Bookstore Rental Analysis

Wenyan Ge

June 12, 2019

This report is created for HarvardX: PH125.9xData Science: Capstone as the submit to "Chose Your Own Project". The report is made of 4 parts. * **Overview**, which describe the data, summarize the goal. * **Analyze the data**, including data exploration, data cleaning, and some models used during analyze. * **Results** * **Conclusion**

Overview

Image if you are an owner of a bookstore. Your business is to rent books to your customers. You find some books are very popular and out of stocks very soon, while some books rarely get rented. If you can predict how many of each book will be rented in the next month, you can maximize your business chance by increasing popular books stock and decreasing unpopular ones. Fortunately, you keep the data. Now let's see what we can do.

- **About the data set** Our data set is bookshop. It contains about 200,000 observations of each book checkouts(rented) in 2014 by month. We will explore the data set a bit more in **analyze the data**. The original source of the data can be download here.
- **About the goal** The goal of this report is to **predict future(1 month later) number of rent of each book**. We use RMSE(root mean squared error) to evaluate the models we built. We should try to minimize the RMSE of our models.
- **About the prediction models** We used 4 methods to build the prediction models. They are:

1. Just the average model
2. Just the nearest month model
3. glm
4. randomForest
5. Ensembles

Analyze the data

Meet the data

To begin with, let's download the bookshop data.

```
# load the library
# if you can't load successfully
# use install.packages("library_name") to install

library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
```

```
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
library(tidyr)
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------
```

```
## v tibble  2.1.1     v purrr   0.3.2
## v readr   1.3.1     v stringr 1.4.0
## v tibble  2.1.1     v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------------------------------------------------
## x lubridate::as.difftime() masks base::as.difftime()
## x lubridate::date()        masks base::date()
## x dplyr::filter()          masks stats::filter()
## x lubridate::intersect()   masks base::intersect()
## x dplyr::lag()             masks stats::lag()
## x lubridate::setdiff()     masks base::setdiff()
## x lubridate::union()       masks base::union()
```

```
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(Rborist)
```

```
## Rborist 0.1-17
```

```
## Type RboristNews() to see new features/changes/bug fixes.
```

```r
# download, unzip, and read the data.
dl <- tempfile()
download.file("https://github.com/wenyanfelix/bookshop/raw/master/bookshop.zip", dl)

bookshop <- unzip(dl)

bookshop <- read.csv(bookshop)

rm(dl)
```

The data we use is a subset of 20 million data set, which contains a 10 years records from 2005 to 2015. We only use the data in 2014 due to running time problems.

We explore the data by answer a few questions. 1. How many rows and columns are there in the data? We have 237217 rows and 13 columns.

```r
dim(bookshop)
```

```
## [1] 237217     13
```

2. What are the names of the columns?

The colum names are:

```r
names(bookshop)
```

```
##  [1] "X"                "id"               "usage_class"
##  [4] "checkout_type"    "material_type"    "checkout_year"
##  [7] "checkout_month"   "checkouts"        "title"
## [10] "creator"          "subjects"         "publisher"
## [13] "publication_year"
```

```r
# For simplicity, we will only use
# title, checkout_year, checkout_month, checkouts
# for prediction.

# "checkouts" means on s specific date,
# how many times are the book be rented
```

3. How many unique titles in the data?
    There are 31,087 unique titles in the data.

4. Which are the top 10 titles rented of the year?

Here are the top 10 titles:
I see one of my favorite TV series: A Game of Thrones: A Song of Ice and Fire Series, Book, have been r

```r
bookshop %>%
  group_by(title) %>%
  summarize(sum_checkouts = sum(checkouts)) %>%
  arrange(desc(sum_checkouts))
```

```
## # A tibble: 31,087 x 2
##    title                                                    sum_checkouts
##    <fct>                                                            <int>
##  1 Wild: From Lost to Found on the Pacific Crest Trail               1343
##  2 A Game of Thrones: A Song of Ice and Fire Series, Book 1          1288
##  3 Pure heroine [sound recording] / Lorde.                           1266
##  4 The lowland : a novel / Jhumpa Lahiri.                            1093
##  5 The interestings / Meg Wolitzer.                                  1082
##  6 1Q84                                                              1067
##  7 Wild : from lost to found on the Pacific Crest Trail / Ch~         996
##  8 The Paris Wife: A Novel                                            959
##  9 Sycamore Row: Jake Brigance Series, Book 2                         931
## 10 I broke my trunk! / by Mo Willems.                                 890
## # ... with 31,077 more rows
```

5. Which are the worst? Here are the worst:

```
bookshop %>%
  group_by(title) %>%
  summarize(sum_checkouts = sum(checkouts)) %>%
  filter(sum_checkouts == 1)
```

```
## # A tibble: 1,466 x 2
##    title                                                  sum_checkouts
##    <fct>                                                          <int>
##  1 .45-caliber manhunt [text (large print)] / Peter Brandvol~           1
##  2 ?? có gi?c ng? ngon / Lâm D? Ph??ng ; biên d?ch, Nguy?n K~           1
##  3 ?????Hero 's Troubles and Fortune?                                   1
##  4 ?leanor Rigbi / D?vid Kouplend.                                      1
##  5 ?rini r?l wihan masimello iyagi = Don't eat the marshmall~           1
##  6 ¡Olé flamenco! / George Ancona.                                      1
##  7 100 best plants for the coastal garden : the botanical ba~           1
##  8 101 things you wish you'd invented-- and some you wish no~           1
##  9 1327 - 1547, The Black Prince to Henry V: This Sceptred I~           1
## 10 1938: Hitler's Gamble                                                1
## # ... with 1,456 more rows
```

There are 6 titles have over 1,000 checkouts this year, while 1466 books only get 1 checkout.

6. Mean, Median, SD of the checkouts?

```
bookshop %>%
  group_by(title) %>%
  summarize(sum_checkouts = sum(checkouts)) %>%
  summarise(mean = mean(sum_checkouts),
   median = median(sum_checkouts),
   sd = sd(sum_checkouts))
```

```
## # A tibble: 1 x 3
##    mean median    sd
##   <dbl>  <int> <dbl>
## 1  26.2     12  49.3
```

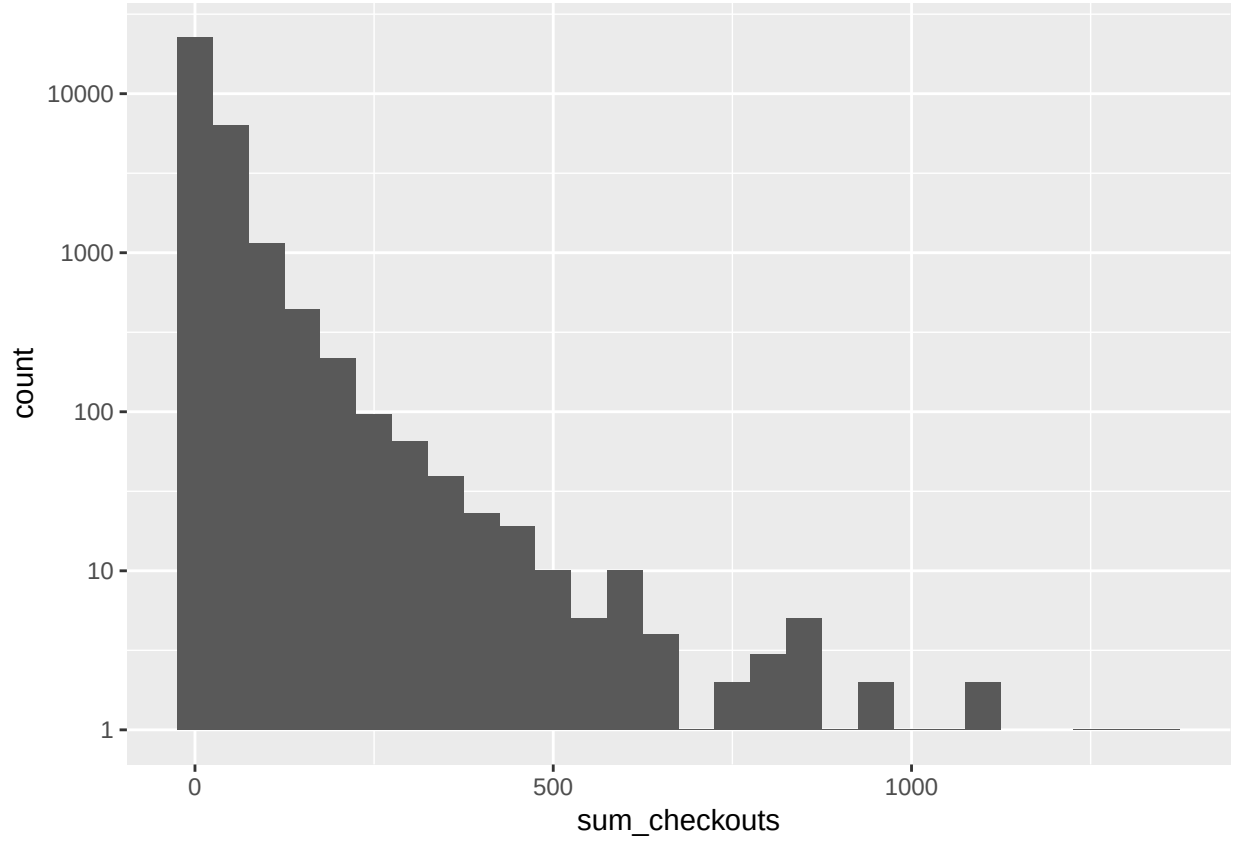
In this graph, we can see most of the titles (or 31,043 precisely, 99.86%) have less than 500 checkouts a year.

Only 44 titles (0.14%) have more or equal to 500 checkouts.

```
bookshop %>%
      group_by(title) %>%
      summarize(sum_checkouts = sum(checkouts)) %>%
      ggplot(aes(sum_checkouts)) +
      geom_histogram(binwidth = 50) +
      scale_y_log10()
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```




```
    bookshop %>%
      group_by(title) %>%
      summarize(sum_checkouts = sum(checkouts)) %>%
      filter(sum_checkouts >= 500) %>%
      arrange(desc(sum_checkouts))
```

```
## # A tibble: 44 x 2
##    title                                                    sum_checkouts
##    <fct>                                                           <int>
##  1 Wild: From Lost to Found on the Pacific Crest Trail              1343
##  2 A Game of Thrones: A Song of Ice and Fire Series, Book 1         1288
##  3 Pure heroine [sound recording] / Lorde.                          1266
##  4 The lowland : a novel / Jhumpa Lahiri.                           1093
##  5 The interestings / Meg Wolitzer.                                 1082
##  6 1Q84                                                             1067
```

```
##  7 Wild : from lost to found on the Pacific Crest Trail / Ch~           996
##  8 The Paris Wife: A Novel                                              959
##  9 Sycamore Row: Jake Brigance Series, Book 2                           931
## 10 I broke my trunk! / by Mo Willems.                                   890
## # ... with 34 more rows
```

We can also visualize the data by such as publication_year, subjects. But in fact the data of publication_year, subjects are quite irregular.

```
unique(bookshop$publication_year)
```

Here are some exmaples:

- 2013, c2009. #2 years in 1 value
- [2005?] #unsure value?
- 92009. #miss input
- Heisei 19 [2007] #Japanese year

Due to the complexity and ambiguity of cleaning data, we will leave it for this report.

clean the data

We are going to shape the data into tidy data before analyze. Here is the code:

```
# First, we mutate a column named "checkout_date"
# just combine the year and the month
# then we drop columns we don't use for our prediction
cleaned_data <- bookshop %>%
  mutate(checkout_date = ymd(paste(checkout_year, checkout_month, 1, sep= "-"))) %>%
  select(title, checkout_date, checkouts)


# Here we reshaping the data into a tidy data.
# Now you can see every title has just one row, which contains
# its checkout number from Jan. 2014 to Dec. 2014
cleaned_data <- reshape(cleaned_data, idvar = "title",
 timevar = "checkout_date", direction = "wide")
```

```
## Warning in reshapeWide(data, idvar = idvar, timevar = timevar, varying =
## varying, : multiple rows match for checkout_date=2014-01-01: first taken

## Warning in reshapeWide(data, idvar = idvar, timevar = timevar, varying =
## varying, : multiple rows match for checkout_date=2014-02-01: first taken

## Warning in reshapeWide(data, idvar = idvar, timevar = timevar, varying =
## varying, : multiple rows match for checkout_date=2014-03-01: first taken

## Warning in reshapeWide(data, idvar = idvar, timevar = timevar, varying =
## varying, : multiple rows match for checkout_date=2014-04-01: first taken

## Warning in reshapeWide(data, idvar = idvar, timevar = timevar, varying =
## varying, : multiple rows match for checkout_date=2014-05-01: first taken

## Warning in reshapeWide(data, idvar = idvar, timevar = timevar, varying =
## varying, : multiple rows match for checkout_date=2014-06-01: first taken

## Warning in reshapeWide(data, idvar = idvar, timevar = timevar, varying =
## varying, : multiple rows match for checkout_date=2014-07-01: first taken

## Warning in reshapeWide(data, idvar = idvar, timevar = timevar, varying =
## varying, : multiple rows match for checkout_date=2014-08-01: first taken
```

```
## Warning in reshapeWide(data, idvar = idvar, timevar = timevar, varying =
## varying, : multiple rows match for checkout_date=2014-09-01: first taken

## Warning in reshapeWide(data, idvar = idvar, timevar = timevar, varying =
## varying, : multiple rows match for checkout_date=2014-10-01: first taken

## Warning in reshapeWide(data, idvar = idvar, timevar = timevar, varying =
## varying, : multiple rows match for checkout_date=2014-11-01: first taken

## Warning in reshapeWide(data, idvar = idvar, timevar = timevar, varying =
## varying, : multiple rows match for checkout_date=2014-12-01: first taken
```

```r
# we can see some NA where the book get 0 rented in that month
# we replace NAs by 0 use this code
cleaned_data[is.na(cleaned_data)] <- 0


# change column names so its easier to read
# we will use x_0 ~ x_10 (checkouts number from Jan. to Nov.)
# as the features to predict y (checkouts number in Dec.)
colnames(cleaned_data) <- c("title", paste("x",0:10,sep = "_"), "y")


# create a train set and test set
# remember y is checkouts number in Dec. which we are trying to predict
y <- cleaned_data$y

# split the data into train(90%) and test data(10%)
set.seed(1)
test_index <- createDataPartition(y, times = 1, p = 0.1, list = FALSE)

train_set <- cleaned_data %>% slice(-test_index)
test_set <- cleaned_data %>% slice(test_index)
```

Now the data is tidy, we can move on to our analyze.

prediction models

Again, we use RMSE(root mean squared error) to evaluate the models we built.

```r
RMSE <- function(true_checkouts, predicted_checkouts){
  sqrt(mean((true_checkouts - predicted_checkouts)^2))
}
```

Let's start with 2 simple models. The "Just the average model" and "Just the nearest month model". ### Model 1: Just the average model
We are going to predict y just using the average of the data from Jan. to Nov.

```r
# just row mean model
# we don't use train data here, because we need to use mean of the
# test sets' mean (features) to predict y
just_row_mean <- test_set %>% select(-title,-y) %>% rowMeans()
rmse_just_mean <- RMSE(test_set$y, just_row_mean)
rmse_just_mean
```

```
## [1] 2.621224
```

Suprisingly, this naive model get a rmse of 2.6, which means our prediction is less than 3 checkouts from the true checkouts. Our conclusion here is each title gets a checkout fluctuates within 3 per month on average.

```
Note: I tried a large set of this data, which rmse_just_mean is about
 18.
 The result of rmse of 2.6 may be due to the data we've chosen.
 However, we will stick to the data we use here.
```

Model 2: Just the nearest month model

Since the checkouts usually don't change much in a month, instead of the average of Jan. to Nov., we try just use the Nov. data to predict the checkouts of Dec...

```
# just use November data to predict
rmse_just_nov <- RMSE(test_set$y, test_set$x_10)
rmse_just_nov
```

```
## [1] 1.992671
```

We can see the rmse is 1.9. This simple model performs pretty good and told us it's not always the case, the more the data, the better the prediction. Sometimes less data performs better, like this case.

Model 3: glm

We have already got a quite good result. However, let's see if we can still improve the result by glm.

```
# model 3: glm
# use all the months Jan. ~ Nov. to predict
train_glm_1 <- train(y ~ x_0 + x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_10, method = "gl
y_hat_glm_1 <- predict(train_glm_1, test_set)
rmse_1 <- RMSE(test_set$y, y_hat_glm_1)
rmse_1
```

```
## [1] 1.853124
```

```
# use 3 latest months Sep. ~ Nov. to predict
train_glm_2 <- train(y ~ x_8 + x_9 + x_10, method = "glm", data = train_set)
y_hat_glm_2 <- predict(train_glm_2, test_set)
rmse_2 <- RMSE(test_set$y, y_hat_glm_2)
rmse_2
```

```
## [1] 1.830864
```

We can see the glm models improve a bit with rmses are about 1.8. Again, less data (3 months) provide slightly better result.

Model 4: randomForest

After the glm model, we try random Forest model.

```
# codes used here may take several minutes to run
# model 4: randomForest
# use 3 latest months Sep. ~ Nov. to predict
train_rf_1 <- randomForest(y ~ x_8 + x_9 + x_10, data = train_set)
y_hat_rf_1 <- predict(train_rf_1, newdata = test_set)
rmse_rf_1 <- RMSE(test_set$y, y_hat_rf_1)
rmse_rf_1
```

```
## [1] 1.795625
```

```
# use all the months Jan. ~ Nov. to predict
train_rf_2 <- randomForest(y ~ x_0 + x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_10, data =
```

```
y_hat_rf_2 <- predict(train_rf_2, newdata = test_set)
rmse_rf_2 <- RMSE(test_set$y, y_hat_rf_2)
rmse_rf_2
```

```
## [1] 1.814296
```

Our rmse is about 1.8 slightly better than glm models. 3 latest months data beat all months data again!

Model 5: Ensembles

We try to ensembles a few models here and see if they do better.

```
# an Ensemble model of randomForests
rmse_ensemble_1 <- RMSE(test_set$y, (y_hat_rf_1 + y_hat_rf_2)/2)

# an Ensemble model of glm and randomForest
rmse_ensemble_2 <- RMSE(test_set$y, (y_hat_glm_2 + y_hat_rf_2)/2)

# an Ensemble model of "just November" + glm + randomForest
rmse_ensemble_3 <- RMSE(test_set$y, (test_set$x_10 + y_hat_glm_2 + y_hat_rf_1 )/3)
```

Our result is improved to about 1.7 better than any single model.

Results

Here are the models and results:

```
models <- c("Just the average",
            "Just the nearest month",
            "glm 1",
            "glm 2",
            "randomForest 1",
            "randomForest 2",
            "Ensembles 1",
            "Ensembles 2",
            "Ensembles 3")

RMSEs <- c(rmse_just_mean,
           rmse_just_nov,
           rmse_1,
           rmse_2,
           rmse_rf_1,
           rmse_rf_2,
           rmse_ensemble_1,
           rmse_ensemble_2,
           rmse_ensemble_3)

rmse_results <- tibble(model = models, RMSE = RMSEs)
rmse_results
```

```
## # A tibble: 9 x 2
##   model                   RMSE
##   <chr>                  <dbl>
## 1 Just the average        2.62
## 2 Just the nearest month  1.99
## 3 glm 1                   1.85
```

```
## 4 glm 2                  1.83
## 5 randomForest 1         1.80
## 6 randomForest 2         1.81
## 7 Ensembles 1            1.78
## 8 Ensembles 2            1.78
## 9 Ensembles 3            1.78
```

Conclusion

As a bookshop owner, we succeeded in predict the future number of each book within a rmse of about 1.7. Actually, the result is quite good, we can save money on purchase new books which do not seem to be rented many times in the future.

However, it's just a start. We can predict books rent a month later, but can we predict 3 months later and also get a good prediction? The subject of the title definitely has some influence on the checkout number, can we use this feature to improve our model? We will not discuss these questions in this report, but we shall keep in mind there are always more we can do with the data.

Some reminder to myself. It does little with the report. * This may be a little wired. Because we shall ask ourselves is data analyze really required? We should always think if we can solve the problem in other ways. Data is just the tool we use to solve problems, not the goal.

- **tradeoff** Time and resource(computer memory, CPU speed, your focus) are limited. We can not analyze every aspect of the data.

  In my case, I tried to download the original data, but just downloading costs some hours. Then an error of code get R to restart and all my data in R is gone. As a result, I select a subset which won't crash R and don't cost much time to run code on my pc. This is a tradeoff.

  You can see there are 13 features in the original data, but we only used 4. This is another a tradeoff. In the real world, we have unlimited feature can be used to machine learning, but you should be very careful to minimize your numbers of feature which can maximize your predict. It's not easy to do so, but we should try our best.

- **more is not always better** In the report, we see fewer data made a better prediction. It makes sense especially during prediction from time. Suppose you have 10 years sales data of a book, the book may be a best seller 10 years ago, nevertheless, how it sells the last year give you much fresh information you need to predict next year sales.

I would like to thank edX and professor Rafael A. Irizarry to offer me a great chance to learn from the Data Science Course. And also like to thank all the online students and TAs, who helped me understand the course better.

References

Rafael A. Irizarry https://rafalab.github.io/dsbook/ Origin data source https://www.kaggle.com/city-of-seattle/seattle-checkouts-by-title