Mercari Price Suggestion - Exploratory Analysis

Introduction

Product pricing is a tough challenge, especially at scale. For example, clothing has strong seasonal pricing trends and is heavily influenced by brand names, while electronics have fluctuating prices based on product specs.

In this Kaggle competition, Mercari, Japan’s biggest shopping app, is reaching out to the community of Kagglers to develop predictive models for suggestion of item price.

Data

**train.tsv, test.tsv**

The files consist of a list of product listings. These files are tab-delimited.

* *train\_id* or *test\_id* - the id of the listing
* *name* - the title of the listing. Note that we have cleaned the data to remove text that look like prices (e.g. $20) to avoid leakage. These removed prices are represented as [rm]
* *item\_condition\_id* - the condition of the items provided by the seller
* *category\_name* - category of the listing
* *brand\_name*
* *price* - the price that the item was sold for. This is the target variable that you will predict. The unit is USD. This column doesn’t exist in test.tsv since that is what you will predict. shipping - 1 if shipping fee is paid by seller and 0 by buyer
* *item\_description* - the full description of the item. Note that we have cleaned the data to remove text that look like prices (e.g. $20) to avoid leakage. These removed prices are represented as [rm]

**sample\_sumbmission.csv**

A sample submission file in the correct format.

* *test\_id* - matches the *test\_id* column in *test.tsv*
* *price*

Without further ado, let’s dive into the analysis!

# Exploratory analysis

library(data.table)

library(dplyr)

library(stringr)

library(ggplot2)

library(treemapify)

library(quanteda)

library(gridExtra)

Let’s load the training data first.

mercari <- fread("data/train.tsv", sep = "\t")

Read 77.6% of 1482535 rows

Read 1482535 rows and 8 (of 8) columns from 0.315 GB file in 00:00:03

First, let’s look into whether any columns are worth dropping straight away because they will not be useful for neither analysis nor training the models.

summary(mercari)

train\_id name item\_condition\_id category\_name brand\_name price shipping

Min. : 0 Length:1482535 Min. :1.000 Length:1482535 Length:1482535 Min. : 0.00 Min. :0.0000

1st Qu.: 370634 Class :character 1st Qu.:1.000 Class :character Class :character 1st Qu.: 10.00 1st Qu.:0.0000

Median : 741267 Mode :character Median :2.000 Mode :character Mode :character Median : 17.00 Median :0.0000

Mean : 741267 Mean :1.907 Mean : 26.74 Mean :0.4473

3rd Qu.:1111900 3rd Qu.:3.000 3rd Qu.: 29.00 3rd Qu.:1.0000

Max. :1482534 Max. :5.000 Max. :2009.00 Max. :1.0000

item\_description

Length:1482535

Class :character

Mode :character

It seems like train\_id is irrelevant for both tasks, so let’s remove it.

mercari$train\_id <- NULL

What immediately stands out about this dataset is that there are only two features which are numeric (apart from price), namely: item\_condition\_idand shipping. This suggests that a lot of potentially useful information is contained in the text columns: item\_description, category\_name, nameand brand.

Let’s look at whether there are any NA values.

apply(mercari, 2, anyNA)

name item\_condition\_id category\_name brand\_name price shipping item\_description

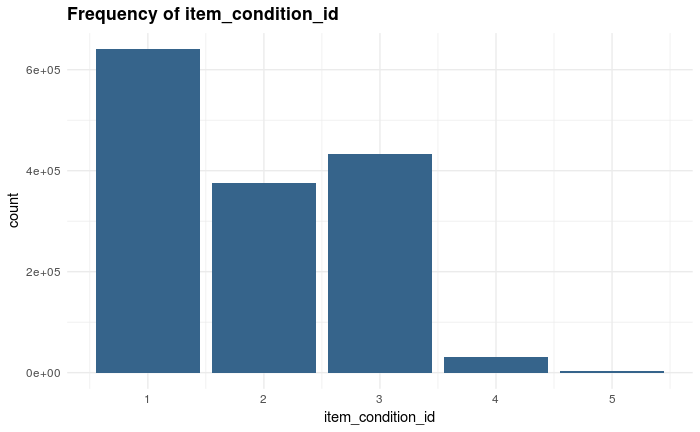
FALSE FALSE FALSE FALSE FALSE FALSE FALSE

Luckily, there are none! Let’s look into numeric columns now.

## item\_condition\_id

common\_theme <- theme(plot.title = element\_text(face = "bold", size = 16)) + theme\_minimal()

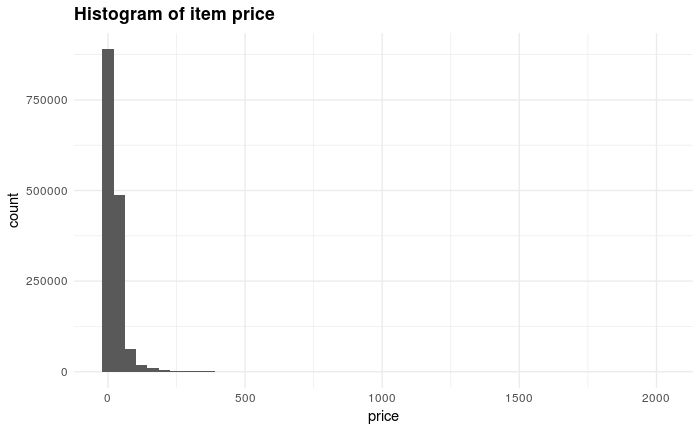
ggplot(mercari, aes(x = item\_condition\_id)) + geom\_bar(fill = "steelblue4") + ggtitle("Frequency of item\_condition\_id") + common\_theme



Clearly, items with condition 1 are the most common, followed by 3, 2, 4 and, finally, 5. It is worth noting that there is a large difference in the frequency for classes 1-3 and 4-5.

It is worth investigating how the item price is related to its condition, so let’s do just that now.

ggplot(mercari, aes(x = price)) + geom\_histogram(bins = 50) + ggtitle("Histogram of item price") + common\_theme

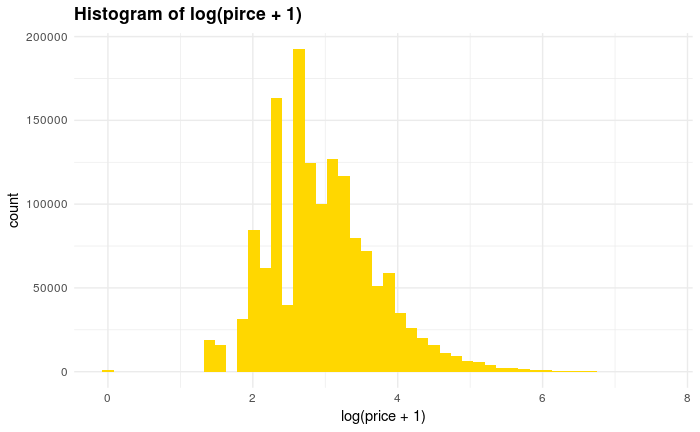


range(mercari$price)

[1] 0 2009

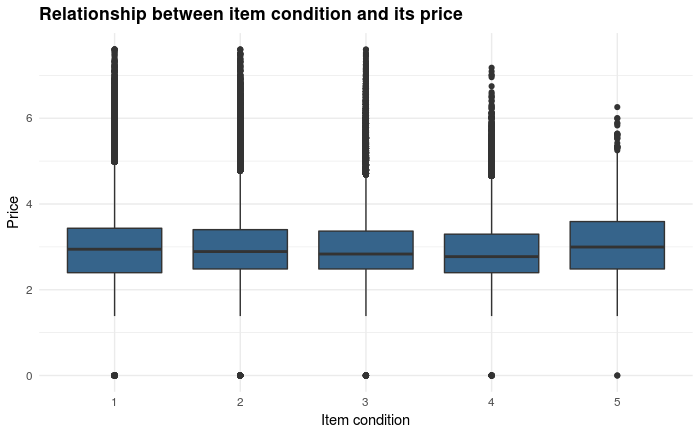
The price variable is very left-skewed, with a very long right tail. The minimal price is 0 (most likely people giving things away for free), while the highest is 2009. Let’s transform the variable using log(x+1) to get rid of skewness (the +1 is there to avoid taking log(0)).

ggplot(mercari, aes(x = log(price + 1))) + geom\_histogram(bins = 50, fill = "gold") + ggtitle("Histogram of log(pirce + 1)") + common\_theme



Let’s now look at the joint relationship between *price* and *item\_condition\_id*

ggplot(mercari, aes(x = as.factor(item\_condition\_id), y = log(price + 1))) + geom\_boxplot(fill = "steelblue4") + ggtitle("Relationship between item condition and its price") + common\_theme + xlab("Item condition") + ylab("Price")



It seems like there is no clear trend between item condition and its price. Moreover, it is not clear whether condition 1 means the best or the worst item quality, and this is impossible to determine from the boxplot above. My best guess is that condition 1 corresponds to the highest quality due to the maximum values of log(price + 1) being the highest.

It is worth noting that condition 5 has the highest median price, but this is a less significant result due to very small sample size for that condition category.

## shipping

Let’s now have a look at the shipping variable.

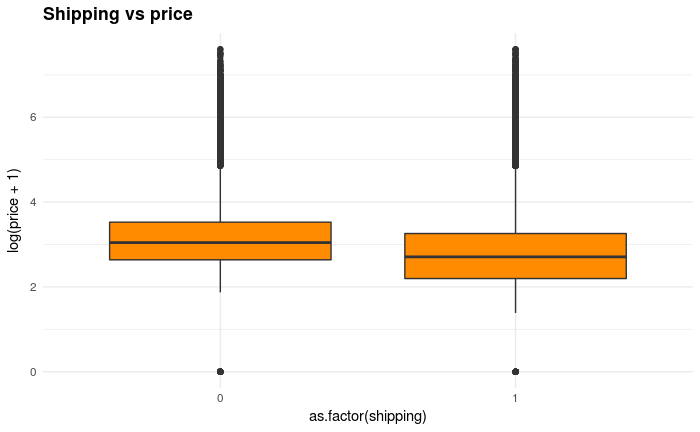
table(mercari$shipping)

0 1

819435 663100

The 0 category is dominant here. We can now look at how shipping is realted to price

ggplot(mercari, aes(x = as.factor(shipping), y = log(price + 1))) + geom\_boxplot(fill = "darkorange") + ggtitle("Shipping vs price") + common\_theme



ggplot(mercari, aes(x = log(price + 1), fill = factor(shipping))) + geom\_density(alpha = 0.75, adjust = 2.5) + common\_theme + ggtitle("Density of price by shipping category")



It seems like the median item price is higher when shipping = 0.

## category\_name, brand\_name

Let’s now take a closer look at category\_name and brand\_name, both of which are textual variables. Note that category actually contains 4 sub-categories, so that we can split this column into 4 new ones.

newcols <- str\_split\_fixed(mercari$category\_name, "/", 4)

mercari <- mercari %>% mutate(cat1 = newcols[, 1], cat2 = newcols[, 2], cat3 = newcols[, 3], cat4 = newcols[, 4])

mercari %>% summarise(cat1\_unique = length(unique(cat1)), cat2\_unique = length(unique(cat2)), cat3\_unique = length(unique(cat3)), cat4\_unique = length(unique(cat4)))

options(repr.plot.width=7, repr.plot.height=7)

mercari %>% group\_by(cat1, cat2) %>% count() %>% ungroup() %>% ggplot(aes(area = n, fill = cat1, label = cat2, subgroup = cat1)) + geom\_treemap() + ggtitle("Hierarchy of 1st and 2nd order categories") + geom\_treemap\_subgroup\_text(min.size = 0, grow = T, alpha = 0.5, colour = "black", fontface = "italic") + geom\_treemap\_text(colour = "white", place = "topleft", reflow = T) + theme(legend.position = "null")

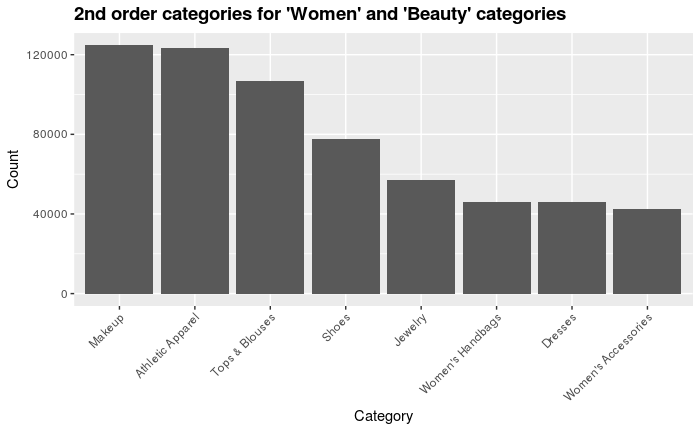


Women and beauty are the two most common majro categories, so we can have a closer look at 2nd order categories for these two only. Since there are a lot of 2nd order categories, we will only look at the top 10 ones (> 40000 associated items).

options(repr.plot.width=7, repr.plot.height=7)

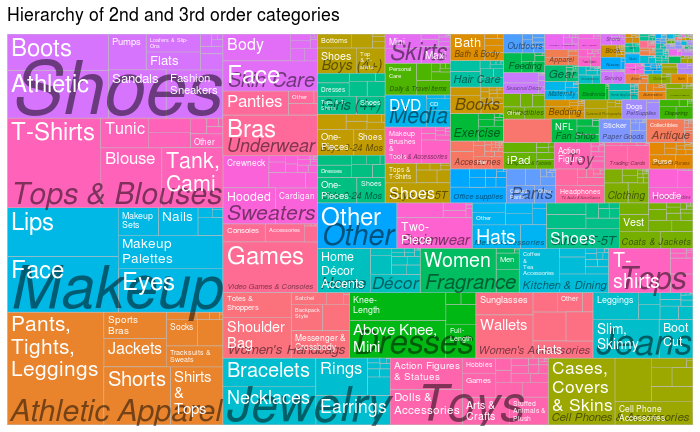
summ <- mercari %>% filter(cat1 == "Women" | cat1 == "Beauty") %>% count(cat2) %>% filter(n > 40000)

ggplot(summ, aes(x = reorder(cat2, -n), y = n)) + geom\_bar(stat = "identity") + ggtitle("2nd order categories for 'Women' and 'Beauty' categories") + theme(axis.text.x = element\_text(angle = 45, hjust = 1)) + theme(axis.text.x = element\_text(angle = 45), plot.title = element\_text(size = 14, face = "bold")) + xlab("Category") + ylab("Count")



We can also have a look at the 2nd and 3rd order categories in a similar way.

mercari %>% group\_by(cat2, cat3) %>% count() %>% ungroup() %>% ggplot(aes(area = n, fill = cat2, label = cat3, subgroup = cat2)) + geom\_treemap() + ggtitle("Hierarchy of 2nd and 3rd order categories") + geom\_treemap\_subgroup\_text(min.size = 0, grow = T, alpha = 0.5, colour = "black", fontface = "italic") + geom\_treemap\_text(colour = "white", place = "topleft", reflow = T) + theme(legend.position = "null")



It’s interesting to look closer at the items which don’t have a brand associated with them. This constitutes a large proportion of the dataset:

Hide

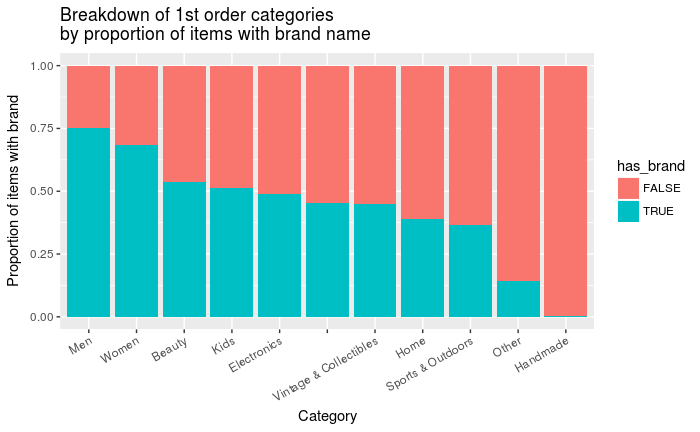
mean(mercari$brand\_name == "")

[1] 0.4267569

Hide

mercari$has\_brand <- mercari$brand\_name != ""

mercari %>% ggplot(aes(x = reorder(cat1, -has\_brand), fill = has\_brand)) + geom\_bar(position = "fill") + xlab("Category") + ylab("Proportion of items with brand") + ggtitle("Breakdown of 1st order categories\nby proportion of items with brand name") + theme(axis.text.x = element\_text(angle = 30, hjust = 1))



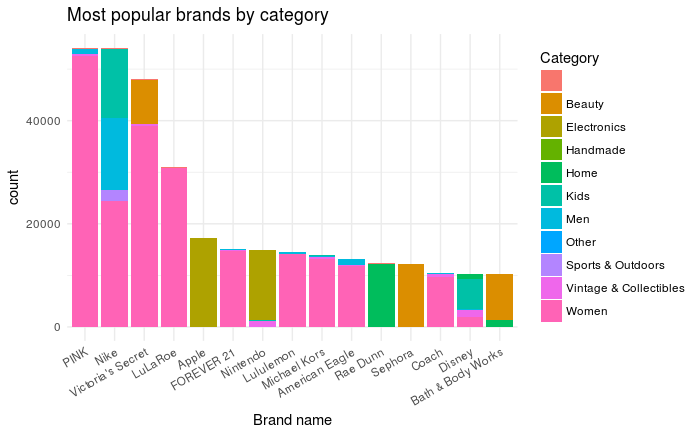
We can see that there are huge differences in proportions of items with a brand; while over 75% of items in the *Men* category have a brand name, nearly no items have one in the *Handmade* category.

Let’s now look at what brands are the most popular.

Hide

top\_brands <- mercari %>% filter(has\_brand == T) %>% count(brand\_name) %>% arrange(desc(n)) %>% head(15)

mercari %>% filter(brand\_name %in% top\_brands$brand\_name) %>% ggplot(aes(x = factor(brand\_name, levels = top\_brands$brand\_name), fill = cat1)) + geom\_bar() + theme\_minimal() + theme(axis.text.x = element\_text(angle = 30, hjust = 1)) + xlab("Brand name") + labs(fill = "Category") + ggtitle("Most popular brands by category")



The top brands are clearly dominated by the *Women* category. We can also look at how expensive each brand is.

Hide

options(repr.plot.width=30, repr.plot.height=30)

brand\_median\_prices <- mercari %>% filter(has\_brand == T & brand\_name %in% top\_brands$brand\_name) %>% group\_by(brand\_name) %>% summarise(median(price))

colnames(brand\_median\_prices)[2] <- "median\_price"

brand\_median\_prices <- brand\_median\_prices %>% arrange(desc(median\_price))

mercari %>% filter(has\_brand == T & brand\_name %in% brand\_median\_prices$brand\_name) %>% ggplot(aes(x = factor(brand\_name, levels = rev(brand\_median\_prices$brand\_name)), y = price, fill = as.factor(shipping))) + geom\_boxplot() + coord\_flip() + xlab("Brand name") + ylab("Price") + ggtitle("Brand name vs price") + labs(fill = "Shipping") + common\_theme

