Sephora

Daniel Mazus, Skye Ritz, Sophia Devoll

Assignment Description

In this project you are going to use the skills that you've learned about regression on a dataset of your own. You may choose any dataset that you wish as long as it is not one that we've already discussed in the course. You may want to consult me about your choice of dataset, just to make sure it is suitable.

After making a suitable dataset choice, you need to complete the following steps:

- Narrative: You need to formulate a question in which you can address using your chosen techniques. This is the overall goal of your analysis.
- You need to perform proper pre-processing and cleaning of the data before your analysis begins. Depending on your data, this step may be fairly short or quite lengthy.
- You need to have a substantial exploratory data analysis (EDA) section. This section should include summaries, graphs (univariate, bivariate, and possibly multivariate), and other techniques from DS 1 to describe your data. You should also investigate possible interactions between variables. Your EDA should show a progression of understanding about the data and your research question.
- You need to choose at least two regression techniques (most likely a multiple linear regression model and a penalized regression method) to use in your analysis. You should explain your modeling choices and how they were informed by your EDA.
- You need to address the assumptions of each method with graphical and/or numeric evidence.
- You need to use cross-validation or a related method to compare the two or more methods.
- You need to come to your final answer using an iterative process that you show throughout your project.
- You need to discuss the shortcomings of your modeling approach. Also, if appropriate, you discuss improvements that could be made.
- You need to discuss how the model approach/output works toward answering the question.

- You need to discuss your major takeaways from the project. This part is meant to be a reflection on what you learned about the data and your increase in knowledge about data science during the process of the project.

Background

We decided to explore a Sephora dataset. Sephora is a well-known beauty store that has built quite an in-store and online empire. At Sephora, one can find skincare, cosmetics, fragrance, beauty tools, haircare, and much more. Sephora's motto is "We Belong to Something Beautiful", let's put this to the test. We propose the question "Are we able to classify purchased categories based off other collected data"?

```r
library(tidymodels)
```

```
-- Attaching packages -------------------------------------- tidymodels 1.1.1 --
```

```
v broom        1.0.5     v recipes      1.0.8
v dials        1.2.0     v rsample      1.2.0
v dplyr        1.1.3     v tibble       3.2.1
v ggplot2      3.4.3     v tidyr        1.3.0
v infer        1.0.5     v tune         1.1.2
v modeldata    1.2.0     v workflows    1.1.3
v parsnip      1.1.1     v workflowsets 1.0.1
v purrr        1.0.2     v yardstick    1.2.0
```

```
-- Conflicts ----------------------------------------- tidymodels_conflicts() --
x purrr::discard() masks scales::discard()
x dplyr::filter()  masks stats::filter()
x dplyr::lag()     masks stats::lag()
x recipes::step()  masks stats::step()
* Dig deeper into tidy modeling with R at https://www.tmwr.org
```

```r
library(tidyverse)
```

```
-- Attaching packages --------------------------------------- tidyverse 1.3.2 --
```

```
v readr   2.1.4     v forcats 1.0.0
v stringr 1.5.0
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x readr::col_factor() masks scales::col_factor()
x purrr::discard()    masks scales::discard()
x dplyr::filter()     masks stats::filter()
x stringr::fixed()    masks recipes::fixed()
x dplyr::lag()        masks stats::lag()
x readr::spec()       masks yardstick::spec()
```

```r
library(rpart)
```

```

Attaching package: 'rpart'

The following object is masked from 'package:dials':

    prune
```

```r
library(baguette)
library(olsrr)
```

```

Attaching package: 'olsrr'

The following object is masked from 'package:datasets':

    rivers
```

```r
library(janitor)
```

```

Attaching package: 'janitor'

The following objects are masked from 'package:stats':

    chisq.test, fisher.test
```

```
sephora <- read_csv(
  "~/Desktop/School/Fall 2023/Data Science II/Project 1/sephora_website_dataset.csv"
  )
```

In uploading our dataset found on Kaggle, we noticed that a few columns would not be beneficial to include. Some of these had too many words, some were inconsistent, and others were specific to every single row. Needless to say, these would not help us in any regression model. We decided to go ahead and remove these columns.

```
sephora_clean <- sephora %>%
  select(-c('id', 'size', 'URL',
            'options', 'details', 'how_to_use', 'ingredients'))
```

Exploratory Data Analysis

Now that we have a clean dataset, lets explore what we have. As stated above, we want to see if we can classify purchased categories. Are we able to do this? Let's see how many categories we are working with.

```
# How many different categories are there?
top_cats <- sephora_clean %>%
  group_by(category) %>%
  count(category) %>%
  arrange(desc(n))
top_cats
```

```
# A tibble: 143 x 2
# Groups:   category [143]
   category                      n
   <chr>                     <int>
 1 Perfume                     665
 2 Moisturizers                451
 3 Face Serums                 384
 4 Value & Gift Sets           378
 5 Face Wash & Cleansers       247
 6 Face Masks                  230
 7 Rollerballs & Travel Size   228
 8 Hair Styling Products       224
 9 Eye Palettes                202
10 Eye Creams & Treatments     191
```

```
# i 133 more rows
```

Wow! There are 143 different categories. To better see different models, let us just look at the top two categories. We see that these two categories are perfume and moisturizers. Below, we are going to name a new variable that only includes the data that are categorized by perfume and moisturizers.

```
# Only looking at the top 2 categories of data as discovered above
sephora_clean <- sephora_clean %>%
  filter((category %in% c("Perfume", "Moisturizers")))
```

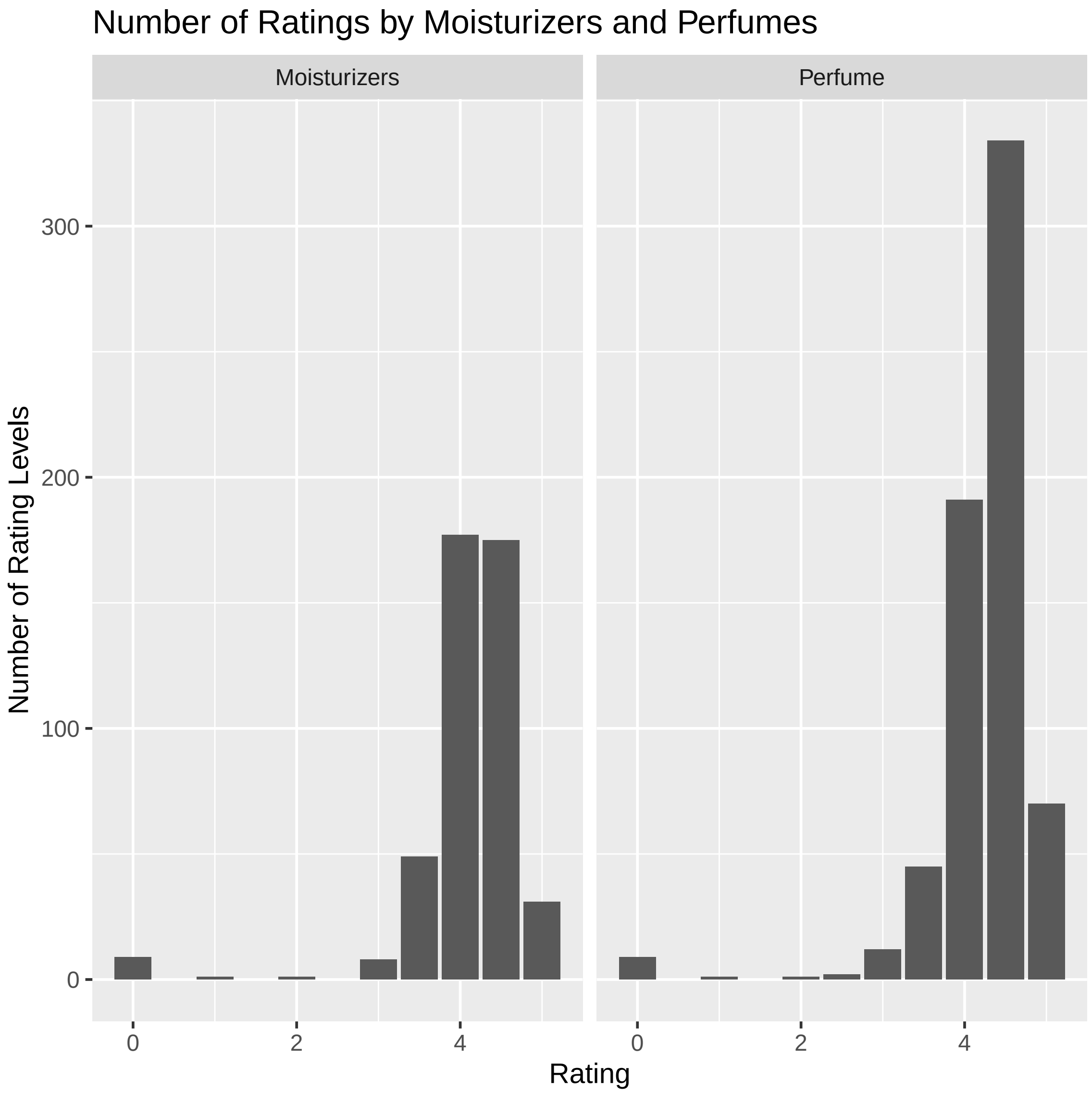
Let us get a feel for our chosen categories and to fully understand how much Sephora actually carries. Brands are different companies that create a product, these products are then categorized (i.e. perfume and moisturizers). Below, we see that there are 155 brands for perfumes and moisturizers. Ultimately, we will not use brand as a predictor in our model. Brand is a categorical variable however, one brand could carry multiple categories. For example, Dior makes perfumes but Dior also makes lip products. Therefore, it would be nearly impossible to classify brands.

```
# How many different brands are there?
sephora_clean %>%
  group_by(brand) %>%
  count(brand) %>%
  arrange(desc(n))
```

```
# A tibble: 155 x 2
# Groups:   brand [155]
   brand                n
   <chr>            <int>
 1 TOM FORD            54
 2 CLINIQUE            39
 3 Jo Malone London    28
 4 Dior                27
 5 CLEAN RESERVE       26
 6 Atelier Cologne     23
 7 CHANEL              23
 8 Fresh               21
 9 Givenchy            21
10 Gucci               21
# i 145 more rows
```

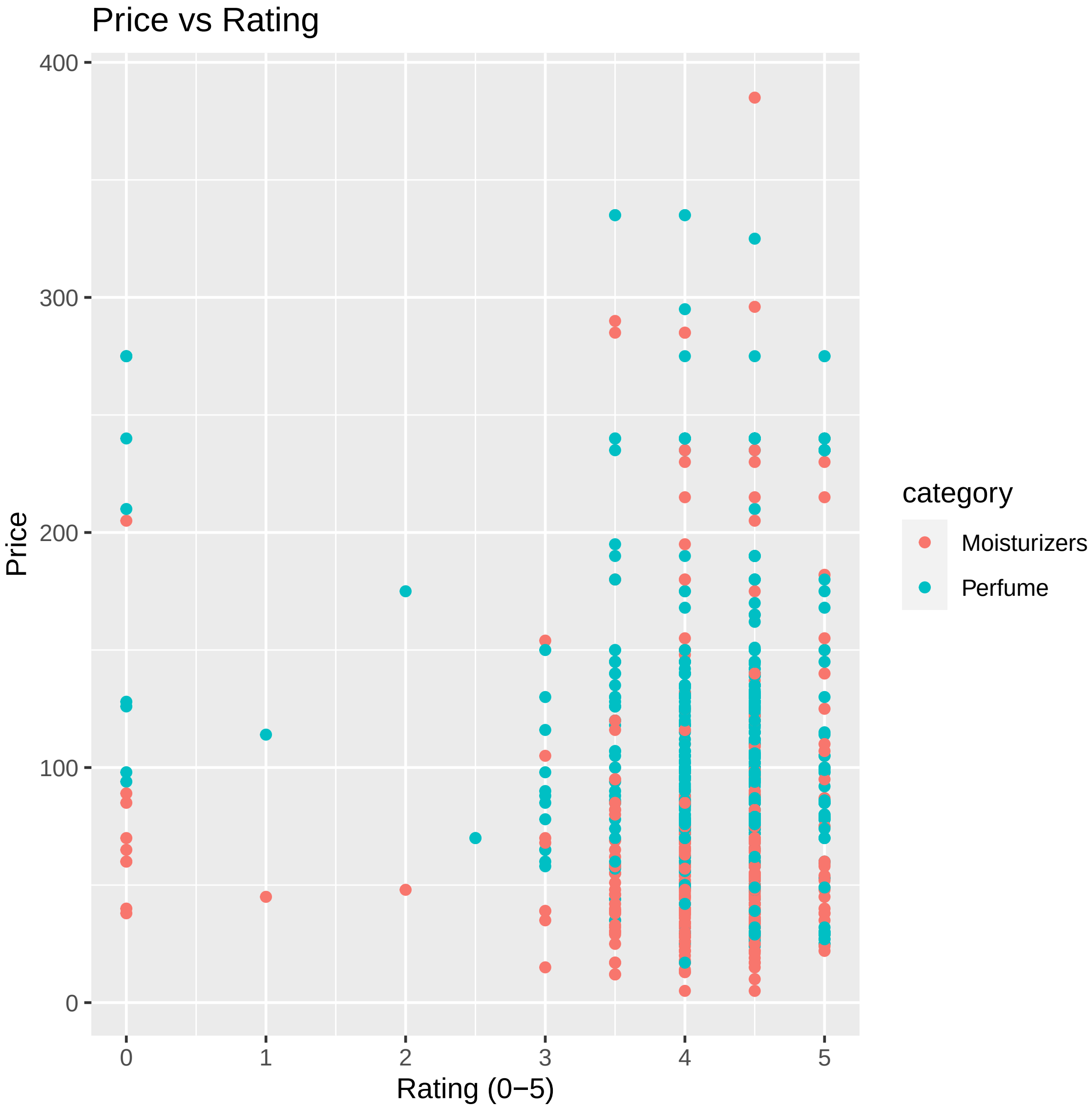
Onto the fun part of EDA's… graphs! Shown below is a bar graph showing different ratings, we split the graph to show ratings for moisturizers and ratings for perfume. The rating in this dataset is 0-5. The rating is given by a customer who purchased a product and decided to give their opinion. Oh no, opinion, not the best word to hear in statistics. But in this case, we want opinions because the higher the ratings, the more people will buy and ultimately, certain perfumes or moisturizers will stay on the market.

```r
sephora_clean %>%
  ggplot(aes(x = rating)) + geom_bar() + facet_wrap(vars(category)) +
  labs(x = "Rating",
       y = "Number of Rating Levels",
       title = "Number of Ratings by Moisturizers and Perfumes")
```



We see that both of these bar graphs are skewed to the left. This is because most of our data lies at 4 or above. For Sephora, this is great since a 4 rating is good. We do see that there is some data at 0 but not a significant amount. This seems to be a good predictor. Now let us look at rating in coherts to price. In other words, is an expensive item going to have a higher rating?

```
sephora_clean %>%
  ggplot(aes(x = rating, y = price, color = category)) + geom_point() +
  labs(x = "Rating (0-5)",
       y = "Price",
       title = "Price vs Rating")
```



This scatterplot has nearly no correlation. We do see most of our data is at the 4 or more mark. But we cannot say that a more expensive item has a higher rating. Our y-axis shows price, and we can see 4 star ratings range from $0-$250. Ultimately, there is not a correlation between price and ratings. Now let's look at whether online purchases can tell us anything about what kinds of categories will be purchased. Let 0 represent in-store and 1 represent online purchase.

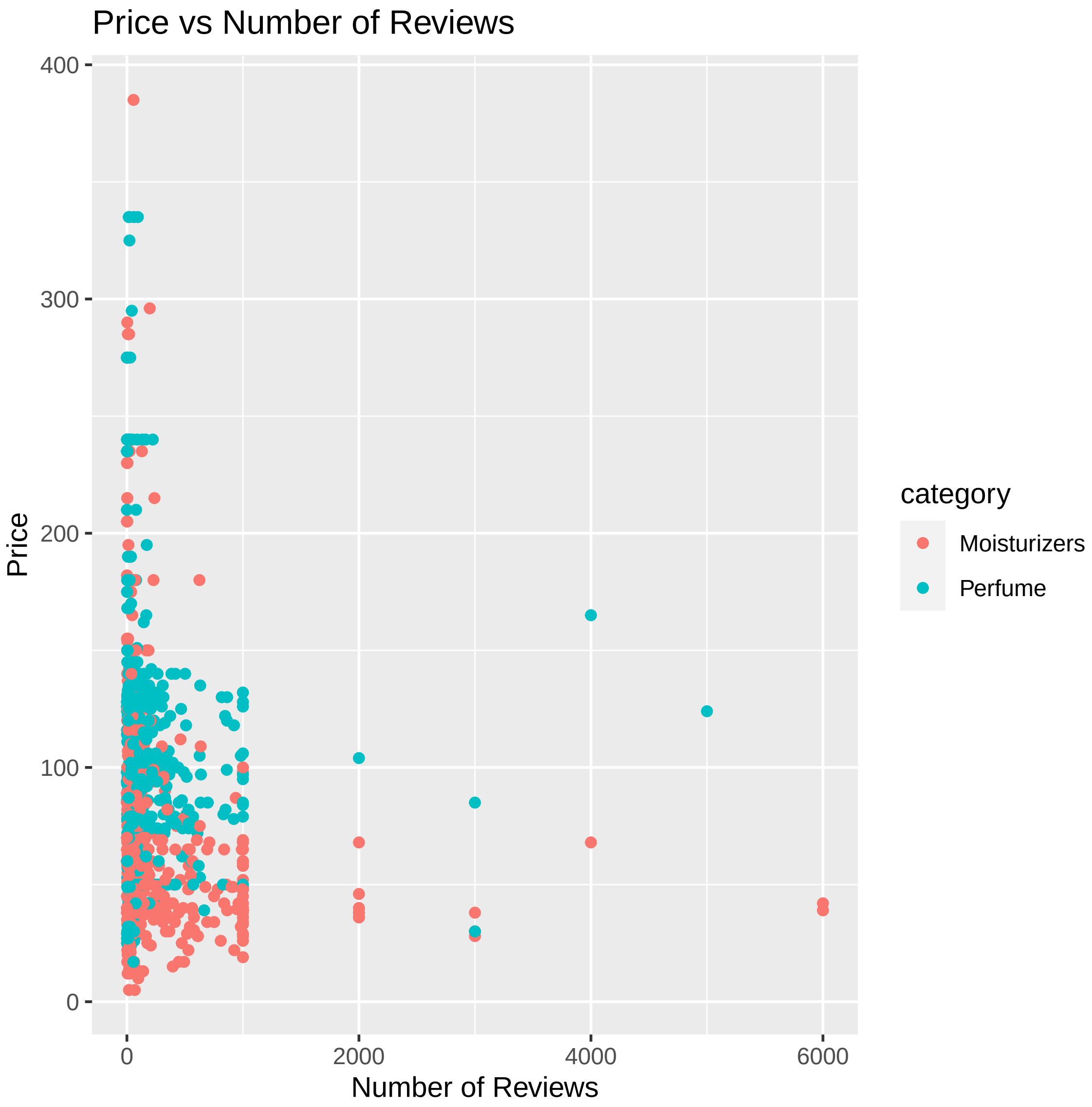
```
sephora_clean %>%
  group_by(category) %>%
  count(online_only)
```

```
# A tibble: 4 x 3
# Groups:   category [2]
  category     online_only     n
  <chr>              <dbl> <int>
1 Moisturizers           0   363
2 Moisturizers           1    88
3 Perfume                0   552
4 Perfume                1   113
```

Based off this table, we can see that there is a significant amount of in-store shopping. Let's note that even though this seems unreasonable as online shopping is so prevalent nowadays, this is a pre-covid dataset. Anyways, the proportions of moisturizers and perfumes bought online and in-store are nearly the same. Therefore, this would be a difficult predictor to use when trying to classify category.

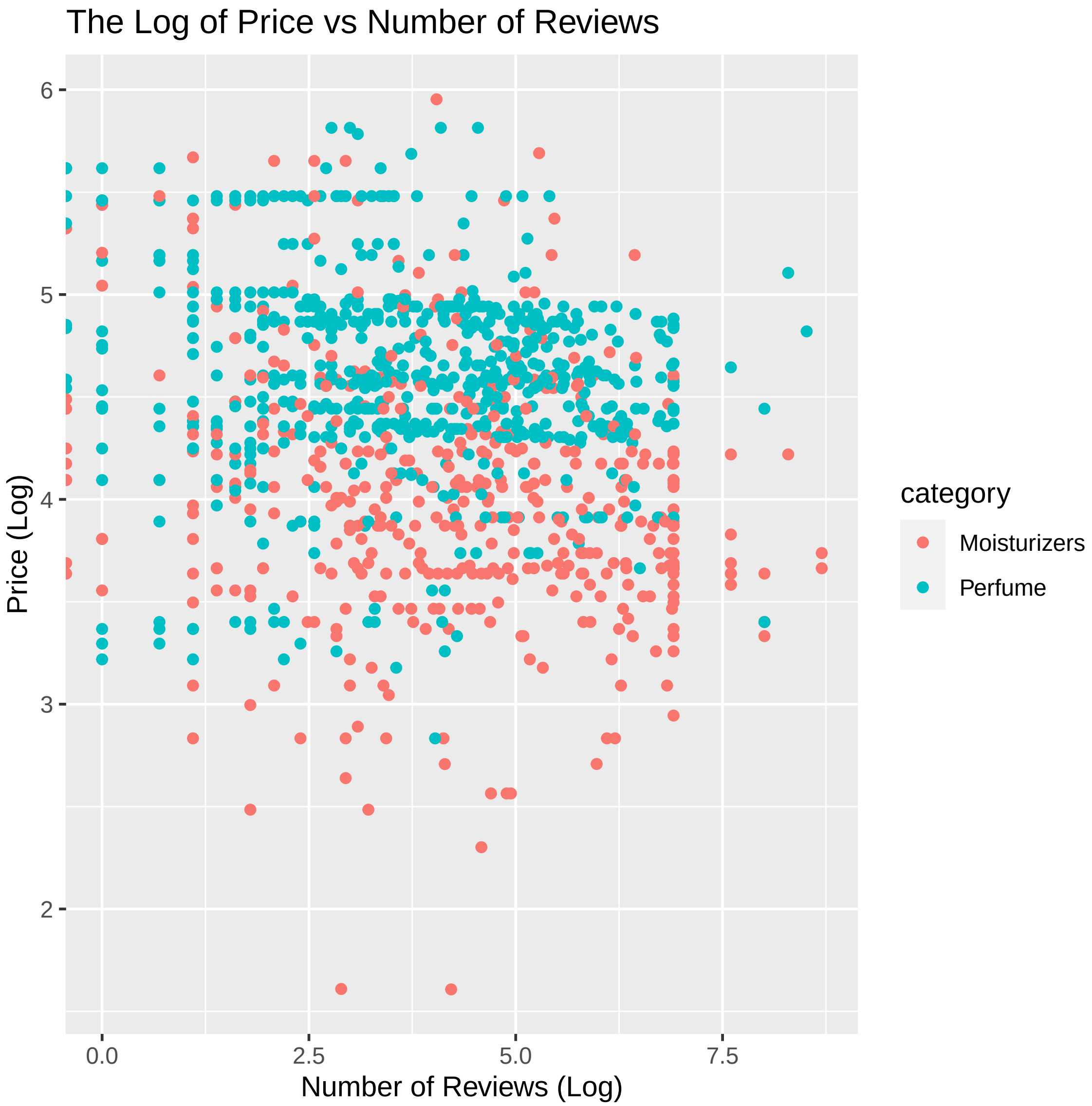
Now let us look at Number of Reviews and Price to see if there was any correlation or significance in these variables while coloring each point by the category specified with it.

```
sephora_clean %>%
  ggplot(aes(x = number_of_reviews, y = price, color = category)) +
  geom_point() + labs(x = "Number of Reviews",
                      y = "Price",
                      title = "Price vs Number of Reviews")
```



Looking at this graph above, we can see that there are a number of outliers here that are most likely affecting the shape and look of the graph. We can see that there are more reviews for lower price levels compared to higher prices, showing some sort of exponential curve or decreasing exponential curve. Let us try a log transformation to see if we can see how the data is shaped when normalized.

```
sephora_clean %>%
  ggplot(aes(x = log(number_of_reviews), y = log(price), color = category)) +
  geom_point() + labs(x = "Number of Reviews (Log)",
                      y = "Price (Log)",
                      title = "The Log of Price vs Number of Reviews")
```



This graph seems better to look at and interpret some meaning out of it beyond just realizing the shape of the data. We can see that there is almost a clear definition between the categories with some overlap in the middle of Number of Reviews and Price. But lower values in Price are more associated with Moisturizers while higher Prices are associated with Perfume. Number of Reivews have some effect to show here but is hard to describe it without a model.

Models

Our first model we will look at will be a logistic regression with price, rating, number of reviews, and love as the predictors and the response variable of category. Recall, we are only looking at the top two categories which are Perfume and Moisturizers. We have to do a little cleaning to show the binary variable of category, so we call Perfume = 1 and Moisturizers = 0.

```r
sephora_clean <- sephora_clean %>%
  mutate(category = case_when(
    category == "Perfume" ~ 1,
    category == "Moisturizers" ~ 0
  ))

first_model <- lm(category ~ price + rating + number_of_reviews + love,
                   data = sephora_clean)
summary(first_model)
```

```
Call:
lm(formula = category ~ price + rating + number_of_reviews + 
    love, data = sephora_clean)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.5434 -0.4320  0.2241  0.3871  0.8123 

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)    
(Intercept)       -7.142e-03  8.471e-02  -0.084    0.933    
price              3.099e-03  2.374e-04  13.054  < 2e-16 ***
rating             8.073e-02  1.915e-02   4.215  2.7e-05 ***
number_of_reviews -6.525e-05  4.211e-05  -1.550    0.122    
love              -1.387e-06  1.076e-06  -1.289    0.198    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.449 on 1111 degrees of freedom
Multiple R-squared:  0.1664,	Adjusted R-squared:  0.1634 
F-statistic: 55.44 on 4 and 1111 DF,  p-value: < 2.2e-16
```

Using these predictors, we can see that only price and rating are statistically significant in predicting category. This does not come as surprising as we saw in our EDA that price and rating were potentially good predictors. We have an extremely low R-squared value. Only 16.64% of variability in the data can be explained in our model. Let us create a new model using only these predictors.

```r
second_model <- lm(category ~ price + rating, data = sephora_clean)
summary(second_model)
```

```

Call:
lm(formula = category ~ price + rating, data = sephora_clean)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.5654 -0.4369  0.2314  0.3991  0.7199 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.0256388  0.0851037  -0.301    0.763    
price        0.0032523  0.0002352  13.828  < 2e-16 ***
rating       0.0753166  0.0192192   3.919 9.44e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4518 on 1113 degrees of freedom
Multiple R-squared:  0.1546,	Adjusted R-squared:  0.1531 
F-statistic: 101.7 on 2 and 1113 DF,  p-value: < 2.2e-16
```

We can see that all of our predictors are significant now. However, this model is not very good. The R-squared value decreased which means even less variability is explained. But, our residual standard error increased (slightly), which is also bad because we want to have a lower residual standard error as that means a model is more reliable. Let's return to our model using all 4 predictors and perform backward, forward, and stepwise selection.

```r
variable_model <- lm(category ~ price + rating +
                       love + number_of_reviews,
                     data = sephora_clean)
```

```r
#forward Selection
ols_step_forward_aic(variable_model)
```

```
                           Selection Summary
--------------------------------------------------------------------------
Variable                 AIC       Sum Sq       RSS       R-Sq      Adj. R-Sq
--------------------------------------------------------------------------
price                 1412.059    38.405     230.336    0.14291      0.14214
rating                1398.766    41.540     227.201    0.15457      0.15305
number_of_reviews     1386.711    44.383     224.358    0.16515      0.16290
--------------------------------------------------------------------------
```

```
#backward Selection
ols_step_backward_aic(variable_model)
```

```
                   Backward Elimination Summary
-------------------------------------------------------------------
Variable        AIC         RSS       Sum Sq      R-Sq      Adj. R-Sq
-------------------------------------------------------------------
Full Model    1387.042    224.022    44.719    0.16640      0.16340
love          1386.711    224.358    44.383    0.16515      0.16290
-------------------------------------------------------------------
```

```
#stepwise
ols_step_both_aic(variable_model)
```

```
                                  Stepwise Summary
--------------------------------------------------------------------------------------
Variable                 Method       AIC          RSS       Sum Sq      R-Sq      Adj. R-Sq
--------------------------------------------------------------------------------------
price                  addition    1412.059     230.336    38.405    0.14291      0.14214
rating                 addition    1398.766     227.201    41.540    0.15457      0.15305
number_of_reviews      addition    1386.711     224.358    44.383    0.16515      0.16290
--------------------------------------------------------------------------------------
```

Forward selection and stepwise selection summarized that the best predictors for category are price, rating, and number of reviews. The backward selection also got rid of the love variable. Therefore we conclude that love is not a good predictor. If we were to make a multiple logistic regression model, we would use price, rating, and number of reviews.

```r
# Factoring the category
sephora_clean <- sephora_clean %>%
  mutate(category = case_when(
    category == 1 ~ factor(1),
    category == 0 ~ factor(0)
  ))
```

This step is here to help build our future models since the response must be an integer or factor instead of numeric or character.

Splitting, Training, and Testing the Data

We will be building a logistic, decision tree, and bagged tree model to compare them and see which one performs the best at trying to predict the category given the variables price, rating, and number of reviews. We are using these variables again given the forward, backward, and stepwise all choosing them as the best variables given the data.

We are using a 80/20 split of training and testing while strataing the categorical variable so we can properly predict without having a training set without one or the other.

```r
sephora_split <- initial_split(sephora_clean,
                               prop = 0.8,
                               strata = category)

sephora_train <- sephora_split %>%
  training()

sephora_train <- sephora_split %>%
  testing()
```

Logistic Model

```r
folds <- vfold_cv(sephora_clean, v = 10)

log_model <- logistic_reg(mixture = tune(),
                          penalty = tune()) %>%
  set_engine("glmnet") %>%
  set_mode("classification")
```

```
log_recipe <- recipe(category ~ price + rating + number_of_reviews,
                     data = sephora_clean)

log_wf <- workflow() %>%
  add_recipe(log_recipe) %>%
  add_model(log_model)

log_grid <- grid_regular(mixture(),
                         penalty(),
                         levels = 10)

log_tune_grid <- tune_grid(
  log_wf,
  resamples = folds,
  grid = log_grid
)

log_tune_grid %>%
  select_best() %>%
  finalize_workflow(log_wf, .) %>%
  last_fit(sephora_split) %>%
  collect_metrics()
```

```
Warning: No value of `metric` was given; metric 'roc_auc' will be used.
```

```
# A tibble: 2 x 4
  .metric  .estimator .estimate .config
  <chr>    <chr>          <dbl> <chr>
1 accuracy binary         0.719 Preprocessor1_Model1
2 roc_auc  binary         0.764 Preprocessor1_Model1
```

We have an accuracy of 75.45% and an ROC-AUC value of 0.7958. This seems pretty good for a logistic model just based off of those variables especially with an R-squared value of about 16% in the variables when using the different variable selection methods.

Decision Tree

```r
sephora_folds <- vfold_cv(sephora_train)

sephora_decision_model <- decision_tree(cost_complexity = tune(),
                                        tree_depth = tune()) %>%
  set_engine("rpart") %>%
  set_mode("classification")

sephora_dec_wf <- workflow() %>%
  add_recipe(log_recipe) %>%
  add_model(sephora_decision_model)

decision_tree_grid <- grid_regular(cost_complexity(),
                                   tree_depth(),
                                   levels = 10)

decision_tune_grid <- tune_grid(
  sephora_dec_wf,
  resamples = sephora_folds,
  grid = decision_tree_grid
)

decision_best <- decision_tune_grid %>%
  select_best("accuracy")

final_decision_wf <- sephora_dec_wf %>%
  finalize_workflow(decision_best)

final_decision_wf %>%
  last_fit(sephora_split) %>%
  collect_metrics()
```

```
# A tibble: 2 x 4
  .metric  .estimator .estimate .config
  <chr>    <chr>          <dbl> <chr>
1 accuracy binary         0.772 Preprocessor1_Model1
2 roc_auc  binary         0.765 Preprocessor1_Model1
```

We have an accuracy of 80.80% and an ROC-AUC value of 0.7898. Comparing to our logistic model, we have done better by about an increase of 5% accuracy but slightly worse in our

ROC-AUC value. I would still say that this model is better in general because of our increase in accuracy being higher despite having the slightly lower ROC-AUC value. Our last model that we will build to compare will be a bagged tree. We are using a bagged tree here to get rid of high variance in the model to see if that is a factor or not since decision trees tend to have high variance in the model.

```
sephora_bag_model <- bag_tree(cost_complexity = tune(),
                              tree_depth = tune()) %>%
  set_engine("rpart") %>%
  set_mode("classification")

sephora_bag_wf <- workflow() %>%
  add_recipe(log_recipe) %>%
  add_model(sephora_bag_model)

bag_grid <- grid_regular(cost_complexity(),
                         tree_depth(),
                         levels = 10)

bag_tune_grid <- tune_grid(
  sephora_bag_wf,
  resamples = sephora_folds,
  grid = bag_grid
)

bag_best <- bag_tune_grid %>%
  select_best("accuracy")

final_bag_wf <- sephora_bag_wf %>%
  finalize_workflow(bag_best)

final_decision_wf %>%
  last_fit(sephora_split) %>%
  collect_metrics()
```

```
# A tibble: 2 x 4
  .metric  .estimator .estimate .config
  <chr>    <chr>          <dbl> <chr>
1 accuracy binary         0.772 Preprocessor1_Model1
2 roc_auc  binary         0.765 Preprocessor1_Model1
```

We have an accuracy of 80.80% and ROC-AUC value of 0.7898. These are the same exact values that we had in the decision tree with about double the computing time. I would say that this would mean that there is not a lot that this model can do about the variance in the models built. It is still better than the logistic model but in some cases might prefer the logistic model because of the computing time. Say we had a much larger dataset than what we have here, the computing time on that would be much larger with only a 5% increase in accuracy.

Concluding

Coming back to our question of can we classify categories correctly, I would say that we would be able to correctly classify them given our Decision Tree model with 80% accuracy. We went through different steps to see how variables were correlated with each other or not and then moved onto basic logistic regression models to see if we could correctly identify variables but chose to use a variable selection model to pick them out. Choosing the variables as number of reviews, price, and rating to predict our model. Despite those having a very low R-squared value, we could still have our accuracy of the Decision Tree Model be about 80%.

The shortcomings of this model though come in the form of it being a data set from 2016 so the data is most likely out of date now especially after COVID, very complicated data set that we had to dim down to help us work with, and the limited number of variables to work with without using text analysis or some other complicated model to help predict. Initially, we thought this would be a simple data set to work with, but after trying different ideas we realized that this data set was not simple and can happen to any real world data. This was probably our biggest takeaway about working with real world things rather than built in data sets or anything of that nature. We learned how to adapt different types of tuning models to logistic regression specifically while combining different components such as the variable selection to help us build those models.