Amazon Beauty Product Recommendation

Recommendation Problem :

The objective of the analysis is building Recommendation Engine that recommends Beauty Products to customer based on Product Type by reviewing customer ratings.

UserId | ProductId | Rating | Timestamp

EDA &
Visualization

Web Scraping

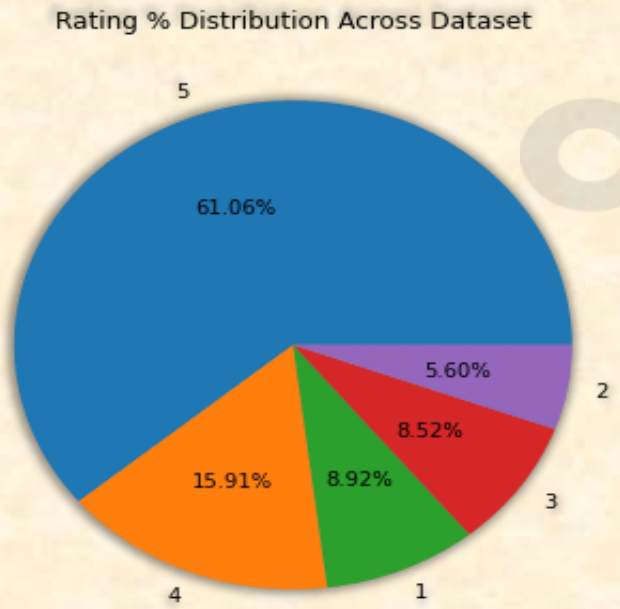
Model Building

Recommendation

Deployment

EDA & Visualization

- Having majority data of rating *5* and least of rating *2*
- Dropped ProductId which are most popular and the ones which are not frequent
- Length of the dataset dropped from *2* Millions to direct *1.34* Million
- With Unique ProductId of *23*




```python
# Creating a function that will drop rows based on value_count of specified column
def drop_by_value_count(data, column, threshold, operator = "="):
    print("Original length of dataset   :", len(data))
    value_counts = data[column].value_counts()
    if operator == "=":
        if isinstance(threshold, list):
            values_to_delete = value_counts[value_counts.isin(threshold)].index
        else:
            values_to_delete = value_counts[value_counts == threshold].index
    elif operator == ">":
        values_to_delete = value_counts[value_counts >= threshold].index
    elif operator == "<":
        values_to_delete = value_counts[value_counts <= threshold].index
    else:
        print("Invalid operator!")
        return
    data.drop(data[data[column].isin(values_to_delete)].index, inplace = True)
    print("Value_counts to drop        :", len(values_to_delete), "\nLength of dataset after drop :", len(data))
```

```python
drop_by_value_count(az, "ProductId", 15, "<")                # droping ProductId which is not frequently bought
drop_by_value_count(az, "ProductId", 7533)                   # droping ProductId which is popularly bought
az.drop(az[az.ProductId == "979078127X"].index, inplace = True)  # droping certain product
```

```
Original length of dataset   : 2023070
Value_counts to drop         : 225411
Length of dataset after drop : 1356461
Original length of dataset   : 1356461
Value_counts to drop         : 1
Length of dataset after drop : 1348928
```

```python
print("Rows containing only Digit ProductId:", len(az["ProductId"][az["ProductId"].apply(lambda x: x.isdigit())]))
print("Unique digit ProductId            :",len(az["ProductId"][az["ProductId"].apply(lambda x: x.isdigit())].value_counts()))
```

```
Rows containing only Digit ProductId: 661
Unique digit ProductId            : 23
```

```python
# Define the product types based on similarities in the product IDs
product_types = {
    "B00L": "Eyeliner & Kajal",
    "B00K": "Talcum Powder",
    "B00J": "Hair Oil",
    "B00I": "Bath Salts",
    "B00H": "Shaving Foam & Gels",
    "B00G": "Trimmers & Clippers",
    "B00F": "Lipstick",
    "B00E": "Body Scrubs",
    "B00D": "Sheet Mask",
    "B00C": "Eye Shadow & Mascara",
    "B00B": "Kits & Accessories",
    "B00A": "Cream & Moisturizer",
    "B009": "Deo & Perfume",
    "B008": "Conditioner",
    "B007": "Body Lotion",
    "B006": "Face Serum",
    "B005": "Hair Color & Heena",
    "B004": "Sunscreen",
    "B003": "Shampoo",
    "B002": "Face Wash & Cleansers",
    "B001": "Shower Gel",
    "B000": "Nail Polish"
}

# Assign product types based on similarities in the product IDs
def assign_product_type(product_id):
    prefix = product_id[:4]
    if prefix in product_types:
        return product_types[prefix]
    else:
        return "Other"

az["ProductType"] = az["ProductId"].apply(assign_product_type)
az = az.reindex(columns = ["UserId", "ProductId", "ProductType", "Rating", "Timestamp"])
```

- Assigning Beauty Product Category to unique 23 ProductId based on starting similarities of ProductId
- This will help to scrape data based on Product Types
- Having *Nail Polish* with highest number of frequency i.e. 5993 and *Eyeliner & Kajal* the least i.e. 11
- And other Product Type within its range

Web Scraping

- Web Scraped Product URL from Amazon using *Selenium* library
- Appending URL to original dataframe based on unique ProductId
- Merging all and have a single dataset by saving it
- Final Dataset have the columns:

  UserId | ProductId | ProductType | Rating | Timestamp | URL

```python
# Function to scrape ProductPage URL using selenium
def extract_product_urls(urls_dict, num_pages):
    browser = webdriver.Chrome()
    dfs = {}

    for product_type, url_template in urls_dict.items():
        urls = []

        for page in range(1, num_pages + 1):
            url = url_template.format(page)
            #print(f"Extracting URLs for {product_type} from page {page}: {url}")

            browser.get(url)
            browser.implicitly_wait(10)

            products = browser.find_elements(By.CSS_SELECTOR, "a.a-link-normal.a-text-normal")
            urls.extend([p.get_attribute("href") for p in products])

        df = pd.DataFrame({"url": urls})
        dfs[product_type] = df

    browser.quit()

    return dfs
```

```python
# Appending URL to original dataframe based on unique ProductId
def merge_url(df1, df2, product_type, length):
    url_dict_list = []
    for i in range(len(df2)):
        unique_product_rows = df1[df1['ProductType'] == product_type[i]]
        unique_product_rows = unique_product_rows.groupby("ProductType")["ProductId"].unique().apply(pd.Series).T
        new_url = df2[i].head(length[i])
        url_dict = pd.concat([unique_product_rows, new_url], axis = 1)
        url_dict.columns = ["ProductId", "URL"]
        url_dict_list.append(url_dict)

    merged_url_dict = pd.concat(url_dict_list, ignore_index = True)
    df1 = pd.merge(df1, merged_url_dict, on = "ProductId", how = "left")
    return df1
```

	UserId	ProductId	ProductType	Rating	Timestamp	URL
0	A3NHUQ33CFH3VM	B00LLPT4HI	Eyeliner & Kajal	5	1405814400	https://www.amazon.in/Maybelline-Colossal-Kaja...
1	A1TIRNQ7O4REOH	B00LLPT4HI	Eyeliner & Kajal	4	1405987200	https://www.amazon.in/Maybelline-Colossal-Kaja...
2	A2Y36BR4YSY9F7	B00LLPT4HI	Eyeliner & Kajal	5	1405728000	https://www.amazon.in/Maybelline-Colossal-Kaja...
3	A23H6FAOLEMAKC	B00LLPT4HI	Eyeliner & Kajal	5	1405814400	https://www.amazon.in/Maybelline-Colossal-Kaja...
4	A3CHYZGF3OO6WD	B00LLPT4HI	Eyeliner & Kajal	5	1405641600	https://www.amazon.in/Maybelline-Colossal-Kaja...

```
# Loading and splitting the data
reader = Reader(rating_scale = (1, 5))
beauty_data = Dataset.load_from_df(beauty[["UserId", "ProductId", "Rating"]], reader)
trainset, testset = train_test_split(beauty_data, test_size = 0.25, random_state = 42)
```

```
# Create Model for recommending
model = SVD(n_factors = 50, reg_all = 0.02, lr_all = 0.005, n_epochs = 20)
model.fit(trainset)

# Predict ratings for testset
test_predictions = model.test(testset)

# Calculate MSE and RMSE
test_mse = accuracy.mse(test_predictions)
test_rmse = accuracy.rmse(test_predictions)
```

```
MSE: 1.5178
RMSE: 1.2320
```

```
model = SVD(n_factors = 50, reg_all = 0.02, lr_all = 0.005, n_epochs = 20)
model.fit(beauty_data.build_full_trainset())
```

Model Building

- On the Final dataset building model using *Surprise* library to recommend
- Where Mean Square Error (MSE) is *1.5178* and Root Mean Square Error (RMSE) is *1.2320*
- It defines a good model to recommend
- Final Model build and fitted with hyperparameters

Recommendation

- User-defined function that takes UserId & ProductType as input and returns Top Beauty Products
- If the UserId is not from the trained data then it will recommend top products of that ProductType


```
# Function to recommend products based on user input
# If user is in the list, use recommender system elseif user is not in the list, recommend popular products
def recommend_products(user_id, product_type):
  if user_id in beauty.UserId:
    product_list = beauty.loc[beauty["ProductType"] == product_type, "ProductId"].unique()
    predictions = [(product_id, model.predict(user_id, product_id).est) for product_id in product_list]
    sorted_predictions = sorted(predictions, key = lambda x: x[1], reverse = True)
    print("Top 5 products for user in the", product_type, "category:")
    for i in range(5):
      product_id = sorted_predictions[i][0]
      url = beauty.loc[beauty["ProductId"] == product_id, "URL"].iloc[0]
      print(i + 1, "- Product ID:", product_id, "\nURL:", url)
  else:
    top_products = beauty.loc[beauty["ProductType"] ==
product_type].groupby("ProductId")["Rating"].mean().sort_values(ascending = False).index[:5]
    print("Top 5 products in the", product_type, "category:")
    for i, product_id in enumerate(top_products):
      url = beauty.loc[beauty["ProductId"] == product_id, "URL"].iloc[0]
      print(i + 1, "- Product ID:", product_id, "\nURL:", url)
```

```
recommend_products("AQ97CPL3HC77S", "Lipstick")

Top 5 products in the Lipstick category:
1 - Product ID: B00FPROWWU
URL: https://www.amazon.in/SWISS-BEAUTY-Swiss-Beauty-Lipstick/dp/B07RWJ89ZD/ref=sr_1_48?pf
BF&pf_rd_p=777489a8-ff36-4ced-b8a2-fe296c0f72d2&pf_rd_r=HAW9B7AZ6E1XMSTNXY9X&pf_rd_s=merch
nements=p_85%3A10440599031%2Cp_28%3A-spons&rps=1&s=beauty&sr=1-48
2 - Product ID: B00FJFUW40
URL: https://www.amazon.in/SUGAR-Cosmetics-Smudge-Liquid-Lipstick/dp/B0822KZ799/ref=sr_1_2
L5WCBF&pf_rd_p=777489a8-ff36-4ced-b8a2-fe296c0f72d2&pf_rd_r=HAW9B7AZ6E1XMSTNXY9X&pf_rd_s=m
refinements=p_85%3A10440599031%2Cp_28%3A-spons&rps=1&s=beauty&sr=1-23
3 - Product ID: B00F48C08S
URL: https://www.amazon.in/SWISS-BEAUTY-Swiss-Beauty-Lipstick/dp/B07RWJ7FM5/ref=sr_1_36?pf
BF&pf_rd_p=777489a8-ff36-4ced-b8a2-fe296c0f72d2&pf_rd_r=HAW9B7AZ6E1XMSTNXY9X&pf_rd_s=merch
nements=p_85%3A10440599031%2Cp_28%3A-spons&rps=1&s=beauty&sr=1-36
4 - Product ID: B00FM8AHSU
URL: https://www.amazon.in/SUGAR-Cosmetics-Lipstick-Transferproof-Waterproof/dp/B09F6J7RNX
_m=A1VBAL9TL5WCBF&pf_rd_p=777489a8-ff36-4ced-b8a2-fe296c0f72d2&pf_rd_r=HAW9B7AZ6E1XMSTNXY9
1680065928&refinements=p_85%3A10440599031%2Cp_28%3A-spons&rps=1&s=beauty&sr=1-12
5 - Product ID: B00FHNR5US
URL: https://www.amazon.in/Lakme-Forever-Liquid-Colour-Espresso/dp/B0828W412X/ref=sr_1_31?
WCBF&pf_rd_p=777489a8-ff36-4ced-b8a2-fe296c0f72d2&pf_rd_r=HAW9B7AZ6E1XMSTNXY9X&pf_rd_s=mer
finements=p_85%3A10440599031%2Cp_28%3A-spons&rps=1&s=beauty&sr=1-31
```

Deployment

Amazon Beauty Recommendation System

Enter your user ID:

A32T585IZ0DJX2

Select a product category:

Bath Salts

User Input parameters

	User ID	Product Type
0	A32T585IZ0DJX2	Bath Salts

Recommend

Top 5 products in the Bath Salts category:

1 - Product ID: B00IM0FSCA URL: https://www.amazon.in/gp/slredirect/picassoRedirect.html/ref=sspa_dk_browse_0?ie=UTF8&adId=A05918851V6HIY2OSSD5M&adOffset=0&qualifier=1680073176&id=3527194268279707&widgetName=sp_browse_thematic&url=%2FKimirica-Valentine-Goodness-Extracts-Pro-Vitamin%2Fdp%2FB099F65W77%3Fpsc%3D1%26pd_rd_w%3D01ewU%26content-id%3Damzn1.sym.ae8aecd9-c026-4fdb-87e7-77b5dfadbf30%26pf_rd_p%3Dae8aecd9-c026-4fdb-87e7-77b5dfadbf30%26pf_rd_r%3DRC8QXFMN5TF268CYB487%26pd_rd_wg%3DXhrRL%26pd_rd_r%3Dffb8a1f1-0ebc-4b6b-999d-c492395ff43f%26ref_%3Dsspa_dk_browse_0

2 - Product ID: B00II4NXO0 URL: https://www.amazon.in/gp/slredirect/picassoRedirect.html/ref=sspa_dk_browse_0?ie=UTF8&adId=A05918851V6HIY2OSSD5M&adOffset=0&qualifier=1680073182&id=8376681105232846&widgetName=sp_browse_thematic&url=%2FKimirica-Valentine-Goodness-Extracts-Pro-Vitamin%2Fdp%2FB099F65W77%3Fpsc%3D1%26pd_rd_w%3DfF7p9%26content-id%3Damzn1.sym.ae8aecd9-c026-4fdb-87e7-77b5dfadbf30%26pf_rd_p%3Dae8aecd9-c026-4fdb-87e7-77b5dfadbf30%26pf_rd_r%3DGN43R3FTAWRHRGNCP0PT%26pd_rd_wg%3Due924%26pd_rd_r%3D320f686b-3a05-4c73-a255-4264d219de7b%26ref_%3Dsspa_dk_browse_0

3 - Product ID: B00IU2C148 URL: https://www.amazon.in/EarthenPot-Epsom-Relax-Muscle-Relieves/dp/B08N5QJKGS/ref=lp_1374281031_1_2?sbo=Tc8eqSFhUI4VwMzbE4fw%2Fw%3D%3D

4 - Product ID: B00IXVY8HO URL: https://www.amazon.in/gp/slredirect/picassoRedirect.html/ref=sspa_dk_browse_5?ie=UTF8&adId=A04899513V5MURO92RAZ1&adOffset=0&qualifier=1680073164&id=328991046792571&widgetName=sp_browse_thematic&url=%2FELYSIUM-ELY1005-Elysium-Epsom-Lavender%2Fdp%2FB07BXKNDV6%3Fpsc%3D1%26pd_rd_w%3Dc9ky9%26content-id%3Damzn1.sym.ae8aecd9-c026-4fdb-87e7-77b5dfadbf30%26pf_rd_p%3Dae8aecd9-c026-4fdb-87e7-77b5dfadbf30%26pf_rd_r%3DHG8N242P5RXXNHSFQ3KZ%26pd_rd_wg%3DyVkQv%26pd_rd_r%3D47b2c1d8-86b0-4d7c-8dc6-6ce5316de0fb%26ref_%3Dsspa_dk_browse_5

5 - Product ID: B00IFVHJBO URL: https://www.amazon.in/gp/slredirect/picassoRedirect.html/ref=sspa_dk_browse_2?ie=UTF8&adId=A04757671KGG9KDUK9U26&adOffset=0&qualifier=1680073184&id=1738560609624601&widgetName=sp_browse_thematic&url=%2FNankings-Unscented-Aching-Muscles-Refreshing%2Fdp%2FB07B2RD412%3Fpsc%3D1%26pd_rd_w%3DEFxrk%26content-id%3Damzn1.sym.ae8aecd9-c026-4fdb-87e7-77b5dfadbf30%26pf_rd_p%3Dae8aecd9-c026-4fdb-87e7-

- Deployed model on *Streamlit* using command prompt
- User Id and Product Type as user input
- Recommend Button to Recommend Beauty Products URL


Thank you