

# Kafin Ann Sulaimillah, A.Md.T

## Portofolio



## Hai, Mari berkenalan!

Hai, Perkenalkan saya Kafin Ann Sulaimillah atau bisa dipanggil Kafin. Saya merupakan lulusan D3 Teknik Telekomunikasi dari Politeknik Negeri Bandung. Saat ini, Saya sedang mendalami keahlian di bidang Data Analysis dan tertarik untuk membangun karier di bidang ini.

Melalui portofolio ini, saya ingin membagikan antusiasme saya terhadap Data Analysis. Sebelumnya saya telah memiliki beberapa sertifikat keahlian di bidang Data Analysis dan mengerjakan beberapa proyek.

Portofolio ini sebagai bukti terhadap proyek yang pernah saya lakukan sebelumnya dan kemampuan terhadap Data Analysis. Semoga dengan portofolio ini, kita dapat berdiskusi terkait kemungkinan peluang di masa yang akan datang.

Please contact me at :

Telfon (WhatsApp) : 089665268679

Email : [kafinazkiya@gmail.com](mailto:kafinazkiya@gmail.com)

Linkedin : kafinazkiyaaa

Latar Belakang

E-commerce, atau perdagangan elektronik, adalah metode berbelanja dan berbisnis secara online melalui internet. Pengguna dapat membeli produk atau layanan, melakukan transaksi, dan bertukar informasi secara elektronik. Ulasan dan penilaian dari pelanggan yang telah menggunakan produk atau layanan sangat penting dalam e-commerce. Ulasan memberikan informasi tentang kualitas dan kepuasan pengguna sebelumnya, membantu calon pembeli dalam pengambilan keputusan. Rating juga memberikan gambaran cepat tentang kualitas keseluruhan, mempengaruhi reputasi dan kesuksesan jangka panjang bisnis e-commerce. Rating dan komentar negatif dalam e-commerce memberikan umpan balik penting tentang kekurangan produk atau layanan. Hal ini membantu penjual untuk memperbaiki kualitas dan membangun kepercayaan dengan merespons dengan baik terhadap masukan negatif.

SMART Questions

S	M	A	R	T
Produk apa dan kota mana yang menjadi tujuan pada ulasan negatif ?	Bagaimana kita dapat mengukur tingkat ketidakpuasan dari ulasan dengan rating rendah?	Apakah mungkin untuk mengidentifikasi pola umum dalam bahasa atau frasa yang digunakan dalam ulasan negatif?	Apa tema umum dari ulasan negatif yang dapat dijadikan sebagai panduan untuk perbaikan produk atau layanan?	Berapa jumlah ulasan negatif yang telah diterima selama satu tahun terakhir, dan bagaimana tren keluhan ini selama satu tahun terakhir?










1. Menyiapkan semua library yang dibutuhkan

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
import string
from collections import Counter
from wordcloud import WordCloud
from sklearn.decomposition import PCA
from sklearn.decomposition import TruncatedSVD
from sklearn.pipeline import Pipeline
from sklearn.manifold import TSNE
from sklearn.preprocessing import Normalizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.cluster import KMeans
from googletrans import Translator
from pandas.tseries.offsets import MonthEnd
from datetime import datetime, timedelta
```

Data Wrangling

2. Gathering Data

Langkah pertama yang perlu dilakukan pada tahap ini adalah menyiapkan dataset yang diperlukan untuk analisis, yaitu dataset “E-Commerce Public Data Set ” yang bersumber dari <https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce>, seperti gambar di bawah ini :

Name	Date modified	Type	Size
 customers_dataset	7/23/2023 1:22 PM	Microsoft Excel Com...	8,823 KB
 geolocation_dataset	7/23/2023 1:22 PM	Microsoft Excel Com...	59,838 KB
 order_items_dataset	7/23/2023 1:22 PM	Microsoft Excel Com...	15,077 KB
 order_payments_dataset	7/23/2023 1:22 PM	Microsoft Excel Com...	5,642 KB
 order_reviews_dataset	7/23/2023 1:22 PM	Microsoft Excel Com...	14,113 KB
 orders_dataset	7/23/2023 1:22 PM	Microsoft Excel Com...	17,242 KB
 product_category_name_translation	7/23/2023 1:22 PM	Microsoft Excel Com...	3 KB
 products_dataset	7/23/2023 1:22 PM	Microsoft Excel Com...	2,324 KB
 sellers_dataset	7/23/2023 1:22 PM	Microsoft Excel Com...	171 KB

Setelah mengumpulkan seluruh data dari sumber data set, kita perlu membaca dataset tersebut menggunakan python untuk mengetahui informasi apa saja yang bisa kita dapatkan dari setiap file yang bisa kita gunakan untuk analisis dan pengambilan keputusan, karena file yang kita miliki memiliki format CSV, library pandas menyediakan fungsi untuk membaca file, seperti yang kita gunakan pada salah satu file :

```
# Membaca File CSV
customers_df = pd.read_csv(r".\E-Commerce Public Dataset\customers_dataset.csv")
# Menampilkan 5 teratas
customers_df.head()
```

✓ 4.6s

Python

Lalu dengan fungsi tersebut kita dapat mengetahui informasi setiap file :

a. customers\_df

	customer_id	customer_unique_id	customer_zip_code_prefix	customer_city	customer_state
0	06b8999e2fba1a1fbc88172c00ba8bc7	861eff4711a542e4b93843c6dd7febb0	14409	franca	SP
1	18955e83d337fd6b2def6b18a428ac77	290c77bc529b7ac935b93aa66c333dc3	9790	sao bernardo do campo	SP
2	4e7b3e00288586ebd08712 added0374a03	060e732b5b29e8181a18229c7b0b2b5e	1151	sao paulo	SP
3	b2b6027bc5c5109e529d4dc6358b12c3	259dac757896d24d7702b9acbbff3f3c	8775	mogi das cruces	SP
4	4f2d8ab171c80ec8364f7c12e35b23ad	345ecd01c38d18a9036ed96c73b8d066	13056	campinas	SP

b. geoloc\_df

	geolocation_zip_code_prefix	geolocation_lat	geolocation_lng	geolocation_city	geolocation_state
0	1037	-23.545621	-46.639292	sao paulo	SP
1	1046	-23.546081	-46.644820	sao paulo	SP
2	1046	-23.546129	-46.642951	sao paulo	SP
3	1041	-23.544392	-46.639499	sao paulo	SP
4	1035	-23.541578	-46.641607	sao paulo	SP

c. orderitem\_df

order_id	order_item_id	product_id	seller_id	shipping_limit_date	price	freight va
2cb16214	1	4244733e06e7ecb4970a6e2683c13e61	48436dade18ac8b2bce089ec2a041202	2017-09-19 09:45:35	58.90	13
a144bdd3	1	e5f2d52b802189ee658865ca93d83a8f	dd7ddc04e1b6c2c614352b388efe2d36	2017-05-03 11:05:13	239.90	19
da4fc703e	1	c777355d18b72b67abbeef9df44fd0fd	5b51032eddd242adc84c38acab88f23d	2018-01-18 14:48:30	199.00	17
38114c75	1	7634da152a4610f1595efa32f14722fc	9d7a1d34a5052409006425275ba1c2b4	2018-08-15 10:10:18	12.99	17
e55b4fd9	1	ac6c3623068f30de03045865e4e10089	df560393f3a51e74553ab94004ba5c87	2017-02-13 13:57:51	199.90	18

d. orderpayment\_df

	order_id	payment_sequential	payment_type	payment_installments	payment_value
0	b81ef226f3fe1789b1e8b2acac839d17	1	credit_card	8	99.33
1	a9810da82917af2d9aefd1278f1dcfa0	1	credit_card	1	24.39
2	25e8ea4e93396b6fa0d3dd708e76c1bd	1	credit_card	1	65.71
3	ba78997921bbcdc1373bb41e913ab953	1	credit_card	8	107.78
4	42fdf880ba16b47b59251dd489d4441a	1	credit_card	2	128.45

e. orderreviews\_df

	review_id	order_id	review_score	review_comment_title
0	7bc2406110b926393aa56f80a40eba40	73fc7af87114b39712e6da79b0a377eb	4	NaN
1	80e641a11e56f04c1ad469d5645fdfde	a548910a1c6147796b98fdf73dbeba33	5	NaN
2	228ce5500dc1d8e020d8d1322874b6f0	f9e4b658b201a9f2ecdecbb34bed034b	5	NaN
3	e64fb393e7b32834bb789ff8bb30750e	658677c97b385a9be170737859d3511b	5	NaN
4	f7c4243c7fe1938f181bec41a392bdeb	8e6bfb81e283fa7e4f11123a3fb894f1	5	NaN

review_comment_message	review_creation_date	review_answer_timestamp
NaN	2018-01-18 00:00:00	2018-01-18 21:46:59
NaN	2018-03-10 00:00:00	2018-03-11 03:05:13
NaN	2018-02-17 00:00:00	2018-02-18 14:36:24
Recebi bem antes do prazo estipulado.	2017-04-21 00:00:00	2017-04-21 22:02:06
Parabéns lojas lannister adorei comprar pela l...	2018-03-01 00:00:00	2018-03-02 10:26:53

f. orders\_df

	order_id	customer_id	order_status	order_purchase_timestamp
0	e481f51cbdc54678b7cc49136f2d6af7	9ef432eb6251297304e76186b10a928d	delivered	2017-10-02 10:56:33
1	53cdb2fc8bc7dce0b6741e2150273451	b0830fb4747a6c6d20dea0b8c802d7ef	delivered	2018-07-24 20:41:37
2	47770eb9100c2d0c44946d9cf07ec65d	41ce2a54c0b03bf3443c3d931a367089	delivered	2018-08-08 08:38:49
3	949d5b44dbf5de918fe9c16f97b45f8a	f88197465ea7920adcdbec7375364d82	delivered	2017-11-18 19:28:06
4	ad21c59c0840e6cb83a9ceb5573f8159	8ab97904e6daea8866dbdbc4fb7aad2c	delivered	2018-02-13 21:18:39

order_approved_at	order_delivered_carrier_date	order_delivered_customer_date	order_estimated_delivery_date
2017-10-02 11:07:15	2017-10-04 19:55:00	2017-10-10 21:25:13	2017-10-18 00:00:00
2018-07-26 03:24:27	2018-07-26 14:31:00	2018-08-07 15:27:45	2018-08-13 00:00:00
2018-08-08 08:55:23	2018-08-08 13:50:00	2018-08-17 18:06:29	2018-09-04 00:00:00
2017-11-18 19:45:59	2017-11-22 13:39:59	2017-12-02 00:28:42	2017-12-15 00:00:00
2018-02-13 22:20:29	2018-02-14 19:46:34	2018-02-16 18:17:02	2018-02-26 00:00:00



g. productcategory\_df

	product_category_name	product_category_name_english
0	beleza_saude	health_beauty
1	informatica_acessorios	computers_accessories
2	automotivo	auto
3	cama_mesa_banho	bed_bath_table
4	moveis_decoracao	furniture_decor

h. products\_df

	product_id	product_category_name	product_name_lenght	product_description_lenght	product_photos_qty
0	1e9e8ef04dbcff4541ed26657ea517e5	perfumaria	40.0	287.0	1.0
1	3aa071139cb16b67ca9e5dea641aaa2f	artes	44.0	276.0	1.0
2	96bd76ec8810374ed1b65e291975717f	esporte_lazer	46.0	250.0	1.0
3	cef67bcfe19066a932b7673e239eb23d	bebes	27.0	261.0	1.0
4	9dc1a7de274444849c219cff195d0b71	utilidades_domesticas	37.0	402.0	4.0

product_weight_g	product_length_cm	product_height_cm	product_width_cm
225.0	16.0	10.0	14.0
1000.0	30.0	18.0	20.0
154.0	18.0	9.0	15.0
371.0	26.0	4.0	26.0
625.0	20.0	17.0	13.0

i. sellers\_df

	seller_id	seller_zip_code_prefix	seller_city	seller_state
0	3442f8959a84dea7ee197c632cb2df15	13023	campinas	SP
1	d1b65fc7debc3361ea86b5f14c68d2e2	13844	mogi guacu	SP
2	ce3ad9de960102d0677a81f5d0bb7b2d	20031	rio de janeiro	RJ
3	c0f3eea2e14555b6faeea3dd58c1b1c3	4195	sao paulo	SP
4	51a04a8a6bdc b23deccc82b0b80742cf	12914	braganca paulista	SP

3. Assessing Data

Sebelum masuk ke tahap analisis data, pada tahap assessing ini kita harus mengidentifikasi masalah yang terdapat dalam dataset sehingga dapat memastikan data berkualitas. Dalam proses assessing data kita memeriksa dataframe satu persatu :

a. customers\_df

- Tahap awal dalam assessing data yaitu menilai informasi dataframe tersebut, library pandas memiliki fungsi yaitu .info untuk memeriksa informasi dataset :

```
customers_df.info()
✓ 0.2s

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99441 entries, 0 to 99440
Data columns (total 5 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   customer_id                          99441 non-null  object
1   customer_unique_id                   99441 non-null  object
2   customer_zip_code_prefix             99441 non-null  int64
3   customer_city                        99441 non-null  object
4   customer_state                       99441 non-null  object
dtypes: int64(1), object(4)
memory usage: 3.8+ MB
```

Informasi yang dihasilkan dari dataframe tersebut kita dapat menilai dataframe tidak memiliki **missing value** dalam dataframe-nya, dapat dilihat pada kolom Non-Null tidak ada perbedaan nilai antar kolomnya sehingga tidak perlu memeriksa nilai *missing value* lebih lanjut, lalu pada informasi ini kita dapat menilai kesesuaian tipe data, dapat dilihat setiap kolom memiliki tipe data yang sesuai dengan isi kolomnya.

- Memeriksa **Duplikasi Data**

```
print("Jumlah duplikasi: ", customers_df.duplicated().sum())
✓ 0.4s
Jumlah duplikasi: 0
```

Library pandas memiliki fungsi untuk menghitung data yang terduplikat yaitu `.duplicated().sum()`, dapat dilihat hasil perhitungan duplikasi yaitu tidak ada duplikasi dalam dataframe ini.

- Memeriksa **Inaccurate Value**  
Inaccurate value merupakan masalah yang muncul ketika nilai dalam sebuah data tidak sesuai dengan hasil observasi. Masalah ini umumnya muncul karena adanya human error atau sistem error. Untuk memeriksa *Inaccurate Value* pandas memiliki fungsi untuk menampilkan deskripsi dataframe yaitu `.describe()` :

```
customers_df.describe()
✓ 0.0s
```

	customer_zip_code_prefix
count	99441.000000
mean	35137.474583
std	29797.938996
min	1003.000000
25%	11347.000000
50%	24416.000000
75%	58900.000000
max	99990.000000

Dapat dilihat pada deskripsi diatas, tidak ada data yang tidak masuk akal dari deskripsi tersebut, sehingga nilai bisa dikatakan akurat.

b. geoloc\_df

- Memeriksa informasi dataset :

```
geoloc_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000163 entries, 0 to 1000162
Data columns (total 5 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   geolocation_zip_code_prefix          1000163 non-null  int64
1   geolocation_lat                      1000163 non-null  float64
2   geolocation_lng                      1000163 non-null  float64
3   geolocation_city                     1000163 non-null  object
4   geolocation_state                    1000163 non-null  object
dtypes: float64(2), int64(1), object(2)
memory usage: 38.2+ MB
```

Informasi yang dihasilkan dari dataframe tersebut kita dapat menilai dataframe tidak memiliki **missing value** dalam dataframe-nya, dapat dilihat pada kolom Non-Null tidak ada perbedaan nilai antar kolomnya sehingga tidak perlu memeriksa nilai *missing value* lebih lanjut, lalu pada informasi ini kita dapat menilai kesesuaian tipe data, dapat dilihat setiap kolom memiliki tipe data yang sesuai dengan isi kolomnya.

- Memeriksa **Duplikasi Data**

```
print("Jumlah duplikasi: ",geoloc_df.duplicated().sum())
Jumlah duplikasi: 261831
```

Dapat dilihat hasil perhitungan duplikasi yaitu 261831 duplikasi dalam dataframe ini.

- Memeriksa **Inaccurate Value**

```
geoloc_df.describe()
```

	geolocation_zip_code_prefix	geolocation_lat	geolocation_lng
count	1.000163e+06	1.000163e+06	1.000163e+06
mean	3.657417e+04	-2.117615e+01	-4.639054e+01
std	3.054934e+04	5.715866e+00	4.269748e+00
min	1.001000e+03	-3.660537e+01	-1.014668e+02
25%	1.107500e+04	-2.360355e+01	-4.857317e+01
50%	2.653000e+04	-2.291938e+01	-4.663788e+01
75%	6.350400e+04	-1.997962e+01	-4.376771e+01
max	9.999000e+04	4.506593e+01	1.211054e+02

Dapat dilihat pada deskripsi diatas, tidak ada data yang tidak masuk akal dari deskripsi tersebut, sehingga nilai

bisa dikatakan akurat.

c. orderreview\_df

- Memeriksa informasi dataset :

```
orderreviews_df.info()  
  
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 99224 entries, 0 to 99223  
Data columns (total 7 columns):  
#   Column                      Non-Null Count  Dtype  
---  -----  
0   review_id                   99224 non-null  object  
1   order_id                    99224 non-null  object  
2   review_score                 99224 non-null  int64  
3   review_comment_title        11568 non-null  object  
4   review_comment_message      40977 non-null  object  
5   review_creation_date         99224 non-null  object  
6   review_answer_timestamp      99224 non-null  object  
dtypes: int64(1), object(6)  
memory usage: 5.3+ MB
```

Informasi yang dihasilkan dari dataframe tersebut kita dapat menilai dataframe memiliki **missing value** dalam dataframe-nya, dapat dilihat pada kolom review\_comment\_title dan review\_comment\_message memiliki nilai Non-Null yang berbeda dengan kolom lainnya, perlu diperiksa nilai *missing value* lebih lanjut, lalu pada informasi ini kita dapat menilai **kesesuaian tipe data**, dapat dilihat setiap kolom memiliki tipe data yang sesuai dengan ada yang tidak sesuai dengan isinya, seperti kolom review\_creation\_date dan review\_answer\_timestamp harusnya memiliki tipe data *datetime* bukan *object*.

- Memeriksa Jumlah **Missing Value**

Untuk menghitung jumlah *missing value* lebih lanjut, Library pandas memiliki fungsi yaitu .isna().sum() untuk menghitung *missing value* yang terdapat pada suatu dataset, seperti :

```
orderreviews_df.isna().sum()  
  
review_id          0  
order_id           0  
review_score        0  
review_comment_title    87656  
review_comment_message  58247  
review_creation_date    0  
review_answer_timestamp  0  
dtype: int64
```

Dapat dilihat pada deskripsi table diatas kolom review\_comment\_title memiliki missing value 87656 dan review\_comment\_message memiliki missing value 58247.

- Memeriksa **Duplikasi Data** dan **Inaccurate Value**

```
print("Jumlah duplikasi: ", orderreviews_df.duplicated().sum())  
orderreviews_df.describe()  
  
Jumlah duplikasi: 0  
  
   review_score  
count  99224.000000  
mean      4.086421  
std       1.347579  
min       1.000000  
25%       4.000000  
50%       5.000000  
75%       5.000000  
max       5.000000
```

dapat dilihat hasil perhitungan duplikasi yaitu tidak ada duplikasi dalam dataframe ini dan dapat dilihat pada deskripsi diatas, tidak ada data yang tidak masuk akal dari deskripsi tersebut, sehingga nilai bisa dikatakan akurat.

Lalu assessing data dilakukan selanjutnya hingga ke dataframe terakhir, dan berikut hasil assessing data seluruh dataset :

No	Dataframe	Missing Value	Kesesuaian Datatype	Duplikasi Data	Inaccurate Value
1	customers_df	-	-	-	-
2	geoloc_df	-	-	261831	-
3	orderitem_df	-	√	-	-



4	orderpayment_df	-	-	-	-
5	orderreview_df	√	√	-	-
6	orders_df	√	√	-	-
7	productcategory_df	-	-	-	-
8	products_df	√	-	-	-
9	sellers_df	-	-	-	-

#### 4. Cleaning Data

Setelah menemukan kesalahan kesalahan pada dataset ditahap sebelumnya, pada tahap *cleaning* atau pembersihan ini kita membersihkan atau membenarkan kesalahan yang ditemukan :

a. geoloc\_df

Pada dataframe ini kita menemukan duplikasi data, untuk menghapus duplikasi data Library Pandas memiliki fungsi `.drop_duplicates`, seperti :

```
geoloc_df.drop_duplicates(inplace=True)

print("Jumlah duplikasi: ", geoloc_df.duplicated().sum())

Jumlah duplikasi: 0
```

Dapat dilihat pada gambar diatas setelah melaukan *drop duplicates* duplikasi pada dataframe geoloc\_df tidak ada.

b. orderitem\_df

Pada dataframe ini kita menemukan ketidaksesuaian tipe data pada dataframe ini, Library pandas memiliki fungsi untuk mengubah tipe data pada kasus ini kita harus merubah tipe data ke *datetime* sehingga fungsi yang digunakan yaitu `.to_datetime` seperti :

```
datetime_columns = ["shipping_limit_date"]

for column in datetime_columns:
    orderitem_df[column] = pd.to_datetime(orderitem_df[column])

orderitem_df.info()
✓ 1.5s

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 112650 entries, 0 to 112649
Data columns (total 7 columns):
#   Column                Non-Null Count  Dtype
---  -
0   order_id              112650 non-null object
1   order_item_id         112650 non-null int64
2   product_id           112650 non-null object
3   seller_id             112650 non-null object
4   shipping_limit_date   112650 non-null datetime64[ns]
5   price                 112650 non-null float64
6   freight_value         112650 non-null float64
dtypes: datetime64[ns](1), float64(2), int64(1), object(3)
```

Dapat dilihat dari gambar diatas, setelah melakukan perubahan tipe data, tipe data pada kolom `shipping_limit_date` berubah menjadi `datetime`.

c. orderreview\_df

- Missing Value

Untuk mengatasi missing value ada beberapa cara untuk mengatasinya, dalam kasus dataframe ini data memiliki jenis data kualitatif yaitu komentar (review) yang dikelompokan secara kuantitatif atau dengan rating, cara yang akan dipakai dalam dataframe ini yaitu *imputation* atau mengisi data dengan nilai tertentu seperti nilai tertinggi, namun untuk menghindari sampling bias atau ketika sample tidak mewakili populasi secara keseluruhan, misalkan ketika nilai tertinggi (mode) dari data merupakan komentar positif sementara data yang harus diisi bukan hanya pada range rating tinggi namun rating rendah juga sehingga akan terjadi sampling bias dalam rating negatif namun ada komentar positif didalamnya, untuk menjaga kredibilitas data kita menggunakan teknik multiple imputation, yaitu dengan :

- Membagi dataframe menjadi 5 bagian sesuai masing masing rating :

```
import pandas as pd

# Membagi dataset berdasarkan review score
review_score_1 = orderreviews_df[orderreviews_df['review_score'] == 1]
review_score_2 = orderreviews_df[orderreviews_df['review_score'] == 2]
review_score_3 = orderreviews_df[orderreviews_df['review_score'] == 3]
review_score_4 = orderreviews_df[orderreviews_df['review_score'] == 4]
review_score_5 = orderreviews_df[orderreviews_df['review_score'] == 5]
```

- Lakukan proses cleaning data terhadap setiap dataframe, dimulai dari dataframe review\_score\_1 :
  - Menghitung missing value pada dataframe review\_score\_1 :

```
review_score_1.isna().sum()

[ ] Python

... review_id          0
    order_id          0
    review_score       0
    review_comment_title  9551
    review_comment_message  2679
    review_creation_date  0
    review_answer_timestamp  0
    dtype: int64
```

Dapat dilihat pada kolom review\_comment\_title ada 9551 missing value, sehingga dari missing value itu akan diisi dengan nilai data yang sering muncul pada kolom tersebut.

- Untuk melihat isi data pada kolom dan jumlahnya, dengan menggunakan library pandas kita dapat menggunakan fungsi .value\_counts :

```
review_score_1.review_comment_title.value_counts()

review_comment_title
Não recomendo          44
Ruim                   37
não recomendo          34
Não recebi o produto   30
Produto errado         30
..
Não recebi ainda        1
Comprei dois filtros...  1
Irritante               1
nao funciona telefones  1
Empres não confiável    1
Name: count, Length: 1217, dtype: int64
```

Dapat dilihat pada dari gambar diatas, data yang sering muncul yaitu Não recomendo dengan 44 data, nilai ini akan dijadikan nilai untuk mengisi kolom yang kosong.

- Untuk mengisi kolom yang kosong dengan data yang sebelumnya didapatkan, kita memiliki fungsi .fillna:

```
review_score_1['review_comment_title'].fillna(value='Não recomendo', inplace=True)

✓ 0.0s
```

Setelah kolom berhasil diisi, periksa kembali apakah pada kolom review\_comment\_title masih ada missing value :

```
review_score_1.isna().sum()

✓ 0.0s

review_id          0
order_id          0
review_score       0
review_comment_title  0
review_comment_message  2679
review_creation_date  0
review_answer_timestamp  0
dtype: int64
```

Missing value pada kolom review\_comment\_title berhasil di hilangkan.

- Dapat dilihat pada gambar terakhir kolom review\_comment\_message memiliki missing value sebanyak 2679 kolom yang tidak memiliki isi, seperti proses sebelumnya missing value akan diisi dengan nilai data yang sering muncul pada kolom tersebut.Untuk melihat isi data pada kolom dan jumlahnya, dengan menggunakan library pandas kita dapat menggunakan fungsi .value\_counts :

```
review_score_1.review_comment_message.value_counts()

✓ 0.0s
```

Dan menghasilkan data :

review_comment_message	Não recebi produto
------------------------	--------------------

30	1
Não recebi o produto	Boa noite. Preciso de uma posição da baratheon sobre o Meu pedido. Ou o estorno no meu cartão ja que o produto não foi entregue.
12	1
Não recebi	Não recebi tudo que comprei. Já enviei reclamação para o stark, mas ainda não recebi nenhum retorno
11	1
Ainda não recebi	Eu fiz o cancelamento desse pedido no dia 27/02, até hoje não recebi uma resposta sobre.\r\nNão recebi o produto e não foi estornado o meu cartão \r\nEstou muito insatisfeita.
10	1
Ainda não recebi o produto	meu produto chegou e ja tenho que devolver, pois está com defeito , não segurar carga
10	1
..	
Um dos produtos veio com o lacre violado, vazado, as embalagens muito toda suja. \r\n\r\nNão recomendo...	

Dapat dilihat pada dari gambar diatas, data yang sering muncul yaitu Não recebi produto dengan 30 data, nilai ini akan dijadikan nilai untuk mengisi kolom yang kosong.

5. Mengisi nilai yang kosong dengan data yang telah didapatkan sebelumnya:

```
review_score_1['review_comment_message'].fillna(value='Não recebi o produto', inplace=True)
✓ 0.0s
```

Setelah kolom berhasil diisi, periksa kembali apakah pada kolom review\_comment\_message masih ada missing value :

```
review_score_1.isna().sum()
✓ 0.0s

.. review_id      0
   order_id      0
   review_score   0
   review_comment_title  0
   review_comment_message  0
   review_creation_date  0
   review_answer_timestamp  0
   dtype: int64
```

Dan dapat dilihat di gambar atas dataframe review\_score\_1 sudah tidak ada missing value, proses-proses diatas diatas dilakukan sama terhadap dataframe yang belum dilakukan cleaning data yaitu review\_score\_2 hingga 5.

• Datatype

Pada dataframe ini kita menemukan ketidaksesuaian tipe data pada dataframe ini, Library pandas memiliki fungsi untuk mengubah tipe data pada kasus ini kita harus merubah tipe data ke *datetime* sehingga fungsi yang digunakan yaitu `.to_datetime` seperti :

```
datetime_columns = ["review_creation_date", "review_answer_timestamp"]

for column in datetime_columns:
    review_score_1[column] = pd.to_datetime(review_score_1[column])

for column in datetime_columns:
    review_score_2[column] = pd.to_datetime(review_score_2[column])

for column in datetime_columns:
    review_score_3[column] = pd.to_datetime(review_score_3[column])

for column in datetime_columns:
    review_score_4[column] = pd.to_datetime(review_score_4[column])

for column in datetime_columns:
    review_score_5[column] = pd.to_datetime(review_score_5[column])
```

Lalu setelah berhasil, di cek kembali tipe data ke 5 dataframe ini :

```
review_score_1.info()
review_score_2.info()
review_score_3.info()
review_score_4.info()
review_score_5.info()
✓ 0.3s

<class 'pandas.core.frame.DataFrame'>
Index: 11424 entries, 5 to 99223
Data columns (total 7 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   review_id                             11424 non-null  object
1   order_id                              11424 non-null  object
2   review_score                           11424 non-null  int64
3   review_comment_title                   11424 non-null  object
4   review_comment_message                 11424 non-null  object
5   review_creation_date                   11424 non-null  datetime64[ns]
6   review_answer_timestamp                 11424 non-null  datetime64[ns]
dtypes: datetime64[ns](2), int64(1), object(4)
```

Dan kolom review\_creation\_date dan review\_answer\_timestamp sudah berubah ke tipe data datetime.

d. orders\_df

- Missing Value

Pada dataframe ini ditemukan missing value pada kolom :

```
orders_df.isna().sum()
✓ 0.1s Python
```

order_id	0
customer_id	0
order_status	0
order_purchase_timestamp	0
order_approved_at	160
order_delivered_carrier_date	1783
order_delivered_customer_date	2965
order_estimated_delivery_date	0

dtype: int64

Kolom order\_approved\_at, order\_delivered\_carrier\_date, order\_delivered\_customer\_date memiliki tipe data datetime yaitu termasuk kategori data continue, sehingga jika dilakukan imputation secara langsung tidak khawatir terjadi sampling bias.

1. Melihat data yang sering muncul pada kolom order\_approve\_at :

```
orders_df.order_approved_at.value_counts()
136] ✓ 0.2s
```

```
order_approved_at
2018-02-27 04:31:10    9
2017-11-07 07:30:38    7
2018-02-27 04:31:01    7
2018-02-06 05:31:52    7
2017-11-07 07:30:29    7
..
2018-08-22 11:50:14    1
2017-09-22 11:27:36    1
2018-03-07 16:40:32    1
2017-08-08 10:50:15    1
2018-03-09 11:20:28    1
Name: count, Length: 90733, dtype: int64
```

Dapat dilihat data yang sering muncul yaitu 2018-02-27 04:31:10 sebanyak 9 kali akan dijadikan nilai acuan untuk mengisi kolom yang kosong.

2. Mengisi kolom yang kosong dengan nilai yang sudah didapatkan sebelumnya :

```
orders_df['order_approved_at'].fillna(value="2018-02-27 04:31:10", inplace=True)
✓ 0.0s
```

+ Code + Markdown

Lakukan hal yang sama dengan kolom order\_delivered\_carrier\_date dan order\_delivered\_customer\_date , lalu cek jumlah missing value pada dataframe :

```
orders_df.isna().sum()
✓ 0.0s
```

order_id	0
customer_id	0
order_status	0
order_purchase_timestamp	0
order_approved_at	0
order_delivered_carrier_date	0
order_delivered_customer_date	0
order_estimated_delivery_date	0

dtype: int64

Pembersihan missing value pada dataframe ini berhasil.

- Datatype

Pada dataframe ini kita menemukan ketidaksesuaian tipe data pada dataframe ini, Library pandas memiliki fungsi untuk mengubah tipe data pada kasus ini kita harus merubah tipe data pada kolom order\_purchase\_timestamp,order\_approved\_at,order\_delivered\_carrier\_date,order\_delivered\_customer\_d ate,order\_estimated\_delivery\_time ke *datetime* sehingga fungsi yang digunakan yaitu .to\_datetime seperti:

```
datetime_columns = ["order_purchase_timestamp", "order_approved_at", "order_delivered_carrier_date",
"order_delivered_customer_date", "order_estimated_delivery_date"]
for column in datetime_columns:
    orders_df[column]= pd.to_datetime(orders_df[column])
```

lalu cek tipe data :

```
orders_df.info()
[141] ✓ 0.0s

... <class 'pandas.core.frame.DataFrame'>
RangeIndex: 99441 entries, 0 to 99440
Data columns (total 8 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   order_id                             99441 non-null  object
1   customer_id                         99441 non-null  object
2   order_status                         99441 non-null  object
3   order_purchase_timestamp            99441 non-null  datetime64[ns]
4   order_approved_at                   99441 non-null  datetime64[ns]
5   order_delivered_carrier_date        99441 non-null  datetime64[ns]
6   order_delivered_customer_date       99441 non-null  datetime64[ns]
7   order_estimated_delivery_date       99441 non-null  datetime64[ns]
dtypes: datetime64[ns](5), object(3)
```

Dan kolom order\_purchase\_timestamp, order\_approved\_at, order\_delivered\_carrier\_date, order\_delivered\_customer\_date, order\_estimated\_delivery\_time sudah berubah ke tipe data datetime.

e. products\_df

Pada dataframe ini ditemukan juga kesalahan missing value, selain melakukan teknik imputation missing value bisa diatasi dengan teknik drop atau menghapus semua kolom yang tidak memiliki data atau missing value, pada dataframe ini dilakukan teknik drop dikarenakan kolom yang berisi missing value tidak berperan penting dalam menjawab pertanyaan bisnis :

```
products_df.isna().sum()
✓ 0.0s

product_id                0
product_category_name    610
product_name_lenght      610
product_description_lenght 610
product_photos_qty       610
product_weight_g          2
product_length_cm         2
product_height_cm         2
product_width_cm          2
dtype: int64
```

Sehingga dilakukan teknik .dropna, dan missing value berhasil teratasi :

```
products_df.dropna(inplace=True)
✓ 0.0s

products_df.isna().sum()
✓ 0.0s

product_id                0
product_category_name      0
product_name_lenght        0
product_description_lenght  0
product_photos_qty         0
product_weight_g           0
product_length_cm          0
product_height_cm          0
product_width_cm           0
dtype: int64
```

Eksplorasi Data (*Exploratory Data Analysis*)

Setelah data siap digunakan saat nya untuk eksplorasi data untuk mencari jawaban terbaik dari SMART Questions yang sudah kita susun sebelumnya, pada analisis data kali ini kita berfokus kepada review negatif:

- 1. Mengerucutkan dataframe orderreview\_df hanya rating negatif  
Karena pada saat cleaning data, dataframe orderreview\_df sudah dipisahkan sesuai ratingnya, kali ini kita hanya perlu menyatukan kembali dataframe dengan rating rendah yaitu 1 dan 2, penyatuan dataframe yang memiliki struktur kolom yang sama dapat langsung dilakukan dengan fungsi .concat yang dimiliki oleh library pandas :



```
import pandas as pd

# Gabungkan berdasarkan baris
orderreviews_lowrate_df = pd.concat([review_score_1, review_score_2], axis=0)

# Reset indeks jika diperlukan
orderreviews_lowrate_df.reset_index(drop=True, inplace=True)
orderreviews_lowrate_df.head()
```

✓ 2.8s Python

	review_id	order_id	review_score	review_comment_title	review_comment_mess
0	15197aa66ff4d0650b5434f1b46cda19	b18dcdcf73be66366873cd26c5724d1dc	1	Não recomendo	Não recebi o proc
1	373cbeecea8286a2b66c97b1b157ec46	583174fbe37d3d5f0d6661be3aad1786	1	Não chegou meu produto	Péss
2	2c5e27fc178bde7ac173c9c62c31b070	0ce9a24111d850192a933fcaab6fbad3	1	Não recomendo	Não gostei ! Comprei q por k
3	58044bca115705a48fe0e00a21390c54	68e55ca79d04a79f20d4bfc0146f4b66	1	Não recomendo	Sempre compro   Internet e a enti

2. Melakukan data preprocessing

Untuk mencari lebih detail mengenai ulasan negative, tentunya kita harus mengetahui isi dari kalimat kalimat ulasan yang dikirimkan oleh customer, untuk memproses kalimat per kalimat dalam jumlah yang banyak, kita perlu melakukan data preprocessing sehingga data berubah menjadi format yang lebih bersih dan terstruktur, sehingga mudah untuk dianalisis selanjutnya, tahapan data preprocessing yang dilakukan kali ini yaitu :

```
# Lowercasing
orderreviews_lowrate_df['review_comment_message'] = orderreviews_lowrate_df['review_comment_message'].str.lower()

# Pembersihan teks
def clean_text(text):
    # Hapus tanda baca
    text = text.translate(str.maketrans('', '', string.punctuation))
    return text
orderreviews_lowrate_df['review_comment_message'] = orderreviews_lowrate_df['review_comment_message'].apply(clean_text)

# Tokenisasi
orderreviews_lowrate_df['tokens'] = orderreviews_lowrate_df['review_comment_message'].apply(word_tokenize)

# Stopword Removal
# Menggunakan stopwords bahasa Portugis
stop_words_portuguese = set(stopwords.words('portuguese'))
# Menghapus stopwords bahasa Portugis
orderreviews_lowrate_df['tokens'] = orderreviews_lowrate_df['tokens'].apply(lambda tokens: [word for word in tokens if word not in stop_words_portuguese])

# Stemming
ps = PorterStemmer()
orderreviews_lowrate_df['stemmed_tokens'] = orderreviews_lowrate_df['tokens'].apply(lambda tokens: [ps.stem(word.lower()) for word in tokens])

# Tampilkan hasil
orderreviews_lowrate_df.head()
```

- a. Lowercasing  
Untuk menghindari jika dua kata yang sama dianggap berbeda karena perbedaan huruf besar dan kecil, sehingga semua huruf dalam review\_comment\_message diubah menjadi huruf kecil menggunakan fungsi .str yang disediakan oleh library pandas.
  - b. Pembersihan Text  
Untuk membersihkan teks ulasan dari karakter-karakter tanda baca yang tidak relevan atau tidak diperlukan text dibersihkan dengan cara menghapus tanda baca, teks menjadi lebih bersih dan lebih mudah untuk diproses dan dianalisis menggunakan fungsi clean\_text dan str.maketrans yang disediakan oleh library pandas.
  - c. Tokenisasi  
Untuk memisahkan teks menjadi kata-kata sehingga kita dapat menganalisis setiap kata secara terpisah, proses kali ini menggunakan fungsi word\_tokenize yang disediakan oleh library NLTK.
  - d. Stopword Removal  
untuk menghilangkan kata-kata yang umum dan sering muncul dalam teks, tetapi memiliki sedikit kontribusi makna dalam analisis sehingga fokus analisis lebih tertuju pada kata-kata yang memiliki makna penting dan kontribusi yang lebih besar terhadap interpretasi dan analisis teks. Stopword removal digunakan menggunakan modul stopwords dari library NLTK.
  - e. Stemming  
Stemming dilakukan untuk mengubah kata kata dalam token menjadi betuk dasar kata, seperti berlari menjadi lari, hal ini dilakukan agar membantu dalam analisis dengan mengelompokkan kata-kata yang memiliki dasar kata yang sama meskipun dengan bentuk yang berbeda.
- Berikut hasil Data Preposessing :

	review_id	order_id	review_score	review_comment_title	review_comment_message	review_creation_date	review_answer_timestamp	tokens	stemmed_tokens
0	15197aa66ff4d0650b5434f1b46cda19	b18dcdf73be66366873cd26c5724d1dc	1	Não recomendo	não recebi o produto	2018-04-13	2018-04-16 00:39:37	[recebi, produto]	[recebi, produto]
1	373cbeecea8286a2b66c97b1b157ec46	583174fbe37d3d5f0d6661be3aad1786	1	Não chegou meu produto	péssimo	2018-08-15	2018-08-15 04:10:37	[péssimo]	[péssimo]
2	2c5e27fc178bde7ac173c9c62c31b070	0ce9a24111d850192a933fcaab6fbad3	1	Não recomendo	não gostei comprei gato por lebre	2017-12-13	2017-12-16 07:14:07	[gostei, comprei, gato, lebre]	[gostei, comprei, gato, lebr]
3	58044bca115705a48fe0e00a21390c54	68e55ca79d04a79f20d4bfc0146f4b66	1	Não recomendo	sempre compro pela internet e a entrega ocorre...	2018-04-08	2018-04-09 12:22:39	[sempre, compro, internet, entrega, ocorre, an...]	[sempr, compro, internet, entrega, ocorr, ant,...]
4	9fd59cd04b42f600df9f25e54082a8d1	3c314f50bc654f3c4e317b055681dff9	1	Não recomendo	nada de chegar o meu pedido	2017-04-21	2017-04-23 05:37:03	[nada, chegar, pedido]	[nada, chegar, pedido]

### 3. N-gram Analysis

Untuk menjawab pertanyaan bagian **Achievable** kita dapat mengetahui pola umum dalam frasa/kata yang terdapat review\_comment\_message negative dengan menggunakan N-gram analysis, kita dapat melakukan penghitungan munculnya urutan kata atau karakter yang terdiri dari n elemen berturut-turut (N-gram), dalam analisis data ini elemen yang kita hitung yaitu N=3, berikut kode yang digunakan untuk menganalisis trigram :

```
# Fungsi untuk mendapatkan trigram dari sebuah teks
def extract_trigrams(text):
    words = text.split()
    return [f"{words[i]} {words[i+1]} {words[i+2]}" for i in range(len(words) - 2)]
# Menggabungkan semua teks dalam kolom review_comment_message
all_comments = ' '.join(orderreviews_lowrate_df['review_comment_message'].dropna())
# Ekstraksi trigram dari teks panjang
all_trigrams = extract_trigrams(all_comments)
# Menghitung jumlah munculnya masing-masing trigram
trigram_counts = Counter(all_trigrams)
# Menghitung total trigram dalam dataset
total_trigrams = len(all_trigrams)
# Menghitung persentase masing-masing trigram
trigram_percentages = {trigram: (count / total_trigrams) * 100 for trigram, count in trigram_counts.items()}
# Mengurutkan trigram berdasarkan frekuensi tertinggi
sorted_trigrams = dict(sorted(trigram_counts.items(), key=lambda item: item[1], reverse=True))
# Menampilkan 10 trigram teratas
top_10_trigrams = dict(list(sorted_trigrams.items())[:10])
# Initialize the translator
translator = Translator()
# Fungsi untuk menerjemahkan trigram ke bahasa Indonesia
def translate_to_indonesian(trigram):
    try:
        # Translate trigram to Indonesian
        translated = translator.translate(trigram, src='pt', dest='id')
        return translated.text
    except Exception as e:
        print(f"Translation failed for {trigram}: {str(e)}")
        return trigram
# Menampilkan hasil dengan trigram diterjemahkan ke bahasa Indonesia
print("Top 10 Trigram (Terjemahan ke Bahasa Indonesia):")
for trigram, count in top_10_trigrams.items():
    translated_trigram = translate_to_indonesian(trigram)
    print(f"{translated_trigram}: {count} kali, {trigram_percentages[trigram]:.2f}%")
```

Untuk menjalankan kode itu ada beberapa fungsi dan library yang digunakan, fungsi yang pertama adalah extract\_trigrams, yang membantu mengambil trigram atau grup tiga kata dari setiap ulasan. Untuk mengelola data dengan efisien, digunakan library pandas, yang memfasilitasi pengolahan dan pengelompokan data dalam bentuk DataFrame. Terakhir fungsi translate\_to\_indonesian, yang disediakan oleh library Translator dari googletrans untuk menerjemahkan trigram dari bahasa Portugis ke bahasa Indonesia. lalu menghasilkan data :

```
Top 10 Trigram (Terjemahan ke Bahasa Indonesia):
Saya menerima produk: 4609 kali, 2.27%
Saya tidak menerima: 4446 kali, 2.19%
Produk tidak: 1807 kali, 0.89%
Produk tidak menerima: 1346 kali, 0.66%
tidak dikirim: 407 kali, 0.20%
Saya belum menerimanya: 404 kali, 0.20%
produk o: 368 kali, 0.18%
produk yang dibeli: 267 kali, 0.13%
sampai sekarang: 248 kali, 0.12%
Produk Produk: 245 kali, 0.12%
```

Dari 10 data tertinggi yang dihasilkan, presentase data masih belum bisa mewakili keseluruhan data bahwa data memiliki ulasan negative, sehingga harus melakukan analisis lebih lanjut yaitu menggunakan clustering.

4. Clustering Analysis

Clustering analisis digunakan untuk mengelompokkan kalimat yang serupa atau sejenis berdasarkan fitur/ karakteristik tertentu pada kolom review\_comment\_message yang memiliki kesamaan yang tinggi dalam satu cluster dan perbedaan yang signifikan dengan cluster lainnya, sehingga dapat mengidentifikasi pola kalimat yang alami, dalam clustering analysis digunakan fungsi TfidfVectorizer dari library sklearn.feature\_extraction.text untuk mengekstraksi fitur dari teks ulasan menggunakan metode TF-IDF lalu KMeans dari library sklearn.cluster untuk membentuk kelompok-kelompok berdasarkan fitur-fitur (karakteristik) tersebut dan terakhir menggunakan fungsi fit\_transform untuk memproses data, dan value\_counts untuk mengevaluasi hasil klustering. Berikut kode lengkapnya :

```
# Ekstraksi fitur menggunakan TF-IDF
tfidf_vectorizer = TfidfVectorizer(max_features=1000)
X = tfidf_vectorizer.fit_transform(orderreviews_lowrate_df['review_comment_message'])

# Lakukan K-means clustering dengan inisialisasi yang stabil menggunakan random_state
num_clusters = 4
kmeans = KMeans(n_clusters=num_clusters, random_state=42)
kmeans.fit(X)

# Tambahkan label kcluster ke DataFrame
orderreviews_lowrate_df['cluster_label'] = kmeans.labels_

# Evaluasi hasil clustering
cluster_counts = orderreviews_lowrate_df['cluster_label'].value_counts()
total_data = len(orderreviews_lowrate_df)
cluster_percentages = (cluster_counts / total_data) * 100

# Menampilkan jumlah data dan persentase masing-masing cluster
for label, count, percentage in zip(cluster_counts.index, cluster_counts, cluster_percentages):
    print(f"Cluster Label {label}: {count} data ({percentage:.2f}%")
```

Dan dari kode tersebut menghasilkan clustering :

```
Cluster Label 2: 9103 data (62.46%)
Cluster Label 3: 3825 data (26.24%)
Cluster Label 0: 1022 data (7.01%)
Cluster Label 1: 625 data (4.29%)
```

Dan dihasilkan cluster 2 yang memiliki jumlah data tertinggi, sehingga dapat disimpulkan ada 9103 atau 62.46% dari data keseluruhan yang memiliki karakteristik dan jenis ulasan dengan kata yang sama.

Mari kita lihat isi setiap cluster, menggunakan kode :

```
unique_labels = orderreviews_lowrate_df['cluster_label'].value_counts()

# Tampilkan isi dari setiap cluster untuk 5 cluster teratas
num_clusters_to_display = 5

for idx, label in enumerate(unique_labels.index):
    if idx >= num_clusters_to_display:
        break

    cluster_data = orderreviews_lowrate_df[orderreviews_lowrate_df['cluster_label'] == label]['review_comment_message']
    cluster_texts = cluster_data.tolist()[:5]

    print("Cluster Label:", label)
    for idx, original_text in enumerate(cluster_texts):
        print(f"Text {idx + 1}: {original_text}")
```

Dengan hasil :

Cluster Label: 2
Text 1: péssimo
Text 2: não gostei comprei gato por lebre
Text 3: sempre compro pela internet e a entrega ocorre antes do prazo combinado que acredito ser o prazo máximo no stark o prazo máximo já se esgotou e ainda não recebi o produto
Text 4: nada de chegar o meu pedido
Text 5: recebi somente 1 controle midea split estilo
faltou controle remoto para ar condicionado consul
Cluster Label: 3
Text 1: não recebi o produto
Text 2: não recebi o produto
Text 3: não recebi o produto
Text 4: não recebi o produto
Text 5: não recebi o produto
Cluster Label: 0
Text 1: este foi o pedido
balde com 128 peças blocos de montar 2 un r 2500 cada não foi entregue
vendido e entregue targaryen
tapete de eva nº letras 36 peças crianças 1 un r 3590 este foi entreg

Text 2: comprei o produto dia 25 de fevereiro e hoje dia 29 de marco não fora entregue na minha residência não sei se os correios desse brasil e péssimo ou foi a própria loja que demorou postar

Text 3: aqui está descrevendo como entregue só que ate agora não recebi

Text 4: produto foi entregue com umas das alças com problemas está faltando 1 pino de fixação

Text 5: dos dois produtos comprados foi entregue apenas um

Cluster Label: 1

Text 1: ainda não recebi

Text 2: estou esperando há mais de vinte dias e ainda não recebi meu produto e não consigo nenhuma informação

Text 3: o produto ainda não chegou

Text 4: ainda não recebi o produto

Text 5: ainda não recebi

Karena hasil masih menggunakan Bahasa Portugis sehingga kita perlu menerjemahkan kedalam bahasa Indonesia menggunakan fungsi translator yang disediakan oleh library googletans, berikut hasil terjemahnya:

Cluster Label: 2

Text 1 (Translated): sangat buruk

Text 2 (Translated): Saya tidak suka saya membeli kucing untuk kelinci

Text 3 (Translated): Selalu beli melalui internet dan pengiriman terjadi sebelum tenggat waktu yang saya yakini adalah tenggat waktu maksimum dalam tenggat waktu maksimum telah terjual habis dan belum menerima produk tersebut

Text 4 (Translated): Tidak ada permintaan untuk pesanan saya

Text 5 (Translated): Saya hanya menerima 1 kontrol gaya split midea tidak memiliki remote control untuk konsul pendingin udara

=====

Cluster Label: 3

Text 1 (Translated): Saya tidak menerima produk

Text 2 (Translated): Saya tidak menerima produk

Text 3 (Translated): Saya tidak menerima produk

Text 4 (Translated): Saya tidak menerima produk

Text 5 (Translated): Saya tidak menerima produk

=====

Cluster Label: 0

Text 1 (Translated): Ini adalah pesanan ember dengan 128 keping perakitan 2 un r 2500 masing -masing tidak dikirim dan dikirimkan karpet targaryen eva n° huruf 36 potong anak -anak 1 un r 3590 ini disampaikan

Text 2 (Translated): Saya membeli produk pada tanggal 25 Februari dan hari ini Marco 29 tidak dikirim ke kediaman saya, saya tidak tahu apakah kantor pos Brasil ini dan buruk atau toko itu sendiri yang mengambil posting

Text 3 (Translated): di sini menggambarkan hanya disampaikan sejauh yang belum saya terima

Text 4 (Translated): Produk dikirimkan dengan salah satu pegangan dengan masalah yang hilang 1 pin pin

Text 5 (Translated): dari dua produk yang dibeli hanya dikirimkan satu

=====

Cluster Label: 1

Text 1 (Translated): Saya belum menerimanya

Text 2 (Translated): Saya telah menunggu selama lebih dari dua puluh hari dan saya belum menerima produk saya dan saya tidak bisa mendapatkan informasi apa pun

Text 3 (Translated): Produk belum tiba

Text 4 (Translated): Saya belum menerima produk

Text 5 (Translated): Saya belum menerimanya

=====

## 5. Time Series Analysis

Untuk menjawab pertanyaan **Time-Bond** kita perlu menganalisis jumlah kolom review\_comment\_message dari waktu ke waktu selama kurun 1 tahun terakhir untuk dapat menentukan tren ulasan/review. Untuk menghitung jumlah ulasan dalam satu tahun terakhir tentunya kita menggunakan beberapa fungsi yaitu .max() fungsi python yang digunakan untuk mendapatkan tanggal terbaru pada dataframe, lalu menggunakan fungsi timedelta() dari modul datetime untuk menghitung tanggal 1 tahun terakhir dan menggunakan fungsi resample yang disediakan oleh library pandas digunakan untuk mengelompokkan bulan, berikut kode lengkapnya :

```
# Ambil tanggal terakhir dalam dataset
last_date_in_dataset = orderreviews_lowrate_df['review_creation_date'].max()

# Hitung tanggal 1 tahun sebelum tanggal terakhir
one_year_ago = last_date_in_dataset - timedelta(days=365)

# Filter data hanya untuk satu tahun terakhir
data_last_year = orderreviews_lowrate_df[orderreviews_lowrate_df['review_creation_date'] >= one_year_ago]

# Hitung jumlah ulasan per bulan
review_counts_per_month = data_last_year.resample('M', on='review_creation_date').size()
print(review_counts_per_month)
```

Lalu menghasilkan data :

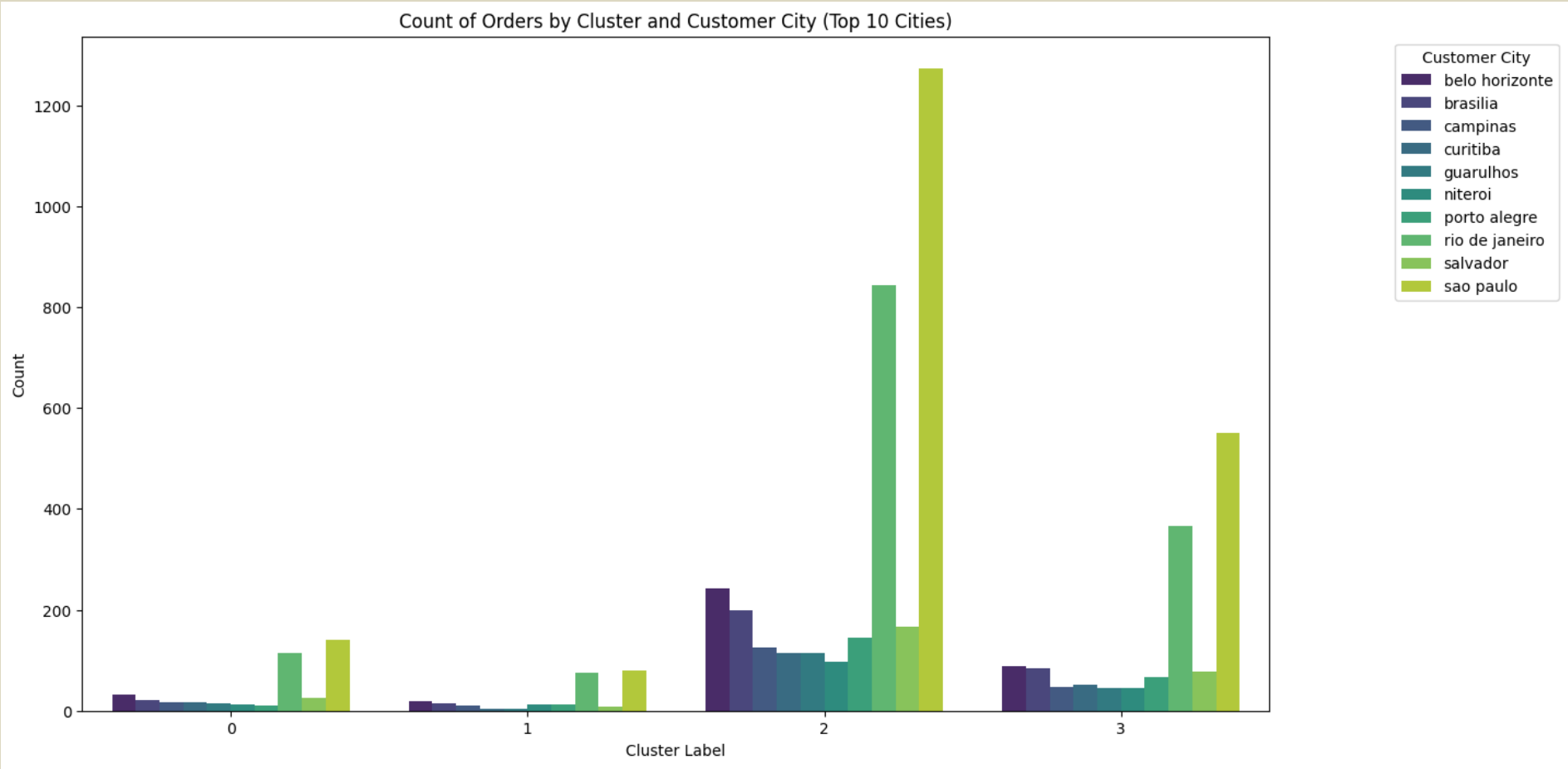
review_creation_date	
2017-08-31	20
2017-09-30	521
2017-10-31	540
2017-11-30	640
2017-12-31	1478
2018-01-31	926
2018-02-28	981
2018-03-31	1832
2018-04-30	1392
2018-05-31	903
2018-06-30	823
2018-07-31	583
2018-08-31	1092
Freq: M, dtype: int64	

## Visualisasi Data dan Kesimpulan

1. **Spesific** : Produk apa dan kota mana yang menjadi tujuan pada ulasan negatif ?
- Untuk mencari kota customer terbanyak yang dituju pengiriman, kita perlu membuat grafik dengan membandingkan kolom cluster label dengan kolom customer city pada dataframe menggunakan library matplotlib, berikut :

```
# Ambil 10 kota terbanyak
top_10_cities = city_df['customer_city'].value_counts().nlargest(10).index
# Filter DataFrame hanya untuk 10 kota teratas
top_10_cities_df = city_df[city_df['customer_city'].isin(top_10_cities)]
# Group by 'cluster_label' and 'customer_city', then count the occurrences
grouped_data = top_10_cities_df.groupby(['cluster_label', 'customer_city']).size().reset_index(name='count')
# Create the plot
plt.figure(figsize=(14, 8))
sns.barplot(data=grouped_data, x='cluster_label', y='count', hue='customer_city', palette='viridis')
# Add title and labels
plt.title('Count of Orders by Cluster and Customer City (Top 10 Cities)')
plt.xlabel('Cluster Label')
plt.ylabel('Count')
# Show legend
plt.legend(title='Customer City', loc='upper right', bbox_to_anchor=(1.25, 1))
plt.show()
```

Dan menghasilkan grafik :



Dari grafik yang dihasilkan, kota customer tertinggi yaitu kota Sao Paulo, dan kota kedua tertinggi yaitu Kota Rio de Janeiro pada keempat cluster.

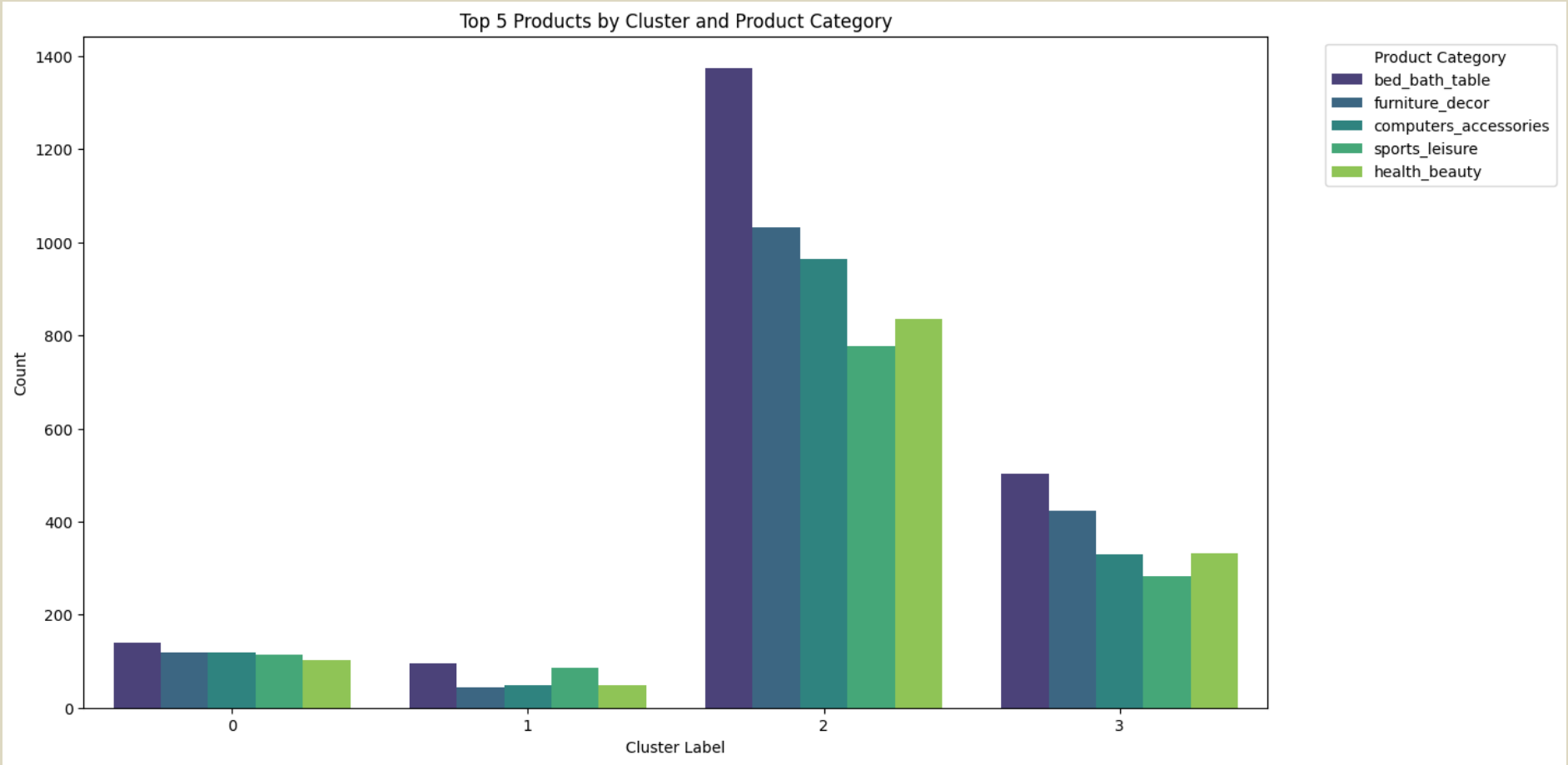
Untuk mencari produk terbanyak yang dibeli oleh customer, kita perlu membuat grafik dengan membandingkan kolom cluster label dengan kolom product\_category pada dataframe menggunakan library matplotlib, berikut :

```
# Function to get top n products in each cluster
def get_top_products(cluster_data, n=5):
    top_products = cluster_data.sort_values(by='count', ascending=False).head(n)
    return top_products
# Group by 'cluster_label' and 'product_category_name_english', then count the occurrences
grouped_data = all_df.groupby(['cluster_label', 'product_category_name_english']).size().reset_index(name='count')
# Get the top 5 products in each cluster
top_products_by_cluster = grouped_data.groupby('cluster_label', group_keys=False).apply(get_top_products, n=5)
```



```
# Create the plot
plt.figure(figsize=(14, 8))
sns.barplot(data=top_products_by_cluster, x='cluster_label', y='count', hue='product_category_name_english', palette='viridis')
# Add title and labels
plt.title('Top 5 Products by Cluster and Product Category')
plt.xlabel('Cluster Label')
plt.ylabel('Count')
# Show legend
plt.legend(title='Product Category', loc='upper right', bbox_to_anchor=(1.25, 1))
plt.show()
```

dari kode tersebut menghasilkan grafik :



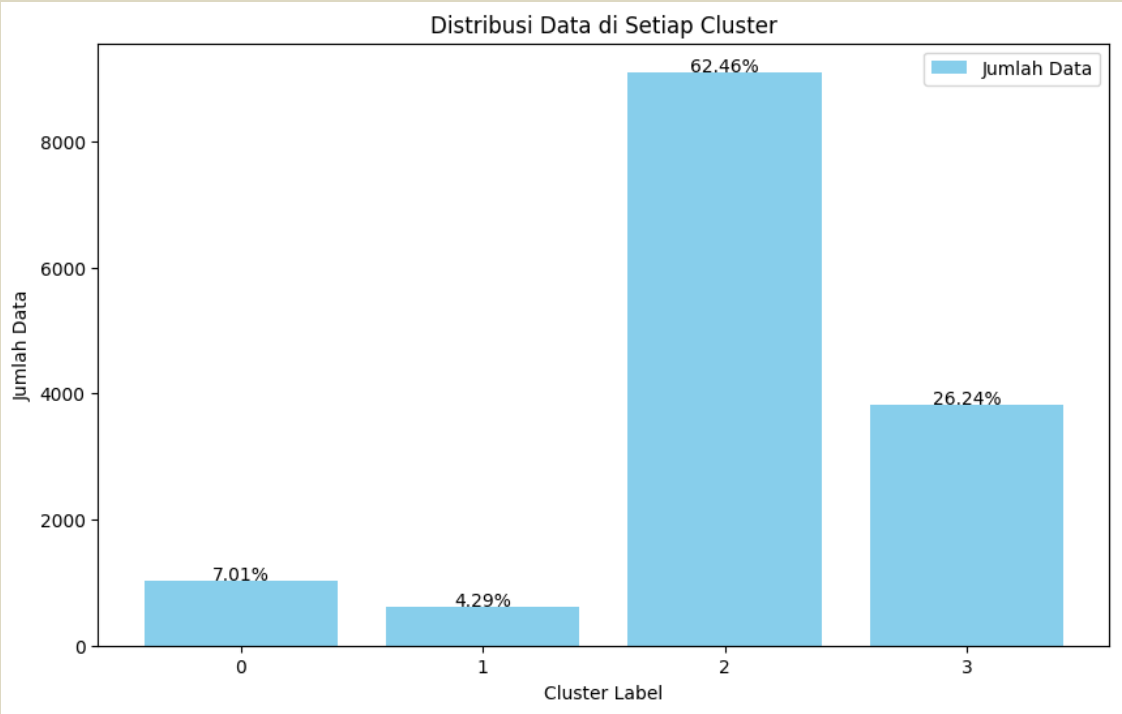
Dari grafik yang dihasilkan, product kategori tertinggi yang di beli oleh customer adalah kategori bed\_bath\_table.

2. **Measurable** : Bagaimana kita dapat mengukur tingkat ketidakpuasan dari ulasan dengan rating rendah?

Mengukur ketidakpuasan dari ulasan yaitu dengan melihat pola kata dan mengelompokan pola kata yang mirip terhadap satu kelompok atau disebut clustering, clustering yang digunakan dalam dataset ini yaitu kmeans clustering, lalu divisualisasikan menggunakan matplotlib untuk melihat distribusi data setiap cluster, dengan kode berikut :

```
# Hitung jumlah data di setiap cluster
cluster_counts = orderreviews_lowrate_df['cluster_label'].value_counts()
# Hitung total data
total_data = len(orderreviews_lowrate_df)
# Hitung persentase jumlah data di setiap cluster
cluster_percentages = (cluster_counts / total_data) * 100
# Plot grafik batang dengan presentase
plt.figure(figsize=(10, 6))
bars = plt.bar(cluster_counts.index, cluster_counts.values, color='skyblue', label='Jumlah Data')
plt.xlabel('Cluster Label')
plt.ylabel('Jumlah Data')
plt.title('Distribusi Data di Setiap Cluster')
plt.xticks(cluster_counts.index)
# Tambahkan label persentase di atas setiap bar
for bar, percentage in zip(bars, cluster_percentages):
    plt.text(bar.get_x() + bar.get_width() / 2, bar.get_height() + 10, f'{percentage:.2f}%', ha='center')
plt.legend()
plt.show()
```

Dan menghasilkan grafik :



Dari grafik diatas dihasilkan cluster 2 yang memiliki jumlah data tertinggi, sehingga dapat disimpulkan ada 9103 atau 62.46% dari data keseluruhan yang memiliki karakteristik dan jenis ulasan dengan kata yang sama, dengan isi ulasan setiap clustering:

Cluster Label: 2  
Text 1 (Translated): sangat buruk  
Text 2 (Translated): Saya tidak suka saya membeli kucing untuk kelinci  
Text 3 (Translated): Selalu beli melalui internet dan pengiriman terjadi sebelum tenggat waktu yang saya yakini adalah tenggat waktu maksimum dalam tenggat waktu maksimum telah terjual habis dan belum menerima produk tersebut  
Text 4 (Translated): Tidak ada permintaan untuk pesanan saya  
Text 5 (Translated): Saya hanya menerima 1 kontrol gaya split midea tidak memiliki remote control untuk konsul pendingin udara  
=====

Cluster Label: 3  
Text 1 (Translated): Saya tidak menerima produk  
Text 2 (Translated): Saya tidak menerima produk  
Text 3 (Translated): Saya tidak menerima produk  
Text 4 (Translated): Saya tidak menerima produk  
Text 5 (Translated): Saya tidak menerima produk  
=====

Cluster Label: 0  
Text 1 (Translated): Ini adalah pesanan ember dengan 128 keping perakitan 2 un r 2500 masing -masing tidak dikirim dan dikirimkan karpet targaryen eva n° huruf 36 potong anak -anak 1 un r 3590 ini disampaikan  
Text 2 (Translated): Saya membeli produk pada tanggal 25 Februari dan hari ini Marco 29 tidak dikirim ke kediaman saya, saya tidak tahu apakah kantor pos Brasil ini dan buruk atau toko itu sendiri yang mengambil posting  
Text 3 (Translated): di sini menggambarkan hanya disampaikan sejauh yang belum saya terima  
Text 4 (Translated): Produk dikirimkan dengan salah satu pegangan dengan masalah yang hilang 1 pin pin  
Text 5 (Translated): dari dua produk yang dibeli hanya dikirimkan satu  
=====

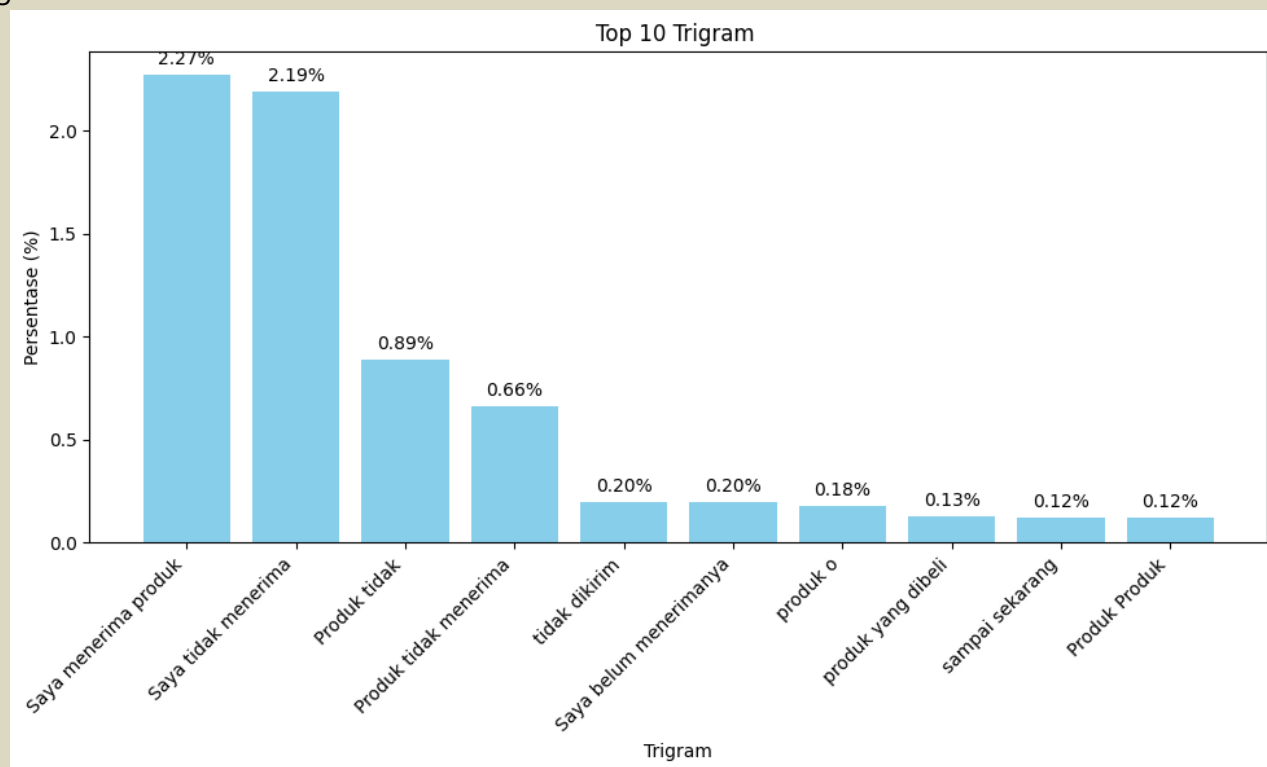
Cluster Label: 1  
Text 1 (Translated): Saya belum menerimanya  
Text 2 (Translated): Saya telah menunggu selama lebih dari dua puluh hari dan saya belum menerima produk saya dan saya tidak bisa mendapatkan informasi apa pun  
Text 3 (Translated): Produk belum tiba  
Text 4 (Translated): Saya belum menerima produk  
Text 5 (Translated): Saya belum menerimanya  
=====

dari ulasan ulasan yang sudah diclustering sesuai dengan kemiripan kata, dan kita bisa menyimpulkan bahwa :  
Cluster 1, 2, dan 3 memiliki topik atau inti ulasan bahwa customer tidak menerima produknya, dan cluster 0 adalah customer dengan ulasa negative lainya seperti produk tidak lengkap dan lain lain, sehingga kita dapat menyimpulkan bahwa **92.99% ulasan negatif** memiliki topik yaitu produk yang tidak diterima oleh customer.

3. **Achievable:** Apakah mungkin untuk mengidentifikasi pola umum dan kemiripan dalam bahasa atau frasa yang digunakan dalam ulasan negatif?  
Mengidentifikasi pola umum dan kemiripan dalam bahasa atau frasa sangat memungkinkan dilakukan dengan N-Gram Analysis, dan telah divisualisasikan menggunakan matpolib dengan kode :

```
import matplotlib.pyplot as plt
# Data
trigrams = [
    "Saya menerima produk",
    "Saya tidak menerima",
    "Produk tidak",
    "Produk tidak menerima",
    "tidak dikirim",
    "Saya belum menerimanya",
    "produk o",
    "produk yang dibeli",
    "sampai sekarang",
    "Produk Produk"
]
counts = [4609, 4446, 1807, 1346, 407, 404, 368, 267, 248, 245]
percentages = [2.27, 2.19, 0.89, 0.66, 0.20, 0.20, 0.18, 0.13, 0.12, 0.12]
# Plot
fig, ax = plt.subplots(figsize=(10, 6))
ax.bar(trigrams, percentages, color='skyblue')
ax.set_ylabel('Persentase (%)')
ax.set_xlabel('Trigram')
ax.set_title('Top 10 Trigram ')
# Rotate x-axis labels for better readability
plt.xticks(rotation=45, ha='right')
# Display the percentages on top of the bars
for i in range(len(trigrams)):
    ax.text(i, percentages[i] + 0.05, f'{percentages[i]:.2f}%', ha='center')
plt.tight_layout()
plt.show()
```

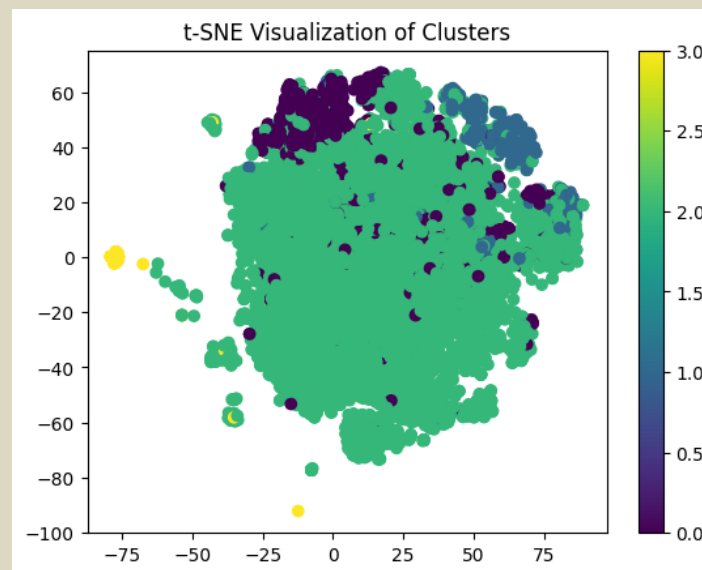
dan menghasilkan grafik :



Dari Trigram yang telah dibuat urutan kata atau karakter yang terdiri dari 3 elemen berturut-turut dan didapatkan 10 pola/frasa dari ulasan yang sering disebutkan, namun belum mewakili keseluruhan data sehingga dilakukan clustering dan didapatkan 4 cluster untuk kombinasi pola/ frasa yang serupa dan divisualisasikan menggunakan t-SNE grafik dengan library matpolib, seperti berikut :

```
# Reduksi dimensi menggunakan TruncatedSVD (untuk mengubah matriks sparse menjadi dense)
# dan kemudian menggunakan t-SNE pada matriks dense hasil SVD
tsvd = TruncatedSVD(n_components=50) # Ubah n_components sesuai kebutuhan
normalizer = Normalizer(copy=False)
tsne = TSNE(n_components=2, random_state=42)
# Pipeline untuk menggabungkan proses reduksi dimensi
pipeline = Pipeline([
    ('svd', tsvd),
    ('norm', normalizer),
    ('tsne', tsne)
])
# Perform t-SNE on the TF-IDF data
tsne_result = pipeline.fit_transform(X.toarray())
# Get the cluster labels
cluster_labels = orderreviews_lowrate_df['cluster_label']
# Plot hasil clustering menggunakan t-SNE
plt.scatter(tsne_result[:, 0], tsne_result[:, 1], c=cluster_labels, cmap='viridis')
plt.title('t-SNE Visualization of Clusters')
plt.colorbar()
plt.show()
```

Dan menghasilkan grafik :



Dapat dilihat dari kepadatan dan perbedaan warna pada grafik, titik titik memiliki jarak yang sangat dekat dan padat itu menunjukkan bahwa pola kalimat/ frasa/kata dari ulasan memiliki kemiripan yang tinggi, dan dapat dilihat warna hijau atau menunjukkan cluster dua mendominasi sesuai dengan yang disebutkan pada pertanyaan sebelumnya.

4. **Relevant** : Apa tema umum dari ulasan negatif yang dapat dijadikan sebagai panduan untuk perbaikan produk atau layanan?

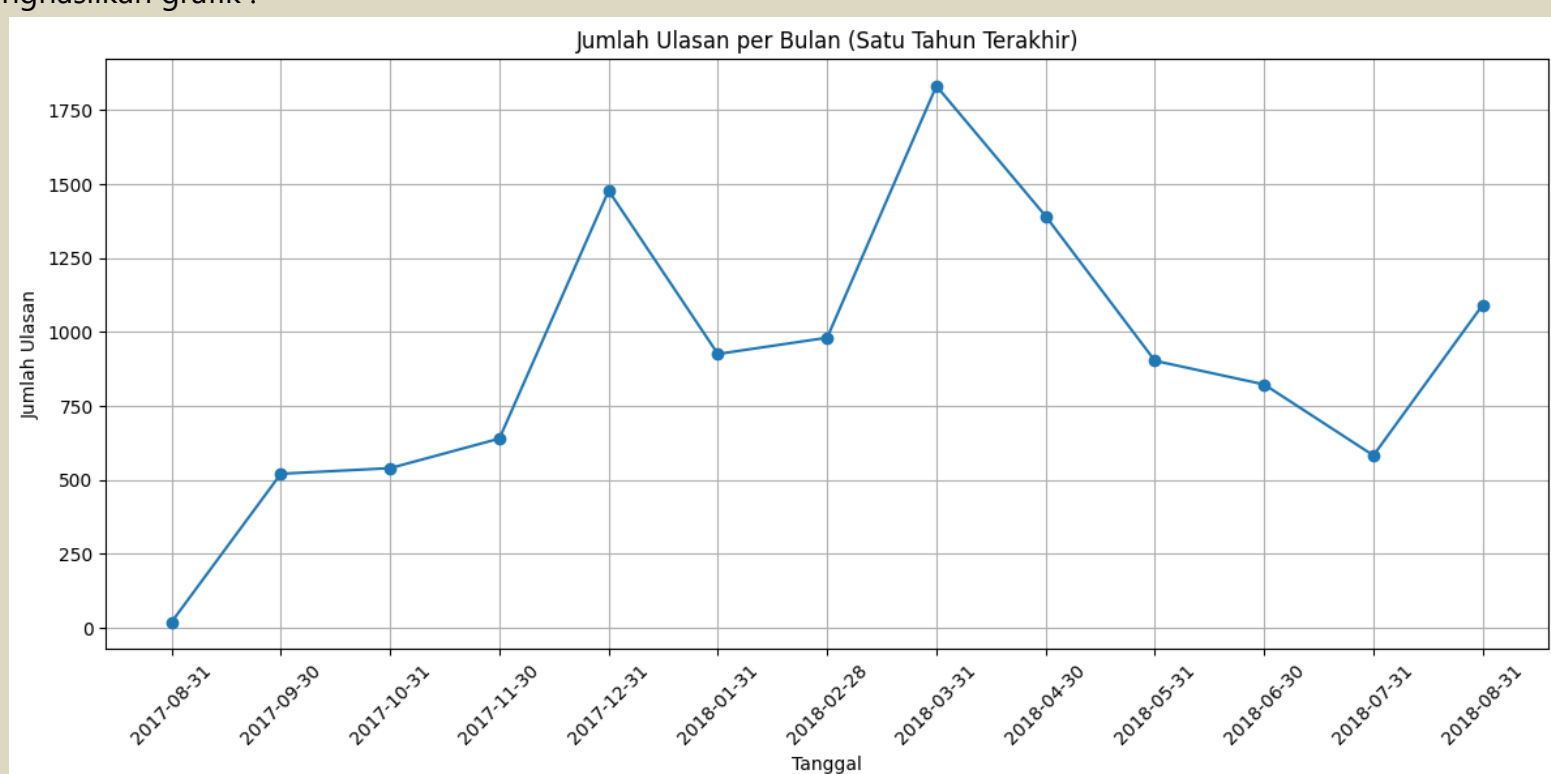
Dari pertanyaan no 2, tema umum atau topik dari ulasan negatif yaitu disimpulkan **produk tidak diterima oleh customer** , hal ini dapat dijadikan panduan untuk perbaikan layanan agar ulasan dapat menjadi baik di tahun berikutnya.

5. **Time-Bound** : Berapa jumlah ulasan negatif yang telah diterima selama satu tahun terakhir, dan bagaimana tren keluhan ini selama satu tahun terakhir?

Hasil dari Time Series Analysis divisualisasikan menggunakan matplotlib, melalui kode :

```
# Ambil tanggal terakhir dalam dataset
last_date_in_dataset = orderreviews_lowrate_df['review_creation_date'].max()
# Hitung tanggal 1 tahun sebelum tanggal terakhir
one_year_ago = last_date_in_dataset - timedelta(days=365)
# Filter data hanya untuk satu tahun terakhir
data_last_year = orderreviews_lowrate_df[orderreviews_lowrate_df['review_creation_date'] >= one_year_ago]
# Hitung jumlah ulasan per bulan
review_counts_per_month = data_last_year.resample('M', on='review_creation_date').size()
# Buat urutan tanggal dengan frekuensi M (akhir setiap bulan)
date_sequence = pd.date_range(start=one_year_ago, periods=len(review_counts_per_month), freq=MonthEnd())
# Ambil tanggal dalam format tahun-bulan-hari
formatted_dates = [date.strftime('%Y-%m-%d') for date in date_sequence]
# Plot time series
plt.figure(figsize=(12, 6))
plt.plot(formatted_dates, review_counts_per_month.values, marker='o')
plt.xlabel('Tanggal')
plt.ylabel('Jumlah Ulasan')
plt.title('Jumlah Ulasan per Bulan (Satu Tahun Terakhir)')
plt.grid(True)
plt.xticks(rotation=45) # Rotasi label tanggal sebesar 45 derajat
plt.tight_layout()
plt.show()
```

Dan menghasilkan grafik :



Dari data yang kita dapatkan selama satu tahun terakhir dapat dilihat pada tabel, selama periode dari Agustus 2017 hingga Agustus 2018, terlihat variasi yang cukup signifikan dalam jumlah ulasan yang dikirimkan. Terjadi lonjakan yang mencolok pada bulan Desember 2017, dengan jumlah ulasan mencapai puncak tertingginya, yakni 1478 ulasan. Namun, setelah itu, terjadi

penurunan pada bulan Januari 2018, meskipun mengalami sedikit peningkatan pada bulan Februari 2018, kemudian pada bulan Maret 2018, jumlah ulasan kembali meningkat secara mencolok, bahkan melewati jumlah tertinggi sebelumnya setelah bulan Desember 2017, yaitu 1832 ulasan. Setelah Maret 2018, terjadi penurunan yang berkelanjutan hingga Juni 2018. Namun, pada bulan Juli 2018, terlihat peningkatan jumlah ulasan kembali. Secara keseluruhan, meskipun terdapat variasi data bulanan yang tidak teratur, dapat dilihat bahwa Desember 2017 dan Maret 2018 merupakan bulan-bulan di mana jumlah ulasan mencapai puncak tertinggi selama periode tersebut.

**Kesimpulan keseluruhan :**

92.99% ulasan negatif memiliki topik yaitu produk yang tidak diterima oleh customer dengan mayoritas kota penerimanya adalah Sao Paulo dan mayoritas kategori produk yang dibeli adalah bed\_bath\_table dan pada t1 tahun terakhir ulasan tidak memiliki jumlah yang teratur pada setiap bulannya dan puncak tertinggi di capai pada Desember 2017 dan Maret 2018.

Terima Kasih! Anda bisa melihat kode keseluruhan pada link :