

a. Read the dataset and identify the right feature

Read train dataset

```
train = fread("blogData_train.csv")
```

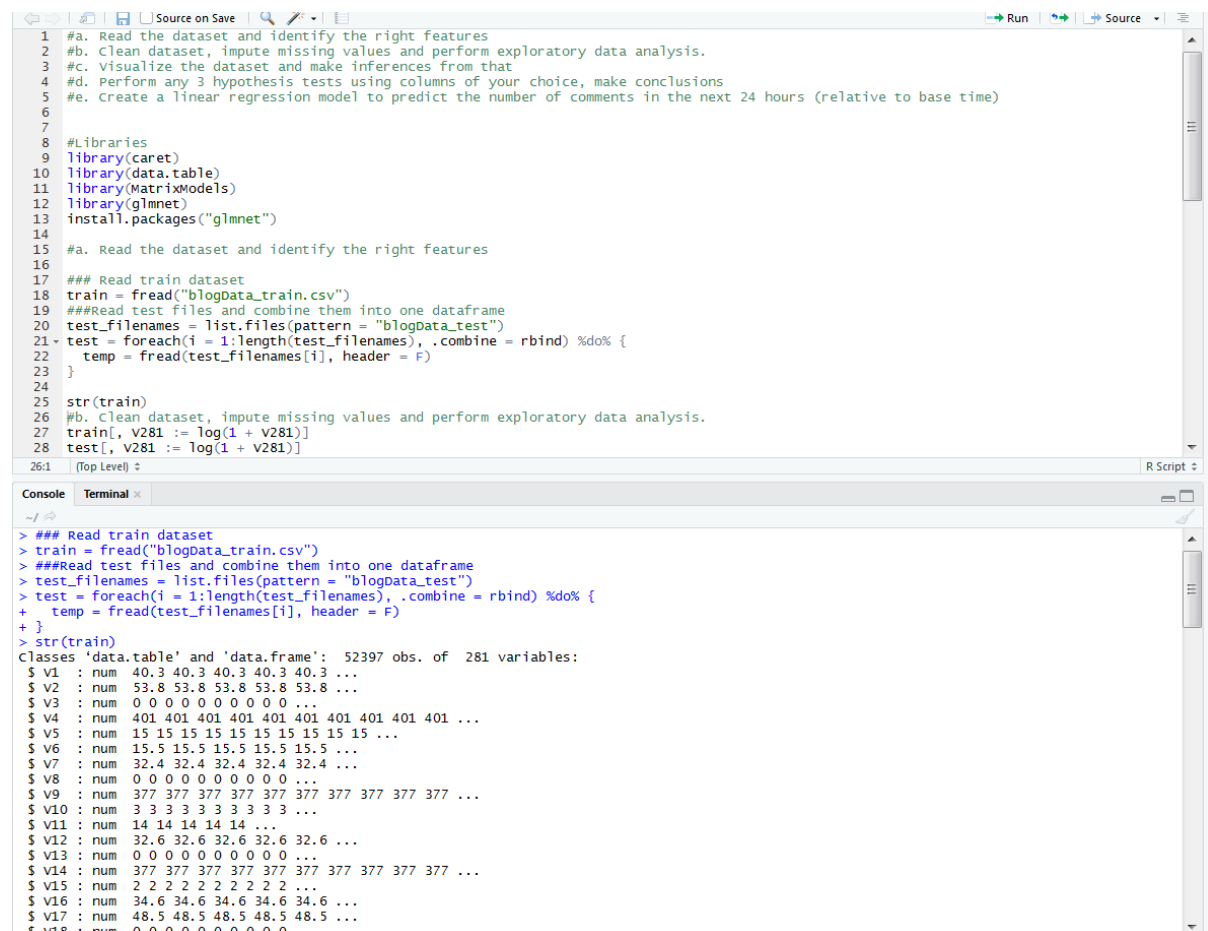
Read test files and combine them into one dataframe

```
test_filenames = list.files(pattern = "blogData_test")
```

```
test = foreach(i = 1:length(test_filenames), .combine = rbind) %do% {
```

```
  temp = fread(test_filenames[i], header = F)
```

```
}
```



```
1 #a. Read the dataset and identify the right features
2 #b. Clean dataset, impute missing values and perform exploratory data analysis.
3 #c. Visualize the dataset and make inferences from that
4 #d. Perform any 3 hypothesis tests using columns of your choice, make conclusions
5 #e. Create a linear regression model to predict the number of comments in the next 24 hours (relative to base time)
6
7
8 #Libraries
9 library(caret)
10 library(data.table)
11 library(MatrixModels)
12 library(glmnet)
13 install.packages("glmnet")
14
15 #a. Read the dataset and identify the right features
16
17 ### Read train dataset
18 train = fread("blogData_train.csv")
19 ### Read test files and combine them into one dataframe
20 test_filenames = list.files(pattern = "blogData_test")
21 test = foreach(i = 1:length(test_filenames), .combine = rbind) %do% {
22   temp = fread(test_filenames[i], header = F)
23 }
24
25 str(train)
26 #b. Clean dataset, impute missing values and perform exploratory data analysis.
27 train[, V281 := log(1 + V281)]
28 test[, V281 := log(1 + V281)]
```

```
> ### Read train dataset
> train = fread("blogData_train.csv")
> ### Read test files and combine them into one dataframe
> test_filenames = list.files(pattern = "blogData_test")
> test = foreach(i = 1:length(test_filenames), .combine = rbind) %do% {
+   temp = fread(test_filenames[i], header = F)
+ }
> str(train)
Classes 'data.table' and 'data.frame': 52397 obs. of 281 variables:
 $ V1 : num 40.3 40.3 40.3 40.3 40.3 ...
 $ V2 : num 53.8 53.8 53.8 53.8 53.8 ...
 $ V3 : num 0 0 0 0 0 0 0 0 0 ...
 $ V4 : num 401 401 401 401 401 401 401 401 401 ...
 $ V5 : num 15 15 15 15 15 15 15 15 15 ...
 $ V6 : num 15.5 15.5 15.5 15.5 15.5 15.5 ...
 $ V7 : num 32.4 32.4 32.4 32.4 32.4 ...
 $ V8 : num 0 0 0 0 0 0 0 0 0 ...
 $ V9 : num 377 377 377 377 377 377 377 377 377 ...
 $ V10 : num 3 3 3 3 3 3 3 3 3 ...
 $ V11 : num 14 14 14 14 14 ...
 $ V12 : num 32.6 32.6 32.6 32.6 32.6 ...
 $ V13 : num 0 0 0 0 0 0 0 0 0 ...
 $ V14 : num 377 377 377 377 377 377 377 377 377 ...
 $ V15 : num 2 2 2 2 2 2 2 2 2 ...
 $ V16 : num 34.6 34.6 34.6 34.6 34.6 ...
 $ V17 : num 48.5 48.5 48.5 48.5 48.5 ...
 $ V18 : num 0 0 0 0 0 0 0 0 0 ...
```

b. Clean dataset, impute missing values and perform exploratory data

#b. Clean dataset, impute missing values and perform exploratory data analysis.

```
train[, V281 := log(1 + V281)]
```

```
test[, V281 := log(1 + V281)]
```

drop continuous variables without variation

```
drop = c(8, 13, 28, 33, 38, 40, 43, 50, 278)
```

```
train[, (drop) := NULL]
```

test[, (drop) := NULL]

```
20 test_filenames = list.files(pattern = "blogData_test")
21 test = foreach(i = 1:length(test_filenames), .combine = rbind) %do% {
22   temp = fread(test_filenames[i], header = F)
23 }
24
25 str(train)
26 #b. Clean dataset, impute missing values and perform exploratory data analysis.
27 train[, v281 := log(1 + v281)]
28 test[, v281 := log(1 + v281)]
29 # drop continuous variables without variation
30 drop = c(8, 13, 28, 33, 38, 40, 43, 50, 278)
31 train[, (drop) := NULL]
32 test[, (drop) := NULL]
33
34 #c. Visualize the dataset and make inferences from that
35
36 view(train)
37
38 #Our target variable for prediction is v281(which is the number of comments in the next 24hrs)
39 #Other variables are statistical information (mean,sd,min,max) of some variables on the original blog posts
40
41 #d. Perform any 3 hypothesis tests using columns of your choice, make conclusions
42
43 chisq.test(table(train$v1, train$v2))
44
45 t.test(train$v4, train$v280, paired=TRUE)
46
47
48 #e. Create a linear regression model to predict the number of comments in the next 24 hours (relative to base time)
49 mse = function(y_hat, y) {
50   mse = mean((y - y_hat)^2)
51   return(mse)
52 }
53 # create design matrices
54
55 train_x = model.matrix(X0.0.57 ~ . - 1, data = blogTrain, sparse = F)
56 train_x_sparse = model.matrix(X0.0.57 ~ . - 1, data = blogTrain, sparse = T)
57 train_y = blogTrain$X0.57
58
59 test_x = model.matrix(v281 ~ . - 1, data = blogTest, sparse = F)
60 test_y = blogTest$v281
61
62 # Linear Model Using LASSO
63 mdl_lasso = cv.glmnet(train_x_sparse, train_y, family = "gaussian", alpha = 1)
64 pred_lasso = predict(mdl_lasso, newx = test_x)
65 mae(pred_lasso, test_y)
66
0:11 (Top Level) ↕
```

insol

Terminal ✕

```
#b. Clean dataset, impute missing values and perform exploratory data analysis.
train[, v281 := log(1 + v281)]
test[, v281 := log(1 + v281)]
# drop continuous variables without variation
drop = c(8, 13, 28, 33, 38, 40, 43, 50, 278)
train[, (drop) := NULL]
test[, (drop) := NULL]
```

c. Visualize the dataset and make inferences from that

View(train)

#Our target variable for prediction is V281(which is the number of comments in the next 24hrs)

#Other variables are statistical information (mean,sd,min,max) of some variables on the original blog posts

d. Perform any 3 hypothesis tests using columns of your choice, make

```
40
41 #d. Perform any 3 hypothesis tests using columns of your choice, make conclusions
42
43 chisq.test(table(train$V1, train$V2))
44
45 t.test(train$V4, train$V280, paired=TRUE)
46
47
```

45:42 (Top Level) R Script

Console Terminal

```
> chisq.test(table(train$V1, train$V2))

Pearson's Chi-squared test

data:  table(train$V1, train$V2)
X-squared = 22143000, df = 207360, p-value < 2.2e-16

warning message:
In chisq.test(table(train$V1, train$V2)) :
  chi-squared approximation may be incorrect
>
> t.test(train$V4, train$V280, paired=TRUE)

Paired t-test

data:  train$V4 and train$V280
t = 175.51, df = 52396, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 335.2969 342.8703
sample estimates:
mean of the differences
      339.0836

> |
```

P-Value is insignificant on both t.test and chi-square and less than 0.05 so we fail to reject null hypothesis

e. Create a linear regression model to predict the number of comments in the next 24 hours (relative to base time)

Linear Model Using LASSO

I used Lasso and got a success rate of 0.6345 as seen below

Steps Taken below:

#e. Create a linear regression model to predict the number of comments in the next 24 hours (relative to base time)

```
train = fread("bf-Train.csv")
```

```
test = fread("bf-Test.csv")
```

```
mse = function(y_hat, y) {
```

```
  mse = mean((y - y_hat)^2)
```

```
  return(mse)
```

```
}
```

```
# create design matrices
```

```
train_x = model.Matrix(V281 ~ . - 1, data = train, sparse = F)
```

```
train_x_sparse = model.Matrix(V281 ~ . - 1, data = train, sparse = T)
```

```
train_y = train$V281
```

```
test_x = model.Matrix(V281 ~ . - 1, data = test, sparse = F)
```

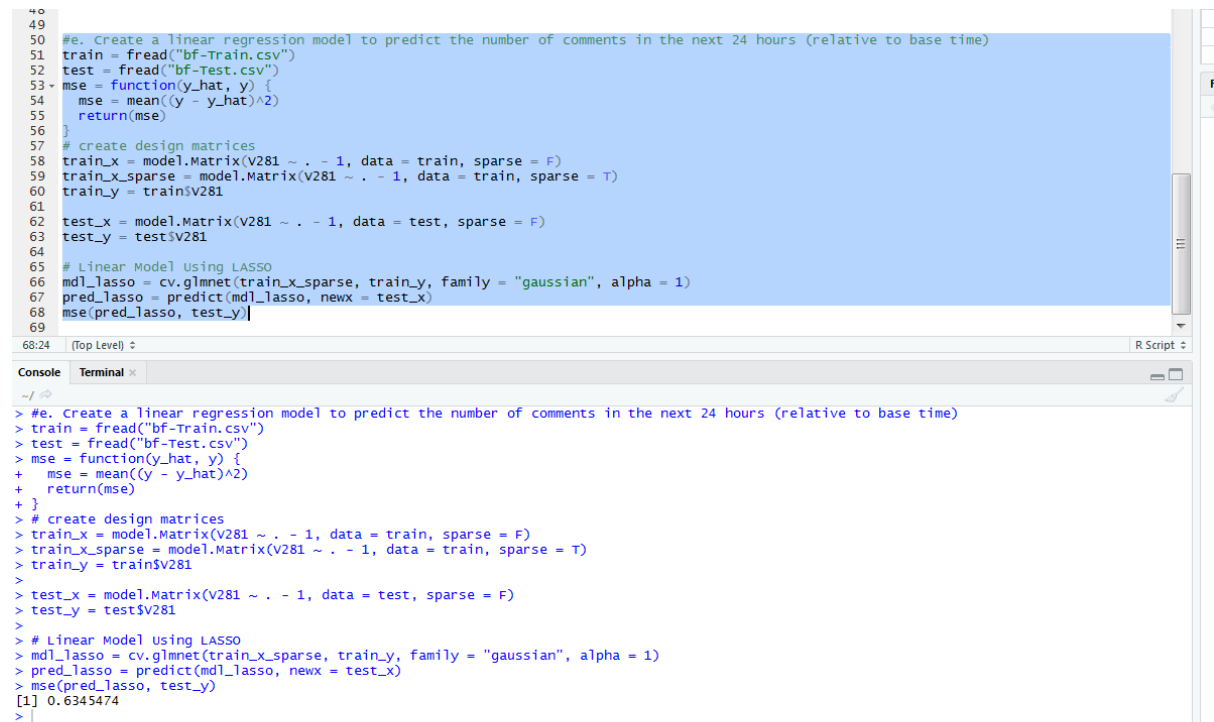
```
test_y = test$V281
```

```
# Linear Model Using LASSO
```

```
mdl_lasso = cv.glmnet(train_x_sparse, train_y, family = "gaussian", alpha = 1)
```

```
pred_lasso = predict(mdl_lasso, newx = test_x)
```

```
mse(pred_lasso, test_y)
```



```
40
49
50 #e. Create a linear regression model to predict the number of comments in the next 24 hours (relative to base time)
51 train = fread("bf-Train.csv")
52 test = fread("bf-Test.csv")
53 mse = function(y_hat, y) {
54   mse = mean((y - y_hat)^2)
55   return(mse)
56 }
57 # create design matrices
58 train_x = model.matrix(V281 ~ . - 1, data = train, sparse = F)
59 train_x_sparse = model.matrix(V281 ~ . - 1, data = train, sparse = T)
60 train_y = train$V281
61
62 test_x = model.matrix(V281 ~ . - 1, data = test, sparse = F)
63 test_y = test$V281
64
65 # Linear Model using LASSO
66 mdl_lasso = cv.glmnet(train_x_sparse, train_y, family = "gaussian", alpha = 1)
67 pred_lasso = predict(mdl_lasso, newx = test_x)
68 mse(pred_lasso, test_y)
69
```

68:24 (Top Level) R Script

Console Terminal

```
> #e. Create a linear regression model to predict the number of comments in the next 24 hours (relative to base time)
> train = fread("bf-Train.csv")
> test = fread("bf-Test.csv")
> mse = function(y_hat, y) {
+   mse = mean((y - y_hat)^2)
+   return(mse)
+ }
> # create design matrices
> train_x = model.matrix(V281 ~ . - 1, data = train, sparse = F)
> train_x_sparse = model.matrix(V281 ~ . - 1, data = train, sparse = T)
> train_y = train$V281
>
> test_x = model.matrix(V281 ~ . - 1, data = test, sparse = F)
> test_y = test$V281
>
> # Linear Model using LASSO
> mdl_lasso = cv.glmnet(train_x_sparse, train_y, family = "gaussian", alpha = 1)
> pred_lasso = predict(mdl_lasso, newx = test_x)
> mse(pred_lasso, test_y)
[1] 0.6345474
>
```