1_Conversion

CathyQian

April 25, 2018

Challenge Description

We have data about users who hit our site: whether they converted or not as well as some of their characteristics such as their country, the marketing channel, their age, whether they are repeat users and the number of pages visited during that session (as a proxy for site activity/time spent on site).

Your project is to: - Predict conversion rate - Come up with recommendations for the product team and the marketing team to improve conversion rate

Solution

Part I. Data Analysis

Below are libraries needed.

```
require(dplyr)
require(rpart)
require(ggplot2)
require(randomForest)
```

Read the dataset into R and we get the following table:

```
data = read.csv('1_conversion_data.csv')
head(data)

##   country age new_user source total_pages_visited converted
## 1      UK  25        1    Ads                   1         0
## 2      US  23        1    Seo                   5         0
## 3      US  28        1    Seo                   4         0
## 4   China  39        1    Seo                   5         0
## 5      US  30        1    Seo                   6         0
## 6      US  31        0    Seo                   1         0
```

Let's check the structure of the data.

```
str(data)

## 'data.frame':    316200 obs. of  6 variables:
##  $ country            : Factor w/ 4 levels "China","Germany",..: 3 4 4 1 4
4 1 4 3 4 ...
##  $ age                : int  25 23 28 39 30 31 27 23 29 25 ...
##  $ new_user           : int  1 1 1 1 1 0 1 0 0 0 ...
## $ source             : Factor w/ 3 levels "Ads","Direct",..: 1 3 3 3 3 3
3 1 2 1 ...
## $ total_pages_visited: int  1 5 4 5 6 1 4 4 4 2 ...
## $ converted          : int  0 0 0 0 0 0 0 0 0 0 ...
```

It has 316200 observations of 6 variables.‘age’, ‘new_user’, ‘total_page_visited’ and ‘converted’ are numerical.‘country’ and ‘source’ are factor with multiple levels.

Next, let’s identify and deal with wrong data.

```
summary(data)
```

```
##     country            age            new_user         source
##  China  : 76602   Min.   : 17.00   Min.   :0.0000   Ads   : 88740
##  Germany: 13056   1st Qu.: 24.00   1st Qu.:0.0000   Direct: 72420
##  UK     : 48450   Median : 30.00   Median :1.0000   Seo   :155040
##  US     :178092   Mean   : 30.57   Mean   :0.6855
##                   3rd Qu.: 36.00   3rd Qu.:1.0000
##                   Max.   :123.00   Max.   :1.0000
##  total_pages_visited   converted
##  Min.   : 1.000      Min.   :0.00000
##  1st Qu.: 2.000      1st Qu.:0.00000
##  Median : 4.000      Median :0.00000
##  Mean   : 4.873      Mean   :0.03226
##  3rd Qu.: 7.000      3rd Qu.:0.00000
##  Max.   :29.000      Max.   :1.00000
```

A few quick obervations from the above data: 1) The site is probably a US site given the largest number of users from US. 2) The user base is quite young, with a median of 30.00 and mean of 30.57. 3) Mean conversion rate is only 3.2%, which is industry standard and make sense. 4) The maximum age is 123! This is probably an outlier and worth investigating.

```
sort(unique(data$age), decreasing = TRUE)
```

```
##  [1] 123 111  79  77  73  72  70  69  68  67  66  65  64  63  62  61  60
## [18]  59  58  57  56  55  54  53  52  51  50  49  48  47  46  45  44  43
## [35]  42  41  40  39  38  37  36  35  34  33  32  31  30  29  28  27  26
## [52]  25  24  23  22  21  20  19  18  17
```

Both 123 and 111 sames unrealistic. Let’s see how many users we are talking about:

```
subset(data, age>79)
```

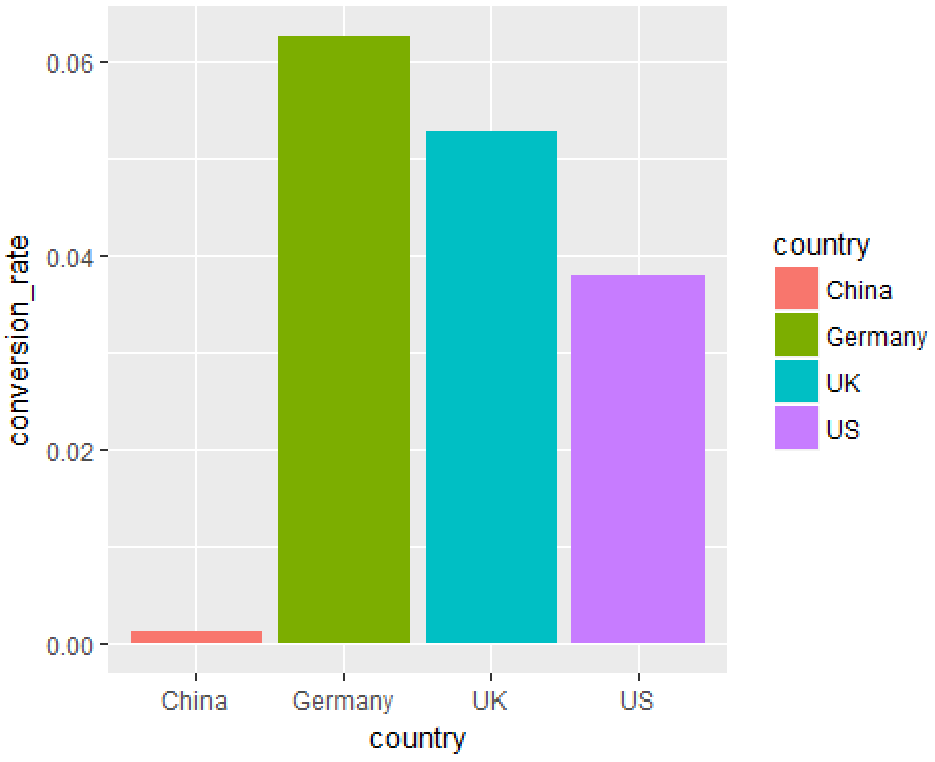
```
##        country age new_user source total_pages_visited converted
## 90929  Germany 123        0    Seo                  15         1
## 295582      UK 111        0    Ads                  10         1
```

It’s just two users. To be safe, let’s remove these two rows.

`r  data = subset(data, age < 80)`

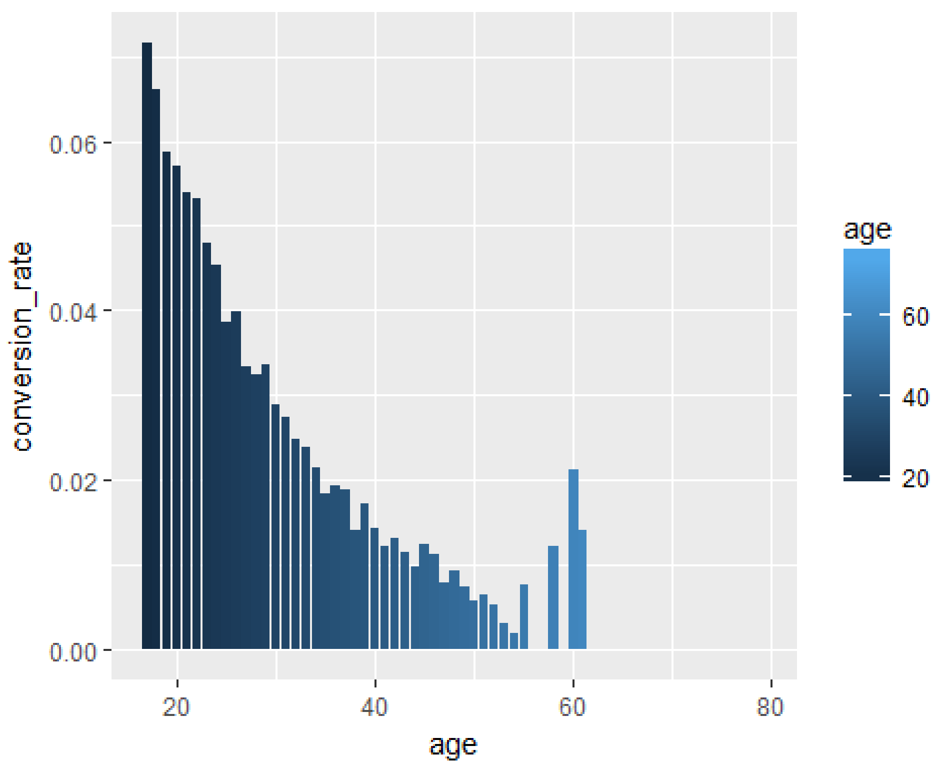
Next, let's get a sense of the data by investigating the variables and how their distribution differs for two-classes.

```
data_country = data %>%
               group_by(country) %>%
               summarise(conversion_rate = mean(converted))
ggplot(data = data_country, aes(x=country, y = conversion_rate)) +
geom_bar(stat = 'identity', aes(fill = country))
```



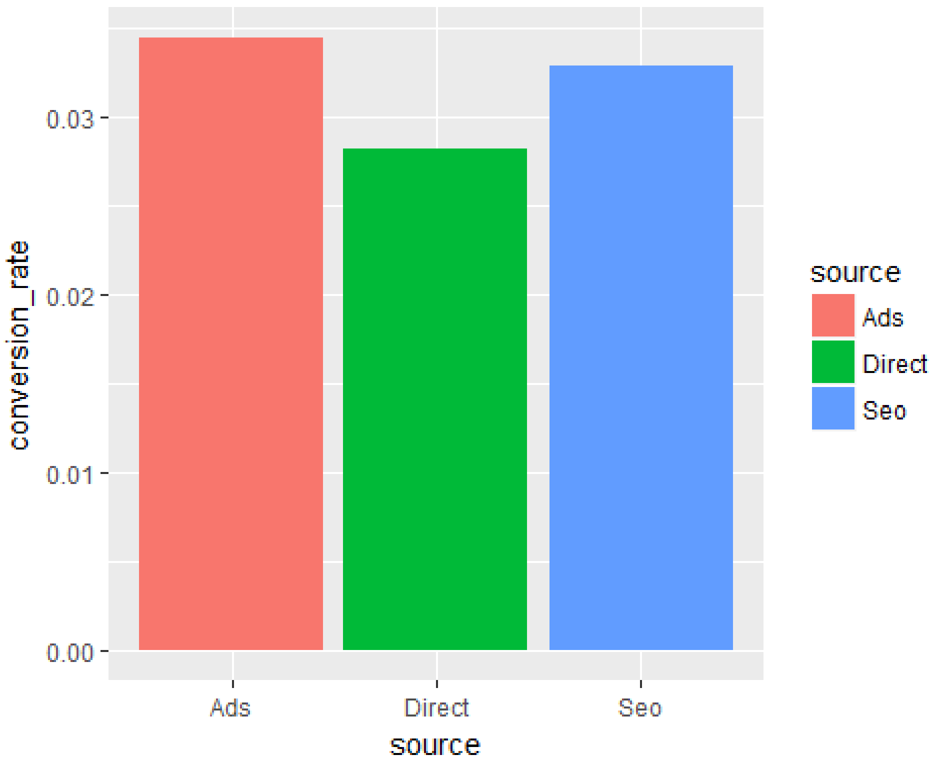
It's clear that China has a much lower conversion rate than other countries!

```
data_age = data %>%
           group_by(age) %>%
           summarise(conversion_rate = mean(converted))
ggplot(data = data_age, aes(x=age, y = conversion_rate)) + geom_bar(stat =
'identity', aes(fill = age))
```



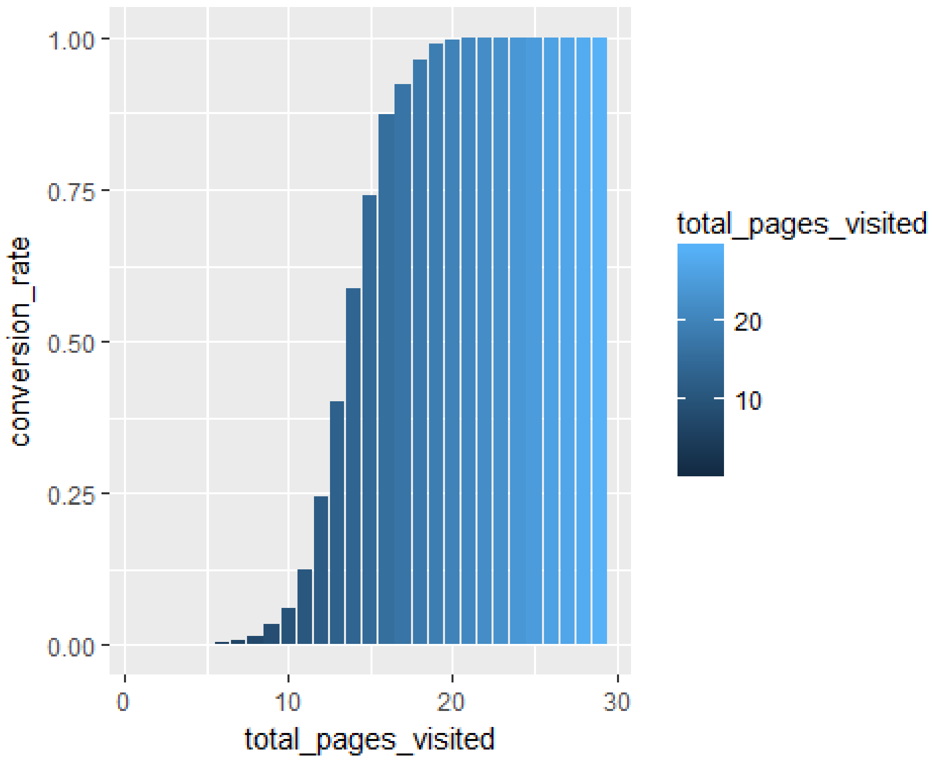
In general, the conversion rate decreases with increasing age, whereas it resumes a peak at age around 60.

```
data_source = data %>%
            group_by(source) %>%
            summarise(conversion_rate = mean(converted))
ggplot(data = data_source, aes(x = source, y = conversion_rate)) +
geom_bar(stat = 'identity', aes(fill = source))
```



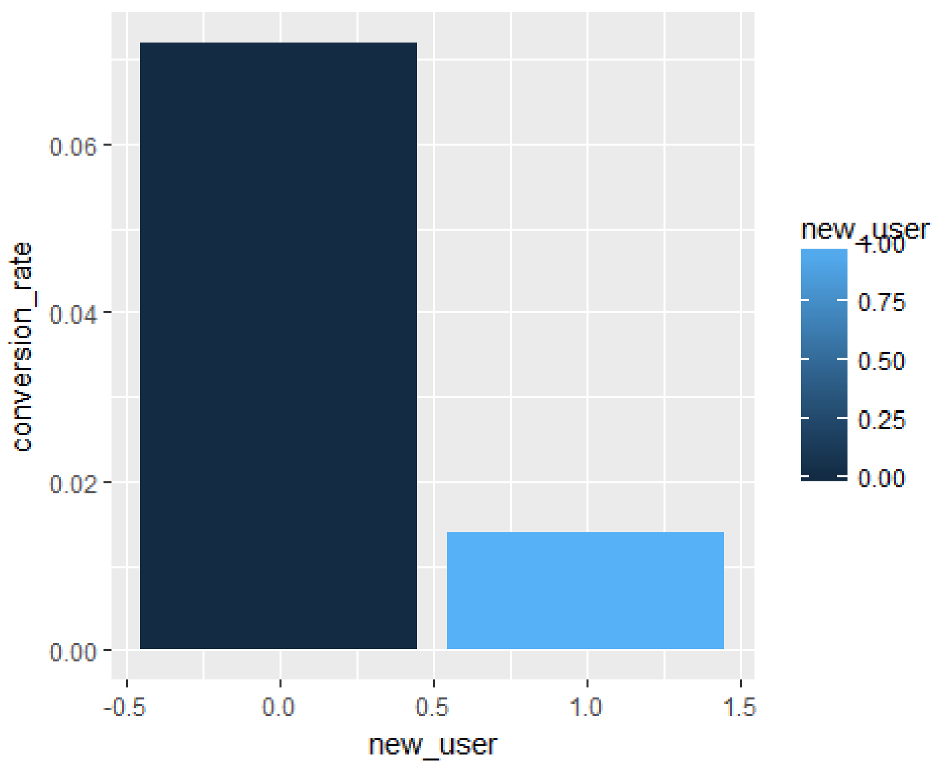
The conversion rate are comparable for users coming from the three different sources.

```
data_visited = data %>%
               group_by(total_pages_visited) %>%
               summarise(conversion_rate = mean(converted))
ggplot(data = data_visited, aes(x = total_pages_visited, y =
conversion_rate)) + geom_bar(stat = 'identity', aes(fill =
total_pages_visited))
```



It's intutive that the conversion rate increases with increasing total times the user visited the page and it gets close to 1 when the total times the user visited the page is over 20.

```
data_new_user = data %>%
                group_by(new_user) %>%
                summarise(conversion_rate = mean(converted))
ggplot(data = data_new_user, aes(x = new_user, y = conversion_rate)) +
geom_bar(stat = 'identity', aes(fill = new_user))
```



The conversion rate of old users is about 4 times of that of old users.

Part II. Machine Learning

Given that the output is binary, this is a classification problem. Let's pick random forests machine learning algorithm because 1) it usually requires very little time to optimize 2) it is strong with outliers, irrelevant variables, continuous and discrete variables.

Let's start with changing 'converted' and 'new_user' into factors which contain only a limited number of values.

```
data$converted = as.factor(data$converted)
data$new_user = as.factor(data$new_user)
```

Replace Germany with DE for brevity.

```
levels(data$country)[levels(data$country) == 'Germany'] = "DE"
```

Create test/training set with a standard 66% split and then build the forest with the standard values for the 3 most important parameters.

```
train_sample = sample(nrow(data), size = nrow(data)*0.66)
train_data = data[train_sample, ]
test_data = data[-train_sample, ]

rf = randomForest(y=train_data$converted, x = train_data[, -
ncol(train_data)], ytest = test_data$converted, xtest = test_data[, -
ncol(test_data)], ntree = 100, mtry = 3, keep.forest = TRUE)
```

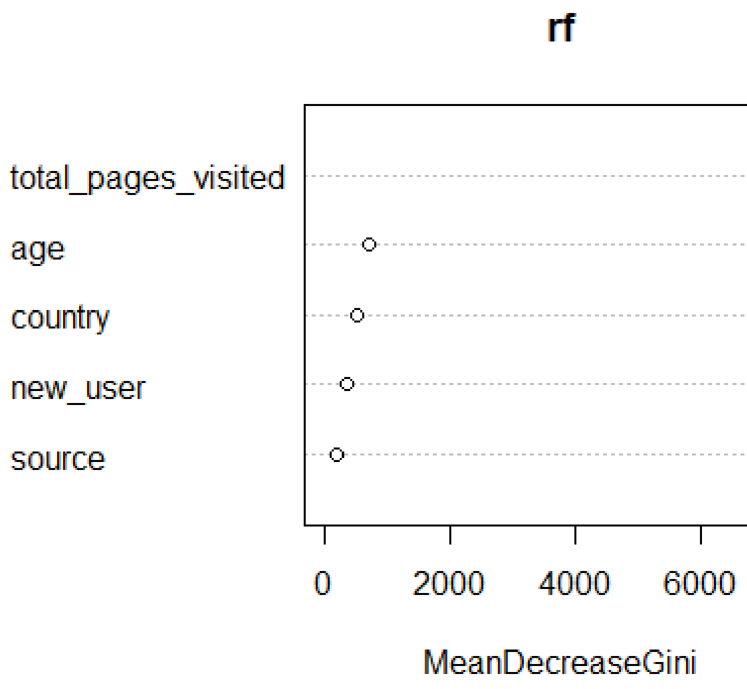
```
rf

##
## Call:
##  randomForest(x = train_data[, -ncol(train_data)], y =
train_data$converted,      xtest = test_data[, -ncol(test_data)], ytest =
test_data$converted,      ntree = 100, mtry = 3, keep.forest = TRUE)
##                Type of random forest: classification
##                      Number of trees: 100
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 1.47%
## Confusion matrix:
##        0    1 class.error
## 0 201155  825 0.004084563
## 1   2245 4465 0.334575261
##                 Test set error rate: 1.44%
## Confusion matrix:
##        0    1 class.error
## 0 103590  430  0.00413382
## 1   1114 2374  0.31938073
```

Both OOB error and test error are pretty similar: 1.5% and 1.4%. Thus we are confident that we are not overfitting.However, the initial aveage conversion are 3%, which means without doing nothing, we can easily get 97% prediction accuracy. So 1 - 1.5% = 98.5% is not shockingly good. Indeed, 30% of conversions are predicted as non-conversion (see class.error).

If we need to get the best possible accuracy or specifically minimizing false positve/false negative, we need to use ROCR and find the best cut-off point.We are statisfied with our current results for now.

Let’s check variable importance.

```
varImpPlot(rf, type = 2)
```



Total pages visited is the most important one. Unfortunately, this is proably the least “actionable” parameter. Because people visit many pages because they already want to buy. Also, in order to buy you have to click on multiple pages.

Let’s rebuild the RF without total_page_visited. Also, since classes are heavily unbalanced and we don’t have that powerful variable anymore, let’s change the weight a bit, just to make sure we will get something classified as 1.

```
rf = randomForest(y=train_data$converted, x = train_data[, -c(5,
ncol(train_data))], ytest = test_data$converted, xtest = test_data[, -c(5,
ncol(test_data))], ntree = 100, mtry = 3, keep.forest = TRUE, classwt =
c(0.7, 0.3))
rf

##
## Call:
##  randomForest(x = train_data[, -c(5, ncol(train_data))], y =
train_data$converted,      xtest = test_data[, -c(5, ncol(test_data))], ytest
= test_data$converted,      ntree = 100, mtry = 3, classwt = c(0.7, 0.3),
keep.forest = TRUE)
##                Type of random forest: classification
##                      Number of trees: 100
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 13.94%
## Confusion matrix:
##         0     1 class.error
## 0 175976 26004   0.1287454
## 1   3089  3621   0.4603577
##                  Test set error rate: 13.9%
## Confusion matrix:
##        0     1 class.error
## 0 90668 13352   0.1283599
## 1  1592  1896   0.4564220
```

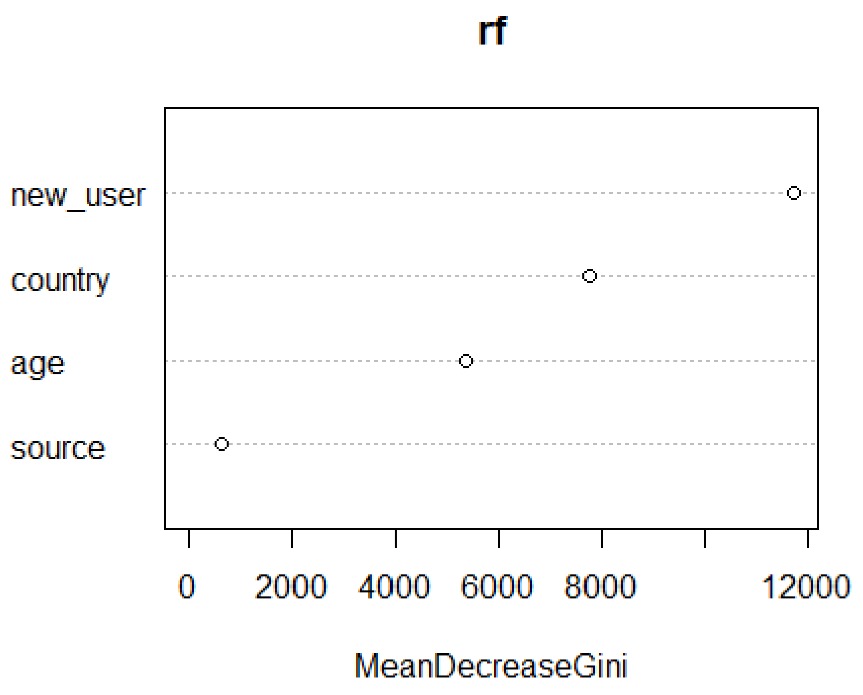
```r
rf
```

```
##
## Call:
##  randomForest(x = train_data[, -c(5, ncol(train_data))], y =
train_data$converted,      xtest = test_data[, -c(5, ncol(test_data))], ytest
= test_data$converted,      ntree = 100, mtry = 3, classwt = c(0.7, 0.3),
keep.forest = TRUE)
##                 Type of random forest: classification
##                       Number of trees: 100
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 13.94%
## Confusion matrix:
##        0     1 class.error
## 0 175976 26004   0.1287454
## 1   3089  3621   0.4603577
##                  Test set error rate: 13.9%
## Confusion matrix:
##        0     1 class.error
## 0 90668 13352   0.1283599
## 1  1592  1896   0.4564220
```

Accuracy went down, but the model is still good enough to give us some insights.

Let's recheck the variable importance:

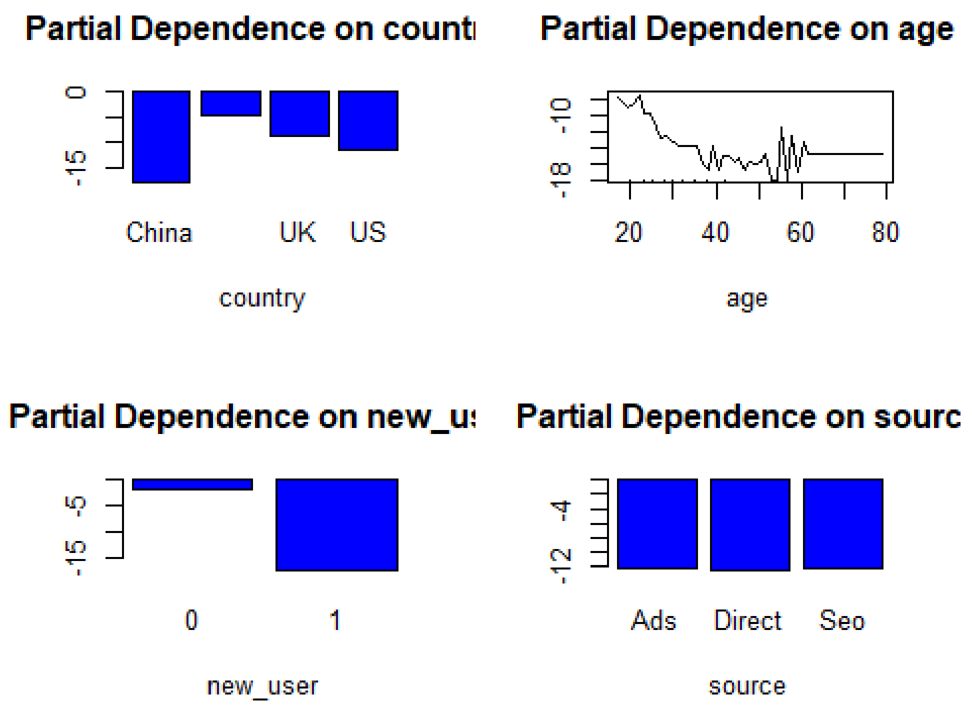
```r
varImpPlot(rf, type = 2) # 2=mean decrease in node impurity
```



Now new users is the most important one while source doesn't seem to matter at all.

Let's check partial dependence plots for the 4 vars.

```
op <- par(mfrow = c(2,2))
partialPlot(rf, train_data, country, 1)
partialPlot(rf, train_data, age, 1)
partialPlot(rf, train_data, new_user, 1)
partialPlot(rf, train_data, source, 1)
```



In partial dependence plots, we just care about the trend, not the actual y value. So this shows that: * Users with an old account are much better than new users * China is really bad, all other countries are similar with Germany being the best * The site works very well for young people and bad for less young people (>30 yrs old) * Source is irrelevant

Let’s now build a simple decision tree and check the 2 or 3 most important segments:

```
tree = rpart(data$converted ~., data[, -c(5, ncol(data))], control =
rpart.control(maxdepth = 3), parms = list(prior = c(0.7, 0.3)))
tree
```

```
## n= 316198
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 316198 94859.4000 0 (0.70000000 0.30000000)
##    2) new_user=1 216744 28268.0600 0 (0.84540048 0.15459952) *
##    3) new_user=0 99454 66591.3400 0 (0.50063101 0.49936899)
##      6) country=China 23094   613.9165 0 (0.96445336 0.03554664) *
##      7) country=DE,UK,US 76360 50102.8100 1 (0.43162227 0.56837773)
##       14) age>=29.5 38341 19589.5200 0 (0.57227507 0.42772493) *
##       15) age< 29.5 38019 23893.0000 1 (0.33996429 0.66003571) *
```

This model confirms previous findings from random forests: new_user, country and age are the three most important factors.

Conclusions and suggestions

1. The site is working very well for young users. Definitely let's tell marketing to advertise and use marketing channel which are more likely to reach young people.
2. The site is working very well for Germany in terms of conversion. But the summary showed that there are few Germans coming to the site which is much less than UK despite a larger population. Thus there is big opportunity in getting more German users.
3. Users with old accounts do much better. Targeted emails with offers to bring them back to the site could be a good idea to try.
4. Something is wrong with the Chinese version of the site. It is either poorly translated, doesn't fit the local culture, some payment issue or maybe it is just in English! Given how many users are based in China, fixing this should be a top priority.
5. Maybe go through the UI and figure out why older users perform so poorly? From 30 y/o conversion clearly starts dropping.
6. If I know someone has visited many pages, but hasn't converted, she almost surely has high purchase intent. I could email her targeted offers or sending her reminders. Overall, these are probably the easiest users to make convert.

As you can see, conclusions usually end up being about: 1. Tell marketing to get more of the good performing user segments. 2. Tell product to fix the experience for the bad performing ones.