factor(shares) ~ ", f))
fit.nb.top <- naiveBayes(top_formula, data = news.train,
                    laplace = 0.01)
fit.nb.top
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##        0        1
```

```
## 0.507601 0.492399
##
## Conditional probabilities:
##    kw_avg_avg
## Y       [,1]     [,2]
##   0 2903.353 1195.640
##   1 3332.595 1438.295
##
##    self_reference_avg_sharess
## Y       [,1]     [,2]
##   0 4762.339 16814.64
##   1 8182.664 27888.39
##
##    self_reference_min_shares
## Y       [,1]     [,2]
##   0 2897.951 12682.52
##   1 5297.637 22843.08
##
##    kw_max_avg
## Y       [,1]     [,2]
##   0 5199.327 6138.151
##   1 6140.077 7474.803
##
##    kw_min_avg
## Y        [,1]     [,2]
##   0  996.6734 1036.156
##   1 1203.3945 1201.867
##
##    LDA_02
## Y       [,1]      [,2]
##   0 0.259582 0.3096707
##   1 0.168008 0.2394007
##
##    self_reference_max_shares
## Y        [,1]     [,2]
##   0  7723.222 28577.69
##   1 12857.940 46606.65
##
##    data_channel_is_world
## Y        [,1]      [,2]
##   0 0.2717949 0.4449075
##   1 0.1412561 0.3483039
##
##    is_weekend
## Y         [,1]      [,2]
##   0 0.08307692 0.2760126
##   1 0.17657010 0.3813246
##
##    data_channel_is_entertainment
## Y        [,1]      [,2]
##   0 0.2207179 0.4147519
##   1 0.1308945 0.3373027
##
##    num_imgs
```

```
## Y       [,1]     [,2]
##   0 4.015385 7.859911
##   1 5.128040 8.636760
##
##    timedelta
## Y       [,1]     [,2]
##   0 351.8253 212.3286
##   1 372.6382 210.9193
##
##    LDA_04
## Y        [,1]      [,2]
##   0 0.2133305 0.2751248
##   1 0.2637750 0.3054033
##
##    global_subjectivity
## Y        [,1]       [,2]
##   0 0.4482801 0.08728411
##   1 0.4668890 0.08680308
##
##    data_channel_is_tech
## Y        [,1]      [,2]
##   0 0.1512821 0.3583420
##   1 0.2307042 0.4213057
##
##    num_hrefs
## Y       [,1]      [,2]
##   0 10.14390  9.801597
##   1 12.31254 12.461688
##
##    kw_avg_min
## Y       [,1]     [,2]
##   0 291.4152 628.9478
##   1 342.4846 763.8474
##
##    min_positive_polarity
## Y         [,1]       [,2]
##   0 0.09998028 0.06912497
##   1 0.09760649 0.07177736
##
##    LDA_01
## Y        [,1]      [,2]
##   0 0.1578189 0.2351664
##   1 0.1219013 0.1958560
##
##    LDA_00
## Y        [,1]      [,2]
##   0 0.1686540 0.2487415
##   1 0.2117662 0.2826035
##
##    kw_max_min
## Y       [,1]     [,2]
##   0 1055.735 4080.226
##   1 1282.170 4865.931
##
```

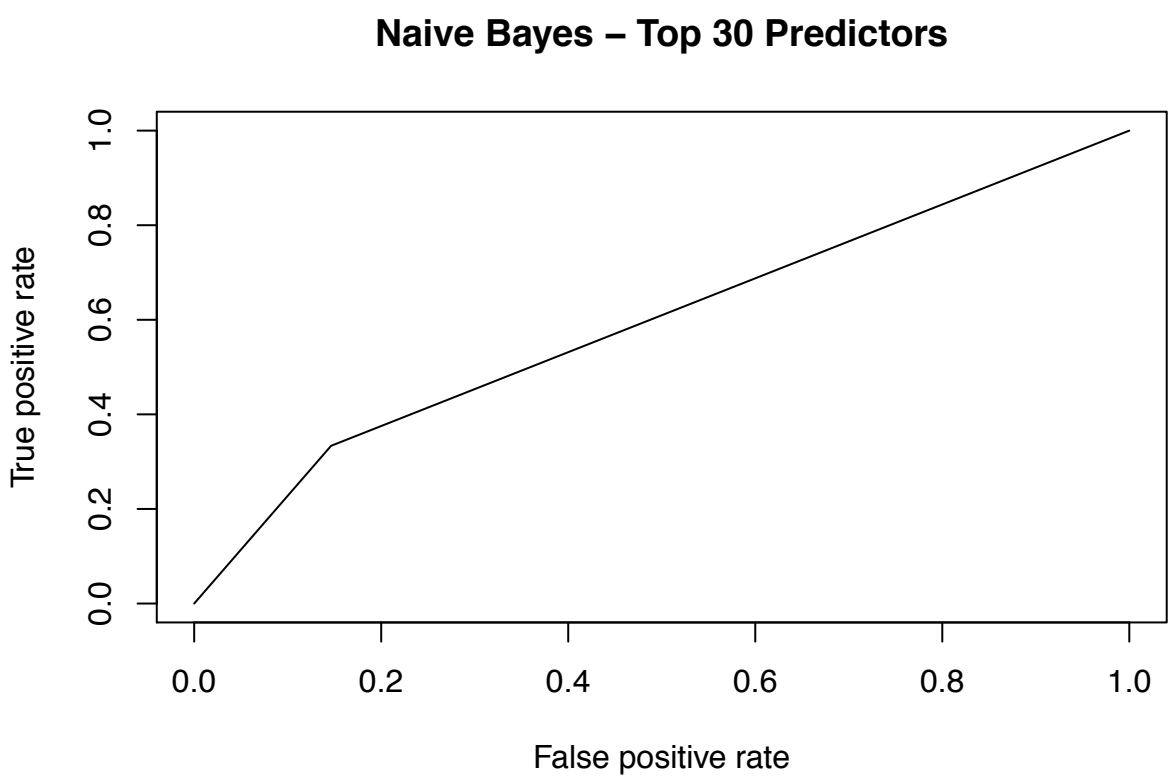
```
##    global_sentiment_polarity
## Y        [,1]       [,2]
##   0 0.1169798 0.09635949
##   1 0.1306176 0.09534242
##
##    LDA_03
## Y        [,1]      [,2]
##   0 0.2006147 0.2810526
##   1 0.2345495 0.2992509
##
##    rate_negative_words
## Y        [,1]      [,2]
##   0 0.3050053 0.1543457
##   1 0.2871370 0.1462852
##
##    kw_min_max
## Y       [,1]     [,2]
##   0 12662.25 59460.38
##   1 13006.42 52133.18
##
##    n_unique_tokens
## Y        [,1]       [,2]
##   0 0.5509231 0.09825618
##   1 0.5429947 0.10496439
##
##    global_rate_positive_words
## Y         [,1]       [,2]
##   0 0.03984364 0.01646563
##   1 0.04207235 0.01589851
##
##    n_non_stop_unique_tokens
## Y        [,1]       [,2]
##   0 0.6989485 0.09705316
##   1 0.6878740 0.10284656
##
##    rate_positive_words
## Y        [,1]      [,2]
##   0 0.6949947 0.1543457
##   1 0.7128630 0.1462852
##
##    kw_avg_max
## Y       [,1]     [,2]
##   0 254086.1 128065.7
##   1 257064.3 134977.8
```

```r
summary(fit.nb.top)
```

```
##         Length Class  Mode
## apriori  2     table  numeric
## tables  30     -none- list
## levels   2     -none- character
## call     4     -none- call
```

```r
## Q-Q plots
#qqnorm(fit.nb.top$residuals)
#qqline(fit.nb.top$residuals)

## ROC Curves
scores <- predict(fit.nb.top, news.validation)
prob <- prediction(as.numeric(scores), labels=news.validation$shares)
perf <- performance(prob, "tpr", "fpr")
plot(perf, colorize=F, main="Naive Bayes - Top 30 Predictors")
```




```r
## AUC
auc.fit.nb.top <- unlist(attributes(performance(prob, "auc"))$y.values)
auc.fit.nb.top
```

```
## [1] 0.5935392
```

```r
## Confusion Matrix
confusionMatrix(scores, news.validation$shares, positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 4239 3160
##          1  727 1581
##
##                Accuracy : 0.5996
```

```
##                  95% CI : (0.5897, 0.6093)
##     No Information Rate : 0.5116
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.1893
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.3335
##             Specificity : 0.8536
##          Pos Pred Value : 0.6850
##          Neg Pred Value : 0.5729
##              Prevalence : 0.4884
##          Detection Rate : 0.1629
##    Detection Prevalence : 0.2378
##       Balanced Accuracy : 0.5935
##
##        'Positive' Class : 1
##
```

```
## Prediction Error on Validation Set
mse.fit.nb.top = mean((news.validation$shares-as.numeric(scores))^2)
mse.fit.nb.top
```

```
## [1] 0.8991449
```

```
## Variable Importance
#fit.nb.top.imp <- varImp(fit.nb.top)
#fit.nb.top.imp$Names <- row.names(fit.nb.top.imp)
#imp.values <- fit.nb.top.imp[order(fit.nb.top.imp$Overall, decreasing=TRUE)[1:30],1]
#imp.names <- fit.nb.top.imp[order(fit.nb.top.imp$Overall, decreasing=TRUE)[1:30],2]
#barplot(imp.values, names = imp.names, las = 2)
#imp.values.plot <- barplot(imp.values, axes = FALSE, axisnames = FALSE)
#text(imp.values.plot, par("usr")[3], labels = imp.names, srt = 45, adj = c(1.0,1.0), xpd = TRUE, cex=.
#axis(2)
```

3.13 Naive Bayes - Cross Validation

```
#######################################################################
# Naive Bayes - Cross Validation
#######################################################################
## Model fitting
ctrl <- trainControl(method="repeatedcv", number = 2, savePredictions = TRUE)
fit.nb.cv <- train(news.train[, -53], as.factor(news.train$shares), method="nb",
                   trControl=ctrl, tuneLength = 5)
fit.nb.cv
```

```
## Naive Bayes
##
## 19208 samples
##    52 predictor
##     2 classes: '0', '1'
```

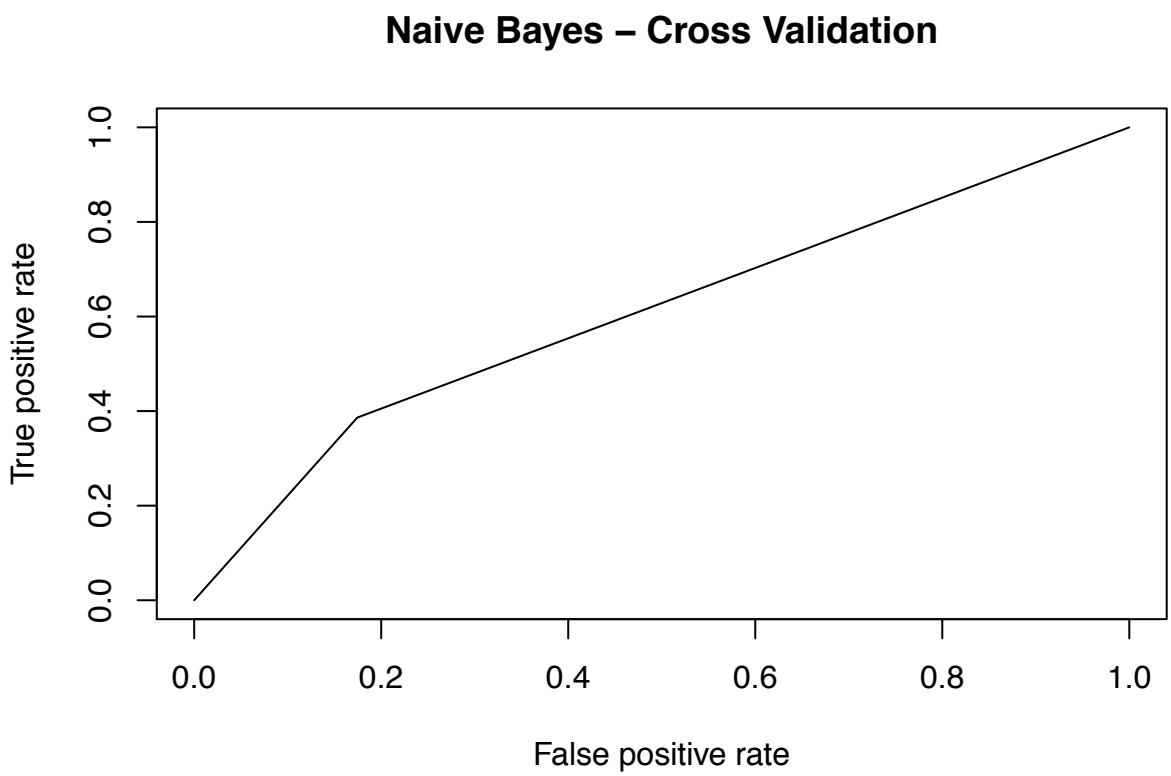
```
##
## No pre-processing
## Resampling: Cross-Validated (2 fold, repeated 1 times)
## Summary of sample sizes: 9604, 9604
## Resampling results across tuning parameters:
##
##   usekernel  Accuracy   Kappa
##   FALSE      0.6094856  0.2137947
##    TRUE      0.5989171  0.1909823
##
## Tuning parameter 'fL' was held constant at a value of 0
## Tuning
##  parameter 'adjust' was held constant at a value of 1
## Accuracy was used to select the optimal model using  the largest value.
## The final values used for the model were fL = 0, usekernel = FALSE
##  and adjust = 1.
```

```r
summary(fit.nb.cv)
```

```
##             Length Class      Mode
## apriori      2     table      numeric
## tables      52     -none-     list
## levels       2     -none-     character
## call         5     -none-     call
## x           52     data.frame list
## usekernel    1     -none-     logical
## varnames    52     -none-     character
## xNames      52     -none-     character
## problemType  1     -none-     character
## tuneValue    3     data.frame list
## obsLevels    2     -none-     character
```

```r
## Q-Q plots
#qqnorm(fit.nb.cv$residuals)
#qqline(fit.nb.cv$residuals)

## ROC Curves
scores <- predict(fit.nb.cv, news.validation)
prob <- prediction(as.numeric(scores), labels=news.validation$shares)
perf <- performance(prob, "tpr", "fpr")
plot(perf, colorize=F, main="Naive Bayes - Cross Validation")
```




```r
## AUC
auc.fit.nb.cv <- unlist(attributes(performance(prob, "auc"))$y.values)
auc.fit.nb.cv
```

```
## [1] 0.6059098
```

```r
## Confusion Matrix
confusionMatrix(scores, news.validation$shares, positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 4100 2910
##          1  866 1831
##
##                Accuracy : 0.611
##                  95% CI : (0.6012, 0.6207)
##     No Information Rate : 0.5116
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.2139
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.3862
##             Specificity : 0.8256
##          Pos Pred Value : 0.6789
##          Neg Pred Value : 0.5849
```

```
##              Prevalence : 0.4884
##          Detection Rate : 0.1886
##    Detection Prevalence : 0.2778
##       Balanced Accuracy : 0.6059
##
##        'Positive' Class : 1
##
```

```
## Prediction Error on Validation Set
mse.fit.nb.cv = mean((news.validation$shares-as.numeric(scores))^2)
mse.fit.nb.cv
```

```
## [1] 0.9678582
```

```
## Variable Importance
#fit.nb.top.imp <- varImp(fit.nb.top)
#fit.nb.top.imp$Names <- row.names(fit.nb.top.imp)
#imp.values <- fit.nb.top.imp[order(fit.nb.top.imp$Overall, decreasing=TRUE)[1:30],1]
#imp.names <- fit.nb.top.imp[order(fit.nb.top.imp$Overall, decreasing=TRUE)[1:30],2]
#barplot(imp.values, names = imp.names, las = 2)
#imp.values.plot <- barplot(imp.values, axes = FALSE, axisnames = FALSE)
#text(imp.values.plot, par("usr")[3], labels = imp.names, srt = 45, adj = c(1.0,1.0), xpd = TRUE, cex=.
#axis(2)
```

4 Artificially unbalancing the data

```
#######################################################################
# Artificially altering data
#######################################################################
## 95 percentile
news.data <- news.data.copy
quantile(news.data$shares, c(.90, .95, .99))
```

```
##   90%   95%   99%
##  6100 10700 31500
```

```
## Data Prep - Convert shares to 1 and 0
news.data$shares <- ifelse(news.data$shares > 10700, 1, 0)
sample.index <- sample(1:3, size=nrow(news.data), replace=TRUE,
                       prob=c(0.5,0.25,0.25))
news.train <- news.data[sample.index == 1, ]
news.validation <- news.data[sample.index == 2, ]
news.test <- news.data[sample.index == 3, ]

## Model fitting
fit.lm.95 <- glm(shares ~ ., data = news.train,
                       family=binomial)
fit.lm.95
```

```
##
## Call:  glm(formula = shares ~ ., family = binomial, data = news.train)
##
## Coefficients:
##                   (Intercept)                      timedelta
##                     1.067e+07                      1.072e-03
##                n_tokens_title                n_tokens_content
##                     2.779e-02                      2.374e-04
##               n_unique_tokens                n_non_stop_words
##                     2.697e+00                     -1.067e+07
##      n_non_stop_unique_tokens                      num_hrefs
##                    -1.524e+00                      1.124e-02
##                num_self_hrefs                       num_imgs
##                    -3.223e-02                      3.028e-03
##                    num_videos            average_token_length
##                     5.432e-03                     -3.671e-01
##                  num_keywords       data_channel_is_lifestyle
##                     8.427e-03                     -1.959e-01
## data_channel_is_entertainment             data_channel_is_bus
##                    -2.572e-01                     -5.530e-01
##        data_channel_is_socmed            data_channel_is_tech
##                    -3.415e-02                     -2.707e-01
##         data_channel_is_world                      kw_min_min
##                    -3.129e-02                      1.897e-03
##                    kw_max_min                      kw_avg_min
##                     3.627e-05                     -2.294e-04
##                    kw_min_max                      kw_max_max
```

```
##                    -1.403e-06                     8.307e-07
##                    kw_avg_max                    kw_min_avg
##                    -1.991e-07                    -1.645e-04
##                    kw_max_avg                    kw_avg_avg
##                    -8.832e-05                     6.449e-04
##     self_reference_min_shares      self_reference_max_shares
##                    -1.464e-06                    -2.162e-06
##    self_reference_avg_sharess                    is_weekend
##                     7.101e-06                     7.677e-02
##                        LDA_00                        LDA_01
##                    -3.436e-01                    -2.802e-01
##                        LDA_02                        LDA_03
##                    -1.131e+00                    -2.132e-01
##                        LDA_04            global_subjectivity
##                            NA                     1.369e+00
##     global_sentiment_polarity     global_rate_positive_words
##                     6.181e-01                    -8.984e+00
##    global_rate_negative_words            rate_positive_words
##                    -1.307e+01                    -8.859e-01
##           rate_negative_words          avg_positive_polarity
##                            NA                    -4.099e-02
##         min_positive_polarity          max_positive_polarity
##                    -2.035e-01                     5.492e-01
##         avg_negative_polarity          min_negative_polarity
##                    -4.400e-01                    -1.889e-01
##         max_negative_polarity             title_subjectivity
##                     5.918e-01                     1.141e-01
##      title_sentiment_polarity         abs_title_subjectivity
##                     5.202e-02                    -1.313e-02
##  abs_title_sentiment_polarity
##                     1.303e-01
##
## Degrees of Freedom: 19283 Total (i.e. Null);  19233 Residual
## Null Deviance:       7537
## Residual Deviance: 6871  AIC: 6973
```

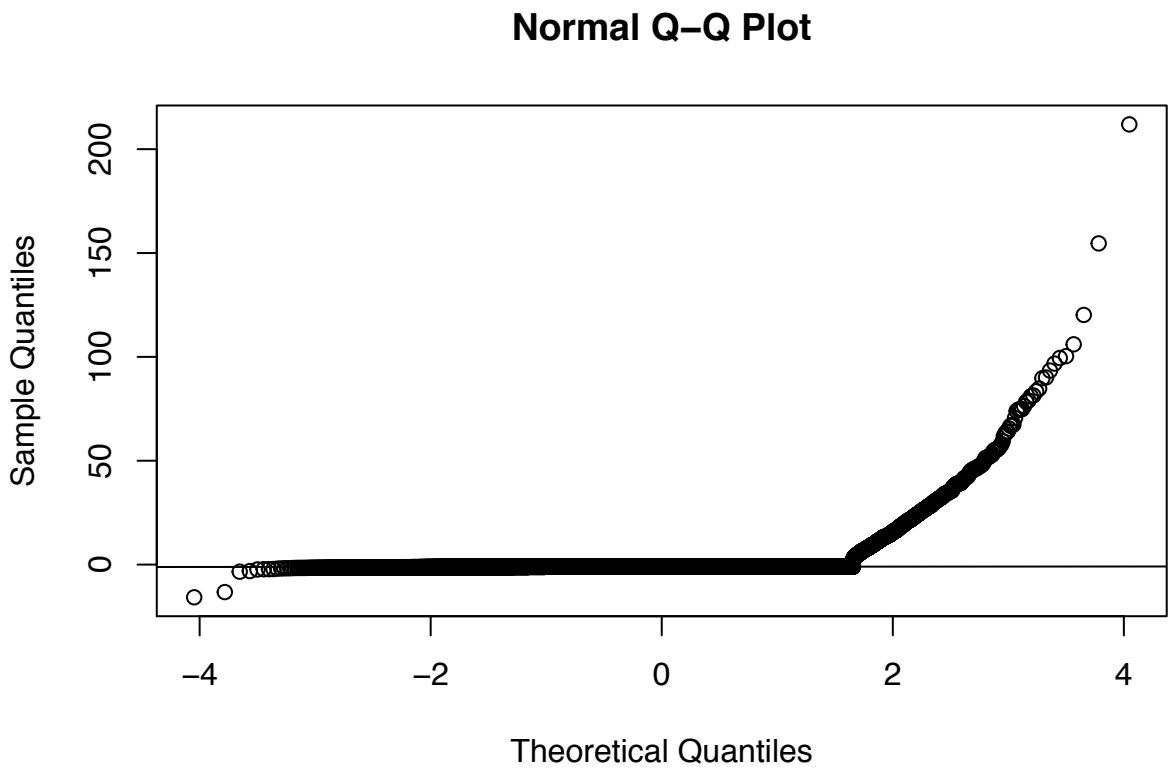
```r
summary(fit.lm.95)
```

```
##
## Call:
## glm(formula = shares ~ ., family = binomial, data = news.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3482  -0.3287  -0.2527  -0.1969   3.2730
##
## Coefficients: (2 not defined because of singularities)
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    1.067e+07  1.510e+07   0.707 0.479772
## timedelta                      1.072e-03  2.360e-04   4.543 5.54e-06 ***
## n_tokens_title                 2.779e-02  1.755e-02   1.584 0.113230
## n_tokens_content               2.374e-04  1.026e-04   2.314 0.020674 *
## n_unique_tokens                2.697e+00  1.200e+00   2.246 0.024688 *
## n_non_stop_words              -1.067e+07  1.510e+07  -0.707 0.479772
```

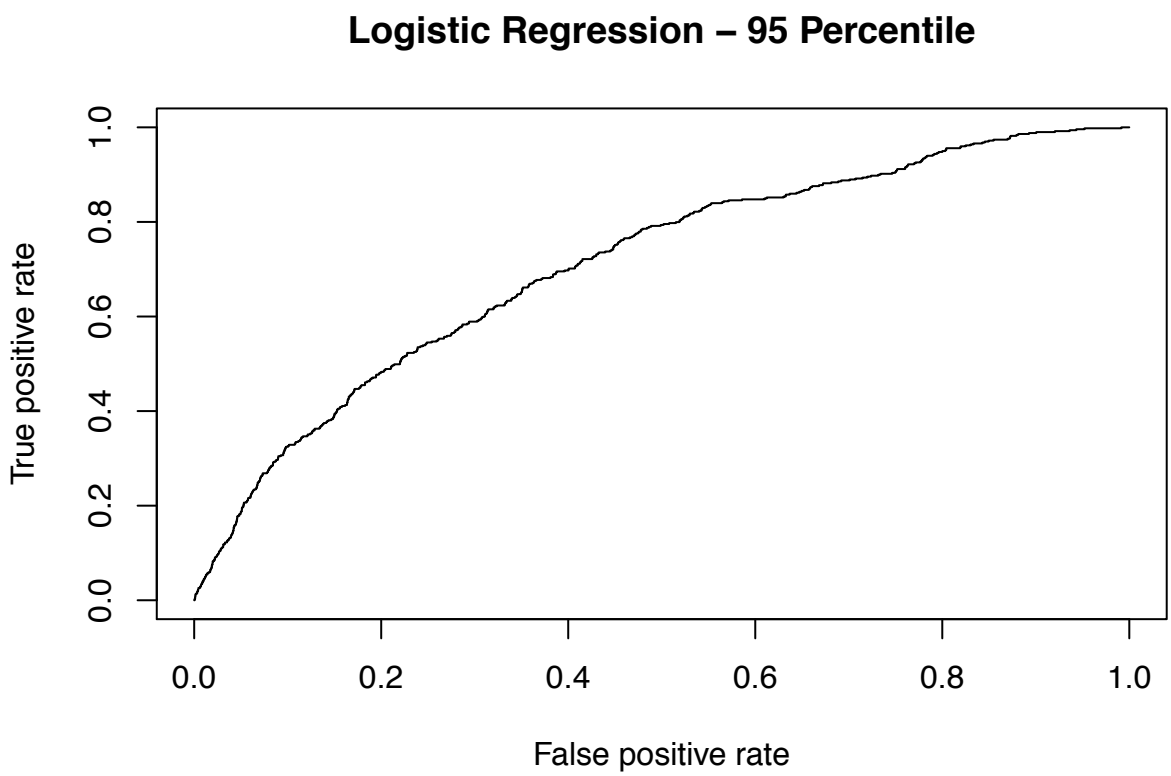
```
## n_non_stop_unique_tokens      -1.524e+00  9.029e-01  -1.688 0.091365 .
## num_hrefs                      1.124e-02  2.851e-03   3.941 8.10e-05 ***
## num_self_hrefs                -3.223e-02  1.147e-02  -2.809 0.004967 **
## num_imgs                       3.028e-03  4.377e-03   0.692 0.488991
## num_videos                     5.432e-03  7.391e-03   0.735 0.462377
## average_token_length          -3.671e-01  1.381e-01  -2.658 0.007870 **
## num_keywords                   8.427e-03  2.216e-02   0.380 0.703695
## data_channel_is_lifestyle     -1.959e-01  2.027e-01  -0.967 0.333778
## data_channel_is_entertainment -2.572e-01  1.319e-01  -1.950 0.051184 .
## data_channel_is_bus           -5.530e-01  2.239e-01  -2.470 0.013501 *
## data_channel_is_socmed        -3.415e-02  2.016e-01  -0.169 0.865492
## data_channel_is_tech          -2.707e-01  2.047e-01  -1.322 0.186099
## data_channel_is_world         -3.129e-02  2.234e-01  -0.140 0.888615
## kw_min_min                     1.897e-03  8.878e-04   2.136 0.032651 *
## kw_max_min                     3.627e-05  3.180e-05   1.141 0.254065
## kw_avg_min                    -2.294e-04  1.846e-04  -1.243 0.213937
## kw_min_max                    -1.403e-06  9.384e-07  -1.495 0.134833
## kw_max_max                     8.307e-07  3.526e-07   2.356 0.018465 *
## kw_avg_max                    -1.991e-07  4.572e-07  -0.436 0.663170
## kw_min_avg                    -1.645e-04  3.932e-05  -4.184 2.86e-05 ***
## kw_max_avg                    -8.832e-05  1.427e-05  -6.188 6.08e-10 ***
## kw_avg_avg                     6.449e-04  7.685e-05   8.391  < 2e-16 ***
## self_reference_min_shares     -1.464e-06  3.462e-06  -0.423 0.672305
## self_reference_max_shares     -2.163e-06  2.367e-06  -0.914 0.360841
## self_reference_avg_sharess     7.101e-06  5.445e-06   1.304 0.192172
## is_weekend                     7.677e-02  9.677e-02   0.793 0.427559
## LDA_00                        -3.436e-01  2.870e-01  -1.197 0.231200
## LDA_01                        -2.802e-01  2.940e-01  -0.953 0.340662
## LDA_02                        -1.131e+00  3.193e-01  -3.542 0.000398 ***
## LDA_03                        -2.132e-01  2.692e-01  -0.792 0.428376
## LDA_04                                 NA         NA      NA       NA
## global_subjectivity            1.369e+00  4.645e-01   2.948 0.003202 **
## global_sentiment_polarity      6.181e-01  8.258e-01   0.749 0.454132
## global_rate_positive_words    -8.984e+00  4.076e+00  -2.204 0.027536 *
## global_rate_negative_words    -1.307e+01  7.462e+00  -1.752 0.079784 .
## rate_positive_words           -8.859e-01  6.590e-01  -1.344 0.178825
## rate_negative_words                    NA         NA      NA       NA
## avg_positive_polarity         -4.099e-02  7.411e-01  -0.055 0.955891
## min_positive_polarity         -2.035e-01  5.933e-01  -0.343 0.731553
## max_positive_polarity          5.492e-01  2.610e-01   2.105 0.035330 *
## avg_negative_polarity         -4.400e-01  6.981e-01  -0.630 0.528537
## min_negative_polarity         -1.889e-01  2.649e-01  -0.713 0.475823
## max_negative_polarity          5.918e-01  5.541e-01   1.068 0.285509
## title_subjectivity             1.141e-01  1.476e-01   0.773 0.439350
## title_sentiment_polarity       5.202e-02  1.272e-01   0.409 0.682604
## abs_title_subjectivity        -1.313e-02  2.028e-01  -0.065 0.948382
## abs_title_sentiment_polarity   1.303e-01  2.090e-01   0.623 0.533133
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 7536.9  on 19283  degrees of freedom
## Residual deviance: 6870.7  on 19233  degrees of freedom
```

```
## AIC: 6972.7
##
## Number of Fisher Scoring iterations: 6
```

```r
## Q-Q plot
qqnorm(fit.lm.95$residuals)
qqline(fit.lm.95$residuals)
```




```r
## ROC Curves
scores <- predict(fit.lm.95, newdata=news.validation, type = "response")
prob <- prediction(scores, labels=news.validation$shares)
perf <- performance(prob, "tpr", "fpr")
plot(perf, colorize=F, main="Logistic Regression - 95 Percentile")
```




```
## AUC
auc.fit.lm.95 <- unlist(attributes(performance(prob, "auc"))$y.values)
auc.fit.lm.95
```

```
## [1] 0.7087991
```

```
## Confusion Matrix
pred.fit.lm.95 <- ifelse(scores>0.5, 1, 0)
confusionMatrix(pred.fit.lm.95, news.validation$shares,
                positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 9081  498
##          1    5    1
##
##                Accuracy : 0.9475
##                  95% CI : (0.9429, 0.9519)
##     No Information Rate : 0.9479
##     P-Value [Acc > NIR] : 0.5845
##
##                   Kappa : 0.0027
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.0020040
##             Specificity : 0.9994497
```

```
##          Pos Pred Value : 0.1666667
##          Neg Pred Value : 0.9480113
##              Prevalence : 0.0520605
##          Detection Rate : 0.0001043
##    Detection Prevalence : 0.0006260
##       Balanced Accuracy : 0.5007269
##
##        'Positive' Class : 1
##
```

```
## Prediction Error on Validation Set
mse.fit.lm.95 = mean((news.validation$shares -
                              pred.fit.lm.95)^2)
mse.fit.lm.95
```

```
## [1] 0.05247783
```

```
## Changing validation set to a balanced validation set by undersampling majority class
news.validation.popular <- news.validation[news.validation[, 53] == 1, ]

news.validation.unpopular <- news.validation[news.validation[, 53] == 0, ]
dim(news.validation.popular)
```

```
## [1] 499  53
```

```
dim(news.validation.unpopular)
```

```
## [1] 9086   53
```

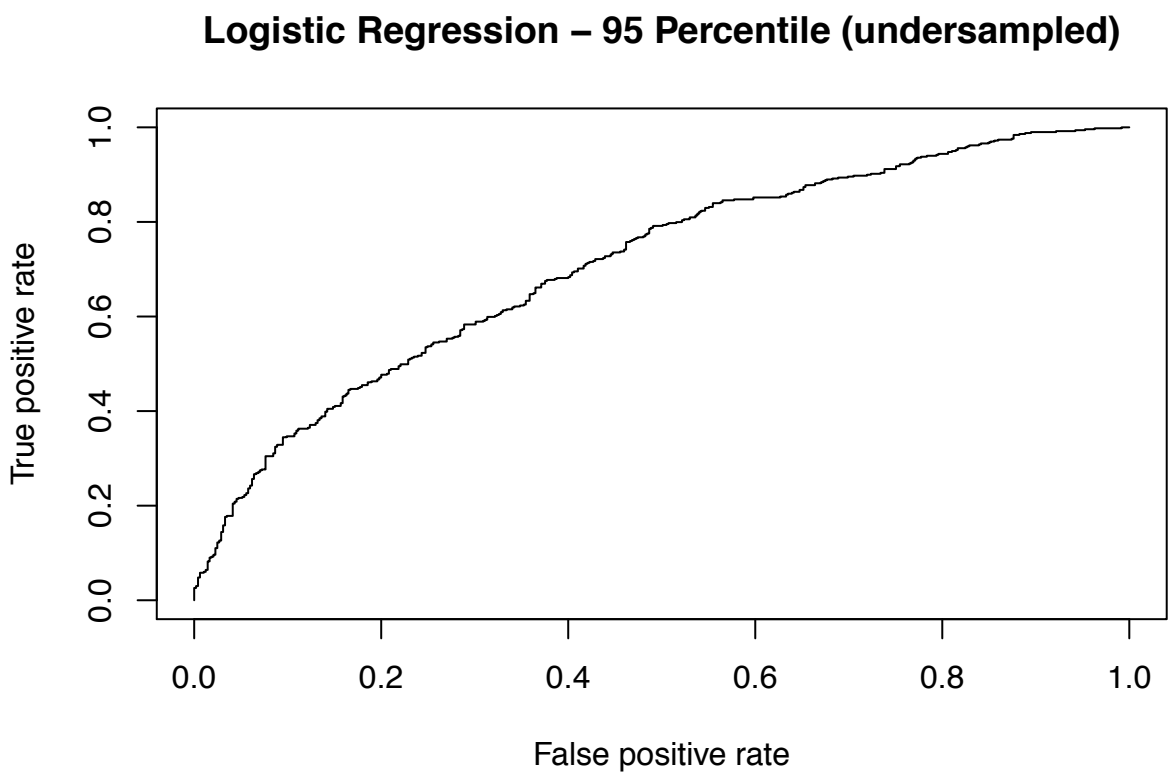
```
news.validation.unpopular <- news.validation.unpopular[sample(nrow(news.validation.unpopular), 485), ]
dim(news.validation.unpopular)
```

```
## [1] 485  53
```

```
news.validation <- rbind(news.validation.popular, news.validation.unpopular)
dim(news.validation)
```

```
## [1] 984  53
```

```
## ROC Curves
scores <- predict(fit.lm.95, newdata=news.validation,
                  type = "response")
prob <- prediction(scores, labels=news.validation$shares)
perf <- performance(prob, "tpr", "fpr")
plot(perf, colorize=F, main="Logistic Regression - 95 Percentile (undersampled)")
```




```
## AUC
auc.fit.lm.undersampled <- unlist(attributes(performance(prob, "auc"))$y.values)
auc.fit.lm.undersampled
```

```
## [1] 0.7092205
```

```
## Confusion Matrix
pred.fit.lm.undersampled <- ifelse(scores>0.5, 1, 0)
confusionMatrix(pred.fit.lm.undersampled, news.validation$shares,
                positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 485 498
##          1   0   1
##
##                Accuracy : 0.4939
##                  95% CI : (0.4622, 0.5256)
##     No Information Rate : 0.5071
##     P-Value [Acc > NIR] : 0.8053
##
##                   Kappa : 0.002
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.002004
##             Specificity : 1.000000
```

```
##          Pos Pred Value : 1.000000
##          Neg Pred Value : 0.493388
##              Prevalence : 0.507114
##          Detection Rate : 0.001016
##    Detection Prevalence : 0.001016
##       Balanced Accuracy : 0.501002
##
##        'Positive' Class : 1
##
```

```
######################################################################
## Fixing Oversampling issue
news.train.popular <- news.train[news.train[, 53] == 1, ]

news.train.unpopular <- news.train[news.train[, 53] == 0, ]
dim(news.train.popular)
```

```
## [1] 944  53
```

```
dim(news.train.unpopular)
```

```
## [1] 18340    53
```

```
news.train.unpopular <- news.train.unpopular[sample(nrow(news.train.unpopular), 940), ]
dim(news.validation.unpopular)
```

```
## [1] 485  53
```

```
news.train <- rbind(news.train.popular, news.train.unpopular)
dim(news.train)
```

```
## [1] 1884   53
```

```
## Model fitting
fit.lm.95.over <- glm(shares ~ ., data = news.train,
                      family=binomial)
fit.lm.95.over
```

```
##
## Call:  glm(formula = shares ~ ., family = binomial, data = news.train)
##
## Coefficients:
##                   (Intercept)                      timedelta
##                    -1.678e+07                      7.744e-04
##                n_tokens_title               n_tokens_content
##                     5.385e-02                      1.450e-04
##               n_unique_tokens               n_non_stop_words
##                     3.592e+00                      1.678e+07
##      n_non_stop_unique_tokens                      num_hrefs
##                    -1.660e+00                      1.314e-02
```

```
##                num_self_hrefs                       num_imgs
##                    -4.012e-02                      1.162e-02
##                    num_videos           average_token_length
##                     8.480e-03                     -6.743e-01
##                  num_keywords      data_channel_is_lifestyle
##                    -8.875e-03                     -2.333e-01
## data_channel_is_entertainment            data_channel_is_bus
##                    -4.947e-01                     -2.676e-01
##        data_channel_is_socmed           data_channel_is_tech
##                    -6.815e-02                     -9.593e-02
##         data_channel_is_world                     kw_min_min
##                    -4.341e-02                      2.623e-05
##                    kw_max_min                     kw_avg_min
##                    -2.753e-05                      9.939e-05
##                    kw_min_max                     kw_max_max
##                    -7.835e-07                     -1.478e-07
##                    kw_avg_max                     kw_min_avg
##                    -9.430e-07                     -2.953e-04
##                    kw_max_avg                     kw_avg_avg
##                    -1.063e-04                      9.859e-04
##     self_reference_min_shares      self_reference_max_shares
##                     8.307e-06                     -1.182e-06
##    self_reference_avg_sharess                     is_weekend
##                     3.758e-06                      2.079e-01
##                        LDA_00                         LDA_01
##                    -5.513e-01                     -7.084e-02
##                        LDA_02                         LDA_03
##                    -7.323e-01                     -3.042e-01
##                        LDA_04            global_subjectivity
##                            NA                      1.337e+00
##     global_sentiment_polarity     global_rate_positive_words
##                    -3.711e-01                     -3.513e+00
##    global_rate_negative_words            rate_positive_words
##                    -9.717e+00                     -7.165e-01
##           rate_negative_words          avg_positive_polarity
##                            NA                      8.431e-01
##         min_positive_polarity          max_positive_polarity
##                    -4.422e-01                      4.504e-01
##         avg_negative_polarity          min_negative_polarity
##                     9.127e-01                     -5.208e-01
##         max_negative_polarity             title_subjectivity
##                    -3.686e-01                     -2.077e-01
##      title_sentiment_polarity         abs_title_subjectivity
##                     2.992e-01                     -3.089e-01
##  abs_title_sentiment_polarity
##                     5.959e-01
##
## Degrees of Freedom: 1883 Total (i.e. Null);  1833 Residual
## Null Deviance:       2612
## Residual Deviance: 2209  AIC: 2311
```

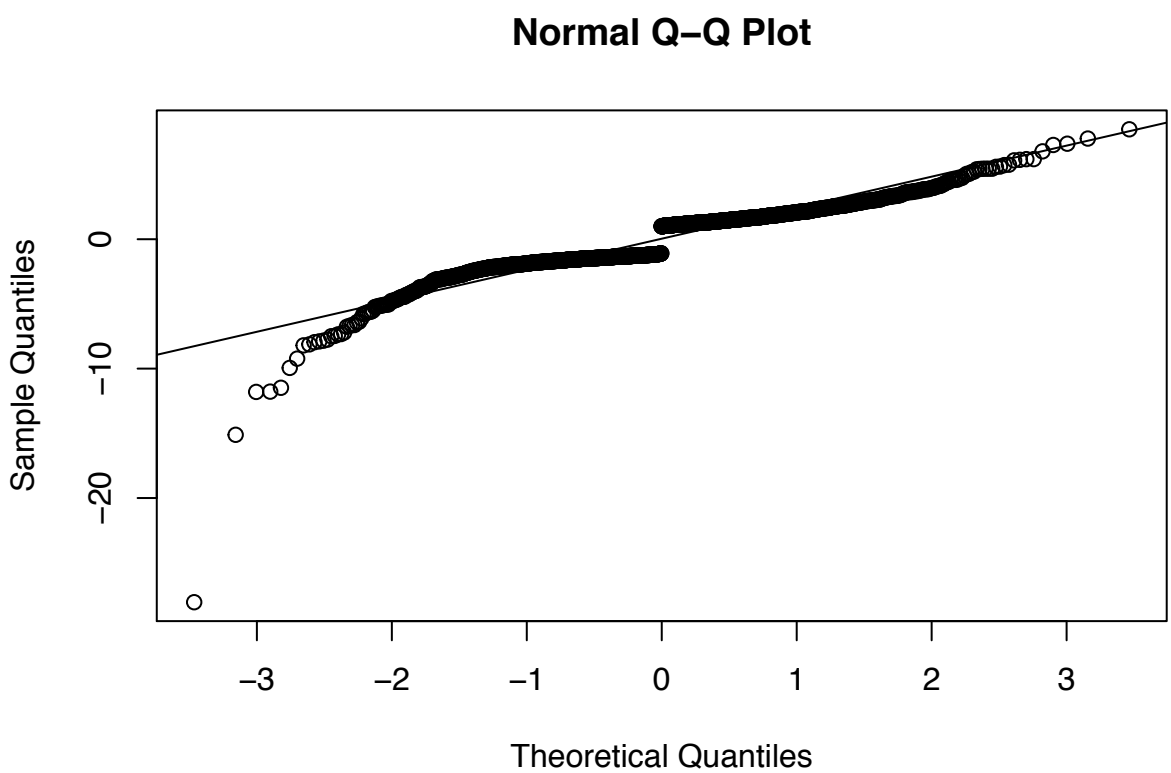
```r
summary(fit.lm.95.over)
```

```
##
```

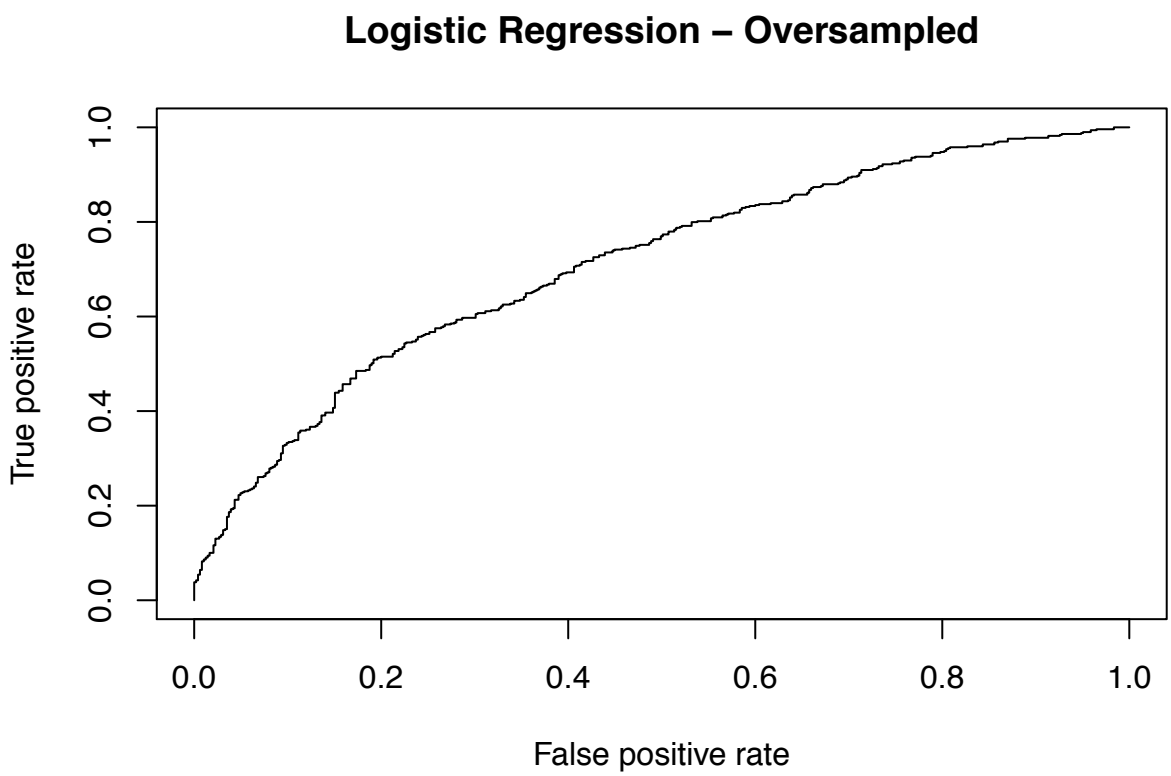
```
## Call:
## glm(formula = shares ~ ., family = binomial, data = news.train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.58226  -0.95417   0.02478   1.00535   2.06853
##
## Coefficients: (2 not defined because of singularities)
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   -1.678e+07  2.588e+07  -0.648 0.516785
## timedelta                      7.744e-04  3.700e-04   2.093 0.036350 *
## n_tokens_title                 5.385e-02  2.614e-02   2.060 0.039359 *
## n_tokens_content               1.450e-04  1.777e-04   0.816 0.414518
## n_unique_tokens                3.592e+00  1.935e+00   1.856 0.063433 .
## n_non_stop_words               1.678e+07  2.588e+07   0.648 0.516785
## n_non_stop_unique_tokens      -1.660e+00  1.440e+00  -1.153 0.249014
## num_hrefs                      1.314e-02  5.339e-03   2.461 0.013837 *
## num_self_hrefs                -4.012e-02  1.676e-02  -2.393 0.016715 *
## num_imgs                       1.162e-02  7.722e-03   1.505 0.132225
## num_videos                     8.480e-03  1.409e-02   0.602 0.547171
## average_token_length          -6.743e-01  2.215e-01  -3.044 0.002335 **
## num_keywords                  -8.875e-03  3.247e-02  -0.273 0.784612
## data_channel_is_lifestyle     -2.333e-01  3.206e-01  -0.728 0.466685
## data_channel_is_entertainment -4.947e-01  2.090e-01  -2.366 0.017964 *
## data_channel_is_bus           -2.676e-01  3.262e-01  -0.820 0.412096
## data_channel_is_socmed        -6.815e-02  3.236e-01  -0.211 0.833227
## data_channel_is_tech          -9.593e-02  3.074e-01  -0.312 0.754940
## data_channel_is_world         -4.341e-02  3.250e-01  -0.134 0.893733
## kw_min_min                     2.623e-05  1.334e-03   0.020 0.984320
## kw_max_min                    -2.753e-05  6.076e-05  -0.453 0.650457
## kw_avg_min                     9.939e-05  3.687e-04   0.270 0.787475
## kw_min_max                    -7.835e-07  1.158e-06  -0.677 0.498569
## kw_max_max                    -1.478e-07  5.012e-07  -0.295 0.768029
## kw_avg_max                    -9.430e-07  7.484e-07  -1.260 0.207637
## kw_min_avg                    -2.953e-04  6.690e-05  -4.414 1.01e-05 ***
## kw_max_avg                    -1.063e-04  3.049e-05  -3.488 0.000486 ***
## kw_avg_avg                     9.859e-04  1.410e-04   6.990 2.74e-12 ***
## self_reference_min_shares      8.307e-06  7.271e-06   1.142 0.253256
## self_reference_max_shares     -1.182e-06  3.777e-06  -0.313 0.754348
## self_reference_avg_sharess     3.758e-06  9.407e-06   0.399 0.689534
## is_weekend                     2.079e-01  1.473e-01   1.412 0.158040
## LDA_00                        -5.513e-01  4.007e-01  -1.376 0.168861
## LDA_01                        -7.084e-02  4.490e-01  -0.158 0.874635
## LDA_02                        -7.323e-01  4.232e-01  -1.730 0.083561 .
## LDA_03                        -3.042e-01  4.167e-01  -0.730 0.465444
## LDA_04                                 NA         NA      NA       NA
## global_subjectivity            1.337e+00  7.362e-01   1.816 0.069352 .
## global_sentiment_polarity     -3.711e-01  1.381e+00  -0.269 0.788115
## global_rate_positive_words    -3.513e+00  6.481e+00  -0.542 0.587767
## global_rate_negative_words    -9.717e+00  1.269e+01  -0.766 0.443754
## rate_positive_words           -7.165e-01  1.012e+00  -0.708 0.478834
## rate_negative_words                    NA         NA      NA       NA
## avg_positive_polarity          8.431e-01  1.188e+00   0.710 0.477892
## min_positive_polarity         -4.422e-01  9.891e-01  -0.447 0.654836
```

```
## max_positive_polarity          4.504e-01  3.822e-01   1.178 0.238663
## avg_negative_polarity          9.127e-01  1.110e+00   0.823 0.410704
## min_negative_polarity         -5.208e-01  4.045e-01  -1.288 0.197893
## max_negative_polarity         -3.686e-01  9.274e-01  -0.397 0.691046
## title_subjectivity            -2.077e-01  2.320e-01  -0.895 0.370540
## title_sentiment_polarity       2.992e-01  2.120e-01   1.411 0.158229
## abs_title_subjectivity        -3.089e-01  3.191e-01  -0.968 0.333009
## abs_title_sentiment_polarity   5.959e-01  3.402e-01   1.752 0.079803 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2611.8  on 1883  degrees of freedom
## Residual deviance: 2208.5  on 1833  degrees of freedom
## AIC: 2310.5
##
## Number of Fisher Scoring iterations: 6
```

```r
## Q-Q plot
qqnorm(fit.lm.95.over$residuals)
qqline(fit.lm.95.over$residuals)
```




```r
## ROC Curves
scores <- predict(fit.lm.95.over, newdata=news.validation,
                  type = "response")
prob <- prediction(scores, labels=news.validation$shares)
perf <- performance(prob, "tpr", "fpr")
plot(perf, colorize=F, main="Logistic Regression - Oversampled")
```




```
## AUC
auc.fit.lm.95.over <- unlist(attributes(performance(prob, "auc"))$y.values)
auc.fit.lm.95.over
```

```
## [1] 0.71008
```

```
## Confusion Matrix
pred.fit.lm.95.over <- ifelse(scores>0.5, 1, 0)
confusionMatrix(pred.fit.lm.95.over, news.validation$shares,
                positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 339 201
##          1 146 298
##
##                Accuracy : 0.6474
##                  95% CI : (0.6166, 0.6772)
##     No Information Rate : 0.5071
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.2957
##  Mcnemar's Test P-Value : 0.003745
##
##             Sensitivity : 0.5972
##             Specificity : 0.6990
```

```
##          Pos Pred Value : 0.6712
##          Neg Pred Value : 0.6278
##              Prevalence : 0.5071
##          Detection Rate : 0.3028
##    Detection Prevalence : 0.4512
##       Balanced Accuracy : 0.6481
##
##        'Positive' Class : 1
##
```

```
## Prediction Error on Validation Set
mse.fit.lm.95.over = mean((news.validation$shares -
                              pred.fit.lm.95.over)^2)
mse.fit.lm.95.over
```

```
## [1] 0.3526423
```

```
######################################################################
## Testing with balanced Validation set
news.validation.popular <- news.validation[news.validation[, 53] == 1, ]

news.validation.unpopular <- news.validation[news.validation[, 53] == 0, ]
dim(news.validation.popular)
```

```
## [1] 499  53
```

```
dim(news.validation.unpopular)
```

```
## [1] 485  53
```

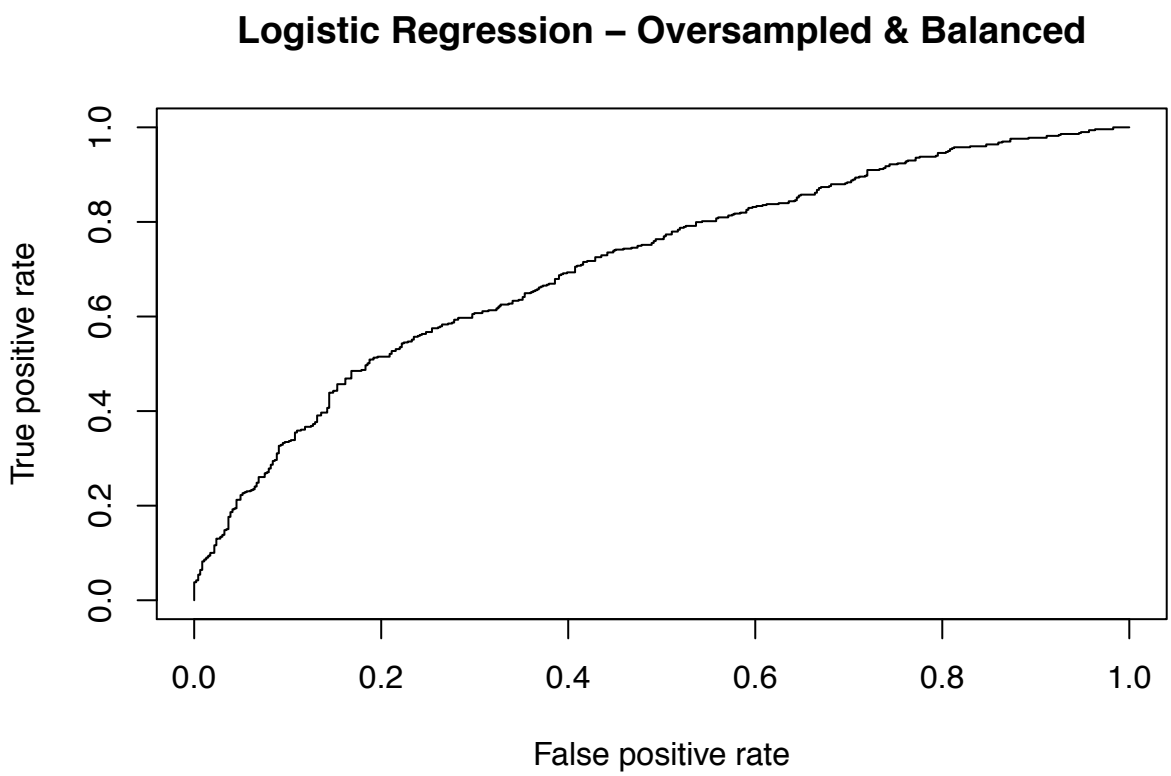
```
news.validation.unpopular <- news.validation.unpopular[sample(nrow(news.validation.unpopular), 464), ]
dim(news.validation.unpopular)
```

```
## [1] 464  53
```

```
news.validation <- rbind(news.validation.popular, news.validation.unpopular)
dim(news.validation)
```

```
## [1] 963  53
```

```
## ROC Curves
scores <- predict(fit.lm.95.over, newdata=news.validation,
                  type = "response")
prob <- prediction(scores, labels=news.validation$shares)
perf <- performance(prob, "tpr", "fpr")
plot(perf, colorize=F, main="Logistic Regression - Oversampled & Balanced")
```




```
## AUC
auc.fit.lm.95.correct <- unlist(attributes(performance(prob, "auc"))$y.values)
auc.fit.lm.95.correct
```

```
## [1] 0.709976
```

```
## Confusion Matrix
pred.fit.lm.95.correct <- ifelse(scores>0.5, 1, 0)
confusionMatrix(pred.fit.lm.95.correct, news.validation$shares,
                positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 326 201
##          1 138 298
##
##                Accuracy : 0.648
##                  95% CI : (0.6169, 0.6782)
##     No Information Rate : 0.5182
##     P-Value [Acc > NIR] : 2.74e-16
##
##                   Kappa : 0.2984
##  Mcnemar's Test P-Value : 0.0007589
##
##             Sensitivity : 0.5972
##             Specificity : 0.7026
```

```
##          Pos Pred Value : 0.6835
##          Neg Pred Value : 0.6186
##              Prevalence : 0.5182
##          Detection Rate : 0.3094
##    Detection Prevalence : 0.4528
##       Balanced Accuracy : 0.6499
##
##        'Positive' Class : 1
##
```

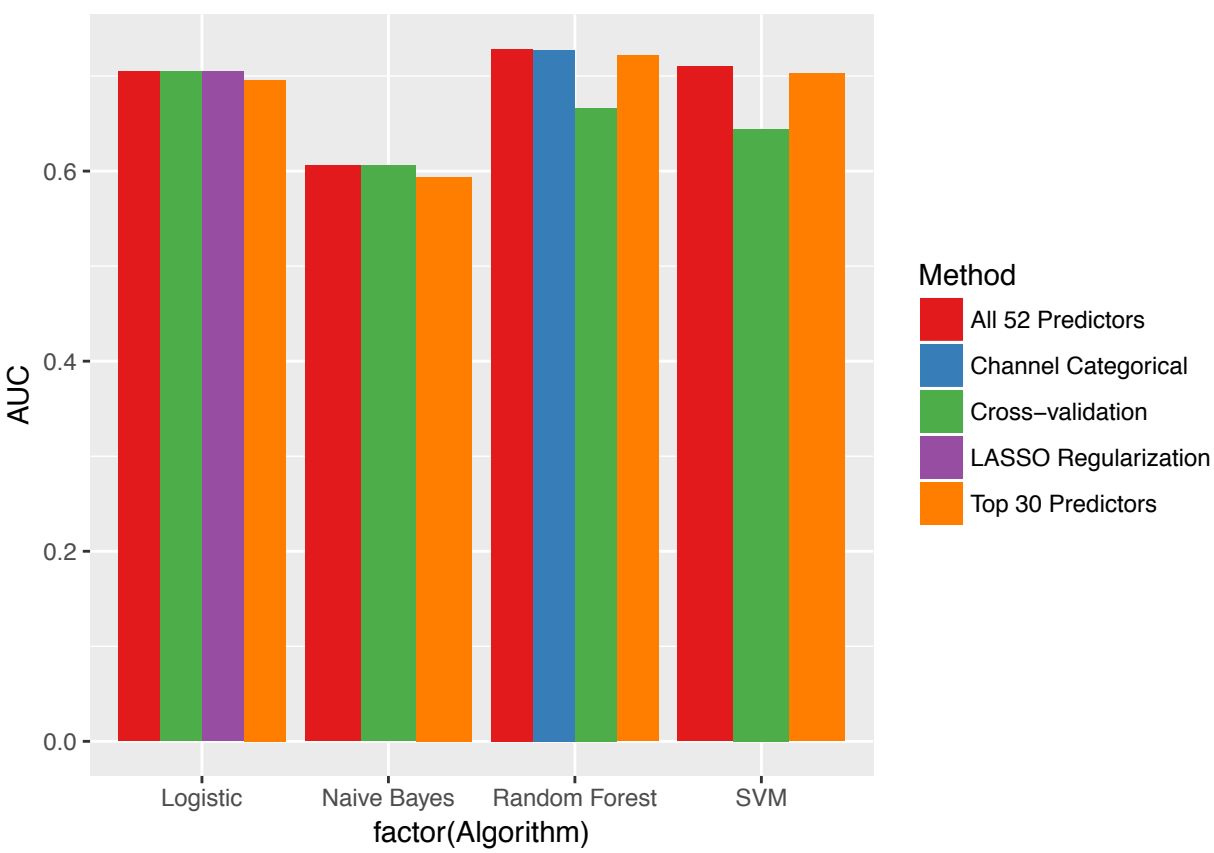
```
## Prediction Error on Validation Set
mse.fit.lm.95.correct = mean((news.validation$shares -
                              pred.fit.lm.95.correct)^2)
mse.fit.lm.95.correct
```

```
## [1] 0.3520249
```

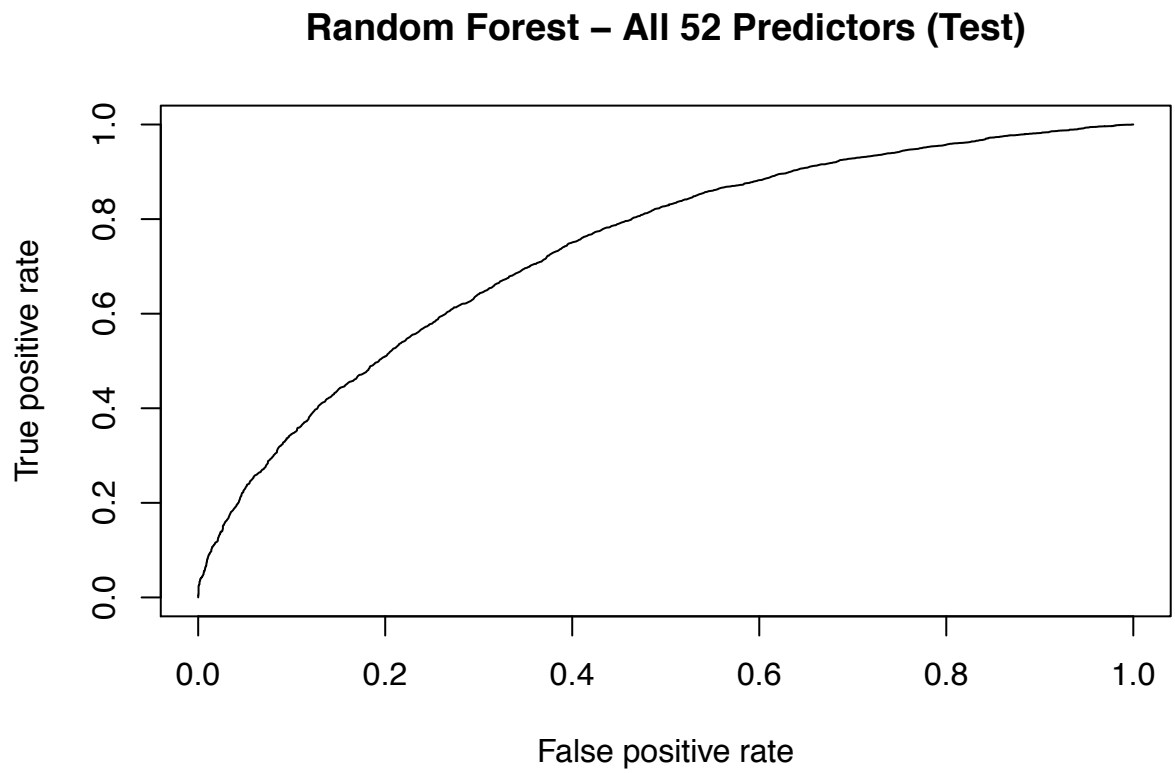
5 Results

```
#######################################################################
# Results
#######################################################################
## AUC Comparison plot
results.df <- data.frame(c("Logistic", "Logistic", "Logistic", "Logistic", "SVM",
                           "SVM", "SVM", "Random Forest", "Random Forest",
                           "Random Forest", "Random Forest", "Naive Bayes",
                           "Naive Bayes", "Naive Bayes"),
                          c("All 52 Predictors", "Top 30 Predictors", "LASSO Regularization",
                            "Cross-validation", "All 52 Predictors", "Top 30 Predictors",
                            "Cross-validation", "All 52 Predictors", "Top 30 Predictors",
                            "Channel Categorical", "Cross-validation", "All 52 Predictors",
                            "Top 30 Predictors", "Cross-validation"),
                         c(auc.fit.lm.all, auc.fit.lm.top, auc.refit.lasso,
                           auc.fit.lm.cv, auc.fit.svm.all, auc.fit.svm.top,
                           auc.fit.svm.cv, auc.fit.rf.1800, auc.fit.rf.top,
                           auc.fit.rf.cat, auc.fit.rf.cv, auc.fit.nb.all,
                           auc.fit.nb.top, auc.fit.nb.cv))

names(results.df) <- c("Algorithm", "Method", "AUC")
ggplot(results.df, aes(factor(Algorithm), AUC, fill = Method)) +
  geom_bar(stat="identity", position = "dodge") +
  scale_fill_brewer(palette = "Set1")
```



```r
## Test set on best model (Random Forest - All 52 Predictors)
## ROC Curves
scores <- predict(fit.rf.1800, newdata= news.test, type = "prob")[, 2]
prob <- prediction(scores, labels=news.test$shares)
perf <- performance(prob, "tpr", "fpr")
plot(perf, colorize=F, main="Random Forest - All 52 Predictors (Test)")
```




```r
## AUC
auc.fit.rf.1800.test <- unlist(attributes(performance(prob, "auc"))$y.values)
auc.fit.rf.1800.test
```

```
## [1] 0.7382278
```

```r
## Confusion Matrix
pred.fit.rf.1800.test <- ifelse(scores>0.5, 1, 0)
confusionMatrix(pred.fit.rf.1800.test, news.test$shares, positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 3286 1574
##          1 1556 3122
##
##                Accuracy : 0.6718
##                  95% CI : (0.6623, 0.6813)
##     No Information Rate : 0.5077
```

```
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.3435
##  Mcnemar's Test P-Value : 0.7612
##
##             Sensitivity : 0.6648
##             Specificity : 0.6786
##          Pos Pred Value : 0.6674
##          Neg Pred Value : 0.6761
##              Prevalence : 0.4923
##          Detection Rate : 0.3273
##    Detection Prevalence : 0.4905
##       Balanced Accuracy : 0.6717
##
##        'Positive' Class : 1
##
```

```
## Prediction Error on Validation Set
mse.fit.rf.1800.test <- mean((news.test$shares-pred.fit.rf.1800.test)^2)
mse.fit.rf.1800.test
```

```
## [1] 0.328161
```