Code----

---

title: "Mercari Price Suggestion - Exploratory Analysis"

output: html\_notebook

---

# 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

```{r,message=FALSE,warning=FALSE}

library(data.table)

library(dplyr)

library(stringr)

library(ggplot2)

library(treemapify)

library(quanteda)

library(gridExtra)

```

Let's load the training data first.

```{r}

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

```

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.

```{r}

summary(mercari)

```

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

```{r}

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\_id\* and \*shipping\*. This suggests that a lot of potentially useful information is contained in the text columns: \*item\_description\*, \*category\_name\*, \*name\* and \*brand\*.

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

```{r}

apply(mercari, 2, anyNA)

```

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

##item\_condition\_id

```{r}

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.

```{r}

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

range(mercari$price)

```

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)).

```{r}

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\*

```{r}

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.

```{r}

table(mercari$shipping)

```

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

```{r}

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.

```{r}

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)))

```

It seems like the most unique categories are in the 3rd category level. Let's now look at the hierarchy of 1st and 2nd categories.

```{r}

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).

```{r}

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.

```{r}

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:

```{r}

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

```

```{r}

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.

```{r}

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.

```{r, fig.width=10, fig.height=10}

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

```

We can see that Michael Kors is the most expensive brand, while Apple seems to have a highly left-skewed distribution, with a lot of highly-priced items. Moreover, it seems like the buyers pay the highest premium for shipping for Apple products too.

Let's now do a little analysis of item descriptions.

First, we should perform some basic preprocessing by setting the description to "NA" where == "no description yet". We will not yet transform the text to lowercase as this prevents the corpus() function of package quanteda from counting the sentences properly.

```{r}

#mercari$item\_description <- tolower(mercari$item\_description)

mercari[mercari$item\_description == "No description yet", "item\_description"] = NA

```

Let's count the number of characters and plot it against mean of log(price + 1).

```{r}

mercari$desc\_len <- nchar(mercari$item\_description)

mean\_log\_price <- mercari %>% group\_by(desc\_len) %>% summarise(mean(log(price + 1)))

colnames(mean\_log\_price)[2] <- "mean\_log\_price"

ggplot(mean\_log\_price, aes(x = desc\_len, y = mean\_log\_price)) + geom\_point() + stat\_smooth(method = "loess") + xlab("Description length") + ylab("Mean log price") + ggtitle("Mean log price vs item description length") + common\_theme

```

There seems to be no clear relationship between description length and item price.

Let's now create a corpus and a document term matrix and count the number of 1, 2 and 3-grams.

```{r, fig.width=10}

# 1-grams

desc\_corpus <- corpus(mercari$item\_description)

dtm1 <- dfm(desc\_corpus, ngrams = 1, remove = c("rm", stopwords("english")), remove\_punct = T, remove\_numbers = T, stem = T)

top\_1grams <- data.frame(term = names(topfeatures(dtm1, n = 20)), count = topfeatures(dtm1, n = 20))

p1 <- ggplot(top\_1grams, aes(x = reorder(term, count), y = count)) + geom\_bar(stat = "identity", fill = "steelblue4") + xlab("1-gram") + ylab("Count") + ggtitle("Frequency of 1-grams") + coord\_flip() + common\_theme

# Take 20% of corpus for computing 2-grams

desc\_corpus\_20 <- corpus\_sample(desc\_corpus, size = floor(ndoc(desc\_corpus)\*0.2))

# 2-grams

dtm2 <- dfm(desc\_corpus\_20, ngrams = 2, remove = c("rm", stopwords("english")), remove\_punct = T, remove\_numbers = T, concatenator = " ")

top\_2grams <- data.frame(term = names(topfeatures(dtm2, n = 20)), count = topfeatures(dtm2, n = 20))

p2 <- ggplot(top\_2grams, aes(x = reorder(term, count), y = count)) + geom\_bar(stat = "identity", fill = "darkgreen") + xlab("2-gram") + ylab("Count") + ggtitle("Frequency of 2-grams") + coord\_flip() + common\_theme

# 3-grams

dtm3 <- dfm(desc\_corpus\_20, ngrams = 3, remove = c("rm", stopwords("english")), remove\_punct = T, remove\_numbers = T, concatenator = " ")

top\_3grams <- data.frame(term = names(topfeatures(dtm3, n = 20)), count = topfeatures(dtm3, n = 20))

p3 <- ggplot(top\_3grams, aes(x = reorder(term, count), y = count)) + geom\_bar(stat = "identity", fill = "darkred") + xlab("3-gram") + ylab("Count") + ggtitle("Frequency of 3-grams") + coord\_flip() + common\_theme

grid.arrange(p1, p2, p3, nrow = 1)

```

For 1-grams, the most common ones are \*new\* and \*size\*. There is a relatively sharp count drop for 2-grams, where \*brand new\* is the most common by far, followed by roughly 2 times less common \*free shipping\*. When it comes to 3-grams, there is no marked sharp drop in frequency, and the most common trigrams are \*price is firm\* and \*new with tags\*.