Rakamin Academy

Homework

Machine Learning Preparation

Bytesquad
Gede Verel Aditya Setiabudi
Tarisha Zhafira
Muhammad Raditya Nur Aziz
Ida Bagus Putu Basma Yoga
Bunga Anggun Chintamy
Muhammad Abigail Anargya
Yusma Cantika Parhati
Ida Ayu Tri Sabina Putri
Egydia Alfariza Ramadhani
Atqiya Trianda Putra Anugrah

Jhordy Wong Abuhasan

Product Classification: Memprediksi Eksklusivitas Produk

- **Deskripsi**:

  Memprediksi apakah suatu produk eksklusif atau tidak berdasarkan fitur yang tersedia

- **Link dataset**: Dataset

- **Link google colab:** https://colab.research.google.com/drive/1TuS7--Os4YEKyG6VzJfZd7YtvL8m7QGf?usp=sharing

- **Link github:** https://github.com/Rhutless29/HW8-Finpro1.2.git

Load Data

```
[7] from pyspark.sql import SparkSession
    spark = SparkSession.builder.getOrCreate()
    airports = spark.read.csv("/content/drive/MyDrive/Big data 1/Product_Exclusive_Classification.csv",

    header=True, inferSchema=True)
```

1. Descriptive Statistics (5 poin)

Gunakan function **`info`** dan **`describe`** pada dataset final project kalian. Tuliskan hasil observasinya, seperti:

A. Apakah ada kolom dengan tipe data kurang sesuai, atau nama kolom dan isinya kurang sesuai?

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8000 entries, 0 to 7999
Data columns (total 9 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   id                 8000 non-null   int64
 1   brand              8000 non-null   object
 2   category           7987 non-null   object
 3   rating             7905 non-null   float64
 4   number_of_reviews  7991 non-null   float64
 5   love               7966 non-null   float64
 6   price              7992 non-null   float64
 7   value_price        7983 non-null   float64
 8   exclusive          8000 non-null   int64
dtypes: float64(5), int64(2), object(2)
memory usage: 562.6+ KB
```

```
# Menampilkan informasi dasar tentang data
df.info()

# Menampilkan statistik deskriptif tentang data
df.describe(include='all')
```

Semua tipe data terlihat sudah sesuai. Kolom brand dan category memiliki tipe data object yang sesuai dengan isinya (teks). Tidak ada kolom yang berisi angka namun bertipe object.

1. Descriptive Statistics (5 poin)

B. Apakah ada kolom yang memiliki nilai kosong? Jika ada, apa saja?

```
Missing values per column:
 id                   0
brand                 0
category             13
rating               95
number_of_reviews     9
love                 34
price                 8
value_price          17
exclusive             0
dtype: int64
```

Beberapa kolom memiliki nilai kosong (null):

- category memiliki 23 nilai kosong.
- rating memiliki 95 nilai kosong.
- number_of_reviews memiliki 9 nilai kosong.
- ove memiliki 34 nilai kosong.
- price memiliki 8 nilai kosong.
- value_price memiliki 17 nilai kosong.

1. Descriptive Statistics (5 poin)

C. Apakah ada kolom yang memiliki nilai summary agak aneh?
(min/mean/median/max/unique/top/freq)

	id	brand	category	rating	number_of_reviews	love	price	value_price	exclusive
count	8.000000e+03	8000	7987	7905.000000	7991.000000	7.966000e+03	7992.000000	7983.000000	8000.000000
unique	NaN	310	142	NaN	NaN	NaN	NaN	NaN	NaN
top	NaN	SEPHORA COLLECTION	Perfume	NaN	NaN	NaN	NaN	NaN	NaN
freq	NaN	492	619	NaN	NaN	NaN	NaN	NaN	NaN
mean	1.910231e+06	NaN	NaN	4.085136	303.574396	1.756396e+04	49.900935	50.983300	0.255875
std	3.858353e+05	NaN	NaN	0.761069	931.724460	4.425339e+04	46.864764	48.473049	0.436379
min	5.000000e+01	NaN	NaN	0.000000	0.000000	0.000000e+00	2.000000	2.000000	0.000000
25%	1.773379e+06	NaN	NaN	4.000000	14.000000	2.000000e+03	24.000000	24.000000	0.000000
50%	2.030360e+06	NaN	NaN	4.000000	56.000000	5.500000e+03	35.000000	35.000000	0.000000
75%	2.185074e+06	NaN	NaN	4.500000	231.500000	1.530000e+04	59.000000	60.000000	1.000000
max	2.293801e+06	NaN	NaN	5.000000	19000.000000	1.300000e+06	549.000000	549.000000	1.000000

1. Descriptive Statistics (5 poin)

C. Apakah ada kolom yang memiliki nilai summary agak aneh? (min/mean/median/max/unique/top/freq)

	id
count	8.000000e+03
unique	NaN
top	NaN
freq	NaN
mean	1.910231e+06
std	3.858353e+05
min	5.000000e+01
25%	1.773379e+06
50%	2.030360e+06
75%	2.185074e+06
max	2.293801e+06

id:
Tidak ada masalah, nilai id bervariasi secara unik sebagai penanda setiap baris, dengan tidak ada nilai summary yang mencurigakan.

	brand	c
count	8000	
unique	310	
top	SEPHORA COLLECTION	
freq	492	
mean	NaN	
std	NaN	
min	NaN	
25%	NaN	
50%	NaN	
75%	NaN	
max	NaN	

brand:
Tidak ada masalah, kolom brand memiliki 310 nilai unik, dengan brand SEPHORA COLLECTION sebagai yang paling sering muncul (freq = 492). Ini masuk akal jika dataset mencakup berbagai merek.

1. Descriptive Statistics (5 poin)

C. Apakah ada kolom yang memiliki nilai summary agak aneh? (min/mean/median/max/unique/top/freq)

	category
count	7987
unique	142
top	Perfume
freq	619
mean	NaN
std	NaN
min	NaN
25%	NaN
50%	NaN
75%	NaN
max	NaN

category:
Tidak ada masalah yang jelas, meskipun kolom ini memiliki 142 kategori unik, yang dapat diharapkan dari berbagai produk kecantikan. Nilai top adalah Perfume dengan frekuensi 619, yang dapat dianggap normal jika parfum adalah kategori yang dominan.

	rating	n
count	7905.000000	
unique	NaN	
top	NaN	
freq	NaN	
mean	4.085136	
std	0.761069	
min	0.000000	
25%	4.000000	
50%	4.000000	
75%	4.500000	
max	5.000000	

rating:
mean rating adalah 4.08, yang masuk akal untuk produk dengan ulasan cenderung positif. Nilai min adalah 0, yang mungkin aneh jika seharusnya rating berkisar antara 1 hingga 5, ini bisa jadi kesalahan input atau produk yang belum mendapat rating.

1. Descriptive Statistics (5 poin)

C. Apakah ada kolom yang memiliki nilai summary agak aneh? (min/mean/median/max/unique/top/freq)

	number_of_reviews
count	7991.000000
unique	NaN
top	NaN
freq	NaN
mean	303.574396
std	931.724460
min	0.000000
25%	14.000000
50%	56.000000
75%	231.500000
max	19000.000000

number_of_reviews:
min adalah 0, yang bisa terjadi jika ada produk yang belum memiliki ulasan. max adalah 19,000, yang mungkin agak tinggi tetapi mungkin valid untuk produk yang sangat populer. Tidak ada masalah signifikan selain min = 0, yang mungkin perlu pengecekan lebih lanjut.

	love
count	7.966000e+03
unique	NaN
top	NaN
freq	NaN
mean	1.756396e+04
std	4.425339e+04
min	0.000000e+00
25%	2.000000e+03
50%	5.500000e+03
75%	1.530000e+04
max	1.300000e+06

love:
max adalah 1,300,000, yang tampak sangat tinggi dibandingkan dengan mean sekitar 17,563 dan median 2,000. Ini mungkin indikasi outlier atau produk yang sangat populer. Perbedaan yang signifikan antara mean dan median menunjukkan distribusi yang sangat miring, mungkin karena beberapa produk memiliki jumlah “love” yang jauh lebih tinggi dari yang lain.

1. Descriptive Statistics (5 poin)

C. Apakah ada kolom yang memiliki nilai summary agak aneh? (min/mean/median/max/unique/top/freq)

	price
count	7992.000000
unique	NaN
top	NaN
freq	NaN
mean	49.900935
std	46.864764
min	2.000000
25%	24.000000
50%	35.000000
75%	59.000000
max	549.000000

price:
min adalah 2, yang cukup rendah untuk produk kecantikan. max adalah 549, yang bisa masuk akal untuk produk mewah atau kit yang lebih besar, tetapi dapat dianggap sebagai outlier tergantung pada jenis produk.

	value_price
count	7983.000000
unique	NaN
top	NaN
freq	NaN
mean	50.983300
std	48.473049
min	2.000000
25%	24.000000
50%	35.000000
75%	60.000000
max	549.000000

value_price:
Nilai max adalah 549, yang sama dengan kolom price. Ini masuk akal jika value_price digunakan untuk menunjukkan nilai produk sebelum diskon atau sebagai perbandingan nilai. min adalah 2, sama seperti kolom price, yang tidak tampak aneh karena mungkin beberapa produk memang berharga rendah.

1. Descriptive Statistics (5 poin)

C. Apakah ada kolom yang memiliki nilai summary agak aneh? (min/mean/median/max/unique/top/freq)

	exclusive
count	8000.000000
unique	NaN
top	NaN
freq	NaN
mean	0.255875
std	0.436379
min	0.000000
25%	0.000000
50%	0.000000
75%	1.000000
max	1.000000

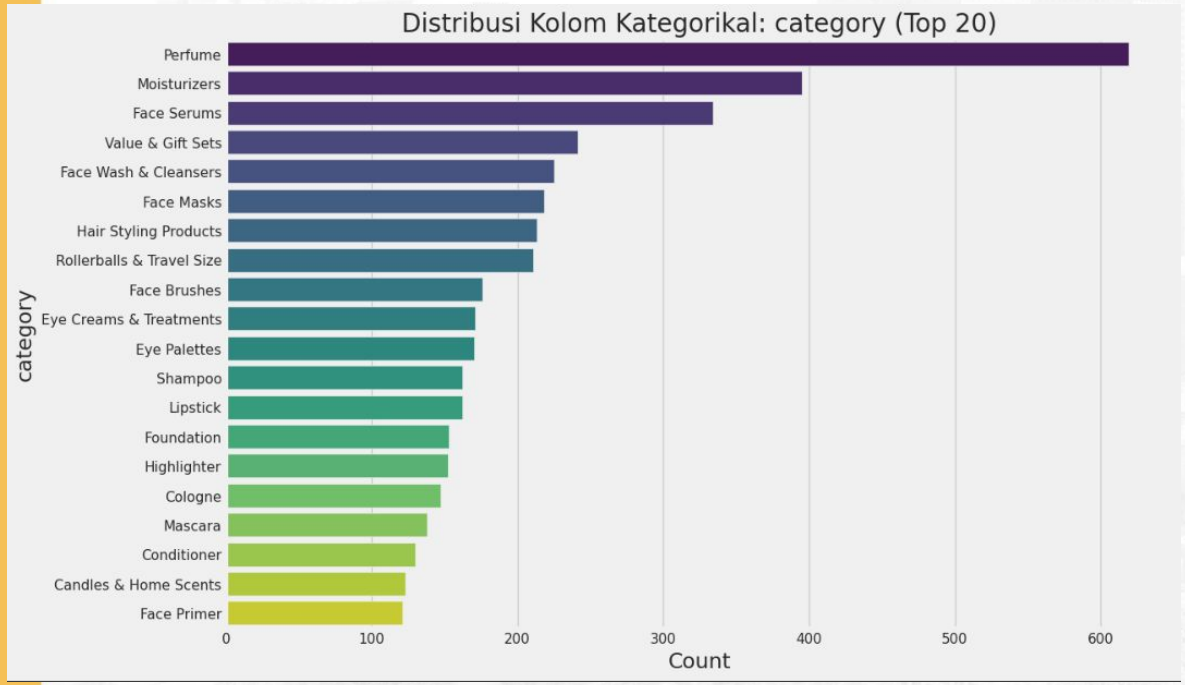
exclusive:
Kolom ini memiliki nilai rata-rata 0.256 dan terdiri dari nilai 0 dan 1 saja, dengan 1 sebagai penanda eksklusifitas. Tidak ada masalah karena nilai ini sepertinya biner dan konsisten.

Kesimpulan:
Nilai yang tampak agak aneh adalah pada kolom rating (nilai min = 0 yang mungkin kesalahan), number_of_reviews (nilai max = 19000, yang bisa menjadi outlier tetapi mungkin valid), dan love (dengan max = 1,300,000 yang tampak sangat tinggi dan mungkin merupakan outlier).

2. Univariate Analysis : Code Documentation (10 poin)

```
# Visualisasi distribusi untuk kolom numerik
numeric_cols = df.select_dtypes(include=['float64', 'int64']).columns
for col in numeric_cols:
    plt.figure(figsize=(10, 6))
    sns.histplot(df[col], kde=True, bins=30)
    plt.title(f'Distribusi Kolom Numerik: {col}')
    plt.xlabel(col)
    plt.ylabel('Frequency')
    plt.show()

# Visualisasi distribusi untuk kolom kategorikal
categorical_cols = df.select_dtypes(include=['object', 'category']).columns
for col in categorical_cols:
    plt.figure(figsize=(12, 8))

    # Menampilkan hanya 20 kategori teratas (sesuaikan jumlah sesuai kebutuhan)
    top_categories = df[col].value_counts().nlargest(20)

    # Grafik batang horizontal
    sns.barplot(y=top_categories.index, x=top_categories.values, palette="viridis")

    plt.title(f'Distribusi Kolom Kategorikal: {col} (Top 20)')
    plt.xlabel('Count')
    plt.ylabel(col)
    plt.show()
```

2. Univariate Analysis : Category (10 poin)

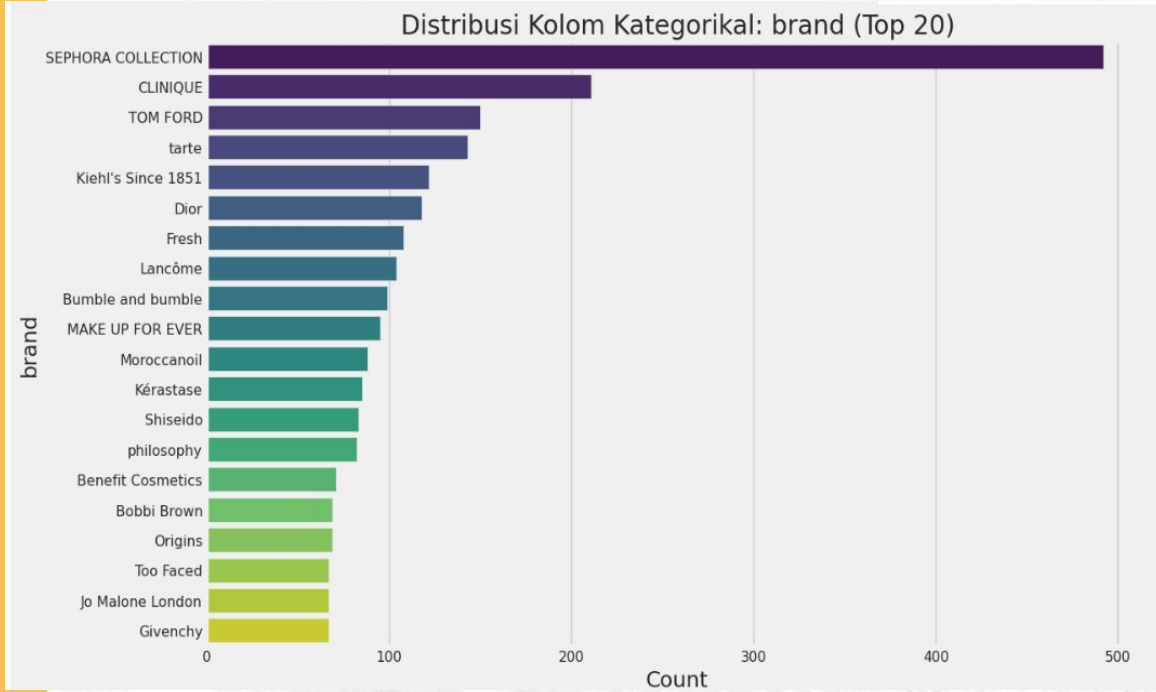


Kategori Produk: "Perfume" adalah kategori dominan, diikuti oleh "Moisturizers," "Face Serums," dan lainnya.

Distribusi Brand: Sangat skewed ke kiri, dengan beberapa brand memiliki frekuensi tinggi, sementara sebagian besar brand memiliki frekuensi rendah.

Rekomendasi Preprocessing: Mengelompokkan kategori dengan data sedikit agar lebih mudah dianalisis.

2. Univariate Analysis : Brand (10 poin)



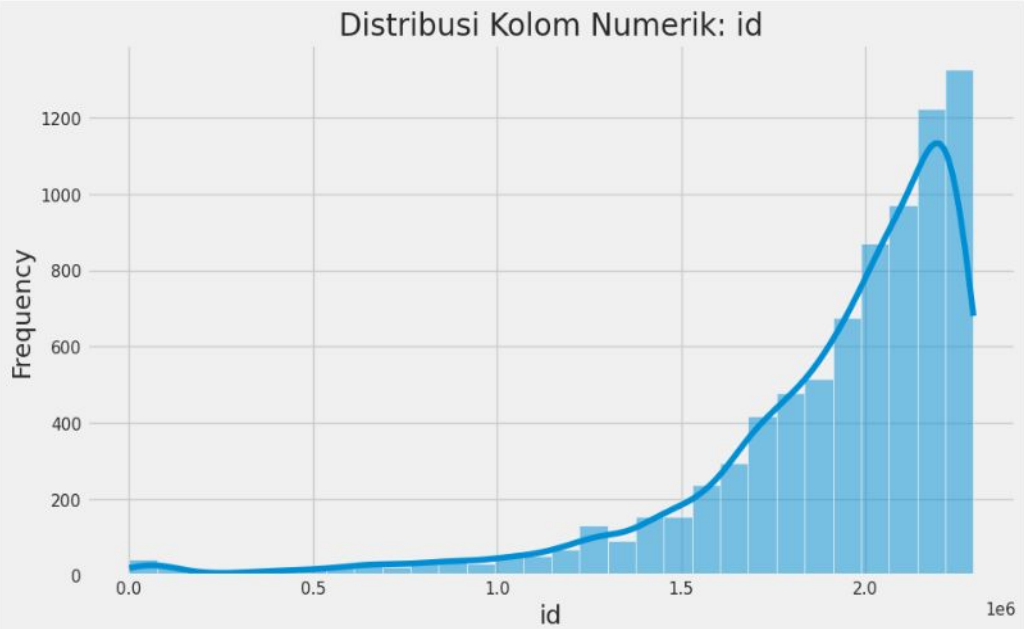
Dominasi Brand: Brand seperti "SEPHORA COLLECTION," "CLINIQUE," dan "TOM FORD" memiliki frekuensi tertinggi.

Kategori Produk: Distribusi skewed ke kiri, dengan "Perfume" sebagai kategori paling dominan.

Rekomendasi Preprocessing:

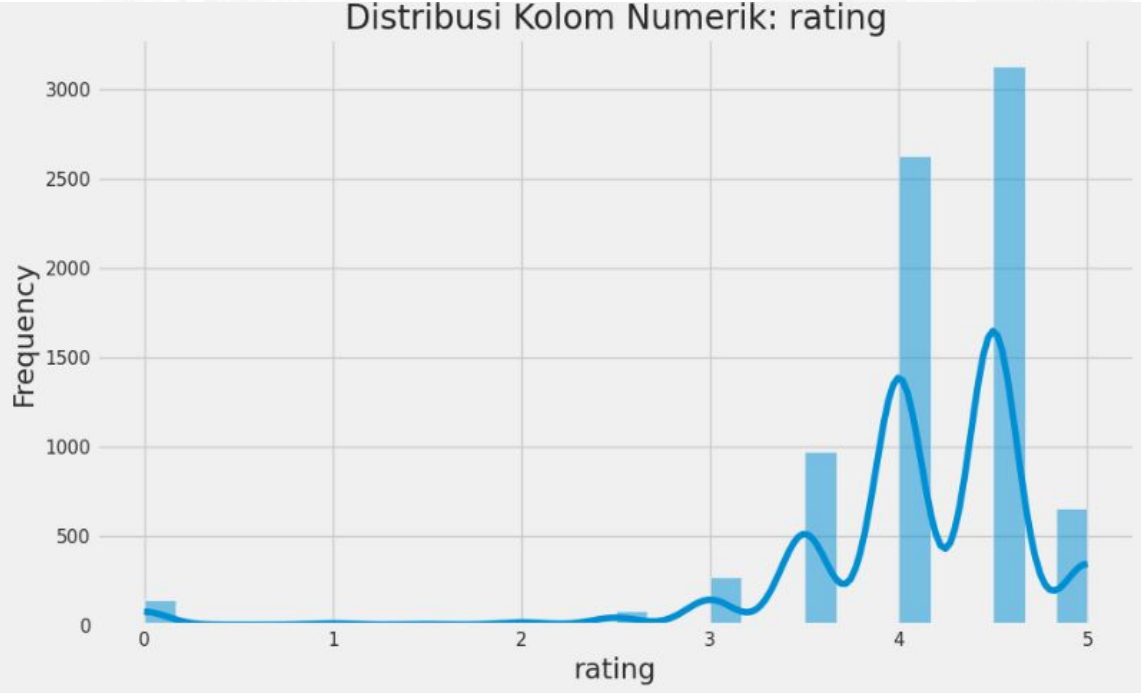
- Fokus analisis pada brand populer.
- Kelompokkan brand dengan data yang lebih sedikit untuk menyederhanakan analisis.

2. Univariate Analysis : id (10 poin)



Kolom ini menunjukkan identitas unik untuk produk dan sepertinya tidak memiliki pola distribusi yang perlu diperhatikan.

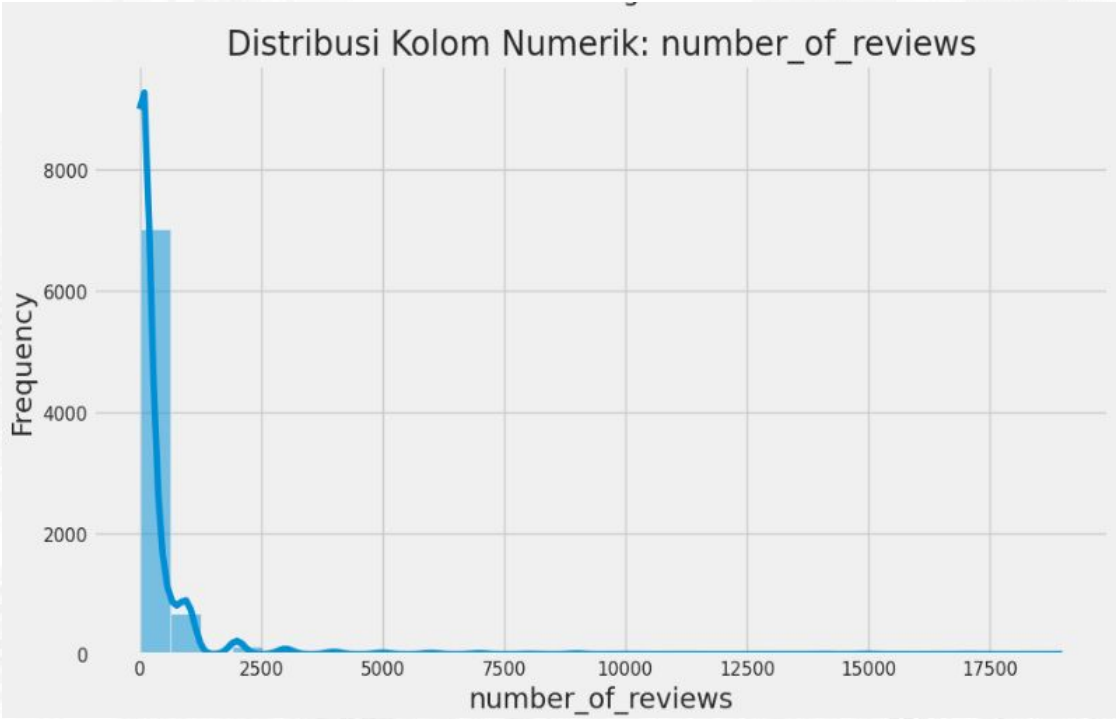
2. Univariate Analysis : Rating (10 poin)



Distribusi rating terlihat **bimodal,** dengan Puncak utama berada di sekitar rating 4 dan 5, menunjukkan mayoritas produk memiliki rating baik, dengan beberapa produk di rating rendah sekitar 1.

Preprocessing: Tidak diperlukan tindakan khusus karena distribusi rating sudah cukup baik dan tidak ada outlier yang signifikan.

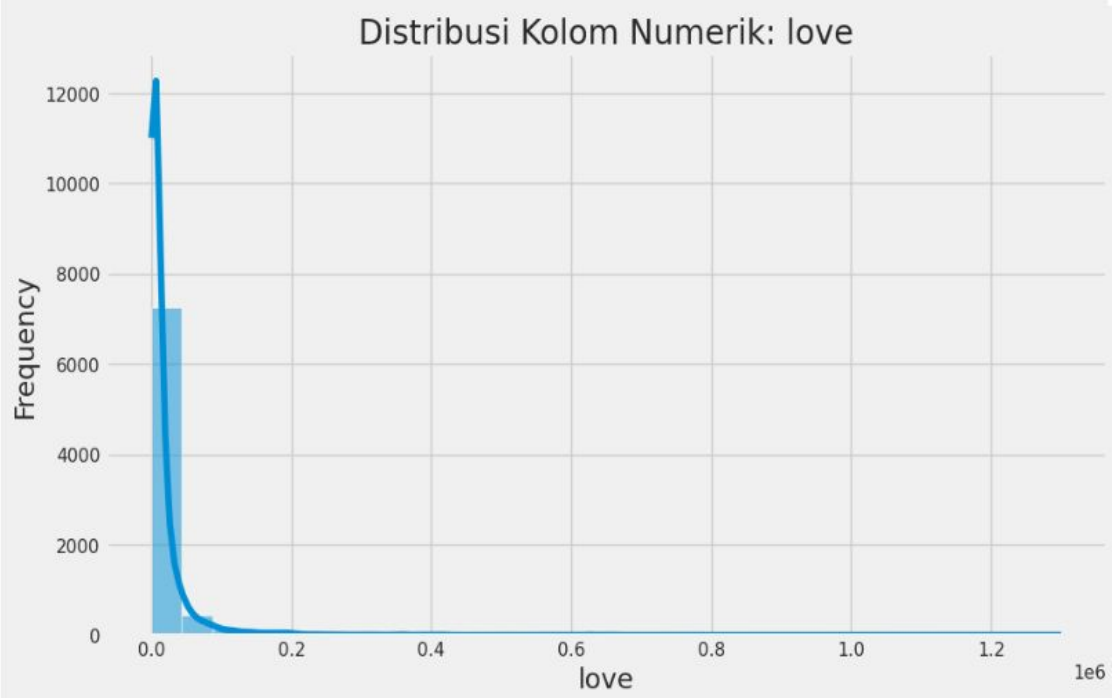
2. Univariate Analysis : Number of Review (10 poin)



Skewness: Sangat skewed ke kanan, dengan sebagian besar produk memiliki sedikit ulasan dan love, namun ada beberapa produk dengan jumlah ulasan yang sangat tinggi.

Outlier: Produk dengan ulasan sangat tinggi dianggap outlier, menunjukkan popularitas tinggi.

2. Univariate Analysis : Love (10 poin)

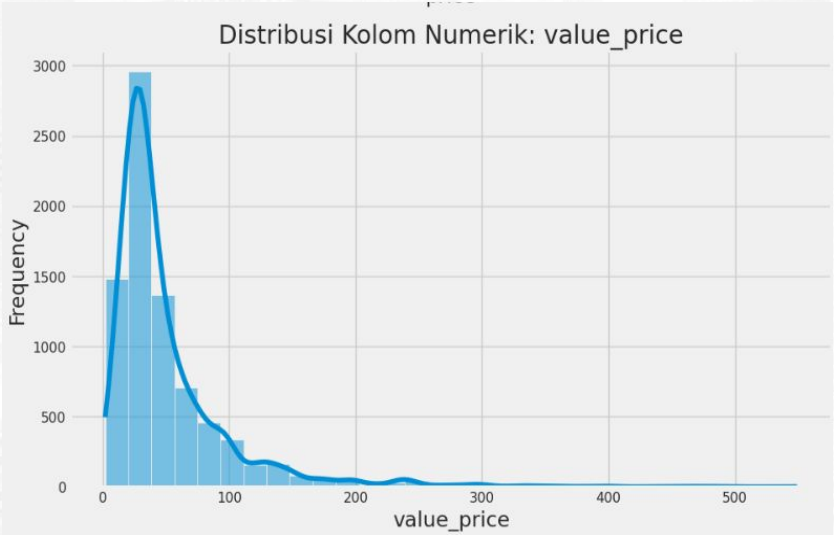


Skewness: Sangat skewed ke kanan, dengan sebagian besar nilai "love" berada di angka rendah, tetapi terdapat beberapa outlier dengan nilai sangat tinggi.

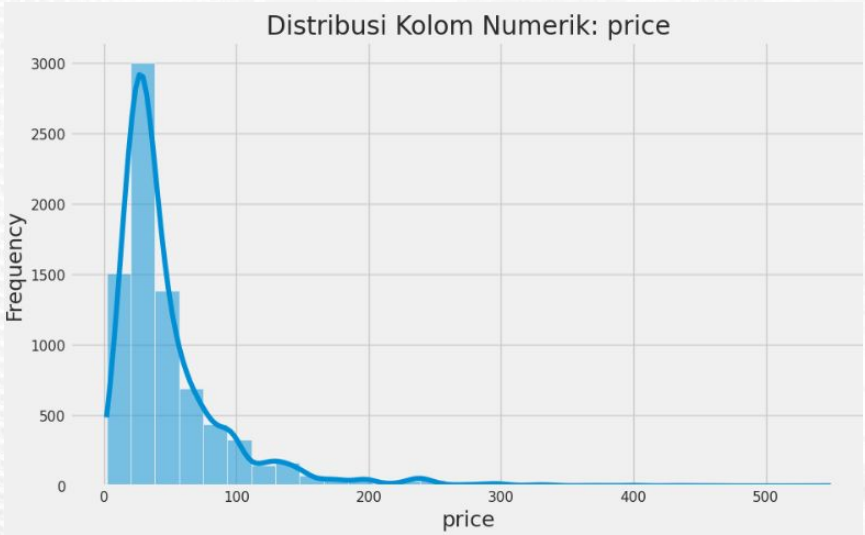
Outlier: Jika ada produk dengan nilai "love" yang sangat tinggi (sedang populer atau tren) dianggap outlier nilai "love" yang tinggi dapat menyebabkan ketidakseimbangan dalam analisis dan mungkin perlu diatasi (misalnya, dengan winsorizing).

2. Univariate Analysis : Value Price & Price (10 poin)

distribusi kolom value_price



distribusi kolom price



2. Univariate Analysis : Value Price & Price (10 poin)

Distribusi: Kedua kolom memiliki distribusi yang sangat skewed ke kanan dengan sebagian besar harga berada di rentang rendah (0–100) dan beberapa outlier di harga tinggi.

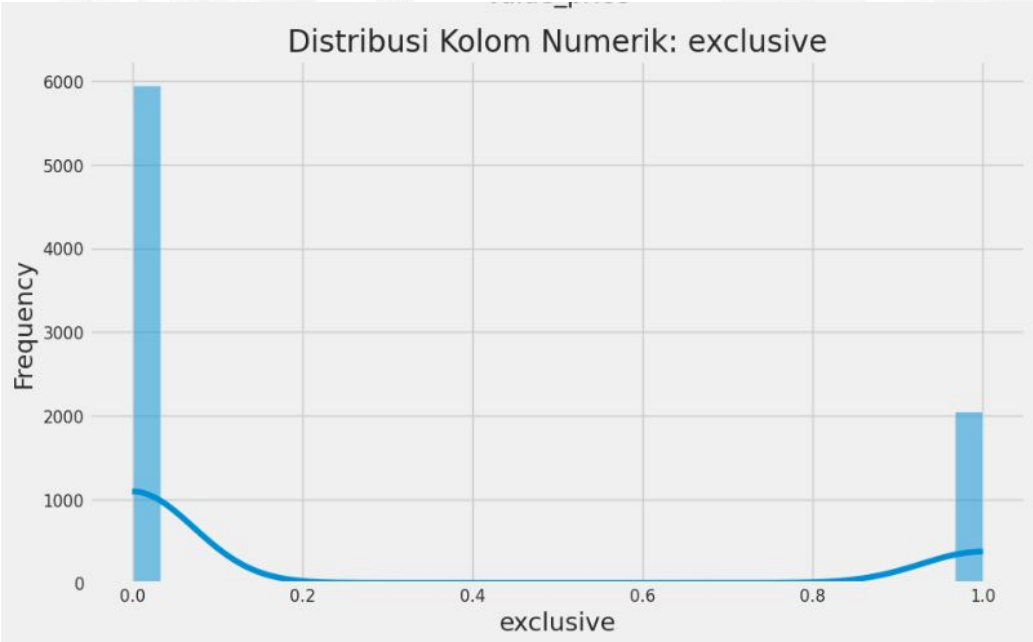
Karakteristik:

- Produk berharga rendah mendominasi data.
- Outlier di sisi harga tinggi kemungkinan adalah produk premium atau paket yang lebih mahal.

Transformasi yang Direkomendasikan: Log Transform atau Winsorizing dapat digunakan untuk menangani skewness yang tinggi dan mengurangi dampak outlier pada kedua kolom.

Catatan: Produk dengan harga sangat tinggi mungkin perlu analisis khusus, karena berbeda dari mayoritas produk lainnya.

2. Univariate Analysis : Exclusive (10 poin)



Kolom ini sepertinya merupakan variabel biner (0 dan 1) yang menunjukkan apakah suatu produk eksklusif atau tidak. Nilai 0 mendominasi, yang berarti sebagian besar produk tidak eksklusif. Kolom ini adalah variabel biner (0 dan 1), sehingga tidak ada skewness dalam arti tradisional. Namun, ada ketidakseimbangan karena sebagian besar nilai adalah 0 (produk non-eksklusif), dan hanya sebagian kecil adalah 1 (produk eksklusif). Tidak ada outlier karena ini adalah variabel biner.

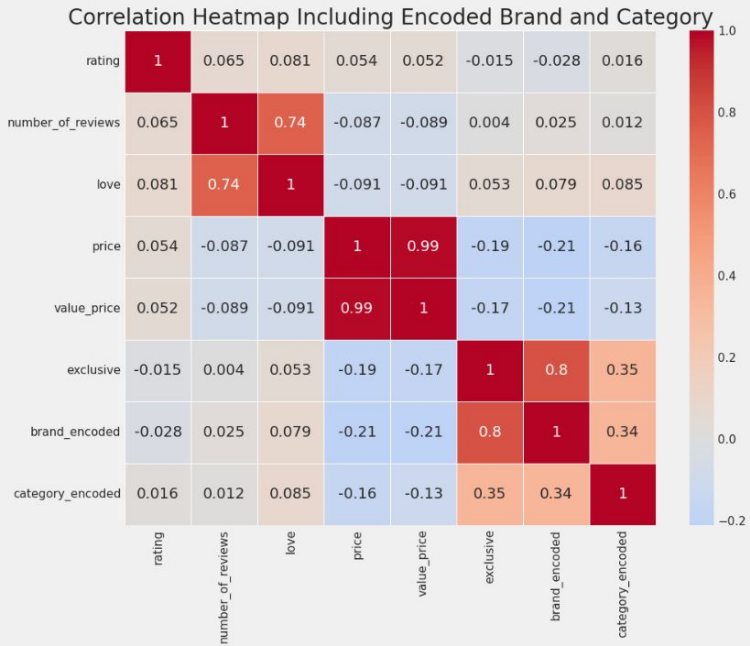
2. Univariate Analysis (10 poin)

Apa yang harus di-follow up saat data pre-processing :

1. **Grouping:** Pengelompokan kategori dan brand yang jumlah datanya sedikit untuk memperkecil variasi yang terlalu banyak.
2. **Handling Outliers:** Kolom price, value_price, love, dan number_of_reviews memiliki outlier yang mungkin perlu ditangani untuk mengurangi skewness.
3. **Normalization/Transformation:** Kolom numerik yang sangat skewed mungkin bisa dinormalisasi atau ditransformasi (misalnya, log transform) untuk membuat distribusi lebih normal.
4. **Focus on Popular Categories/Brands:** Untuk efisiensi, bisa fokus pada produk di kategori atau brand populer.
5. **Merging :** Karena Distribusi dan konten data yang serupa Value_price dan Price bisa di merge untuk mengurang redundansi data

3. Multivariate Analysis (15 poin)

A. Bagaimana korelasi antara masing-masing feature dan label. Kira-kira feature mana saja yang paling relevan dan harus dipertahankan?

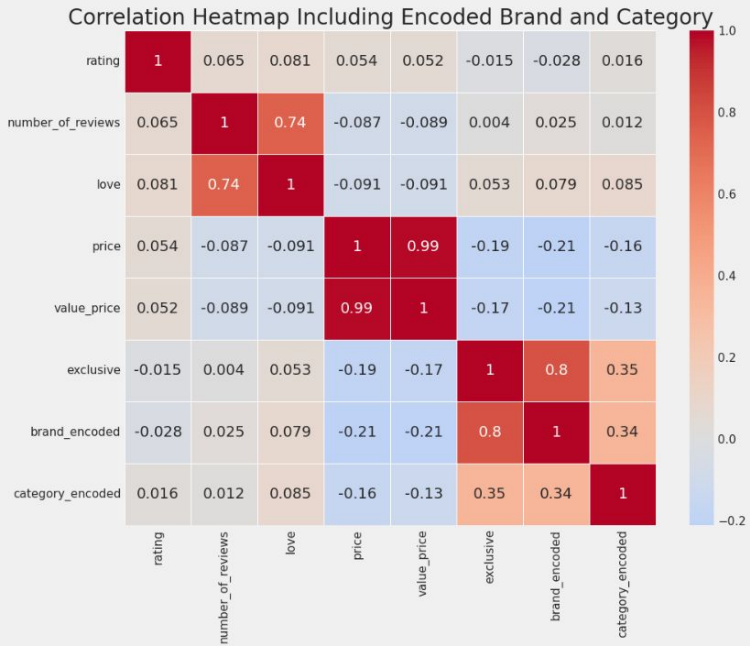


Korelasi dengan rating:

- **number_of_reviews:** Korelasi positif sangat lemah / Tidak Ada Korelasi (0.065) menunjukkan ada korelasi lemah bahwa produk dengan lebih banyak ulasan tidak berarti akan memiliki rating yang tinggi
- **love:** Korelasi positif sangat lemah/ Tidak Ada Korelasi (0.081) menunjukkan bahwa produk yang banyak disukai belum tentu akan memiliki rating tinggi
- **price dan value_price:** (0.054) Tidak ada korelasi signifikan dengan rating, menunjukkan bahwa harga tidak mempengaruhi rating produk.
- Rating tidak memiliki korelasi signifikan dengan fitur lain

3. Multivariate Analysis (15 poin)

A. Bagaimana korelasi antara masing-masing feature dan label. Kira-kira feature mana saja yang paling relevan dan harus dipertahankan?

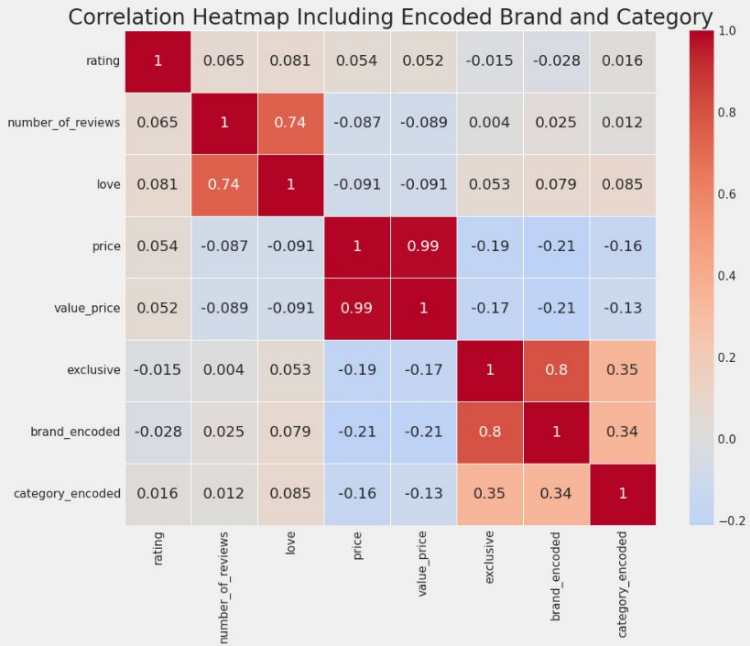


Korelasi dengan number_of_reviews:

- **love:** Korelasi positif (0.74) yang kuat, menunjukkan bahwa produk yang disukai cenderung memiliki lebih banyak ulasan. Ini adalah hubungan yang penting dan relevan.
- **rating:** Korelasi positif yang lebih rendah (0.065), menandakan bahwa ulasan banyak tidak selalu berkorelasi dengan rating tinggi.
- Tidak ada korelasi signifikan lain selain dengan love

3. Multivariate Analysis (15 poin)

A. Bagaimana korelasi antara masing-masing feature dan label. Kira-kira feature mana saja yang paling relevan dan harus dipertahankan?

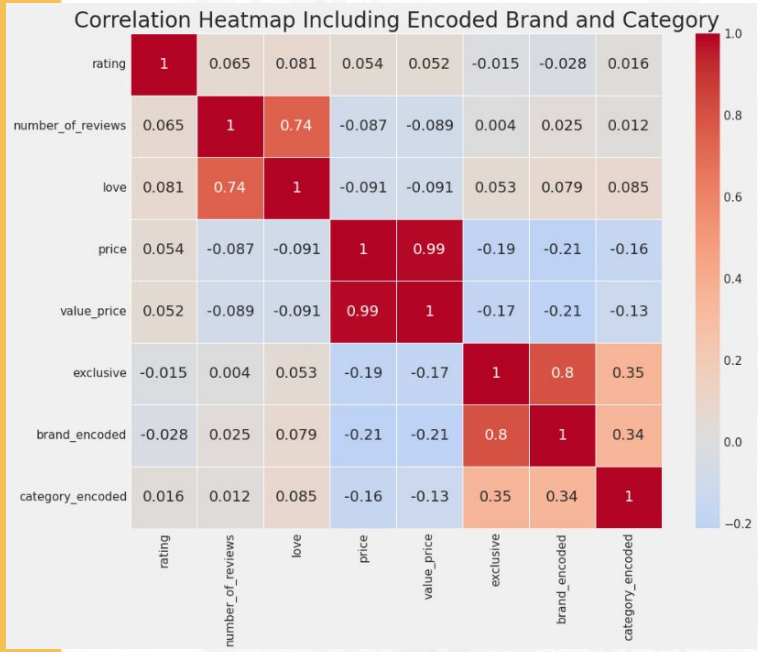


Korelasi dengan love:

- **number_of_reviews:** Korelasi positif tinggi (0.74) menunjukkan bahwa produk yang sangat disukai bisa memiliki banyak ulasan.
- **rating**: Korelasi positif (0.081) menunjukkan bahwa produk yang lebih disukai tidak berarti akan mendapatkan rating tinggi.
- Tidak ada korelasi signifikan lain selain dengan number_of_reviews

3. Multivariate Analysis (15 poin)

A. Bagaimana korelasi antara masing-masing feature dan label. Kira-kira feature mana saja yang paling relevan dan harus dipertahankan?

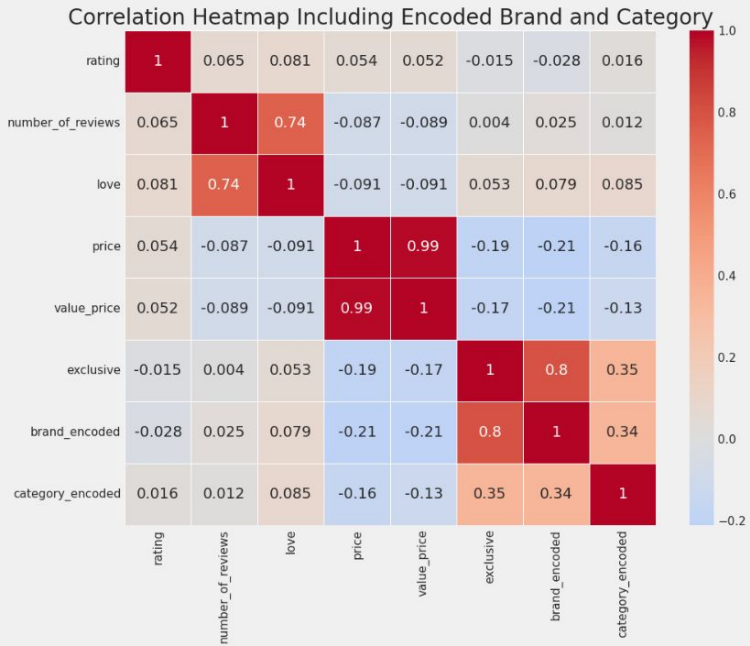


Korelasi dengan price dan value_price:

- **Korelasi Sangat Tinggi:** 'Price' dan 'Value Price' memiliki korelasi hampir identik (0.99), sehingga salah satu kolom dapat dihapus.
- **Korelasi Negatif Lemah dengan 'Exclusive':** 'Price' (-0.19) dan 'Value Price' (-0.17) menunjukkan bahwa produk eksklusif mungkin sedikit lebih murah, tetapi tidak signifikan.
- **Korelasi Negatif Lemah dengan 'Brand':** Keduanya memiliki korelasi -0.21, mengindikasikan bahwa harga beberapa brand ternama mungkin sedikit lebih murah, tetapi tidak signifikan.
- **Korelasi Negatif Lemah dengan 'Category':** 'Price' (-0.16) dan 'Value Price' (-0.13) menunjukkan beberapa kategori mungkin sedikit lebih murah, tetapi juga tidak signifikan sebagai penentu korelasi.

3. Multivariate Analysis (15 poin)

A. Bagaimana korelasi antara masing-masing feature dan label. Kira-kira feature mana saja yang paling relevan dan harus dipertahankan?

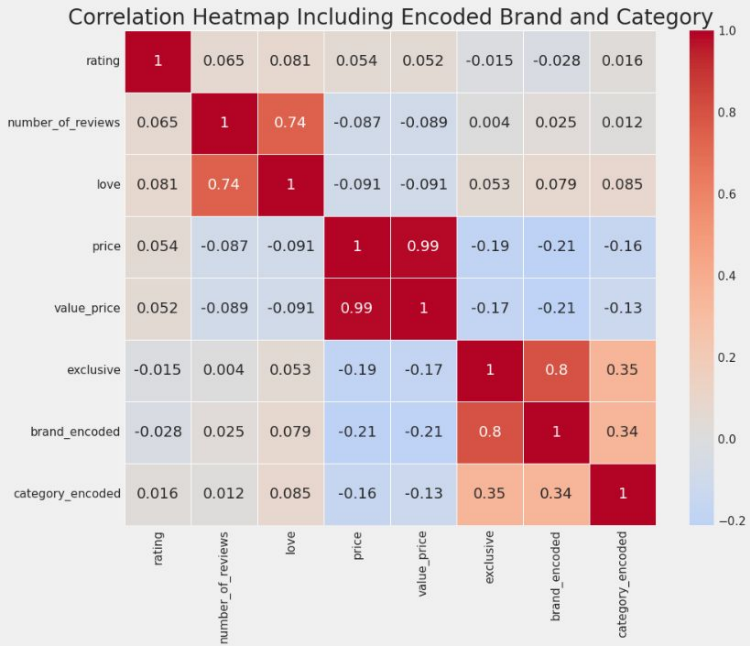


Korelasi dengan exclusive:

- **brand_encoded:** Korelasi positif tinggi (0.80) menunjukkan bahwa nama brand cenderung membuat sebuah produk eksklusif
- **category_encoded:** Korelasi positif sedang (0.35) menunjukkan bahwa kategori produk bisa jadi mempengaruhi ke eksklusifitas sebuah produk
- love dan rating tidak memiliki korelasi yang signifikan dengan exclusive.

3. Multivariate Analysis (15 poin)

A. Bagaimana korelasi antara masing-masing feature dan label. Kira-kira feature mana saja yang paling relevan dan harus dipertahankan?

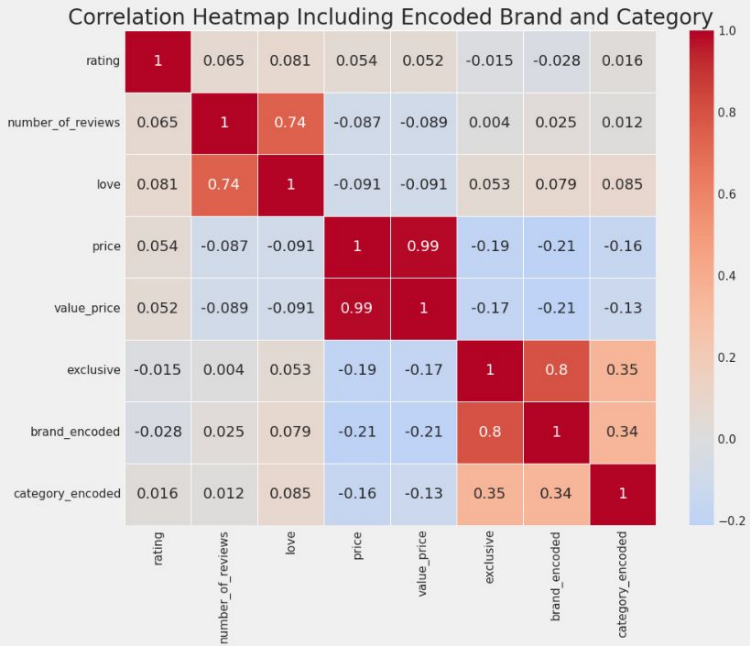


Korelasi dengan category_encoded:

- Category_encoded juga memiliki korelasi positif dengan exclusive (0.35), meskipun tidak sekuat brand_encoded. Ini menunjukkan bahwa kategori produk mungkin relevan untuk menentukan eksklusivitas sebuah produk.
- Brand_encoded memiliki korelasi sedang (0.34), menunjukkan bahwa ada beberapa keterkaitan antara brand dan kategori produk contohnya beberapa brand berfokus lebih banyak memiliki produk di kategori tertentu, namun tidak cukup kuat untuk menyebabkan redundansi.

3. Multivariate Analysis (15 poin)

A. Bagaimana korelasi antara masing-masing feature dan label. Kira-kira feature mana saja yang paling relevan dan harus dipertahankan?



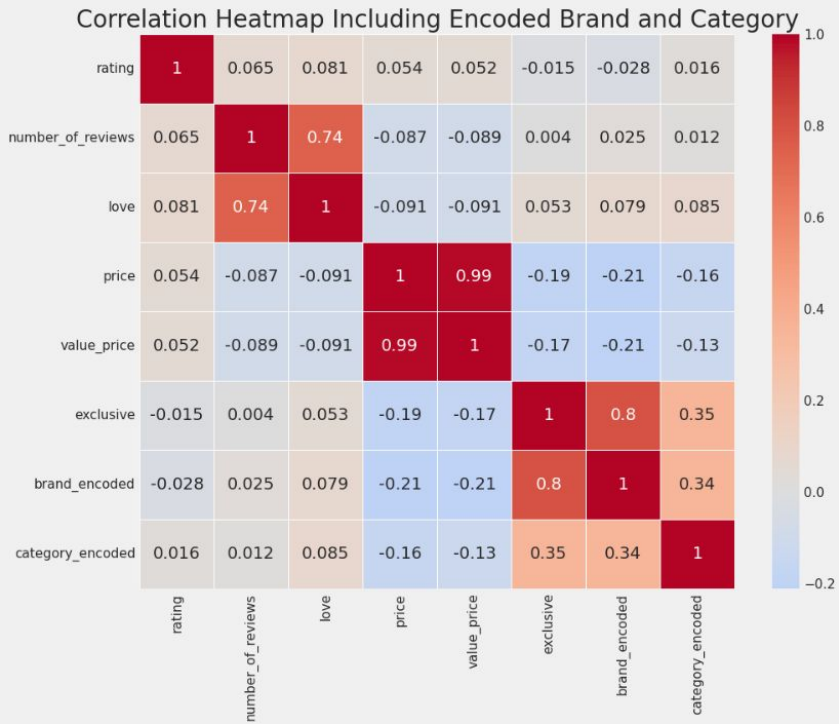
korelasi yang ditunjukkan oleh brand dan love memiliki pengaruh yang lebih besar terhadap eksklusivitas dan jumlah ulasan, sementara rating tidak memiliki korelasi signifikan dengan fitur lain. Korelasi dari price dan value price, menunjukkan hubungan yang hampir identik.

Fitur yang paling relevan dan harus dipertahankan:

- brand_encoded: Sangat relevan dengan eksklusivitas.
- category_encoded: Relevan meskipun tidak sekuat brand_encoded.
- number_of_reviews dan love: mereka memiliki hubungan kuat satu sama lain yang menunjukkan bahwa popularitas dan penilaian produk saling terkait.

3. Multivariate Analysis (15 poin)

B. Bagaimana korelasi antar-feature, apakah ada pola yang menarik? Apa yang perlu dilakukan terhadap feature itu?



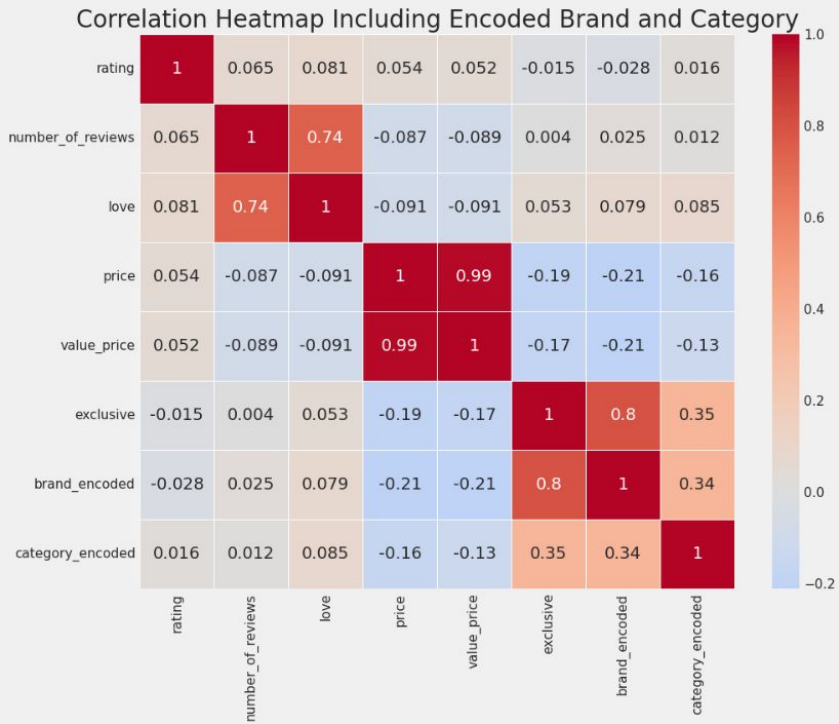
Price dan value_price: Korelasi sangat tinggi (0.99) menunjukkan bahwa mereka memberikan informasi yang hampir sama. Salah satu dari keduanya harus dihapus untuk mengurangi redundansi.

number_of_reviews dan love: Korelasi positif tinggi (0.74) menunjukkan bahwa produk yang banyak disukai juga cenderung memiliki banyak ulasan. Hal ini bisa memberikan informasi yang lebih baik dalam analisis produk lebih baik kedua fitur ini disimpan.

brand_encoded dan category_encoded: Korelasi sedang (0.34) menunjukkan bahwa ada hubungan korelasi sedang antara kategori dan merek. Ini bisa berarti bahwa merek tertentu mungkin lebih sering muncul dalam kategori tertentu.

3. Multivariate Analysis (15 poin)

B. Bagaimana korelasi antar-feature, apakah ada pola yang menarik? Apa yang perlu dilakukan terhadap feature itu?



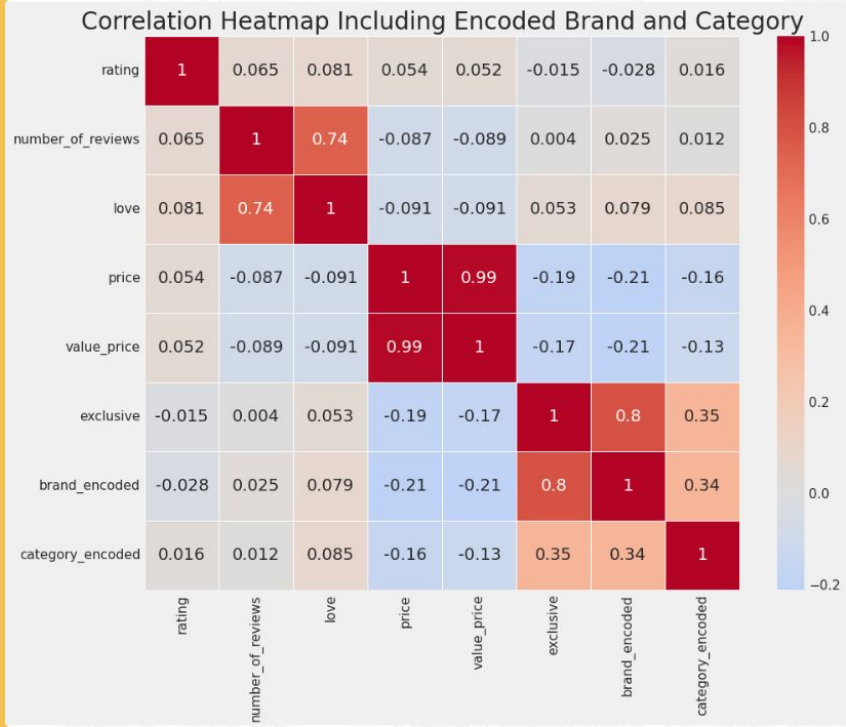
Brand_encoded dan exclusive: Korelasi Sangat Kuat (0.80), menunjukkan bahwa nama brand sangat relevan dalam menentukan apakah produk sebuah produk eksklusif atau bukan. Kedua fitur ini sangat dianjurkan untuk disimpan

Category_encoded dan exclusive (0.35): meskipun tidak serelevan brand_encoded. Ini menunjukkan bahwa beberapa kategori produk mungkin relevan untuk menentukan eksklusivitas sebuah produk.

Ada pola menarik antar korelasi dengan Category, Brand dan Exclusive yang mungkin saja menunjukkan bahwa brand dan category mungkin menentukan apakah sebuah produk eksklusif atau non eksklusif

3. Multivariate Analysis (15 poin)

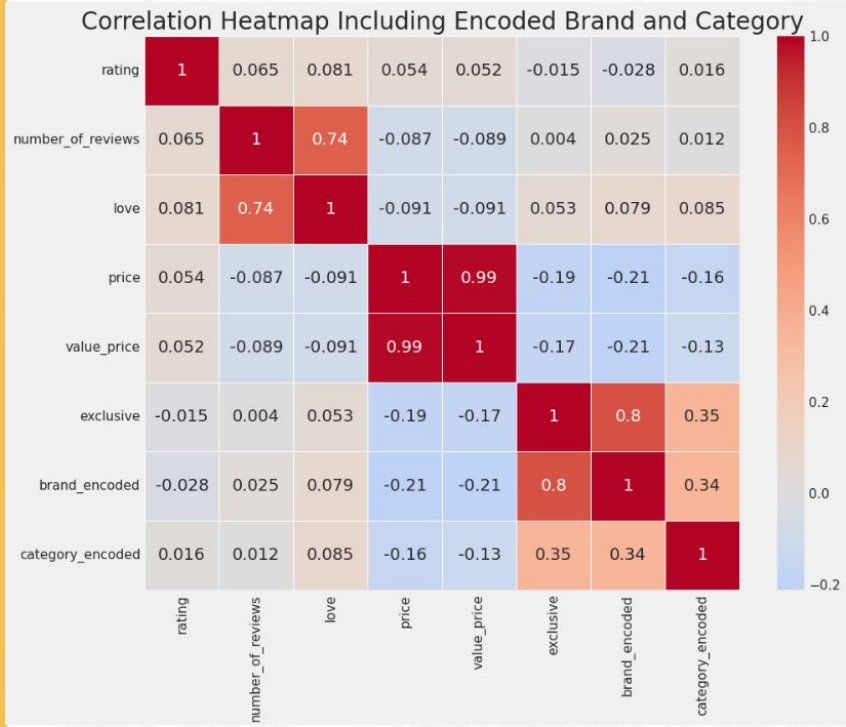
B. Bagaimana korelasi antar-feature, apakah ada pola yang menarik? Apa yang perlu dilakukan terhadap feature itu?



Category_encoded dan Brand_encoded dianjurkan untuk disimpan karena mempunyai korelasi dengan exclusive dan dengan satu sama lain selain itu fitur ini merupakan identias produk

3. Multivariate Analysis (15 poin)

B. Bagaimana korelasi antar-feature, apakah ada pola yang menarik? Apa yang perlu dilakukan terhadap feature itu?



Tindakan yang Direkomendasikan:

- Hapus Salah Satu dari price atau value_price: karena keduanya serupa dengan korelasi 0.99
- Pertahankan number_of_reviews dan love: Keduanya memberikan informasi berharga tentang popularitas produk dan ulasan pelanggan.
- Pertimbangkan untuk Mempertahankan Kategori dan Merek: Dengan mengingat hasil korelasi, kedua fitur ini berpotensi memberikan wawasan tentang eksklusivitas produk.

4. Data Cleansing (40 poin)

Lakukan pembersihan data, sesuai yang diajarkan di kelas, seperti:

A. Handle missing values : **Code**

```
from sklearn.preprocessing import LabelEncoder

# Target encoding for categorical columns: brand and category
# Replace NaN in category with a placeholder for consistent encoding
df['category'].fillna('Unknown', inplace=True)

# Calculate target-encoded values (mean of exclusive for each category in brand and category)
brand_target_encoded = df.groupby('brand')['exclusive'].transform('mean')
category_target_encoded = df.groupby('category')['exclusive'].transform('mean')

# Add encoded columns to the dataset
df['brand_encoded'] = brand_target_encoded
df['category_encoded'] = category_target_encoded

# Prepare numerical data for correlation heatmap
numeric_data_with_encoded = df[['rating', 'number_of_reviews', 'love', 'price', 'value_price', 'exclusive', 'brand_encoded', 'category_encoded']].dropna()

# Recalculate correlation matrix including encoded categorical columns
correlation_matrix_with_encoded = numeric_data_with_encoded.corr()

# Plot updated correlation heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix_with_encoded, annot=True, cmap="coolwarm", center=0, linewidths=0.5)
plt.title("Correlation Heatmap Including Encoded Brand and Category")
plt.show()
```

4. Data Cleansing (40 poin)

Lakukan pembersihan data, sesuai yang diajarkan di kelas, seperti:

A. Handle missing values: **output**

```
Missing values per column:
 id                   0
brand                 0
category             13
rating               95
number_of_reviews     9
love                 34
price                 8
value_price          17
exclusive             0
dtype: int64
Missing values in 'price' filled with median.
Missing values in 'number_of_reviews' filled with median.
Missing values in 'rating' filled with median.
Missing values in 'value_price' filled with median.
Missing values in 'love' filled with median.
Missing values in 'category' filled with 'Unknown'.
```

Cek missing values pada setiap kolom, kemudian menggunakan metode imputasi atau mengganti data yang missing dengan nilai median untuk kolom numerik dan value dominan untuk kolom kategorikal/non-numerik. Jika ada kolom kategorikal yang tidak memiliki nilai dominan yang jelas, nilai kosong diisi dengan kategori "Unknown".

4. Data Cleansing (40 poin)

B. Handle duplicate data

Code

```
duplicates = df.duplicated().sum()
print("Number of duplicate rows:", duplicates)

# Menghapus data duplikat jika ada
if duplicates > 0:
    data = df.drop_duplicates()
    print("Duplicate rows removed.")
else:
    print("No duplicate rows found.")
```

mengecek duplikat data dan menghapus data duplikat jika ada.

Output

```
Number of duplicate rows: 0
No duplicate rows found.
```

Tidak perlu di handle karena tidak ada data duplikat

4. Data Cleansing (40 poin)

C. Handle outliers

Code

```
# Fungsi untuk mendeteksi dan menghapus outliers menggunakan IQR
def remove_outliers(df, column):
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    outliers = df[(df[column] < lower_bound) | (df[column] > upper_bound)]
    if not outliers.empty:
        print(f"Outliers detected in '{column}':", len(outliers))
        return df[(df[column] >= lower_bound) & (df[column] <= upper_bound)]
    else:
        print(f"No outliers detected in '{column}'.")
        return df

# Memastikan hanya kolom dengan outliers yang di-handle
df = remove_outliers(df, 'price')
df = remove_outliers(df, 'number_of_reviews')
df = remove_outliers(df, 'love')
df = remove_outliers(df, 'value_price')
df = remove_outliers(df, 'rating')
df = remove_outliers(df, 'exclusive')
```

Mengidentifikasi outliers pada kolom numerik menggunakan metode Interquartile Range (IQR) yaitu ukuran penyebaran data yang mengabaikan nilai ekstrim. Jika ditemukan nilai yang berada di luar batas IQR, nilai tersebut dihapus dari dataset.

Cara ini diulang terus menerus sampai tidak ada lagi outlier karena itu output menunjukkan "No outliers detected in..."

Output yang diharapkan

```
No outliers detected in 'price'.
No outliers detected in 'number_of_reviews'.
No outliers detected in 'love'.
No outliers detected in 'value_price'.
No outliers detected in 'rating'.
No outliers detected in 'exclusive'.
```

4. Data Cleansing (40 poin)

D. Feature transformation: **Code**

```
import numpy as np

# Tentukan threshold untuk skewness
high_skew_threshold = 1
moderate_skew_threshold = 0.5


# Daftar kolom yang akan dikecualikan dari pengecekan
excluded_columns = ['id', 'exclusive_log', 'id_log_log','id_log','exclusive_log_log','exclusive_log_log_log', 'exclusive_log_log_log_log']

# Loop melalui kolom numerik di DataFrame, kecuali kolom yang dikecualikan
for col in df.select_dtypes(include=np.number).columns:
    if col not in excluded_columns:
        # Hitung skewness kolom
        skewness = df[col].skew()
        print(f"Skewness of '{col}': {skewness}")

        # Terapkan transformasi sesuai dengan tingkat skewness
        if abs(skewness) > high_skew_threshold:
            # Skewness tinggi, gunakan transformasi log
            df[f'{col}_log'] = np.log1p(df[col])
            print(f"Applied log transformation to '{col}'.")
        elif abs(skewness) > moderate_skew_threshold:
            # Skewness sedang, gunakan transformasi akar kuadrat atau akar kubik
            df[f'{col}_sqrt'] = np.sqrt(df[col])
            print(f"Applied square root transformation to '{col}'.")
```

Pengecekan Skewness:

mengecek skewness dari setiap kolom numerik dalam DataFrame untuk menentukan tingkat simetri distribusinya. Skewness dihitung menggunakan metode .skew() pada setiap kolom numerik.

Jika skewness mendekati 0, distribusi dianggap simetris; jika skewness lebih besar dari 1 atau lebih rendah dari -1 distribusi sangat skew.

4. Data Cleansing (40 poin)

D. Feature transformation: melakukan pengecekan skewness, menentukan threshold untuk skewness dan menentukan daftar kolom yang akan dikecualikan pada saat pengecekan, menerapkan transformasi sesuai dengan tingkat skewness

Threshold Skewness:

- Skewness Tinggi: |skewness| > 1 → Distribusi sangat skew. diterapkan transformasi log menggunakan np.log1p(). Metode ini mengurangi skewness secara signifikan, sehingga distribusi menjadi lebih simetris.
- Skewness Sedang: 0.5 < |skewness| ≤ 1 → Distribusi sedikit skew. diterapkan transformasi akar kuadrat (np.sqrt()), yang merupakan transformasi lebih ringan dibandingkan log. Ini mengurangi skewness tanpa mengubah distribusi secara drastis.
- Skewness Rendah: |skewness| ≤ 0.5 → Distribusi simetris, tidak perlu transformasi.

Pengecualian Kolom:
Kolom tertentu, seperti id dan exclusive_log, dikecualikan untuk menjaga relevansi dan menghindari pengolahan ulang data.

4. Data Cleansing (40 poin)

D. Feature transformation

```
Skewness of 'rating': 0.04624594180016424
Skewness of 'number_of_reviews': 0.9490536457328569
Applied square root transformation to 'number_of_reviews'.
Skewness of 'love': 0.7844861888197289
Applied square root transformation to 'love'.
Skewness of 'price': 0.660921552705658
Applied square root transformation to 'price'.
Skewness of 'value_price': 0.6322656023728316
Applied square root transformation to 'value_price'.
Skewness of 'exclusive': 1.0518806526185684
Applied log transformation to 'exclusive'.
```

Kolom 'rating'
- Skewness: 0.046
- Tindakan: Tidak dilakukan transformasi karena distribusi sudah simetris.

Kolom 'number_of_reviews'
- Skewness: 0.949 (Skewness sedang)
- Tindakan: Transformasi akar kuadrat diterapkan untuk mengurangi skewness.

Kolom 'love'
- Skewness: 0.784 (Skewness sedang)
- Tindakan: Transformasi akar kuadrat diterapkan untuk mengurangi skewness.

4. Data Cleansing (40 poin)

D. Feature transformation

```
Skewness of 'rating': 0.04624594180016424
Skewness of 'number_of_reviews': 0.9490536457328569
Applied square root transformation to 'number_of_reviews'.
Skewness of 'love': 0.7844861888197289
Applied square root transformation to 'love'.
Skewness of 'price': 0.660921552705658
Applied square root transformation to 'price'.
Skewness of 'value_price': 0.6322656023728316
Applied square root transformation to 'value_price'.
Skewness of 'exclusive': 1.0518806526185684
Applied log transformation to 'exclusive'.
```

Kolom 'price'
- Skewness: 0.661 (Skewness sedang)
- Tindakan: Transformasi akar kuadrat diterapkan untuk mengurangi skewness.

Kolom 'value_price'
- Skewness: 0.632 (Skewness sedang)
- Tindakan: Transformasi akar kuadrat diterapkan untuk mengurangi skewness.

Kolom 'exclusive'
- Skewness: 1.052 (Skewness tinggi)
- Tindakan: Transformasi log diterapkan untuk mengurangi skewness yang signifikan.

4. Data Cleansing (40 poin)

D. Feature transformation: Metode Yang Digunakan

Metode Transformasi

- Log Transform (np.log1p()): Digunakan untuk mengurangi skewness tinggi pada kolom dengan skewness lebih besar dari 1. Metode ini efektif dalam menangani skewness yang ekstrem.
- Square Root Transform (np.sqrt()): Digunakan untuk mengurangi skewness sedang pada kolom dengan skewness antara 0.5 dan 1. Transformasi ini lebih ringan dibandingkan log, sehingga cocok untuk distribusi yang sedikit skew.

4. Data Cleansing (40 poin)

D. Feature transformation

```
Skewness of 'rating': 0.04624594180016424
Skewness of 'number_of_reviews': 0.9490536457328569
Applied square root transformation to 'number_of_reviews'.
Skewness of 'love': 0.7844861888197289
Applied square root transformation to 'love'.
Skewness of 'price': 0.660921552705658
Applied square root transformation to 'price'.
Skewness of 'value_price': 0.6322656023728316
Applied square root transformation to 'value_price'.
Skewness of 'exclusive': 1.0518806526185684
Applied log transformation to 'exclusive'.
```

Ringkasan Transformasi

- Akar Kuadrat diterapkan pada kolom dengan skewness sedang untuk menyeimbangkan distribusi tanpa perubahan drastis.
- Transformasi Log diterapkan pada kolom dengan skewness tinggi untuk mengurangi ketidaksimetrisan yang lebih ekstrem.

4. Data Cleansing (40 poin)

E. Feature encoding : **Code**

Mengonversi kolom kategorikal brand dan category menggunakan target encoding untuk menyederhanakan representasi kolom dan menambah relevansi informasi. Target encoding dilakukan berdasarkan rata-rata nilai label (exclusive) pada setiap kategori.

```
# Mengecek tipe data kategorikal
categorical_cols = df.select_dtypes(include=['object']).columns
print("Categorical columns:", categorical_cols)

# Lakukan target encoding untuk kolom 'brand' dan 'category' jika mereka ada
if 'brand' in categorical_cols:
    df['brand_encoded'] = df.groupby('brand')['exclusive'].transform('mean')
    print("Encoded 'brand' with target encoding.")

if 'category' in categorical_cols:
    df['category_encoded'] = df.groupby('category')['exclusive'].transform('mean')
    print("Encoded 'category' with target encoding.")

# Drop kolom asli setelah encoding jika sudah tidak dibutuhkan
df = df.drop(columns=['brand', 'category'])
```

4. Data Cleansing (40 poin)

E. Feature encoding : **output**

```
Categorical columns: Index(['brand', 'category'], dtype='object')
Encoded 'brand' with target encoding.
Encoded 'category' with target encoding.
```

Hasil:
Kolom brand dan category dikonversi menjadi nilai rata-rata eksklusivitas masing-masing kategori. Hal ini membantu mengurangi dimensi data dan membuat fitur ini lebih informatif dalam proses modeling. Setelah encoding, kolom asli brand dan category dihapus untuk menghindari redundansi.

4. Data Cleansing (40 poin)

F. Handle class imbalance : **Code**

```
# Mengecek distribusi kelas pada label
class_distribution = df['exclusive'].value_counts()
print("Class distribution before handling imbalance:\n", class_distribution)

# Menggunakan SMOTE jika ada ketidakseimbangan kelas
if class_distribution.min() / class_distribution.max() < 0.5:
    from imblearn.over_sampling import SMOTE
    from sklearn.model_selection import train_test_split

    # Pisahkan fitur dan label
    X = df.drop(columns=['exclusive'])
    y = df['exclusive']

    # Split data sebelum melakukan SMOTE
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    # Menggunakan SMOTE untuk oversample data training
    smote = SMOTE(random_state=42)
    X_train_res, y_train_res = smote.fit_resample(X_train, y_train)

    print("Class distribution after SMOTE:\n", y_train_res.value_counts())
else:
    print("No class imbalance detected.")
```

Mengecek distribusi kelas pada label exclusive. Jika distribusi kelas tidak seimbang, digunakan SMOTE (Synthetic Minority Over-sampling Technique) untuk menyeimbangkan data training.

Penanganan ketidakseimbangan kelas umumnya dilakukan pada kolom target (label) atau fitur kategorikal yang mewakili kelas atau kategori tertentu.

4. Data Cleansing (40 poin)

F. Handle class imbalance

```
Class distribution before handling imbalance:
 exclusive
0    2143
1     782
Name: count, dtype: int64
Class distribution after SMOTE:
 exclusive
0    1715
1    1715
Name: count, dtype: int64
```

Hasil:

Terdapat Class Imbalance pada label exclusive. Menggunakan SMOTE pada data training berhasil menyeimbangkan jumlah sampel antara kedua kelas. Penyeimbangan kelas dapat meningkatkan performa model untuk menghindari bias terhadap kelas mayoritas.

5. Feature Engineering (30 poin)

Cek feature yang ada sekarang, lalu lakukan:

A. Feature selection (membuang feature yang kurang relevan atau redundan)

```
# Import libraries
import pandas as pd

# Drop 'value_price' due to redundancy with 'price'
df = df.drop(columns=['value_price'])
```

Hapus Salah Satu dari price atau value_price:

Alasan: Keduanya memiliki korelasi hampir sempurna (0.99), menunjukkan bahwa mereka memberikan informasi yang sangat mirip. Untuk mengurangi redundansi, kita akan menghapus salah satu.

5. Feature Engineering (30 poin)

B. Feature extraction : **Code** (membuat feature baru dari feature yang sudah ada)

```
# Mengisi nilai NaN pada 'number_of_reviews' untuk menghindari pembagian nol
df['number_of_reviews'] = df['number_of_reviews'].replace(0, 1)

# 1. Average Rating per Review
# Hitung rata-rata rating per ulasan untuk setiap produk
df['avg_rating_per_review'] = df['rating'] / df['number_of_reviews']

# 2. Love-to-Review Ratio
# Hitung rasio "love" terhadap jumlah ulasan
df['love_to_review_ratio'] = df['love'] / df['number_of_reviews']

# 3. Price per Category Mean
# Hitung rata-rata harga per kategori
category_price_mean = df.groupby('category_encoded')['price'].transform('mean')
# Buat fitur baru dengan selisih harga produk terhadap rata-rata harga di kategorinya
df['price_vs_category_mean'] = df['price'] - category_price_mean

# 4. Brand Exclusivity Score
# Hitung skor eksklusivitas per brand berdasarkan rata-rata nilai 'exclusive'
brand_exclusivity_score = df.groupby('brand_encoded')['exclusive'].transform('mean')
df['brand_exclusivity_score'] = brand_exclusivity_score

# Cek hasilnya
df[['avg_rating_per_review', 'love_to_review_ratio', 'price_vs_category_mean', 'brand_exclusivity_score']].head()
```

5. Feature Engineering (30 poin)

B. Feature extraction : **Output** (membuat feature baru dari feature yang sudah ada)

	avg_rating_per_review	love_to_review_ratio	price_vs_category_mean	brand_exclusivity_score
0	0.108696	0.000000	25.000000	0.950980
23	0.160714	46.428571	-22.840226	0.065789
24	0.095745	117.021277	-22.840226	0.065789
25	0.142857	107.142857	36.159774	0.000000
27	0.128571	22.542857	-5.800000	0.065789

Average Rating per Review:
Tujuan: Untuk menilai apakah produk yang populer (banyak ulasan) juga memiliki penilaian rata-rata yang baik.

Love-to-Review Ratio:
Tujuan: Ini dapat membantu menilai apakah produk yang disukai juga mendapatkan banyak ulasan, atau sebaliknya.

Price per Category Mean:
Tujuan: Ini dapat membantu mengidentifikasi apakah produk lebih mahal atau lebih murah dari rata-rata harga di kategorinya, yang mungkin berpengaruh pada eksklusivitas.

Brand Exclusivity Score:
Tujuan: Membantu menilai kecenderungan eksklusivitas suatu merek dalam dataset.

5. Feature Engineering (30 poin)

C. Tuliskan minimal 4 feature tambahan (selain yang sudah tersedia di dataset) yang mungkin akan sangat membantu membuat performansi model semakin bagus (ini hanya ide saja, untuk menguji kreativitas teman-teman, tidak perlu benar-benar dicari datanya dan tidak perlu diimplementasikan)

1. **Seasonal Popularity:**
Fitur yang menunjukkan popularitas musiman dari produk, misalnya berdasarkan data penjualan selama periode tertentu seperti musim liburan, hari besar, atau musim tertentu.

Tujuan Fitur: Menilai apakah produk tertentu memiliki peningkatan popularitas di musim atau waktu tertentu, yang dapat membantu memahami relevansi produk dengan periode waktu.

2. Customer Loyalty Score:
Fitur menggunakan data pelanggan untuk menghitung tingkat loyalitas, misalnya berdasarkan jumlah produk yang sering dibeli kembali atau tingkat engagement dari pelanggan yang sama.

Tujuan Fitur: Menunjukkan bahwa produk dengan tingkat loyalitas tinggi mungkin lebih bernilai bagi konsumen setia.

5. Feature Engineering (30 poin)

C. Tuliskan minimal 4 feature tambahan (selain yang sudah tersedia di dataset) yang mungkin akan sangat membantu membuat performansi model semakin bagus (ini hanya ide saja, untuk menguji kreativitas teman-teman, tidak perlu benar-benar dicari datanya dan tidak perlu diimplementasikan)

3. Marketing Budget or Advertisement Exposure:
Fitur yang menunjukkan berapa banyak anggaran yang dikeluarkan untuk iklan produk atau eksposur iklan dari masing-masing merek.

Tujuan Fitur: Mengukur apakah popularitas atau eksklusivitas suatu produk dipengaruhi oleh strategi pemasaran dan anggaran yang dialokasikan untuk iklan.

4. Influencer Endorsements:
Jumlah atau tingkat endorsement dari influencer terhadap produk tertentu.

Tujuan Fitur: Mengukur apakah endorsement dari influencer berpengaruh pada eksklusivitas, popularitas, atau penilaian konsumen terhadap produk.

Rakamin Academy

Terima kasih!