

```

1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import matplotlib.gridspec as gridspec
5 import seaborn as sns
6 import plotly.express as px

1 dataset = pd.read_excel("../Data/Womens Clothing Reviews Data New.xlsx")
2 dataset

```

	Product_ID	Category	SubCategory1	SubCategory2	Location	Channel	Customer_Age	Rating	Recommend_Flag	Merged_Review
0	767	Initmates	Intimate	Intimates	Mumbai	Mobile	33	4	1	Absolutely wonderful - silky and sexy and com...
1	1080	General	Dresses	Dresses	Bangalore	Mobile	34	5	1	Love this dress! it's sooo pretty. i happen...
2	1077	General	Dresses	Dresses	Gurgaon	Mobile	60	3	0	Some major design flaws I had such high hopes ...
3	1049	General Petite	Bottoms	Pants	Chennai	Web	50	5	1	My favorite buy! I love, love, love this jumps...
4	847	General	Tops	Blouses	Bangalore	Web	47	5	1	Flattering shirt This shirt is very flattering...
...
22637	1104	General Petite	Dresses	Dresses	Gurgaon	Mobile	34	5	1	Great dress for many occasions I was very happ...

```
1 dataset.columns
```

```

Index(['Product_ID', 'Category', 'SubCategory1', 'SubCategory2', 'Location',
      'Channel', 'Customer_Age', 'Rating', 'Recommend_Flag', 'Merged_Review'],
      dtype='object')

```

```

1 import plotly.graph_objects as go
2 from plotly.subplots import make_subplots
3 import plotly.express as px
4
5 # Create a 4x2 grid of subplots with larger size
6 fig = make_subplots(rows=4, cols=2, subplot_titles=(
7     'Distribution of Products across Categories',
8     'Distribution of Products across Subcategory1',
9     'Distribution of Products across SubCategory2',
10    'Distribution of Products across Locations',
11    'Distribution of Products across Channels',
12    'Distribution of Customer Age',
13    'Distribution of Rating',
14    'Distribution of Recommend_Flag'),
15    vertical_spacing=0.2) # Adjust the vertical spacing as needed
16
17 # Plot the first graph - Distribution of Products across Categories
18 fig.add_trace(go.Bar(x=dataset['Category'].value_counts().index, y=dataset['Category'].value_counts().values),
19              row=1, col=1)
20 fig.update_xaxes(title_text='Category', tickangle=45, row=1, col=1)
21 fig.update_yaxes(title_text='Count', row=1, col=1)
22
23 # Plot the second graph - Distribution of Products across Subcategory1
24 fig.add_trace(go.Bar(x=dataset['SubCategory1'].value_counts().index, y=dataset['SubCategory1'].value_counts().values),
25              row=1, col=2)
26 fig.update_xaxes(title_text='SubCategory1', tickangle=45, row=1, col=2)
27 fig.update_yaxes(title_text='Count', row=1, col=2)
28
29 # Plot the third graph - Distribution of Products across SubCategory2
30 fig.add_trace(go.Bar(x=dataset['SubCategory2'].value_counts().index, y=dataset['SubCategory2'].value_counts().values),
31              row=2, col=1)
32 fig.update_xaxes(title_text='SubCategory2', tickangle=45, row=2, col=1)
33 fig.update_yaxes(title_text='Count', row=2, col=1)
34
35 # Plot the fourth graph - Distribution of Products across Locations

```

```

36 fig.add_trace(go.Bar(x=dataset['Location'].value_counts().index, y=dataset['Location'].value_counts().values),
37                 row=2, col=2)
38 fig.update_xaxes(title_text='Location', tickangle=45, row=2, col=2)
39 fig.update_yaxes(title_text='Count', row=2, col=2)
40
41 # Plot the fifth graph - Distribution of Products across Channels
42 fig.add_trace(go.Bar(x=dataset['Channel'].value_counts().index, y=dataset['Channel'].value_counts().values),
43                 row=3, col=1)
44 fig.update_xaxes(title_text='Channel', tickangle=45, row=3, col=1)
45 fig.update_yaxes(title_text='Count', row=3, col=1)
46
47 # Plot the sixth graph - Distribution of Customer Age
48 fig.add_trace(go.Histogram(x=dataset['Customer_Age'], nbinsx=20), row=3, col=2)
49 fig.update_xaxes(title_text='Customer Age', row=3, col=2)
50 fig.update_yaxes(title_text='Count', row=3, col=2)
51
52 # Plot the seventh graph - Distribution of Rating
53 fig.add_trace(go.Bar(x=dataset['Rating'].value_counts().index, y=dataset['Rating'].value_counts().values),
54                 row=4, col=1)
55 fig.update_xaxes(title_text='Rating', row=4, col=1)
56 fig.update_yaxes(title_text='Count', row=4, col=1)
57
58 # Plot the eighth graph - Distribution of Recommend_Flag
59 fig.add_trace(go.Bar(x=dataset['Recommend_Flag'].value_counts().index, y=dataset['Recommend_Flag'].value_counts().values),
60                 row=4, col=2)
61 fig.update_xaxes(title_text='Recommend_Flag', row=4, col=2)
62 fig.update_yaxes(title_text='Count', row=4, col=2)
63
64 # Update subplot layout
65 fig.update_layout(title='Exploratory Data Analysis (EDA)', height=1200,
66                 showlegend=False) # Set height as needed
67 fig.show()

```

```

1
2 # Plot the first countplot - Rating vs. Category
3 fig1 = px.histogram(dataset, x='Rating', color='Category', barmode='group',
4                     title='Rating vs. Category', labels={'Rating': 'Rating', 'Category': 'Category'})
5 fig1.show()
6
7 # Plot the second countplot - Rating vs. SubCategory1
8 fig2 = px.histogram(dataset, x='Rating', color='SubCategory1', barmode='group',
9                     title='Rating vs. SubCategory1', labels={'Rating': 'Rating', 'SubCategory1': 'SubCategory1'})
10 fig2.show()
11
12 # Plot the third countplot - Rating vs. SubCategory2
13 fig3 = px.histogram(dataset, x='Rating', color='SubCategory2', barmode='group',
14                     title='Rating vs. SubCategory2', labels={'Rating': 'Rating', 'SubCategory2': 'SubCategory2'})
15 fig3.show()
16
17 # Plot the fourth countplot - Rating vs. Location
18 fig4 = px.histogram(dataset, x='Rating', color='Location', barmode='group',
19                     title='Rating vs. Location', labels={'Rating': 'Rating', 'Location': 'Location'})
20 fig4.show()
21
22 # Plot the fifth countplot - Rating vs. Channel
23 fig5 = px.histogram(dataset, x='Rating', color='Channel', barmode='group',
24                     title='Rating vs. Channel', labels={'Rating': 'Rating', 'Channel': 'Channel'})
25 fig5.show()
26

```

```

1
2 # Plot the first countplot - Rating vs. Category
3 fig1 = px.histogram(dataset, x='Recommend_Flag', color='Category', barmode='group',
4                     title='Recommend_Flag vs. Category', labels={'Recommend_Flag': 'Recommend Flag', 'Category': 'Category'})
5 fig1.show()
6
7 # Plot the second countplot - Rating vs. SubCategory1
8 fig2 = px.histogram(dataset, x='Recommend_Flag', color='SubCategory1', barmode='group',
9                     title='Recommend_Flag vs. SubCategory1', labels={'Recommend_Flag': 'Recommend Flag', 'SubCategory1': 'SubCategory1'})
10 fig2.show()
11
12 # Plot the third countplot - Rating vs. SubCategory2

```

```

13 fig3 = px.histogram(dataset, x='Recommend_Flag', color='SubCategory2', barmode='group',
14                       title='Recommend_Flag vs. SubCategory2', labels={'Recommend_Flag': 'Recommend Flag', 'SubCategory2': 'SubCategory2'})
15 fig3.show()
16
17 # Plot the fourth countplot - Rating vs. Location
18 fig4 = px.histogram(dataset, x='Recommend_Flag', color='Location', barmode='group',
19                       title='Recommend_Flag vs. Location', labels={'Recommend_Flag': 'Recommend Flag', 'Location': 'Location'})
20 fig4.show()
21
22 # Plot the fifth countplot - Rating vs. Channel
23 fig5 = px.histogram(dataset, x='Recommend_Flag', color='Channel', barmode='group',
24                       title='Recommend_Flag vs. Channel', labels={'Recommend_Flag': 'Recommend Flag', 'Channel': 'Channel'})
25 fig5.show()
26

```

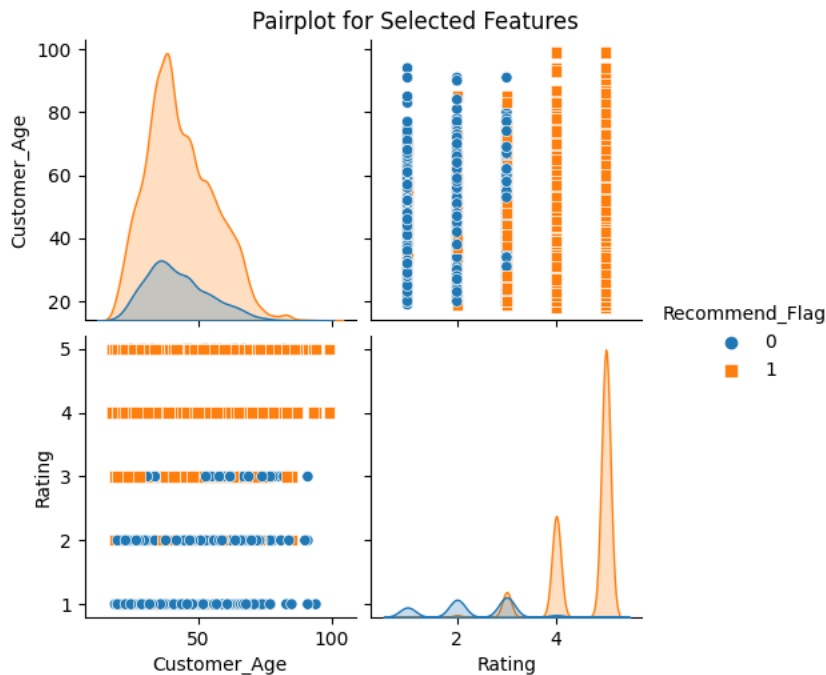
```

1 # Multivariate Analysis (example: Pairplot)
2 # Select a subset of features for multivariate analysis
3 selected_features = ['Customer_Age', 'Rating']
4
5 # Add the target variable to the selected features for better insights
6 selected_features.append('Recommend_Flag')
7
8 # Pairplot
9 sns.pairplot(data=dataset[selected_features], hue='Recommend_Flag', diag_kind='kde', markers=["o", "s"])
10 plt.suptitle("Pairplot for Selected Features", y=1.02)
11 plt.show()

```

c:\Users\pholl\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\axisgrid.py:118: UserWarning

The figure layout has changed to tight



```

1 # Scatter plot for 'Customer_Age' vs. 'Rating' using Plotly Express
2 fig = px.scatter(dataset, x='Customer_Age', y='Rating', color='Recommend_Flag',
3                  color_continuous_scale='viridis', title='Scatter plot: Customer Age vs. Rating')
4 fig.update_layout(xaxis_title='Customer Age', yaxis_title='Rating')
5 fig.show()

```

```

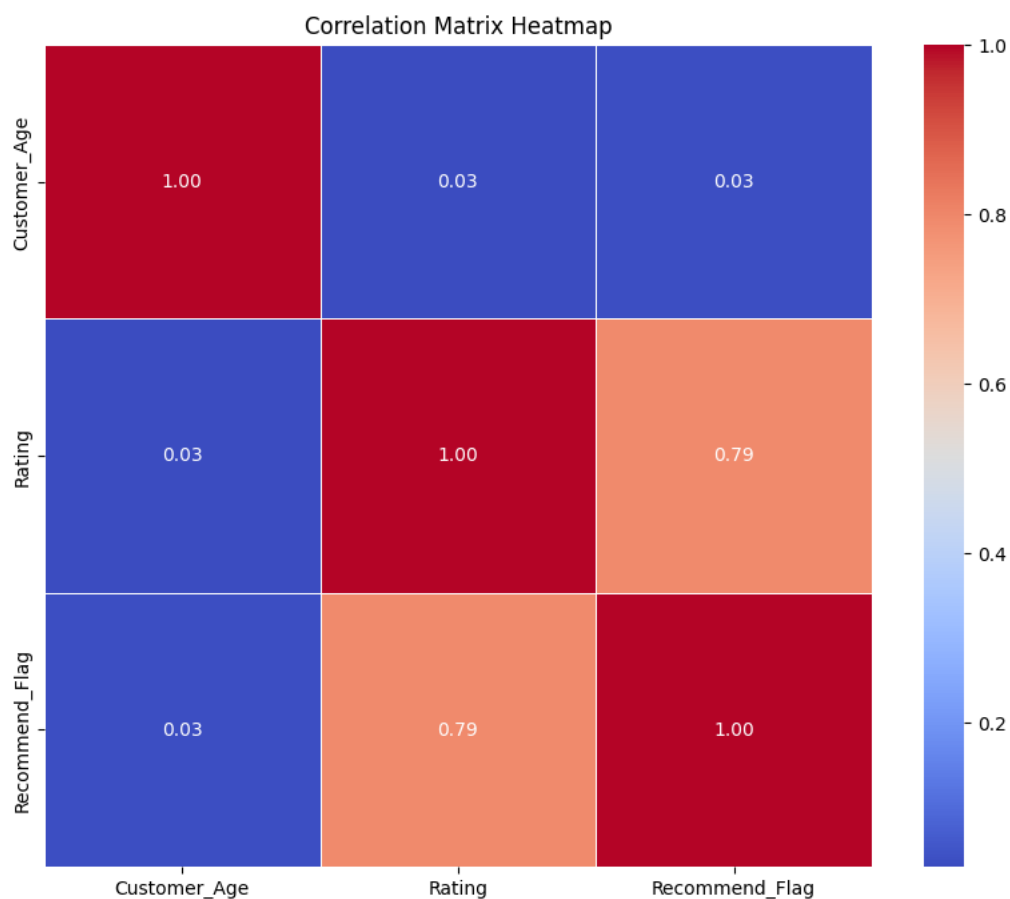
1 # Assuming 'ProductID' is the non-numeric column that needs to be excluded
2 numeric_df = dataset.drop(columns=['Product_ID'])
3
4 # Step 1: Calculate the correlation matrix
5 correlation_matrix = numeric_df.corr()
6
7 # Step 2: Create the heatmap

```

```
8 plt.figure(figsize=(10, 8))
9 sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)
10 plt.title('Correlation Matrix Heatmap')
11 plt.show()
```

C:\Users\pholl\AppData\Local\Temp\ipykernel_13124\57608799.py:5: FutureWarning:

The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default



```
1 dataset.columns
```

```
Index(['Product_ID', 'Category', 'SubCategory1', 'SubCategory2', 'Location',  
      'Channel', 'Customer_Age', 'Rating', 'Recommend_Flag', 'Merged_Review'],  
      dtype='object')
```