Report

2020/12/13

```
library(corrplot)

library(car)

library(knitr)
library(plot.matrix)
library(class)

library(MASS)

library(leaps)

library(caret)

library(nnet)

library(dplyr)

library(randomForest)

library(gbm)

library(glmnet)

library(lubridate)
```

Preprocessing of the Data

```
testing <- read.csv("test.csv")
training <- read.csv("training.csv")
training$PublishedDate <- mdy_hm(training$PublishedDate)
training$month <- month(training$PublishedDate)
training$day<- day(training$PublishedDate)
training$hour<- hour(training$PublishedDate)
training$minute<- minute(training$PublishedDate)

testing$PublishedDate <- mdy_hm(testing$PublishedDate)
testing$month <- month(testing$PublishedDate)
testing$day<- day(testing$PublishedDate)
testing$hour<- hour(testing$PublishedDate)
testing$minute<- minute(testing$PublishedDate)


set.seed(123456)
index <- sample(seq_len(nrow(training)), size = 0.8 * nrow(training))
train <- training[index,-c(1,2)]
train <- na.omit(train)
test <- training[-index,-c(1,2)]
```

Statisctic Model Selection

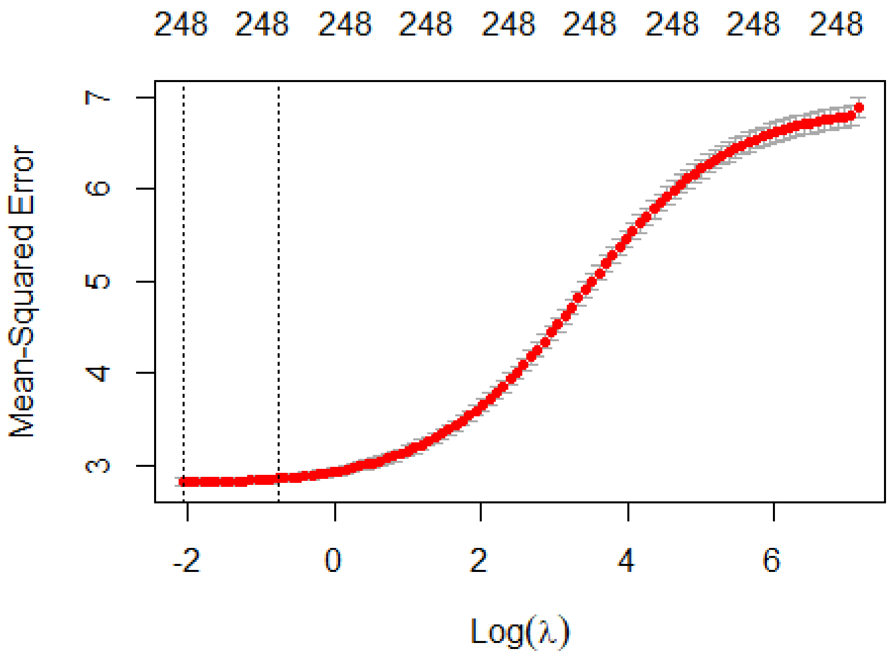
GLM

```
model_glm <- glm(train$growth_2_6 ~ ., data = train)
yhat.glm <- predict(model_glm, newdata = test)

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if
(type == :
## prediction from a rank-deficient fit may be misleading

glm.err <- mean((yhat.glm - test$growth_2_6)^2)
```

GLM model MSE: 2.5475668

Ridge

```
library(glmnet)
xtrain <- model.matrix(growth_2_6~., data = train)
ytrain <- train$growth_2_6
xtest <- model.matrix(growth_2_6~., data = test)
ytest <- test$growth_2_6

ridge.fit <- cv.glmnet(xtrain,ytrain,alpha = 0)
plot(ridge.fit)
```

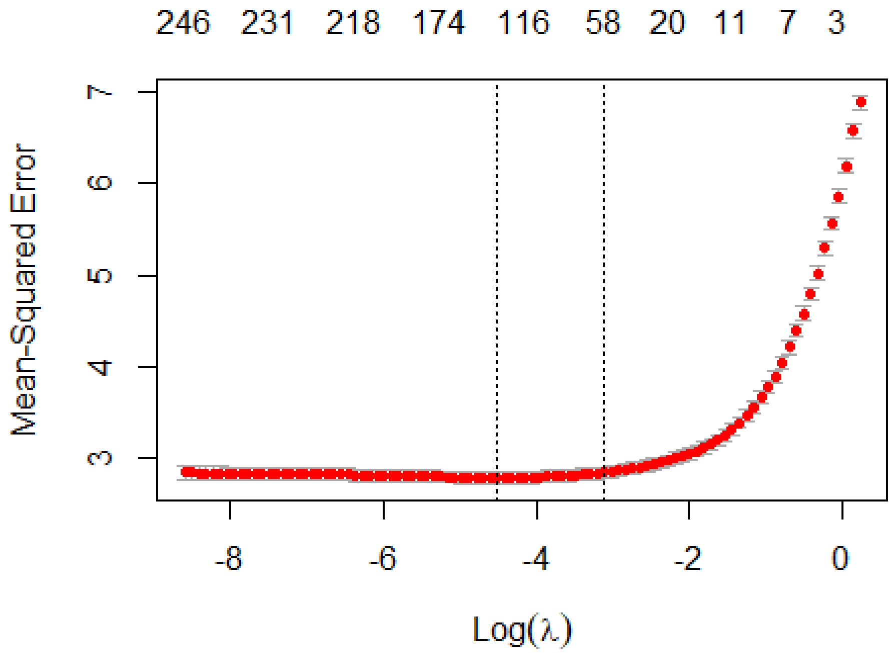


```r
ridge.lambda <- ridge.fit$lambda.min

ridge.pred <- predict(ridge.fit, s = ridge.lambda, newx = xtest)
ridge.err <- mean((ridge.pred - ytest)^2)
```

Ridge test MSE: 2.5400127.

Lasso

```r
lasso.fit <- cv.glmnet(xtrain,ytrain,alpha = 1)
plot(lasso.fit)
```

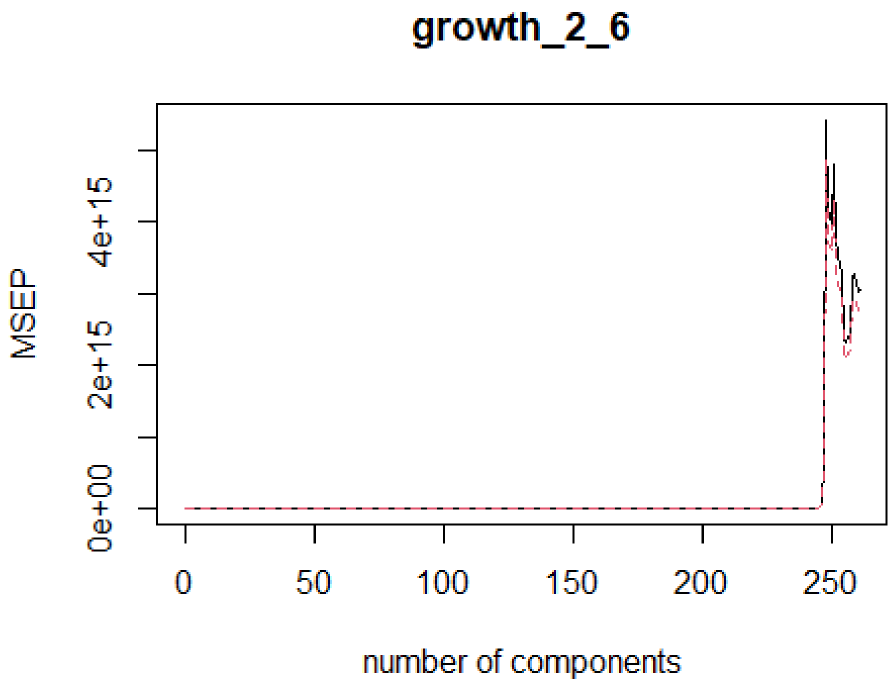


```r
lasso.lambda <- lasso.fit$lambda.min
#lasso.lambda

lasso.pred <- predict(lasso.fit, s = lasso.lambda, newx = xtest)
lasso.err <- mean((lasso.pred - ytest)^2)
```

Lasso test MSE: 2.5599742

PCR

```r
library(pls)

pcr.fit <- pcr(growth_2_6~.,data = train, scale= FALSE, validation = "C
V")
validationplot(pcr.fit, val.type = "MSEP")
```

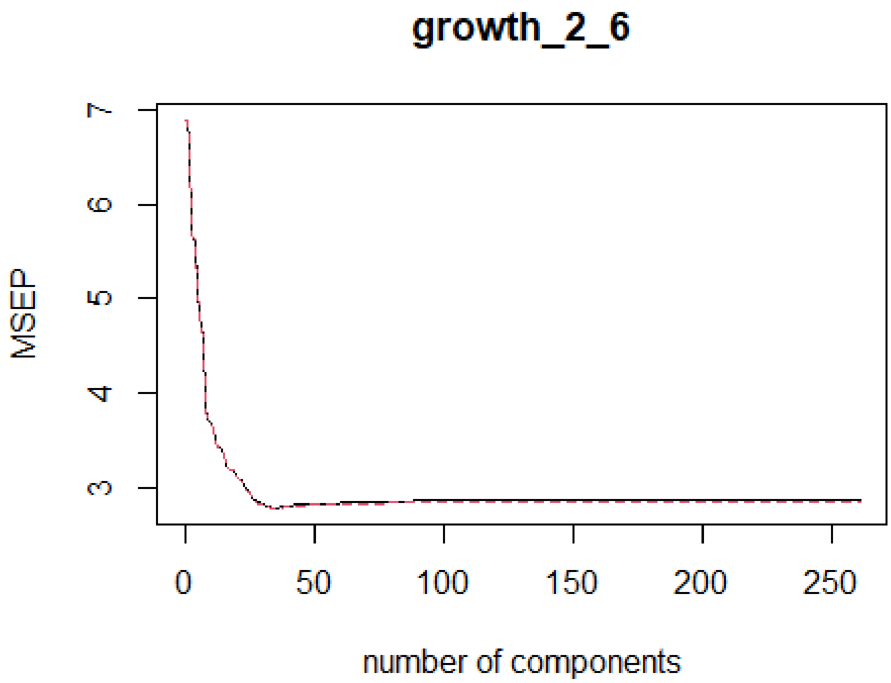


```
#summary(pcr.fit)
pcr.pred <- predict(pcr.fit, test, ncomp = 109)
pcr.err = mean((pcr.pred - test$growth_2_6)^2)
```

PCR test error rate : 2.5657244.

PLS

```
pls.fit <- plsr(growth_2_6~.,data = train, scale= FALSE, validation = "
CV")
validationplot(pls.fit, val.type = "MSEP")
```

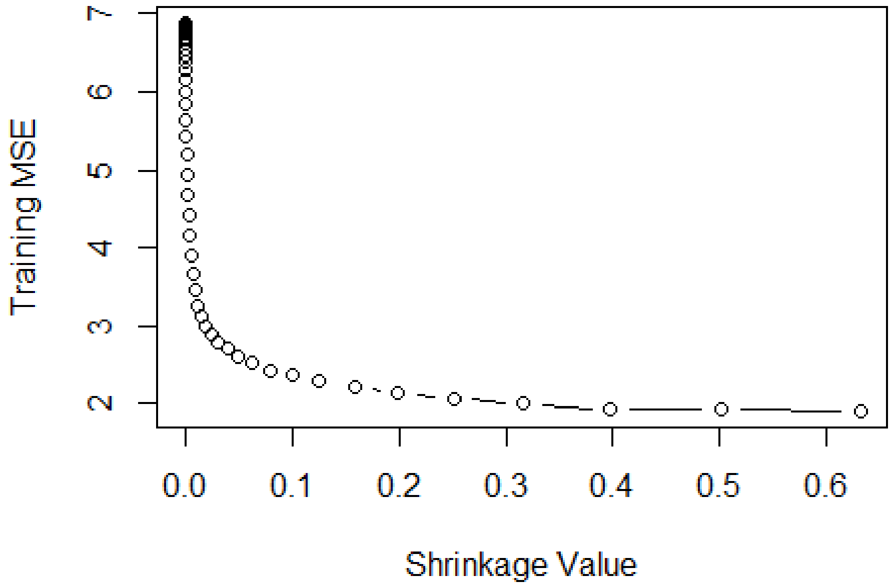


```
#summary(pls.fit)
pls.pred <- predict(pls.fit, test, ncomp = 34)
pls.err = mean((pls.pred - test$growth_2_6)^2)
```

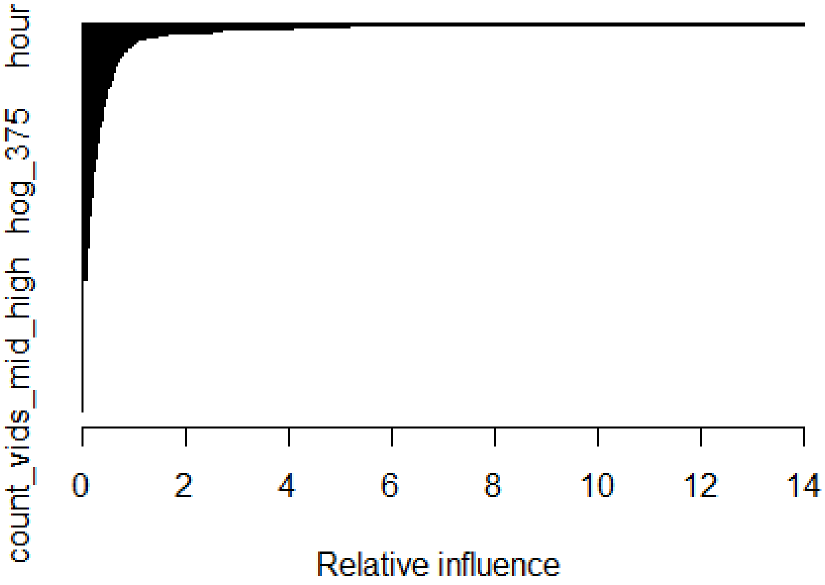
PLS test error rate : 2.5387545.

Boosting

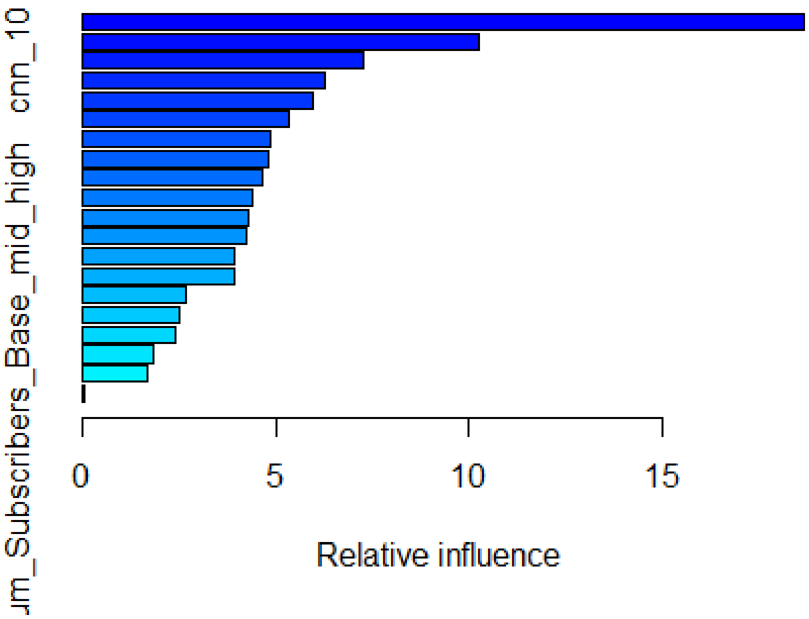
```
library(gbm)
set.seed(123)
power <- seq(-10, -0.2, by = 0.1)
lambda <- 10^power
trainMSE <- rep(NA, length(lambda))
for (i in 1:length(lambda)){
  boost <- gbm(growth_2_6~., data = train, distribution = "gaussian",
n.trees = 500,verbose = FALSE, shrinkage = lambda[i])
  pred.train <- predict(boost, train, n.trees = 1000)
  trainMSE[i] <- mean((pred.train - train$growth_2_6)^2)
}

plot(lambda, trainMSE, type = "b", xlab = "Shrinkage Value", ylab = "Tr
aining MSE")
```



```
#min(trainMSE)
#lambda[which.min(trainMSE)]

model_gbm <- gbm(growth_2_6~., data = train, distribution = "gaussian",
 n.trees = 500,  shrinkage = lambda[which.min(trainMSE)])

yhat.gbm <- predict(model_gbm, newdata = test)

## Using 500 trees...

gbm.err <- mean((yhat.gbm - test$growth_2_6)^2)

a <- summary(model_gbm)
```



```
gbm_x <- head(a,20)[,1]
model_gbm1 <- gbm(growth_2_6~., data = train[,c(gbm_x, "growth_2_6")],
distribution = "gaussian", n.trees = 500,  shrinkage = lambda[which.min
(trainMSE)])

yhat.gbm1 <- predict(model_gbm1, newdata = test)

gbm1.err <- mean((yhat.gbm1 - test$growth_2_6)^2)

summary(model_gbm1)
```



```
##                                                               var      rel.
inf
## cnn_17                                                     cnn_17 18.64845
190
## Num_Views_Base_mid_high                   Num_Views_Base_mid_high 10.22589
007
## cnn_10                                                     cnn_10  7.27887
473
## avg_growth_low                                     avg_growth_low  6.26273
596
## cnn_89                                                     cnn_89  5.94869
388
## pct_nonzero_pixels                             pct_nonzero_pixels  5.33038
857
## hog_643                                                   hog_643  4.87490
171
## avg_growth_low_mid                             avg_growth_low_mid  4.82927
820
## num_words                                               num_words  4.63475
139
## cnn_68                                                     cnn_68  4.39411
512
## views_2_hours                                       views_2_hours  4.30614
076
## cnn_25                                                     cnn_25  4.26066
982
## Duration                                                 Duration  3.93968
```

```
681
## cnn_12                                                   cnn_12  3.91091
219
## hog_492                                                 hog_492  2.65284
802
## hour                                                       hour  2.50540
482
## cnn_86                                                   cnn_86  2.38794
966
## Num_Subscribers_Base_low_mid   Num_Subscribers_Base_low_mid  1.84216
823
## avg_growth_mid_high                         avg_growth_mid_high  1.70365
198
## Num_Subscribers_Base_mid_high Num_Subscribers_Base_mid_high  0.06248
617
```

Boosted model MSE: 3.078268.

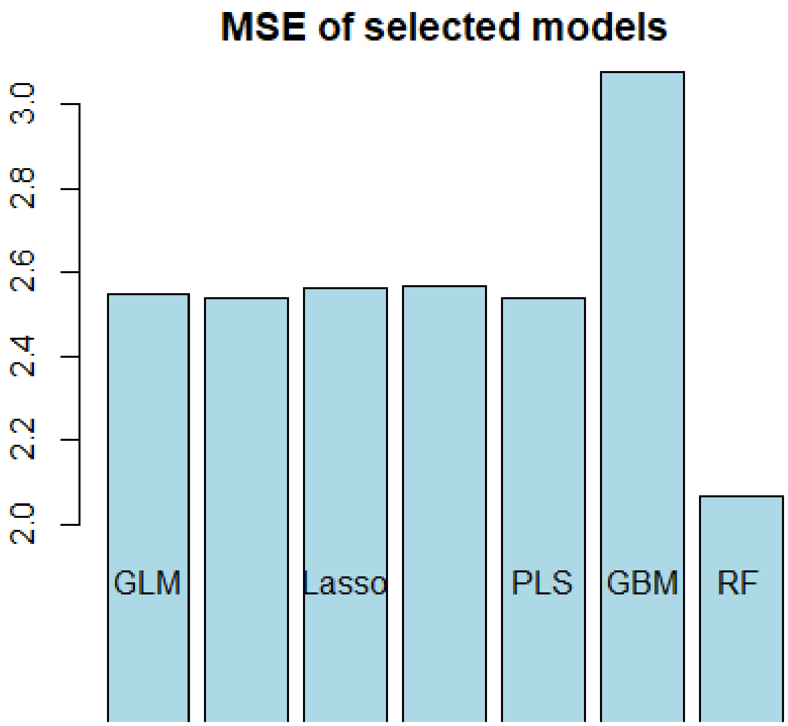
Random Forest

```
library(randomForest)
model_rf <- randomForest(growth_2_6~., data = train, mtry = 262/3, ntre
e= 2000, importance = TRUE) # 2.10

## Warning in randomForest.default(m, y, ...): invalid mtry: reset to w
ithin valid
## range

yhatrf <- predict(model_rf, newdata = test)
rf.err <- mean((yhatrf - test$growth_2_6)^2)
```

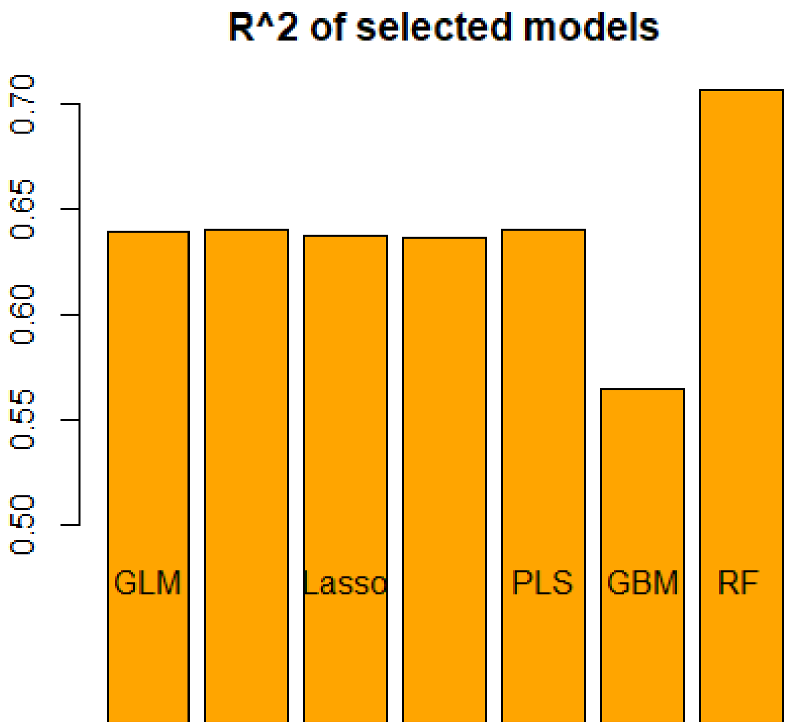
Random forest model MSE: 2.067357.

Summary

```
result <- c(glm.err, ridge.err,lasso.err,pcr.err,pls.err,gbm.err, rf.er
r)
barplot(result,
        names.arg = c("GLM", "Ridge","Lasso", "PCR", "PLS", "GBM", "RF
"),
        ylim = c(2,3),
        col = "lightblue",
        main = "MSE of selected models",
        axes = TRUE)
```



```r
sst <- mean((mean(test$growth_2_6) - test$growth_2_6)^2)

r2 <- c()
for ( i in 1:length(result)){
  r2 <- c(r2, 1 - result[i]/sst)
}
barplot(r2,
        names.arg = c("GLM", "Ridge","Lasso", "PCR", "PLS", "GBM", "RF"),
        col = "ORANGE",
        ylim = c(0.5,0.7),
        main = "R^2 of selected models")
```



Predictor Selection

High correlation

```
cor <- abs(cor(train$growth_2_6,train[,-258]))

pick <-  which(cor > 0.2)
length(pick)

## [1] 19

high_cor <- colnames(train)[pick]

correlationMatrix <- cor(train[,pick])


x <- c(gbm_x, high_cor)
length(x)

## [1] 39

for (i in 1:length(x)){
  for (j in 1:(i-1)) {
    if (x[i] == x[j]){
      x[i] = 0
      break
    }
  }
}
x <- x[-which(x == 0)]
x
```

```
##  [1] "cnn_17"                        "avg_growth_low"               
##  [3] "avg_growth_low_mid"            "cnn_10"                       
##  [5] "cnn_89"                        "num_words"                    
##  [7] "Num_Subscribers_Base_mid_high" "views_2_hours"                
##  [9] "hour"                          "cnn_12"                       
## [11] "Duration"                      "Num_Subscribers_Base_low_mid" 
## [13] "cnn_68"                        "cnn_86"                       
## [15] "avg_growth_mid_high"           "hog_643"                      
## [17] "hog_492"                       "cnn_25"                       
## [19] "pct_nonzero_pixels"            "doc2vec_17"                   
## [21] "num_chars"                     "num_uppercase_chars"          
## [23] "Num_Subscribers_Base_low"      "Num_Views_Base_low"           
## [25] "Num_Views_Base_low_mid"        "Num_Views_Base_mid_high"      
## [27] "count_vids_mid_high"
```

```
model_1.1 <- randomForest(growth_2_6~., data = train[,c(x,"growth_2_6
")], mtry = 27/3, ntree = 500)

summary(model_1.1)
```

```
##                 Length Class  Mode     
## call               5   -none- call     
## type               1   -none- character
## predicted       5793   -none- numeric  
## mse              500   -none- numeric  
## rsq              500   -none- numeric  
## oob.times       5793   -none- numeric  
## importance        27   -none- numeric  
## importanceSD       0   -none- NULL     
## localImportance    0   -none- NULL     
## proximity          0   -none- NULL     
## ntree              1   -none- numeric  
## mtry               1   -none- numeric  
## forest            11   -none- list     
## coefs              0   -none- NULL     
## y               5793   -none- numeric  
## test               0   -none- NULL     
## inbag              0   -none- NULL     
## terms              3   terms  call
```

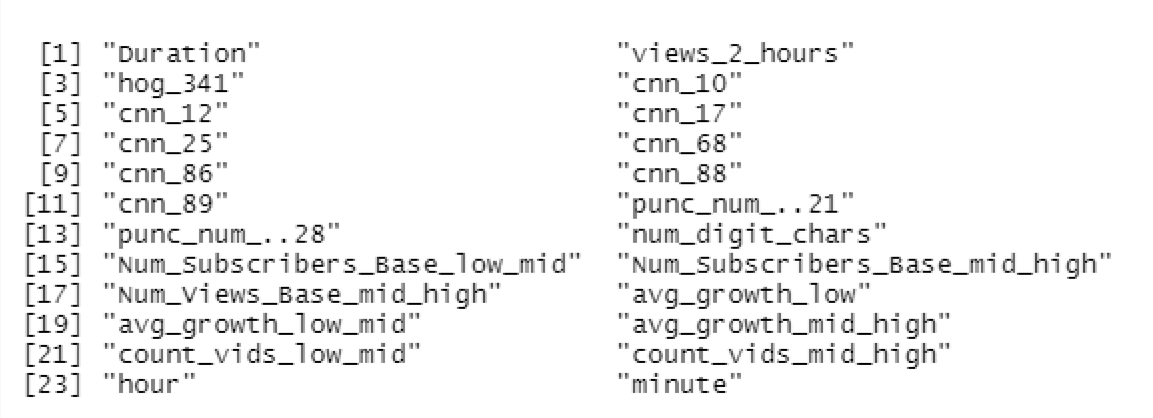
```
yhat.1.1 <- predict(model_1.1, newdata = test)
mse1.1 <- mean((yhat.1.1 - test$growth_2_6)^2)
mse1.1
```

```
## [1] 2.089632
```

MSE: 2.0896315

Importance

```
summary(importance(model_rf))

##     %IncMSE        IncNodePurity
##  Min.   : -2.643   Min.   :   0.00
##  1st Qu.:  2.318   1st Qu.:  40.38
##  Median :  4.566   Median :  55.47
##  Mean   :  9.625   Mean   : 151.59
##  3rd Qu.:  7.066   3rd Qu.:  80.71
##  Max.   :129.356   Max.   :7007.10

rf_imp <- which(importance(model_rf)[,1]>mean(importance(model_rf)[,1])
& importance(model_rf)[,2]>mean(importance(model_rf)[,2]))
rf_imp <- rownames(importance(model_rf))[rf_imp]

rf_imp
```



```
 [1] "Duration"                        "views_2_hours"
 [3] "hog_341"                         "cnn_10"
 [5] "cnn_12"                          "cnn_17"
 [7] "cnn_25"                          "cnn_68"
 [9] "cnn_86"                          "cnn_88"
[11] "cnn_89"                          "punc_num_..21"
[13] "punc_num_..28"                   "num_digit_chars"
[15] "Num_Subscribers_Base_low_mid"    "Num_Subscribers_Base_mid_high"
[17] "Num_Views_Base_mid_high"         "avg_growth_low"
[19] "avg_growth_low_mid"              "avg_growth_mid_high"
[21] "count_vids_low_mid"              "count_vids_mid_high"
[23] "hour"                            "minute"
```

```
set.seed(123)
model_1.2 <- randomForest(growth_2_6~., data = train[,c(rf_imp,"growth_
2_6")], mtry = 24/3, ntree = 500) # 1.988

yhat.1.2 <- predict(model_1.2, newdata = test)
mse1.2 <- mean((yhat.1.2 - test$growth_2_6)^2)
mse1.2

## [1] 1.984859
```