Amazon Kindle Books Sales Analysis 2023

TALITHA ASMATA VERILLA

Background

Dalam *project* ini, portal *marketplace* Amazon ingin melihat review penjualan untuk *Kindle e-books* tahun 2023. Amazon juga ingin mencari tahu faktor yang berpotensi untuk meningkatkan penjualan dan akan menerapkannya di tahun depan.

Terdapat lebih dari 130.000 baris data dan 16 kolom dalam dataset ini. Berikut daftar dan penjelasan dari setiap kolom:

- asin = product ID dari Amazon
- title = judul buku
- author = penulis buku
- soldBy = penjual buku
- imgUrl = URL gambar sampul buku
- productURL = URL buku
- stars = rating rata-rata buku. Jika bernilai 0, belum ada rating yang ditambahkan
- Reviews = jumlah review. Jika bernilai 0, belum ada review yang ditambahkan
- Price = harga buku
- isKindleUnlimited = apakah buku tersebut tersedia di Kindle Unlimited
- category_id = serial ID yang ditetapkan pada kategori buku ini
- isBestSeller = apakah buku berstatus 'Best Seller'
- isEditorsPick = apakah buku berstatus 'Editor's Pick'
- isGoodReadsChoice = apakah buku berstatus 'Good Reads Choice'
- publishedDate = tanggal publikasi buku
- category_name = kategori buku

Worklow


Data Collecting
Data Cleansing
Exploration
Insights
Recommendation

Data Collecting Method

Dalam *project* ini, metode pengambilan data yang dilakukan, yaitu dengan menggunakan pengumpulan data sekunder (*secondary data collection*). Data diambil dari portal free data source, **kaggle**.

Dataset source link: **https://www.kaggle.com/datasets/asaniczka/amazon-kindle-books-dataset-2023-130k-books/data**

Data Cleansing

Dalam proses *data cleansing*, terdapat 4 tahap yang ditangani, yakni:

- Identifikasi *missing value*
- *Handling duplicate data*
- Hapus Outlier
- *Handling Inconsistent Format*

Proses *data cleansing* dilakukan menggunakan ***python***.

Data Cleansing: Missing Value

Terdapat 16 kolom dalam dataset. Untuk menghindari redundansi data, **kolom 'imgUrl' dan 'productURL' dihapus** karena tidak diperlukan dalam proses analisis.

```
# menampilkan daftar kolom di dataset

kindle_data.columns
```

```
Index(['asin', 'title', 'author', 'soldBy', 'imgUrl', 'productURL', 'stars',
       'reviews', 'price', 'isKindleUnlimited', 'category_id', 'isBestSeller',
       'isEditorsPick', 'isGoodReadsChoice', 'publishedDate', 'category_name'],
      dtype='object')
```

```
# menghapus kolom yang tidak diperlukan

kindle_data = kindle_data.drop(['imgUrl', 'productURL'], axis=1)
kindle_data.head(3)
```

Dalam proses pengecekan *missing value*, terdapat **3 kolom yang memiliki *missing value***. Berikut daftar kolom dan presentase *missing value* dalam dataset:

```
# Persentase missing value pada tiap kolom
kindle_data.isnull().sum().sort_values(ascending=False)/len(kindle_data)*100
```

```
publishedDate        36.825893
soldBy                6.936785
author                0.319304
asin                  0.000000
title                 0.000000
stars                 0.000000
reviews               0.000000
price                 0.000000
isKindleUnlimited     0.000000
category_id           0.000000
isBestSeller          0.000000
isEditorsPick         0.000000
isGoodReadsChoice     0.000000
category_name         0.000000
dtype: float64
```

Data Cleansing: Missing Value

Pada kolom **soldBy**, terlihat dari presentase distribusinya, 'Amazon.com Services LLC' memiliki dominasi yang signifikan (sekitar 68.4%) dibanding penjual buku yang lain. Oleh karena itu, kolom ini akan diinput nilai **modus**nya untuk mengisi *missing value*.

```
# Hitung distribusi persentase soldBy
soldby_distribution = kindle_data['soldBy'].value_counts(normalize=True) * 100

# Tampilkan hasil
print(soldby_distribution)
```

```
soldBy
Amazon.com Services LLC                  68.426321
Random House LLC                          4.740492
Hachette Book Group                       3.823394
Penguin Group (USA) LLC                   3.540030
HarperCollins Publishers                  3.494014
Macmillan                                 2.540587
Simon and Schuster Digital Sales Inc      2.459857
Penguin Random House Publisher Services   1.929458
JOHN WILEY AND SONS INC                   1.926229
Simon & Schuster Digital Sales Inc.       1.411168
Pearson Education,Inc.                    1.059183
HarperCollins Publishing                  0.913869
```

Untuk kolom **author**, karena kolom ini penting dan presentase *missing value* nya kecil, menghapus data yang terdapat *missing value* nya bukan lah hal yang tepat. Maka, hal yang dilakukan,yaitu **mengisi nya dengan 'Unknown'**

Untuk kolom **publishedDate**, karena kebutuhan analisis tidak membutuhkan analisis tren waktu dan hal yang berhubungan dengan kolom ini, maka kolom ini akan **dihapus**.

```
# Hapus kolom publishedDate
kindle_data = kindle_data.drop(['publishedDate'], axis=1)

# Input missing value 'soldBy' dengan nilai modus
soldBy_mode = kindle_data['soldBy'].mode()[0]
kindle_data['soldBy'] = kindle_data['soldBy'].fillna(soldBy_mode)

# Input missing value author dengan "Unknown"
kindle_data['author'] = kindle_data['author'].fillna("Unknown")
```

Data Cleansing: Handling Duplicate

Dalam proses mengecek data duplikat, terdapat empat kali cara pengecekan, yaitu:

- Pengecekan berdasarkan kolom asin
- Pengecekan data yang dikategorikan sebagai data duplikat namun berbeda kode asin
- Pengecekan data yang dikategorikan sebagai data duplikat dengan jumlah reviews yg berbeda
- Pengecekan data yang dikategorikan sebagai data duplikat dengan rerata stars yg berbeda

Alasan mengapa pengecekan data duplikat tidak hanya dilakukan berdasarkan asin yang merupakan product ID buku, dikarenakan buku bisa saja sama namun di publikasikan untuk versi atau edisi terbaru.

Data Cleansing: Handling Duplicate

- **Pengecekan berdasarkan kolom asin**

  Kolom asin yang merupakan product ID yang diberikan Amazon biasanya bersifat *unique*. Pengecekan ini dilakukan untuk mencegah adanya redundansi data karena salah penginputan data.

```
# Menyeleksi data yang dikategorikan sebagai data duplikat berdasarkan kolom asin
duplicate_asin = kindle_data[kindle_data.duplicated(subset='asin', keep=False)]
print(f"Jumlah data duplikat berdasarkan kolom asin = {len(duplicate_asin)}")
```

```
Jumlah data duplikat berdasarkan kolom asin = 0
```

  Dari hasil terlihat bahwa tidak ada data dengan ID asin yang sama.

Data Cleansing: Handling Duplicate

- **Pengecekan data yang dikategorikan sebagai data duplikat namun berbeda code asin**

```
# Menyeleksi data yang dikategorikan sebagai data duplikat namun berbeda code asin
duplicate_non_asin = kindle_data[kindle_data.duplicated(subset=['title', 'author', 'soldBy', 'stars','reviews', 'price',
                                                                 'isKindleUnlimited', 'category_id', 'isBestSeller','isEditorsPi
                                                                 'isGoodReadsChoice', 'category_name'], keep=False)]
print(f"Jumlah data duplikat namun berbeda code asin = {len(duplicate_non_asin)}")
```

```
Jumlah data duplikat namun berbeda code asin = 12
```

Dari hasil terlihat sejumlah 12 data yang memiliki kode asin yang berbeda namun isi kolom lain yang sama. Maka, data duplikat ini harus dihapus dengan menggunakan **drop_duplicates**.

```
# Menghapus data duplikat
kindle_data.drop_duplicates(subset=['title', 'author', 'soldBy', 'stars',
       'reviews', 'price', 'isKindleUnlimited', 'category_id', 'isBestSeller',
       'isEditorsPick', 'isGoodReadsChoice', 'category_name'], keep='first',inplace=True, ignore_index=True)
```

Data Cleansing: Handling Duplicate

- **Pengecekan data yang dikategorikan sebagai data duplikat dengan jumlah reviews yang berbeda**

```
# Menyeleksi data yang dikategorikan sebagai data duplikat dengan jumlah reviews yg berbeda
duplicate_reviews = kindle_data[kindle_data.duplicated(subset=['title', 'author','soldBy','stars', 'price', 'category_name'],
                                                       keep=False)]
print(f"Jumlah data duplikat dengan jumlah reviews yg berbeda = {len(duplicate_reviews)}")
```

```
Jumlah data duplikat dengan jumlah reviews yg berbeda = 4
```

Hasil menunjukan terdapat 4 data dengan isi kolom yang sama namun berbeda jumlah reviews. Untuk menangani ini, menghapus salah satu data dengan drop_duplicates bukanlah hal yang tepat karena akan mempengaruhi nilai agregasi reviews saat di analisis. Maka, hal yang dilakukan yaitu mengelompokkannya dengan menggunakan **groupby()** dan melakukan agregasi dengan menjumlahkan reviews untuk data yang sama.

```
# Agregasi dengan groupby
kindle_data = kindle_data.groupby(
    ['title', 'author', 'soldBy', 'stars', 'price', 'category_name'],
    as_index=False).agg({
    'reviews': 'sum',
    'asin': 'first',
    'isKindleUnlimited': 'first',
    'category_id': 'first',
    'isBestSeller': 'first',
    'isEditorsPick': 'first',
    'isGoodReadsChoice': 'first',})
```

Data Cleansing: Handling Duplicate

- **Pengecekan data yang dikategorikan sebagai data duplikat dengan rerata stars yang berbeda**

```
# Menyeleksi data yang dikategorikan sebagai data duplikat dengan rerata stars yg berbeda
duplicate_stars = kindle_data[kindle_data.duplicated(subset=['title', 'author','soldBy','reviews', 'price', 'category_name'],
                                                     keep=False)]
print(f"Jumlah data duplikat dengan rerata stars yg berbeda = {len(duplicate_stars)}")
```

```
Jumlah data duplikat dengan rerata stars yg berbeda = 100
```

Terdapat 100 data duplikat dengan rerata stars (ratings) yang berbeda. Karena kolom stars merupakan hasil dari rerata yang diberikan konsumen, jika kita langsung mengagregasi menggunakan groupby seperti cara menangani duplikat dengan jumlah reviews yang berbeda, hasilnya tidak akan selalu valid (contoh, buku dengan judul 'Writing about Writing‘ memiliki dua data dengan ratings 0 dan 4.1 jika di rata-ratakan akan bernilai 2.05). Untuk menghindari hal ini, kita **ubah data dengan kolom stars yang bernilai 0 menjadi NaN** supaya data tidak salah hitung saat dihitung mean nya baru kita agregasikan menggunakan **groupby**.

```
# Ubah data yg kolom stars nya 0 menjadi NaN supaya data tidak salah hitung saat dihitung mean nya
kindle_data['stars'] = kindle_data['stars'].replace(0, np.nan)

# Agregasi dengan groupby
kindle_data = kindle_data.groupby(
    ['title', 'author','soldBy','reviews', 'price', 'category_name'],
    as_index=False).agg({
    'stars': 'mean',
    'asin': 'first',
    'isKindleUnlimited': 'first',
    'category_id': 'first',
    'isBestSeller': 'first',
    'isEditorsPick': 'first',
    'isGoodReadsChoice': 'first',})
```

Data Cleansing: Outlier

- **Cek terlebih dahulu statistik deskriptif dari dataset**

```
# Cek statistik deskriptif setelah menangani duplicate value
kindle_data.describe()
```

	reviews	price	stars	category_id
count	133044.000000	133044.000000	133044.000000	133044.000000
mean	887.762627	15.116903	4.404590	16.286642
std	5105.972412	22.235318	0.744177	8.418266
min	0.000000	0.000000	0.000000	1.000000
25%	0.000000	4.990000	4.400000	9.000000
50%	4.000000	9.990000	4.500000	16.000000
75%	366.000000	14.990000	4.700000	23.000000
max	618227.000000	682.000000	5.000000	31.000000

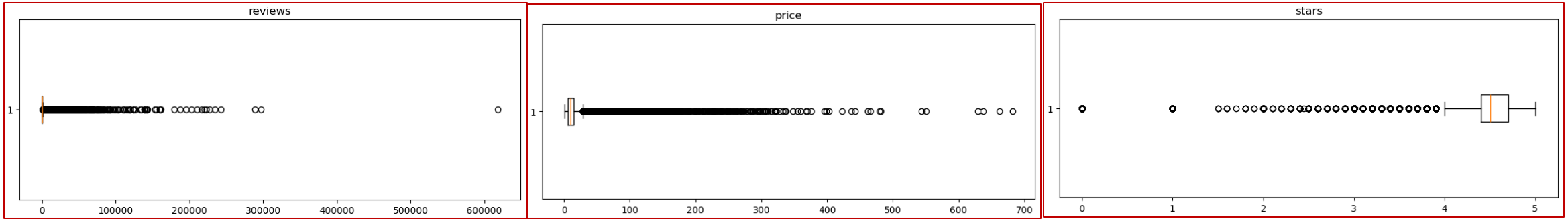
Dari hasil statistik deskriptif terlihat nilai minimum kolom price bernilai 0. Untuk mendapatkan total penjualan tiap buku, kita akan mengalikan kolom reviews dan price. Oleh karena itu, kita akan **mengecualikan data dengan kolom price yang bernilai 0**.

```
# hapus data dengan nilai price = 0
before = len(kindle_data)
kindle_data = kindle_data[kindle_data['price'] != 0]
after = len(kindle_data)
print(f"{before - after} baris dengan price = 0 telah dihapus.")
```

```
4063 baris dengan price = 0 telah dihapus.
```

Data Cleansing: Outlier

- **Cek visualisasi outlier nya dengan menggunakan boxplot dan presentase outliernya.**




```
for col in outlier_columns:

    outlier = detect_outliers_iqr(kindle_data[col])

    print("number of outliers in column", f"'{str(col)}'", "is", len(outlier))
    print("percentage of outliers in column",
          f"'{str(col)}'", "is", np.round(len(outlier)*100/len(kindle_data),2),"%")
    print()
```

```
lower: -544.5 upper: 907.5
number of outliers in column 'reviews' is 20112
percentage of outliers in column 'reviews' is 15.59 %

lower: -7.51 upper: 28.490000000000002
number of outliers in column 'price' is 14516
percentage of outliers in column 'price' is 11.25 %

lower: 3.9500000000000006 upper: 5.15
number of outliers in column 'stars' is 7018
percentage of outliers in column 'stars' is 5.44 %
```

Hasil visualisasi dan presentase menunjukan nilai ***outlier*** yang cukup besar, terutama pada kolom **price** dan **reviews**.

Data Cleansing: Outlier

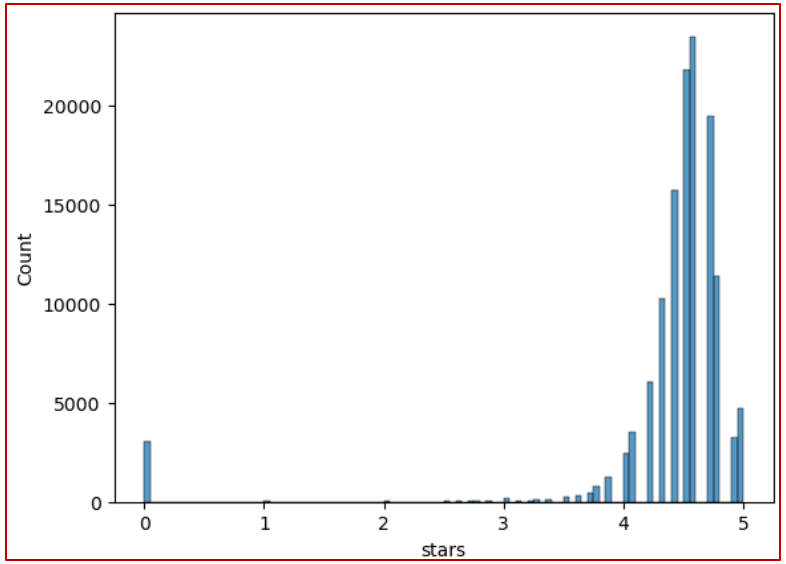
```
reviews_is_0 = len(kindle_data[kindle_data["reviews"] == 0])

print(f"Presentase data dengan jumlah reviews = 0 : {np.round(reviews_is_0*100/len(kindle_data),2)}%")
```

```
Presentase data dengan jumlah reviews = 0 : 48.23%
```

Untuk nilai **reviews = 0**, karena presentase nya besar, menghapusnya akan mempengaruhi analisis. Maka akan dibiarkan dan nantinya akan dipakai untuk analisis lebih lanjut.

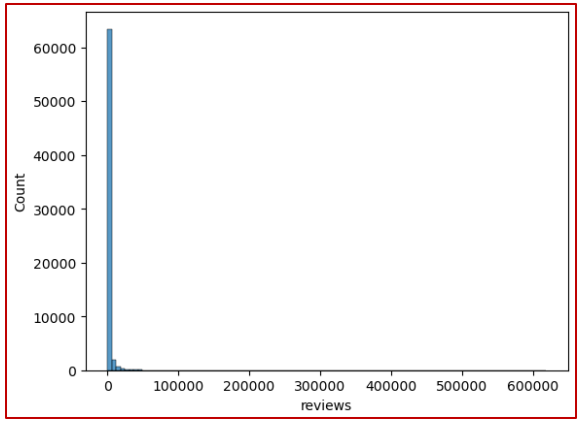
```
lower: 3.9500000000000006 upper: 5.15
number of outliers in column 'stars' is 7018
percentage of outliers in column 'stars' is 5.44 %
```

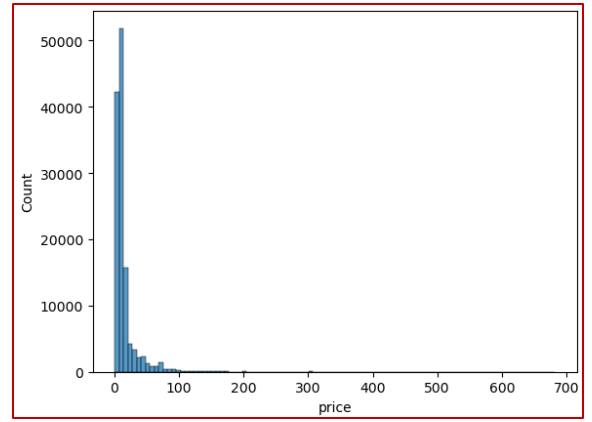
Untuk kolom **stars**, karena persentase outlier relatif kecil dan skor yang rendah belum tentu sama dengan *error* (hanya kurang populer), maka cara menanganinya dengan melakukan ***binning*** atau ***grouping*** (*low*, *medium*, *high* rating) pada saat analisis.



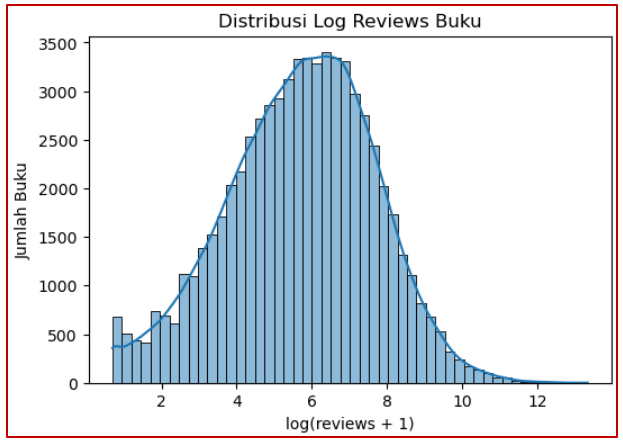
Data Cleansing: Outlier

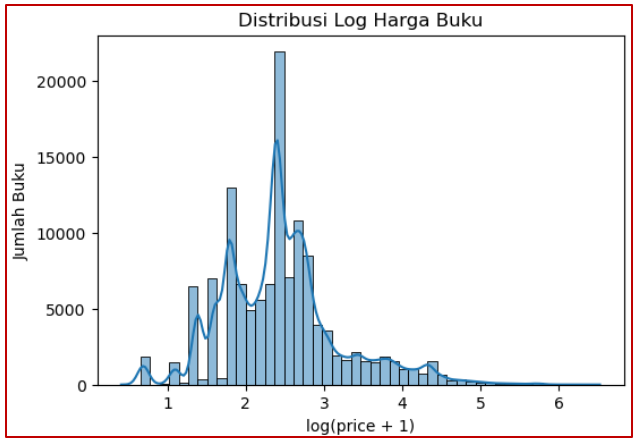
- Jika dilihat dari grafik distribusi histogram, kolom **reviews** dan **price** memiliki distribusi data yang sangat miring (***skewed***). Oleh karena itu, dilakukan ***log transform*** agar distribusi lebih stabil dan mengurangi efek outlier.





- Hasil transformasi





*Catatan: reviews diambil dengan mengecualikan reviews = 0 untuk menghindari outlier

Data Cleansing: Inconsistent Format

- Pengecekan *inconsistent format* penting dilakukan agar meningkatkan akurasi analisis dan menghindari duplikasi terselubung.
- Terdapat tiga kolom yang dilakukan pengecekan inconsistent format: **author, soldBy, dan category_name**.
- Kolom title tidak dilakukan pengecekan karena jumlah data unique yang terlalu banyak. Untuk kolom author hanya dilakukan pengecekan cepat dengan men *tracing* 2000 data teratas.

Data Cleansing: Inconsistent Format

- Contoh pengecekan *inconsistent format* untuk kolom soldBy:

```
# cek inconsistent format kolom soldBy
kindle_data["soldBy"].value_counts().sort_index()
```

soldBy	
Amazon Digital Services LLC GU	1
Amazon Digital Services LLC HN	1
Amazon Digital Services LLC MK	1
Amazon.com	3
Amazon.com Services LLC	45008
Book Republic	2
Cengage Learning	362
DC Comics	2
De Marque	34
Disney Book Group	94
EDIGITA	16
Editorial Planeta, S.A.U.	498
Flammarion Lt.	21
Gallimard Lt.	35
Games Workshop	79
GeMS SpA	10
Giunti Editore S.p.A.	1
Hachette Book Group	2420
Harlequin Digital Sales Corp.	313
Harper Collins	105
HarperCollins Publishers	2541
HarperCollins Publishing	873

Inconsistent Format

- Perbaiki format dengan *mapping* ke format yang benar

```
# mapping inkonsisten kolom soldBy ke format yang benar
publisher_format = { "HarperCollins Publishers" : "Harper Collins",
                     "HarperCollins Publishing" : "Harper Collins",
                     "Simon and Schuster Digital Sales Inc" : "Simon & Schuster Digital Sales Inc."}

# mengubah hasil mapping sehingga nilai menjadi seragam
kindle_data["soldBy"] = kindle_data["soldBy"].replace(publisher_format)

# cek ulang format soldBy
kindle_data["soldBy"].value_counts().sort_index()
```

soldBy	
Amazon Digital Services LLC GU	1
Amazon Digital Services LLC HN	1
Amazon Digital Services LLC MK	1
Amazon.com	3
Amazon.com Services LLC	45008
Book Republic	2
Cengage Learning	362
DC Comics	2
De Marque	34
Disney Book Group	94
EDIGITA	16
Editorial Planeta, S.A.U.	498
Flammarion Lt.	21
Gallimard Lt.	35
Games Workshop	79
GeMS SpA	10
Giunti Editore S.p.A.	1
Hachette Book Group	2420
Harlequin Digital Sales Corp.	313
Harper Collins	3519

Data Cleansing: Inconsistent Format

- Proses data cleansing selesai. Tersisa **128981 data** buku dari hasil data cleansing dalam dataset.

```
# Cek statistik deskriptif
kindle_data.describe()
```

	reviews	price	stars	category_id	log_reviews	log_price
count	128981.000000	128981.000000	128981.000000	128981.000000	128981.000000	128981.000000
mean	847.199727	15.593097	4.405184	16.215877	2.948619	2.446420
std	4863.522486	22.417814	0.744257	8.419084	3.177898	0.759903
min	0.000000	0.500000	0.000000	1.000000	0.000000	0.405465
25%	0.000000	5.990000	4.400000	9.000000	0.000000	1.944481
50%	5.000000	9.990000	4.500000	16.000000	1.791759	2.396986
75%	363.000000	14.990000	4.700000	23.000000	5.897154	2.771964
max	618227.000000	682.000000	5.000000	31.000000	13.334613	6.526495

- Terakhir, ubah urutan kolom dan export data akhir ke dalam format .csv

```
# Ubah urutan kolom
new_order = ['asin', 'title', 'author', 'soldBy', 'stars','reviews', 'price', 'isKindleUnlimited',
             'category_id', 'isBestSeller','isEditorsPick', 'isGoodReadsChoice', 'category_name',
             'log_reviews', 'log_price']
kindle_data = kindle_data[new_order]


# Export_data
kindle_data.to_csv("Kindle_Data_Cleaned_2.csv", index=False)
```

Exploration

Analisis project ini dilakukan dengan mencakup tiga hal sebagai berikut:

- Overview, yang mencakup:
  - Total buku yang tersedia, total eksemplar buku yang terjual dan total sales
  - TOP 5 kategori buku dengan penjualan terbaik
  - TOP 5 Books by Seller
  - TOP 5 Best Selling Books
- The Influence of tags on books sales
- Unreviewed Books Analysis

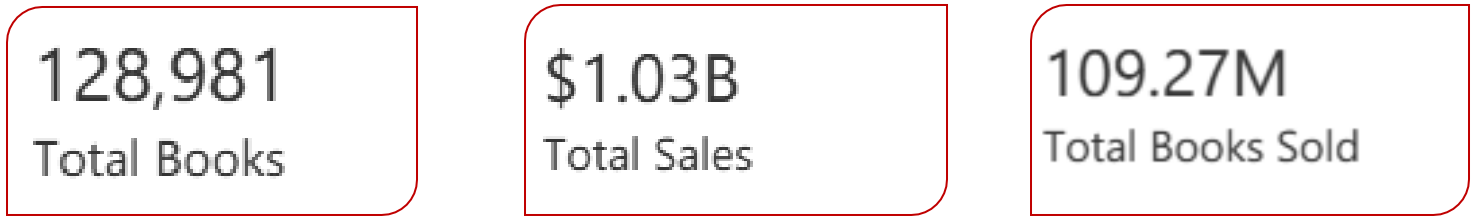
Note:

Tampilan grafik dan table Overview dan Unreviewed Books Analysis dilakukan menggunakan **Tableau**.

Proses analisis regresi dan grafik untuk The Influence of tags on books sales digunakan menggunakan model OLS pada **python**.

Exploration: Overview

Di tahun 2023, terdapat sekitar 129 ribu kindle e-books yang tersedia di Amazon. Dari e-books tersebut, menghasilkan total penjualan sebesar $1.03 Milyar dan 109.27 juta buku terjual.



Dari hasil rata-rata penjualan dari 31 kategori buku, **Literature & Fiction** yang merupakan buku dengan urutan ke 18 dari jumlahnya menghasilkan penghasilan yang paling baik dibandingkan kategori buku lain. Selain itu, genre buku **Nonfiction** juga meraup rata-rata pendapatan yang cukup baik meskipun jumlah bukunya relatif sedikit.

Category Name	Avg. Sales	Total Books	Rank of Total Books
Literature & Fiction	$82,577.01	4,104	18
Teen & Young Adult	$25,542.36	5,623	5
Biographies & Memoirs	$24,512.48	5,302	10
Nonfiction	$22,140.79	876	30
Science Fiction & Fantasy	$18,692.23	3,939	20

Note: Karena tidak ada kolom yang mendukung untuk menghitung sales, nilai sales diambil dari perkalian antara reviews dan price. Asumsi ini diambil karena orang yang memberikan review kemungkinan besar merupakan konsumen yang membeli.

Exploration: Overview

Jika dilihat berdasarkan *seller*, buku dari 'Pottermore' mendapatkan hasil penjualan yang paling baik. Dan jika ditinjau lebih jauh, buku dari distributor ini mendistribusikan buku dengan genre **Literature & Fiction**

Sold By	Avg. Sales	Total Books
Pottermore	$211,807.99	38
Scholastic Trade Publisher	$81,967.10	99
Versilio	$41,569.68	2
Immatériel fr	$39,313.80	11
Amazon Digital Services LLC HN	$37,203.36	1

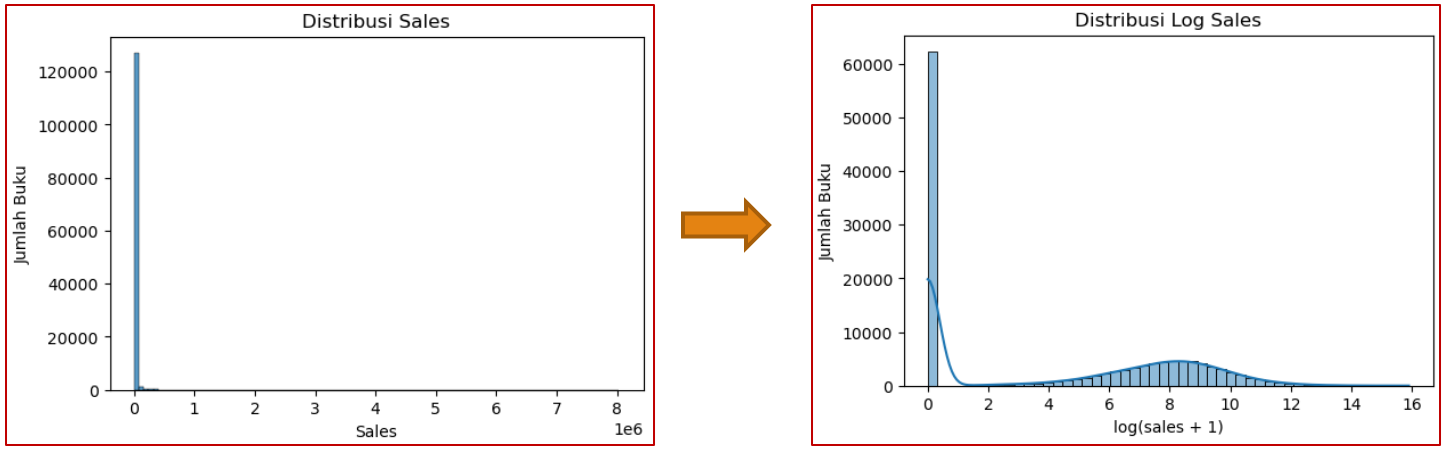
Title	
Harry Potter and the Cursed Child - Parts One and Two: The Official Playscript of t..	$918.29K Literature & Fiction
Harry Potter and the Deathly Hallows	$884.58K Literature & Fiction
Harry Potter and the Chamber of Secrets	$883.71K Literature & Fiction
Harry Potter and the Prisoner of Azkaban	$837.18K Literature & Fiction
Harry Potter and the Order of the Phoenix	$780.25K Literature & Fiction

Untuk 'TOP 5 Best Selling Books' dapat terlihat pada tabel di bawah. Terlihat kelima buku merupakan buku dengan genre **Literature & Fiction**. Hal ini bisa dijadikan masukkan bagi tim untuk menambahkan buku dengan genre tersebut karena banyak peminatnya, terbukti dari hasil penjualan buku yang cukup baik.

Title	Category Name	Sales	Ratings
Where the Crawdads Sing	Literature & Fiction	$8.03M	4.70
It Ends with Us: A Novel	Literature & Fiction	$3.56M	4.70
The Nightingale: A Novel	Literature & Fiction	$3.47M	4.70
Lessons in Chemistry: A Novel	Literature & Fiction	$3.30M	4.60
The Midnight Library: A Novel	Literature & Fiction	$3.29M	4.30

Exploration: The Influence of tags on books sales

- Dalam dataset kindle e-books, terdapat empat *tag* buku yang bisa menjadi faktor penjualan sales:
  - isKindleUnlimited
  - isBestSeller
  - isGoodReadsChoice
  - isEditorsPick
- Kita akan memprediksi tags mana yang paling berpengaruh, dan seberapa besar dampaknya pada kenaikan sales.
- Karena dalam dataset nilai sales memiliki distribusi data yang sangat miring (skewed), dilakukan log transformasi agar terhindar dari outlier.



- Karena banyaknya e-books dengan nilai review 0, kita akan mengecualikan dalam analisis pengaruh *tags* terhadap sales

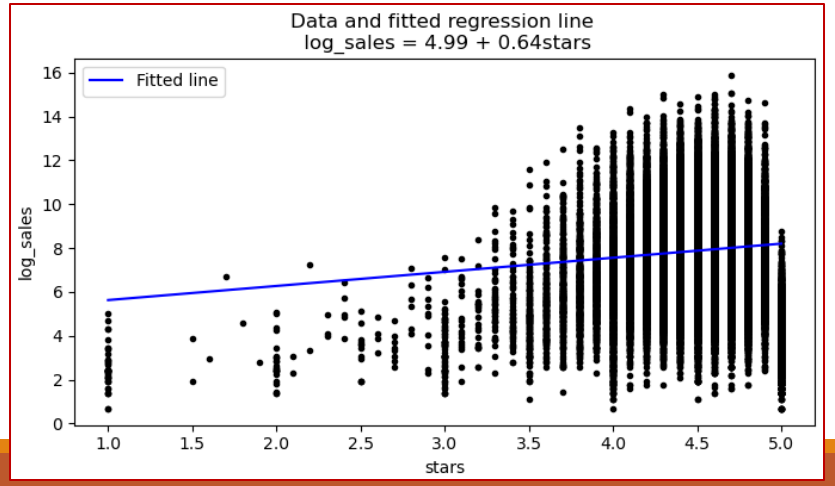
Note: Log transform dilakukan menggunakan function np.log1p (log(1+x)) agar bisa mendefinisikan data dengan nilai sales = 0

Exploration: The Influence of tags on books sales

- Pertama, diasumsikan kita ingin mencari tahu hubungan antara ratings (**stars**) dengan **sales**
- Bangun model menggunakan OLS:

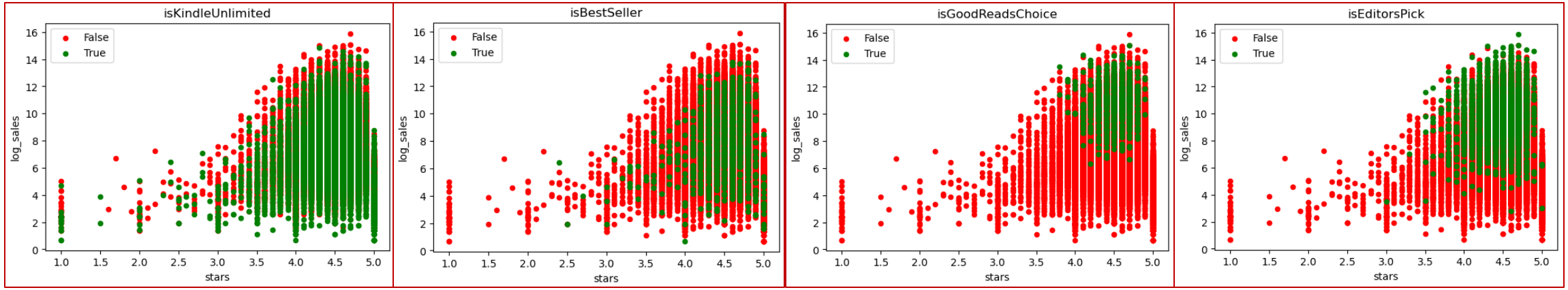
```
# Create OLS model object
model = smf.ols("log_sales ~ stars", kindle_no_0_reviews)

# Fit the model
results = model.fit()

# Extract the results (Coefficient and Standard Error) to DataFrame
results_1 = print_coef_std_err(results)
```

- Visualisasikan data dan garis regresinya:



- Hasil visualisasi garis regresi sederhana menggunakan satu predictor menunjukkan hubungan yang positif antara **stars** dan **log_sales**
- Selanjutnya, mari kita lihat pengaruh keempat *tags* terhadap data.

Exploration: The Influence of tags on books sales



Dari hasil masing-masing visualisasi plot, terlihat variabel **isEditorsPick** dengan value 'True' berkumpul di pojok kanan atas yang menandakan buku dengan *tag* tersebut kemungkinan besar adalah buku dengan penjualan yang besar juga dengan rating yang sangat baik.

Sekarang, kita tambahkan **isEditorsPick** ke dalam model regresi awal untuk mendapat hasil yang lebih dapat diinterpretasikan dan melihat seberapa besar pengaruhnya terhadap sales.

```
# Use LabelEncoder to convert the smoker variable into numeric
from sklearn.preprocessing import LabelEncoder

# Create LabelEncoder Object and transform the smoker variable
kindle_no_0_reviews["isEditorsPick"] = LabelEncoder().fit_transform(kindle_no_0_reviews["isEditorsPick"])

# Display the 5th first row after transforming
kindle_no_0_reviews[["isEditorsPick","log_sales"]].head()
```

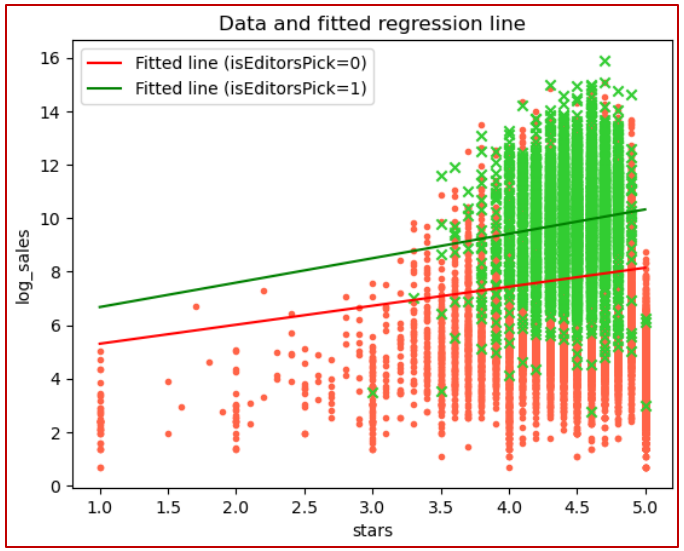
Pertama, kolom **isEditorsPick** dikonversi nilainya menjadi bilangan numeric menggunakan *library* **LabelEncoder**.

Exploration: The Influence of tags on books sales

```
# Create OLS model object
model = smf.ols('log_sales ~ stars + isEditorsPick + isEditorsPick:stars , kindle_no_0_reviews)

# Fit the model
results = model.fit()

# Extract the results (Coefficient and Standard Error) to DataFrame
results_stars_editor_inter = print_coef_std_err(results)
results_stars_editor_inter
```

Kedua variabel diberikan interaksi agar masing-masing kemiringan garis regresi dapat dibedakan

- Hasil visualisasi plot:



Dari hasil visualisasi terlihat untuk data yang memiliki *tag* **isEditorsPick** memiliki *slope* kemiringan yang lebih curam dibandingkan yang tidak ada *tag*. Ini menandakan penambahan *tag* tersebut berpengaruh positif dalam meningkatkan sales. Sekarang, mari kita lihat seberapa besar pengaruhnya, dilihat dari persamaan model regresinya.

Exploration: The Influence of tags on books sales

- Nilai koefisien dari model OLS

	coef	std err
Intercept	4.595067	0.122331
stars	0.710163	0.027048
isEditorsPick	1.168855	0.702473
isEditorsPick:stars	0.202353	0.157610

log_sales = 4.6 + 0.71stars + 1.17isEditorsPick + 0.2isEditorsPick*stars

- **Kasus 1:** isEditorPick = 0

log_sales = 4.6 + 0.71stars

  - **Intercept (4.6):** Estimasi nilai sales untuk buku dengan nilai stars = 0 sebesar **$98.48** ($e^{4.6} - 1 \approx 98.48$)
  - **Slope (0.71):** Estimasi kenaikan sales setiap kenaikan stars sebesar 1 poin, yaitu **103.4%** ($(e^{0.71} - 1) * 100\% \approx 103.4\%$)

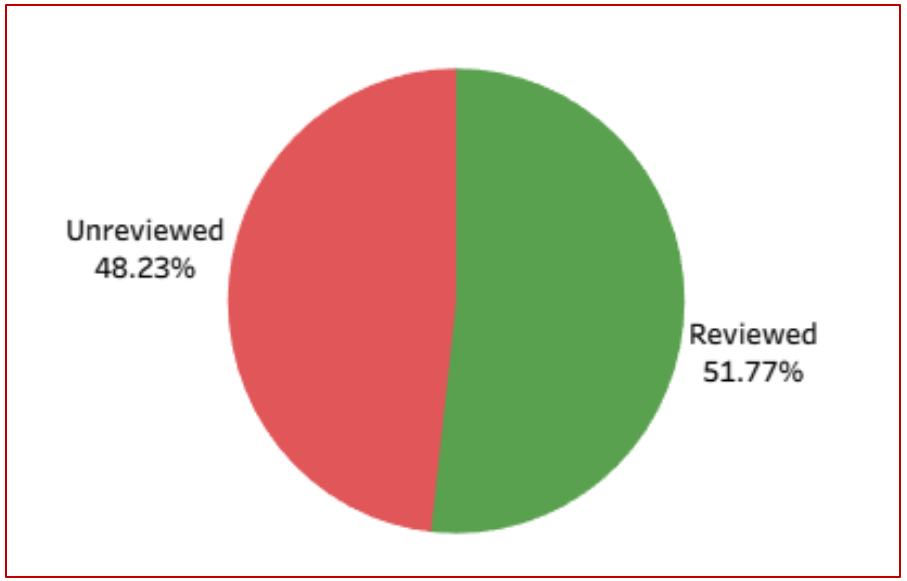
- **Kasus 2:** isEditorPick = 1

log_sales = 5.76 + 0.91stars

  - **Intercept (5.76):** Estimasi nilai sales untuk buku dengan nilai stars = 0 sebesar **$316.35** ($e^{5.76} - 1 \approx 316.35$)
  - **Slope (0.91):** Estimasi kenaikan sales setiap kenaikan stars sebesar 1 poin, yaitu **148.43%** ($(e^{0.91} - 1) * 100\% \approx 148.43\%$)

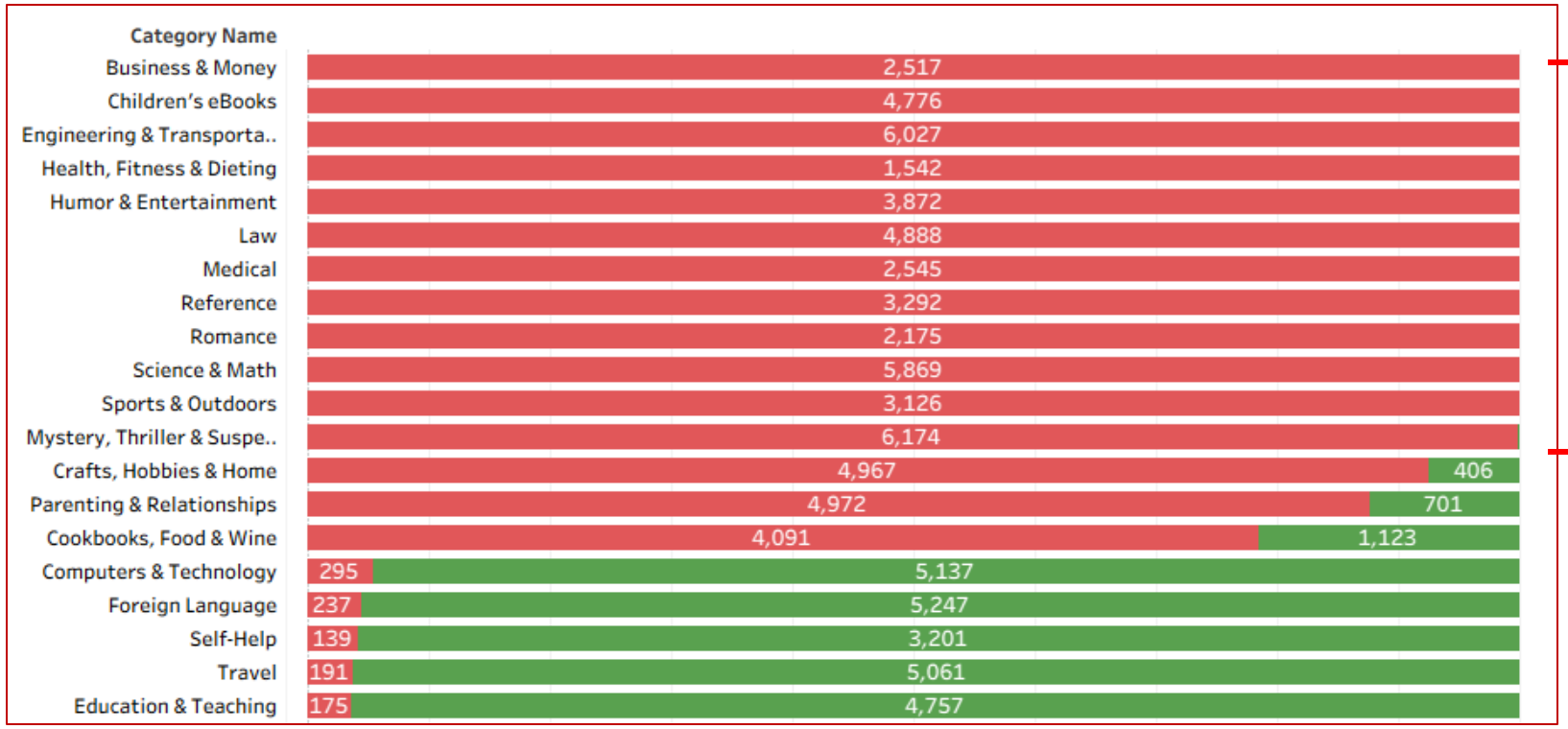
Exploration: Unreviewed Books Analysis

Dari hasil eksplorasi sebelumnya, kita mengetahui bahwa nilai sales memiliki distribusi data yang sangat miring (skewed) dikarenakan pengaruh dari masih banyaknya buku dengan review yang masih kosong. Kita akan meninjau lebih jauh tentang masalah ini.



- Dari hasil *pie chart*, dengan mengelompokkan buku berdasarkan jumlah review nya, terlihat perbedaan presentase buku yang belum ada review (review = 0) dengan buku yang sudah ada review (review > 0) tipis sekali. Hampir separuh kindle e-books yang ada di Amazon masih belum ada review yang ditambahkan konsumen.

Exploration: Unreviewed Books Analysis

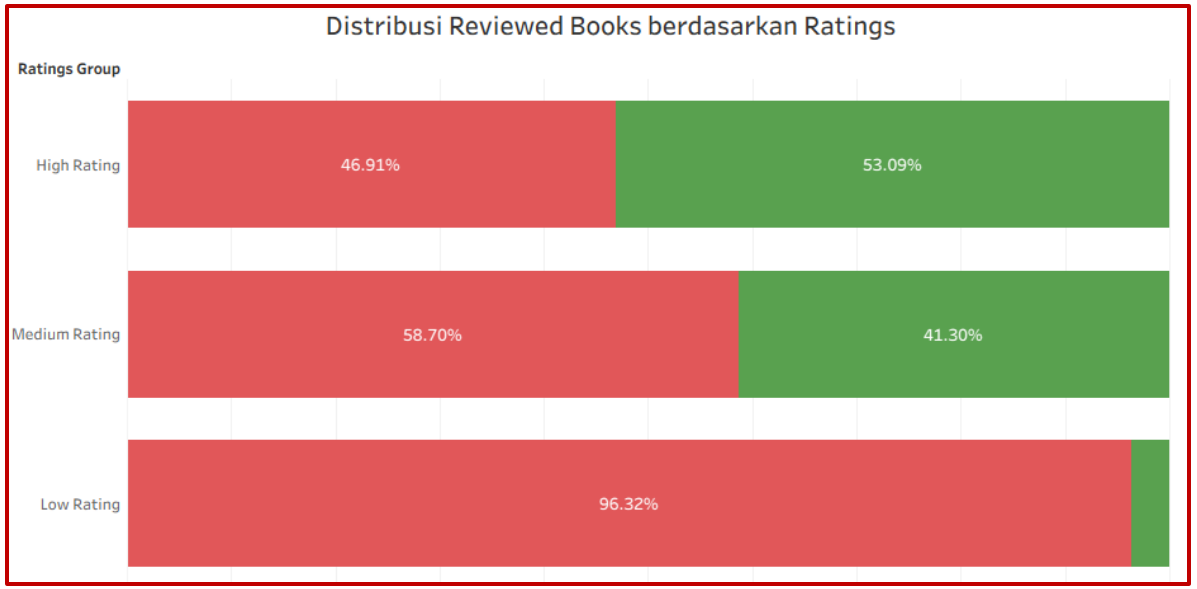


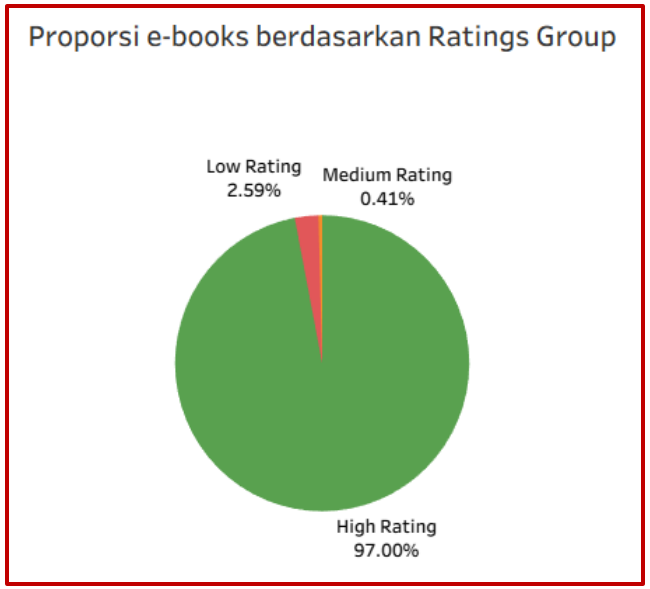
Hampir 100% data dengan review yang kosong

- Jika dilihat dari distribusi berdasarkan genre nya, dari total 31 genre buku, terdapat 12 genre yang setiap bukunya belum memiliki review dari konsumen. Padahal jumlah buku tiap genre tersebut ribuan. Hal ini patut dipertanyakan mengingat review juga menjadi faktor yang mendorong konsumen untuk membeli buku.

Exploration: Unreviewed Books Analysis

- Jika kita lihat berdasarkan ratings (stars), proporsi buku yang belum di review dengan ratings yang rendah jauh lebih banyak dibandingkan yang sudah di review. Akan tetapi, hal itu tidak menjamin faktor yang menjadi alasan masalah ini karena proporsi buku dengan ratings yang tinggi masih jauh lebih besar.





- Atas dasar hal ini, kita dapat menyimpulkan kemungkinan faktor yang menyebabkan masih banyaknya buku dengan review yang masih kosong karena kurangnya dorongan konsumen untuk menambahkan review.

Note: Rentang group ratings (High Rating >= 3.5, Medium Rating >= 3, Low Rating < 3)

Insights

Dari hasil analisis ini dapat disimpulkan:

- Di tahun 2023, kindle e-books memperoleh total penjualan yang cukup baik, sekitar **$1.03 Milyar** dari total **109.27 eksemplar buku** yang terjual.
- Buku dengan kategori **Literature & Fiction, Teen & Young Adult, Biography, Nonfiction, dan Science Fiction & Fantasy** mendapat penjualan yang paling baik. Dari hasil ini menunjukan minat konsumen yang cenderung membeli buku dengan genre yang ringan.
- Walaupun sebagian besar buku yang tersedia di Amazon berasal dari Amazon.com Services LLC, buku dengan penjualan terbaik berasal dari *seller* **Pottermore**.
- Buku dengan label **isEditorsPick** memiliki penjualan **~3.2 kali lebih besar** dibandingkan tanpa label (karena $\$316.35 / \$98.48 \approx 3.2$)
- Efek **stars** untuk kenaikan sales buku dengan label isEdistorsPick **lebih besar** (148.4% per kenaikan stars satu poin) dibandingkan buku dengan tanpa label (103.4%)
- Hampir separuh buku yang tersedia di Amazon kindle e-books masih memiliki review yang kosong, beberapa katergori buku bahkan seluruh buku nya belum ada review sama sekali.

Recommendation

- Fokus promosi di kategori **Literature & Fiction, Teen & Young Adult, Biography, Nonfiction, dan Sci-Fi & Fantasy**, karena kategori ini terbukti memiliki *demand* yang tinggi.
- Buku-buku ***Best Seller*** bisa ditampilkan di laman ***homepage*** untuk mendorong minat beli konsumen.
- Bisa diperhitungkan bagi Amazon untuk **menambahkan kerja sama dari *publisher/seller* lain**, terlihat dari proporsi buku nya masih dominan dari Amazon.com Services LLC, namun penjualannya kurang lebih baik dibandingkan dari *seller* lain.
- Karena label **isEditorsPick meningkatkan penjualan ~3.2x**, disarankan memperluas program **curated picks** ini.
- Bisa dibuat kategori khusus seperti **"Editor's Weekly Picks"** atau **"Reader's Choice"** untuk meningkatkan trust dan daya tarik konsumen.
- Karena ratings (stars) berpengaruh dalam penjualan, ditambah dengan hampir separuh buku tidak memiliki review, Amazon bisa memberikan insentif ke pembaca untuk memberi review & rating (contoh: **reward poin Kindle**).
- Untuk masalah penangan buku tanpa review, Amazon bisa membuat program **membership** dimana salah satunya setiap konsumen memberikan review terbaik akan memberikan poin untuk kenaikan kelas membership. Semakin tinggi kelas membership, semakin banyak pula *benefit* yang didapatkan konsumen.