MA678 Project: Sephora Website Analysis for Products

Hanzhang Song

2022-12-11

1. Abstract

This report uses data from Kaggle, containing products and their ratings, ingredients, and other information from Sephora website. With the data set, I want to explore whether there is a relationship between the number of loves on different categories of product and their actual ratings. I'm curious about if these two measurements of popularity are correlated, and which product gives the most accurate result when we use the number of loves to predict ratings. There are a lot of categories on Sephora website. In this project, I will randomly select 10 of them to run the model. I find that products have larger number of loves will also have higher ratings, but this finding is limited.

2. Introduction

Loves is a way to collect your desired items before adding them to the shopping cart. By clicking the heart icons, you can add the items to your Love list and whenever you visit the website again, you can directly go to your list, putting your final selections into shopping cart and checking out. The Love list gives your extra time to consider your choices, without losing your initial interests. The love list is based on customers' first impression on products based on their appearances and descriptions. From the data set, the most popular item got 1,300,000 loves. However, does this product also have the highest rating? Can we use the number of loves to predict ratings? Which category returns the closest estimation?

The following report will analyze products' information and identify relationships, allowing Sephora to think about how they did in describing products on the website and targeting customers. Overall, the analyses and results obtained from this project can be used to suggest companies to add pertinence on their advertisements.

3. Method

Data background

My data set consists of information collected from 9168 products(from 324 brands and 143 categories) on Sephora website, including their attributes (name, brand, category, price, ingredients), number of reviews and loves, and a few binary variables(if the product is sold online only, if the product is limited edition, if the product has a limited time offer).

Here is a general review on summary statistics:

```
##        id             brand              category             name
##  Min.   :     50   Length:9168        Length:9168        Length:9168
```

```
##  1st Qu.:1819453   Class :character   Class :character   Class :character
##  Median :2072354   Mode  :character   Mode  :character   Mode  :character
##  Mean   :1962952
##  3rd Qu.:2230591
##  Max.   :2359685
##      size               rating     number_of_reviews      love
##  Length:9168        Min.   :0.00   Min.   :    0.0   Min.   :      0
##  Class :character   1st Qu.:4.00   1st Qu.:   10.0   1st Qu.:   1600
##  Mode  :character   Median :4.00   Median :   46.0   Median :   4800
##                     Mean   :3.99   Mean   :  282.1   Mean   :  16279
##                     3rd Qu.:4.50   3rd Qu.:  210.0   3rd Qu.:  13800
##                     Max.   :5.00   Max.   :19000.0   Max.   :1300000
##      price          value_price         URL             MarketingFlags
##  Min.   :  2.00   Min.   :  2.00   Length:9168        Length:9168
##  1st Qu.: 24.00   1st Qu.: 25.00   Class :character   Class :character
##  Median : 35.00   Median : 35.00   Mode  :character   Mode  :character
##  Mean   : 50.06   Mean   : 51.82
##  3rd Qu.: 59.00   3rd Qu.: 60.00
##  Max.   :549.00   Max.   :549.00
##  MarketingFlags_content   options            details
##  Length:9168            Length:9168        Length:9168
##  Class :character       Class :character   Class :character
##  Mode  :character       Mode  :character   Mode  :character
##
##
##
##   how_to_use        ingredients         online_only       exclusive
##  Length:9168        Length:9168        Min.   :0.0000   Min.   :0.0000
##  Class :character   Class :character   1st Qu.:0.0000   1st Qu.:0.0000
##  Mode  :character   Mode  :character   Median :0.0000   Median :0.0000
##                                        Mean   :0.2348   Mean   :0.2647
##                                        3rd Qu.:0.0000   3rd Qu.:1.0000
##                                        Max.   :1.0000   Max.   :1.0000
##  limited_edition   limited_time_offer
##  Min.   :0.00000   Min.   :0.0000000
##  1st Qu.:0.00000   1st Qu.:0.0000000
##  Median :0.00000   Median :0.0000000
##  Mean   :0.09184   Mean   :0.0003272
##  3rd Qu.:0.00000   3rd Qu.:0.0000000
##  Max.   :1.00000   Max.   :1.0000000
```

And let's take a look at all 143 categories(first 10 rows):

```
## # A tibble: 10 x 2
##    category                 Count
##    <chr>                    <int>
##  1 Accessories                  1
##  2 After Sun Care               2
##  3 Aftershave                  13
##  4 Anti-Aging                  37
##  5 Bath & Body                  9
##  6 Bath & Shower               52
##  7 Bath Soaks & Bubble Bath     6
##  8 BB & CC Cream               22
```

```
##  9 BB & CC Creams              11
## 10 Beauty Supplements         118
```

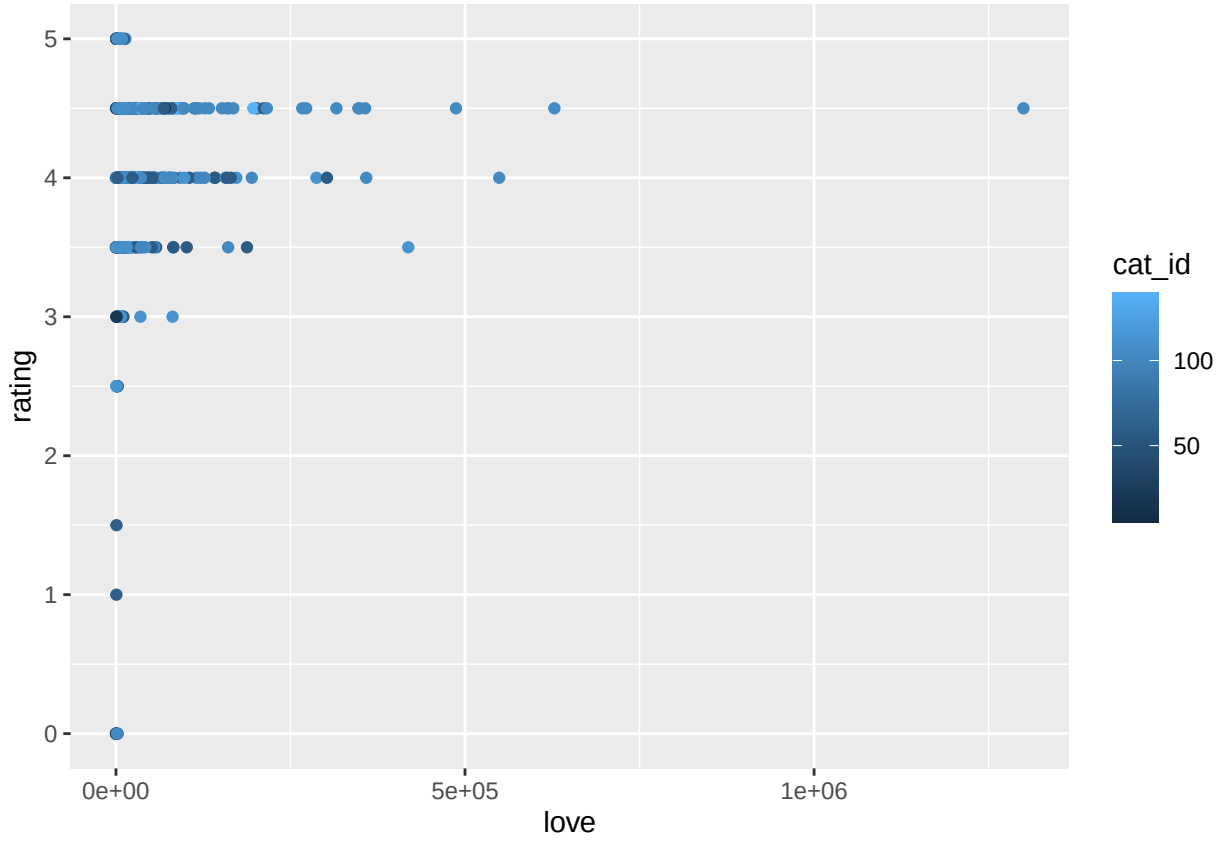
```
## # A tibble: 1 x 2
##   category Count
##   <chr>    <int>
## 1 Perfume    665
```

```
## # A tibble: 1 x 2
##   category    Count
##   <chr>       <int>
## 1 Accessories     1
```

We can see that perfume has the largest number of products and accessories has the least.

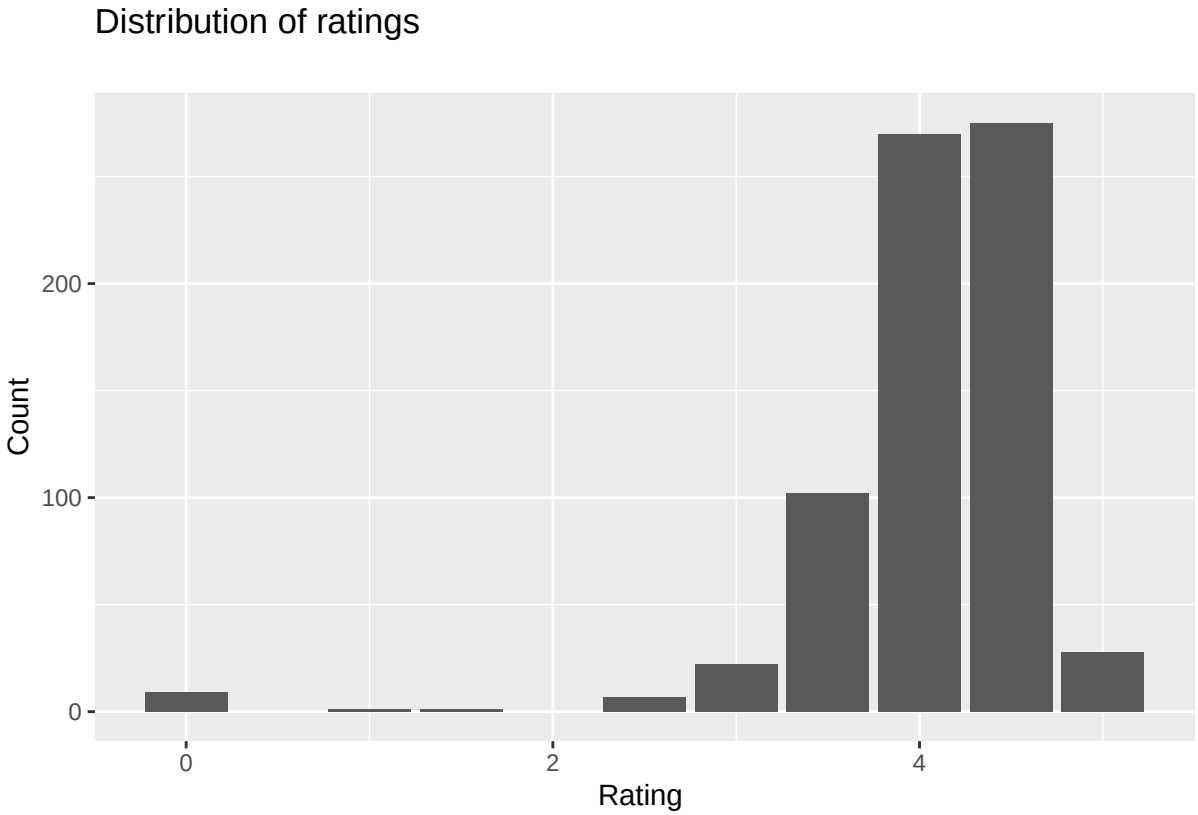
Visulization

In order to make the multilevel model, I assigned a unique ID of each category(cat_id), from 1 to 143. After that, I did random sampling to select 10 categories, and keep products that belong to those 10 categories: Mascara(112), Body Sprays & Deodorant(18), Face Primer(58), Face Sunscreen(61), Toners(140), Hair Styling & Treatments(85), Lipstick(104), Bath & Body(5), Makeup & Travel Cases(108), and Lip Stain(101).




```
##           id            brand category                          name rating
## 4603 1890623 KVD Vegan Beauty Lipstick Everlasting Liquid Lipstick    4.5
##         love
## 4603 1300000
```

As we can see from the scatter plot, the product with largest number of loves, KVD Vegan Beauty lipstick, earns 1,300,000 loves. It has a rating of 4.5, which is relatively high but not the highest.



Most products have ratings above 4, and the most common rating is 4.5.

Set up for the multilevel model

Also, the number of loves and rating are on very different scales, so I divided the number of loves by 1000, and called this new variable love1000, which would be used in my model later.

Now we can build the model as a function of love1000(number of loves divided by 1000) with varying intercepts across categories of product: lmer(rating ~ love1000 + (1 | cat_id))

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: rating ~ love1000 + (1 | cat_id)
##    Data: spr_df
##
## REML criterion at convergence: 1443.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -6.3392 -0.3272  0.1268  0.5517  1.6505
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  cat_id   (Intercept) 0.01872  0.1368
##  Residual             0.42444  0.6515
## Number of obs: 715, groups:  cat_id, 10
##
```

```
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept) 4.0507031  0.0572569  70.746
## love1000    0.0007026  0.0003261   2.154
##
## Correlation of Fixed Effects:
##          (Intr)
## love1000 -0.143
```

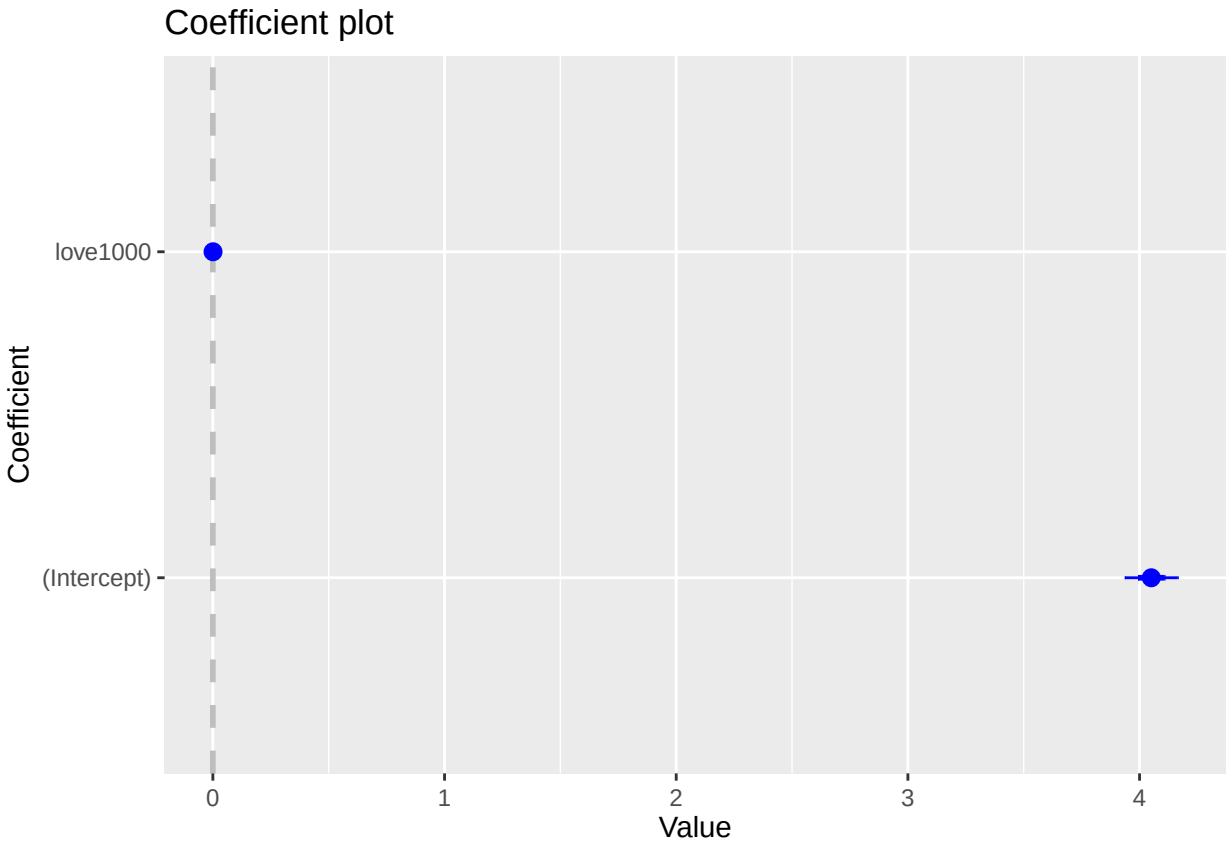
4. Result

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: rating ~ love1000 + (1 | cat_id)
##    Data: spr_df
##
## REML criterion at convergence: 1443.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -6.3392 -0.3272  0.1268  0.5517  1.6505
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  cat_id   (Intercept) 0.01872  0.1368
##  Residual             0.42444  0.6515
## Number of obs: 715, groups:  cat_id, 10
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept) 4.0507031  0.0572569  70.746
## love1000    0.0007026  0.0003261   2.154
##
## Correlation of Fixed Effects:
##          (Intr)
## love1000 -0.143
```

Love1000 = 0.001. For every category, as love1000 increases by 1, rating is predicted to increase by 0.001.

```
## lmer(formula = rating ~ love1000 + (1 | cat_id), data = spr_df)
##             coef.est coef.se
## (Intercept) 4.0507   0.0573
## love1000    0.0007   0.0003
##
## Error terms:
##  Groups   Name        Std.Dev.
##  cat_id   (Intercept) 0.1368
##  Residual             0.6515
## ---
## number of obs: 715, groups: cat_id, 10
## AIC = 1451.9, DIC = 1407.5
## deviance = 1425.7
```

$(0.14)\hat{}2 : (0.65)\hat{}2 = 0.046$




```
## $cat_id
##     (Intercept)     love1000
## 5      4.080642 0.0007025823
## 18     3.991440 0.0007025823
## 58     3.981401 0.0007025823
## 61     3.924591 0.0007025823
## 85     4.048345 0.0007025823
## 101    4.064676 0.0007025823
## 104    4.128024 0.0007025823
## 108    4.161027 0.0007025823
## 112    3.902640 0.0007025823
## 140    4.224245 0.0007025823
##
## attr(,"class")
## [1] "coef.mer"
```

Looking at the coefficients, we can conclude that the regression for category 5(Bath & Body) is y = 4.081 + 0.001 * love1000, y = 3.991 + 0.001 * love1000 for category 18(Body Sprays & Deodorant), y = 4.262 + 0.002 * love1000 for category 58(Face Primer), and so on so forth. The slope(coefficient on love1000) is identical here because they were specified in the model.

Alternatively, if we look at "fixed effects", estimated model averaging over the categories.

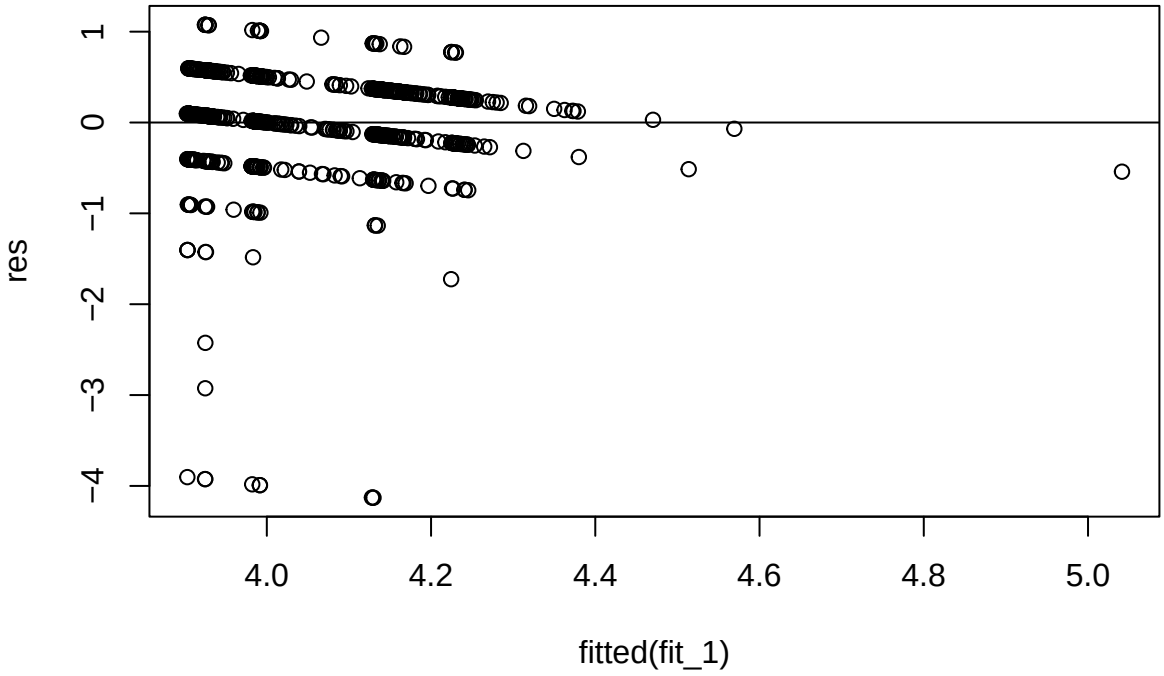
```
##  (Intercept)     love1000
## 4.0507030572 0.0007025823
```

The estimated regression line of an average category is thus y = 4.051 - 0.001*love1000. We can then look at "random effects", category-level errors.

```
## $cat_id
##      (Intercept)
## 5    0.029939085
## 18  -0.059262697
## 58  -0.069302039
## 61  -0.126112441
## 85  -0.002358226
## 101  0.013972761
## 104  0.077321120
## 108  0.110323467
## 112 -0.148063375
## 140  0.173542345
##
## with conditional variances for "cat_id"
```

The intercept is shifted up or down in particular categories. For example, in category 5(Bath & Body), the estimated intercept is 0.03 higher than average, so that the regression line is (4.051 + 0.03) - 0.001 * love1000 = 4.081 - 0.001 * love1000. Categories that have shifted-up intercepts are: Bath & Body(5), Lip Stain(101), Lipstick(104), Makeup & Travel Cases(108), and Toners(140). Categories that have shifted-down intercepts are: Body Sprays & Deodorant(18), Face Primer(58), Face Sunscreen(61), Hair Styling & Treatments(85), and Mascara(112).

Residual Plot



5. Discussion

The results of modeling suggests that larger number of loves will lead to higher ratings of products as well based on the 10 categories I selected randomly. However, due to the size of data set, we cannot conclude that all products follow this rule. Also, the data set is from 3 years ago, Sephora must have updated list of products in this time interval. Therefore, there is limitation of this project. In the future, I will explore this data set further by adding more predictors.

6. Appendix

Code Appendix

```
library(ggplot2)
library(knitr)
library(arm)
library(data.table)
library(foreign)
library(gridExtra)
library(car)
library(stringr)
library(rstan)
library(rstanarm)
library(zoo)
library(dplyr)
library(coefplot)
spr <- read.csv("sephora_website_dataset.csv")
summary(spr)
cat_count <- spr %>%
  group_by(category) %>%
  summarise(`Count` = n())
head(cat_count,10)
cat_count[which.max(cat_count$Count),]
cat_count[which.min(cat_count$Count),]
spr_id <- spr %>%
  group_by(category) %>%
  mutate(`cat_id` = cur_group_id())
set.seed(678)
cats <- sample(unique(spr_id$cat_id),10)
spr_df <- spr_id[spr_id$cat_id %in% cats,]
ggplot(spr_df, aes(x = love, y = rating, color = cat_id)) +
  geom_point()
kvd <- spr[which.max(spr$love),]
kvd[,c("id", "brand", "category", "name", "rating", "love")]
ggplot(spr_df) +
  geom_bar(aes(x = rating,fill = rating)) +
  labs(x = "Rating \n", y = "\n Count ", title = "Distribution of ratings \n")
spr_df$love/1000 -> spr_df$love1000
fit_1 <- lmer(rating ~ love1000 + (1 | cat_id), data = spr_df)
summary(fit_1)
summary(fit_1)
display(fit_1,4)
coefplot(fit_1,title = "Coefficient plot")
coef(fit_1)
fixef(fit_1)
ranef(fit_1)
res <- resid(fit_1)
plot(fitted(fit_1), res)
abline(0,0)
```

7. Supplement

Reference

https://www.kaggle.com/datasets/raghadalharbi/all-products-available-on-sephora-website?resource=download