LAZADA DATA REVIEW

Analyzed by Upwiz

Content

1. **Business Understanding** is Determine Business Objectives by understanding from a business perspective
2. **Data Understanding** is the knowledge that you have about the data, the needs that the data will satisfy, its content and location
3. **Data Preparation and Exploration** is the process of cleaning and transforming raw data prior to processing and analysis
4. **Modelling** is the process of analyzing the data objects and their relationship to the other objects
5. **Business Insight** is the conclusions related to the business obtained from the data that we have analyzed

Business Understanding

Lazada is the number one online shopping and selling destination in Southeast Asia. As a pioneer of the eCommerce ecosystem in Southeast Asia, through a marketplace platform supported by a variety of unique marketing services, data services, and other services, Lazada offers a variety of products in various categories ranging from electronics to household goods, toys, fashion, sports equipment and daily necessities.

Business Understanding

Our team aim to analyze :

- Peak transaction period
- influence of unfamiliar keywords in product names on rating,
- influence of certain keywords in product reviews to predict its rating,
- brands that have the most contribution in sales and transactions,

and hopefully impact to maximize sales and reviews.

Data Understanding

The Dataset Retrieved From
kaggle.com/datasets/grikomsn/lazada-indonesian-reviews

It contains two csv files and one txt file, scraped on 2 October 2019 :

- 20191002-items.csv
- 20191002-reviews.csv
- categories.txt

categories txt File

research bounded for consumer electronics review from 5 product categories:

1. beli-harddisk-eksternal
2. beli-laptop
3. beli-smart-tv
4. jual-flash-drives
5. shop-televisi-digital

20191002-items.csv

in the 20191002-items.csv file there are description for the product that appeared under categories discussed earlier. There are **10942 products** described with 9 columns. The column description are as follows :

No.	Column	Data Type	Description
1.	itemId	Integer	Id on each item
2.	category	String	Category of product
3.	name	String	Name of product
4.	brandName	String	Brand Name of product
5.	url	String	Link of product
6.	price	Integer	Price of product
7.	averageRating	Float	Average rating of product
8.	totalReviews	Integer	Total reviews of product
9.	retrievedDate	Date Time	2 October 2019

int (3), float (1), str (4), date (1)

20191002-reviews.csv

in the 20191002-reviews.csv file there are reviews made by customer for products from previous csv. There are **203787 reviews** made from **consumer electronic products** sold between the year **2014** until **2019.** Here are the details:

No.	Column	Data Type	Description
1.	itemId	Integer	Id on each item
2.	category	String	Category of product
3.	name	String	Name of customers
4.	rating	Integer	rating of product
5.	originalRating	Integer	Original rating of product
6.	reviewTitle	String	Review title of product
7.	reviewContent	String	Review content of product
8.	likeCount	Integer	Total like of product
9.	upVotes	Integer	Total up votes of product

20191002-reviews.csv

No.	Column	Data Types	Description
10.	downVotes	Integer	Total down votes of product
11.	helpful	Boolean	
12.	relevance	Integer	Relevancy score of a review to a product
13.	boughtDate	String	Bought date of product
14.	clientType	String	Client operating system
15.	retrievedDate	Date Time	20191002

int (6), str (6), bool (1), date (1)

Data Preparation

1

EXPLORATORY DATA ANALYSIS
EDA using pandas_profiling package to generate a summary for the two csv's

1. Profiling report from file items.csv

From Overview

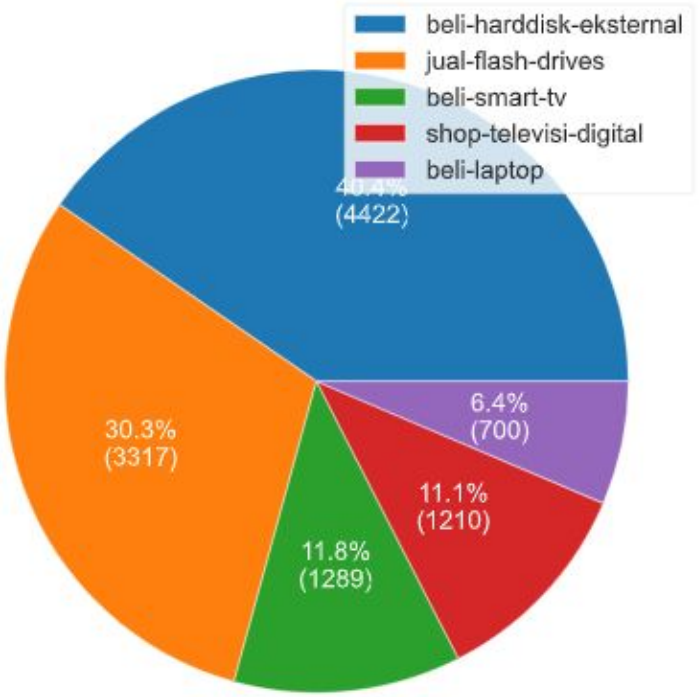
Overview

retrievedDate is a constant column,
there are 235 unique *brandName*

Prep – EDA

From Variables

The top *category* is



Prep – EDA

The *brands* with the most products are, in order:

Value	Count	Frequency (%)
SanDisk	1258	11.5%
Asus	946	8.6%
No Brand	945	8.6%
Lenovo	910	8.3%
Toshiba	674	6.2%

Prep – EDA

price ranged from 1000 to 275.000.000, with half of it under 1.095.000 (median value)

Statistics | Histogram | Common values | Extreme values

Quantile statistics

Minimum	1000
5-th percentile	27900
Q1	79200
median	1095000
Q3	3959000
95-th percentile	11999000
Maximum	275000000
Range	274999000
Interquartile range (IQR)	3879800

Descriptive statistics

Standard deviation	6681452.569
Coefficient of variation (CV)	2.21224137
Kurtosis	762.1288081
Mean	3020218.616
Median Absolute Deviation (MAD)	1051544
Skewness	20.29046952
Sum	3.30472321e+10
Variance	4.464180844e+13
Monotocity	Not monotonic

Prep – EDA

averageRating 47.5% fall at the value of 5, and progressively less for 4, 3, 2, and 1

Statistics | Histogram | Common values | Extreme values

Value	Count	Frequency (%)
5	5202	47.5%
4	3708	33.9%
3	1173	10.7%
1	433	4.0%
2	426	3.9%

Prep – EDA

totalReview column have 40% of its value 1

Value	Count	Frequency (%)
1	4456	40.7%
2	1508	13.8%
3	875	8.0%
4	504	4.6%
5	403	3.7%

Prep – EDA

rating of 81.5% reviews are 5 star rating

Value	Count	Frequency (%)
5	166016	81.5%
4	17567	8.6%
1	9749	4.8%
3	7073	3.5%
2	3382	1.7%

Data Preparation

2. Profiling report from file reviews.csv
   From Overview (warnings)

Warnings

`helpful` has constant value "203787"	Constant
`retrievedDate` has constant value "203787"	Constant
Dataset has 3113 (1.5%) duplicate rows	Duplicates
`name` has a high cardinality: 40099 distinct values	High cardinality
`reviewTitle` has a high cardinality: 5725 distinct values	High cardinality
`reviewContent` has a high cardinality: 38071 distinct values	High cardinality
`boughtDate` has a high cardinality: 1690 distinct values	High cardinality
`upVotes` is highly correlated with `likeCount`	High correlation
`likeCount` is highly correlated with `upVotes`	High correlation
`originalRating` has 203779 (> 99.9%) missing values	Missing
`reviewTitle` has 180383 (88.5%) missing values	Missing
`reviewContent` has 96758 (47.5%) missing values	Missing
`boughtDate` has 7107 (3.5%) missing values	Missing
`likeCount` is highly skewed (γ1 = 98.02694893)	Skewed
`upVotes` is highly skewed (γ1 = 98.02694893)	Skewed
`downVotes` is highly skewed (γ1 = 20.07366216)	Skewed
`likeCount` has 184611 (90.6%) zeros	Zeros
`upVotes` has 184611 (90.6%) zeros	Zeros
`downVotes` has 199268 (97.8%) zeros	Zeros

helpful and *retrievedDate* are constant columns,
dataset still have a lot of duplicated rows,
originalRating is almost 100% missing,
likeCount, *upVotes*, and *downVotes* are **more than 90% zero**

Prep – EDA

3.5% of *boughtDate* are missing.
The common dates are 09-Sep-2019, 27-Mar-2019, 11-Nov-2018, 12-Jul-2019, and 12-Des-2018

Value	Count	Frequency (%)
09 Sep 2019	7421	3.6%
27 Mar 2019	6093	3.0%
11 Nov 2018	4981	2.4%
12 Jul 2019	4316	2.1%
12 Des 2018	3460	1.7%
14 Mei 2019	2479	1.2%
10 Des 2018	2428	1.2%
08 Agu 2019	1805	0.9%
16 Mei 2019	1539	0.8%
11 Des 2018	1406	0.7%
Other values (1680)	160752	78.9%
(Missing)	7107	3.5%

Prep – EDA

most customers (82.2%) uses androidApp, followed by desktop, iosApp, and mobile.

Value	Count	Frequency (%)
androidApp	167478	82.2%
desktop	12251	6.0%
iosApp	9105	4.5%
mobile	8298	4.1%
mobile-app	6655	3.3%

Data Preparation

2

DATA CLEANING

1. Removing low-information columns

```
items = items.drop(columns = ['url', 'retrievedDate'])
reviews = reviews.drop(columns = ['originalRating', 'likeCount', 'upVotes', 'downVotes',
                                  'helpful', 'relevanceScore', 'retrievedDate'])
display(items.head(), reviews.head())
```

- From items

  url column with **unique values**, and *retrievedDate* column with **constant value**.

- From reviews,

  originalRating, *likeCount*, *upVotes*, and *downVotes* columns with **>90% missing or zero values**, *helpful* and *retrievedDate* columns with **constant value**, and *relevanceScore* for its **unknown** interpretation.

Prep – Cleaning

2. Joining the two dataframes
   Join using *itemId*, and preserving the review table.

```
df = items.merge(reviews, how = 'right', on = 'itemId')
df.head()
```

itemId	category_x	name_x	brandName	price	averageRating	totalReviews	category_y	name_y	rating	reviewTitle	reviewContent	boughtDate	clientType
100002528	beli-harddisk-eksternal	TOSHIBA Smart HD LED TV 32" - 32L5650VJ Free B...	Toshiba	2499000	4	8	beli-harddisk-eksternal	Kamal U.	5	NaN	bagus mantap dah sesui pesanan	09 Apr 2019	androidApp

3. Merging *reviewTitle* and *reviewContent* column
   At the end, both of them are review in text.

```
df['reviewContent'] = df.reviewTitle.fillna('').str.cat(df.reviewContent, sep = ' ')
df.head()
```

Prep – Cleaning

4. Renaming and deleting columns

```
df = df.rename(columns = {'name_x'        : 'productName',
                          'reviewContent' : 'review',
                          'category_x'    : 'category',
                          'name_y'        : 'customerName'})
df = df.drop(columns = ['category_y', 'reviewTitle', 'customerName'])
df.head()
```

	itemId	category	productName	brandName	price	averageRating	totalReviews	rating	review	boughtDate	clientType
0	100002528	beli-harddisk-eksternal	TOSHIBA Smart HD LED TV 32" - 32L5650VJ Free B...	Toshiba	2499000	4	8	5	bagus mantap dah sesui pesanan	09 Apr 2019	androidApp
1	100002528	beli-smart-tv	TOSHIBA Smart HD LED TV 32" - 32L5650VJ Free B...	Toshiba	2499000	4	8	5	bagus mantap dah sesui pesanan	09 Apr 2019	androidApp
2	100002528	jual-flash-drives	TOSHIBA Smart HD LED TV 32" - 32L5650VJ Free B...	Toshiba	2499000	4	8	5	bagus mantap dah sesui pesanan	09 Apr 2019	androidApp

- First, there are two *category* column. we only need one
- Second, the *name* on items refer the product's *name*, but on reviews it refer to the customer's name
- Then the *reviewTitle* can be deleted since its already absorbed into *review*
- Also we propose deleting *customerName* column due to lack of *customerId* as a unique customer identifier, so the name serves no purpose

Prep – Cleaning

5. Removing the duplicated rows

   As you can see in previous slide, the first three rows is exactly the same transaction and review, but flagged under different *category*. This indicated that seller often put their product to an array of categories to increase impression, even if its inaccurate.

   So, we propose deleting the *category* column altogether.

```
df = df.drop(columns = 'category').drop_duplicates().reset_index(drop = True)
df
```

	itemId	productName	brandName	price	averageRating	totalReviews	rating	review	boughtDate	clientType
0	100002528	TOSHIBA Smart HD LED TV 32" - 32L5650VJ Free B...	Toshiba	2499000	4	8	5	bagus mantap dah sesui pesanan	09 Apr 2019	androidApp
1	100002528	TOSHIBA Smart HD LED TV 32" - 32L5650VJ Free B...	Toshiba	2499000	4	8	4	Bagus, sesuai foto	24 Sep 2017	androidApp
2	100002528	TOSHIBA Smart HD LED TV 32" - 32L5650VJ Free B...	Toshiba	2499000	4	8	5	ok mantaaapppp barang sesuai pesanan.. good ok...	04 Apr 2018	androidApp
3	100002528	TOSHIBA Smart HD LED TV 32" - 32L5650VJ Free B...	Toshiba	2499000	4	8	4	bagus sesuai	22 Sep 2017	androidApp
4	100002528	TOSHIBA Smart HD LED TV 32" - 32L5650VJ Free B...	Toshiba	2499000	4	8	5	NaN	17 Agu 2018	androidApp

Prep – Cleaning

6. Fixing datetime format

   Python by default is using month abbreviation for US locale, so we need to change our locale to ID, then using pandas.to_datetime to convert *boughtDate* column

```
import locale

locale.setlocale(locale.LC_ALL, 'ID')
loc = locale.getlocale()
print(loc)
```

```
('id_ID', 'ISO8859-1')
```

```
df.boughtDate = pd.to_datetime(df.boughtDate,  format = '%d %b %Y')
df.head()
```

	itemId	productName	brandName	price	averageRating	totalReviews	rating	review	boughtDate	clientType
0	100002528	TOSHIBA Smart HD LED TV 32" - 32L5650VJ Free B...	Toshiba	2499000	4	8	5	bagus mantap dah sesui pesanan	2019-04-09	androidApp
1	100002528	TOSHIBA Smart HD LED TV 32" - 32L5650VJ Free B...	Toshiba	2499000	4	8	4	Bagus, sesuai foto	2017-09-24	androidApp
2	100002528	TOSHIBA Smart HD LED TV 32" - 32L5650VJ Free B...	Toshiba	2499000	4	8	5	ok mantaaapppp barang sesuai pesanan.. good ok...	2018-04-04	androidApp
3	100002528	TOSHIBA Smart HD LED TV 32" - 32L5650VJ Free B...	Toshiba	2499000	4	8	4	bagus sesuai	2017-09-22	androidApp

Prep – Cleaning

7. Generate categories

   This time we will **re-assign product category** based on the product name, from this we can inspect whether **unfamiliar product name can affect sales.**

   We use **LDA (latent dirichlet allocation) model** under **pycaret's nlp package** to **split** our products into **4 categories**, topic model will consider intrinsic property of a product, that is its name, brand, and price.

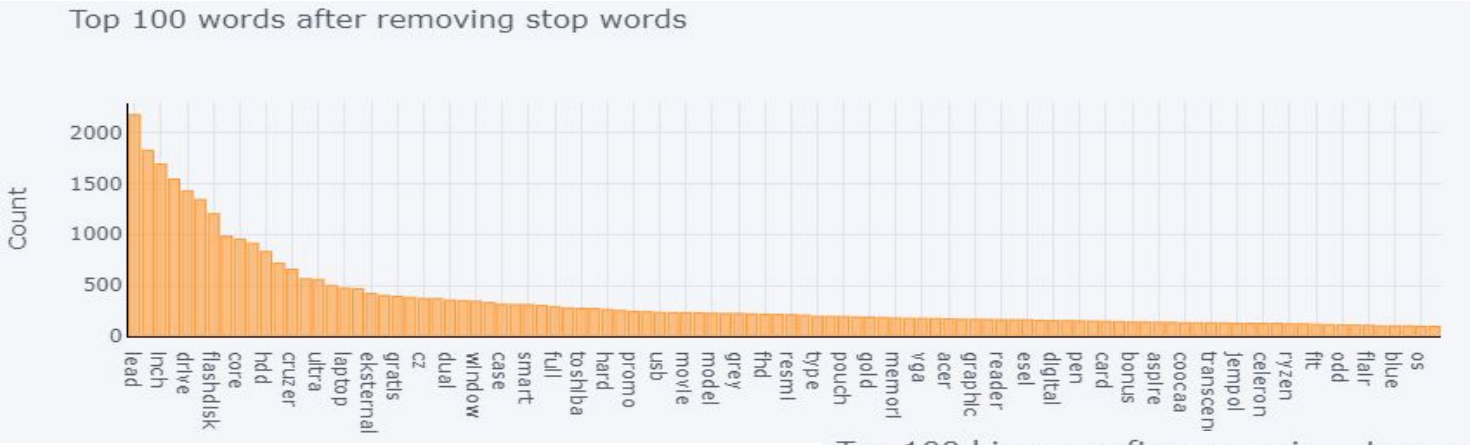
GENERATING MODEL

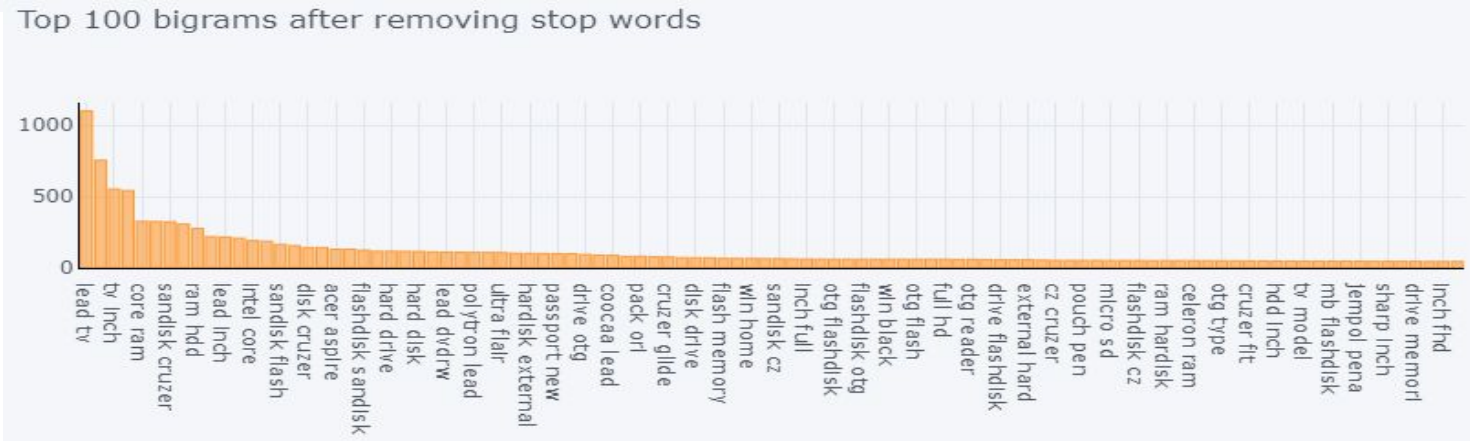
```
[ ]  from pycaret.nlp import *

     topic = setup(data = items[['name', 'brandName', 'price']],
                   target = 'name',
                   custom_stopwords = ['gb', 'free'])
```

Description	Value
session_id	8843
Documents	10942
Vocab Size	1445
Custom Stopwords	True

Prep – Cleaning

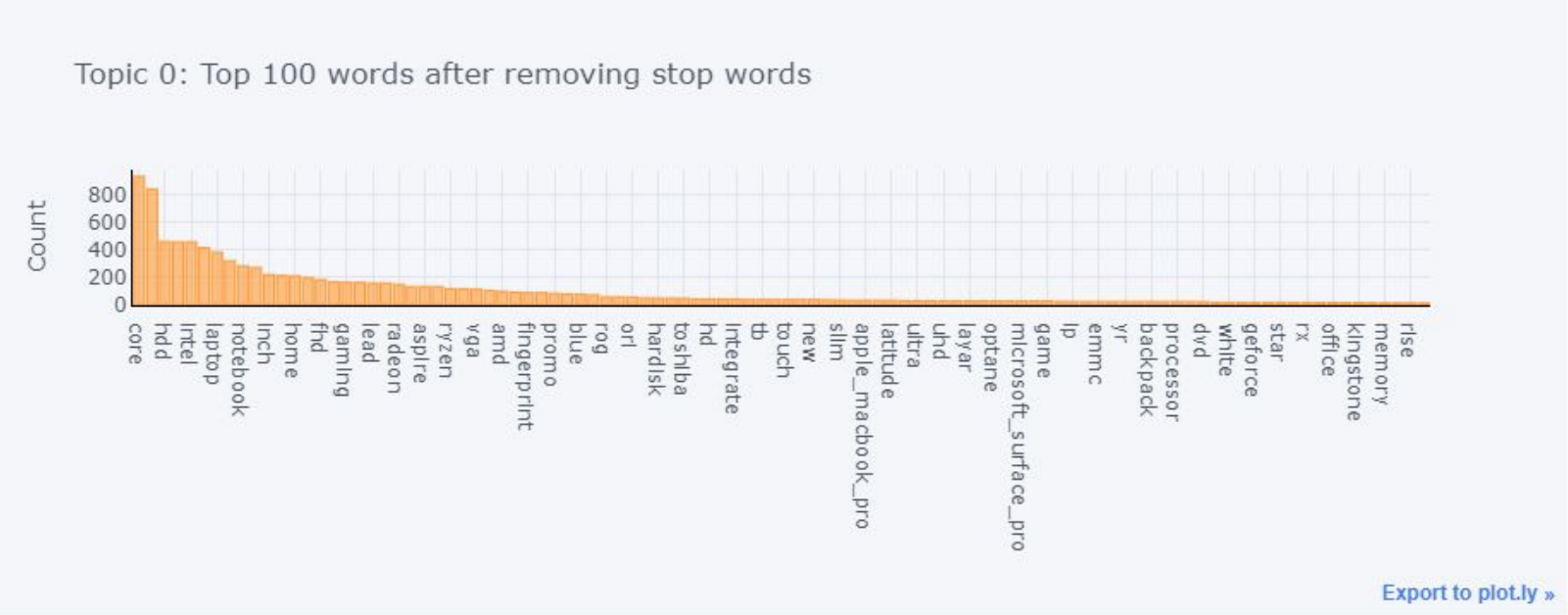
PLOT INSPECTION





Prep – Cleaning

PLOT INSPECTION

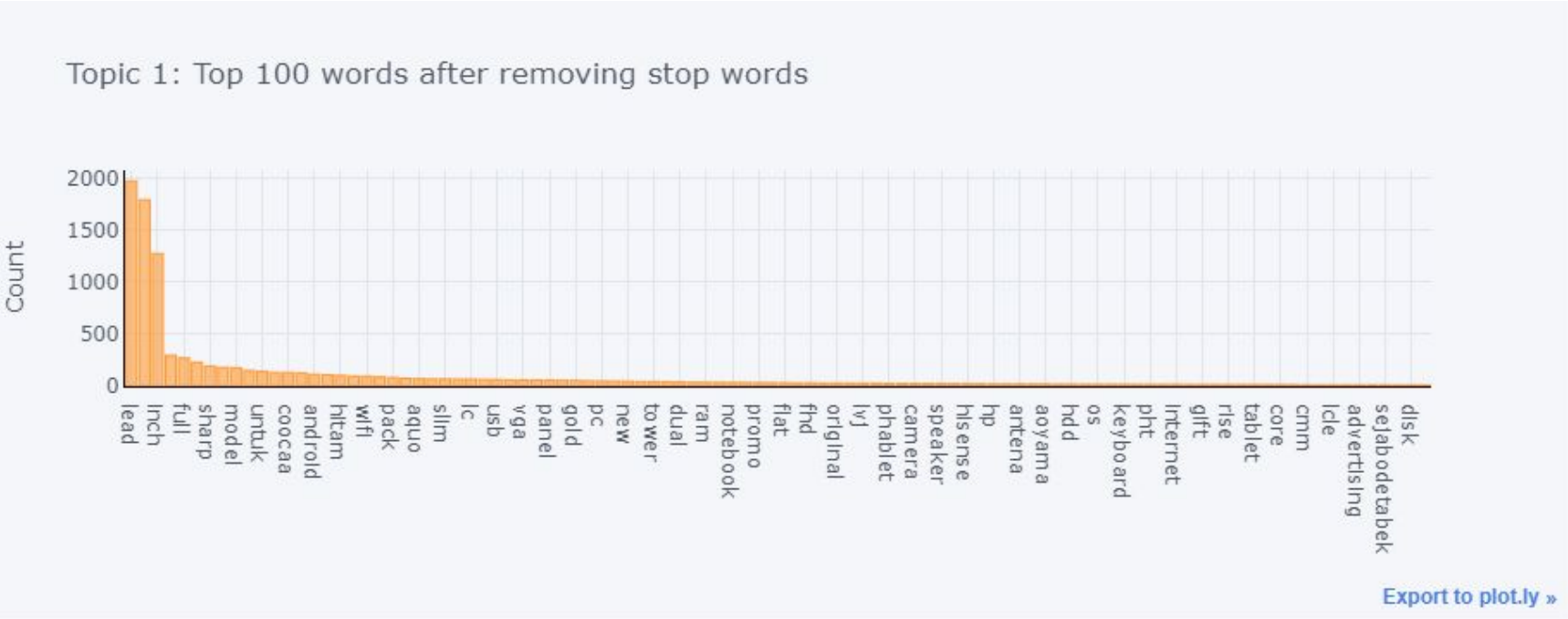


based of words such as core, hdd, vga, amd, intel, etc, we can conclude:

Topic 0 akeen to **laptop** products,

Prep – Cleaning

PLOT INSPECTION

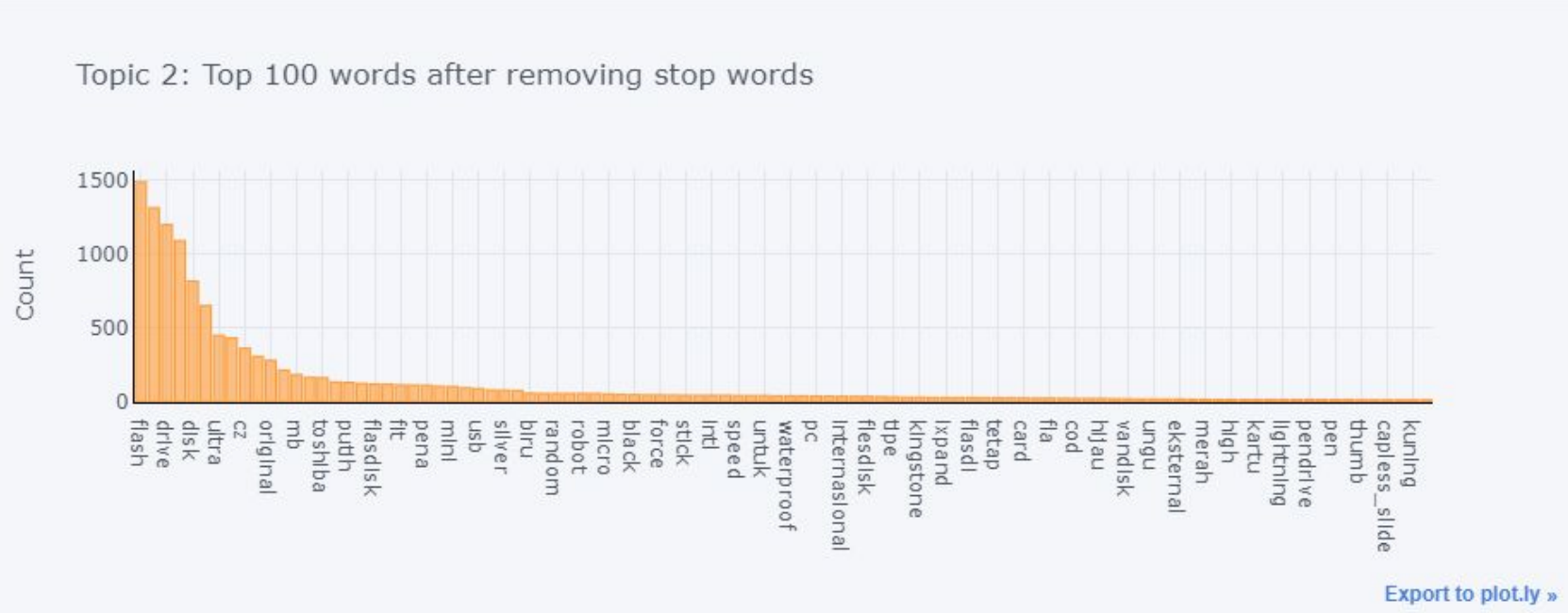


based of words such as led, inch, android tv, usb port, vga port, etc, we can conclude that:

Topic 1 akeen to **tv** products,

Prep – Cleaning

PLOT INSPECTION

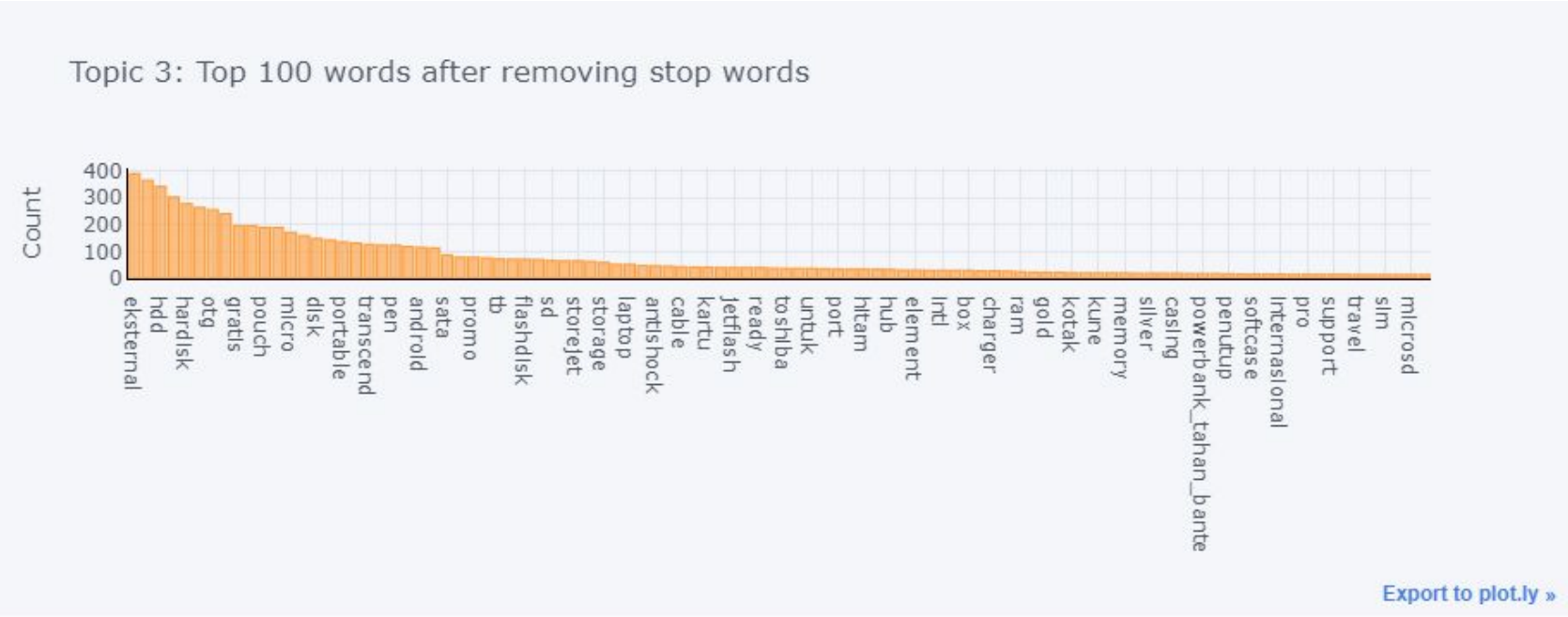


based of word such as flash, drive, usb, cz, etc, we conclude that:

Topic 2 akeen to **flashdrive** products,

Prep – Cleaning

PLOT INSPECTION

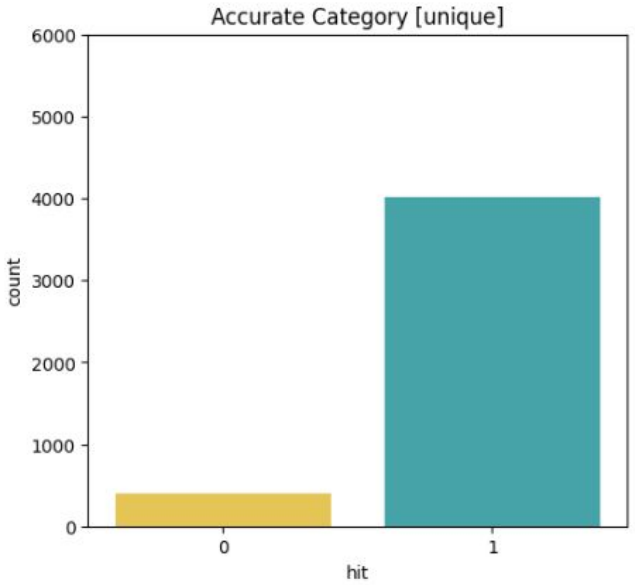


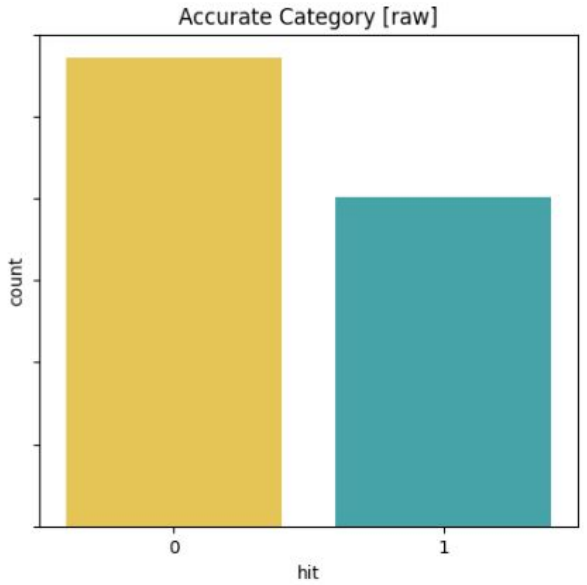
based of words such as hard disk, hdd, external, storage, etc, we conclude that:

Topic 3 akeen to **harddrive** products,

Prep – Cleaning

Model Evaluation





The left bar plot is checking whether prediction is among the listed category, and the right is checking against all listed categories.

We can see that the prediction landed on at least one of the listed category most of the time, and eliminate most of the duplicate category issue.

Prep – Cleaning

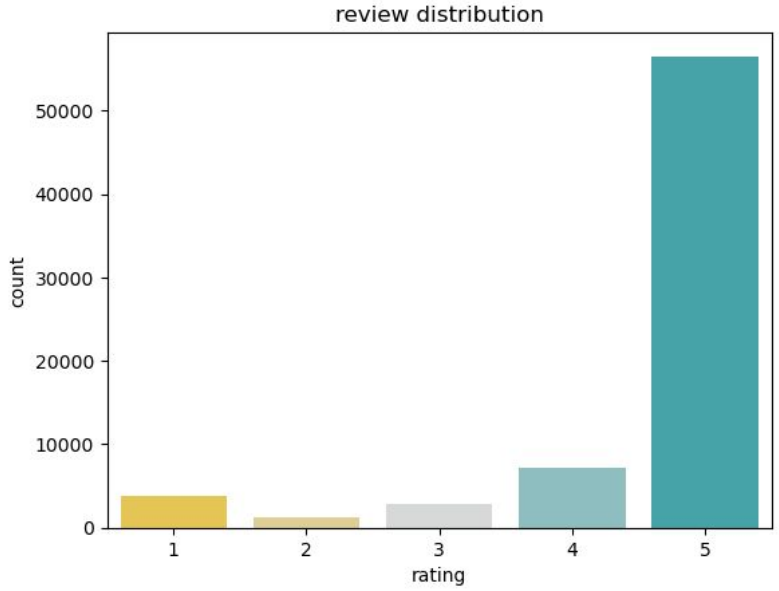
Assign model

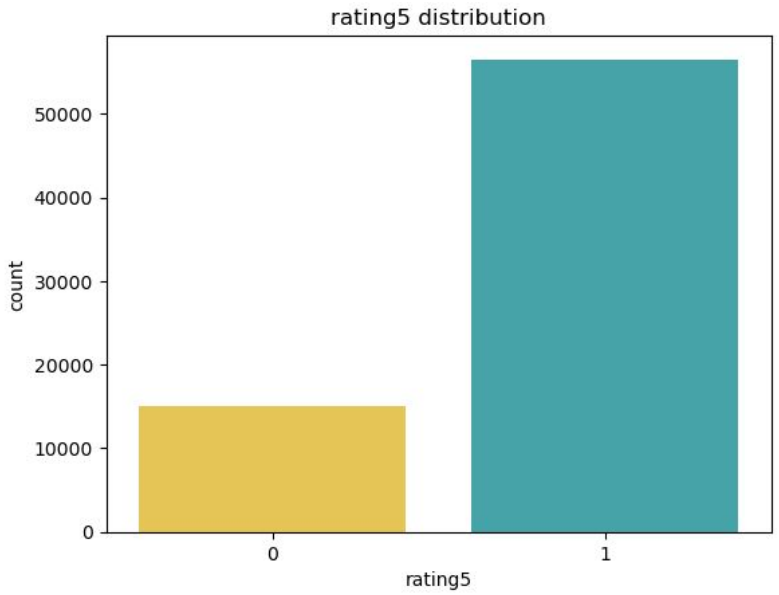
	name	brandName	price	Topic_0	Topic_1	Topic_2	Topic_3	Dominant_Topic	Perc_Dominant_Topic
0	toshiba lead tv lvj bracket tv	Toshiba	2499000	0.021561	0.795407	0.173026	0.010007	Topic 1	0.80
1	toshiba full smart lead tv	Toshiba	3788000	0.025494	0.758088	0.204587	0.011832	Topic 1	0.76
2	inch full flat lead digital tv	LG	3850000	0.021561	0.949657	0.018776	0.010007	Topic 1	0.95
3	lead tv lc lei	Sharp	1275000	0.031182	0.927191	0.027154	0.014472	Topic 1	0.93
4	laptop multimedia hdd lead dvdrw	Lenovo	3984100	0.755140	0.210826	0.022201	0.011832	Topic 0	0.76
...	...	...	...	...	...	...	...	...	...

Results dataframe **consist of the probability** of a product **belonging to a topic**, the dominant topic, and its probability.
We can view the **dominant** probability as the **confidence of our model** to assign that product to a category.
Products with **common keywords** will have **higher confidence value**.

Prep – Cleaning

8. Compiling rating under 5





As seen from the countplot above, majority of ratings are 5, with the rest of rating values making up the minority this will result in a biassed analysis

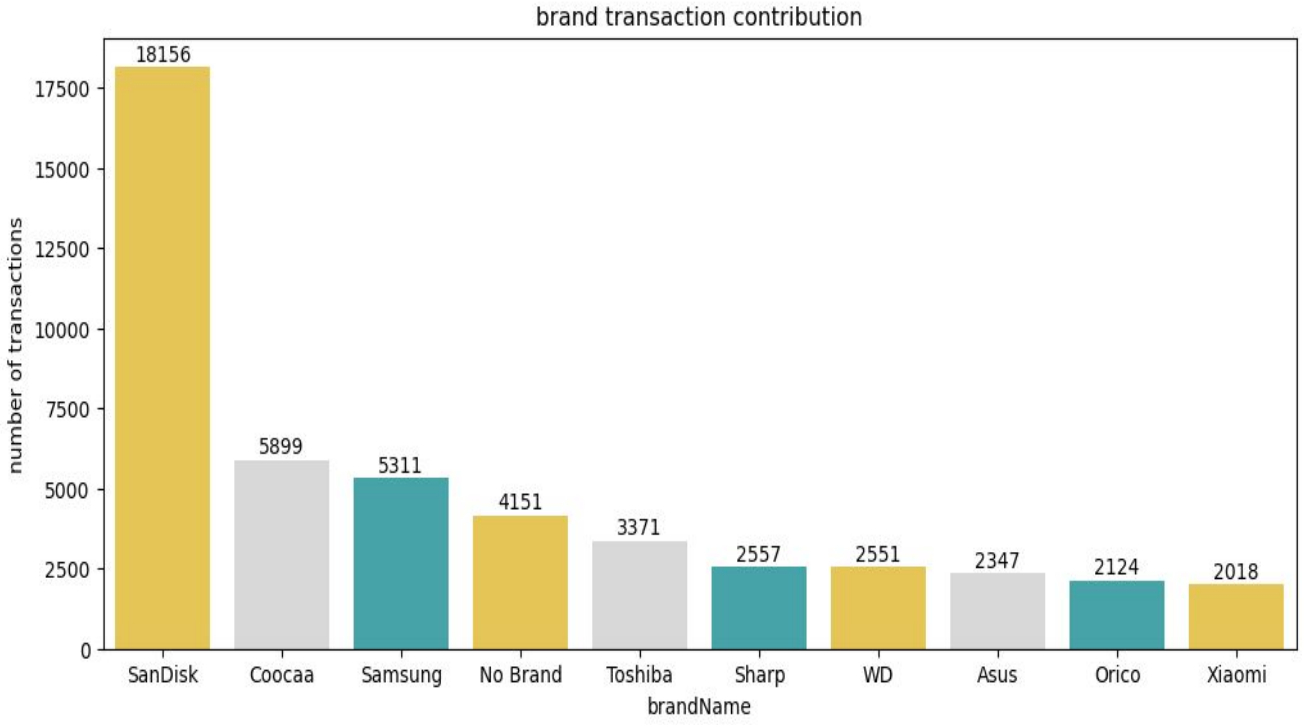
So, we decide to merge all the <5 rating

Data Exploration

1

Brands that have the most contribution in sales

- by transaction



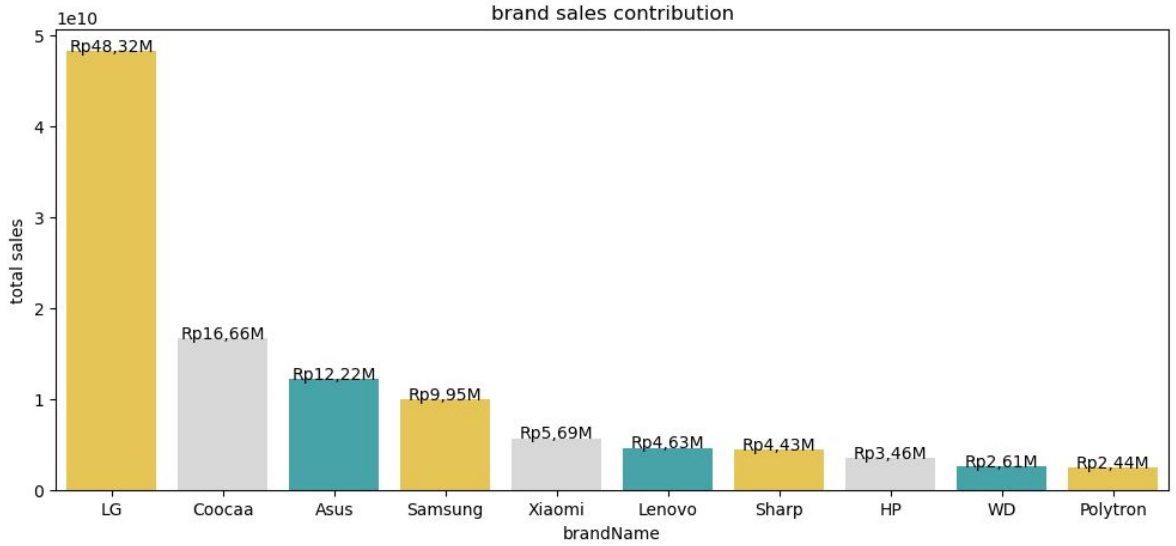
Based on the chart beside, from the **top 10 brands** whose **transactions contributed the most** is **Sandisk**, with a total of 18.156 transactions. Followed by **Coocaa** with a total of 5,899 transactions, and eight other brands, where the **sandisk and Coocaa** brands have a **significant difference** in the number of transactions, compared to the coocaa brand and the other eight brands.

Data Exploration

1

Brands that have the most contribution in sales

- by sales



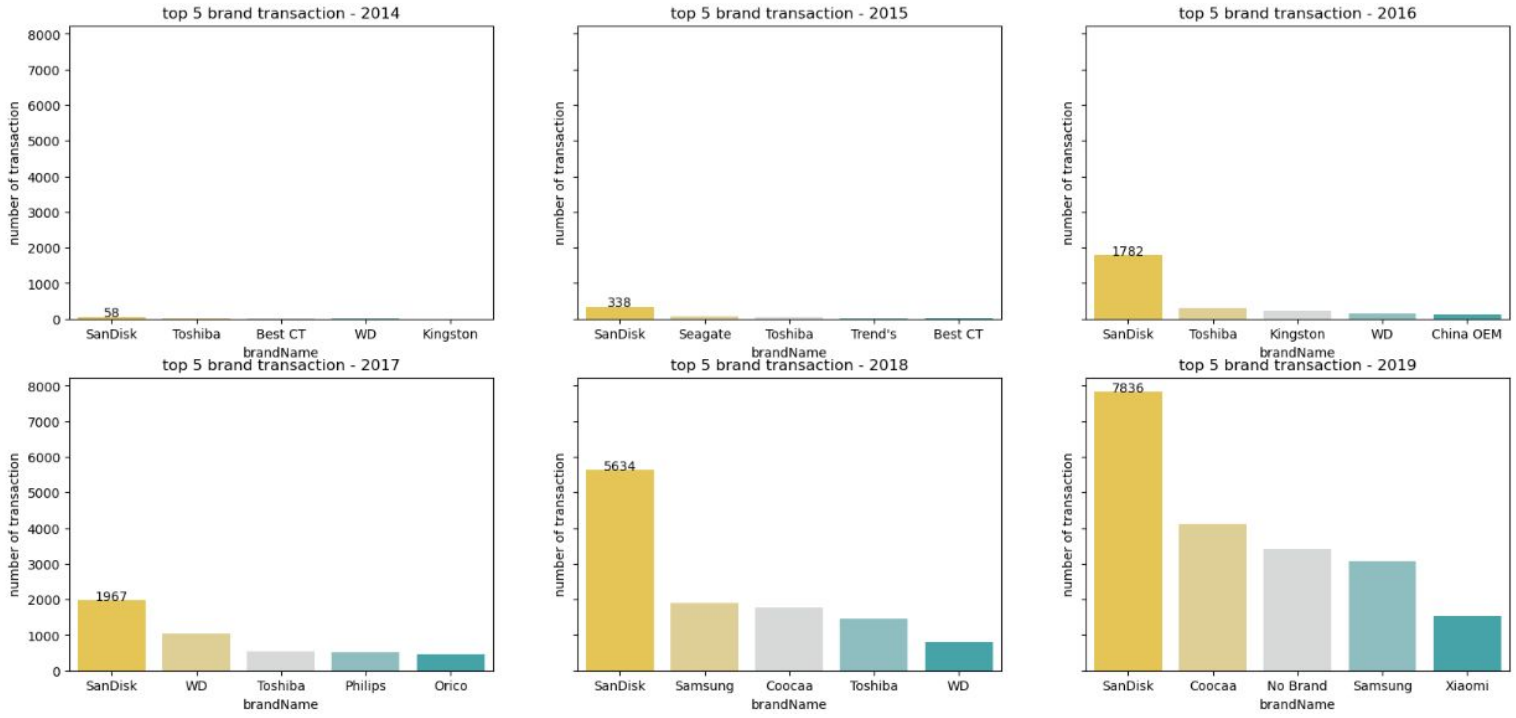
Based on the chart beside, from the **top 10 brands** that sell the most is **LG** with total sales of IDR 48.32 billion. Followed by **Coocaa** with total sales of IDR 16,66 billion, and eight other brands, where the **LG and Coocaa** brands have a **significant difference** in total sales, compared to the coocaa brand and the other eight brands.

Data Exploration

2

Brands contribution overtime

- by transaction



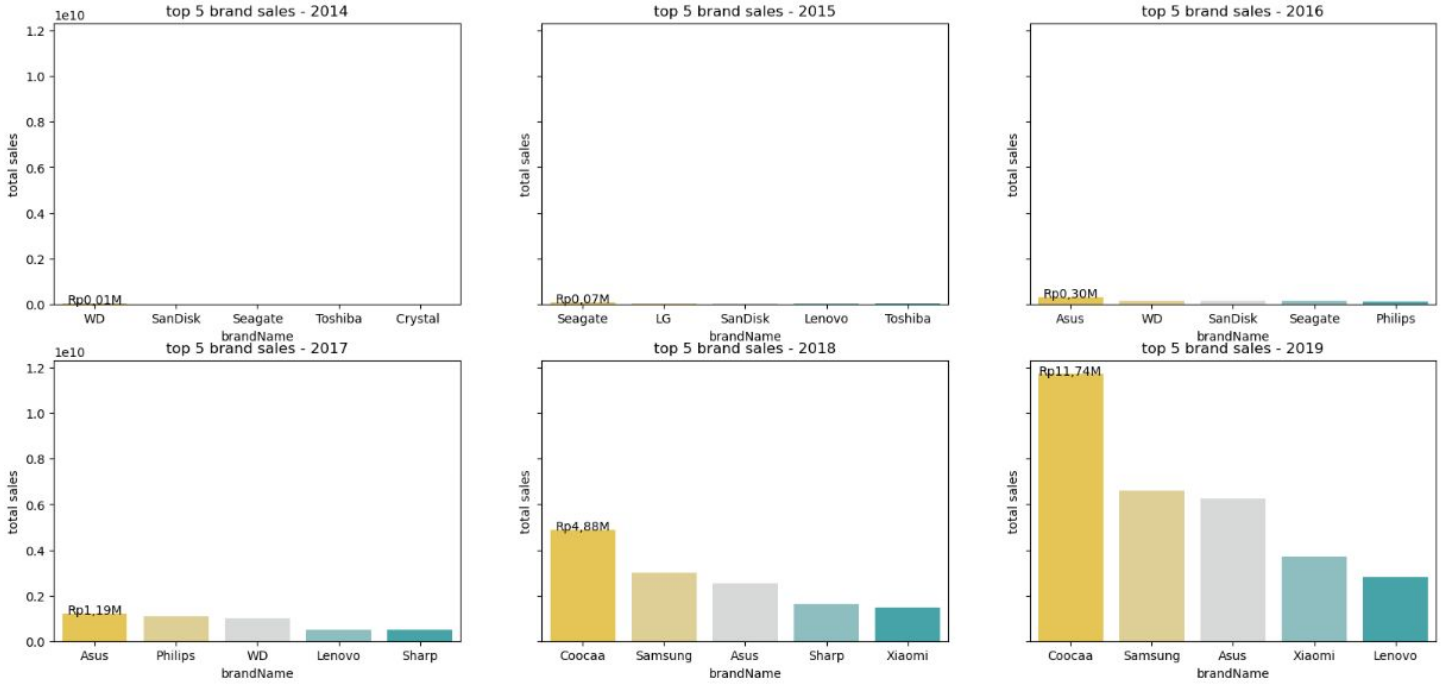
Based on the **annual chart** aside, from the **top 5** brands, **sandisk** brand **always** get **the highest** number of transactions, while the **four brands** in the top five always **change every year**.

Data Exploration

2

Brands contribution overtime

- by sales



Based on the annual chart beside, total sales of the **brands** that are in the **top 5 always change every year**.

Even though **Sandisk** brand has **the most transaction**, it **does not make a lot of sales** because the **price** of the products are **cheap**.

Data Exploration

As you probably have noticed, the brand that supposedly have **the largest sales** overall (LG) is **missing**

this turns out is largely the effect of one particular product labeled under LG brand (itemId 9092536), that have an astronomical price at Rp275 Million, have a high transaction record at 519 transactions, yet all of them missed the date

this should ring the fraud product alarm, but for completeness sake we will not discard this particular product

```
items = pd.read_csv('20191002-items.csv')
items[items.itemId == 9092536][['itemId', 'name', 'brandName', 'price']].drop_duplicates()
```

	itemId	name	brandName	price
4386	9092536	LG Cinema 3D Smart TV in 4K Ultra HD Resoluti...	LG	275000000

```
reviews = pd.read_csv('20191002-reviews.csv')
reviews[reviews.itemId == 9092536].shape[0]
```

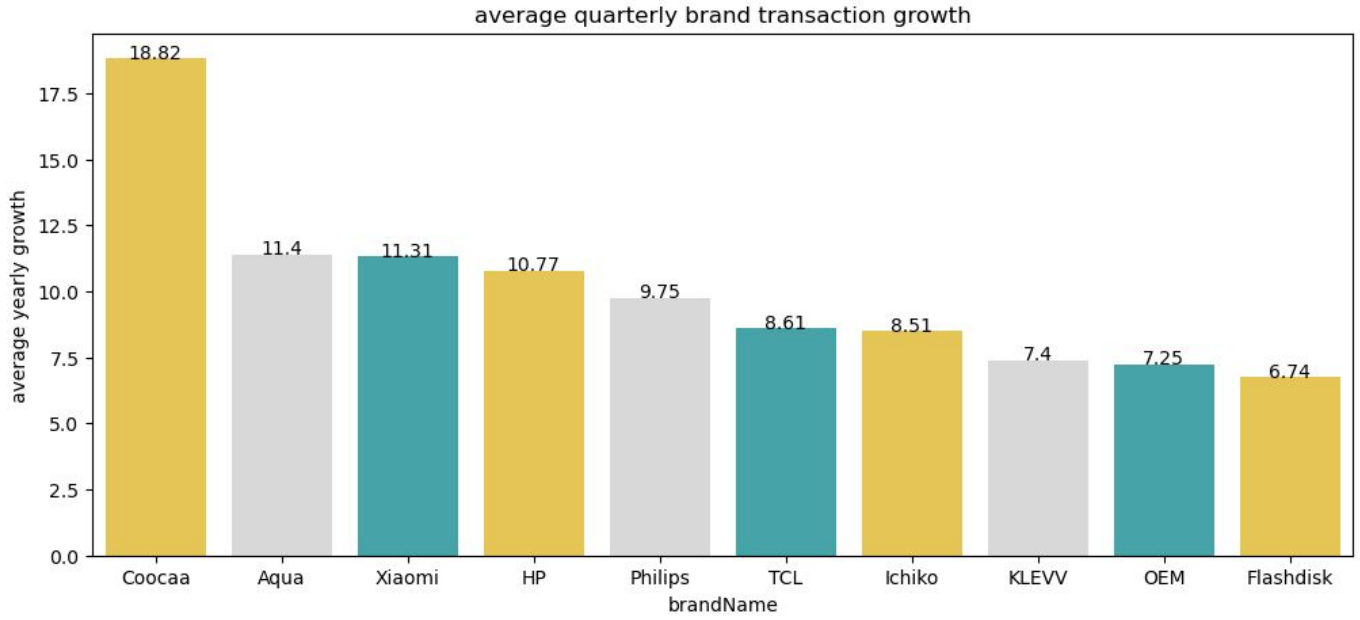
```
519
```

Data Exploration

3

Brands growth

To measure brands growth, we will use the average quarter to quarter transaction percentage increase.



Data Exploration

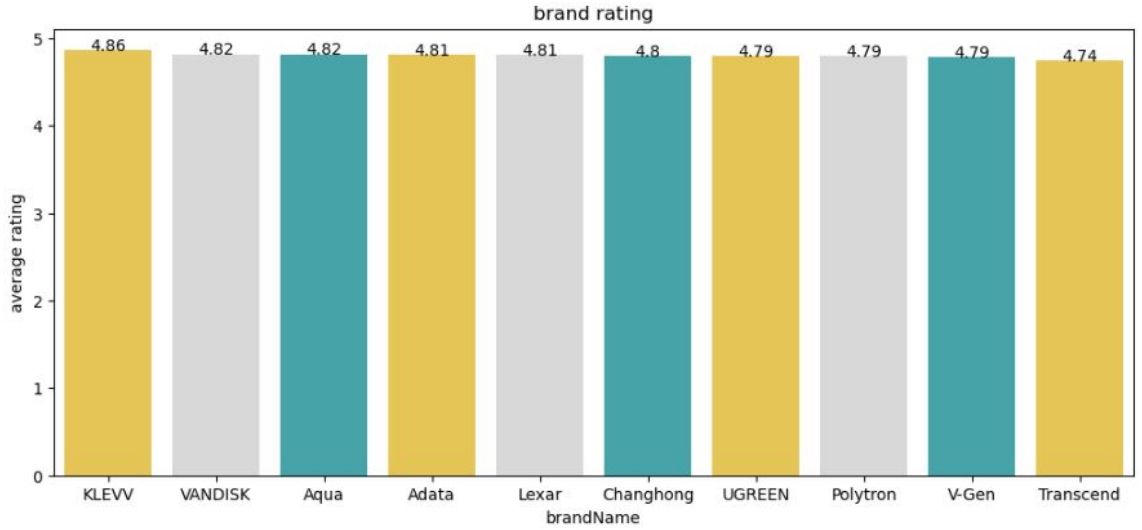
4

Brands rating

Laplace Rule of Succession

We will **balance** brand that **have low number of transaction** with **adding** a review with **rating 1 and rating 5** to each brands.

With this, the brands that have consistently **higher rating** among large number of reviews will **stand on top**.

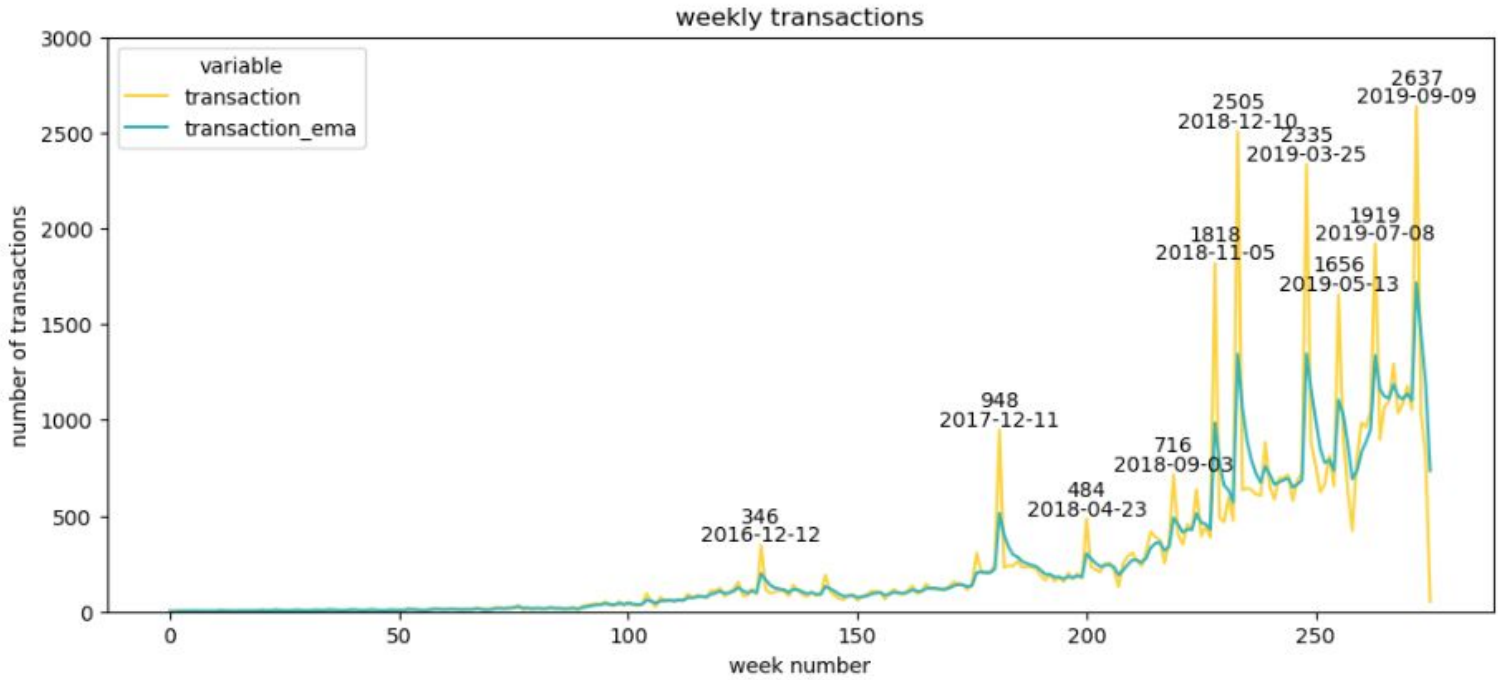


Data Exploration

5

Peak transaction period

- overall transaction behaviour (weekly basis)

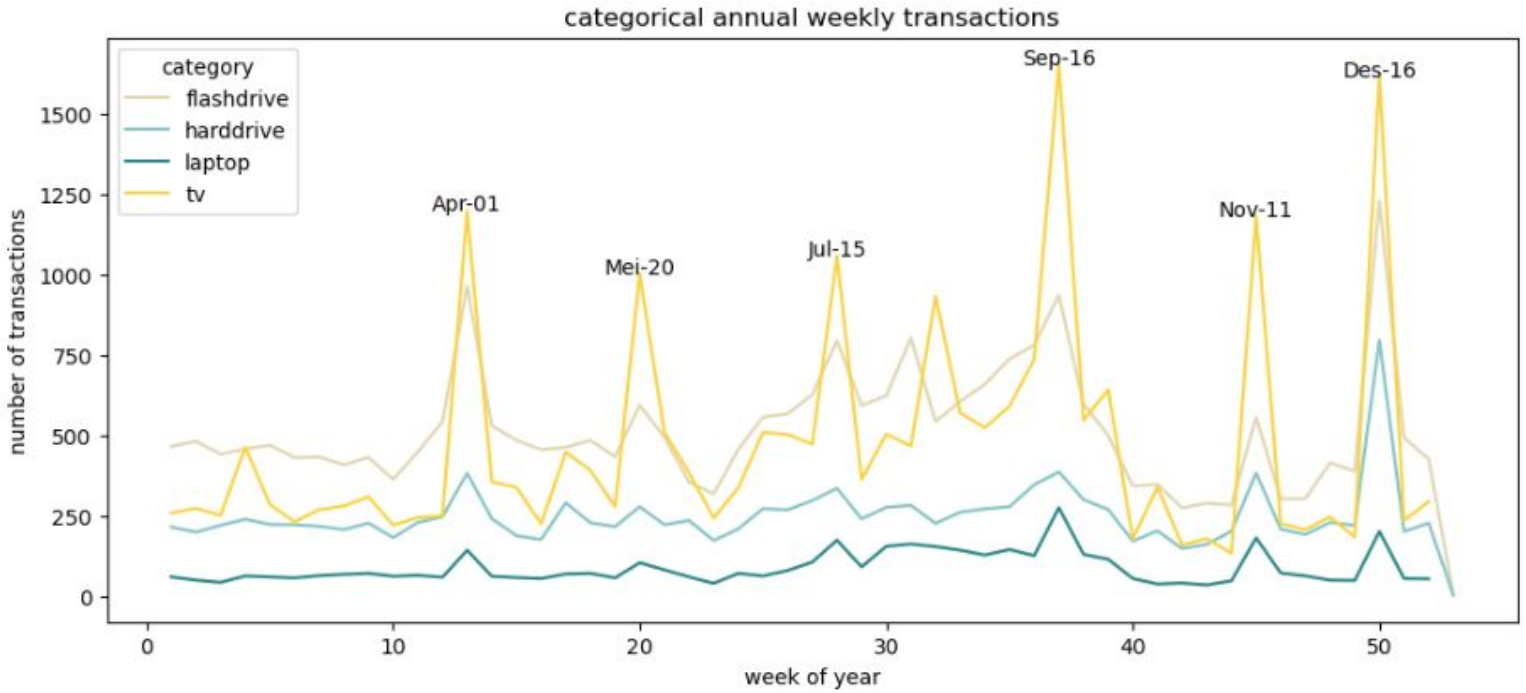


Data Exploration

5

Peak transaction period

- overall transaction in each category

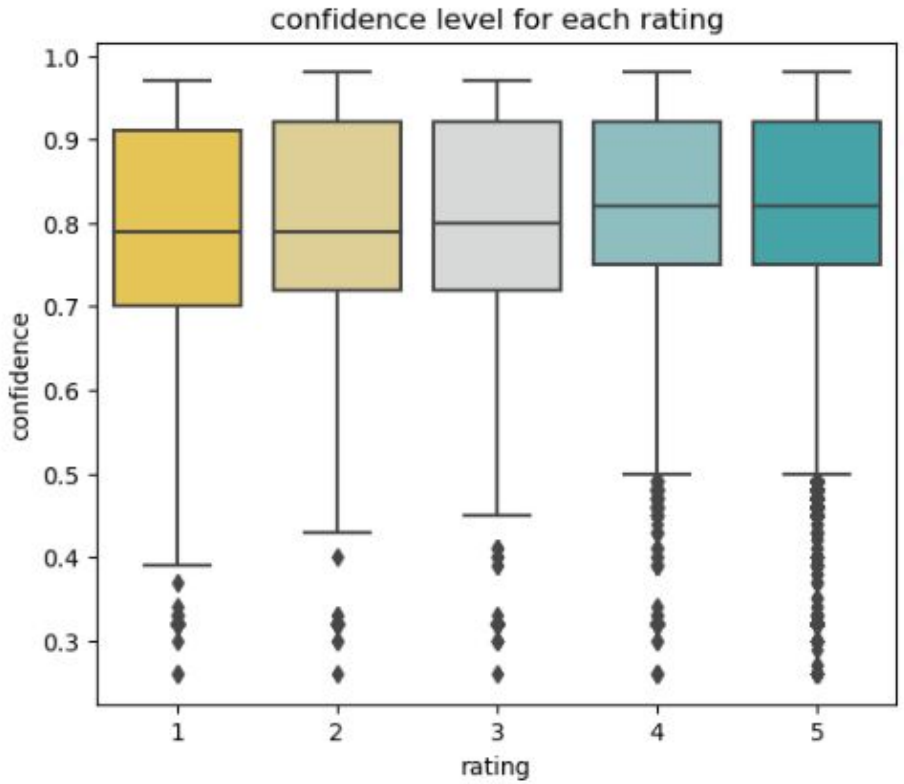


Data Exploration

6

Brand title effect

- without grouping



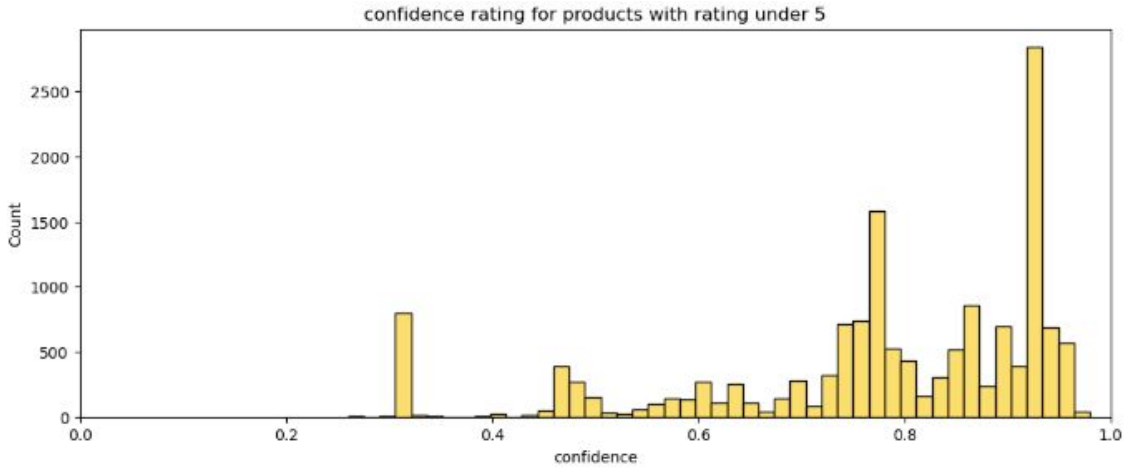
Visually, there are increase in median value and tightening of IQR across ratings, indicating that there are a **difference** in **confidence level** across all rating values

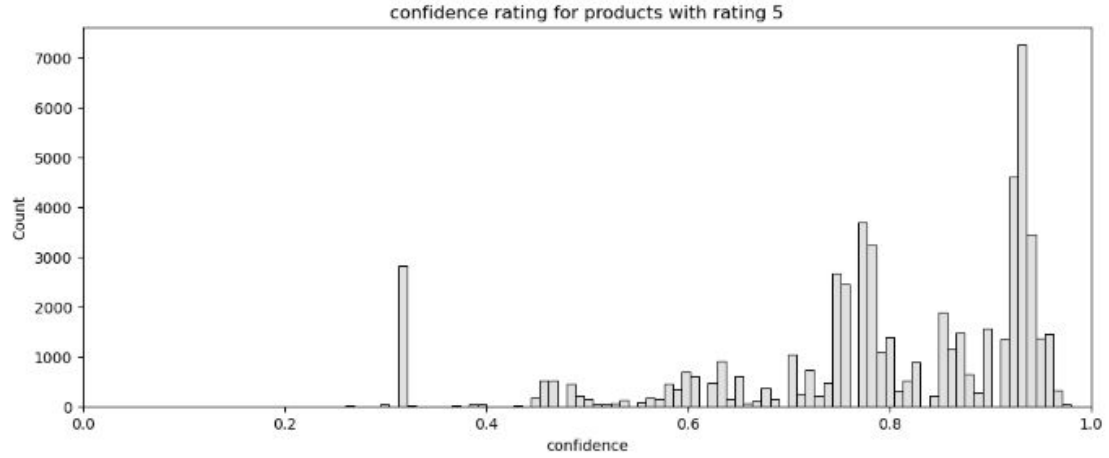
Data Exploration

6

Brand title effect

- with grouping
  product with rating 5 will be compared against product that have rating less than 5





Visually, **neither** of them seemed to be **normal distributed**
with **shapiro test**, both of them are **not normally distributed under 5% significance**

Data Exploration

6

Brand title effect

We will use **mann-whitney u test** (a nonparametric for uneven sample) to see whether those **two distributions are significantly different.**

```
stats.mannwhitneyu(rating1conf, rating5conf)
```

```
MannwhitneyuResult(statistic=404784754.0, pvalue=3.560152728277421e-23)
```

The above **shows** that the **two distributions are significantly different.**

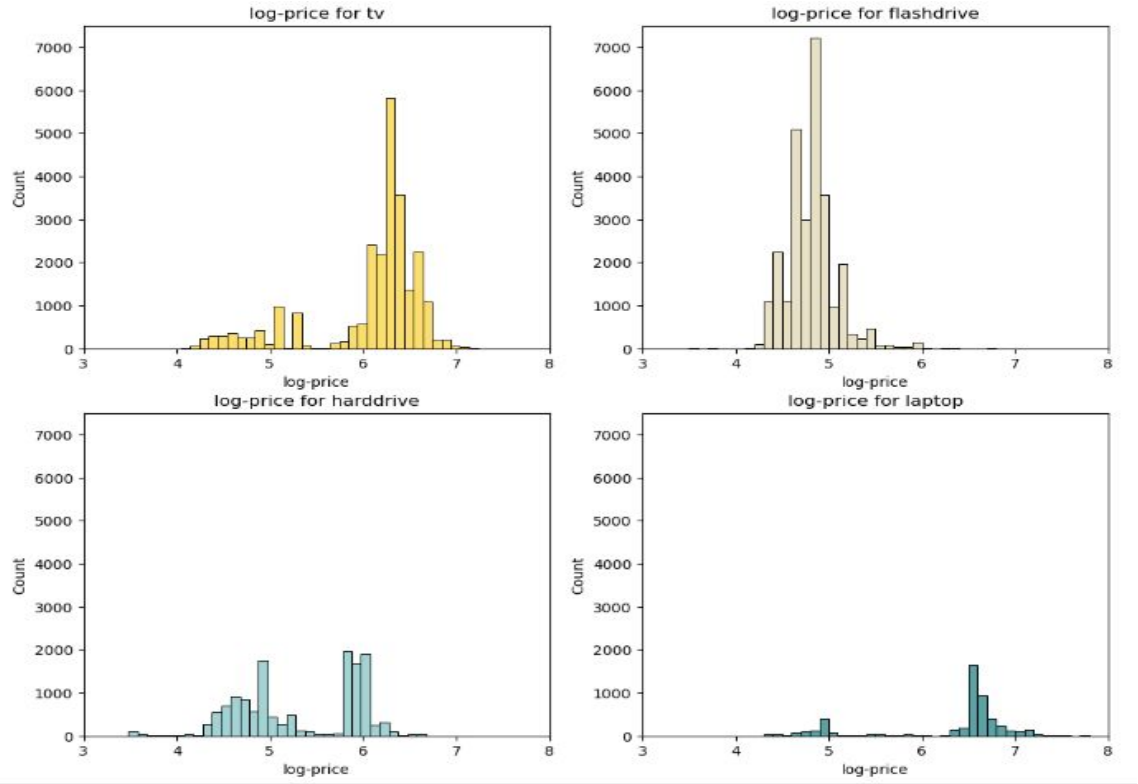
rating5	mean	median
0	0.776231	0.80
1	0.789553	0.82

Comparing the mean and median, the confidence of rating 5 is higher, meaning that they use more common, directive words.

Data Exploration

7

Price distribution



From the log-distribution plots, we can see that **tv and flashdrive** products are **dominating the sales**.

- Most of **tv** products sold are in **1 Million to 10 Million rupiah** price range
- Most of **flashdrives** are sold in the **range of 100 thousand rupiah**
- **harddrives** have two price frequency peaks, that is in the **range of 100 thousand rupiah**, and **around 1 million rupiah**
- Then, most of **laptops** are sold in the **1 million to 10 million rupiah** price range

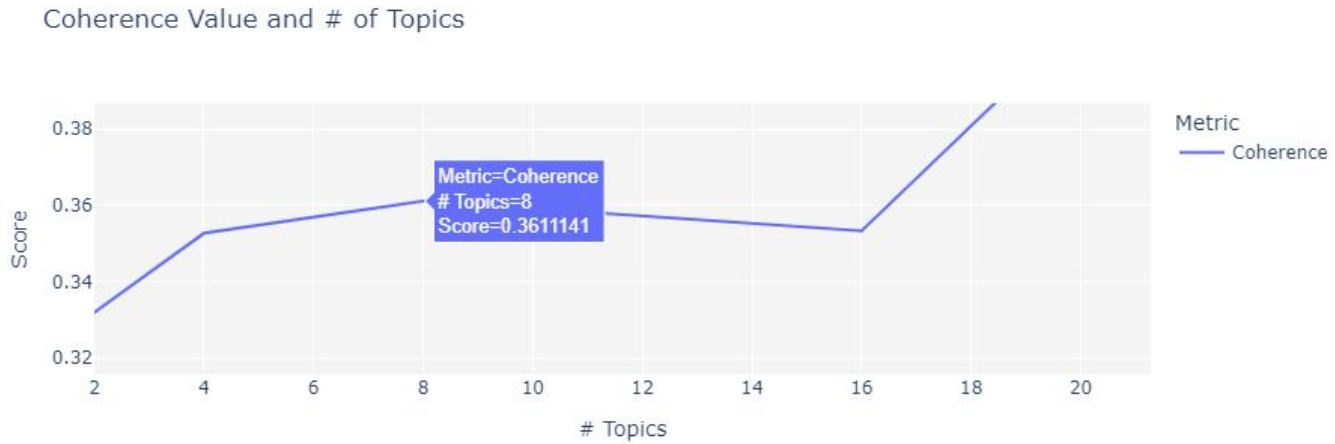
Modelling

1

Aim: predict rating from a given review

We use **LDA algorithm** to answer this aim. Latent Dirichlet Allocation or LDA is a dimension reduction algorithm that is used in topic modelling.

Using Tune Model



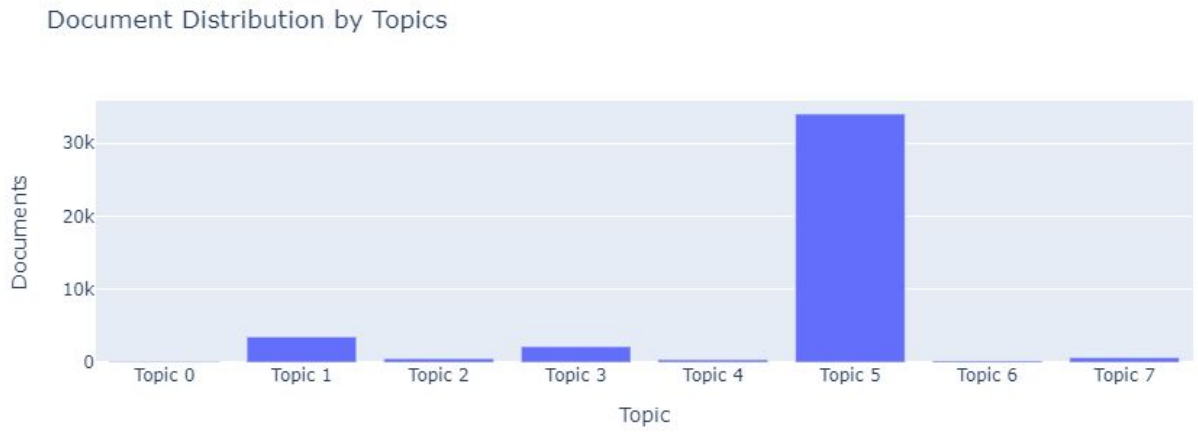
We select 8 topic model that correspond to a local maxima in coherence curve.

Modelling

1

Aim: predict rating from a given review

Topic Model



Topic 0 : disappointed, pretty, steady, still, transfer, free, ask, year, need, warranty
Topic 1 : accord, come, try, satisfied, durable, arrive, description, safe, picture, real
Topic 2 : use, quickly, even, already, ori, courier, directly, work, complete, content
Topic 3 : buy, original, capacity, file, datum, bro, sell, damage, nice, also
Topic 4 : time, long, package, condition, accept, service, delivery, problem, okay, yesterday

Topic 5 : item, fast, function, receive, neat, recommend, seller, bonus, day, hopefully
Topic 6 : great, message, order, stuff, want, disappoint, sip, process, color, immediately
Topic 7 : product, price, quality, cheap, really, easy, image, small, quite, wrong

Modelling

2

Aim : Predict rating using review with 6 topics

We use **Classification Model** for answer this aim. **Classification Model** separates the data based on a **categorical labels (classes)**.

	Description	Value
0	session_id	8538
1	Target	rating5
2	Target Type	Binary
3	Label Encoded	None
4	Original Data	(41666, 26)
5	Missing Values	True
6	Numeric Features	13
7	Categorical Features	2
8	Ordinal Features	False
9	High Cardinality Features	False
10	High Cardinality Method	None
11	Transformed Train Set	(27707, 53)
12	Transformed Test Set	(12500, 53)
13	Shuffle Train-Test	True
14	Stratify Train-Test	False
15	Fold Generator	StratifiedKFold
16	Fold Number	10
17	CPU Jobs	-1
18	Use GPU	False
19	Log Experiment	False
20	Experiment Name	clf-default-name
21	USI	420a
22	Imputation Type	simple
23	Iterative Imputation Iteration	None
24	Numeric Imputer	mean
25	Iterative Imputation Numeric Model	None
26	Categorical Imputer	constant
27	Iterative Imputation Categorical Model	None
28	Unknown Categoricals Handling	least_frequent
29	Normalize	False
30	Normalize Method	None
31	Transformation	False
32	Transformation Method	None
33	PCA	False
34	PCA Method	None
35	PCA Components	None
36	Ignore Low Variance	False
37	Combine Rare Levels	False

Modelling

2

Aim : Predict rating using review with 6 topics

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
gbc	Gradient Boosting Classifier	0.8069	0.7655	0.9680	0.8169	0.8861	0.2811	0.3325	7.7390
lightgbm	Light Gradient Boosting Machine	0.8054	0.7690	0.9579	0.8211	0.8842	0.2961	0.3350	0.6180
ada	Ada Boost Classifier	0.8033	0.7610	0.9588	0.8187	0.8832	0.2835	0.3239	1.8980
ridge	Ridge Classifier	0.8020	0.0000	0.9750	0.8090	0.8842	0.2384	0.3014	0.1090
lda	Linear Discriminant Analysis	0.8017	0.7477	0.9592	0.8171	0.8824	0.2746	0.3156	0.6210
lr	Logistic Regression	0.7950	0.7157	0.9707	0.8051	0.8802	0.2105	0.2674	1.6820
rf	Random Forest Classifier	0.7923	0.7387	0.9410	0.8184	0.8754	0.2679	0.2937	6.0890
knn	K Neighbors Classifier	0.7797	0.6802	0.9278	0.8141	0.8673	0.2352	0.2540	0.9550
et	Extra Trees Classifier	0.7768	0.7190	0.9180	0.8169	0.8645	0.2432	0.2573	5.7640
dummy	Dummy Classifier	0.7758	0.5000	1.0000	0.7758	0.8737	0.0000	0.0000	0.0900
svm	SVM - Linear Kernel	0.7243	0.0000	0.8466	0.8180	0.7910	0.1744	0.2112	0.7470
dt	Decision Tree Classifier	0.7200	0.6014	0.8138	0.8233	0.8185	0.2065	0.2066	0.6300
nb	Naive Bayes	0.7090	0.6666	0.7756	0.8373	0.8052	0.2332	0.2360	0.1190
qda	Quadratic Discriminant Analysis	0.5155	0.5000	0.5281	0.7676	0.6053	0.0102	0.0030	0.3240

Based on compare models table, we pick **light gradient boosting machine** as our **primary model** due to its **high performance** and **short running time**.

Modelling

2

Summary of the modeling process

```
classifier = create_model('lightgbm')
```

Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.8102	0.7765	0.9600	0.8242	0.8869	0.3166	0.3566
1	0.8030	0.7595	0.9539	0.8210	0.8825	0.2937	0.3289
2	0.7986	0.7651	0.9544	0.8169	0.8803	0.2696	0.3061
3	0.8196	0.7897	0.9600	0.8329	0.8920	0.3612	0.3978
4	0.8044	0.7692	0.9553	0.8216	0.8834	0.2966	0.3329
5	0.8084	0.7787	0.9656	0.8196	0.8866	0.2942	0.3418
6	0.8051	0.7707	0.9628	0.8182	0.8846	0.2838	0.3286
7	0.7960	0.7522	0.9525	0.8155	0.8787	0.2609	0.2960
8	0.8054	0.7587	0.9581	0.8210	0.8843	0.2959	0.3350
9	0.8032	0.7698	0.9563	0.8200	0.8829	0.2891	0.3267
Mean	0.8054	0.7690	0.9579	0.8211	0.8842	0.2961	0.3350
Std	0.0061	0.0104	0.0040	0.0046	0.0035	0.0261	0.0264

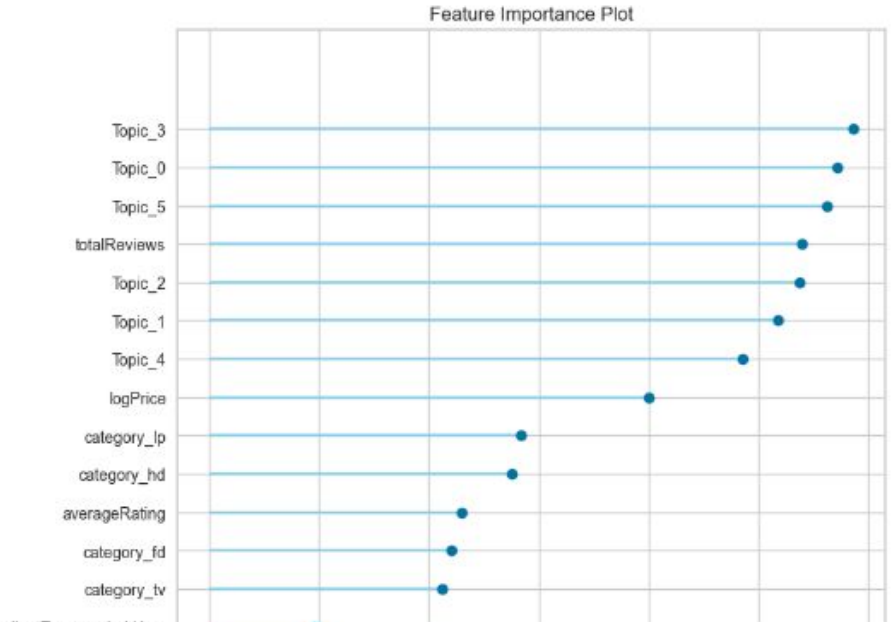
Trial 1: Using *brandName* and *clientType*, alongside the topic results from ***productName*** topic results from ***review*** (6)**, *logPrice*, *averageRating*, and *totalReview*** as a **predictor** for whether the products is **rated 5** (class 1) or **not** (class 0).

Price were **subjected to log** transformation so that the **distribution** is closer to a **normal**.

Modelling

2

Summary of the modeling process



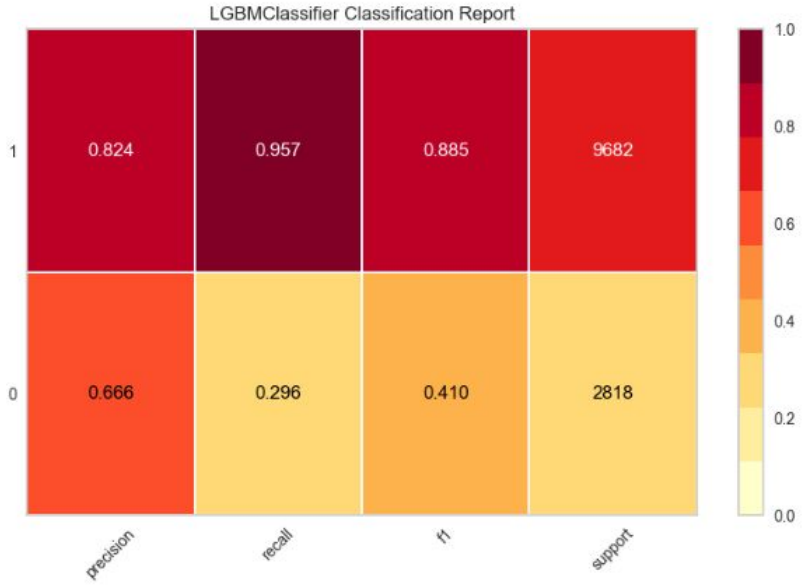
Turns out that *brandNames* and *clientTypes* were **not significant.**

Modelling

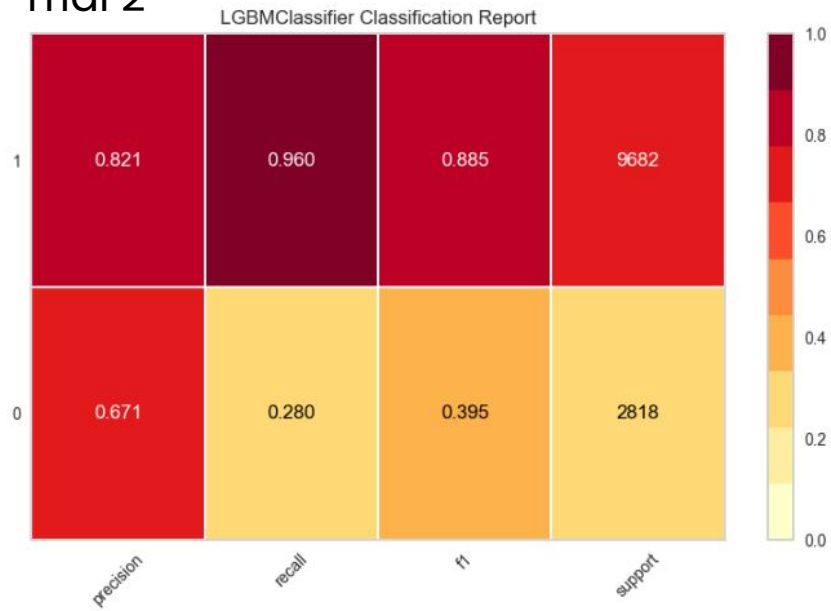
2

Summary of the modeling process

Trial 1



Trial 2



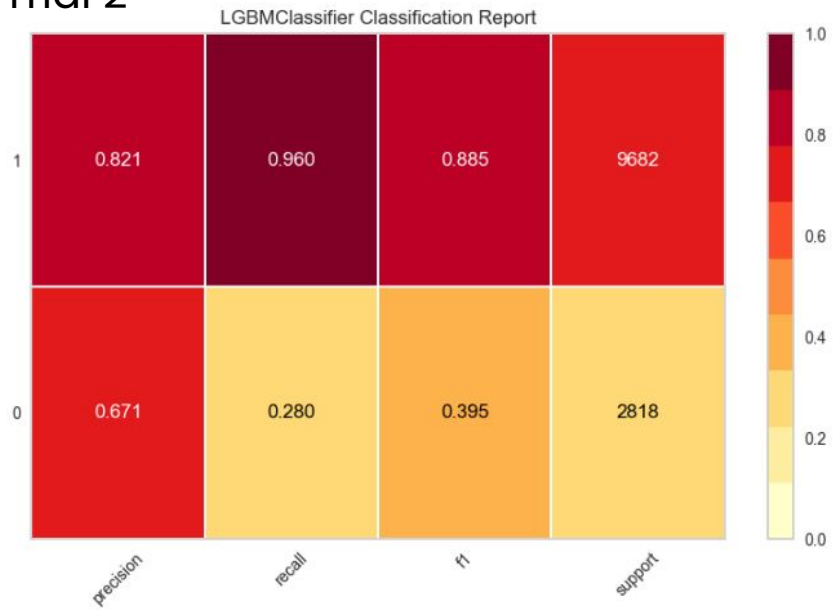
Trial 2: **Eliminate** *brandName* and *clientType*. Overall **model** performance **are maintained**. With parsimonious principle, we'll discard both columns.

Modelling

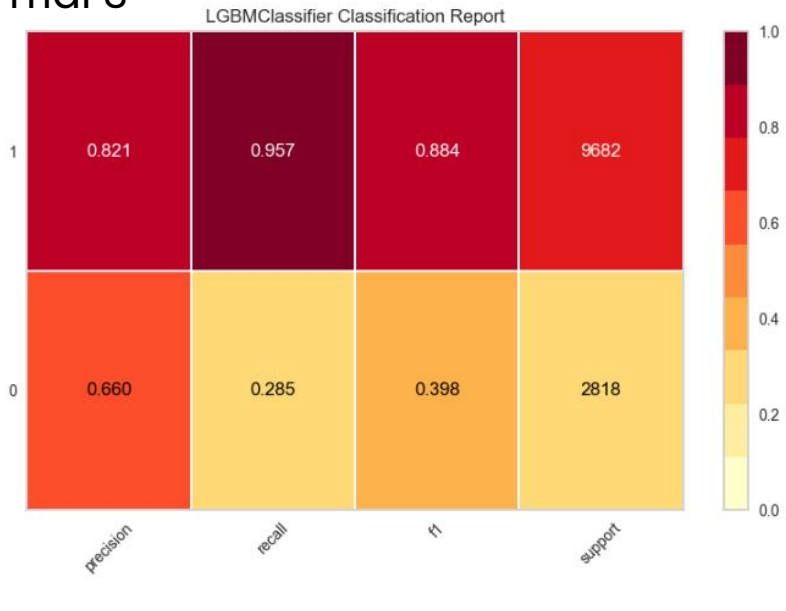
2

Summary of the modeling process

Trial 2



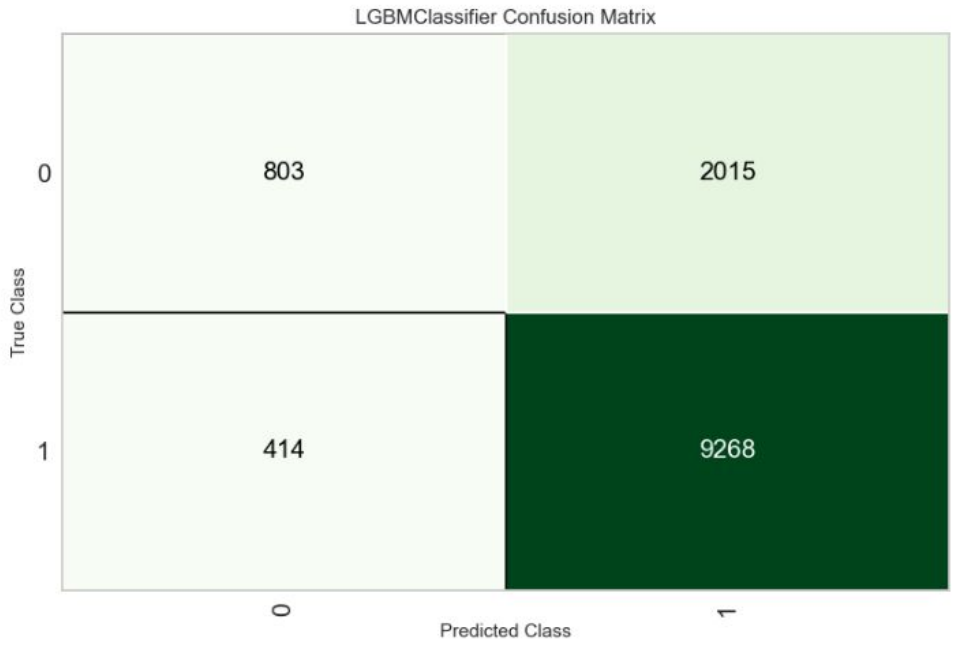
Trial 3



Trial 3: Trying to **tune the hyperparameter**. The **result isn't** significantly **better** considering the time effort. We will **forego** this **tuning** step.

Modelling

2 Summary of the modeling process



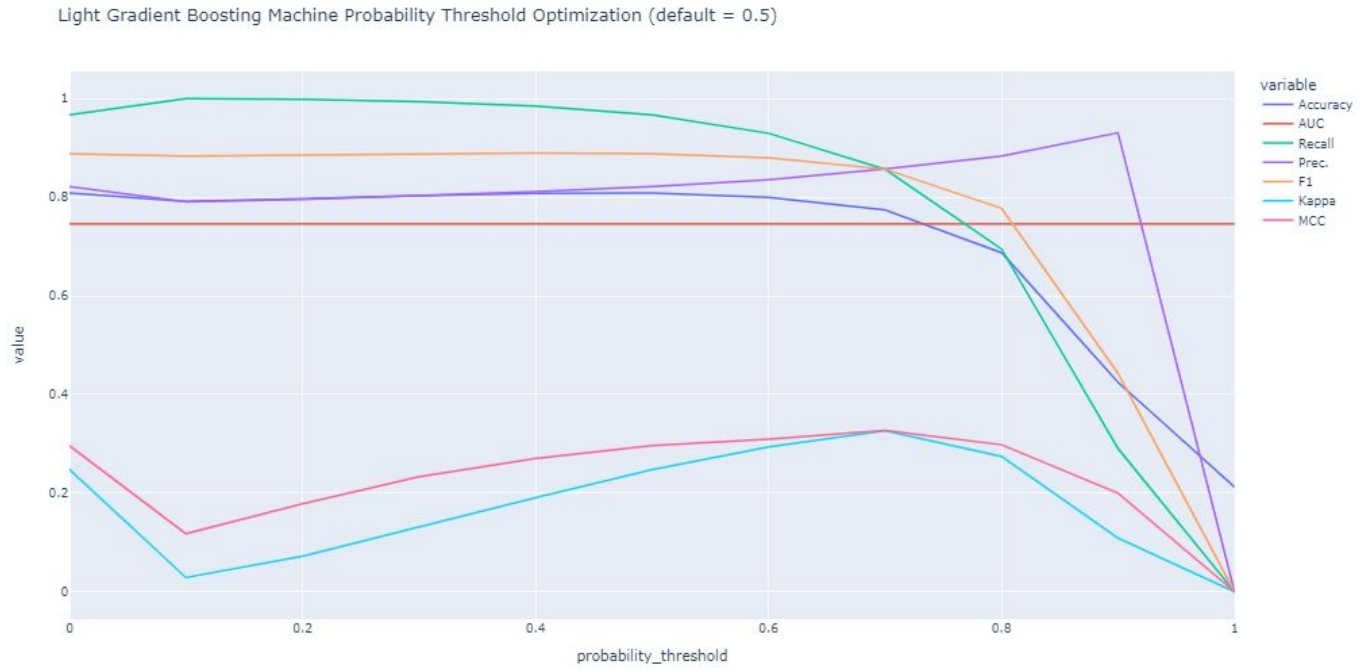
Trial 4: **Updating topic result** from review using **8 topics** that results in higher coherence metric. The **model** still have a **high false positive rate of 0.715**.

True positive (**TP**): Observation is predicted positive and is actually positive.
False positive (**FP**): Observation is predicted positive and is actually negative.
True negative (**TN**): Observation is predicted negative and is actually negative.
False negative (**FN**): Observation is predicted negative and is actually positive.

Modelling

3

Aim : Predict rating using review - Updated with 8 topics



Trial 5: **Optimizing** the **classification threshold** to **Kappa** metric, results in **lower false positive** rate with **higher false negative** rate that is still tolerable

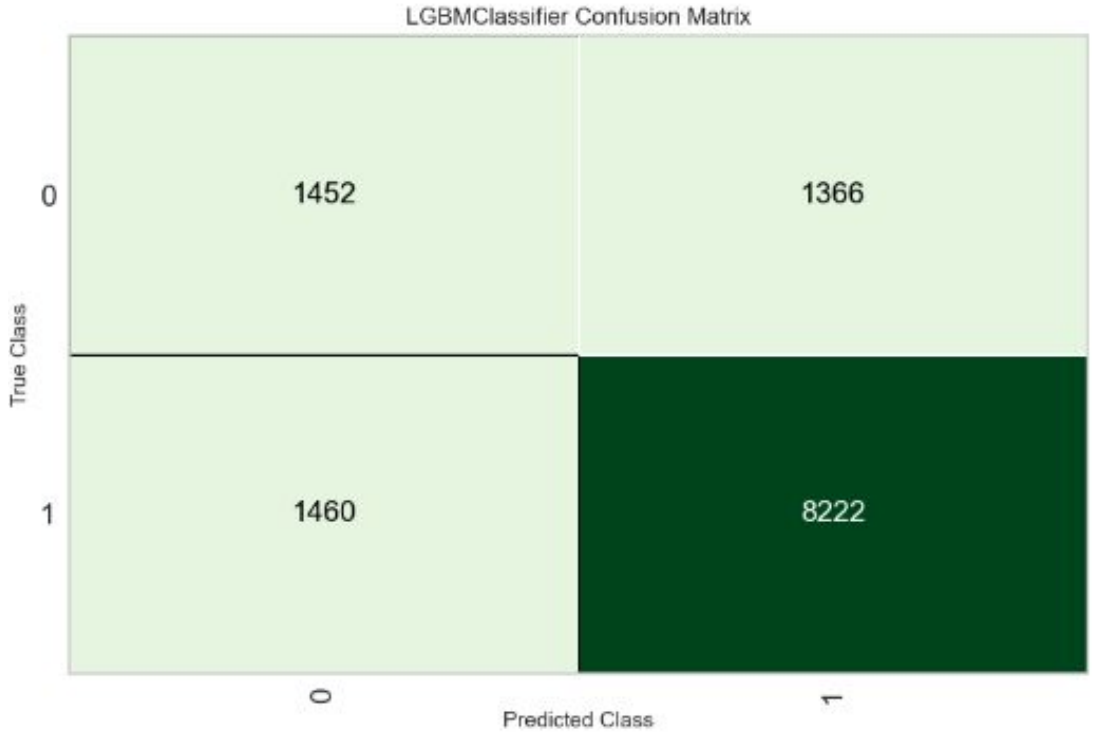
Kappa stats is the measure of accuracy of the model compared to random classifier that produce the same confusion matrix

Modelling

3

Aim : Predict rating using review - Updated with 8 topics

Classifier model with 8 topic, without brand/client, optimized for kappa metric



This Confusion Matrix shows that:

- TN = 1452
- FP = 1366
- FN = 1460
- TP = 8222

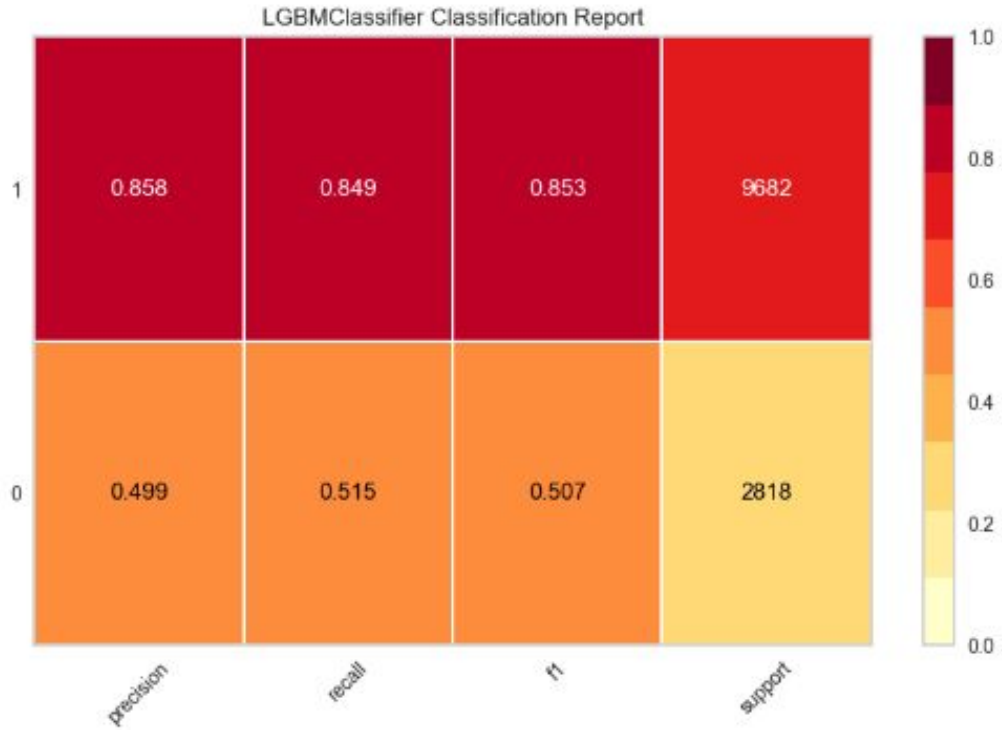
False positive rate: 0.4735

Modelling

3

Aim : Predict rating using review - Updated with 8 topics

Classifier model with 8 topic, without brand/client



The proportion of True predictions that are correct (**precision class 1**) is **0.858**, meanwhile the same metric for **class 0** is **0.499**

The proportion of class 1 that are predicted correctly (**recall class 1**) is **0.849**, meanwhile the same metric for **class 0** is **0.515**

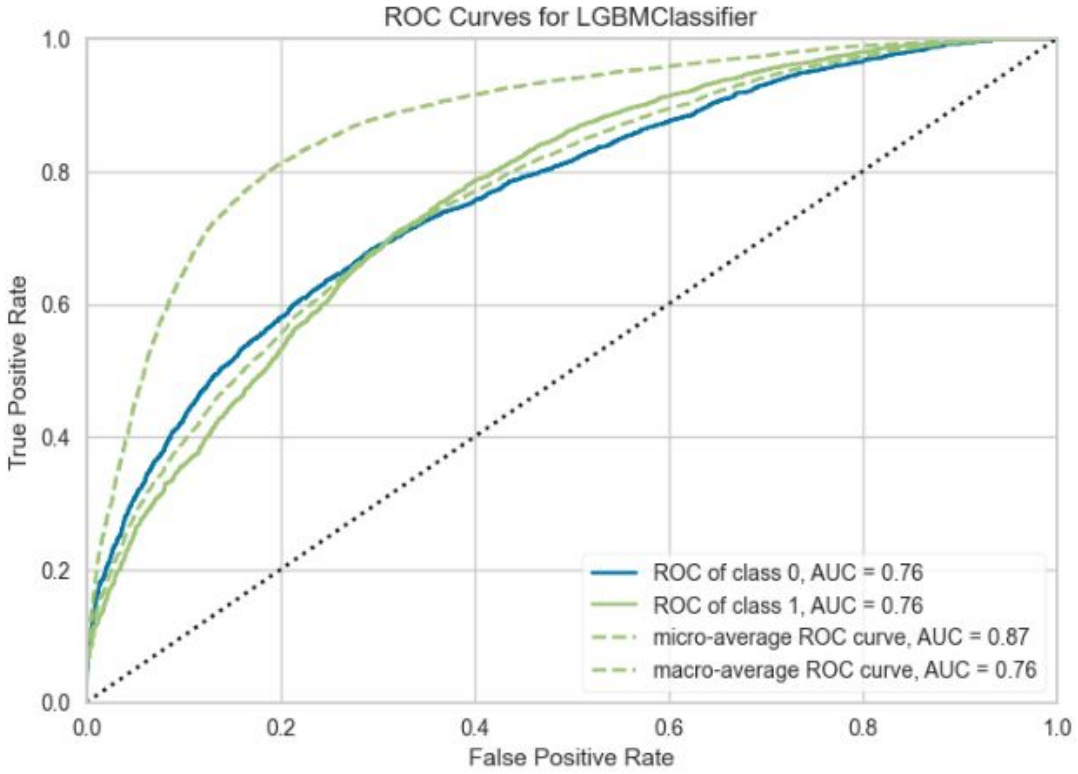
Harmonic mean of precision and recall (**F1 class 1**) is **0.853**, and the same metric for **class 0** is **0.507**

Modelling

3

Aim : Predict rating using review - Updated with 8 topics

Classifier model with 8 topic, without brand/client



From the ROC (Receiver Operating Characteristic) curve shows the effect of various probability threshold to True Positive Rate to False Positive Rate.

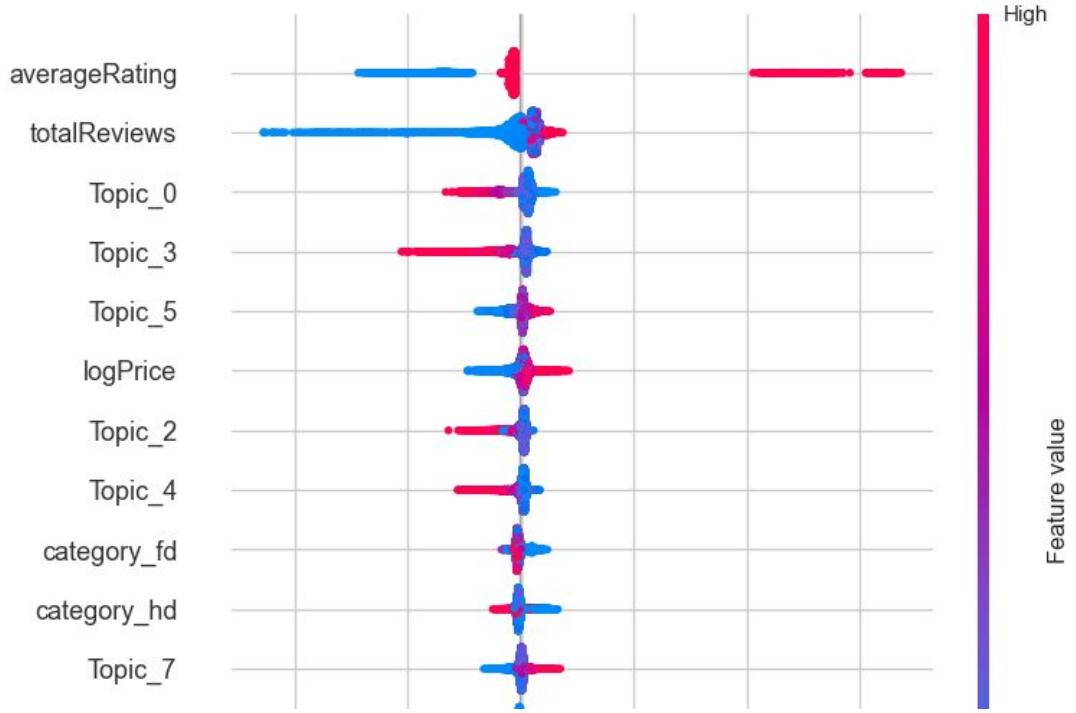
The AUC (area under the ROC Curve) is 0.76

Modelling

3

Aim : Predict rating using review - Updated with 8 topics

Classifier model with 8 topic, without brand/client



averageRating and **totalReviews** are two numerical values with high positive correlation to rating 5, along with **logPrice**.

Reviews affect the rating most by the presence of words in **Topic 0**, **Topic 3**, and **Topic 5**.
Topic 0 and Topic 3 have negative correlation, unlike Topic 5 with a positive correlation.

Modelling

3

Aim : Predict rating using review

With this, we have arrived at the model that we'll propose.

Model : Light-GBM (Kappa optimized)
Predictors : topic results from **brandName** (4 topics/categories),
topic results from **review** (8 topics),
logPrice, **averageRating**, **totalReview**
Label : rating5 (1 if rating is equal to 5, 0 else)

Below are the metrics of our final model, trained for the whole dataset.

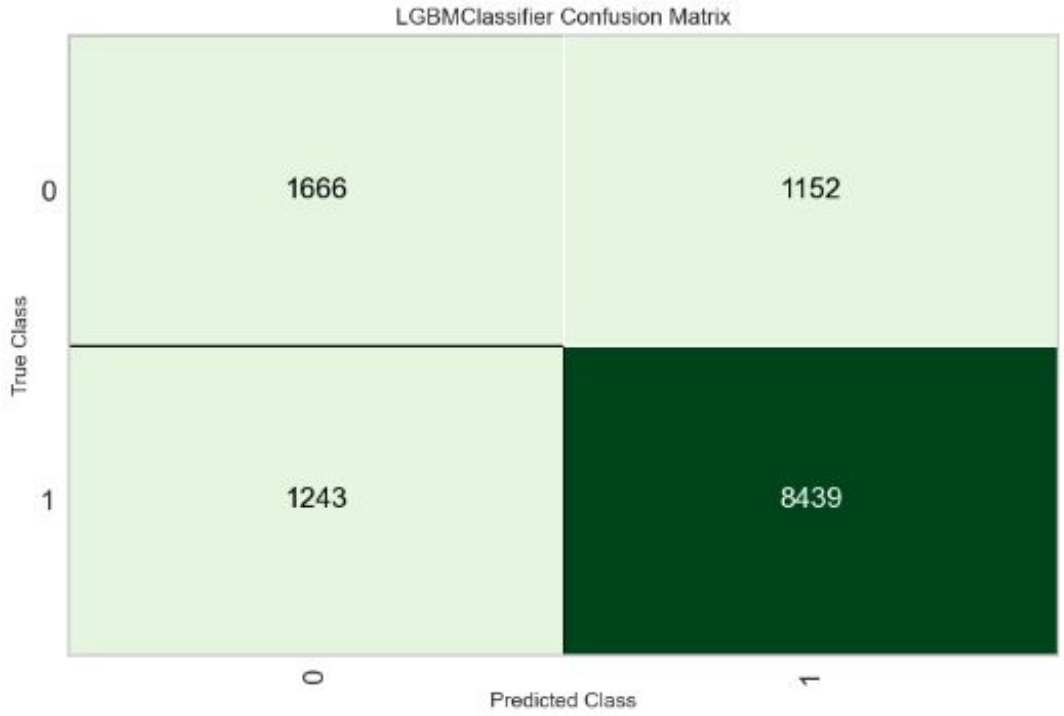
	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	Light Gradient Boosting Machine	0.8084	0.8235	0.8716	0.8799	0.8757	0.4576	0.4577

Modelling

3

Aim : Predict rating using review

After finalize model



This Confusion Matrix shows that:

- TN = 1666 data
- FP = 1152 data
- FN = 1243 data
- TP = 8439 data

Precision : 0.879
Recall : 0.871
F1-score : 0.875
False positive rate: 0.408

Business Insight

From our analysis, we conclude that :

- The overall transactions have a peak in notable dates, such as dates around April 1st, May 20th, July 15th, Sep 16th, Nov 11th, and Dec 16th, with a clear increasing trend in the recent years.
  Due to this pattern we advice to increase the availability of Lazada's resources to handle the increase in traffic.

- Products that have common and directive words tend to have a higher rating. Such words are:
  - TV products:
    Features : led, inch, android tv, usb port, vga port, wifi
    Color : black
    Brand : sharp/coocaa

Business Insight

From our analysis, we conclude that :

- Laptop products:
  Features : core, hdd, vga, amd (ryzen, radeon), processor: intel, inch, fingerprint, gaming, slim, touch
  Brand : rog, toshiba, aspire, macbook
  Color : blue
  Various promos

- Flashdisk Products:
  Features : flash drive, usb drive
  Brand : cz, toshiba, kingstone
  Color : white, silver, black

Business Insight

From our analysis, we conclude that :

- Hardisk Products:
  Features : hdd, hard disk, external drive, storage, sd card
  Accessories : pouch, softcase, cover, portable
  Brand : transcend, antishock, jetflash, toshiba, element
  Various promos

Business Insight

From our analysis, we conclude that :

- influence of certain keywords in product reviews to predict its rating
  We identify that words in Topic 0 and Topic 3 correlate negatively with review, and the reverse for Topic 5. Those words are:
  - **Topic 0** : disappointed, pretty, steady, still, transfer, free, ask, year, need, warranty
  - **Topic 3** : buy, original, capacity, file, datum, bro, sell, damage, nice, also
  - **Topic 5**: item, fast, function, receive, neat, recommend, seller, bonus, day, hopefully

This can be further applied into **alternative rating system** and **suggestive review templates**

Business Insight

From our analysis, we conclude that :

- Brands that have the most contribution in sales and transactions
  - By transaction: Transactions contributed the most is **Sandisk**, with a total of 14,150 transactions.
  - By sales: Brands that sell the most is **LG** with total sales of IDR 48.32 billion.

Further findings:

- Categories listed on most products isn't accurate, but product titles can be used to suggest the more suitable category.
- There are products that have suspicious transaction record and ridiculous prices that may have been used for fraud

Thank you!

See u on the next event ☺️