```
import pandas as pd

df = pd.read_csv(r"C:\Users\adars\OneDrive\Pictures\Documents\
sephora_website_dataset.csv")
df.head(5)
```

```
        id           brand   category                              name
\
0  2218774  Acqua Di Parma  Fragrance  Blu Mediterraneo MINIATURE Set

1  2044816  Acqua Di Parma    Cologne                          Colonia

2  1417567  Acqua Di Parma    Perfume                Arancia di Capri

3  1417617  Acqua Di Parma    Perfume                Mirto di Panarea

4  2218766  Acqua Di Parma  Fragrance           Colonia Miniature Set


              size  rating  number_of_reviews  love  price  value_price
...  \
0  5 x 0.16oz/5mL     4.0                  4  3002   66.0         75.0
...
1   0.7 oz/ 20 mL     4.5                 76  2700   66.0         66.0
...
2    5 oz/ 148 mL     4.5                 26  2600  180.0        180.0
...
3   2.5 oz/ 74 mL     4.5                 23  2900  120.0        120.0
...
4  5 x 0.16oz/5mL     3.5                  2   943   72.0         80.0
...

  MarketingFlags  MarketingFlags_content  \
0           True             online only
1           True             online only
2           True             online only
3           True             online only
4           True             online only

                                          options  \
0                                      no options
1   - 0.7 oz/ 20 mL  Spray  - 1.7 oz/ 50 mL Eau d...
2   - 1oz/30mL Eau de Toilette  - 2.5 oz/ 74 mL E...
3   - 1 oz/ 30 mL Eau de Toilette Spray - 2.5 oz/...
4                                      no options

                                          details  \
0  This enchanting set comes in a specially handc...
1  An elegant timeless scent filled with a fresh-...
2  Fragrance Family: Fresh Scent Type: Fresh Citr...
3  Panarea near Sicily is an an island suspended ...
```

```
4  The Colonia Miniature Set comes in an iconic A...

                                          how_to_use  \
0  Suggested Usage:-Fragrance is intensified by t...
1                                    no instructions
2                                    no instructions
3                                    no instructions
4  Suggested Usage:-Fragrance is intensified by t...

                                        ingredients online_only
exclusive  \
0  Arancia di Capri Eau de Toilette: Alcohol Dena...           1
0
1                                           unknown           1
0
2  Alcohol Denat.- Water- Fragrance- Limonene- Li...           1
0
3                                           unknown           1
0
4  Colonia: Alcohol Denat.- Water- Fragrance- Lim...           1
0

   limited_edition  limited_time_offer
0                0                   0
1                0                   0
2                0                   0
3                0                   0
4                0                   0

[5 rows x 21 columns]
```

```
df.describe()
```

```
            rating  reviews_count         likes        price
value_price
count  9168.000000    9168.000000  9.168000e+03  9168.000000
9168.00000
mean      3.990020     282.139180  1.627859e+04    50.063237
51.82359
std       1.007707     890.642028  4.260651e+04    47.164989
49.45902
min       0.000000       0.000000  0.000000e+00     2.000000
2.00000
25%       4.000000      10.000000  1.600000e+03    24.000000
25.00000
50%       4.000000      46.000000  4.800000e+03    35.000000
35.00000
75%       4.500000     210.000000  1.380000e+04    59.000000
60.00000
```

```
max       5.000000   19000.000000  1.300000e+06   549.000000
549.00000

df.count()

brand             9168
category          9168
name              9168
rating            9168
reviews_count     9168
likes             9168
price             9168
value_price       9168
dtype: int64

df.isnull().sum()

brand             0
category          0
name              0
rating            0
reviews_count     0
likes             0
price             0
value_price       0
dtype: int64

df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9168 entries, 0 to 9167
Data columns (total 8 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   brand          9168 non-null   object
 1   category       9168 non-null   object
 2   name           9168 non-null   object
 3   rating         9168 non-null   float64
 4   reviews_count  9168 non-null   int64
 5   likes          9168 non-null   int64
 6   price          9168 non-null   float64
 7   value_price    9168 non-null   float64
dtypes: float64(3), int64(2), object(3)
memory usage: 573.1+ KB

df.duplicated()

0       False
1       False
2       False
3       False
```

```
4       False
        ...
9163    False
9164    False
9165    False
9166    False
9167    False
Length: 9168, dtype: bool
```

```
## Data Cleaning

#1-Removing Unwanted Columns

columns_to_keep = ['brand','category',     'name',    'rating',
'number_of_reviews', 'love', 'price', 'value_price']
df = df[columns_to_keep]
df.head(5)
```

```
            brand   category                              name
rating  \
0  Acqua Di Parma  Fragrance  Blu Mediterraneo MINIATURE Set     4.0

1  Acqua Di Parma    Cologne                          Colonia     4.5

2  Acqua Di Parma    Perfume                Arancia di Capri     4.5

3  Acqua Di Parma    Perfume                Mirto di Panarea     4.5

4  Acqua Di Parma  Fragrance           Colonia Miniature Set     3.5


   number_of_reviews  love  price  value_price
0                  4  3002   66.0         75.0
1                 76  2700   66.0         66.0
2                 26  2600  180.0        180.0
3                 23  2900  120.0        120.0
4                  2   943   72.0         80.0
```

```
#2-Change Column Names

df.rename(columns={'number_of_reviews': 'review_count', 'love':
'likes'}, inplace=True)
df.head(5)
```

```
            brand   category                              name
rating  \
0  Acqua Di Parma  Fragrance  Blu Mediterraneo MINIATURE Set     4.0
```

```
1  Acqua Di Parma    Cologne                          Colonia     4.5

2  Acqua Di Parma    Perfume                Arancia di Capri     4.5

3  Acqua Di Parma    Perfume                Mirto di Panarea     4.5

4  Acqua Di Parma  Fragrance           Colonia Miniature Set     3.5


   reviews_count  likes  price  value_price
0              4   3002   66.0         75.0
1             76   2700   66.0         66.0
2             26   2600  180.0        180.0
3             23   2900  120.0        120.0
4              2    943   72.0         80.0
```

```
#3-Removing Duplicate Product if any

df.duplicated().sum()
```

```
0
```

```
#4-Finding duplicate Product with same name

dup_name = df[df.duplicated(subset='name', keep=False)]
# Show them if any duplicate name
display(dup_name[['brand', 'category',     'name',    'rating',
'reviews_count', 'likes', 'price', 'value_price']])
```

```
                         brand                     category  \
276   Anastasia Beverly Hills                     Eyebrow
293   Anastasia Beverly Hills                     Lipstick
306   Anastasia Beverly Hills                    Lip Gloss
348                   Anthony                      Shaving
539              bareMinerals                        Blush
...                       ...                          ...
9058       SEPHORA COLLECTION                    Lip Stain
9067       SEPHORA COLLECTION       Hair Styling Products
9125       SEPHORA COLLECTION  Makeup Bags & Travel Cases
9151       SEPHORA COLLECTION           Scrub & Exfoliants
9155       SEPHORA COLLECTION           Scrub & Exfoliants

                                  name  rating  reviews_count   likes
price  \
276                    Clear Brow Gel     4.5           4000  159800
22.0
293                   Liquid Lipstick     4.0           4000  549000
20.0
306                         Lip Gloss     4.5           1000  208100
16.0
348                       Shave Cream     4.5            194    1800
```

```
20.0
539                              Blush     4.5           1000   34000
22.0
...                                ...     ...            ...     ...
...
9058  Cream Lip Stain Liquid Lipstick     4.0            205   16900
14.0
9067                        Curl Cream     4.0             19    2100
14.0
9125                Hanging Organizer     4.5             23    2900
35.0
9151                 Sugar Body Scrub     4.5            163    3100
17.0
9155                 Sugar Body Scrub     4.0              6    1900
17.0

      value_price
276          22.0
293          20.0
306          16.0
348          20.0
539          22.0
...           ...
9058         14.0
9067         14.0
9125         35.0
9151         17.0
9155         17.0

[116 rows x 8 columns]
```

```
dup_name = df[df.duplicated(subset=['name','price', 'rating',
'value_price'], keep=False)]
# Show them if any duplicate name
display(dup_name[['brand', 'category',     'name',    'rating',
'reviews_count', 'likes', 'price', 'value_price']])
```

```
                   brand     category       name  rating
reviews_count  \
7031  SEPHORA COLLECTION   Face Masks  Face Mask     4.5
1000
7191  SEPHORA COLLECTION  Sheet Masks  Face Mask     4.5
1000

       likes  price  value_price
7031  117100    6.0          6.0
7191  106900    6.0          6.0
```

```
# Removing Duplicates
```

```
df = df.drop_duplicates(subset=['name', 'price', 'rating',
'value_price'], keep='first')
df.reset_index(drop=True, inplace=True)

# Rechecking

dup_name = df[df.duplicated(subset=['name','price', 'rating',
'value_price'], keep=False)]
display(dup_name[['brand', 'category',     'name',    'rating',
'reviews_count', 'likes', 'price', 'value_price']])
```

```
Empty DataFrame
Columns: [brand, category, name, rating, reviews_count, likes, price,
value_price]
Index: []
```

```
df.head(5)
```

	brand	category	name	rating	\
0	Acqua Di Parma	Fragrance	Blu Mediterraneo MINIATURE Set	4.0	
1	Acqua Di Parma	Cologne	Colonia	4.5	
2	Acqua Di Parma	Perfume	Arancia di Capri	4.5	
3	Acqua Di Parma	Perfume	Mirto di Panarea	4.5	
4	Acqua Di Parma	Fragrance	Colonia Miniature Set	3.5	

	reviews_count	likes	price	value_price
0	4	3002	66.0	75.0
1	76	2700	66.0	66.0
2	26	2600	180.0	180.0
3	23	2900	120.0	120.0
4	2	943	72.0	80.0

```
# finding the best_value_score column by comparing price and
value_price

df['best_value_score']= df['value_price']/ df['price']

df.sort_values(by='best_value_score', ascending=False)[
[ 'brand', 'category', 'name', 'price',    'value_price',
'best_value_score']
].head(10)

# Higher the best_value_score will be good for customers
```

```
                  brand           category  \
6653   PLAY! by SEPHORA  Value & Gift Sets
8037              tarte    Makeup Palettes
7328  Sephora Favorites  Value & Gift Sets
1872           CLINIQUE       Eye Palettes
6654   PLAY! by SEPHORA  Value & Gift Sets
6656   PLAY! by SEPHORA  Value & Gift Sets
7627           Smashbox       Eye Palettes
6658   PLAY! by SEPHORA  Value & Gift Sets
6032           NUDESTIX           Eye Sets
7329  Sephora Favorites           Lip Sets

                                          name  price  value_price  \
6653    PLAY! by Sephora: Beauty For Self-Care   10.0         62.0
8037     Lele Pons x Tarte Eye & Cheek Palette   35.0        164.0
7328                          Sun Safety Kit   39.0        178.0
1872  Light Up Your Eyes Eyeshadow Palette Set   39.5        172.0
6654     PLAY! by Sephora: Stress-Free Beauty   10.0         42.0
6656    PLAY! by Sephora: Award Worthy Beauty   10.0         42.0
7627           LA Cover Shot Eyeshadow Palette   45.0        147.0
6658              PLAY! LUXE by Sephora Vol. 5   25.0         79.0
6032                   Nude Metallics For Eyes   25.0         78.0
7329                 Give Me Some Balm Lip Set   29.0         89.0

      best_value_score
6653          6.200000
8037          4.685714
7328          4.564103
1872          4.354430
6654          4.200000
6656          4.200000
7627          3.266667
6658          3.160000
6032          3.120000
7329          3.068966
```

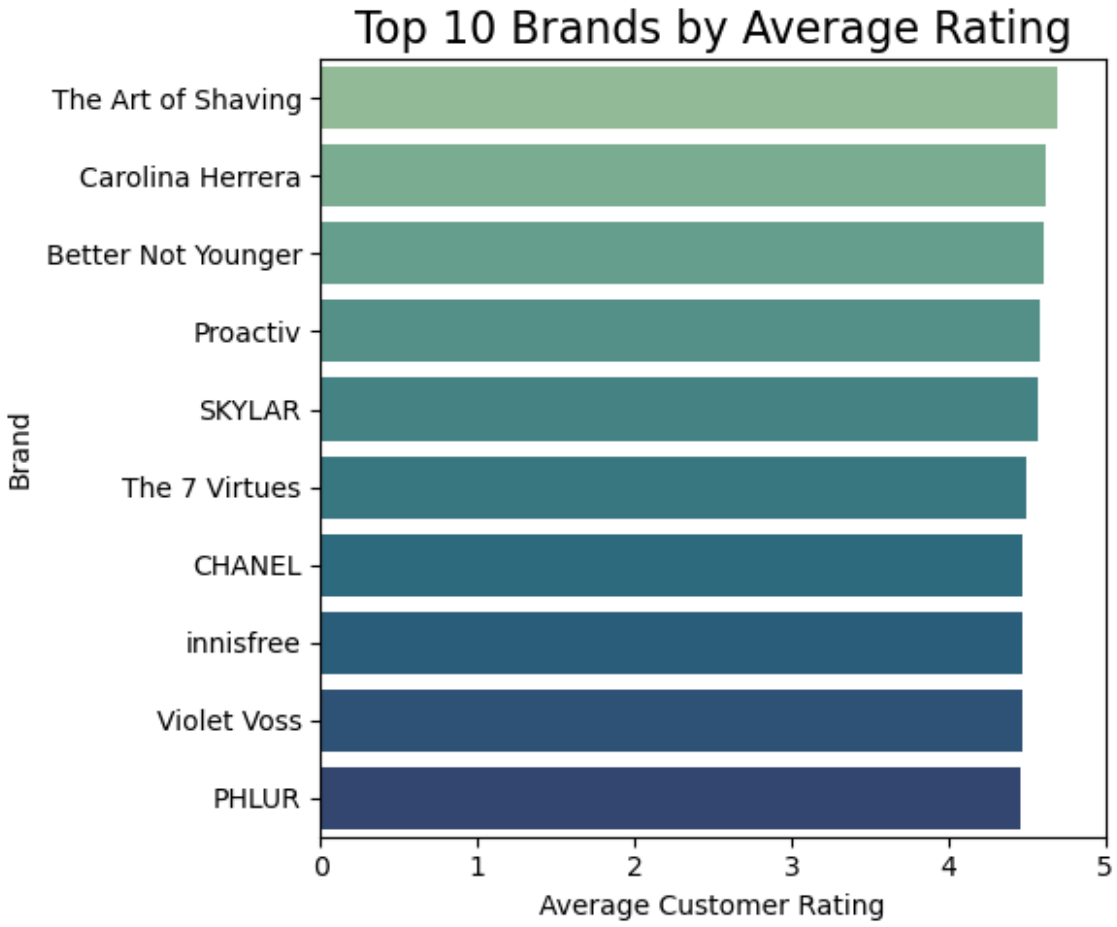
```python
import matplotlib.pyplot as plt
import seaborn as sns

import numpy as np
num_points = len(df)
colors = np.random.rand(num_points)

plt.figure(figsize=(8, 5))
sns.scatterplot(
    data=df,
    x='best_value_score',
    y='rating',
    hue=colors,
    palette='viridis',
```

```
    legend=False,
    alpha=0.7
)

plt.title("Best Value Score vs Customer Rating", fontsize=16)
plt.xlabel("Best Value Score")
plt.ylabel("Customer Rating")
plt.tight_layout()
plt.show()
```


```
#2- Top 10 Brands by Avg Rating

#Grouping and Calculating avg and count
#Filter brands
#Top 10 brands by avg rating

brand_ratings = df.groupby('brand').agg(avg_rating=('rating', 'mean'),
total_products=('rating', 'count'))
brand_ratings = brand_ratings[brand_ratings['total_products']>=10]
top_brands = brand_ratings.sort_values(by='avg_rating',
ascending=False).head(10)

#Plotting
plt.figure(figsize=(6, 5))
```

```
sns.barplot(data=top_brands, x='avg_rating', y='brand',
palette='crest')

plt.title("Top 10 Brands by Average Rating", fontsize=16)
plt.xlabel("Average Customer Rating")
plt.ylabel("Brand")
plt.xlim(0, 5)
plt.tight_layout()
plt.show()
```

```
C:\Users\adars\AppData\Local\Temp\ipykernel_17848\2758840922.py:11:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `y` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(data=top_brands, x='avg_rating', y='brand',
palette='crest')
```

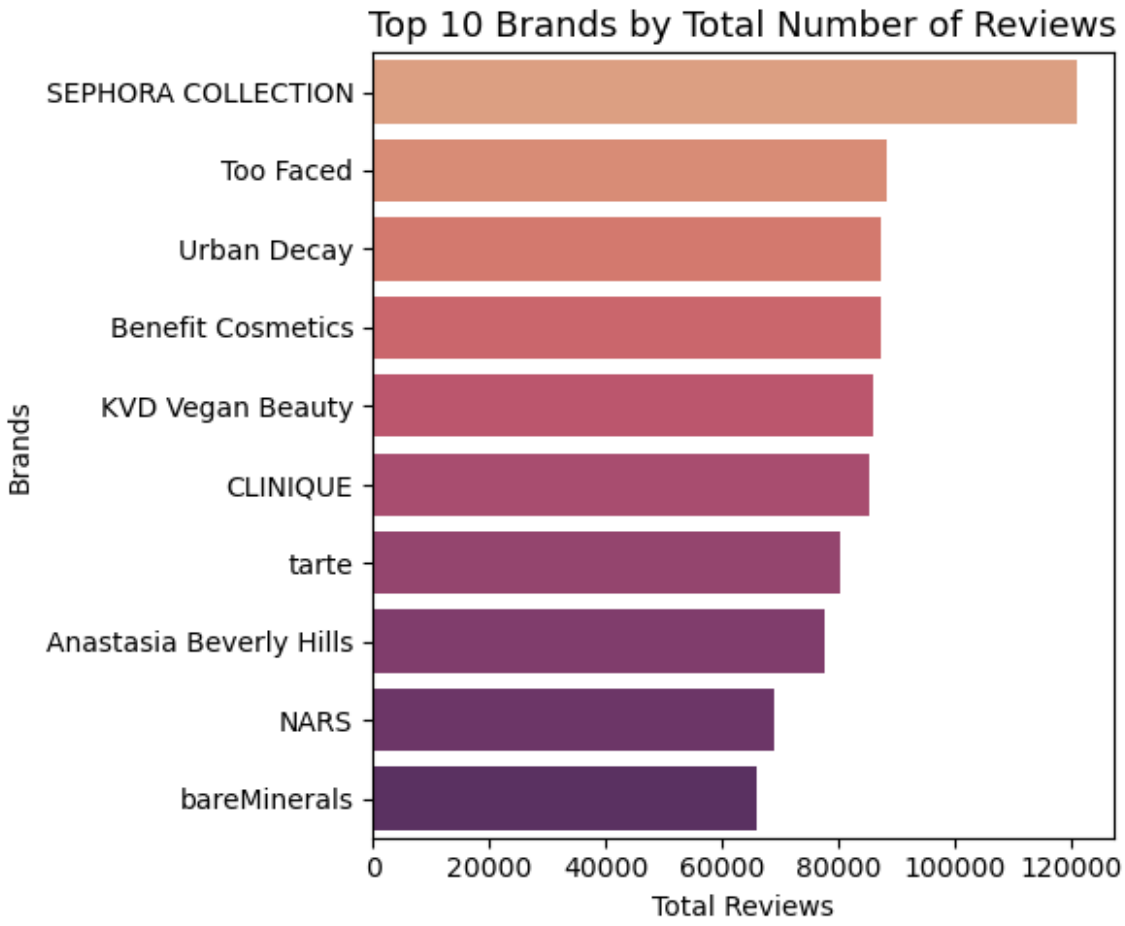


```
#3- Top 10 Brands by  number of reviws
df.head(1
       )
```

```
        id           brand   category                                name
\
0  2218774  Acqua Di Parma  Fragrance  Blu Mediterraneo MINIATURE Set


              size  rating  number_of_reviews  love  price  value_price
...  \
0  5 x 0.16oz/5mL     4.0                  4  3002   66.0         75.0
...

  MarketingFlags  MarketingFlags_content     options  \
0           True             online only  no options

                                              details  \
0  This enchanting set comes in a specially handc...

                                           how_to_use  \
0  Suggested Usage:-Fragrance is intensified by t...

                                         ingredients online_only
exclusive  \
0  Arancia di Capri Eau de Toilette: Alcohol Dena...           1
0

   limited_edition  limited_time_offer
0                0                   0

[1 rows x 21 columns]
```

```
#Group by Brand and Sum reviws
#Top 10 brands by reviews count

brand_reviews = df.groupby('brand')
['number_of_reviews'].sum().reset_index()
top_reviewed_brands =
brand_reviews.sort_values(by='number_of_reviews',
ascending=False).head(10)

plt.figure(figsize=(6, 5))
sns.barplot(data=top_reviewed_brands, x='number_of_reviews',
y='brand', palette='flare')

plt.title("Top 10 Brands by Total Number of Reviews", fontsize=13)
plt.xlabel("Total Reviews")
plt.ylabel("Brands")
plt.tight_layout()
plt.show()
```

```
C:\Users\adars\AppData\Local\Temp\ipykernel_17848\605629553.py:8:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `y` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(data=top_reviewed_brands, x='number_of_reviews',
y='brand', palette='flare')
```



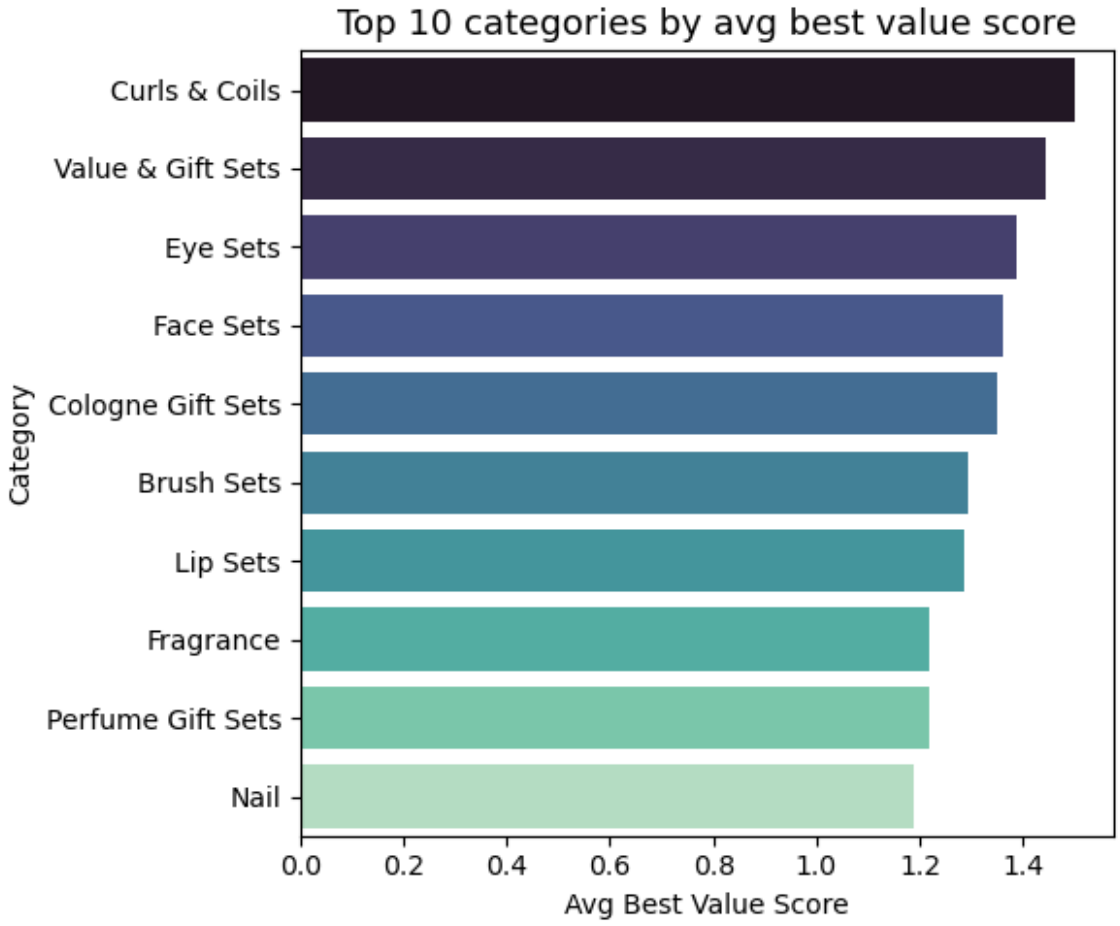

```
#4-Price vs Customer Rating
#--to check if higher priced products gets better  rating, or if price
doesn't matter for satisafaction

plt.figure(figsize=(6, 5))
sns.scatterplot(data=df, x='price', y='rating', alpha=0.6,
color='teal')

plt.title("Price vs Customer Rating", fontsize=13)
```

```
plt.xlabel("Product Price")
plt.ylabel("Customer Rating")
plt.tight_layout()
plt.show()
```


```
#5Average best value_score by category
#-- to see which categories offer the best deals in terms of value per
rupee

category_value = df.groupby('category')
['best_value_score'].mean().sort_values(ascending=False).head(10)

plt.figure(figsize=(6, 5))
sns.barplot(x=category_value.values, y=category_value.index,
palette='mako')

plt.title("Top 10 categories by avg best value score", fontsize=13)
plt.xlabel("Avg Best Value Score")
plt.ylabel("Category")
```

```
plt.tight_layout()
plt.show()
```

```
C:\Users\adars\AppData\Local\Temp\ipykernel_20704\3684351960.py:4:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `y` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(x=category_value.values, y=category_value.index,
palette='mako')
```




```
#6-Distributation of Best Value Score

plt.figure(figsize=(6, 5))
sns.histplot(df['best_value_score'], bins=30, kde=True, color='green')

plt.title("Distributiom of Best Value Score", fontsize=13)
```

```
plt.xlabel("Best Value Score")
plt.ylabel("Number of Products")
plt.tight_layout()
plt.show()
```