BigBasket Product Recommendor

A Beginner Friendly Guide to Creating Recommendation Systems

Logo

Description automatically generated with medium confidence

Table Of Contents

1. [Introduction](https://www.kaggle.com/code/ayushv322/bigbasket-product-recommendation-system/notebook#section-1)
2. [Importing Necessary Libraries](https://www.kaggle.com/code/ayushv322/bigbasket-product-recommendation-system/notebook#section-2)
3. [Data Loading and Cleaning](https://www.kaggle.com/code/ayushv322/bigbasket-product-recommendation-system/notebook#section-3)
4. [Exploratory Data Analysis](https://www.kaggle.com/code/ayushv322/bigbasket-product-recommendation-system/notebook#section-4)
5. [Demographic Filter Recommendor](https://www.kaggle.com/code/ayushv322/bigbasket-product-recommendation-system/notebook#section-5)
6. [Content Based Recommendor](https://www.kaggle.com/code/ayushv322/bigbasket-product-recommendation-system/notebook#section-6)
7. [Thank You](https://www.kaggle.com/code/ayushv322/bigbasket-product-recommendation-system/notebook#section-99)

Introduction

[Back To Top](https://www.kaggle.com/code/ayushv322/bigbasket-product-recommendation-system/notebook#section-0)

Have you ever noticed how when you seach for a product on Amazon, similar products are shown below it. An example of this is when you search for self-help books, Amazon shows other self-help books that might interest you. This is the job of a recommendation system.  
  
Recommendation Systems are a type of information filtering systems as they improve the quality of search results and provides items that are more relevant to the search item or are realted to the search history of the user. Recommendation system finds similarity between the product a user clicks on and the other products and then recommends if their is some sustainable similarity.  
  
Youtube recommends videos and even decides which video to play next on autoplay. Facebook and Instagram recommend friends and pages to folow. Instagram also recommends reels that you might be interested in. Netflix recommends movies based on user taste and genre selection, it even recommends thumbnails of movies based on user data. Amazon, BigBasket and other E-Commerce websites recommend ads and products based on the items we buy or search for.  
  
In this notebook, we will be building a product recommendor with detailed explanations.  
  
**So Let's Start!!!** There are basically three types of recommender systems:-

1. **Demographic Filtering-** They offer generalized recommendations to every user, based on movie popularity and/or genre. The System recommends the same movies to users with similar demographic features. Since each user is different , this approach is considered to be too simple. The basic idea behind this system is that movies that are more popular and critically acclaimed will have a higher probability of being liked by the average audience.

1. **Content Based Filtering-** They suggest similar items based on a particular item. This system uses item metadata, such as genre, director, description, actors, etc. for movies, to make these recommendations. The general idea behind these recommender systems is that if a person liked a particular item, he or she will also like an item that is similar to it.

1. **Collaborative Filtering-** This system matches persons with similar interests and provides recommendations based on this matching. Collaborative filters do not require item metadata like its content-based counterparts.

Here is the deployment link of the recommendation system: [Deployment Link](https://vermaayush680-bigbasket-product-recommendation-s-mainapp-bp8r67.streamlitapp.com/) Will be creating a separate deployment notebook to teach it.

linkcode

Importing Necessary Libraries

[Back To Top](https://www.kaggle.com/code/ayushv322/bigbasket-product-recommendation-system/notebook#section-0)

In [1]:

*#Basic Libraries*

import numpy as np

import pandas as pd

*#Visualization Libraries*

import matplotlib.pyplot as plt

import seaborn as sns

import plotly.express as px

*#Text Handling Libraries*

import re

from sklearn.feature\_extraction.text import TfidfVectorizer, CountVectorizer

from sklearn.metrics.pairwise import linear\_kernel, cosine\_similarity

Data Loading and Cleaning

[Back To Top](https://www.kaggle.com/code/ayushv322/bigbasket-product-recommendation-system/notebook#section-0)

In [2]:

df = pd.read\_csv('../input/bigbasket-entire-product-list-28k-datapoints/BigBasket Products.csv',index\_col='index')

In [3]:

df.head()

Out[3]:

|  | product | category | sub\_category | brand | sale\_price | market\_price | type | rating | description |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| index |  |  |  |  |  |  |  |  |  |
| 1 | Garlic Oil - Vegetarian Capsule 500 mg | Beauty & Hygiene | Hair Care | Sri Sri Ayurveda | 220.0 | 220.0 | Hair Oil & Serum | 4.1 | This Product contains Garlic Oil that is known... |
| 2 | Water Bottle - Orange | Kitchen, Garden & Pets | Storage & Accessories | Mastercook | 180.0 | 180.0 | Water & Fridge Bottles | 2.3 | Each product is microwave safe (without lid), ... |
| 3 | Brass Angle Deep - Plain, No.2 | Cleaning & Household | Pooja Needs | Trm | 119.0 | 250.0 | Lamp & Lamp Oil | 3.4 | A perfect gift for all occasions, be it your m... |
| 4 | Cereal Flip Lid Container/Storage Jar - Assort... | Cleaning & Household | Bins & Bathroom Ware | Nakoda | 149.0 | 176.0 | Laundry, Storage Baskets | 3.7 | Multipurpose container with an attractive desi... |
| 5 | Creme Soft Soap - For Hands & Body | Beauty & Hygiene | Bath & Hand Wash | Nivea | 162.0 | 162.0 | Bathing Bars & Soaps | 4.4 | Nivea Creme Soft Soap gives your skin the best... |

In [4]:

df.shape

Out[4]:

(27555, 9)

In [5]:

df.isnull().sum()

Out[5]:

product 1

category 0

sub\_category 0

brand 1

sale\_price 0

market\_price 0

type 0

rating 8626

description 115

dtype: int64

In [6]:

print('Percentage Null Data In Each Column')

print('-'\*30)

for col **in** df.columns:

null\_count = df[col].isnull().sum()

total\_count = df.shape[0]

print("**{}** : **{:.2f}**".format(col,null\_count/total\_count \* 100))

Percentage Null Data In Each Column

------------------------------

product : 0.00

category : 0.00

sub\_category : 0.00

brand : 0.00

sale\_price : 0.00

market\_price : 0.00

type : 0.00

rating : 31.30

description : 0.42

In [7]:

print('Total Null Data')

null\_count = df.isnull().sum().sum()

total\_count = np.product(df.shape)

print("**{:.2f}**".format(null\_count/total\_count \* 100))

Total Null Data

3.53

So overall 3% data is missing but 31% of ratings are missing. Since we are going to create a recommender system, let's drop the null values as their will still be over 69% data for recommendation purposes which is enough for us.

In [8]:

df = df.dropna()

In [9]:

df.isnull().sum()

Out[9]:

product 0

category 0

sub\_category 0

brand 0

sale\_price 0

market\_price 0

type 0

rating 0

description 0

dtype: int64

In [10]:

df.shape

Out[10]:

(18840, 9)

In [11]:

*# df.to\_csv('data\_cleaned.csv')*

So even after dropping null data, 18000+ products are available for recommendation.  
Let's recommend now!!

Exploratory Data Analysis

[Back To Top](https://www.kaggle.com/code/ayushv322/bigbasket-product-recommendation-system/notebook#section-0)

In [12]:

df.head()

Out[12]:

|  | product | category | sub\_category | brand | sale\_price | market\_price | type | rating | description |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| index |  |  |  |  |  |  |  |  |  |
| 1 | Garlic Oil - Vegetarian Capsule 500 mg | Beauty & Hygiene | Hair Care | Sri Sri Ayurveda | 220.0 | 220.0 | Hair Oil & Serum | 4.1 | This Product contains Garlic Oil that is known... |
| 2 | Water Bottle - Orange | Kitchen, Garden & Pets | Storage & Accessories | Mastercook | 180.0 | 180.0 | Water & Fridge Bottles | 2.3 | Each product is microwave safe (without lid), ... |
| 3 | Brass Angle Deep - Plain, No.2 | Cleaning & Household | Pooja Needs | Trm | 119.0 | 250.0 | Lamp & Lamp Oil | 3.4 | A perfect gift for all occasions, be it your m... |
| 4 | Cereal Flip Lid Container/Storage Jar - Assort... | Cleaning & Household | Bins & Bathroom Ware | Nakoda | 149.0 | 176.0 | Laundry, Storage Baskets | 3.7 | Multipurpose container with an attractive desi... |
| 5 | Creme Soft Soap - For Hands & Body | Beauty & Hygiene | Bath & Hand Wash | Nivea | 162.0 | 162.0 | Bathing Bars & Soaps | 4.4 | Nivea Creme Soft Soap gives your skin the best... |

In [13]:

counts = df['category'].value\_counts()

counts\_df = pd.DataFrame({'Category':counts.index,'Counts':counts.values})

In [14]:

px.bar(data\_frame=counts\_df,

x='Category',

y='Counts',

color='Counts',

color\_continuous\_scale='blues',

text\_auto=True,

title=f'Count of Items in Each Category')

546024942468236421732091665630495Beauty & HygieneKitchen, Garden & PetsSnacks & Branded FoodsGourmet & World FoodFoodgrains, Oil & MasalaCleaning & HouseholdBakery, Cakes & DairyBeveragesBaby Care010002000300040005000

10002000300040005000CountsCount of Items in Each CategoryCategoryCounts

In [15]:

counts = df['sub\_category'].value\_counts()

counts\_df\_1 = pd.DataFrame({'Category':counts.index,'Counts':counts.values})[:10]

In [16]:

px.bar(data\_frame=counts\_df\_1,

x='Category',

y='Counts',

color='Counts',

color\_continuous\_scale='blues',

text\_auto=True,

title=f'Top 10 Bought Sub\_Categories')

1641818808764658649627621557550Skin CareHair CareBath & Hand WashMasalas & SpicesStorage & AccessoriesMen's GroomingFragrances & DeosCrockery & CutleryReady To Cook & EatOrganic Staples02004006008001000120014001600

6008001000120014001600CountsTop 10 Bought Sub\_CategoriesCategoryCounts

In [17]:

counts = df['brand'].value\_counts()

counts\_df\_brand = pd.DataFrame({'Brand Name':counts.index,'Counts':counts.values})[:10]

In [18]:

px.bar(data\_frame=counts\_df\_brand,

x='Brand Name',

y='Counts',

color='Counts',

color\_continuous\_scale='blues',

text\_auto=True,

title=f'Top 10 Brand Items based on Item Counts')

2781721531391041031011019997bb RoyalBB HomeAmulHimalayaCelloBIOTIQUEDPKeyaOrganic TattvaMTR050100150200250

100150200250CountsTop 10 Brand Items based on Item CountsBrand NameCounts

In [19]:

counts = df['type'].value\_counts()

counts\_df\_type = pd.DataFrame({'Type':counts.index,'Counts':counts.values})[:10]

In [20]:

px.bar(data\_frame=counts\_df\_type,

x='Type',

y='Counts',

color='Counts',

color\_continuous\_scale='blues',

text\_auto=True,

title=f'Top 10 Types of Products based on Item Counts')

1094404390343332322263251234212Face CareMen's DeodorantsShampoo & ConditionerBlended MasalasContainers SetsBathing Bars & SoapsGlasswareBody CareNamkeen & Savoury SnacksHand Wash & Sanitizers02004006008001000

4006008001000CountsTop 10 Types of Products based on Item CountsTypeCounts

Demographic Filter Recommendor

[Back To Top](https://www.kaggle.com/code/ayushv322/bigbasket-product-recommendation-system/notebook#section-0)

Demographic Filtering is like recommeding items based on a feature. Like the top 10 rated items or the top 10 items in a particular category.

In [21]:

def sort\_recommendor(col='rating',sort\_type = False):

*"""*

*A recommendor based on sorting products on the column passed.*

*Arguments to be passed:*

*col: The Feature to be used for recommendation.*

*sort\_type: True for Ascending Order*

*"""*

rated\_recommend = df.copy()

if rated\_recommend[col].dtype == 'O':

col='rating'

rated\_recommend = rated\_recommend.sort\_values(by=col,ascending = sort\_type)

return rated\_recommend[['product','brand','sale\_price','rating']].head(10)

In [22]:

help(sort\_recommendor)

Help on function sort\_recommendor in module \_\_main\_\_:

sort\_recommendor(col='rating', sort\_type=False)

A recommendor based on sorting products on the column passed.

Arguments to be passed:

col: The Feature to be used for recommendation.

sort\_type: True for Ascending Order

In [23]:

sort\_recommendor(col='sale\_price',sort\_type=True)

Out[23]:

|  | product | brand | sale\_price | rating |
| --- | --- | --- | --- | --- |
| index |  |  |  |  |
| 21313 | Serum | Livon | 3.0 | 2.5 |
| 18291 | Sugar Coated Chocolate | Cadbury Gems | 5.0 | 4.2 |
| 21229 | Dish Shine Bar | Exo | 5.0 | 4.2 |
| 14539 | Cadbury Perk - Chocolate Bar | Cadbury | 5.0 | 4.2 |
| 19539 | Layer Cake - Chocolate | Winkies | 5.0 | 4.2 |
| 2979 | Sugar Free Chewing Gum - Mixed Fruit | Orbit | 5.0 | 4.2 |
| 15927 | Dreams Cup Cake - Choco | Elite | 5.0 | 3.9 |
| 6015 | Good Day Butter Cookies | Britannia | 5.0 | 4.1 |
| 27414 | Layer Cake - Orange | Winkies | 5.0 | 4.1 |
| 11307 | Happy Happy Choco-Chip Cookies | Parle | 5.0 | 4.2 |

Notice that our top product has rating of 2.5 which is quite bad so let's filter down by setting a threshold rating.

In [24]:

C= df['rating'].mean()

C

Out[24]:

3.9430626326963902

So the average rating of products is 3.94 Let's use 3.5 as the threshold.

In [25]:

def sort\_recommendor(col='rating',sort\_type = False):

*"""*

*A recommendor based on sorting products on the column passed.*

*Arguments to be passed:*

*col: The Feature to be used for recommendation.*

*sort\_type: True for Ascending Order*

*"""*

rated\_recommend = df.copy().loc[df['rating'] >= 3.5]

if rated\_recommend[col].dtype == 'O':

col='rating'

rated\_recommend = rated\_recommend.sort\_values(by=col,ascending = sort\_type)

return rated\_recommend[['product','brand','sale\_price','rating']].head(10)

In [26]:

sort\_recommendor(col='sale\_price',sort\_type=True)

Out[26]:

|  | product | brand | sale\_price | rating |
| --- | --- | --- | --- | --- |
| index |  |  |  |  |
| 2762 | Orbit Sugar-Free Chewing Gum - Lemon & Lime | Wrigleys | 5.0 | 4.2 |
| 3446 | Marie Light Biscuits - Active | Sunfeast | 5.0 | 4.5 |
| 14604 | 50-50 Timepass Biscuits | Britannia | 5.0 | 3.9 |
| 17641 | Hand Wash - Moisture Shield | Savlon | 5.0 | 4.4 |
| 27491 | 50-50 Timepass Salted Biscuits | Britannia | 5.0 | 4.2 |
| 26585 | Polo - The Mint With The Hole | Nestle | 5.0 | 4.4 |
| 2979 | Sugar Free Chewing Gum - Mixed Fruit | Orbit | 5.0 | 4.2 |
| 19539 | Layer Cake - Chocolate | Winkies | 5.0 | 4.2 |
| 19203 | Bounce Biscuits - Choco Creme | Sunfeast | 5.0 | 4.2 |
| 14539 | Cadbury Perk - Chocolate Bar | Cadbury | 5.0 | 4.2 |

Notice that the 2.5 rated product is not recommended now!! This was our first recommendor.  
Quite easy yet effective and used a lot !!

Content Based Recommendor

[Back To Top](https://www.kaggle.com/code/ayushv322/bigbasket-product-recommendation-system/notebook#section-0)

In [27]:

df.head()

Out[27]:

|  | product | category | sub\_category | brand | sale\_price | market\_price | type | rating | description |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| index |  |  |  |  |  |  |  |  |  |
| 1 | Garlic Oil - Vegetarian Capsule 500 mg | Beauty & Hygiene | Hair Care | Sri Sri Ayurveda | 220.0 | 220.0 | Hair Oil & Serum | 4.1 | This Product contains Garlic Oil that is known... |
| 2 | Water Bottle - Orange | Kitchen, Garden & Pets | Storage & Accessories | Mastercook | 180.0 | 180.0 | Water & Fridge Bottles | 2.3 | Each product is microwave safe (without lid), ... |
| 3 | Brass Angle Deep - Plain, No.2 | Cleaning & Household | Pooja Needs | Trm | 119.0 | 250.0 | Lamp & Lamp Oil | 3.4 | A perfect gift for all occasions, be it your m... |
| 4 | Cereal Flip Lid Container/Storage Jar - Assort... | Cleaning & Household | Bins & Bathroom Ware | Nakoda | 149.0 | 176.0 | Laundry, Storage Baskets | 3.7 | Multipurpose container with an attractive desi... |
| 5 | Creme Soft Soap - For Hands & Body | Beauty & Hygiene | Bath & Hand Wash | Nivea | 162.0 | 162.0 | Bathing Bars & Soaps | 4.4 | Nivea Creme Soft Soap gives your skin the best... |

Let's try using other features such as Category, Sub Category, Brand, Type and Description for much better Recommendation.  
We will be using NLP here to extract useful info from the features especially Description so let's understand TF-IDF before using it.

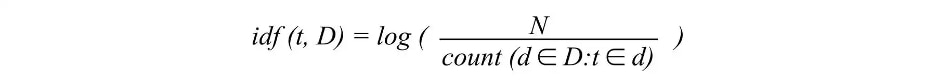
**TF-IDF** stands for **term frequency-inverse document frequency**.

**What is TF(Term Frequency):**

Term frequency works by looking at the frequency of a particular term you are concerned with relative to the document. There are multiple measures, or ways, of defining frequency: Number of times the word appears in a document (raw count).  
  
Term frequency adjusted for the length of the document (raw count of occurences divided by number of words in the document). Logarithmically scaled frequency (e.g. log(1 + raw count)). Boolean frequency (e.g. 1 if the term occurs, or 0 if the term does not occur, in the document).

**What is IDF (inverse document frequency)?**

Inverse document frequency looks at how common (or uncommon) a word is amongst the corpus. IDF is calculated as follows where t is the term (word) we are looking to measure the commonness of and N is the number of documents (d) in the corpus (D)..  
The denominator is simply the number of documents in which the term, t, appears in.



**TF-IDF at work**

Graphical user interface, application, table, Excel

Description automatically generated

[**Source Credits**](https://www.capitalone.com/tech/machine-learning/understanding-tf-idf/)

In [28]:

tfidf = TfidfVectorizer(stop\_words='english')

tfidf\_matrix = tfidf.fit\_transform(df['description'])

tfidf\_matrix.shape

Out[28]:

(18840, 23342)

Now to compute the similarity score, let's use Linear\_Kernel.  
Linear Kernel which Calculates the Dot Product of the tfidf\_matrix and returns an aggreate value depicting the Similarity score.

In [29]:

cosine\_sim = linear\_kernel(tfidf\_matrix, tfidf\_matrix)

cosine\_sim

Out[29]:

array([[1. , 0.01632718, 0.00999603, ..., 0.01056047, 0.01133156,

0. ],

[0.01632718, 1. , 0.00719713, ..., 0. , 0. ,

0. ],

[0.00999603, 0.00719713, 1. , ..., 0.00635776, 0. ,

0. ],

...,

[0.01056047, 0. , 0.00635776, ..., 1. , 0. ,

0. ],

[0.01133156, 0. , 0. , ..., 0. , 1. ,

0. ],

[0. , 0. , 0. , ..., 0. , 0. ,

1. ]])

So we will be recommending items based on similarity score.  
But our problem is that we will be getting back the similarity scores so we will be sorting the scores.  
Now we need a reverse-map to get the title and that is what **indices** is for.

In [30]:

indices = pd.Series(df.index, index=df['product']).drop\_duplicates()

def get\_recommendations\_1(title, cosine\_sim=cosine\_sim):

idx = indices[title]

sim\_scores = list(enumerate(cosine\_sim[idx]))

sim\_scores = sorted(sim\_scores, key=lambda x: x[1], reverse=True)

sim\_scores = sim\_scores[1:11]

movie\_indices = [i[0] for i **in** sim\_scores]

return df['product'].iloc[movie\_indices]

In [31]:

get\_recommendations\_1('Water Bottle - Orange')

Out[31]:

index

1677 Brass Nanda Stand Goblets - No.1

2162 Brass Kachua Stand Deepam - No.1

2756 Brass Angle Deep Stand - Plain, No.2

5400 Brass Lakshmi Deepam - Plain, No.2

6520 Brass Kuber Deepam - No.1

10504 Brass Kuber Deepam - No.2

11226 Brass Angle Deep Stand - Plain, No.3

11504 Brass Angle Deep Stand - Plain, No.1

12699 Brass Kachua Stand Deepam - No.2

18572 Brass Kuber Deepam - No.3

Name: product, dtype: object

In [32]:

get\_recommendations\_1('Cadbury Perk - Chocolate Bar')

Out[32]:

index

27049 Pickle - Mixed

6601 Pickle - Kaduku Mango

17934 Pickle - Mix Vegetable

27105 Pickle - Prawn

3962 Pickle - Tender Mango

16875 Olive Oil - Carrot Pickle

3444 Pickle - Cut Mango

17237 Andhra Special Red Chilli Pickle

27234 Pickle - Lime (South Indian Style)

4955 Pickle - Gooseberry

Name: product, dtype: object

Our search was chocolate yet we got Cashews and Nuts recommended.  
We need to optimize this based on category, sub\_category and brand.

In [33]:

df2 = df.copy()

In [34]:

df2.head()

Out[34]:

|  | product | category | sub\_category | brand | sale\_price | market\_price | type | rating | description |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| index |  |  |  |  |  |  |  |  |  |
| 1 | Garlic Oil - Vegetarian Capsule 500 mg | Beauty & Hygiene | Hair Care | Sri Sri Ayurveda | 220.0 | 220.0 | Hair Oil & Serum | 4.1 | This Product contains Garlic Oil that is known... |
| 2 | Water Bottle - Orange | Kitchen, Garden & Pets | Storage & Accessories | Mastercook | 180.0 | 180.0 | Water & Fridge Bottles | 2.3 | Each product is microwave safe (without lid), ... |
| 3 | Brass Angle Deep - Plain, No.2 | Cleaning & Household | Pooja Needs | Trm | 119.0 | 250.0 | Lamp & Lamp Oil | 3.4 | A perfect gift for all occasions, be it your m... |
| 4 | Cereal Flip Lid Container/Storage Jar - Assort... | Cleaning & Household | Bins & Bathroom Ware | Nakoda | 149.0 | 176.0 | Laundry, Storage Baskets | 3.7 | Multipurpose container with an attractive desi... |
| 5 | Creme Soft Soap - For Hands & Body | Beauty & Hygiene | Bath & Hand Wash | Nivea | 162.0 | 162.0 | Bathing Bars & Soaps | 4.4 | Nivea Creme Soft Soap gives your skin the best... |

In [35]:

df2.shape

Out[35]:

(18840, 9)

Notice that a product can be in multiple catergories and sub\_categories and they are separated with a &.  
Let's split them into a list for futher processes.

In [36]:

rmv\_spc = lambda a:a.strip()

get\_list = lambda a:list(map(rmv\_spc,re.split('& |, |\\*|**\n**', a)))

In [37]:

get\_list('A & B, C')

Out[37]:

['A', 'B', 'C']

In [38]:

for col **in** ['category', 'sub\_category', 'type']:

df2[col] = df2[col].apply(get\_list)

In [39]:

df2.head()

Out[39]:

|  | product | category | sub\_category | brand | sale\_price | market\_price | type | rating | description |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| index |  |  |  |  |  |  |  |  |  |
| 1 | Garlic Oil - Vegetarian Capsule 500 mg | [Beauty, Hygiene] | [Hair Care] | Sri Sri Ayurveda | 220.0 | 220.0 | [Hair Oil, Serum] | 4.1 | This Product contains Garlic Oil that is known... |
| 2 | Water Bottle - Orange | [Kitchen, Garden, Pets] | [Storage, Accessories] | Mastercook | 180.0 | 180.0 | [Water, Fridge Bottles] | 2.3 | Each product is microwave safe (without lid), ... |
| 3 | Brass Angle Deep - Plain, No.2 | [Cleaning, Household] | [Pooja Needs] | Trm | 119.0 | 250.0 | [Lamp, Lamp Oil] | 3.4 | A perfect gift for all occasions, be it your m... |
| 4 | Cereal Flip Lid Container/Storage Jar - Assort... | [Cleaning, Household] | [Bins, Bathroom Ware] | Nakoda | 149.0 | 176.0 | [Laundry, Storage Baskets] | 3.7 | Multipurpose container with an attractive desi... |
| 5 | Creme Soft Soap - For Hands & Body | [Beauty, Hygiene] | [Bath, Hand Wash] | Nivea | 162.0 | 162.0 | [Bathing Bars, Soaps] | 4.4 | Nivea Creme Soft Soap gives your skin the best... |

To avoid duplicacy, we will be converting everything to lowercase and also removing spaces between words.  
This will ensure that our recommendor doesn't consider **Chocolate** of **Cholocate IceCream** and **Chocolate Bar** as the same.

In [40]:

def cleaner(x):

if isinstance(x, list):

return [str.lower(i.replace(" ", "")) for i **in** x]

else:

if isinstance(x, str):

return str.lower(x.replace(" ", ""))

else:

return ''

In [41]:

for col **in** ['category', 'sub\_category', 'type','brand']:

df2[col] = df2[col].apply(cleaner)

In [42]:

df2.head()

Out[42]:

|  | product | category | sub\_category | brand | sale\_price | market\_price | type | rating | description |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| index |  |  |  |  |  |  |  |  |  |
| 1 | Garlic Oil - Vegetarian Capsule 500 mg | [beauty, hygiene] | [haircare] | srisriayurveda | 220.0 | 220.0 | [hairoil, serum] | 4.1 | This Product contains Garlic Oil that is known... |
| 2 | Water Bottle - Orange | [kitchen, garden, pets] | [storage, accessories] | mastercook | 180.0 | 180.0 | [water, fridgebottles] | 2.3 | Each product is microwave safe (without lid), ... |
| 3 | Brass Angle Deep - Plain, No.2 | [cleaning, household] | [poojaneeds] | trm | 119.0 | 250.0 | [lamp, lampoil] | 3.4 | A perfect gift for all occasions, be it your m... |
| 4 | Cereal Flip Lid Container/Storage Jar - Assort... | [cleaning, household] | [bins, bathroomware] | nakoda | 149.0 | 176.0 | [laundry, storagebaskets] | 3.7 | Multipurpose container with an attractive desi... |
| 5 | Creme Soft Soap - For Hands & Body | [beauty, hygiene] | [bath, handwash] | nivea | 162.0 | 162.0 | [bathingbars, soaps] | 4.4 | Nivea Creme Soft Soap gives your skin the best... |

We will now be joining the values of category, sub\_category, type and brand

In [43]:

def couple(x):

return ' '.join(x['category']) + ' ' + ' '.join(x['sub\_category']) + ' '+x['brand']+' ' +' '.join( x['type'])

df2['soup'] = df2.apply(couple, axis=1)

In [44]:

df2['soup'].head()

Out[44]:

index

1 beauty hygiene haircare srisriayurveda hairoil...

2 kitchen garden pets storage accessories master...

3 cleaning household poojaneeds trm lamp lampoil

4 cleaning household bins bathroomware nakoda la...

5 beauty hygiene bath handwash nivea bathingbars...

Name: soup, dtype: object

We need to Count the String Vectors and then compute the Cosine Similarity Score.

In [45]:

df2.head()

Out[45]:

|  | product | category | sub\_category | brand | sale\_price | market\_price | type | rating | description | soup |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| index |  |  |  |  |  |  |  |  |  |  |
| 1 | Garlic Oil - Vegetarian Capsule 500 mg | [beauty, hygiene] | [haircare] | srisriayurveda | 220.0 | 220.0 | [hairoil, serum] | 4.1 | This Product contains Garlic Oil that is known... | beauty hygiene haircare srisriayurveda hairoil... |
| 2 | Water Bottle - Orange | [kitchen, garden, pets] | [storage, accessories] | mastercook | 180.0 | 180.0 | [water, fridgebottles] | 2.3 | Each product is microwave safe (without lid), ... | kitchen garden pets storage accessories master... |
| 3 | Brass Angle Deep - Plain, No.2 | [cleaning, household] | [poojaneeds] | trm | 119.0 | 250.0 | [lamp, lampoil] | 3.4 | A perfect gift for all occasions, be it your m... | cleaning household poojaneeds trm lamp lampoil |
| 4 | Cereal Flip Lid Container/Storage Jar - Assort... | [cleaning, household] | [bins, bathroomware] | nakoda | 149.0 | 176.0 | [laundry, storagebaskets] | 3.7 | Multipurpose container with an attractive desi... | cleaning household bins bathroomware nakoda la... |
| 5 | Creme Soft Soap - For Hands & Body | [beauty, hygiene] | [bath, handwash] | nivea | 162.0 | 162.0 | [bathingbars, soaps] | 4.4 | Nivea Creme Soft Soap gives your skin the best... | beauty hygiene bath handwash nivea bathingbars... |

In [46]:

df2.to\_csv('data\_cleaned\_1.csv')

In [47]:

count = CountVectorizer(stop\_words='english')

count\_matrix = count.fit\_transform(df2['soup'])

We need to Count the String Vectors and then compute the Cosine Similarity Score.

In [48]:

cosine\_sim2 = cosine\_similarity(count\_matrix, count\_matrix)

cosine\_sim2

Out[48]:

array([[1. , 0. , 0. , ..., 0. , 0. ,

0.27216553],

[0. , 1. , 0. , ..., 0. , 0. ,

0. ],

[0. , 0. , 1. , ..., 0. , 0. ,

0. ],

...,

[0. , 0. , 0. , ..., 1. , 0. ,

0. ],

[0. , 0. , 0. , ..., 0. , 1. ,

0. ],

[0.27216553, 0. , 0. , ..., 0. , 0. ,

1. ]])

[Cosine Similarity Documentation](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine_similarity.html#sklearn.metrics.pairwise.cosine_similarity)

In [49]:

df2 = df2.reset\_index()

indices = pd.Series(df2.index, index=df2['product'])

In [50]:

def get\_recommendations\_2(title, cosine\_sim=cosine\_sim):

idx = indices[title]

sim\_scores = list(enumerate(cosine\_sim[idx]))

sim\_scores = sorted(sim\_scores, key=lambda x: x[1], reverse=True)

sim\_scores = sim\_scores[1:11]

movie\_indices = [i[0] for i **in** sim\_scores]

return df2['product'].iloc[movie\_indices]

Comparing Old and New Recommedations

In [51]:

old\_rec = get\_recommendations\_1('Water Bottle - Orange').values

new\_rec = get\_recommendations\_2('Water Bottle - Orange', cosine\_sim2).values

pd.DataFrame({'Old Recommendor': old\_rec,'New Recommendor':new\_rec})

Out[51]:

|  | Old Recommendor | New Recommendor |
| --- | --- | --- |
| 0 | Rectangular Plastic Container - With Lid, Mult... | Glass Water Bottle - Aquaria Organic Purple |
| 1 | Jar - With Lid, Yellow | Glass Water Bottle With Round Base - Transpare... |
| 2 | Round & Flat Storage Container - With lid, Green | H2O Unbreakable Water Bottle - Pink |
| 3 | Premium Rectangular Plastic Container With Lid... | Water Bottle H2O Purple |
| 4 | Premium Round Plastic Container With Lid - Yellow | H2O Unbreakable Water Bottle - Green |
| 5 | Premium Rectangular Plastic Container With Lid... | Regel Tritan Plastic Sports Water Bottle - Black |
| 6 | Premium Round & Flat Storage Container With Li... | Apsara 1 Water Bottle - Assorted Colour |
| 7 | Premium Round Plastic Container With Lid - Blue | Glass Water Bottle With Round Base - Yellow, B... |
| 8 | Premium Round Plastic Container With Lid - Mul... | Trendy Stainless Steel Bottle With Steel Cap -... |
| 9 | Premium Round Plastic Container With Lid - Pink | Penta Plastic Pet Water Bottle - Violet, Wide ... |

In [52]:

old\_rec = get\_recommendations\_1('Cadbury Perk - Chocolate Bar').values

new\_rec = get\_recommendations\_2('Cadbury Perk - Chocolate Bar', cosine\_sim2).values

pd.DataFrame({'Old Recommendor': old\_rec,'New Recommendor':new\_rec})

Out[52]:

|  | Old Recommendor | New Recommendor |
| --- | --- | --- |
| 0 | Cadbury Perk - Chocolate Bar | Nutties Chocolate Pack |
| 1 | Choco Stick - Hexagon Pack | 5 Star Chocolate Bar |
| 2 | Luvit Chocwich White Home Delights 187 g | Dairy Milk Silk - Hazelnut Chocolate Bar |
| 3 | Luvit Chocwich Home Delights 187 g | Perk - Chocolate, Home Treats, 175.5 g, 27 Units |
| 4 | Wafer Biscuits - Chocolate Flavor | Dark Milk Chocolate Bar |
| 5 | Drinking Chocolate - Original | Dairy Milk Silk Mousse - Chocolate Bar |
| 6 | Drinking Chocolate - Original | Dark Milk Chocolate Bar |
| 7 | Biscuit - Bourbon Creams | Chocolate Bar - Fuse |
| 8 | Wafers With Hazelnut Cream | Choclairs Gold Coffee |
| 9 | Choco Stick - Chocolate | 5 Star Chocolate Home Pack, 200 g, 20 units |

Our new recommendation are much better compared to the old ones.

Thank You

If you found it useful, do upvote it!!!

[Back To Top](https://www.kaggle.com/code/ayushv322/bigbasket-product-recommendation-system/notebook#section-0)

Logo

Description automatically generated with medium confidence