# Mercari Price Prediction

# CONTENTS

## 1) Problem Statement:

Product pricing gets even harder at scale, considering just how many products are sold online. Clothing has strong seasonal pricing trends and is heavily influenced by brand names, while electronics have fluctuating prices based on product specs. Mercari, Japan's biggest community-powered shopping app, knows this problem deeply. They'd like to offer pricing suggestions to sellers, but this is tough because their sellers are enabled to put just about anything, or any bundle of things, on Mercari's marketplace.In this competition, Mercari's challenging you to build an algorithm that automatically suggests the right product prices. You'll be provided user-inputted text descriptions of their products, including details like product category name, brand name, and item condition.

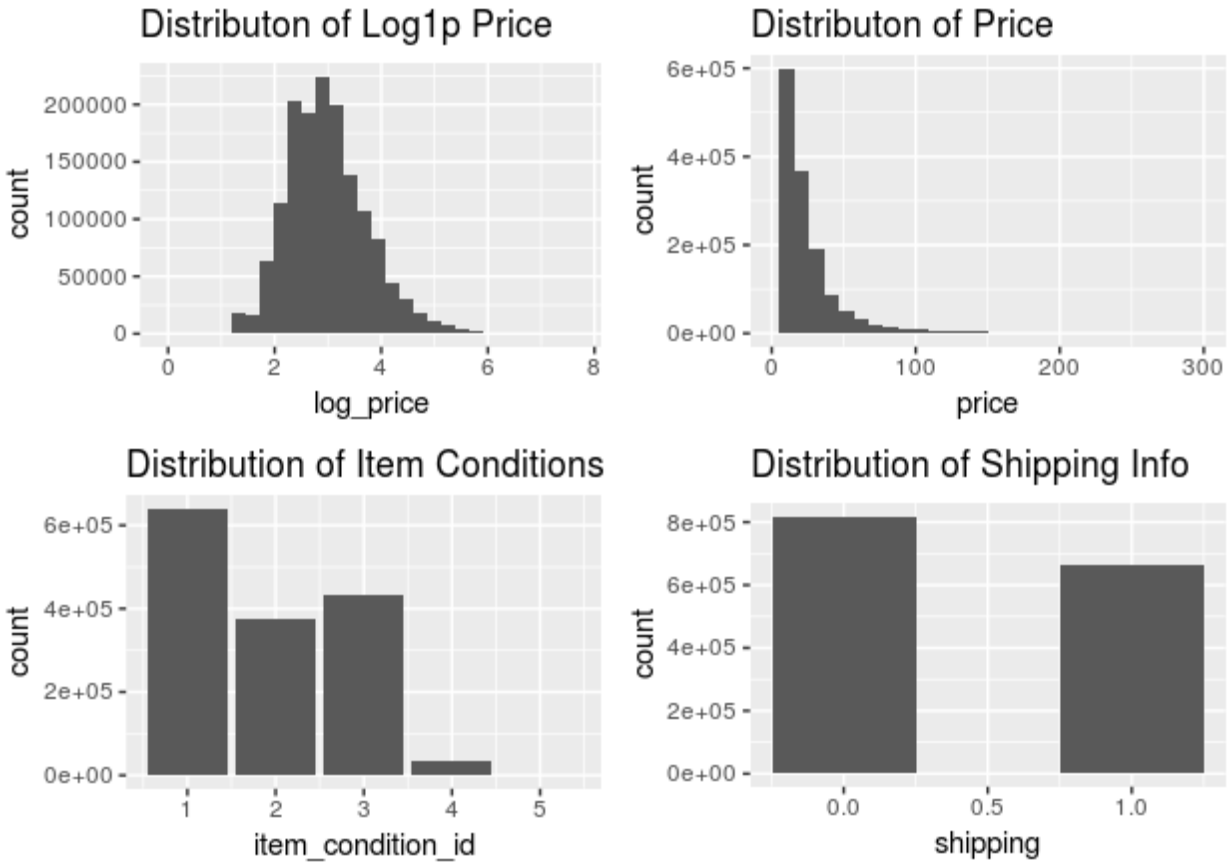
## 2. Data used:

The data has 1482535 observations with 7 features.

## Variable used

1. **train_id :**A unique key for each item.
2. **name**: The item's name as a string.
3. **item_condition_id**: A factor with 5 levels. As the plot below shows, the mean prices for different conditions are really close and it's hard to guess which whether higher / lower condition id is better so far.
4. **category_name**: The category of the item.
5. **brand_name**: The brand name of the item. Nearly half of the items do not have a brand.
6. **shipping**: A binary indicator of the shipping information. (1 if shipping fee is paid by seller and 0 by buyer)
7. **item_description**: A long string containing the raw text of the item description. ~5% of the items do not have a description.
8. **price(target variable)**: Numerical value of the product.It various from 0 – 2009.

**3. Exploration of Numerical Variable:** Among the 8 variables,two of them are numerical.Train_id and price(target variable). The item price ranges from 0 to $2009. Let's look at the histogram of prices. Because price is likely skewed and because there are some 0s, we'll plot the log of price + 1. The (log price + 1) appears to be centered around 3 and has a longer right tail due to the 0 bound on the left.




There are many zero value product ,we can deal with that by 2 methods.Either predicting the price using others or removing it.I have remove the zero valued product.

## 4. Exploration of Categorical Variable:

There are two category variable.

Table of Item_condition :

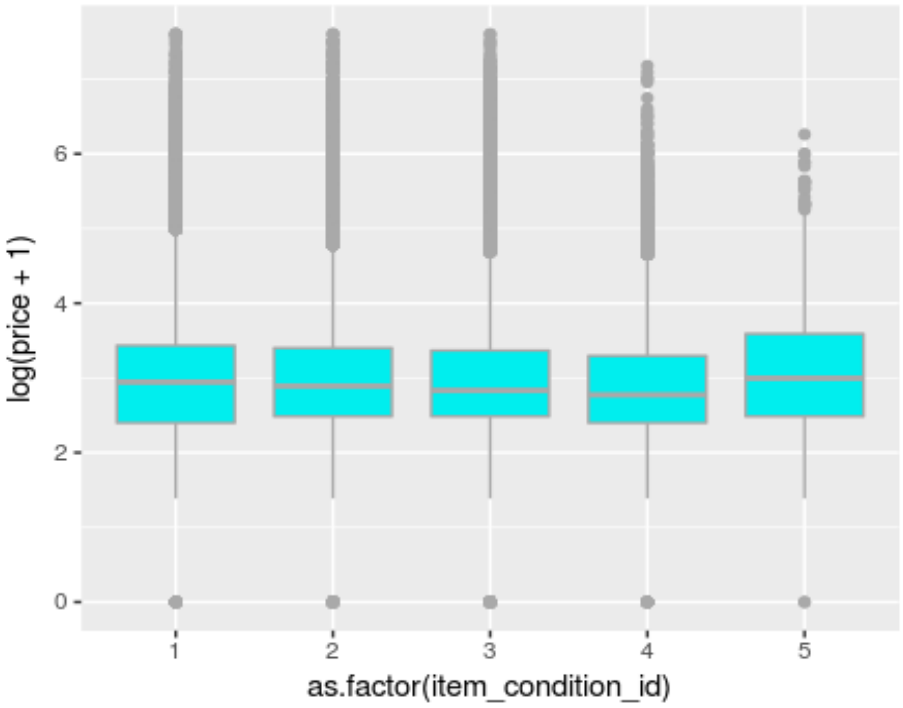
```
     1      2      3      4      5
640549 375479 432161  31962   2384
```

Table of Shipping :

```
     0      1
819435 663100
```

**Item_condition:**

The item condition ranges from 1 to 5. There are more items of condition 1 than any other. Items of condition 4 and 5 are relatively rare. It's not clear from the data description what the ordinality of this variable is. My assumption is that since conditions 4 and 5 are so rare these are likely the better condition items. We can try and verify this. If a higher item condition is better, it should have a positive correlation with price. Let's see if that is the case.



Looking at the average price by .Condition 5 clearly has the highest price, however condition 1 has the next-highest price, followed by condition 2, then 3, then 4.condition 1 is the best product and 5 is the worst. Condition 5 is a bit of an anomaly in that it has the highest price. However, it also has the fewest number of items, so our point estimate has the most uncertainty.

## Shipping:

The shipping variable is a dummy variable indicating whether the shipping for the item is paid for by the seller (1) or not (0).

Items where the shipping fee is paid by the seller will be higher-priced. However, there are a number of conflating factors. This may be true within specific product categories and item conditions, but not when comparing items on the aggregate.


Items where the shipping is paid by the seller have a lower average price.

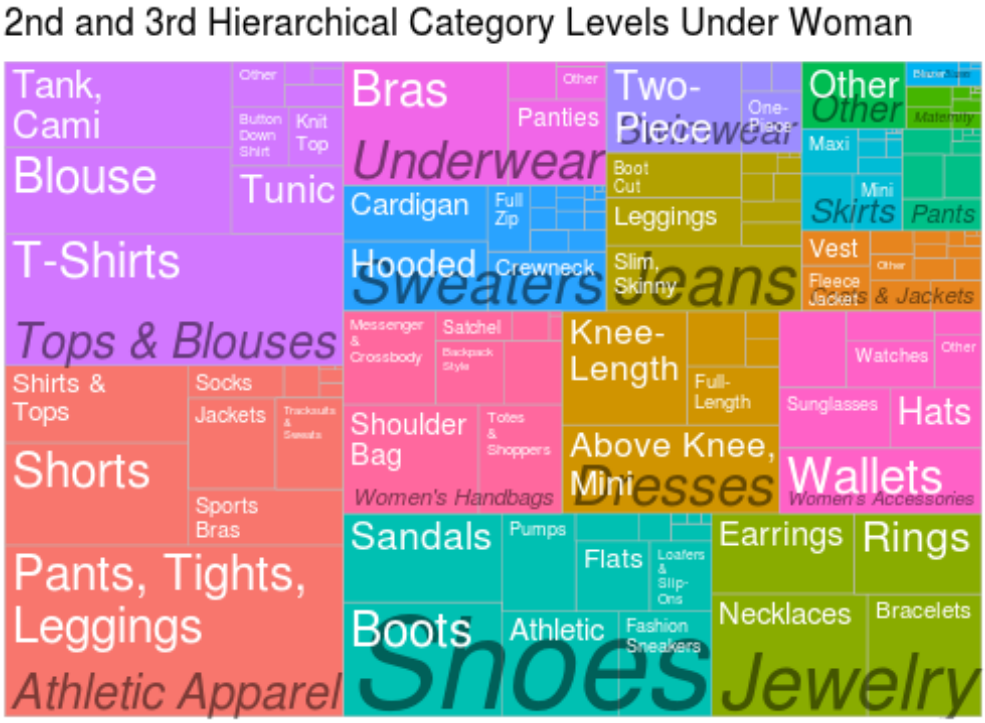
## 5. Exploration of Character Variable:

There are 4 string variable.Name, brand_name, category_name and item_description.

**Category_name** is actually encoded as three or four hierarchical levels splitted by /.We can split the category names and store them into 4 columns. The major category (1st category) only has 11 levels and we can make distinguishable visualizations on them. From the 2nd level on the # of levels are too many to visualize.

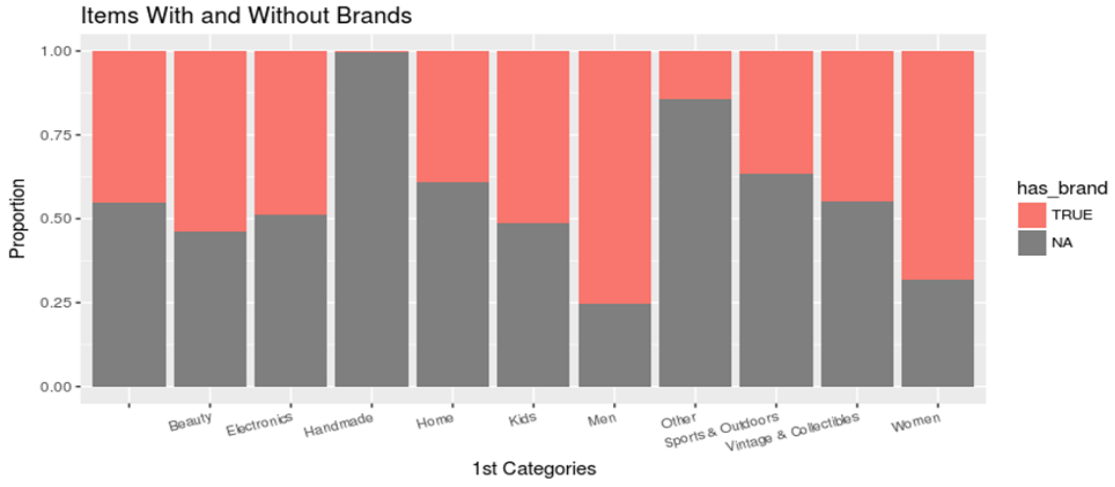
Most items only have three levels of categories. But the 4th level exists with 8 unique sub-categories and 4389 items. For modeling perspective it may be fine to combine it with 3rd levels but for analysis purpose I extract and keep the 4th level here.

1st and 2nd Hierarchical Category Levels

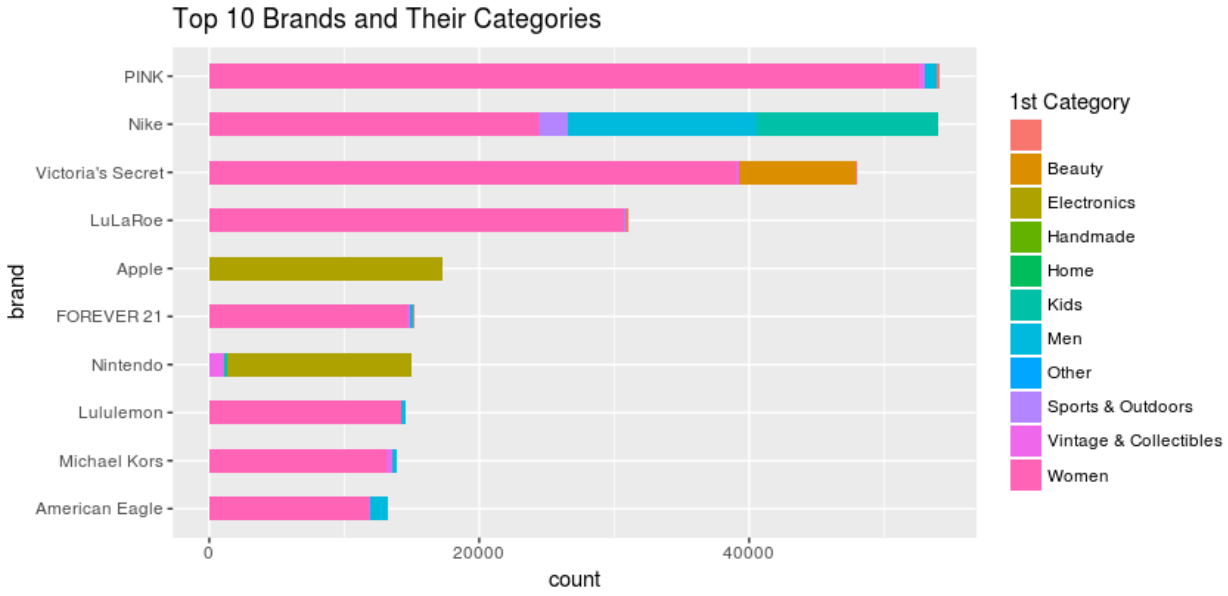



## Brand_Name

Nearly half of the items don't have brands. The proportions of items that have brands vary in different categories. For example, Nearly all handmade items don't have brand names, of course.For brands, they are not in a hierarchical order and there are too many to be fitted in one graph. So I plotted the count of top 10 most frequent brands for a rough look. Each brand contains items from 1 or more major categories. Not surprisingly, the top brands are dominated by women items except Apple and Nintendo.

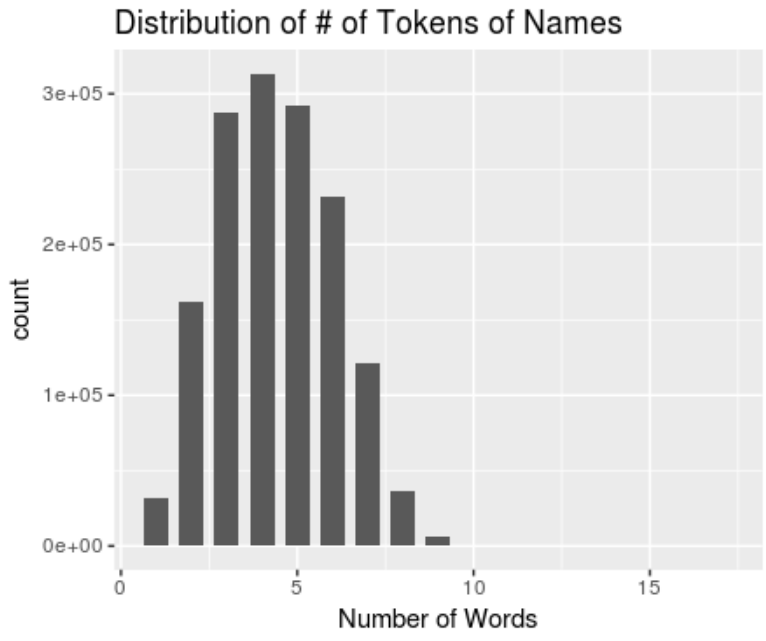


Top 10 most sold brands and categories:



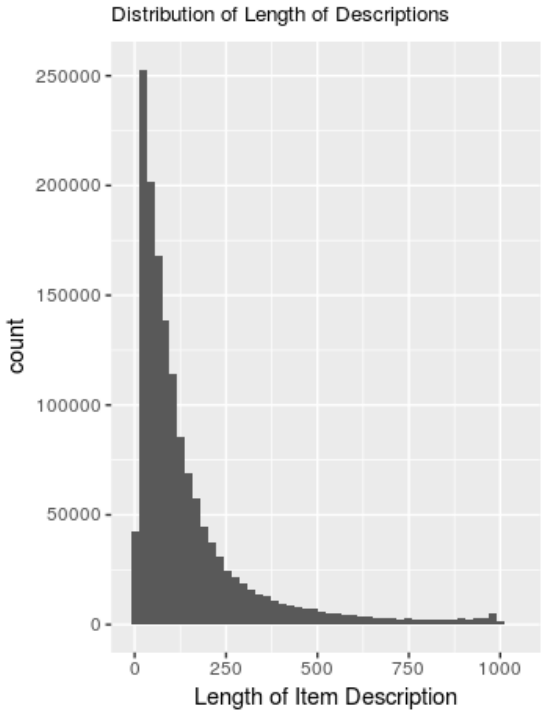

## Name:

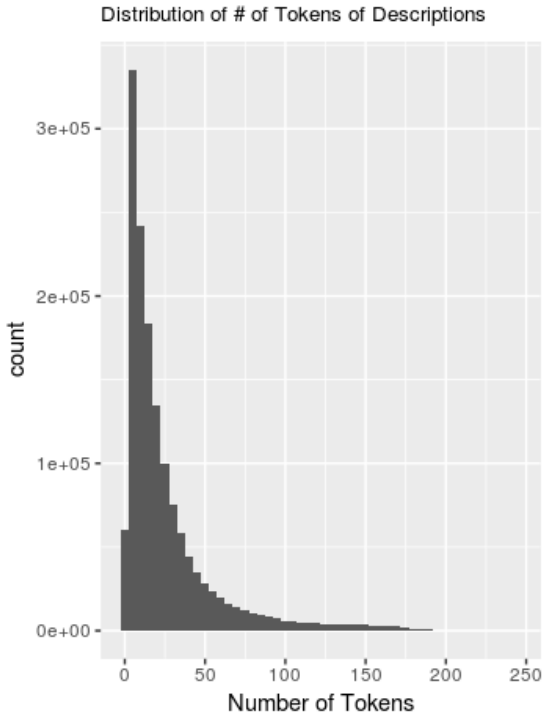
Name is a very important feature —— it directly tell us what the items are. But it's hard to be categorized or one-hot encoded because there are 1225273 uniqle names for all the items. We should find some ways to extract the information from the item name.



## Item_Description:

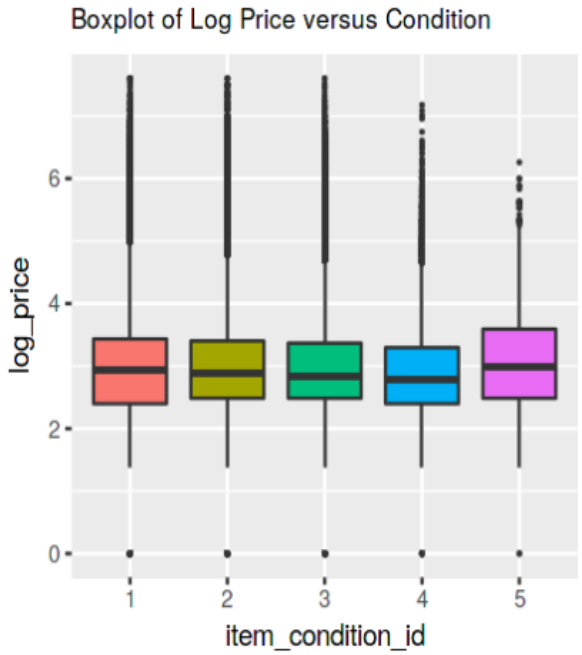
The description in unstructured data, so in order to explore it fully, we'll need to do some text processing and normalization.Let's see a document-term matrix of the descriptions. To do this, we use the dfm() function and pass in our corpus object. We start with individual words, remove english stopwords and punctuation, and stem words.

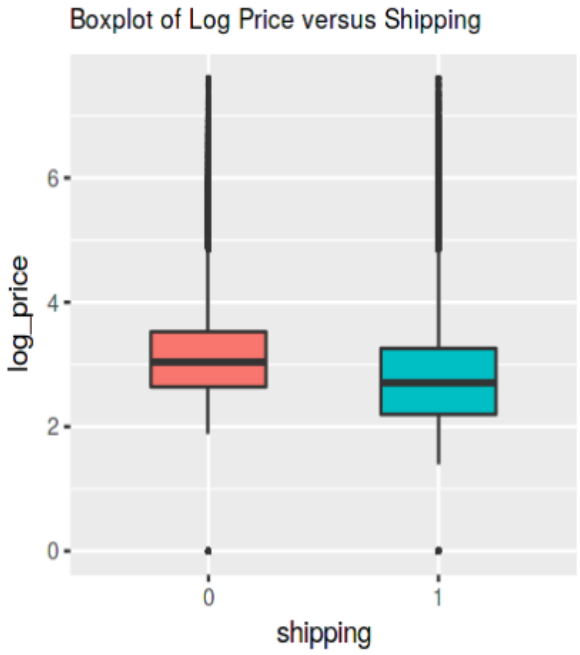



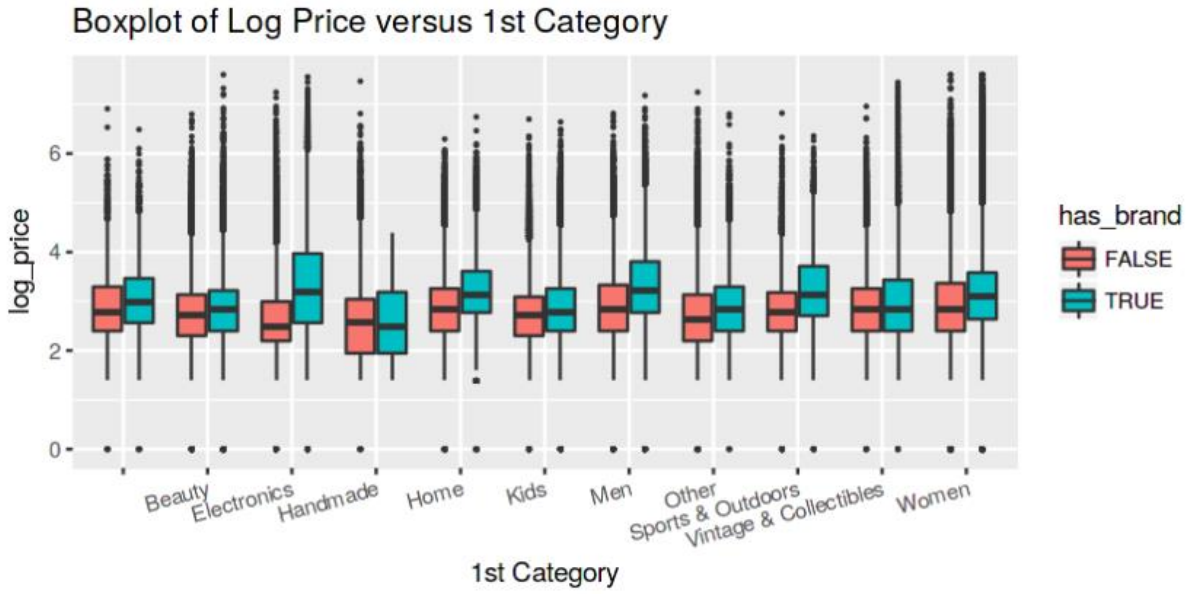



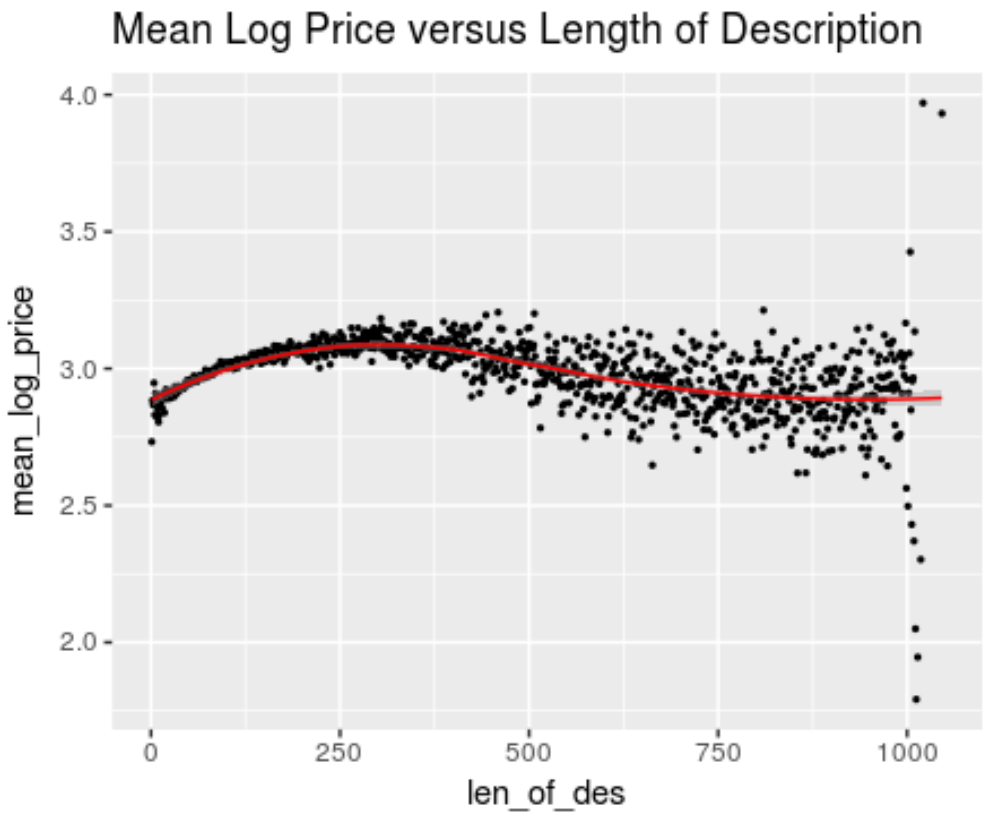
## 5.Boxplot of the entire numerical variable with the dependent variable

The main objective is to predict price.The mean price doesn't vary too much with that of condition and shipping info. We can see that even though we take log to the price, there are still lots of outliers. The item is more expensive If the shipping fee is expected to be paid by the buyer.The price don't differ too much between different categories. However, there is a clear trend that items with brands have higher price than items without brands, especially for the electronics.




Boxplot of Log Price versus 1st Category
log_price
0
2
4
6
Beauty
Electronics
Handmade
Home
Kids
Men
Other
Sports & Outdoors
Vintage & Collectibles
Women
1st Category
has_brand
FALSE
TRUE


Mean Log Price versus Length of Description
mean_log_price
2.0
2.5
3.0
3.5
4.0
0
250
500
750
1000
len_of_des

## 7. Preprocessing of the variable

Character variable are preprocessed before modelling.Various library are used for the preprocessing – quanteda,stringr etc. We can split the category names and store them into 3 column.Cleaning is done on all character variables.

## 8. Building Predictive Models

The data is divided into training and testing set in the proportion of 70:30.The evaluation metric for this prediction is Root Mean Squared Logarithmic Error.

$$\epsilon = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(p_i+1) - \log(a_i+1))^2}$$

Where:

ϵ is the RMSLE value (score)
n is the total number of observations in the (public/private) data set,
pi is your prediction of price, and
ai is the actual sale price for ii.
log(x) is the natural logarithm of x

- Light GBM
  Light GBM is a fast, distributed, high-performance gradient boosting framework based on decision tree algorithm, used for ranking, classification and many other machine learning tasks.
  RMSLE-0.4679472

- XGBoost
  XGBoost is an implementation of gradient boosted decision trees designed for speed and performance.
  RMSLE-0.4347824

THANK YOU