2021-2023

Price Predictions of Second-Hand Products Using Text Analytics & RNN:

MBA-Business Analytics Final Project:

Under the Guidance of - Dr. Pooja Sengupta

Submitted By:

SHREYANSH MOHANTY

ROLL NUMBER: BA033-21

IIM RANCHI, MBA-BA (2021-23)

Contents:

Introduction:

With the growth in E-commerce & ease of accessibility to E-commerce platforms, the speed of exchange of goods between retailers-consumers & between consumers amongst themselves has increased exponentially.

As consumers are spoilt for options with regards to type of products, brands & updated designs, there has been an increase in recent times for the need to discard/resell old or unwanted products as users move on to the next trend in the market. This has also led to a significant rise in the "Sharing-Economy" in which a greater number of consumers are willing to cash-in on their old/used products to fund their new purchases, and also a greater number of people are willing to buy these second-hand products as they are considered a cheaper alternative to the conventional retail market.

The aforementioned growth in E-commerce platforms has also directly contributed to rise in the online second-hand markets, with social media platforms such as Facebook & Instagram being popular destinations to market & exchange used goods. Observing the potential in this second-hand consumer-to-consumer retail market, traditional Ecommerce players such as Amazon have also introduced options for users to sell their used items as refurbished products. Moreover, there are entire E-commerce platforms built around the premise of reselling of goods in the consumer-to-consumer market space such as E-Bay, Mercari etc.

Generally, there is no stipulation on how the price of the second-hand goods needs to be set. This creates some uncertainty amongst the resellers & potential buyers on what the ideal price should be for the product being listed. The ambiguity on the price can be attributed to the following factors:

1. Uncertainty on the depreciation on the product given its condition/period of time it has been used for.
2. Uncertainty on the value of the product's brand. In most cases, the product's brand name itself could be missing due to deterioration in product's appearance.
3. Uncertainty on how the products compares to new offerings in the market, and consequently how much the value should be for older/out-of-date trends.
4. Large number of listings on the second-hand market of the same/similar products of varying qualities with diverse price listings.
5. Impact of the shipping prices in a consumer-to-consumer market space & how it influences the final price of the item being exchanged. Usually the consumer-to-consumer market space does not have economies of scale with respect to logistics, hence the transport of goods is a significant contributor to the final price of an item being sold.

Considering all the above issues, an ideal solution that comes to mind is leveraging past records of second-hand listings that have already been sold to get a rough estimate on the ideal price for a product being put up for resale.

Moreover, another factor that needs to considered is the influence of the item descriptions accompanying a resale listing & how it influences potential buyers.

This is the premise of the current project being undertaken.

Project Summary:

Objective: Predicting the price of product listing in a second-hand consumer-to-consumer market by considering the textual item descriptions accompanying the listing.

Dataset & Source: Mercari E-commerce platform consumer-to-consumer sales data.

- Link - https://www.kaggle.com/c/mercari-price-suggestion-challenge
- https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiA7dqZ6K_9AhXp1TgGHRDEDXoQFnoECA0QAQ&url=https%3A%2F%2Fuploads-ssl.webflow.com%2F61ee41ad88001d7eae3b2752%2F624652e5ed0594320b9a8dba_Mercari%2520Reuse%2520Report%25202022.pdf&usg=AOvVaw3paAeWfGWZ04B9VFkFsqzK

Methodology:

1. Exploratory Data Analysis on the Dataset to get significant associations between features.
2. Pre-processing of textual data in the Item Descriptions to find significant themes/topics being covered in the product listings (LDA – Topic Modelling & TF-IDF)
3. Clustering algorithms (KNN) to identify similar groups of products being listed.
4. Neural Networks models (RNN) to handle a combination of numerical, categorical & textual features.
5. Root Mean Square Logarithmic Error (RMSLE) function to measure the model performance on a test dataset.

Constraints:

The following constraints/risks have been considered before working on the datasets:

1. Non-standard text descriptions: Second-hand sellers usually are not professionals in retail, hence unlike standard merchants/retailers the textual description of the items being listed are usually sub-standard & low effort. As the dataset cannot be filtered to include only item listings based on a arbitrary standard such as quality of the descriptions, accuracy of the overall model could be impacted.
2. User defined grading on item quality: The dataset in consideration has the re-sellers defining the product's current quality on a scale of 1-5. As such the legitimacy of the rating being provided cannot be verified.
3. Limitations of regression models: Due to the nature of the dataset containing a mixture of numerical, categorical & textual data, conventional regression models (Linear Regression) is not an appropriate model which can handle the problem in consideration. Hence, Neural Networks have been used to identify patterns/sequences present in the textual information & clusters of similar product listings to arrive at an acceptable estimate of prices.

Contributions:

The major contributions aimed at being made through this project are as follows:

1. Developing an intelligent price suggestion system to remove some of the ambiguity resellers & customers have with regards to the price of a second-hand product.
2. Customised metric to measure the accuracy of the price prediction model.

Dataset Description:

We are considering an exhaustive dataset containing the sales information of second-hand item listing from the Mercari E-commerce platform.

A snapshot of the dataset is given below:

	train_id	name	item_condition_id	category_name	brand_name	price	shipping	item_description
0	0	MLB Cincinnati Reds T Shirt Size XL	3	Men/Tops/T-shirts	NaN	10.0	1	No description yet
1	1	Razer BlackWidow Chroma Keyboard	3	Electronics/Computers & Tablets/Components & P...	Razer	52.0	0	This keyboard is in great condition and works ...
2	2	AVA-VIV Blouse	1	Women/Tops & Blouses/Blouse	Target	10.0	1	Adorable top with a hint of lace and a key hol...
3	3	Leather Horse Statues	1	Home/Home Décor/Home Décor Accents	NaN	35.0	1	New with tags. Leather horses. Retail for [rm]...
4	4	24K GOLD plated rose	1	Women/Jewelry/Necklaces	NaN	44.0	0	Complete with certificate of authenticity
5	5	Bundled items requested for Ruie	3	Women/Other/Other	NaN	59.0	0	Banana republic bottoms, Candies skirt with ma...
6	6	Acacia pacific tides santorini top	3	Women/Swimwear/Two-Piece	Acacia Swimwear	64.0	0	Size small but straps slightly shortened to fi...
7	7	Girls cheer and tumbling bundle of 7	3	Sports & Outdoors/Apparel/Girls	Soffe	6.0	1	You get three pairs of Sophie cheer shorts siz...
8	8	Girls Nike Pro shorts	3	Sports & Outdoors/Apparel/Girls	Nike	19.0	0	Girls Size small Plus green. Three shorts total.
9	9	Porcelain clown doll checker pants VTG	3	Vintage & Collectibles/Collectibles/Doll	NaN	8.0	0	I realized his pants are on backwards after th...
10	10	Smashbox primer	2	Beauty/Makeup/Face	Smashbox	8.0	1	0.25 oz Full size is 1oz for [rm] in Sephora
11	11	New vs pi k body mists	1	Beauty/Fragrance/Women	Victoria's Secret	34.0	0	(5) new vs pink body mists (2.5 oz each) Fresh...
12	12	Black Skater dress	2	Women/Dresses/Above Knee, Mini	rue	16.0	0	XI, great condition
13	13	Sharpener and eraser	1	Other/Office supplies/School Supplies	Scholastic	4.0	1	No description yet
14	14	HOLD for Dogs2016 Minnetonka boots	3	Women/Shoes/Boots	UGG Australia	43.0	0	Authentic. Suede fringe boots. Great condition...

Description of the features being extracted:

1. train_id: The ID of the item listing.
2. Name: The title of the listing i.e., the headers.
3. item_condition_id: Seller declared condition of the product listing, graded on a scale of 1-5.
4. category_name: Item categories the product listing separated by "/". A maximum of 3 categories can be assigned to any product on the Mercari reseller platform.
5. brand_name: The brand the product listing belongs to.
6. Price: The price that the item was sold for. This is also the target variable which our model will be predicting.
7. Shipping: 1 if shipping fee is paid by seller and 0 by buyer. Actual values of the shipping prices cannot be extracted as logistics is not covered by the E-commerce platform & has to be arranged by the seller/buyer. This was a limitation that has been considered during modelling.
8. item_description: The full description of the item provided by the seller during product listing. This is a major feature being considered for predicting prices as potential buyers consider the listings item description as a deciding factor in their purchase decision.

Other features that are present in the E-commerce platform along with the product listings that users consider (but not taken into consideration for model preparation):

1. Image: The item's image that accompanies a product listing. Potential buyers refer to the image to verify whether the sellers' self-declared item condition rating is accurate. To avoid the model getting complicated by including computer vision, item_condition_id is taken as a surrogate to this feature. (Constraint has been mentioned earlier)
2. Comments: Resale listings usually have a comments thread in which other potential buyers have listed their price offerings in case the seller has put up the price of the item to be negotiable. Actual Price at which the item was sold is considered as a surrogate to this to avoid complicating the model. (Though this could be considered in future models)

Exploratory Data Analysis:

A brief overview of the features & variable types:

1. Numerical features:
    a. Price -> Target Variable
2. Categorical features:
    a. shipping cost -> 1 if shipping fee is paid by retailer and 0 if paid by customer
    b. item_condition_id -> The condition of the items provided by the retailer
    c. name -> Item Name
    d. brand_name -> Producer brand name
    e. category_name -> Item single or multiple categories that are separated by "\"
    f. item_description -> A short description of the item

```
# Different data types in the dataset: categorical (strings) and numeric
train.dtypes
```

```
train_id              int64
name                 object
item_condition_id      int64
category_name        object
brand_name           object
price               float64
shipping              int64
item_description     object
dtype: object
```

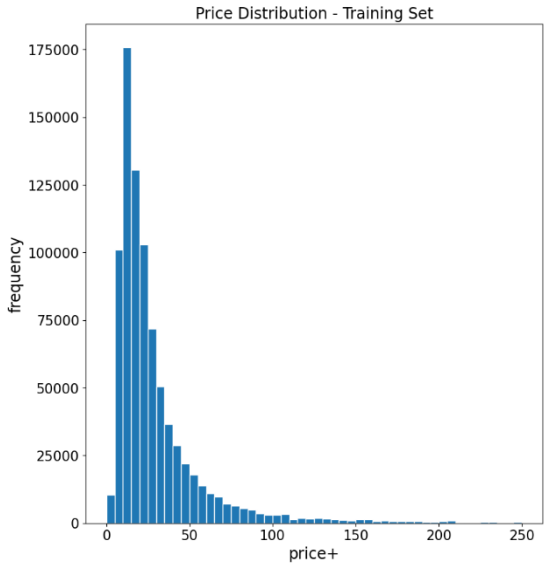
Review of the Target Variable -> Price:

Checking the descriptive statistics of the Price variable in the training dataset we observe:

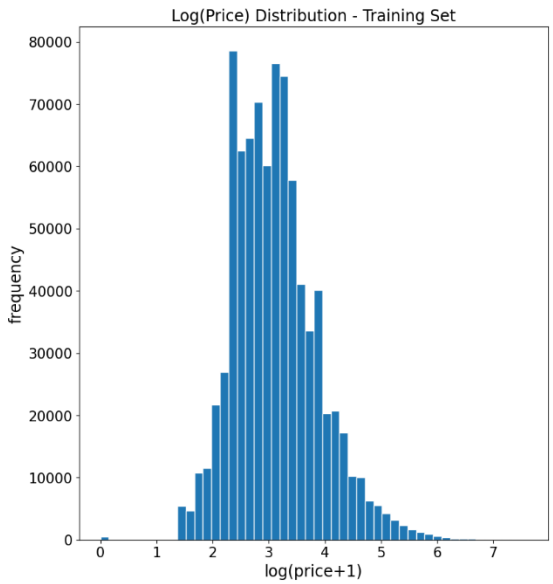
- Mean price in training dataset: 30.92
- Median price in training dataset: 20.00
- Max price in training dataset: 2009.00
- Min price in training dataset: 0.00

(All prices in $s)

Checking the price distribution on the training dataset (via Histogram) we observe:



- From the graph we can observe that the Prices are distributed more or less normally but they are skewed to the right.
- The range of variations in the prices is significant.
- Hence a log() transform of the price variable was considered to get a more symmetric normal distribution towards the centre.



From the log() transformed Price variable we observe:

- Prices are more or less normally distributed in the training dataset.
- The range of variations in the price is now somewhat reduced.
- The Price distribution is now less skewed as compared to the earlier distribution.

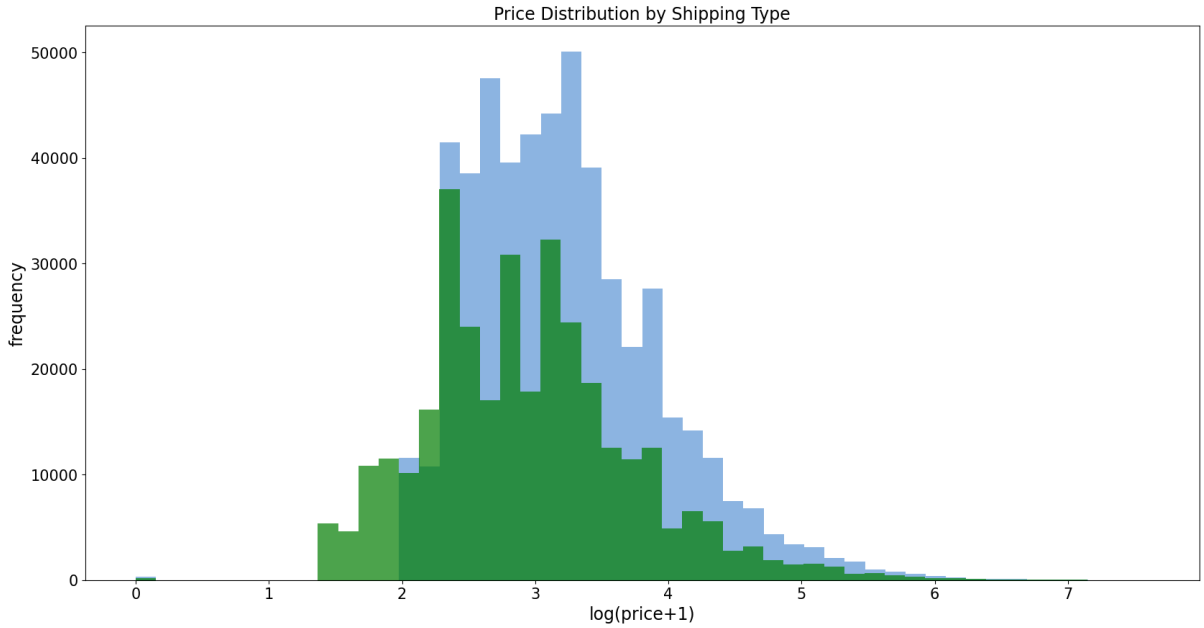
Review of the Shipping Variable:

Since shipping/logistics is not covered by most Ecommerce platforms when it comes to the second-hand reselling market space, it is imperative that we consider who bears the shipping cost (buyer or seller) and how it impacts the overall selling price of the product listing.

Analysing the distribution of shipping variable in the training dataset:

- 0 (Shipping cost paid by the Seller): 0.610865
- 1 (Shipping cost paid by the Buyer): 0.389135

Plotting the Price distributions by Shipping (0/1):



- Blue plots -> Shipping paid by Seller (0)
- Green plots -> Shipping paid by Buyer (1)

Analysing the descriptive statistics of the Shipping categories (0/1):

- Mean price in training dataset for shipping paid by retailers (0): 33.34
- Median price in training dataset for shipping paid by retailers (0): 21.00
- Mean price in training dataset for shipping paid by customers (1): 27.12
- Median price in training dataset for shipping paid by customers (1): 16.00

Conclusions from the plots & descriptive stats summary above:

1. Nearly 39% of the customers/buyers have to bear the shipping expenses themselves.
2. Mean price when the customer bears the shipping expense < Mean price when retailer bears the shipping expense.
3. Median price when the customer bears the shipping expense < Median price when retailer bears the shipping expense.
4. Retailers have to provide customers incentives such as lower prices when the customer bears the shipping expenses.

Review of Item Categories:

As mentioned earlier, the Ecommerce platform allowed for up to 3 categories to be assigned to an individual item in any product listing. The dataset contained the Categories separated by "/". To make the processing of the item categories feasible, it was first necessary to delimit the categories column & separate them into 3 different subcategory columns.

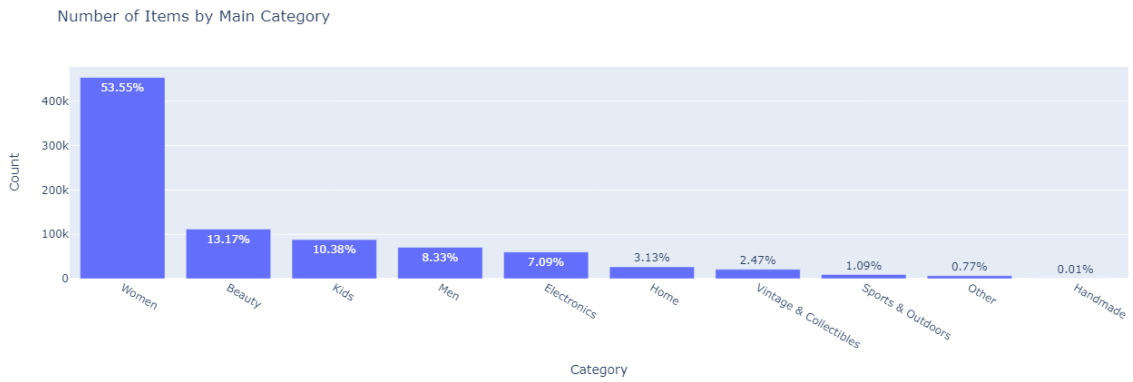
```
# Function to split products with multiple categories...
def split_cat(text):
    try:
        return text.split("/")
    except:
        return ("No Label", "No Label", "No Label")


# Splitting the product categories in the training dataset using the above function...
# Categories split into --> General Cateogry || Sub category 1 || Sub cateogry 2
# Results appended into the dataframe...

train['general_cat'], train['subcat_1'], train['subcat_2'] = \
zip(*train['category_name'].apply(lambda x: split_cat(x)))
train.head()
```

	train_id	name	item_condition_id	category_name	brand_name	price	shipping	item_description	general_cat	subcat_1	subcat_2
1	1	Razer BlackWidow Chroma Keyboard	3	Electronics/Computers & Tablets/Components & P...	Razer	52.0	0	This keyboard is in great condition and works ...	Electronics	Computers & Tablets	Components & Parts
2	2	AVA-VIV Blouse	1	Women/Tops & Blouses/Blouse	Target	10.0	1	Adorable top with a hint of lace and a key hol...	Women	Tops & Blouses	Blouse
6	6	Acacia pacific tides santorini top	3	Women/Swimwear/Two-Piece	Acacia Swimwear	64.0	0	Size small but straps slightly shortened to fi...	Women	Swimwear	Two-Piece
7	7	Girls cheer and tumbling bundle of 7	3	Sports & Outdoors/Apparel/Girls	Soffe	6.0	1	You get three pairs of Sophie cheer shorts siz...	Sports & Outdoors	Apparel	Girls
8	8	Girls Nike Pro shorts	3	Sports & Outdoors/Apparel/Girls	Nike	19.0	0	Girls Size small Plus green. Three shorts total.	Sports & Outdoors	Apparel	Girls

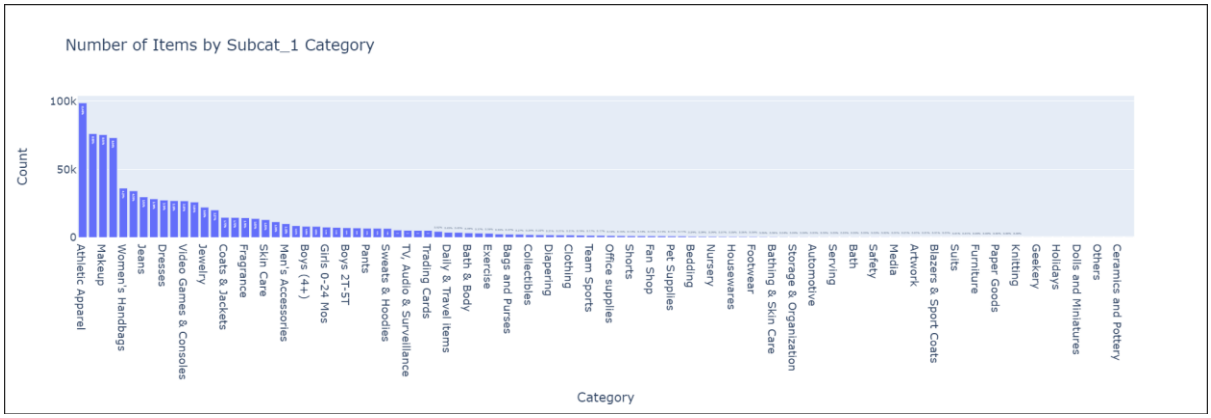
Checking for the number of unique values in each category/subcategory:

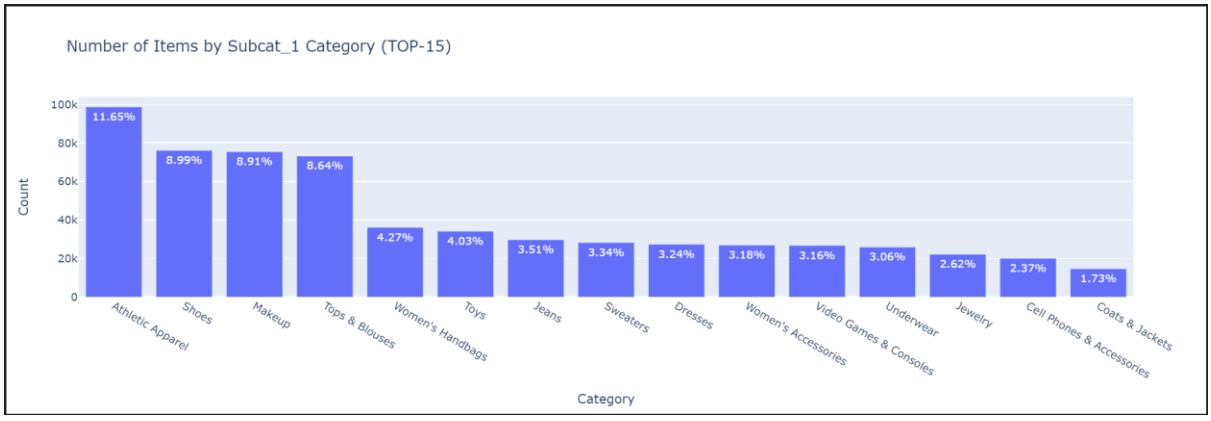
- Number of unique General Category divisions: 10
- Number of unique SubCategory 1 divisions: 104
- Number of unique SubCategory 2 divisions: 669

Plotting the distribution of items in the training dataset by General Category (n = 10):

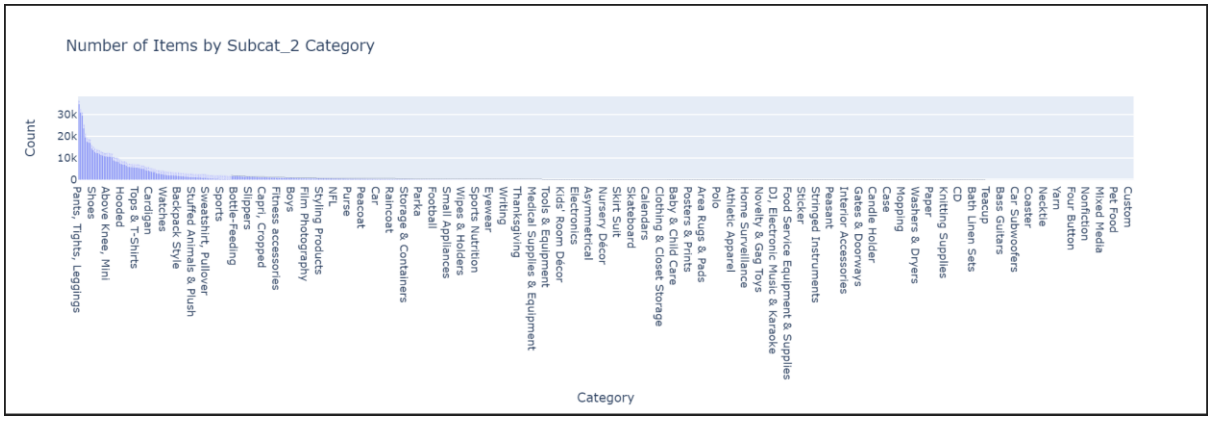


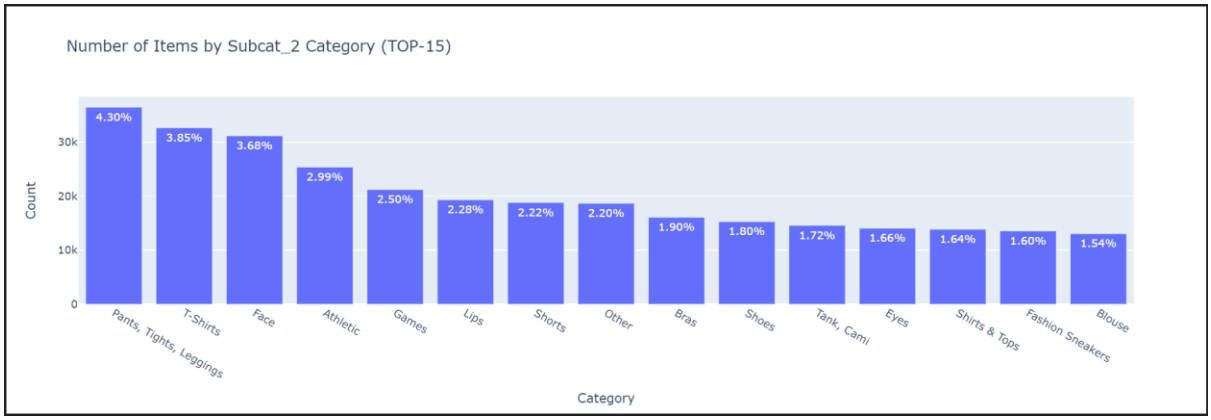
Similarly plotting the distribution of items in SubCategory 1 (n = 104) & checking the top-15 categories:





Similarly plotting the distribution of items in SubCategory 2 (n = 669) & checking the top-15 categories:





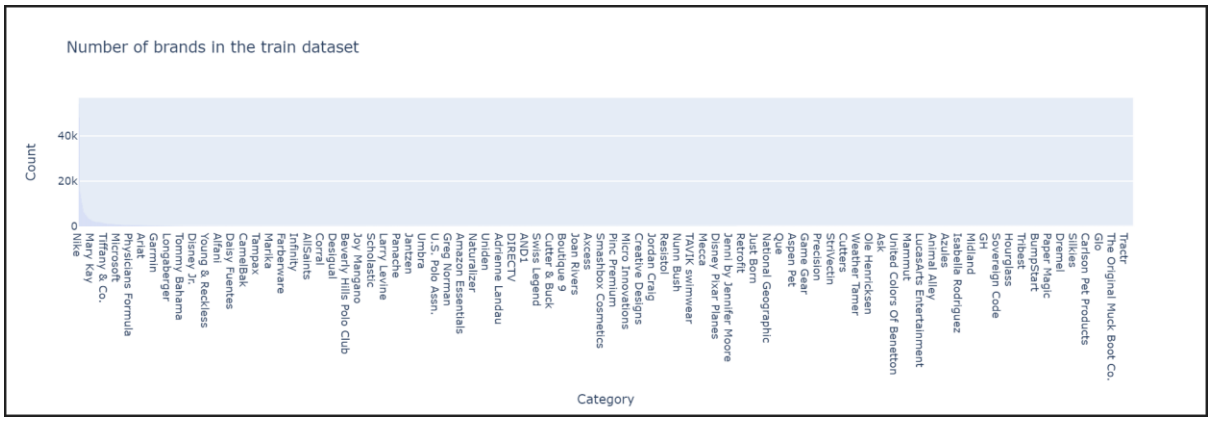
These would further be useful while analysing the TF-IDF of item descriptions & while clustering.

Review of Brands:

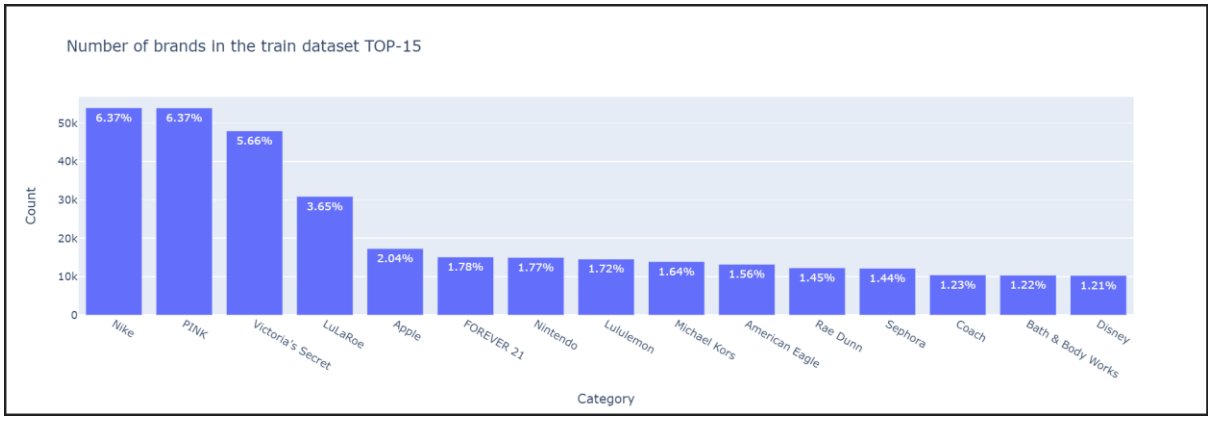
Most of the items being sold in the second-hand market space is expected to be of some brand. Hence analysing the unique brand names present in the training dataset could give some useful insights.

Number of unique brands present = 4791

Plotting the distribution of unique product listing across all brands:



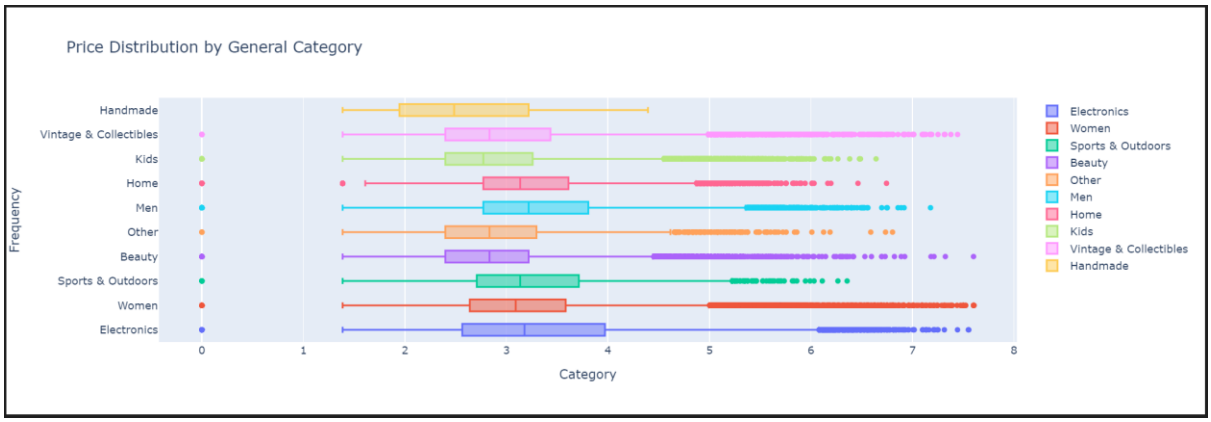
Plotting the top-15 brand's whose products are being listed:



Brand image can sometimes act as a contributing factor to a buyer's perception of item quality. Hence it is important to take into consideration the brand of an item in any product listing.

Price Plots per General Category:

Analysing the distribution of prices per unique product listing in the General Categories (n = 10):



(Interactive plot present in Jupyter Notebook)

Text Pre-Processing & EDA:

Steps involved in Text Pre-Processing of the Item Descriptions feature are as follows:

1. Text Cleaning
2. Tokenisation
3. Wordclouds
4. TF-IDF
5. Topic Modelling – LDA

Text Cleaning & Tokenisation:

Most of the item descriptions contain special characters, emojis etc. In order to effectively work on the textual data, it is important to first clean it.

Steps followed for cleaning the text data & making it ready for analysis are as follows:

1. Removing special characters
2. Removing punctuation
3. Removing regular expressions
4. Converting all characters to lower case

After data is cleaned, commonly occurring words (Stopwords) are also removed to increase the speed of processing & remove redundant words that don't add any value to the model.

Post cleaning, the item description text can be tokenised into its constituent words.

Function to clean the text & tokenise it:

```python
# Defining a function which converts the item description texts to tokens...
# sent_tokenise() --> Tokenise text into sentences
# word_tokenise() --> Tokenise sentences into words

# re library is used for regular expressions...

def tokenize(text):
    try:
        regex = re.compile('[' +re.escape(string.punctuation) + '0-9\\r\\t\\n]') # defining a variable with set of regular expressions & punctuations
        text = regex.sub(" ", text) # remove punctuation

        tokens_ = [word_tokenize(s) for s in sent_tokenize(text)] # Tokenising words per sentence after item_description has been tokenised into sentences
        tokens = []
        for token_by_sent in tokens_:
            tokens += token_by_sent
        tokens = list(filter(lambda t: t.lower() not in stop, tokens)) # Removing STOP words from the tokensied descriptions/tokens list & converting into lower case
        filtered_tokens = [w for w in tokens if re.search('[a-zA-Z]', w)]
        filtered_tokens = [w.lower() for w in filtered_tokens if len(w)>=3]

        return filtered_tokens

    except TypeError as e: print(text,e) # To handle cases with No item descriptions...
```

Python

	train_id	name	item_condition_id	category_name	brand_name	price	shipping	item_description	general_cat	subcat_1	subcat_2	tokens
0	1	Razer BlackWidow Chroma Keyboard	3	Electronics/Computers & Tablets/Components & P...	Razer	52.0	0	This keyboard is in great condition and works ...	Electronics	Computers & Tablets	Components & Parts	[keyboard, great, condition, works, like, came...
1	2	AVA-VIV Blouse	1	Women/Tops & Blouses/Blouse	Target	10.0	1	Adorable top with a hint of lace and a key hol...	Women	Tops & Blouses	Blouse	[adorable, top, hint, lace, key, hole, back, p...
2	6	Acacia pacific tides santorini top	3	Women/Swimwear/Two-Piece	Acacia Swimwear	64.0	0	Size small but straps slightly shortened to fi...	Women	Swimwear	Two-Piece	[size, small, straps, slightly, shortened, fit...
3	7	Girls cheer and tumbling bundle of 7	3	Sports & Outdoors/Apparel/Girls	Soffe	6.0	1	You get three pairs of Sophie cheer shorts siz...	Sports & Outdoors	Apparel	Girls	[get, three, pairs, sophie, cheer, shorts, siz...
4	8	Girls Nike Pro shorts	3	Sports & Outdoors/Apparel/Girls	Nike	19.0	0	Girls Size small Plus green. Three shorts total.	Sports & Outdoors	Apparel	Girls	[girls, size, small, plus, green, three, short...

TF-IDF:

- Term Frequency (TF): Measures how often a particular word occurs in a document (in this case, an individual item description)
  - $TF = \frac{n(Word)}{N(Total\ Words\ in\ Document)}$
- Inverse Document Frequency (IDF): Measure how unique the term is over the entire set of all documents (in this case, all item descriptions in the training dataset)
  - $IDF = \log\frac{D(Total\ number\ of\ Documents)}{1+d(Number\ of\ Documents\ the\ Word\ occurs\ in)}$
- Higher TF-IDF score indicates that the word is unique, and hence can be considered to be a significant indicator as to what topic is being discussed in a particular document (item description)

List of the Top-10 words per TF-IDF scores:

	tfidf
fast heat	12.635476
discounts offers	12.635476
flynn skye	12.635476
stripped polo	12.635476
trivet	12.635476
disinfecting wipes	12.635476
pretreated	12.635476
gold shift	12.635476
topics	12.635476
gifts items	12.635476

List of Bottom-10 words per TF-IDF scores:

	tfidf
size	2.210096
new	2.213893
condition	2.585938
brand	2.852017
worn	2.930135
brand new	2.968557
free	3.005863
used	3.129891
shipping	3.244118
never	3.258112

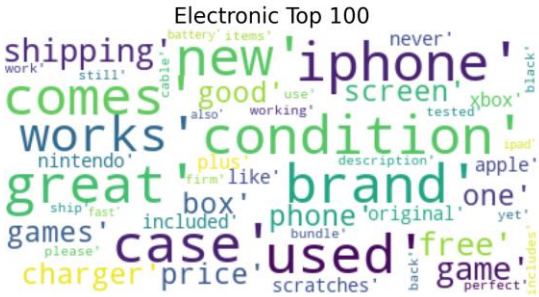
Comparing the Top-10 & Bottom-10 words, the following can be concluded:

1. Checking the Bottom-10 TF-IDF score words: Difficult to predict which categories the item description might belong to. More or less generic words used when it comes to providing Item descriptions.
2. Checking the Top-10 TF-IDF score words: Get a rough idea about the product being listed. Some words/phrases are descriptive as far as Items are concerned.

Word Cloud:

A visual representation of the most frequently occurring words in a corpus of text.

In this case we prepared a word cloud for the top-4 general categories, which has been displayed below:



Topic Modelling:

Topic modelling can help in generating a user specified number of groups. The documents (Item Descriptions) can then be divided into these groups.

Topic modelling can give a better understanding of the general themes/patterns emerging from the item descriptions of the product listings that are being analysed. This can further be used to Cluster similar product listings which could then be useful in creating a robust model for price predictions.

Preparing a list of topics taking n = 10 (given that we have 10 general item categories) & listing the top-20 words occurring in each, we observe:

```
Topic 0: description | yet | description yet | navy | couple | blue | american | times | wash | couple times | green | boots | disney | forever | eagle | american eagle | skinny | tried | old | fast
Topic 1: excellent | used | condition | excellent condition | inside | stains | pocket | clean | gently | gently used | rips | photo | bag | air | tears | zipper | come | lots | use | weight
Topic 2: box | used | comes | great | case | condition | included | original | iphone | works | one | bag | scratches | new | silver | authentic | game | charger | gold | wallet
Topic 3: size | worn | condition | like | new | great | women | perfect | fit | like new | dress | men | flaws | black | color | light | one | shorts | long | jeans
Topic 4: condition | size | good | small | medium | good condition | great | wear | worn | top | great condition | size medium | white | size small | back | little | black | bottom | feel | fit
Topic 5: new | brand | brand new | never | never worn | tags | used | never used | new never | box | new tags | worn | size | set | body | new box | listings | without | opened | one
Topic 6: free | shipping | price | home | free shipping | smoke | firm | price firm | free home | smoke free | new | authentic | pet | retail | color | pet free | full | comes | brush | shade
Topic 7: shipping | please | bundle | items | free | ship | save | price | ask | questions | day | item | check | bundle save | purchase | make | free shipping | thank | save shipping | get
Topic 8: pink | size | black | secret | victoria | victoria secret | new | cute | color | nwt | large | leather | worn | super | tags | super cute | zip | leggings | secret pink | times
Topic 9: material | soft | brown | find | right | perfect | super | washed | hard | piece | beautiful | made | hair | little | big | natural | look | dog | long | hard find
```

- Most of the topics are related to clothing.
- Shipping is a crucial term occurring in most of the item descriptions (per topic).
- Topics related to clothing can be differentiated by the Sub Category of the type of clothing item being listed. Major Sub Categories that can be identified through the Topics are Men's, Women's, Children's etc.
- Pet items are also a prominent topic being mapped.

Preparing a list of topics taking n = 40 (to cover a more diverse range of listings) & listing the top-20 words occurring in each topic, we observe:

```
Topic 0: condition | box | perfect | excellent | used | excellent condition | times | new box | perfect condition | worn | couple | still | couple times | size | super |
well | skinny | soft | little | material
Topic 1: gold | silver | orange | stretch | crop | dog | womens | beautiful | earrings | tone | metal | neon | cardigan | cold | scott | kendra | authentic | blouse |
welcome | crop top
Topic 2: box | comes | original | works | included | great | used | charger | scratches | card | original box | water | deal | come | work | ipad | watch | perfectly | xbox
| cable
Topic 3: cute | super | size | women | flaws | super cute | light | full | dress | bought | twice | worn | also | high | full size | gold | skirt | shoes | rose | worn
twice
Topic 4: price | firm | price firm | bag | feel | strap | make | feel free | inside | edition | purse | adjustable | limited | back | shoulder | offer | questions | two |
around | straps
Topic 5: purchased | looking | pocket | nice | wallet | dunn | rae | rae dunn | michael | kors | gorgeous | pack | michael kors | care | authentic | final | leather |
lining | thanks looking | imperfections
Topic 6: online | lowest | dust | bag | bracelet | separate | new size | double | dust bag | ring | necklace | size | shop | custom | designed | lowest price | cups |
pandora | charm | diamond
Topic 7: pink | secret | victoria | victoria secret | secret pink | tag | logo | new | new tag | brown | sizes | interior | sale | hot | polo | medium | lauren | retail |
ralph | ralph lauren
Topic 8: good | condition | size | good condition | worn | shirt | large | pink | size large | used | times | lularoe | sleeve | print | worn times | see | black | used
condition | wear | minor
Topic 9: nwt | big | leggings | find | right | black | months | hard | lace | adidas | size | jordan | much | hard find | need | classic | run | shoe | per | colored
Topic 10: new | used | authentic | skin | color | shade | brush | makeup | matte | lip | sealed | palette | colors | light | eye | eyeshadow | swatched | foundation |
brandy | lipstick
Topic 11: black | leather | retails | authentic | mini | wore | use | mascara | details | tax | let | let know | real | heel | guaranteed | sorry | made | zip | sandals |
plus
Topic 12: description | yet | description yet | case | iphone | game | phone | pop | photo | plus | one | everything | size worn | unopened | working | play | funko | clear
| last | screen
Topic 13: check | items | get | listings | listing | buy | ship | check listings | air | pictured | available | weight | days | purchase | inch | want | within | closet |
pre | total
Topic 14: great | condition | great condition | stains | holes | size | hoodie | rips | worn | used | tears | flaws | zip | one | washed | band | rips stains | face |
pullover | half
Topic 15: free | shipping | bundle | free shipping | please | price | home | smoke | free home | smoke free | firm | save | item | ask | ship | new | pet | items | bundle
save | price firm
Topic 16: color | grey | left | blue | clean | baby | look | washed | looks | trades | navy | product | tried | like | one | negotiable | life | dark | cover | last
Topic 17: shorts | american | size | jeans | waist | great | pretty | shape | old | forever | eagle | style | american eagle | cute | hollister | girl | navy | old navy |
people | outfitters
Topic 18: size | small | medium | black | fit | top | white | size small | back | wear | size medium | color | front | worn | bra | pockets | large | bottom | length |
cotton
Topic 19: new | brand | brand new | never | tags | size | like | worn | never worn | new tags | like new | never used | new never | used | men | fits | body | nike | set |
jacket
```

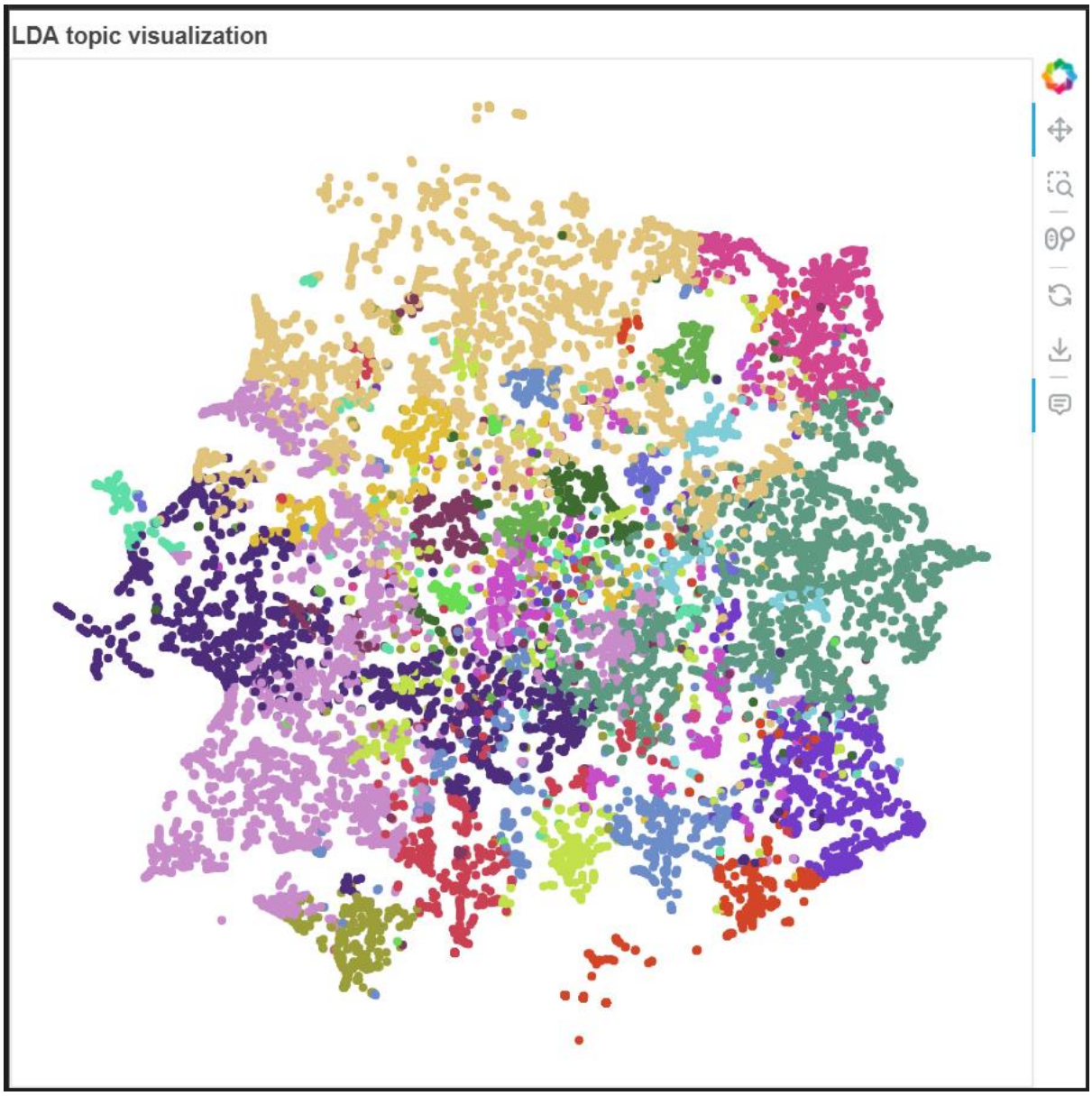
- A more exhaustive list of categories is now covered in the topics.
- Watches & accessories are now assigned topics.

Moving ahead with n = 20 for number of topics, assigning the topics to the item descriptions present in the training dataset:

	x	y	description	category	topic
0	10.637218	-24.829309	Super soft and in perfect condition. A little ...	Women	0
1	5.995852	17.090754	Pink Nike sneakers size 4. Like new never worn...	Kids	4
2	35.391930	3.771287	Lego Disney Princess Cogsworth and Lumiere fre...	Kids	15
3	5.136758	9.705489	Little black stretchy strapless mini dress wit...	Women	11
4	23.749315	-11.924470	Michael Kors watch and Diesel watch Free shipp...	Women	3
...	...	...	...	...	...
14995	23.079021	-5.879655	This is a New-in-Box Otterbox Defender Case fo...	Electronics	15
14996	-5.941325	-22.362772	1/4 zip fleece Black Size M	Women	14
14997	13.499397	22.102354	Experienced shipper.	Home	5
14998	26.953966	11.934688	New & full. No box. The shade is Golden. All i...	Beauty	15
14999	5.148448	9.500727	White patent leather size 18mm	Women	11

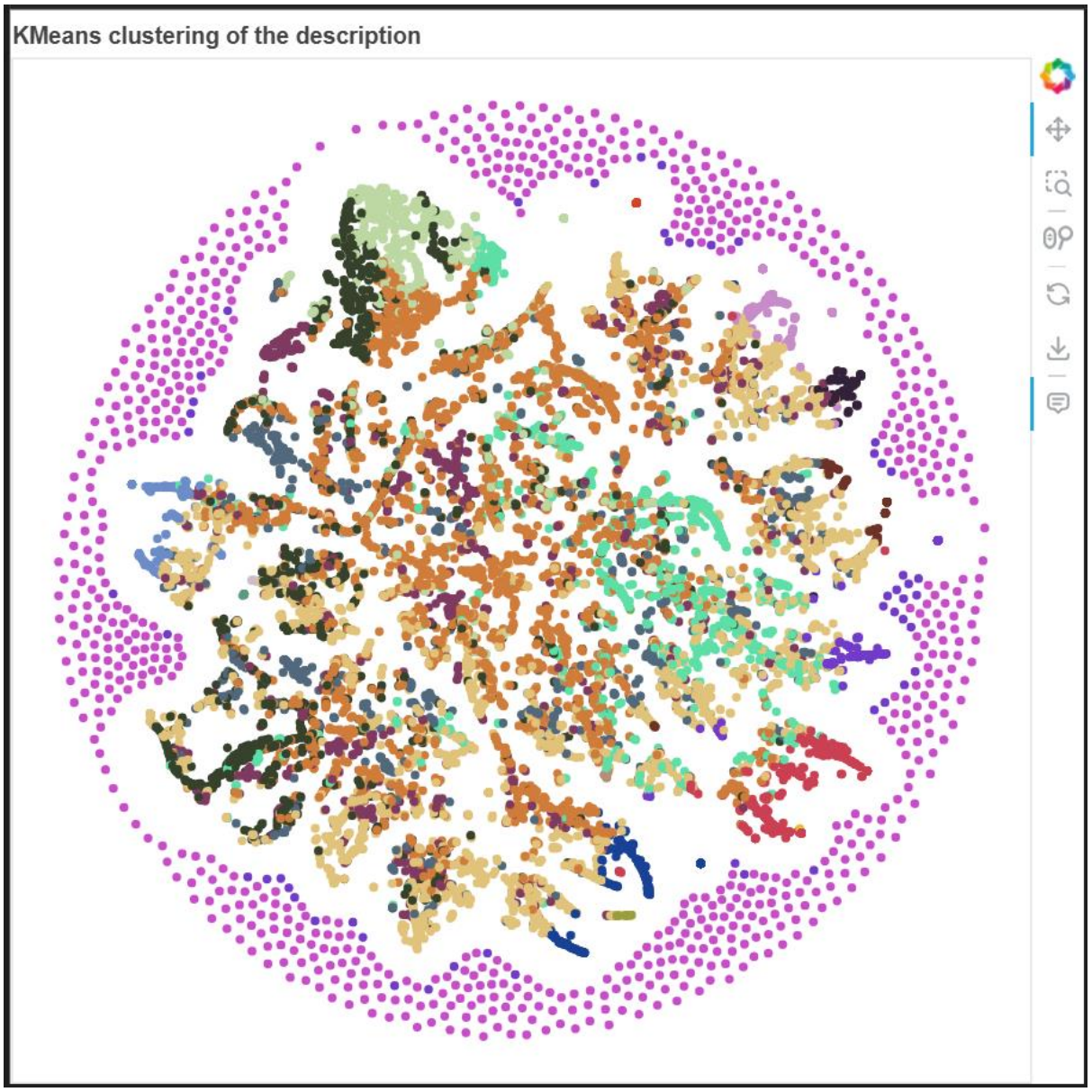
15000 rows × 5 columns

Preparing a K-Means cluster for the topics:



(Interactive plot present in Jupyter Notebook)

K-Means clusters of Topics taking into account Categories:



(Interactive plot present in Jupyter Notebook)

Note:

- LDA algorithm used for Topic Modelling.
- tSNE dimensionality reduction used on the training dataset before K-Means clustering performed.

Neural Networks Modelling:

We have considered RNN (Recurrent Neural Networks) for the price prediction model due to its superior capability in handling textual data.

Steps involved in modelling are listed as follows:

1. Data cleaning -> Performed as part of the EDA & Text cleaning portions.
2. Filling in missing values.
3. Feature Engineering -> Adding features, encoding categorical data & converting textual data (Item descriptions) into sequences to fit into the RNN model.
4. Splitting the entire dataset into Test/Train data.
5. Defining the parameters/layers of the RNN model.
6. Model fitting.
7. Declaring the RMSLE error function.
8. Model evaluation.

Feature Engineering:

- Assigned Topics to each item description via Topic Modelling as discussed in the previous sections.
- Added Description length & Name length features that count the total number of words present in the Item Description & the Product Listing's heading (before the removal of stop words & tokenisation).
- Encoded the categorical features:
    - Category_name: Divided this into subcat_0, subcat_1 Y subcat_2 and encoded each.
    - Brand_name
- Converted the item descriptions (post cleaning & text pre-processing) into numerical sequences.
- Converted Price into its log() transformed values & saved them into a "Target" feature.

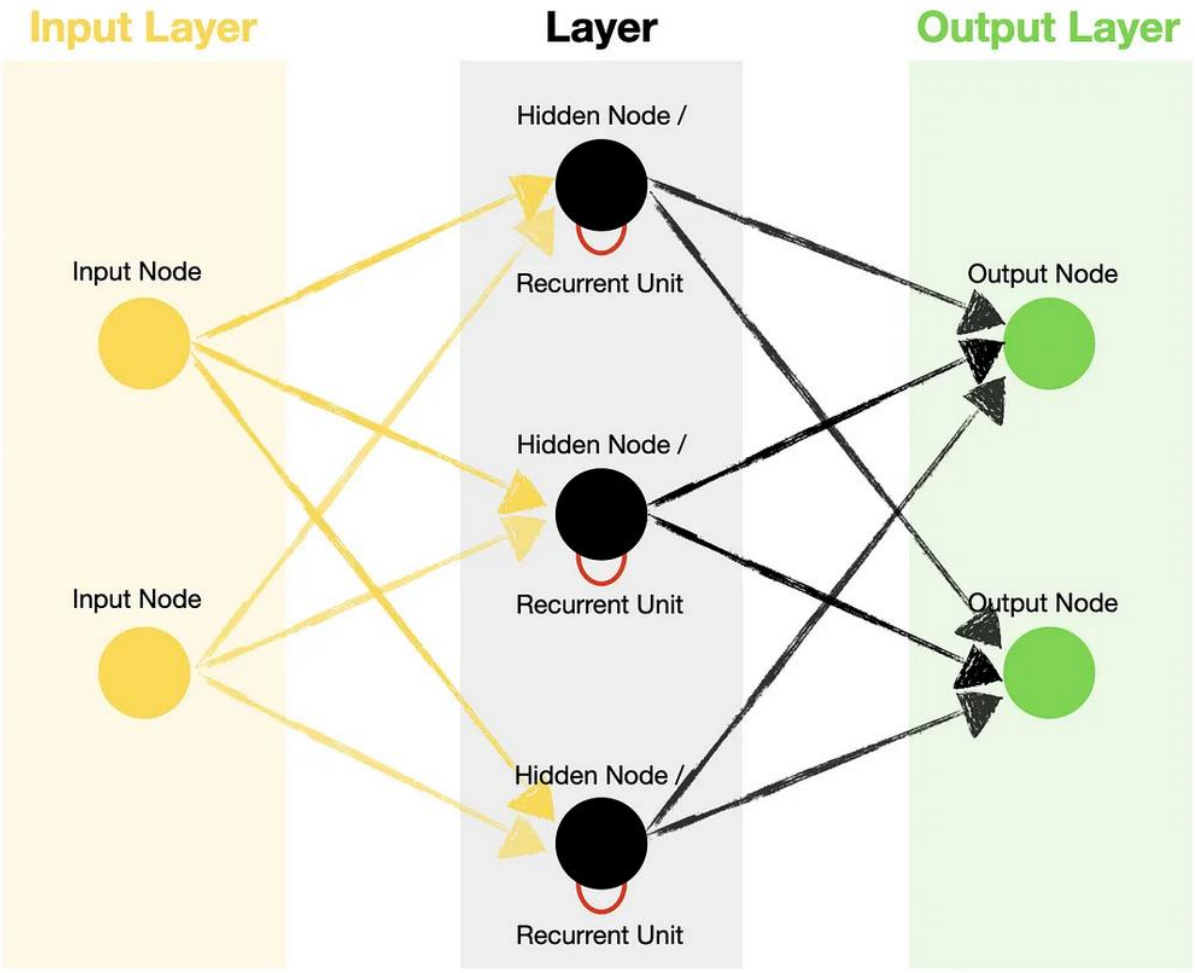
The dataset post cleaning has been displayed as follows:

	train_id	name	item_condition_id	category_name	brand_name	price	shipping	item_description	desc_len	name_len	subcat_0	subcat_1	subcat_2	target	category	seq_item_description	seq_name
295549	295549	Ava Anderson	1	Women/Jeans/Straight Leg	101042	6	1	Ava Anderson	2	2	10	57	726	1.94591	1150	[7759, 8321]	[7759, 8321]
297540	297540	Kendra Scott Jackie Necklace	2	Women/Jewelry/Necklaces	44145	79	0	Kendra Scott Necklace / Gunmetal/ NWOT	6	4	10	58	530	4.382027	1154	[959, 900, 273, 4040, 307]	[959, 900, 10591, 273]
583443	583443	FreeShip Striped Tee Brandy Style	2	Women/Tops & Blouses/T-Shirts	16959	6	1	Super soft Brandy Melville like material Maroo...	16	5	10	104	749	1.94591	1233	[89, 138, 596, 650, 42, 208, 905, 53, 1184, 27...	[1299, 802, 262, 596, 172]
978262	978262	Alo yoga vitality leggings	1	Women/Athletic Apparel/Pants, Tights, Leggings	7725	64	0	Size small. Never worn, new tags Retails [rm]	8	4	10	5	569	4.174387	1110	[3, 36, 27, 19, 2, 54, 376, 9]	[6278, 652, 6687, 33]
727950	727950	Girls 12/24 Mo John Deere Socks Baby 4	1	Kids/Girls 0-24 Mos/Accessories	101042	8	1	New 4 pairs John Deere socks Size 12/24 months...	14	8	4	46	5	2.197225	603	[2, 28, 381, 2421, 6715, 457, 3, 146, 113...]	[78, 146, 113, 3301, 2421, 6715, 457, 153, 28]

RNN Model Fitting:

The advantage of using Recurrent Neural Networks is that, in their architecture, they contain a recurrent node attached to the nodes in the hidden layers which keeps a track of previous inputs that have passed through it. This is specifically helpful when we are trying to identify patterns/sequences present in the dataset. This makes RNN models very adept in dealing with time series forecasting, predictions & textual data.

A simplified architecture of the RNN model is displayed below:



Types of RNN:

1. Simple RNN
2. LSTM (Long Short Term Memory) RNN
3. GRU (Gated Recurrent Unit) RNN

We are considering GRU RNN to model the price prediction model as they are faster compared to the other RNN types.

For the purpose of our model, we want to consider each of the features as an input node, hence converting the pandas dataframe into a dictionary with key values being the individual features considered in the model:

```python
# Converting the datasets from a Pandas dataframe to a dictionary to train the RNN models...

def get_rnn_data(dataset):
    X = {
        'name': pad_sequences(dataset.seq_name, maxlen=MAX_NAME_SEQ),
        'item_desc': pad_sequences(dataset.seq_item_description, maxlen=MAX_ITEM_DESC_SEQ),
        'brand_name': np.array(dataset.brand_name),
        'category': np.array(dataset.category),
        #         'category_name': pad_sequences(dataset.seq_category, maxlen=MAX_CATEGORY_SEQ),
        'item_condition': np.array(dataset.item_condition_id),
        'num_vars': np.array(dataset[["shipping"]]),
        'desc_len': np.array(dataset[["desc_len"]]),
        'name_len': np.array(dataset[["name_len"]]),
        'subcat_0': np.array(dataset.subcat_0),
        'subcat_1': np.array(dataset.subcat_1),
        'subcat_2': np.array(dataset.subcat_2),
    }
    return X


train = full_df[:n_trains]
test = full_df[n_trains:n_trains+n_devs]
# test = full_df[n_trains+n_devs:]

X_train = get_rnn_data(train)
Y_train = train["target"].values.reshape(-1, 1)

X_test = get_rnn_data(test)
Y_test = test["target"].values.reshape(-1, 1)

# X_test = get_rnn_data(test)
```

Python

Defining the RNN layers:

1. Input Layers: Each feature is considered as an input node in the input layer to the RNN model.

```python
def new_rnn_model(lr=0.001, decay=0.0):
    # Inputs
    name = Input(shape=[X_train["name"].shape[1]], name="name")
    item_desc = Input(shape=[X_train["item_desc"].shape[1]], name="item_desc")
    brand_name = Input(shape=[1], name="brand_name")
#     category = Input(shape=[1], name="category")
#     category_name = Input(shape=[X_train["category_name"].shape[1]], name="category_name")
    item_condition = Input(shape=[1], name="item_condition")
    num_vars = Input(shape=[X_train["num_vars"].shape[1]], name="num_vars")
    desc_len = Input(shape=[1], name="desc_len")
    name_len = Input(shape=[1], name="name_len")
    subcat_0 = Input(shape=[1], name="subcat_0")
    subcat_1 = Input(shape=[1], name="subcat_1")
    subcat_2 = Input(shape=[1], name="subcat_2")
```

2. Embedding layers: To adjust the outputs of the internal layers/nodes.

```python
    # Embeddings layers (adjust outputs to help model)
    emb_name = Embedding(MAX_TEXT, 20)(name)
    emb_item_desc = Embedding(MAX_TEXT, 60)(item_desc)
    emb_brand_name = Embedding(MAX_BRAND, 10)(brand_name)
#     emb_category_name = Embedding(MAX_TEXT, 20)(category_name)
#     emb_category = Embedding(MAX_CATEGORY, 10)(category)
    emb_item_condition = Embedding(MAX_CONDITION, 5)(item_condition)
    emb_desc_len = Embedding(MAX_DESC_LEN, 5)(desc_len)
    emb_name_len = Embedding(MAX_NAME_LEN, 5)(name_len)
    emb_subcat_0 = Embedding(MAX_SUBCAT_0, 10)(subcat_0)
    emb_subcat_1 = Embedding(MAX_SUBCAT_1, 10)(subcat_1)
    emb_subcat_2 = Embedding(MAX_SUBCAT_2, 10)(subcat_2)
```

3. RNN layers:

```
    # rnn layers (GRUs are faster than LSTMs and speed is important here)
    rnn_layer1 = GRU(16)(emb_item_desc)
    rnn_layer2 = GRU(8)(emb_name)
#     rnn_layer3 = GRU(8) (emb_category_name)
```

4. Main layers/Hidden layers:

```
    # main layers
    main_l = concatenate([
        Flatten()(emb_brand_name)
        # , Flatten() (emb_category)
        , Flatten()(emb_item_condition)
        , Flatten()(emb_desc_len)
        , Flatten()(emb_name_len)
        , Flatten()(emb_subcat_0)
        , Flatten()(emb_subcat_1)
        , Flatten()(emb_subcat_2)
        , rnn_layer1
        , rnn_layer2
        # , rnn_layer3
        , num_vars
    ])
    # (incressing the nodes or adding layers does not effect the time quite as much as the rnn layers)
    main_l = Dropout(0.1)(Dense(512, kernel_initializer='normal', activation='relu')(main_l))
    main_l = Dropout(0.1)(Dense(256, kernel_initializer='normal', activation='relu')(main_l))
    main_l = Dropout(0.1)(Dense(128, kernel_initializer='normal', activation='relu')(main_l))
    main_l = Dropout(0.1)(Dense(64, kernel_initializer='normal', activation='relu')(main_l))
```

5. Output layer & model optimiser:

```
    # the output layer.
    output = Dense(1, activation="linear")(main_l)

    model = Model([name, item_desc, brand_name, item_condition, num_vars, desc_len, name_len, subcat_0, subcat_1, subcat_2], output)

    optimizer = Adam(lr=lr, decay=decay)
    # (mean squared error loss function works as well as custom functions)
    model.compile(loss='mse', optimizer=optimizer)

    return model
```

Snippet of the RNN Model fitting summary defining the layers in given below:

```
Output exceeds the size limit. Open the full output data in a text editor
Model: "model"
__________________________________________________________________________________________________
 Layer (type)                Output Shape              Param #     Connected to
==================================================================================================
 brand_name (InputLayer)     [(None, 1)]               0           []

 item_condition (InputLayer) [(None, 1)]               0           []

 desc_len (InputLayer)       [(None, 1)]               0           []

 name_len (InputLayer)       [(None, 1)]               0           []

 subcat_0 (InputLayer)       [(None, 1)]               0           []

 subcat_1 (InputLayer)       [(None, 1)]               0           []

 subcat_2 (InputLayer)       [(None, 1)]               0           []

 item_desc (InputLayer)      [(None, 75)]              0           []

 name (InputLayer)           [(None, 10)]              0           []

 embedding_2 (Embedding)     (None, 1, 10)             1017650     ['brand_name[0][0]']

 embedding_3 (Embedding)     (None, 1, 5)              30          ['item_condition[0][0]']
...
Total params: 19,006,916
Trainable params: 19,006,916
Non-trainable params: 0
__________________________________________________________________________________________________
```

RNN Model Training & Evaluation:

Post declaring the RNN model, it is trained using the training split of the dataset over 2 epochs.

The accuracy of the predictions is measured using the RMSLE (Root Mean Square Logarithmic Error) function, which is calculated as below:

$$RMSLE = \sqrt{(\log(y_i + 1) - \log(\hat{y}_i + 1))^2}$$

Where:

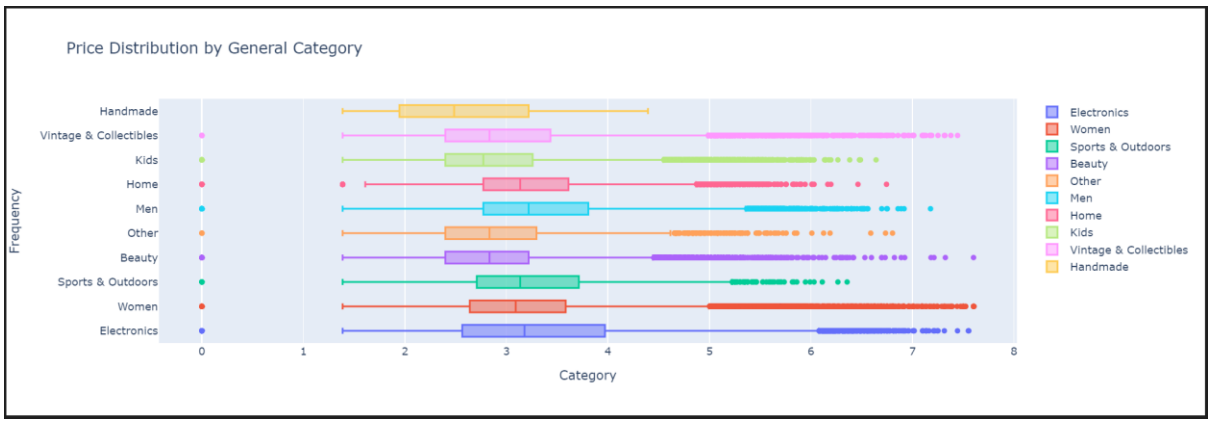
- y_i: Actual Price
- $\hat{y}_i$: Predicted Price

Accuracy obtained using the training dataset: 37.99%

Accuracy obtained using the test dataset: 44.77%

Inferences:

1. The accuracy is on the lower side (~40%) suggesting that the model is not the best predictor of prices at which the goods were sold.
2. But the model can act as a good reference point for sellers & buyers in approximating what the price of a second-hand product listing should be.
3. A reason for the relatively average accuracy of the model could be the existence of a large number of outliers as was observed in the price distribution box plots shown below:



Conclusion, Future Scope & Business Implications:

Although the model accuracy was fairly average (~40%) it could possible be explained by the presence of a large number of outliers in the prices of the product listings present in the dataset. Considering that the dataset was extracted from the second-hand sales information from the Mercari Ecommerce platform which does not have a standardised control on the prices of product listings (as of yet) the presence of outliers was to be expected.

We can conclude that the model is viable & can be considered as a good reference point to recommend prices to the seller & buyer for any particular product listing.

Future Scope:

1. Model can further be extrapolated to recommend product prices to established brands as they introduce new products into the market by comparing/clustering with similar offerings which are already available.
2. Model can be made more robust by including item image data. This can be a good indicator of quality of a product since degree of degradation is an important factor when considering the price of a second-hand product. At the moment, item quality was being accounted for the seller defined “item_condition_id” feature. With the introduction of image data, the model can be made more robust & human error arising from the manually entered “item_condition_id” rating can be eliminated.
3. Model could include the bids placed on any second-hand product listing, along with the winning bid. This might be a feature worth considering but tracking these values & extracting them could be difficult. Moreover, their actual contribution towards improving the model’s performance needs to be studied.
4. The RNN Model itself could be made more robust by including more internal/hidden layers & training the model over more epochs (at the moment trained over 2 epochs). This is a limitation of the computing power of the local system being used. More powerful machines could handle more robust models.