07. Email

Jing Xie

11/20/2021

1. What percentage of users opened the email and what percentage clicked on the link within the email?
2. The VP of marketing thinks that it is stupid to send emails to a random subset and in a random way. Based on all the information you have about the emails that were sent, can you build a model to optimize in future email campaigns to maximize the probability of users clicking on the link inside the email?
3. By how much do you think your model would improve click through rate (defined as # of users who click on the link / total users who received the email). How would you test that?
4. Did you find any interesting pattern on how the email campaign performed for different segments of users? Explain. ## Input libraries needed

Input data

```
email_data = read.csv('email_table.csv')
email_open = read.csv('email_opened_table.csv')
link_click = read.csv('link_clicked_table.csv')
#are there duplicates?
nrow(email_data) == length(unique(email_data$email_id))
```

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

```
nrow(email_open) == length(unique(email_open$email_id))
```

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

```
nrow(link_click) == length(unique(link_click$email_id))
```

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

```
# No duplicates
# Are there any missing values?
sum(is.na(email_data))
```

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

```
sum(is.na(email_open))
```

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

```
sum(is.na(link_click))
```

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

```
# No missing values
```

Add two new columns email_data to indicate email_open and link_click.

Work out the percentage of users opened the email and clicked on the link within the email.

```
email_data$email_open = ifelse(email_data$email_id %in% email_open$email_id,
                               1, 0 )
email_data$link_click = ifelse(email_data$email_id %in% link_click$email_id,
                               1, 0 )

link_click %>% filter((link_click$email_id %in% email_open$email_id) ==FALSE )
```

```
##    email_id
## 1      9912
## 2    858449
## 3    505169
## 4    846127
## 5    763513
## 6    503034
## 7    841517
## 8    327248
## 9    645444
## 10   647421
## 11   931469
## 12      257
## 13   507413
## 14   958485
## 15   952396
## 16   359955
## 17   123727
## 18   570586
## 19   502713
## 20   428375
## 21   921085
## 22    26429
## 23   181168
## 24   104883
## 25   392441
## 26   297589
## 27   167345
## 28   772717
## 29   403381
## 30    64962
## 31    31052
## 32   439416
## 33   426464
## 34   505055
## 35   801271
## 36   954218
## 37   446716
## 38   115028
## 39    81601
## 40   547593
## 41   505481
## 42   611019
## 43   665829
```

```
## 44    25129
## 45   435495
## 46   873162
## 47   435454
## 48   206772
## 49   742967
## 50   916564
```

```
# There are 50 link clicks that don't open email open firstly, which is weird.

sum(email_data$email_open)/length(email_data$email_id)
```

```
## [1] 0.10345
```

```
sum(email_data$link_click)/sum(email_data$email_open)
```

```
## [1] 0.2048333
```

```
sum(email_data$link_click)/length(email_data$email_id)
```

```
## [1] 0.02119
```

10.345% of all the emails sent will be opened to read 20.48% of all opened emails, the link will be clicked to direct to the website. 2.119% of all the emails sent, the link will be clicked.

Build a model to find out the probabilty of click the email based on user characteristic

Have a look at the data file

```
summary(email_data)
```

```
##     email_id       email_text        email_version           hour
##  Min.   :     8   Length:100000      Length:100000      Min.   : 1.000
##  1st Qu.:246708   Class :character   Class :character   1st Qu.: 6.000
##  Median :498447   Mode  :character   Mode  :character   Median : 9.000
##  Mean   :498690                                         Mean   : 9.059
##  3rd Qu.:749943                                         3rd Qu.:12.000
##  Max.   :999998                                         Max.   :24.000
##    weekday           user_country       user_past_purchases   email_open
##  Length:100000      Length:100000      Min.   : 0.000      Min.   :0.0000
##  Class :character   Class :character   1st Qu.: 1.000      1st Qu.:0.0000
##  Mode  :character   Mode  :character   Median : 3.000      Median :0.0000
##                                        Mean   : 3.878      Mean   :0.1035
##                                        3rd Qu.: 6.000      3rd Qu.:0.0000
##                                        Max.   :22.000      Max.   :1.0000
##    link_click
##  Min.   :0.00000
##  1st Qu.:0.00000
##  Median :0.00000
##  Mean   :0.02119
##  3rd Qu.:0.00000
##  Max.   :1.00000
```

```
email_data$email_text = as.factor(email_data$email_text)
email_data$email_version= as.factor(email_data$email_version)
email_data$weekday= as.factor(email_data$weekday)
```

```
email_data$user_country= as.factor(email_data$user_country)
email_data$email_open= as.factor(email_data$email_open)
email_data$link_click= as.factor(email_data$link_click)
```

Split train and test dataset and build a model

```
train_sample = sample(nrow(email_data), size = nrow(email_data)*0.7)
train_data = email_data[train_sample,]
test_data = email_data[-train_sample,]

# Deal with imbalance data
table(email_data$link_click)
```

```
##
##     0     1
## 97881  2119
```

```
prop.table(table(email_data$link_click))
```

```
##
##       0       1
## 0.97881 0.02119
```

```
# The original data is highly imbalanced.

bal_train_data <- ROSE(link_click ~ ., data=train_data,seed=5)$data
bal_train_data <- bal_train_data[,-c(1,8)]
table(bal_train_data$link_click)
```

```
##
##     0     1
## 35328 34672
```

```
prop.table(table(bal_train_data$link_click))
```

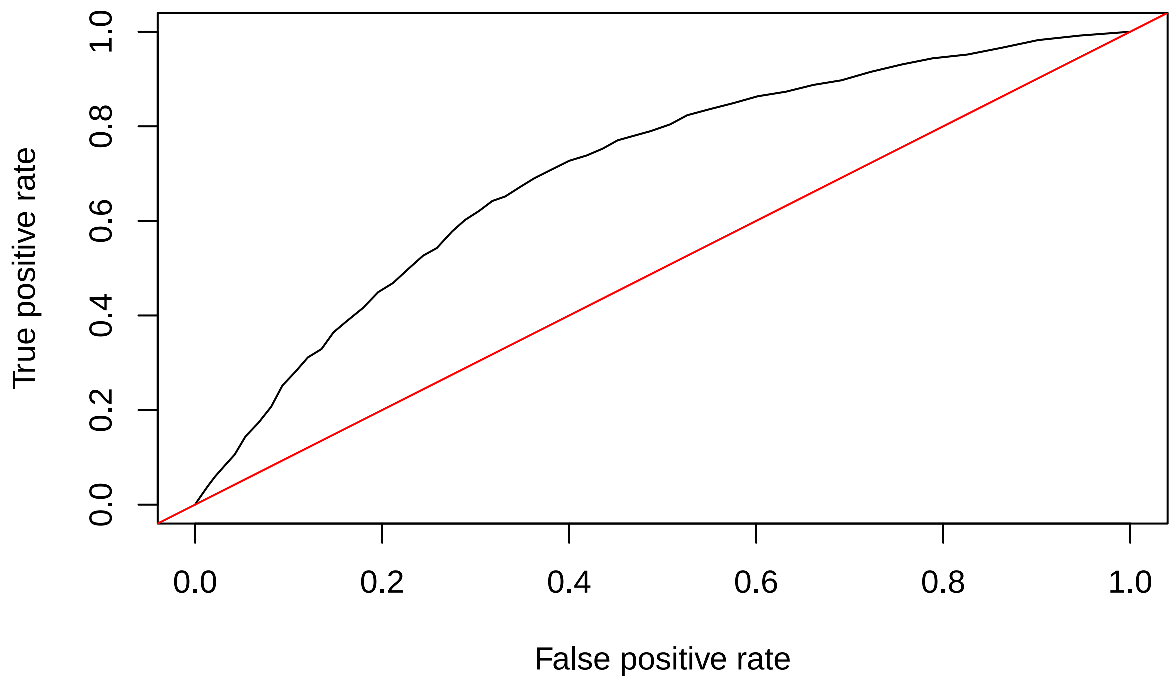
```
##
##         0         1
## 0.5046857 0.4953143
```

```
rf = randomForest(y=bal_train_data$link_click,
                  x = bal_train_data[,-7],
                  ytest = test_data$link_click,
                  xtest = test_data[, c(2:7)],
                  ntree = 50, mtry = 3, keep.forest = TRUE)
rf
```

```
##
## Call:
##  randomForest(x = bal_train_data[, -7], y = bal_train_data$link_click,      xtest = test_data[, c(2:
##                Type of random forest: classification
##                      Number of trees: 50
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 30.05%
## Confusion matrix:
##       0     1 class.error
```

```
## 0 23746 11582   0.3278419
## 1  9453 25219   0.2726407
##                 Test set error rate: 32.29%
## Confusion matrix:
##       0    1 class.error
## 0 19910 9467   0.3222589
## 1   219  404   0.3515249
```

```
#this creates an object with all the information you can possibly need about how
# different cutoff values impact all possible metrics: true positive, true
# negative, false positive, false negative...
rf_results = data.frame (true_values = test_data$link_click,predictions = rf$test$votes[,2])
pred = prediction(rf_results$predictions, rf_results$true_values)
#now let's just plot the ROC and look at true positive vs false positive
perf = performance (pred, measure = 'tpr', x.measure = "fpr")
plot(perf) + abline(a=0, b=1, col = 'red') # the red line is randomness
```




```
## integer(0)
```

```
auc_ROCR <- performance(pred, measure = "auc")
print(auc_ROCR@y.values[[1]])
```

```
## [1] 0.7063589
```

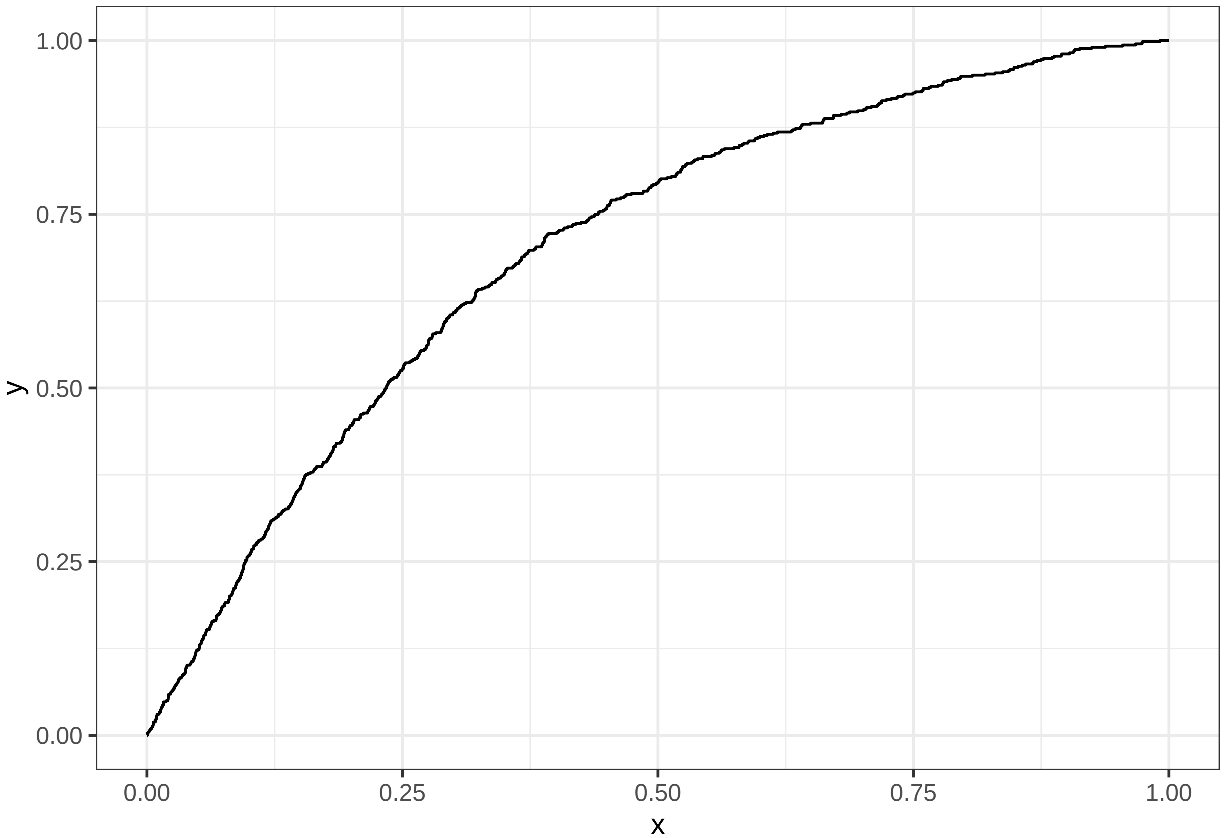
There is only 0.557771 AUC, very bad performance. After balancing data, AUC becomes 0.7073413.

By how much do you think your model would improve click through rate (defined as # of users who click on the link / total users who received the email). How would you test that?

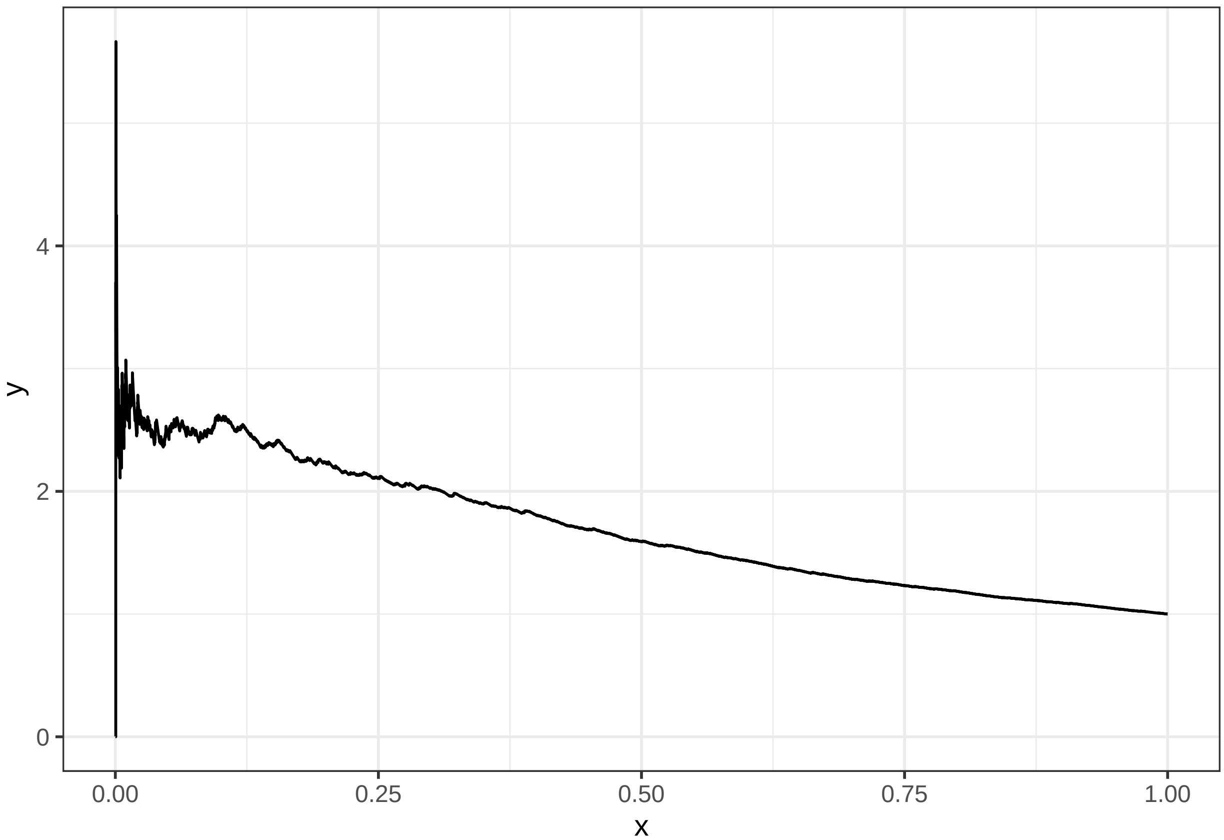
```
old_ctr = sum(email_data$link_click==1)/length(email_data$email_id)
old_ctr
```

```
## [1] 0.02119
```

```r
# Use LIFT to measure!!
# Make Cumulative Response Curve - Use Definition
test_cr = test_data %>%
  mutate(prob = rf_results$predictions) %>%
  arrange(desc(prob)) %>%
  mutate(click_yes = link_click) %>%
# the following two lines make the cumulative response curve
  mutate(y = cumsum(click_yes==1)/sum(click_yes==1),
         x = row_number()/nrow(test_data))
# Then, simply plot it.
ggplot(data = test_cr, aes(x = x, y = y)) + geom_line() + theme_bw()
```




```r
# Plot lift
test_lift = test_data %>%
  mutate(prob = rf_results$predictions) %>%
  arrange(desc(prob)) %>%
  mutate(click_yes = link_click) %>%
  # the following two lines make the lift curve
  mutate(x = row_number()/nrow(test_data),
         y = (cumsum(click_yes==1)/sum(click_yes==1))/x)
# Then, simply plot it.
ggplot(data = test_lift, aes(x = x, y = y)) + geom_line() +
  theme_bw()
```

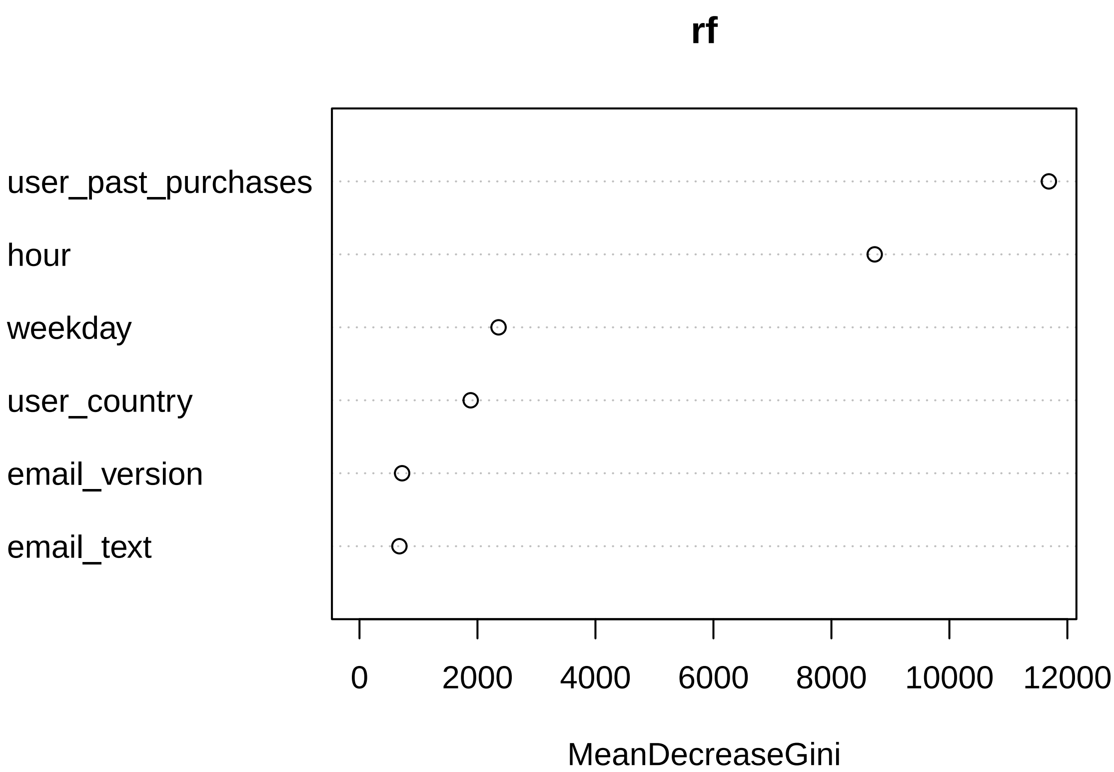


Old click through rate = 2% So comparing with randomly selected email pools, this model would improve click through rate by more than 2 times sending emails to top 25% users that has highest probability to click the link that predicted by this model.

More precisely, we can conduct a A/B Test to see whether the prediction model actually help increase the click through rate.

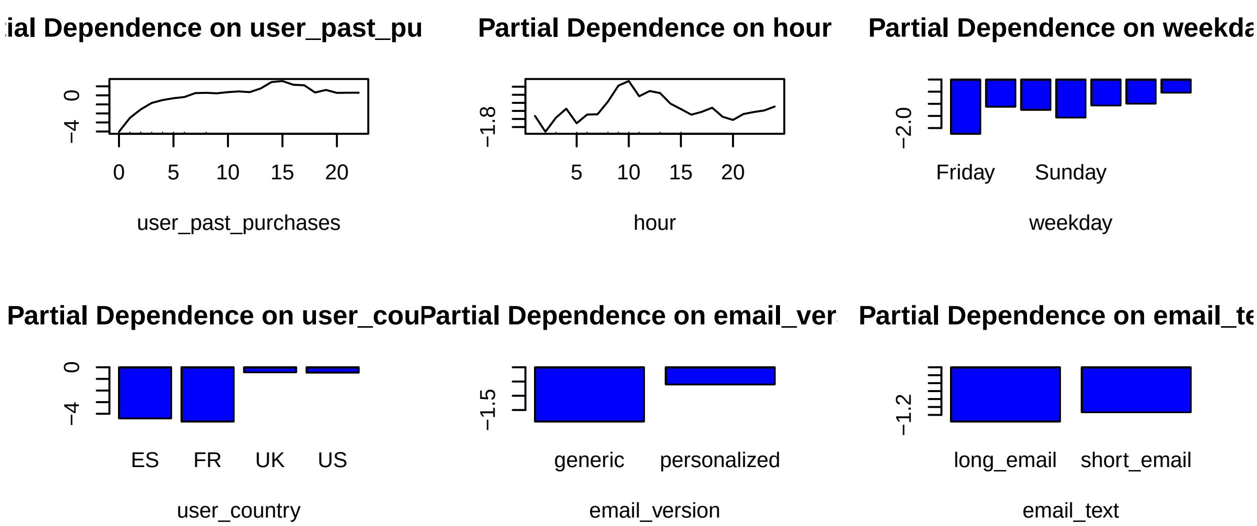
4. Did you find any interesting pattern on how the email campaign performed for different segments of users? Explain.

```
# Check variance importance:
varImpPlot(rf,type=2)
```



Let's check partial dependence plots:

```
op <- par(mfrow=c(3, 3)) # Put below 6 plots in a 3*3 grid.
partialPlot(rf, train_data, user_past_purchases, 1)
partialPlot(rf, train_data, hour, 1)
partialPlot(rf, train_data, weekday, 1)
partialPlot(rf, train_data, user_country, 1)
partialPlot(rf, train_data, email_version, 1)
partialPlot(rf, train_data, email_text, 1)
```

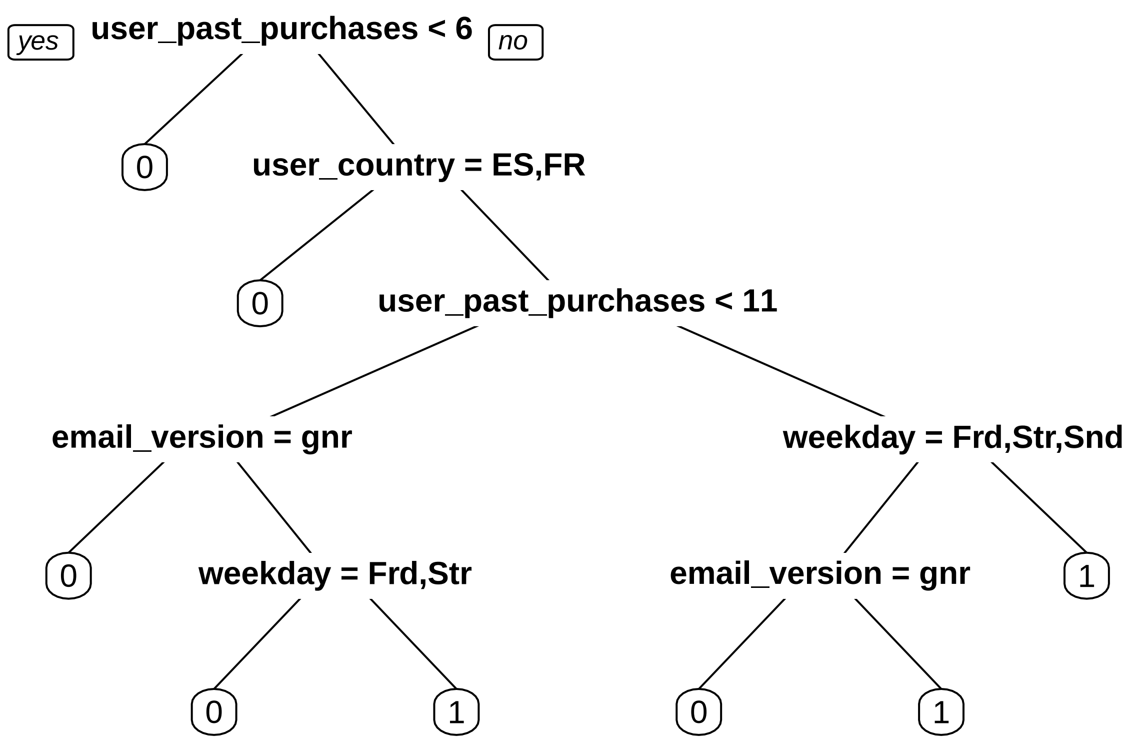


From the partial dependence plot, we can see that: 1. users that have more purchases in the past, is more likely to click the link, probably indicating that we need to send emails focusing on the old, loyal customers. 2. 10 am peaks on the CTR, we might change our sending email time to 10AM. 3. Email sent at weekday(middle of the week) has higher CTR compared with weekends! 4. UK and US have significantly higher CTR compared with other countries, so we can put more priority to these two countries. 5. Personalized and short email is more attractive to customers to click.

```
tree = rpart(train_data$link_click ~ ., train_data[,c(2:7)],
             control = list(maxdepth = 5,
                            cp = 0.002), # Complexity parameter!!
             parms = list(prior = c(0.7, 0.3)))
tree
```

```
## n= 70000
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 70000 21000.0000 0 (0.7000000 0.3000000)
##    2) user_past_purchases< 5.5 50799  9756.0160 0 (0.7860276 0.2139724) *
##    3) user_past_purchases>=5.5 19201 11243.9800 0 (0.5392803 0.4607197)
##      6) user_country=ES,FR 3854   912.4332 0 (0.7481313 0.2518687) *
##      7) user_country=UK,US 15347 10331.5500 0 (0.5028751 0.4971249)
##       14) user_past_purchases< 10.5 13199  7790.7750 0 (0.5372236 0.4627764)
##         28) email_version=generic 6602  3018.0480 0 (0.6021862 0.3978138) *
##         29) email_version=personalized 6597  4475.5490 1 (0.4839333 0.5160667)
##           58) weekday=Friday,Saturday 1908  1010.6950 0 (0.5650981 0.4349019) *
##           59) weekday=Monday,Sunday,Thursday,Tuesday,Wednesday 4689  3162.2820 1 (0.4566925 0.543307
##       15) user_past_purchases>=10.5 2148  1406.9690 1 (0.3563982 0.6436018)
##         30) weekday=Friday,Saturday,Sunday 927   625.8759 1 (0.4616208 0.5383792)
##           60) email_version=generic 444   210.5615 0 (0.5930546 0.4069454) *
##           61) email_version=personalized 483   319.0179 1 (0.3805066 0.6194934) *
##         31) weekday=Monday,Thursday,Tuesday,Wednesday 1221   781.0931 1 (0.3013567 0.6986433) *
```

```
prp(tree, varlen = 0)
```



Same patten as we saw in random forest partial dependence plot.

4. Did you find any interesting pattern on how the email campaign performed for different segments of users? Explain.

So there can be segments like: US/UK and ES/FR: US/UK has much higher CTR through the email campaign Loyal users: Users purchases more than 9 items are very likely to click the link in the email campaign.