

Q:Match similar products from the Flipkart dataset with the Amazon dataset

Importing libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

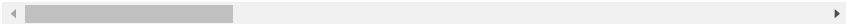
Loading the Dataset

```
amazon = pd.read_excel('/content/amz_com.xlsx')

flipkart = pd.read_excel('/content/flipkart_com.xlsx')

amazon.head()
```

|   | uniq_id                          | crawl_timestamp           | product_url                                      |
|---|----------------------------------|---------------------------|--|
| 0 | ec475b0eb06a90ee1ec5794a314e065d | 2015-12-01 06:13:00 +0000 | http://www.flipkart.com/jewelizer alloy-neckla.. |
| 1 | 6eade120ebeade57fcbc1120a6a2eeae | 2015-12-01 06:13:00 +0000 | http://www.flipkart.com/abb premium-make-6-one.. |
| 2 | f4ff5c04e3bb1171fbf06aa9154cf90c | 2016-06-10 11:36:05 +0000 | http://www.flipkart.com/monmione women-s-push-.. |
| 3 | 72251c555e30c587a23bd01ca73868ac | 2015-12-01 12:40:44 +0000 | http://www.flipkart.com/allure auto-cm-839-car.. |
| 4 | cf519942dbf21758504367ad8795d418 | 2015-12-30 00:17:46 +0000 | http://www.flipkart.com/india-inc women-s-prin.. |



```
flipkart.head()
```

|   | uniq_id                          | crawl_timestamp              | product_url                                       | product_name                        | product_category_tree                             |           |
|---|----------------------------------|------------------------------|---|-------------------------------------|---|-----------|
| 0 | c2d766ca982eca8304150849735ffe9  | 2016-03-25<br>22:59:23 +0000 | http://www.flipkart.com/alisha-solid-women-s-c... | Alisha Solid Women's Cycling Shorts | ["Clothing >> Women's Clothing >> Lingerie, Sl... | SRTEH2FF5 |
| 1 | 7f7036a6d550aaa89d34c77bd39a5e48 | 2016-03-25<br>22:59:23 +0000 | http://www.flipkart.com/fabhomedecor-fabric-do... | FabHomeDecor Fabric Double Sofa Bed | ["Furniture >> Living Room Furniture >> Sofa B... | SBEEH3QG  |

Pre Processing

```
amazon.isnull().sum()
amazon=amazon.dropna(axis=0)
amazon.shape

(14121, 15)

flipkart.isnull().sum()
flipkart=flipkart.dropna(axis=0)
flipkart.shape

(14058, 15)

amazon.columns

Index(['uniq_id', 'crawl_timestamp', 'product_url', 'product_name',
      'product_category_tree', 'pid', 'retail_price', 'discounted_price',
      'image', 'is_FK_Advantage_product', 'description', 'product_rating',
      'overall_rating', 'brand', 'product_specifications'],
      dtype='object')

amazon.drop(['crawl_timestamp', 'product_url',
            'product_category_tree', 'pid',
            'image', 'is_FK_Advantage_product', 'description', 'product_rating',
            'overall_rating', 'product_specifications'], axis=1, inplace=True)

flipkart.columns

Index(['uniq_id', 'crawl_timestamp', 'product_url', 'product_name',
      'product_category_tree', 'pid', 'retail_price', 'discounted_price',
      'image', 'is_FK_Advantage_product', 'description', 'product_rating',
      'overall_rating', 'brand', 'product_specifications'],
      dtype='object')

flipkart.drop(['crawl_timestamp', 'product_url',
              'product_category_tree', 'pid',
              'image', 'is_FK_Advantage_product', 'description', 'product_rating',
              'overall_rating', 'product_specifications'], axis=1, inplace=True)

/usr/local/lib/python3.8/dist-packages/pandas/core/frame.py:4906: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
return super().drop(

amazon.columns

Index(['uniq_id', 'product_name', 'retail_price', 'discounted_price', 'brand'], dtype='object')

amazon.head()
```

```
flipkart.columns

Index(['uniq_id', 'product_name', 'retail_price', 'discounted_price', 'brand'], dtype='object')

flipkart.head()
```

|   | uniq_id                          | product_name                        | retail_price | discounted_price | brand |
|---|----------------------------------|-------------------------------------|--------------|------------------|-------|
| 0 | c2d766ca982eca8304150849735ffef9 | Alisha Solid Women's Cycling Shorts | 999.0        | 379.0            |       |
| 1 | 7f7036a6d550aaa89d34c77bd39a5e48 | FabHomeDecor Fabric Double Sofa Bed | 32157.0      | 22646.0          |       |
| 2 | f449ec65dcb041b6ae5e6a32717d01b  | AW Relies                           | 999.0        | 499.0            |       |

```
amazon.describe

<bound method NDFrame.describe of      uniq_id \
0      ec475b0eb06a90ee1ec5794a314e065d
1      6eade120ebeade57fcbc1120a6a2eeae
2      f4ff5c04e3bb1171fbf06aa9154cf90c
3      72251c555e30c587a23bd01ca73868ac
4      cf519942dbf21758504367ad8795d418
...
19993  c4b045288524a8770c760ed2bbca2ed5
19994  dd96000fa1d9e408a4fc47ea5c1123e5
19996  07b0df742cdcac28d09c29a1e246fff2
19997  d9fa5b1d8917b841abae2a1ce032114
19998  3a2546675bc399953779e58d84d56650

      product_name  retail_price \
0      Jewelizer Alloy Necklace      -20
1      ABB Premium Make 6 One Way Electrical Switch      -20
2      Monmione Women's Push-up Bra      -20
3      Allure Auto CM 839 Car Mat Toyota Corolla Altis      -19
4      India Inc Women's Printed Casual Shirt      -19
...
19993  Audeze Lcd2 In Bamboo, High Quality Planar Mag...    116289
19994  Durian Club/3 Leather 3 Seater Sofa    161982
19996  Durian Leather 2 Seater Sofa    204592
19997  Durian Laze/3 Leather 3 Seater Sofa    217495
19998  Durian Leather 2 Seater Sofa    250483

      discounted_price      brand
0      0      Jewelizer
1      0      ABB
2      0      Monmione
3      0      Allure Auto
4      0      Normal Fit
...
19993    144560      Audeze
19994    135928      Durian
19996    163788      Durian
19997    157779      Durian
19998    207175      Durian

[14121 rows x 5 columns]>
```

```
flipkart.describe

<bound method NDFrame.describe of      uniq_id \
0      c2d766ca982eca8304150849735ffef9
1      7f7036a6d550aaa89d34c77bd39a5e48
2      f449ec65dcb041b6ae5e6a32717d01b
3      0973b37acd0c664e3de26e97e5571454
4      bc940ea42ee6bef5ac7cea3fb5cfbee7
...
19995  7179d2f6c4ad50a17d014ca1d2815156
19996  71ac419198359d37b8fe5e3fffdfee09
19997  93e9d343837400ce0d7980874ece471c
19998  669e79b8fa5d9ae020841c0c97d5e935
19999  cb4fa87a874f715fff567f7b7b3be79c

      product_name  retail_price \
0      Alisha Solid Women's Cycling Shorts    999.0
```

```

1      FabHomeDecor Fabric Double Sofa Bed      32157.0
2                      AW Bellies              999.0
3      Alisha Solid Women's Cycling Shorts      699.0
4      Sicons All Purpose Arnica Dog Shampoo    220.0
...
19995      WallDesign Small Vinyl Sticker      1500.0
19996 Wallmantra Large Vinyl Stickers Sticker  1429.0
19997 Elite Collection Medium Acrylic Sticker  1299.0
19998 Elite Collection Medium Acrylic Sticker  1499.0
19999 Elite Collection Medium Acrylic Sticker  1499.0

```

```

      discounted_price      brand
0          379.0      Alisha
1      22646.0      FabHomeDecor
2          499.0      AW
3          267.0      Alisha
4          210.0      Sicons
...
19995      730.0      WallDesign
19996      1143.0      Wallmantra
19997      999.0      Elite Collection
19998      1199.0      Elite Collection
19999      999.0      Elite Collection

```

```
[14058 rows x 5 columns]>
```

```
display(amazon.dtypes)
```

```

uniq_id      object
product_name  object
retail_price  int64
discounted_price  int64
brand         object
dtype: object

```

```
display(flipkart.dtypes)
```

```

uniq_id      object
product_name  object
retail_price  float64
discounted_price  float64
brand         object
dtype: object

```

Checking Duplicate Entries

```
amazon.duplicated().sum()
```

```
0
```

```
flipkart.duplicated().sum()
```

```
0
```

## ▼ Flipkart EDA

```
flipkart["brand"].value_counts()
```

```

Allure Auto      468
Regular          308
Voylla           299
Slim             284
TheLostPuppy     229
...
ORIFLAME SWEDEN   1
Wella            1
Cayman           1
Nineteen         1
Fun To See       1
Name: brand, Length: 3481, dtype: int64

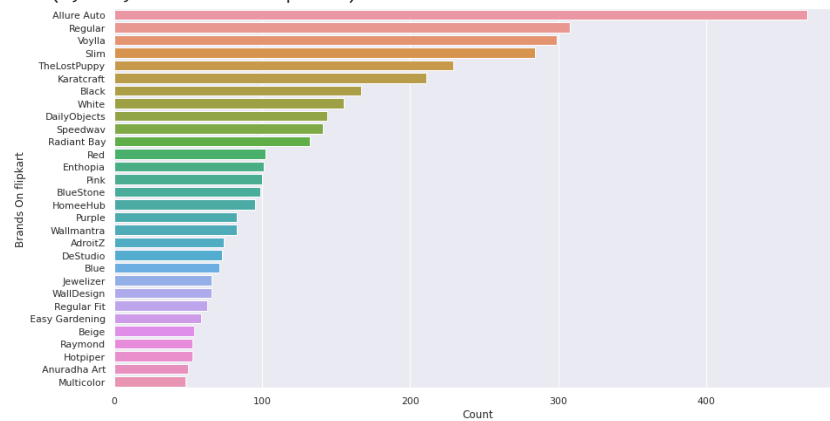
```

```
L1=list(flipkart["brand"].value_counts().keys()[0:30])
```

```
L2=list(flipkart["brand"].value_counts().values[0:30])
```

```
plt.figure(figsize=(15,8))
sns.set(style="darkgrid")
sns.barplot(x=L2,y=L1,data=flipkart)
plt.xlabel("Count")
plt.ylabel("Brands On flipkart")
```

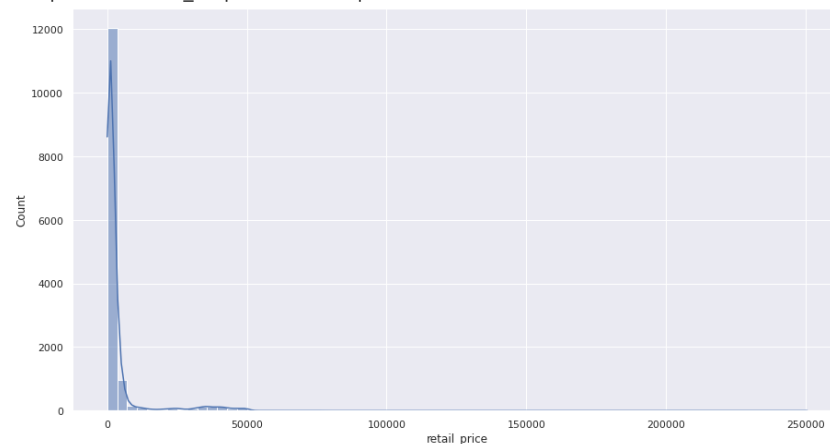
Text(0, 0.5, 'Brands On flipkart')



Allure Auto Brand has highest products, Multicolor Brand has lowest number of products.

```
plt.figure(figsize=(15,8))
sns.set(style="darkgrid")
sns.histplot(data=flipkart,x="retail_price",kde=True, legend=True,bins=70)
```

<matplotlib.axes.\_subplots.AxesSubplot at 0x7fd25dd0c730>



```
flipkart['retail_price'].describe().astype(int)
```

```
count    14058
mean      3640
std       9272
min         35
25%        699
50%       1100
```

```
75%      2199
max      250500
Name: retail_price, dtype: int64
```

#Minimum Product Price id INR 35 and Maximun is INR 250500

Amazon EDA

[ ] 16 cells hidden

Merging Both Data Frames

```
merge=amazon.merge(flipkart,left_on='uniq_id',right_on='uniq_id',how='inner')
```

```
merge.head()
```

|   | uniq_id                          | Product name in Amazon                               | Retail Price in amazon | Discounted Price in Amazon | Product name in flipkart                             | Pr f] |
|---|----------------------------------|--|------------------------|----------------------------|--|-------|
| 0 | 7c88739f05009f3ed8b06dab0cb204c2 | Havells<br>Havells - Oro 32 One Way Electrical Sw... | 21.0                   | 44                         | Havells<br>Havells - Oro 32 One Way Electrical Sw... |       |
| 1 | aa68675f50a0551b8dadb954017a50a1 | Geol<br>Wooden Wet and Dry Broom                     | 27.0                   | 39                         | Geol<br>Wooden Wet and Dry Broom                     |       |
|   |                                  | Maped  |                        |                            | Maped  |       |

```
merge.to_csv('df.csv', index=False)
```

```
df=pd.read_csv('/content/df.csv')
```

```
#df.isnull().sum()
#df=df.dropna(axis=0)
#df.shape
```

```
#df.head()
```

```
df.columns


Index(['uniq_id', 'Product name in Amazon', 'Retail Price in amazon',
      'Discounted Price in Amazon', 'Product name in flipkart',
      'Retail Price in flipkart', 'Discounted Price in flipkart'],
      dtype='object')
```

Query

```
import re
#search=[]
search=input("Enter Product To Search:\n")
#search=search.split()
```

Enter Product To Search:  
havells

```
df.loc[df['Product name in Amazon'].str.contains(search,na=False, flags=re.IGNORECASE)]
```



|    |                                  | uniq_id | Product<br>name in<br>Amazon                                     | Retail<br>Price<br>in<br>amazon | Discounted<br>Price in<br>Amazon | Product<br>name in<br>flipkart                                   | Retail<br>Price i<br>flipkart |
|----|----------------------------------|---------|--|---------------------------------|----------------------------------|--|-------------------------------|
| 0  | 7c88739f05009f3ed8b06dab0cb204c2 |         | Havells<br>Havells -<br>Oro 32<br>One Way<br>Electrical<br>Sw... | 21.0                            | 44                               | Havells<br>Havells -<br>Oro 32<br>One Way<br>Electrical<br>Sw... | 36.0                          |
| 66 | 447a60a4ffffe50ab35276eb3df37263 |         | Havells<br>Crabtree -<br>Murano<br>10 One<br>Way<br>Electrica... | 169.0                           | 216                              | Havells<br>Crabtree -<br>Murano<br>10 One<br>Way<br>Electrica... | 170.0                         |

