A Project Report On

Myntra Data Set Analysis

Submitted in partial fulfillment of the requirement for the award of the degree

MASTER OF COMPUTER APPLICATIONS
from
Marwadi University

Academic Year 2023 – 24

Yamunesh Patadia (92200584028)
Sahil Sheikh (92200584037)

Internal Guide
Dr. Jaypalsinh Gohil

Marwadi University

Rajkot-Morbi Road, At & PO : Gauridad, Rajkot 360 003. Gujarat. India.

Marwadi University
Marwadi Chandarana Group

Faculty of Computer Applications (FoCA)

Certificate

This is to certify that the project work entitled

Myntra Data Set Analysis

submitted in partial fulfillment of the requirement for the award of the degree of

Master of Computer Applications

of the

Marwadi University

is a result of the bonafide work carried out by

Yamunesh Patadia (92200584028)
Sahil Sheikh (92200584037)

during the academic year 2023 – 2024

Faculty Guide	**HOD**

External Viva

Name of the Examiners	**Signature with Date**

DECLARATION

We hereby declare that this project work entitled Myntra Data Set Analysis is a record done by us.

We also declare that the matter embodied in this project is genuine work done by us and has not been submitted whether to this University or to any other University / Institute for the fulfillment of the requirement of any course of study.

Place: Rajkot

Date: 23th Sep, 2023

Yamunesh Patadia (92200584028) Signature:____________
Sahil Sheikh (92200584037) Signature:____________

ACKNOWLEDGEMENT

It is indeed a great pleasure to express our thanks and gratitude to all those who helped us. No serious and lasting achievement or success one can ever achieve without the help of friendly guidance and co-operation of so many people involved in the work.

We are very thankful to our guide **Dr. Jaypalsinh Gohil,** the person who makes us to follow the right steps during our project work. We express our deep sense of gratitude to for his/her guidance, suggestions and expertise at every stage. A part from that his/her valuable and expertise suggestion during documentation of our report indeed help us a lot.

Thanks to our friend and colleague who have been a source of inspiration and motivation that helped to us during our project work.

We are heartily thankful to the Dean of our department **Dr. R. Sridaran** for giving us an opportunity to work over this project and for their end-less and great support. And to all other people who directly or indirectly supported and help us to fulfil our task.

Yamunesh Patadia (92200584028) Signature:___________
Sahil Sheikh (92200584037) Signature:___________

CONTENTS

1. Introduction

1.1 Objective of the New System:

The objective of the new system is to provide a more efficient and effective way to analyze and visualize Myntra sales data. The current system is manual and time-consuming, and it is difficult to get insights from the data. The new system will automate the data analysis and visualization process, making it easier to identify trends, patterns, and insights.

1.2 Problem Definition:

The current Myntra sales data analysis and visualization process is manual and time-consuming. Data analysts have to spend a lot of time extracting, transforming, cleaning, and analyzing the data before they can create visualizations. This process is also error-prone, and it is difficult to get consistent results.

1.3 Core Components:

The new system will consist of the following core components:

- Data extraction component: This component will be responsible for extracting data from the Myntra sales database.
- Data transformation component: This component will be responsible for transforming the data into a format that is suitable for analysis and visualization.
- Data cleaning component: This component will be responsible for identifying and correcting errors in the data.
- Data analysis component: This component will be responsible for performing statistical analysis on the data to identify trends, patterns, and insights.
- Data visualization component: This component will be responsible for creating visualizations of the data, such as charts, graphs, and maps.

1.4 Project Profile:

The project will be completed in three phases:

Phase 1: Requirements gathering and analysis.
Phase 2: System design and development.
Phase 3: System testing and deployment.

The project is expected to take six months to complete.

1.5 Assumptions and Constraints:

The following assumptions and constraints have been identified for the project:

- The Myntra sales database is accessible and well-maintained.
- The required hardware and software resources are available.
- The project team has the necessary skills and experience.
- The project budget is sufficient.

1.6 Advantages and Limitations of the Proposed System

The advantages of the proposed system include:

- Increased efficiency and effectiveness of data analysis and visualization.
- Reduced errors in the data analysis and visualization process.
- Improved ability to identify trends, patterns, and insights from the data.
- Increased accessibility of the data analysis and visualization results to users.

The limitations of the proposed system include:

- The system is dependent on the accuracy and completeness of the Myntra sales database.
- The system requires users to have some basic knowledge of data analysis and visualization.
- The system may not be able to handle all types of Myntra sales data analysis and visualization requests.

Overall, the proposed system is a viable solution to the problem of manual and time-consuming Myntra sales data analysis and visualization. The system has the potential to improve the efficiency and effectiveness of the data analysis and visualization process, and to provide users with more insights into the Myntra sales data.

2. Requirement Determination & Analysis

2.1 Requirement Determination

The requirements for the proposed system were determined through a process of user interviews, surveys, and focus groups. The following are the key requirements for the system:

- The system should be able to extract data from the Myntra sales database.
- The system should be able to transform the data into a format that is suitable for analysis and visualization.
- The system should be able to clean the data to identify and correct errors.
- The system should be able to perform statistical analysis on the data to identify trends, patterns, and insights.
- The system should be able to create visualizations of the data, such as charts, graphs, and maps.
- The system should be easy to use and accessible to users with a variety of skill levels.
- The system should be scalable to handle large datasets.

2.2 Targeted Users

The targeted users of the proposed system are data analysts and business users at Myntra. The system will be used by data analysts to analyze the Myntra sales data and identify trends, patterns, and insights. The system will be used by business users to make informed decisions about product development, marketing, and sales.

2.3 Tool details (Python / PowerBI/ Tableau)

The proposed system will be developed using Python. Python is a popular programming language that is well-suited for data analysis and visualization. There are a number of Python libraries that can be used for data extraction, transformation, cleaning, analysis, and visualization.

The following Python libraries will be used to develop the proposed system:

- Pandas: A Python library for data manipulation and analysis.
- NumPy: A Python library for working with arrays.
- Matplotlib: A Python library for creating visualizations.
- Seaborn: A Python library for creating statistical graphics.

2.4 Library description (Details on various libraries / packages used)

- Pandas: Pandas is a Python library for data manipulation and analysis. It provides a number of data structures and tools for working with tabular data, such as DataFrames and Series. Pandas also provides a number of functions for data manipulation, such as filtering, sorting, and grouping.
- NumPy: NumPy is a Python library for working with arrays. It provides a number of functions for creating, manipulating, and analyzing arrays. NumPy is also used to perform mathematical operations on arrays.
- Matplotlib: Matplotlib is a Python library for creating visualizations. It provides a number of plotting functions for creating charts, graphs, and other types of visualizations.
- Seaborn: Seaborn is a Python library for creating statistical graphics. It builds on Matplotlib and provides a higher-level interface for creating visualizations. Seaborn also provides a number of pre-defined statistical plots, such as histograms, scatter plots, and box plots.

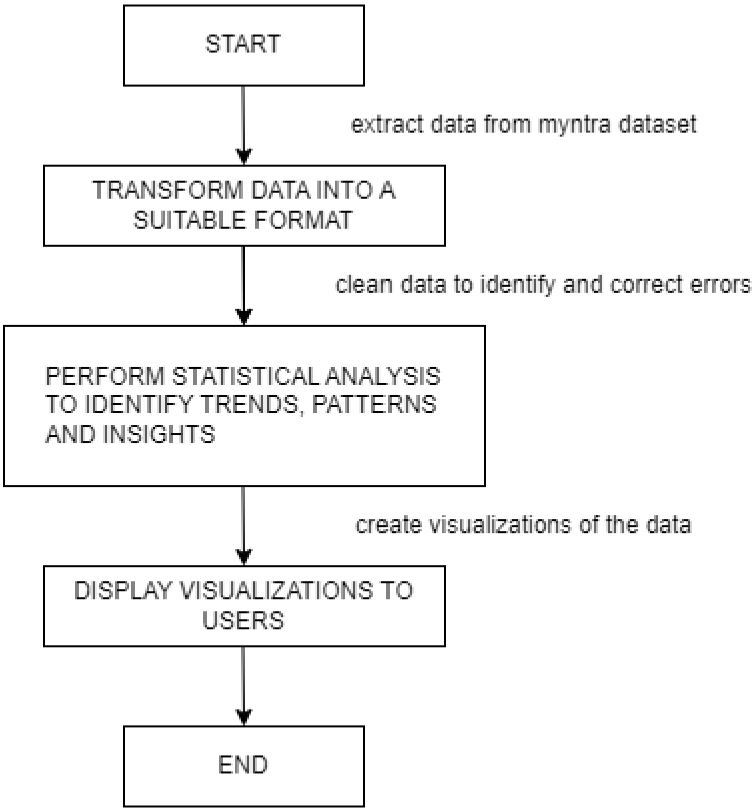
In addition to the above libraries, a number of other Python libraries may be used to develop the proposed system, such as:

- Scikit-learn: A Python library for machine learning.
- Statsmodels: A Python library for statistical modeling.
- BeautifulSoup: A Python library for parsing HTML and XML documents.
- Requests: A Python library for making HTTP requests.

The specific libraries that are used will depend on the specific requirements of the system.

3. System Design

3.1 Flowchart


START
extract data from myntra dataset
TRANSFORM DATA INTO A SUITABLE FORMAT
clean data to identify and correct errors
PERFORM STATISTICAL ANALYSIS TO IDENTIFY TRENDS, PATTERNS AND INSIGHTS
create visualizations of the data
DISPLAY VISUALIZATIONS TO USERS
END

3.2 Dataset Design

The dataset for the proposed system will be stored in a relational database. The database will consist of the following tables:

- Products: This table will store information about the products that are sold on Myntra, such as product name, product category, and product price.
- Sales: This table will store information about the sales of the products on Myntra, such as order date, ship date, and quantity sold.
- Customers: This table will store information about the customers who purchase products from Myntra, such as customer name, customer email address, and customer shipping address.

The tables will be linked together using foreign keys. For example, the Sales table will have a foreign key to the Products table and a foreign key to the Customers table.

3.3 Details on preprocessing steps applied

The following preprocessing steps will be applied to the data before it is analyzed and visualized:

- Data cleaning: This will involve identifying and correcting errors in the data, such as duplicate records, missing values, and incorrect data types.
- Data transformation: This will involve converting the data into a format that is suitable for analysis and visualization. For example, the data may be converted from a string format to a numeric format.
- Feature engineering: This will involve creating new features from the existing data. For example, a new feature could be created to represent the total sales of a product in each period.

3.4 UI design

The UI for the proposed system will be a web-based application. The UI will be designed to be easy to use and accessible to users with a variety of skill levels. The UI will consist of the following main components:

- Data selection: This component will allow users to select the data that they want to analyze and visualize.
- Analysis options: This component will allow users to select the type of analysis that they want to perform.
- Visualization options: This component will allow users to select the type of visualization that they want to create.
- Visualization display: This component will display the visualizations that have been created.

The UI will also provide users with the ability to export the visualizations to other formats, such as PDF, PNG, and JPEG.

Additional considerations

The following additional considerations should be taken into account when designing the system:

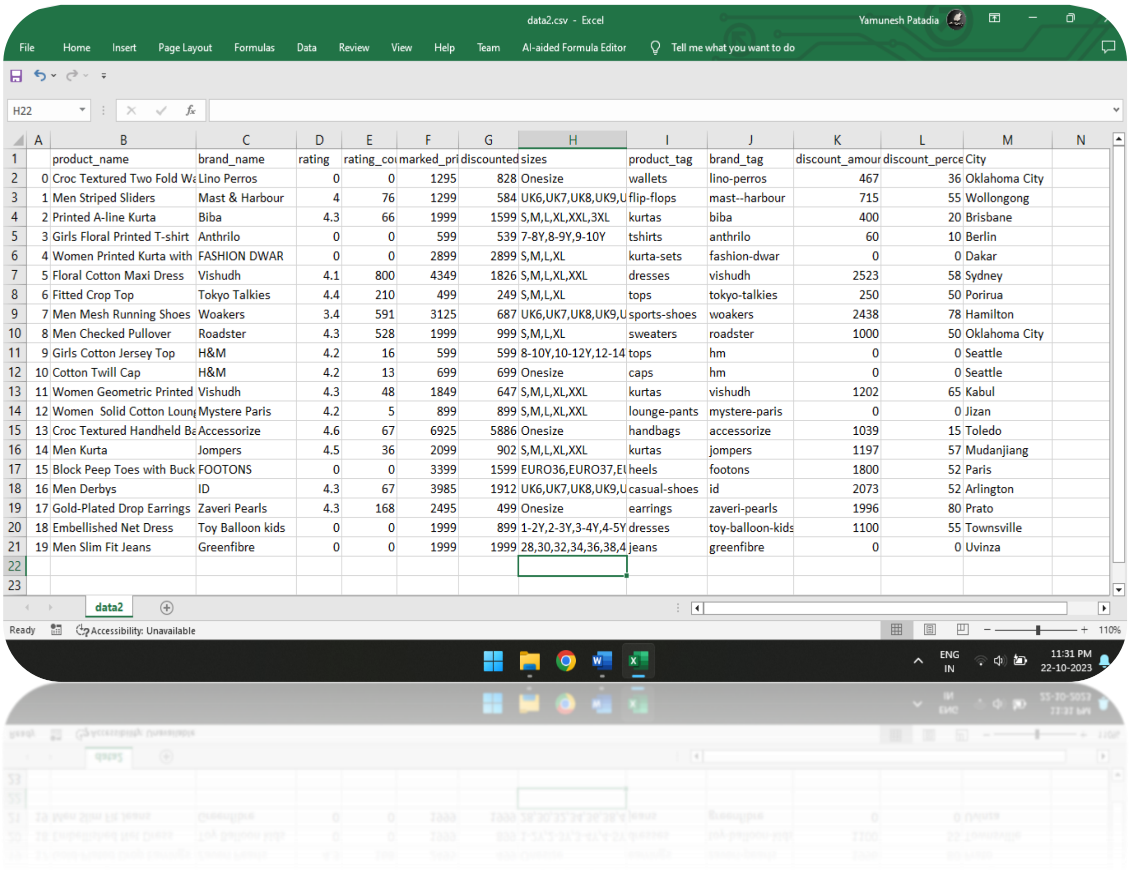
- Scalability: The system should be designed to be scalable to handle large datasets.
- Security: The system should be designed to be secure and protect the data from unauthorized access.
- Documentation: The system should be well-documented so that users can easily understand how to use it.

The system will be developed using Agile development methodology. This will allow the system to be developed and delivered in a timely and iterative manner. The system will be tested using a variety of testing methods, including unit testing, integration testing, and system testing.

4. Development

4.1 Code

Myntra Sales Dataset:



	product_name	brand_name	rating	rating_co	marked_pri	discounted	sizes	product_tag	brand_tag	discount_amour	discount_perce	City
0	Croc Textured Two Fold Wa	Lino Perros	0	0	1295	828	Onesize	wallets	lino-perros	467	36	Oklahoma City
1	Men Striped Sliders	Mast & Harbour	4	76	1299	584	UK6,UK7,UK8,UK9,U	flip-flops	mast--harbour	715	55	Wollongong
2	Printed A-line Kurta	Biba	4.3	66	1999	1599	S,M,L,XL,XXL,3XL	kurtas	biba	400	20	Brisbane
3	Girls Floral Printed T-shirt	Anthrilo	0	0	599	539	7-8Y,8-9Y,9-10Y	tshirts	anthrilo	60	10	Berlin
4	Women Printed Kurta with	FASHION DWAR	0	0	2899	2899	S,M,L,XL	kurta-sets	fashion-dwar	0	0	Dakar
5	Floral Cotton Maxi Dress	Vishudh	4.1	800	4349	1826	S,M,L,XL,XXL	dresses	vishudh	2523	58	Sydney
6	Fitted Crop Top	Tokyo Talkies	4.4	210	499	249	S,M,L,XL	tops	tokyo-talkies	250	50	Porirua
7	Men Mesh Running Shoes	Woakers	3.4	591	3125	687	UK6,UK7,UK8,UK9,U	sports-shoes	woakers	2438	78	Hamilton
8	Men Checked Pullover	Roadster	4.3	528	1999	999	S,M,L,XL	sweaters	roadster	1000	50	Oklahoma City
9	Girls Cotton Jersey Top	H&M	4.2	16	599	599	8-10Y,10-12Y,12-14	tops	hm	0	0	Seattle
10	Cotton Twill Cap	H&M	4.2	13	699	699	Onesize	caps	hm	0	0	Seattle
11	Women Geometric Printed	Vishudh	4.3	48	1849	647	S,M,L,XL,XXL	kurtas	vishudh	1202	65	Kabul
12	Women Solid Cotton Loung	Mystere Paris	4.2	5	899	899	S,M,L,XL,XXL	lounge-pants	mystere-paris	0	0	Jizan
13	Croc Textured Handheld Ba	Accessorize	4.6	67	6925	5886	Onesize	handbags	accessorize	1039	15	Toledo
14	Men Kurta	Jompers	4.5	36	2099	902	S,M,L,XL,XXL	kurtas	jompers	1197	57	Mudanjiang
15	Block Peep Toes with Buck	FOOTONS	0	0	3399	1599	EURO36,EURO37,EU	heels	footons	1800	52	Paris
16	Men Derbys	ID	4.3	67	3985	1912	UK6,UK7,UK8,UK9,U	casual-shoes	id	2073	52	Arlington
17	Gold-Plated Drop Earrings	Zaveri Pearls	4.3	168	2495	499	Onesize	earrings	zaveri-pearls	1996	80	Prato
18	Embellished Net Dress	Toy Balloon kids	0	0	1999	899	1-2Y,2-3Y,3-4Y,4-5Y	dresses	toy-balloon-kids	1100	55	Townsville
19	Men Slim Fit Jeans	Greenfibre	0	0	1999	1999	28,30,32,34,36,38,4	jeans	greenfibre	0	0	Uvinza

```python
# Import necessary libraries
import pandas as pd
```

```python
# Load the dataset
data = pd.read_csv('MyntraDataSet.csv')
```

```python
# Display the first five rows of the dataset
print(data.head())
```

Output :

```
                        product_name       brand_name  rating  rating_count  \
 0    Croc  Textured Two Fold Wallet      Lino  Perros     0.0             0
 1               Men Striped Sliders   Mast & Harbour     4.0            76
```

```
2           Printed A-line Kurta              Biba     4.3           66
3    Girls Floral Printed T-shirt         Anthrilo     0.0            0
4  Women Printed Kurta with Skirt     FASHION DWAR     0.0            0


          marked_price  discounted_price                   sizes      \
0         1295          828                              Onesize
1         1299          584               UK6,UK7,UK8,UK9,UK10,UK11
2         1999          1599                        S,M,L,XL,XXL,3XL
3         599           539                          7-8Y,8-9Y,9-10Y
4         2899          2899                                S,M,L,XL


                                      product_link             \
0           wallets/lino-perros/lino-perros-women-peach-co...
1           flip-flops/mast--harbour/mast--harbour-men-nav...
2           kurtas/biba/biba-women-off-white--black-printe...
3           tshirts/anthrilo/anthrilo-girls-white-floral-p...
4           kurta-sets/fashion-dwar/fashion-dwar-women-mul...


                                                img_link      product_tag \
0       https://assets.myntassets.com/dpr_2,q_60,w_210...  wallets
1       https://assets.myntassets.com/dpr_2,q_60,w_210...  flip-flops
2       https://assets.myntassets.com/dpr_2,q_60,w_210...  kurtas
3       https://assets.myntassets.com/dpr_2,q_60,w_210...  tshirts
4       https://assets.myntassets.com/dpr_2,q_60,w_210...  kurta-sets

        brand_tag  discount_amount  discount_percent  Unnamed: 13  Order Date\
0      lino-perros              467                36      NaN      11-11-2021
1    mast--harbour              715                55      NaN      05-02-2021
2             biba              400                20      NaN      17-10-2021
3         anthrilo               60                10      NaN      28-01-2021
4      fashion-dwar               0                 0      NaN      05-11-2021

        Ship Date           City
0      13-11-2021  Oklahoma City
1      07-02-2021     Wollongong
2      18-10-2021       Brisbane
3      30-01-2021         Berlin
4      06-11-2021          Dakar
```

```
# Summary statistics
summary = data.describe()
summary
```

Output :

```
             Rating  rating_count  marked_price discounted_price discount_amount \
count  52038.000000  52038.000000  52038.000000     52038.000000    52038.000000
mean       2.066327  60.506514     2472.660248      1481.337696      991.322553
std        2.103646  585.330688    2318.276451      1689.222533     1266.709366
min        0.000000  0.000000        55.000000        49.000000        0.000000
25%        0.000000  0.000000     1248.000000       664.000000      188.000000
50%        0.000000  0.000000     1990.000000       999.000000      700.000000
75%        4.200000  20.000000    2995.000000      1708.750000     1320.000000
max        5.000000  55900.0000   113999.0000     45900.000000    68400.000000

discount_percent      Unnamed: 13
    52038.000000              0.0
```

```
             37.148757              NaN
             24.889723              NaN
              0.000000              NaN
             15.000000              NaN
             40.000000              NaN
             60.000000              NaN
             90.000000              NaN
```

```
# Data types and missing values
info = data.info()
info
```

Output :

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 52038 entries, 0 to 52037
Data columns (total 17 columns):
#      Column               Non-Null Count         Dtype
0      product_name         52038 non-null          object
1      brand_name           52038 non-null         object
2      rating               52038 non-null         float64
3      rating_count         52038 non-null         int64
4      marked_price         52038 non-null         int64
5      discounted_price     52038 non-null         int64
6      sizes                52038 non-null         object
7      product_link         52038 non-null         object
8      img_link             52038 non-null         object
9      product_tag          52038 non-null         object
10     brand_tag            52038 non-null         object
11     discount_amount      52038 non-null         int64
12     discount_percent     52038 non-null         int64
13     Unnamed: 13          0 non-null             float64
14     Order Date           51290  non-null        object
15     Ship Date            51290  non-null        object
16     City                 51290  non-null        object
dtypes: float64(2), int64(5), object(10)
memory usage: 6.7+ MB
```

```
# Unique values in categorical columns
unique_values = data.nunique()
unique_values
```

Output :

```
 product_name      18710
 brand_name         2658
 rating             40
 rating_count       870
 marked_price       2020
 discounted_price  3169
 sizes              1694
 product_link       43646
 img_link           43645
 product_tag        315
 brand_tag          2658
 discount_amount   3075
 discount_percent  91
```

```
Unnamed: 13          0
Order Date         366
Ship Date          373
City              3650
dtype: int64
```

```
# Get the data types of each column in the 'data' DataFrame
data_types = data.dtypes
data_types
```

Output :

```
product_name          object
brand_name            object
rating               float64
rating_count           int64
marked_price           int64
discounted_price       int64
sizes                 object
product_link          object
img_link              object
product_tag           object
brand_tag             object
discount_amount        int64
discount_percent       int64
Unnamed: 13          float64
Order Date            object
Ship Date             object
City                  object
dtype: object
```

```
# Check for missing values
missing_values = data.isnull().sum()
missing_values
```

Output :

```
brand_name              0
rating                  0
rating_count            0
marked_price            0
discounted_price        0
sizes                   0
product_link            0
img_link                0
product_tag             0
brand_tag               0
discount_amount         0
discount_percent        0
Unnamed: 13         52038
Order Date            748
Ship Date             748
City                  748
dtype: int64
```

```
# Removing unwanted columns from the dataset.
data.drop(['product_link', 'img_link','Unnamed: 13'], axis=1,
inplace=True)
```

```
# filling missing of Order Date column in dataset
mode_of_order_date=data['Order Date'].mode()[0]
data['Order Date'].fillna(value=mode_of_order_date,inplace=True)
```

```
# filling missing of Ship Date column in dataset
mode_of_ship_date=data['Ship Date'].mode()[0]
data['Ship Date'].fillna(value=mode_of_ship_date,inplace=True)
```

```
# filling missing of City column in dataset
mode_of_city=data['City'].mode()[0]
data['City'].fillna(value=mode_of_city,inplace=True)
```

```
# obtain the shape of a DataFrame
data.shape
```

Output :

```
(52038, 14)
```

```
# Calculate the average rating and total rating count from the 'data'
dataset.
avg_rating = data['rating'].mean()  # Calculate the mean of the
'rating' column.
total_rating_count = data['rating_count'].sum()  # Calculate the sum
of 'rating_count' column.

# Display the results.
print("Average Rating:", avg_rating)
print("Total Rating Count:", total_rating_count)
```

Output :

```
Average Rating: 2.066326530612245
Total Rating Count: 3148638
```

```
# Calculate the average discount percentage by brand and display the
top 10 results.

# Group the data by 'brand_name' and calculate the mean of
'discount_percent' within each group.
avg_discount=
data.groupby('brand_name')['discount_percent'].mean().head(10)

# Print the calculated average discounts for the top 10 brands.
print("Average Discount by Brand:")
print(avg_discount)
```

Output :

```
Average Discount by Brand:
brand_name
1 Stop Fashion        75.000000
20Dresses             27.041885
39 THREADS            40.000000
3PIN                  43.000000
4711                  27.500000
513                   40.000000
7Threads              70.454545
883 Police            33.600000
98 Degree North       63.090909
999Store              64.941176
Name: discount_percent, dtype: float64
```

```
# Sort the 'data' DataFrame by 'discounted_price' in descending order
# to show the highest discounted prices first, and display the top
rows.
sorted_df = data.sort_values(by='discounted_price', ascending=False)
sorted_df.head()
```

Output :

```
       product_name                 brand_name  rating  rating_count    marked_price  \

29599  Men Automatic Motion Watch    D1 Milano     0.0             0           51000
27039   Lord Krishna Showpiece     eCraftIndia     0.0             0          113999
13309   Women Square Sunglasses       Tom Ford     0.0             0           41900
20538  Women Aviator Sunglasses       Tom Ford     0.0             0           40900
 8809   Casual Shirt Polo Ralph         Lauren     0.0             0           39000


discounted_price        sizes    product_tag          brand_tag   discount_amount  \

           45900      Onesize        watches          d1-milano              5100
           45599      Onesize    showpieces         ecraftindia             68400
           41900            M    sunglasses            tom-ford                 0
           40900            L    sunglasses            tom-ford                 0
           39000   38,42.5,44        shirts   polo-ralph-lauren                 0


discount_percent      Order Date      Ship Date             City
```

```
              10      01-11-2021    05-11-2021    Los Angeles
              60      20-08-2021    24-08-2021       Lawrence
               0      08-07-2021    14-07-2021         Sincan
               0      03-08-2021    08-08-2021         Riyadh
               0      17-06-2021    17-06-2021      Carrefour
```

```
# Find the index of the brand and product with the highest total
sales.
sales_by_brand_tag =
data.groupby(['brand_tag', 'product_tag'])['discounted_price'].sum()
sales_by_brand_tag

max_sales_index = sales_by_brand_tag.idxmax()
print("Brand and Product with Highest Total Sales:")
max_sales_index
```

Output :

```
Brand and Product with Highest Total Sales:
('jc-collection', 'dresses')
```

```
# Calculate and retrieve the top 15 cities with the highest average
discount amounts
avg_discount_by_city=
data.sort_values(by='discount_amount', ascending=False).head(15)

# Display the resulting DataFrame containing the cities and their
average discounts
avg_discount_by_city.head()
```

Output :

```
                                           product_name          brand_name rating \

27039                            Lord Krishna Showpiece          eCraftIndia    0.0
18316  Textured 360-Degree Rotation Hard-Sided Trolle...              Safari    0.0
25265             Gold-Plated Stone-Studded Jewellery Set  Silvermerc Designs    0.0
24736                           Gold Plated Jewellery Set  Silvermerc Designs    0.0
15422                           Gold Plated Jewellery Set  Silvermerc Designs    0.0


rating_count  marked_price  discounted_price  sizes   product_tag          brand_tag \

          0        113999             45599  Onesize    showpieces        ecraftindia
          0         32997              9239     Pack   trolley-bag             safari
          0         29500              5900  Onesize  jewellery-set  silvermerc-designs
          0         29000              5800  Onesize  jewellery-set  silvermerc-designs
          0         29000              5800  Onesize  jewellery-set  silvermerc-designs


discount_amount  discount_percent  Order Date   Ship Date       City

          68400                60  20-08-2021  24-08-2021  Lawrence
          23758                72  01-06-2021  06-06-2021    Zanjan
```

```
   23600              80  06-10-2021  10-10-2021     Rugby
   23200              80  27-09-2021  30-09-2021   Detroit
   23200              80  08-08-2021  14-08-2021    Harrow
```

```
# Analyzing Discounts by Brand and City

# Grouping the data by 'brand_name' and calculating the average
discount percentage for each brand.
avg_discount_by_brand =
data.groupby('brand_name')['discount_percent'].mean()

# Grouping the data by 'City' and finding the maximum discount
percentage offered in each city.

max_discount_by_city =
data.groupby('City')['discount_percent'].max()
```

```
# The 'avg_discount_by_brand' Series now contains the average
discount percentage for each brand,
# which provides insights into how brands are pricing their products.
avg_discount_by_brand
```

Output :

```
brand_name
1 Stop Fashion  75.000000
20Dresses       27.041885
39 THREADS      40.000000
3PIN            43.000000
4711            27.500000
                  ...
x2o             74.000000
yelloe          51.666667
yoho            29.000000
zebu            57.400000
zink Z          10.000000
Name: discount_percent, Length: 2658, dtype: float64
```

```
# The 'max_discount_by_city' Series displays the maximum discount
percentage available in each city,
# helping to identify where customers can find the highest discounts.
max_discount_by_city
```

Output :

```
City
Aachen                 83
Aalen                  0
Aalst                  60
```

```
Aba                      86
Abadan                   60
                        ...
Zwedru                   53
Zwickau                  30
Zwolle                   50
eMbalenhle               25
Águas Lindas de Goiás  65
Name: discount_percent, Length: 3650, dtype: int64
```

```
# Calculate the discounted percentage for each item in the dataset.
# The discounted percentage is obtained by dividing the discount
amount
# by the marked price and then multiplying by 100 to express it as a
percentage.

data['discounted_percent'] = (data['discount_amount'] /
data['marked_price']) * 100
```

```
data.head()
```

```
Output :

                     product_name       brand_name  rating  rating_count  marked_price \

0   Croc Textured Two Fold Wallet      Lino Perros     0.0             0          1295
1             Men Striped Sliders   Mast & Harbour     4.0            76          1299
2            Printed A-line Kurta             Biba     4.3            66          1999
3    Girls Floral Printed T-shirt         Anthrilo     0.0             0           599
4  Women Printed Kurta with Skirt      FASHION DWAR     0.0             0          2899


discounted_price                      sizes  product_tag     brand_tag  discount_amount \

             828                    Onesize      wallets   lino-perros              467
             584  UK6,UK7,UK8,UK9,UK10,UK11   flip-flops  mast—harbour              715
             1599            S,M,L,XL,XXL,3XL       kurtas          biba              400
             539               7-8Y,8-9Y,9-10Y      tshirts      anthrilo               60
             2899                    S,M,L,XL   kurta-sets   fashion-dwar                0


discount_percent   Order Date   Ship Date           City  discounted_percent

              36   11-11-2021  13-11-2021  Oklahoma City           36.061776
              55   05-02-2021  07-02-2021     Wollongong           55.042340
              20   17-10-2021  18-10-2021       Brisbane           20.010005
              10   28-01-2021  30-01-2021         Berlin           10.016694
               0   05-11-2021  06-11-2021          Dakar            0.000000
```

```
# Sort the 'data' dataset by 'rating' in descending order to prioritize higher-rated entries.
# This allows us to explore the dataset with the most positively rated items at the top.
sorted_by_rating = data.sort_values(by='rating', ascending=False)

# Display the sorted dataset.
sorted_by_rating.head()
```

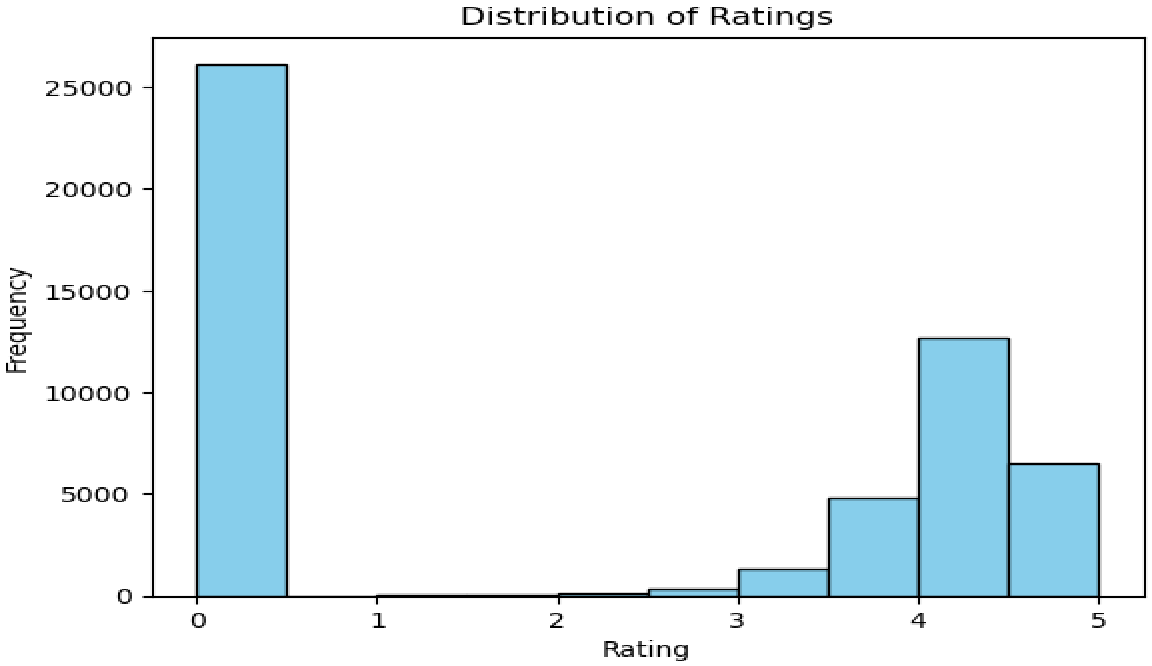
Output :

```
                           product_name             brand_name  rating  rating_count  \

32472               Opaque Casual Shirt                URBANIC     5.0             5
49615            Checked Pinafore Dress             Nauti Nati     5.0             4
6509   Printed Elevated Bottom Jumpsuit                Juniper     5.0             5
45510         Pack of 4 Patterned Socks                Bonjour     5.0             5
5130                      EDGE T-shirts  HRX by Hrithik Roshan     5.0             6


marked_price  discounted_price            sizes  product_tag           brand_tag  \

        1490               745           M,L,XL       shirts             urbanic
        1799               719   4Y,5Y,6Y,7Y,8Y      dresses          nauti-nati
        2799               951     S,M,L,XL,XXL     jumpsuit             juniper
         396               396             6-8Y        socks             bonjour
        2199              1209      XS,S,M,L,XL      tshirts  hrx-by-hrithik-roshan


discount_amount  discount_percent  Order Date   Ship Date          City discounted_percent

            745                50  23-09-2021  29-09-2021       Goiânia          50.000000
           1080                60  18-11-2021  23-11-2021       Brumado          60.033352
           1848                66  26-06-2021  28-06-2021       Pingnan          66.023580
              0                 0  25-11-2021  30-11-2021  Buenos Aires          0.000000
            990                45  16-07-2021  20-07-2021          Genk          45.020464
```

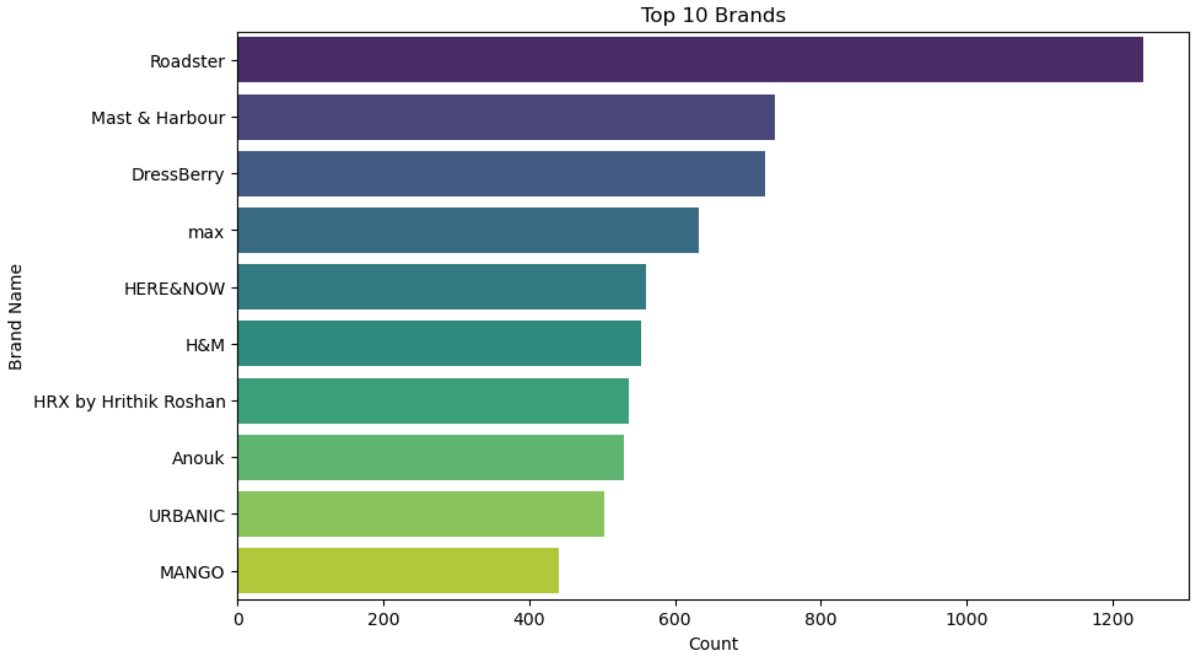
4.2 Screen Shots

```
# Data Visualization
import matplotlib.pyplot as plt
import seaborn as sns
```

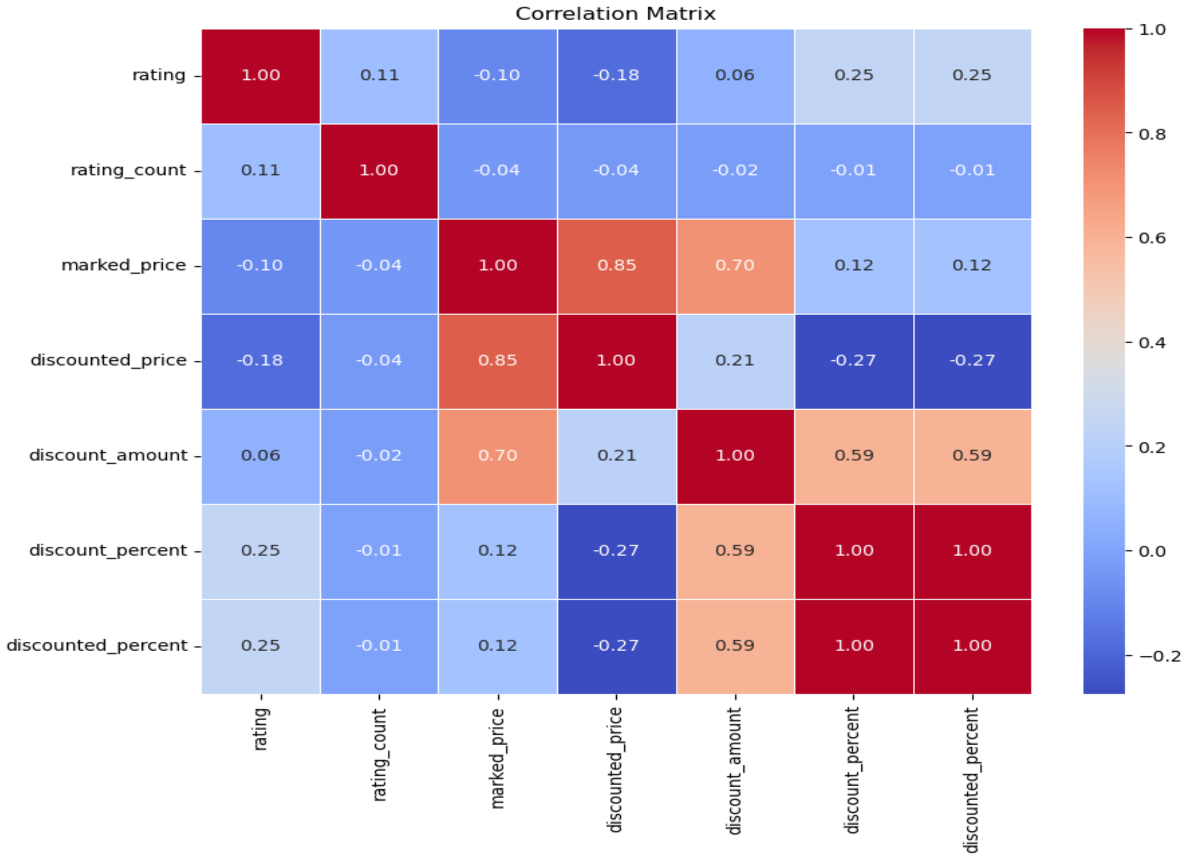
```
# Create a histogram of ratings
# This code generates a histogram to visualize the distribution of
ratings in the 'data' dataset.
# - The 'plt.hist' function is used to create the histogram.
# - We specify 'data['rating']' as the data source, 'bins=10' for
10 equally spaced bins,
#   'color='skyblue'' to set the histogram bars' color, and
'edgecolor='black'' to define the edge color.
# - 'plt.xlabel' and 'plt.ylabel' set labels for the x and y axes,
respectively.
# - 'plt.title' assigns a title to the histogram.
# - Finally, 'plt.show()' displays the histogram.

plt.hist(data['rating'], bins=10, color='skyblue',
edgecolor='black')
plt.xlabel('Rating')
plt.ylabel('Frequency')
plt.title('Distribution of Ratings')
plt.show()
```



```
# Bar Chart of Brands
import seaborn as sns

# Create a bar chart of the top 10 brands from the 'data' dataset
using Seaborn.

# Extract the counts of each brand and select the top 10 most
frequent ones.
top_brands = data['brand_name'].value_counts().head(10)

# Set the figure size for the bar chart.
plt.figure(figsize=(10, 6))

# Generate a bar plot using Seaborn, with brand counts on the x-
axis and brand names on the y-axis.
# The 'viridis' palette is used for coloring the bars.
sns.barplot(x=top_brands.values, y=top_brands.index,
palette='viridis')

# Label the x and y axes.
plt.xlabel('Count')
plt.ylabel('Brand Name')

# Set the title for the chart.
plt.title('Top 10 Brands')

# Display the chart.
plt.show()
```

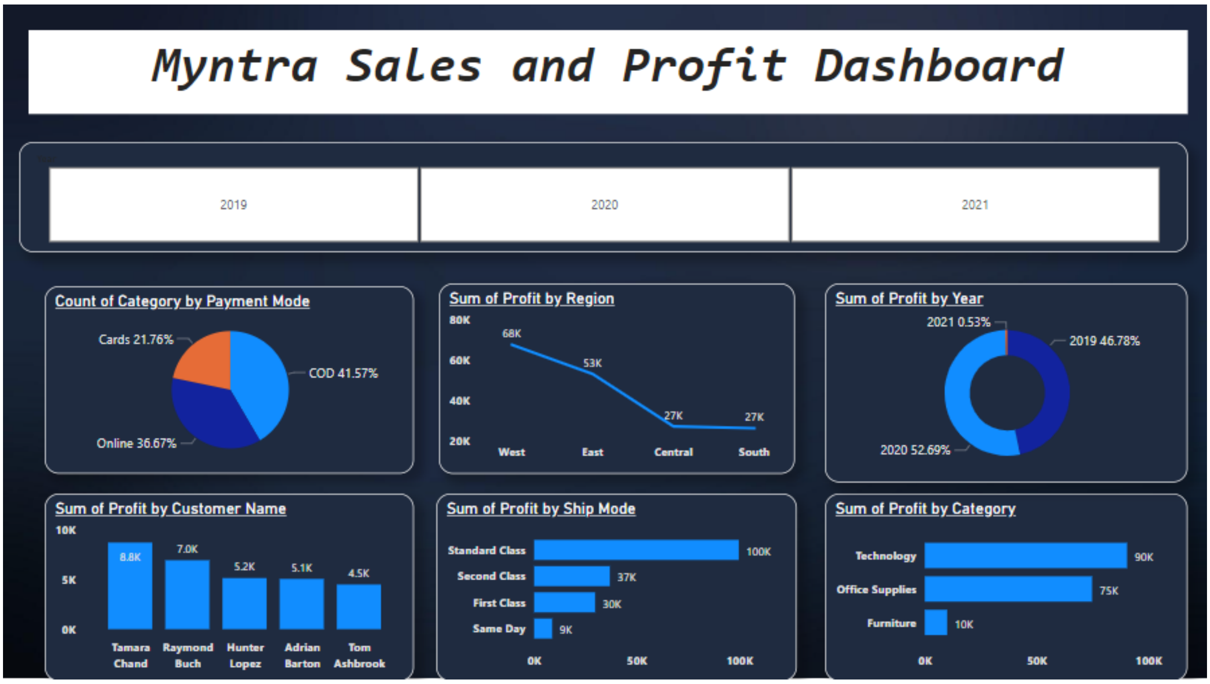


```
# A correlation matrix provides insights into the relationships
between numerical variables in the dataset.

# First, calculate the correlation matrix for the dataset 'data'.
correlation_matrix = data.corr(numeric_only=True)

# Next, create a plot for the correlation matrix using Seaborn and
Matplotlib.
# Set the figure size to 10x8 inches.
plt.figure(figsize=(10, 8))

# Create a heatmap of the correlation matrix with annotations.
# Use the 'coolwarm' color map to represent correlations, format
values with two decimal places,
# and add small gaps between cells.
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm',
fmt='.2f', linewidths=0.5)

# Set the title for the plot.
plt.title('Correlation Matrix')

# Finally, display the plot.
plt.show()
```



	rating	rating_count	marked_price	discounted_price	discount_amount	discount_percent	discounted_percent
rating	1.00	0.11	-0.10	-0.18	0.06	0.25	0.25
rating_count	0.11	1.00	-0.04	-0.04	-0.02	-0.01	-0.01
marked_price	-0.10	-0.04	1.00	0.85	0.70	0.12	0.12
discounted_price	-0.18	-0.04	0.85	1.00	0.21	-0.27	-0.27
discount_amount	0.06	-0.02	0.70	0.21	1.00	0.59	0.59
discount_percent	0.25	-0.01	0.12	-0.27	0.59	1.00	1.00
discounted_percent	0.25	-0.01	0.12	-0.27	0.59	1.00	1.00

Screen Shot of Power BI



5. PROPOSED ENHANCEMENTS

Here are some proposed enhancements for the Myntra sales data analysis and visualization system:

- Real-time data analysis and visualization: The system could be enhanced to provide real-time data analysis and visualization. This would allow users to see the latest trends and patterns in the data as they are happening.
- Predictive analytics: The system could be enhanced to include predictive analytics capabilities. This would allow users to predict future trends and patterns in the data.
- Natural language processing (NLP): The system could be enhanced to include NLP capabilities. This would allow users to interact with the system using natural language, such as asking questions and receiving answers in plain English.
- Integration with other systems: The system could be integrated with other systems, such as CRM systems and marketing automation systems. This would allow users to use the insights from the data to improve their business processes.

Here are some specific examples of how these enhancements could be implemented:

- Real-time data analysis and visualization: The system could use a streaming data processing platform to process the Myntra sales data in real time. The system could then use a visualization library, such as D3.js or Plotly.js, to create real-time visualizations of the data.
- Predictive analytics: The system could use a machine learning library, such as TensorFlow or scikit-learn, to train predictive models on the Myntra sales data. The trained models could then be used to predict future trends and patterns in the data.
- Integration with other systems: The system could use APIs to integrate with other systems, such as CRM systems and marketing automation systems. This would allow users to push the insights from the data to other systems and use them to improve their business processes.

I hope these suggestions are helpful.

6. CONCLUSION

The proposed Myntra sales data analysis and visualization system is a viable solution to the problem of manual and time-consuming Myntra sales data analysis and visualization. The system has the potential to improve the efficiency and effectiveness of the data analysis and visualization process, and to provide users with more insights into the Myntra sales data.

7. BIBLIOGRAPHY

Online references:

https://www.kaggle.com
https://chat.openai.com
https://bard.google.com
https://pandas.pydata.org
https://seaborn.pydata.org
https://scikit-learn.org
https://matplotlib.org