

tests the new solution(s), MM should produce valuable and engaging content for the e-commerce website that attracts and converts the target groups. Their hands-on experience with web analytics tools is much desired to support the new algorithm development and testing.

6. Model selection

```
newdat <- as.data.frame(dat)
# str(newdat)
newdat <- newdat %>% mutate(id = as.factor(id))
# glimpse(newdat)

dat_rfm <- newdat %>%
  group_by(id) %>%
  summarise(recency = as.numeric(r), frequency = as.numeric(f), monitery = as.numeric(m))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
# summary(dat_rfm)

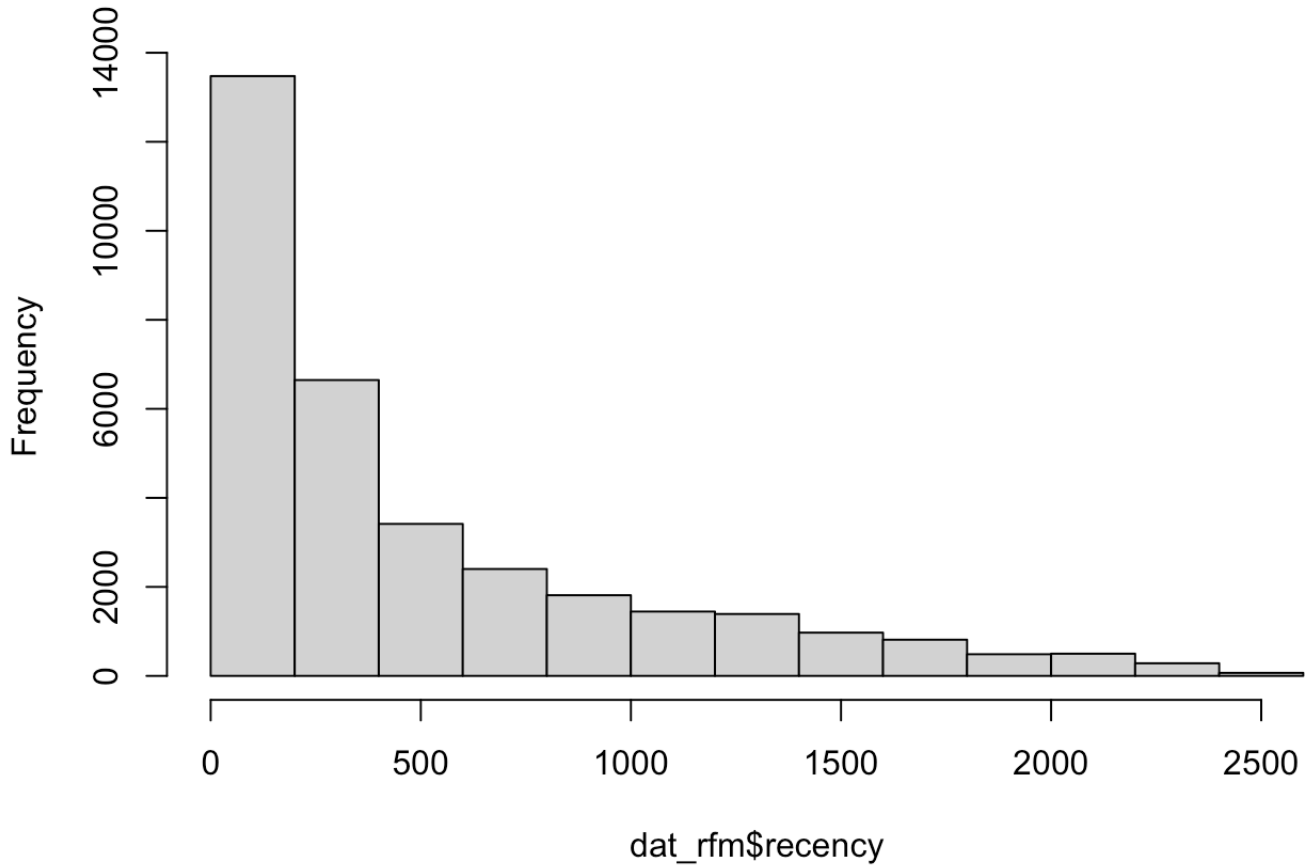
kable(head(dat_rfm,10))
```

id	recency	frequency	monitery
914	194	7	318.89478
957	3	14	368.05225
1406	1489	15	423.29834
1414	155	4	71.21704
1546	194	6	442.63818
1651	1797	2	47.93359
2046	383	3	52.39966
2062	19	16	1078.60742
2232	142	40	1453.84473
2534	80	7	119.28204

Below we perform the RFM analysis:

```
# ===== Clustering =====  
hist(dat_rfm$recency)
```

Histogram of dat_rfm\$recency



```
# Clustering  
dat_rfm2 <- dat_rfm  
row.names(dat_rfm2) <- dat_rfm2$id
```

```
## Warning: Setting row names on a tibble is deprecated.
```

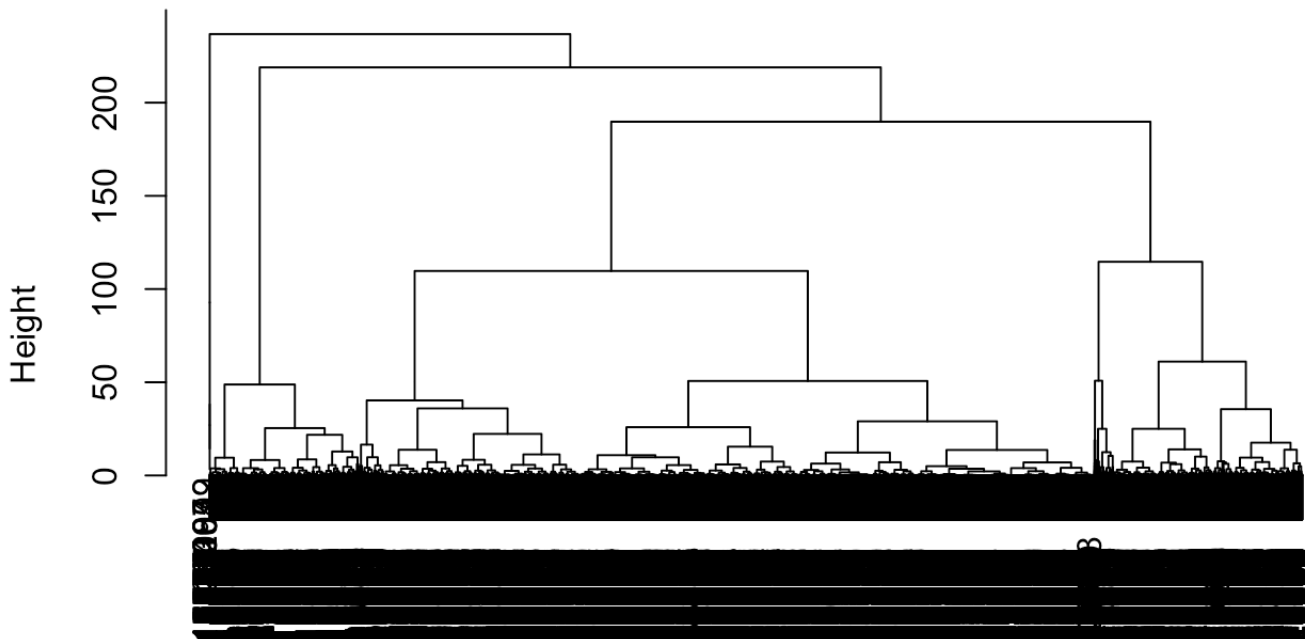
```
dat_rfm2$id <- NULL  
# Centralizing the data frame  
dat_rfm2 <- scale(dat_rfm2)  
summary(dat_rfm2)
```

```
##      recency      frequency      monitery
## Min.    :-0.9373   Min.    :-0.8306   Min.    : -0.06410
## 1st Qu.: -0.7216   1st Qu.: -0.6879   1st Qu.: -0.05395
## Median : -0.3971   Median : -0.4025   Median : -0.03799
## Mean    :  0.0000   Mean    :  0.0000   Mean    :  0.00000
## 3rd Qu.:  0.3753   3rd Qu.:  0.1683   3rd Qu.: -0.00118
## Max.    :  3.5978   Max.    :16.0072   Max.    :135.53425
```

```
d <- dist(dat_rfm2)
c <- hclust(d, method = 'ward.D2')

plot(c)
```

Cluster Dendrogram



d
hclust (*, "ward.D2")

```
# There are roughly 6 clusters
members <- cutree(c,k = 6)
table(members)
```

```
## members
##      1      2      3      4      5      6
## 15905  5823  4612   591  6778    4
```

```
# Segment the data set into clusters
seg <- aggregate(dat_rfm[,2:4], by=list(members), FUN = mean)
seg
```

```
##   Group.1  recency frequency  monitery
## 1      1  182.1896  3.239359   112.6560
## 2      2  221.5277 14.315988   521.8550
## 3      3 1629.4506  2.578708   102.2896
## 4      4  137.0880 38.759729  1446.6133
## 5      5  790.2785  3.907200   150.3676
## 6      6   93.2500 15.750000 329167.3750
```

```
# Assign scores to each cluster
bench <- as.data.frame(lapply(seg, mean, na.rm = TRUE))
order(seg$recency)
```

```
## [1] 6 4 1 2 5 3
```

```
copy <- seg

copy$recency[order(copy$recency)] <- c("6", "5", "4", "3", "2", "1")
copy$frequency[order(copy$frequency, decreasing = TRUE)] <- c("6", "5", "4", "3", "2", "1")
copy$monitery[order(copy$monitery, decreasing = TRUE)] <- c("6", "5", "4", "3", "2", "1")
names(copy)[2:4] <- c("recency_sort", "frequency_sort", "monetery_sort")
copy
```

```
##   Group.1 recency_sort frequency_sort monetery_sort
## 1      1      4      2      2
## 2      2      3      4      4
## 3      3      1      1      1
## 4      4      5      6      5
## 5      5      2      3      3
## 6      6      6      5      6
```

```
# ===== Form a function to segment the customer into the right cluster =====
newseg <- cbind(seg,copy[2:4])
newseg
```

```
##      Group.1    recency frequency    monitery recency_sort frequency_sort
## 1          1  182.1896  3.239359    112.6560             4             2
## 2          2  221.5277 14.315988    521.8550             3             4
## 3          3 1629.4506  2.578708    102.2896             1             1
## 4          4  137.0880 38.759729   1446.6133             5             6
## 5          5  790.2785  3.907200    150.3676             2             3
## 6          6   93.2500 15.750000 329167.3750             6             5
##      monetery_sort
## 1                  2
## 2                  4
## 3                  1
## 4                  5
## 5                  3
## 6                  6
```

```
# Recency
rrangel <- median(c(newseg$recency[newseg$recency_sort == '1'],newseg$recency[newseg$
recency_sort == '2']))
rrange2 <- median(c(newseg$recency[newseg$recency_sort == '2'],newseg$recency[newseg$
recency_sort == '3']))
rrange3 <- median(c(newseg$recency[newseg$recency_sort == '3'],newseg$recency[newseg$
recency_sort == '4']))
rrange4 <- median(c(newseg$recency[newseg$recency_sort == '4'],newseg$recency[newseg$
recency_sort == '5']))
rrange5 <- median(c(newseg$recency[newseg$recency_sort == '5'],newseg$recency[newseg$
recency_sort == '6']))

# The below function will return the group number based on the recency number
new_recency <- function(quantity){
  if (quantity[1] > rrangel) {
    print('1')
  } else if (quantity[1] <= rrangel & quantity[1] > rrange2) {
    print('2')
  } else if (quantity[1] <= rrange2 & quantity[1] > rrange3) {
    print('3')
  } else if (quantity[1] <= rrange3 & quantity[1] > rrange4) {
    print('4')
  } else if (quantity[1] <= rrange4 & quantity[1] > rrange5) {
    print('5')
  } else {
    print('6')
  }
}
```