

1 #a. Performing exploratory analysis on the data to understand the patterns

```
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
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
```

	Product ID	Category	Subcategory1	SubCategory2	Location	Channel	Customer Age	Review Title	Review Text
0	767	Initmates	Intimate	Intimates	Mumbai	Mobile	33	NaN	Absolutely wonderful - silky and sexy and comfy...
1	1080	General	Dresses	Dresses	Bangalore	Mobile	34	NaN	Love this dress! it's sooo pretty. i happene...
2	1077	General	Dresses	Dresses	Gurgaon	Mobile	60	Some major design flaws	I had such high hopes for this dress and reall...
3	1049	General Petite	Bottoms	Pants	Chennai	Web	50	My favorite buy!	I love, love, love this jumpsuit. it's fun, fl...
4	847	General	Tops	Blouses	Bangalore	Web	47	Flattering shirt	This shirt is very flattering to all due to th...
...

1 dataset.shape

(23486, 11)

1 dataset.head()

	Product ID	Category	Subcategory1	SubCategory2	Location	Channel	Customer Age	Review Title	Review Text	Rati
0	767	Initmates	Intimate	Intimates	Mumbai	Mobile	33	NaN	Absolutely wonderful - silky and sexy and comfy...	
4	1080	General	Dresses	Dresses	Bangalore	Mobile	34	NaN	Love this dress! it's sooo pretty. i happene...	

```
1 #Make columns names in proper format by assigning proper variables
2 dataset.columns=['Product_ID', 'Category', 'SubCategory1', 'SubCategory2', 'Location',
3                 'Channel', 'Customer_Age', 'Review_Title', 'Review_Text', 'Rating',
4                 'Recommend_Flag']
```

1 dataset.head()

Product_ID	Category	SubCategory1	SubCategory2	Location	Channel	Customer_Age	Review_Title	Rev
0	767	Intimates	Intimate	Intimates	Mumbai	Mobile	33	NaN

```
1 #Check columns name
2 dataset.columns

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

```
1 # Find duplicate rows
2 duplicated_rows = dataset.duplicated()
3
4 # Count the number of duplicate rows
5 number_of_duplicate_rows = duplicated_rows.sum()
6
7 print("Number of duplicated rows:", number_of_duplicate_rows)
```

Number of duplicated rows: 3

```
1 #Remove duplicate data
2 dataset=dataset.drop_duplicates()
```

```
1 dataset.shape

(23483, 11)
```

```
1 # Validate with duplicate rows again
2 duplicated_rows = dataset.duplicated()
3
4 # Count the number of duplicate rows
5 number_of_duplicate_rows = duplicated_rows.sum()
6
7 print("Number of duplicated rows:", number_of_duplicate_rows)
```

Number of duplicated rows: 0

```
1 #Check null values in dataset
2 dataset.isnull().sum()
```

```
Product_ID      0
Category        14
SubCategory1    14
SubCategory2    14
Location         0
Channel         0
Customer_Age    0
Review_Title    3807
Review_Text     842
Rating          0
Recommend_Flag  0
dtype: int64
```

```
1 rows_missing_category = dataset[dataset['Category'].isnull()]
2 # Display rows with missing 'Category' as a DataFrame
3 rows_missing_category
```

	Product_ID	Category	SubCategory1	SubCategory2	Location	Channel	Customer_Age	Review_Title	
	9444	72	NaN	NaN	NaN	Chennai	Web	25	My favorite socks!!!!
	13767	492	NaN	NaN	NaN	Gurgaon	Web	23	So soft!
	13768	492	NaN	NaN	NaN	Mumbai	Web	49	Wardrobe staple

```
1 dataset['Category'].fillna(dataset['Category'].mode()[0], inplace=True)
2 dataset['SubCategory1'].fillna(dataset['SubCategory1'].mode()[0], inplace=True)
3 dataset['SubCategory2'].fillna(dataset['SubCategory2'].mode()[0], inplace=True)
```

C:\Users\pholl\AppData\Local\Temp\ipykernel_8740\757871235.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
dataset['Category'].fillna(dataset['Category'].mode()[0], inplace=True)

C:\Users\pholl\AppData\Local\Temp\ipykernel_8740\757871235.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
dataset['SubCategory1'].fillna(dataset['SubCategory1'].mode()[0], inplace=True)

C:\Users\pholl\AppData\Local\Temp\ipykernel_8740\757871235.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
dataset['SubCategory2'].fillna(dataset['SubCategory2'].mode()[0], inplace=True)

```
1 dataset.loc[dataset['Product_ID']==492]
```

	Product_ID	Category	SubCategory1	SubCategory2	Location	Channel	Customer_Age	Review_Title
13767	492	General	Tops	Dresses	Gurgaon	Web	23	So soft!

```
1 dataset.isnull().sum()
```

```
Product_ID      0
Category        0
SubCategory1    0
SubCategory2    0
Location        0
Channel         0
Customer_Age    0
Review_Title    3807
Review_Text     842
Rating          0
Recommend_Flag  0
dtype: int64
```

```
1 rows_missing_Review_Text = dataset[dataset['Review_Text'].isnull()]
2 # Display rows with missing 'Review_Text ' as a DataFrame
3 rows_missing_Review_Text
```

	Product_ID	Category	SubCategory1	SubCategory2	Location	Channel	Customer_Age	Review_Title
92	861	General Petite	Tops	Knits	Gurgaon	Mobile	23	NaN
93	1081	General	Dresses	Dresses	Gurgaon	Mobile	31	NaN
98	1133	General	Jackets	Outerwear	Mumbai	Mobile	50	NaN
135	861	General Petite	Tops	Knits	Gurgaon	Web	35	NaN

```
1 # remove rows where Review_Title and Review_Text have NaN values
2 dataset.dropna(subset=['Review_Title', 'Review_Text'], how='all', inplace=True)
```

C:\Users\pholl\AppData\Local\Temp\ipykernel_8740\566270713.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
dataset.dropna(subset=['Review_Title', 'Review_Text'], how='all', inplace=True)

```
1 dataset.isnull().sum()
```

```
Product_ID      0
Category        0
SubCategory1    0
SubCategory2    0
Location        0
Channel         0
Customer_Age    0
Review_Title    2966
Review_Text     1
Rating          0
Recommend_Flag  0
dtype: int64
```

```
1 rows_missing_Review_Text = dataset[dataset['Review_Text'].isnull()]
2 # Display rows with missing 'Review_Text ' as a DataFrame
3 rows_missing_Review_Text
```

	Product_ID	Category	SubCategory1	SubCategory2	Location	Channel	Customer_Age	Review_Title
10220	1096	General	Dresses	Dresses	Chennai	Mobile	30	Such a beautiful

```
1 rows_missing_Review_Title = dataset[dataset['Review_Title'].isnull()]
2 # Display rows with missing 'Review_Text ' as a DataFrame
3 rows_missing_Review_Title
```

	Product_ID	Category	SubCategory1	SubCategory2	Location	Channel	Customer_Age	Review_Title
0	767	Intimates	Intimate	Intimates	Mumbai	Mobile	33	NaN
1	1080	General	Dresses	Dresses	Bangalore	Mobile	34	NaN
11	1095	General Petite	Dresses	Dresses	Mumbai	Mobile	39	NaN
30	1060	General Petite	Bottoms	Pants	Mumbai	Web	33	NaN

```
1 import pandas as pd
2
3 # Replace "NaN" values in both columns with empty strings
4 dataset['Review_Text'].fillna('', inplace=True)
5 dataset['Review_Title'].fillna('', inplace=True)
6
```

```

7 # Merge the columns and store the result in a new column called "Merged_Review"
8 dataset['Merged_Review'] = dataset['Review_Title'] + ' ' + dataset['Review_Text']
9
10 # Drop the individual "Review_Text" and "Review_Title" columns if needed
11 dataset.drop(['Review_Text', 'Review_Title'], axis=1, inplace=True)

C:\Users\pholl\AppData\Local\Temp\ipykernel_8740\984468918.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy
dataset['Review_Text'].fillna('', inplace=True)
C:\Users\pholl\AppData\Local\Temp\ipykernel_8740\984468918.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy
dataset['Review_Title'].fillna('', inplace=True)
C:\Users\pholl\AppData\Local\Temp\ipykernel_8740\984468918.py:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy
dataset['Merged_Review'] = dataset['Review_Title'] + ' ' + dataset['Review_Text']
C:\Users\pholl\AppData\Local\Temp\ipykernel_8740\984468918.py:11: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy
dataset.drop(['Review_Text', 'Review_Title'], axis=1, inplace=True)

```

```
1 dataset.isnull().sum()
```

```

Product_ID      0
Category        0
SubCategory1    0
SubCategory2    0
Location        0
Channel         0
Customer_Age    0
Rating          0
Recommend_Flag  0
Merged_Review   0
dtype: int64

```

```
1 dataset.loc[dataset['Product_ID']==767]
```

	Product_ID	Category	SubCategory1	SubCategory2	Location	Channel	Customer_Age	Rating	Recommend
0	767	Initmates	Intimate	Intimates	Mumbai	Mobile	33	4	

```
1 dataset.shape
```

```
(22642, 10)
```

```
1 #Check info
```

```
2 dataset.info()
```

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 22642 entries, 0 to 23485
Data columns (total 10 columns):
 #   Column                Non-Null Count  Dtype
---  -
 0   Product_ID            22642 non-null  int64
 1   Category              22642 non-null  object
 2   SubCategory1          22642 non-null  object
 3   SubCategory2          22642 non-null  object
 4   Location              22642 non-null  object
 5   Channel               22642 non-null  object
 6   Customer_Age          22642 non-null  int64
 7   Rating               22642 non-null  int64
 8   Recommend_Flag        22642 non-null  int64
 9   Merged_Review         22642 non-null  object
dtypes: int64(4), object(6)
memory usage: 1.9+ MB

```

```
1 dataset.describe()
```

	Product_ID	Customer_Age	Rating	Recommend_Flag
count	22642.000000	22642.000000	22642.000000	22642.000000
mean	919.340164	43.279790	4.183597	0.818876
std	202.265815	12.327023	1.115751	0.385129
min	1.000000	18.000000	1.000000	0.000000
25%	861.000000	34.000000	4.000000	1.000000
50%	936.000000	41.000000	5.000000	1.000000
75%	1078.000000	52.000000	5.000000	1.000000
max	1205.000000	99.000000	5.000000	1.000000

```
1 dataset.columns
```

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

```
1 # Value counts for 'Category'
2 category_counts = dataset['Category'].value_counts()
3 print("Value Counts for Category:")
4 print(category_counts)
```

```
Value Counts for Category:
General      13379
General Petite  7837
Initmates    1426
Name: Category, dtype: int64
```

```
1 # Value counts for 'SubCategory1'
2 subcategory1_counts = dataset['SubCategory1'].value_counts()
3 print("Value Counts for SubCategory1:")
4 print(subcategory1_counts)
```

```
Value Counts for SubCategory1:
Tops      10061
Dresses   6146
Bottoms   3662
Intimate  1653
Jackets   1002
Trend     118
Name: SubCategory1, dtype: int64
```

```
1 # Value counts for 'SubCategory2'
2 subcategory2_counts = dataset['SubCategory2'].value_counts()
3 print("Value Counts for SubCategory2:")
4 print(subcategory2_counts)
```

```
Value Counts for SubCategory2:
Dresses      6159
Knits        4626
Blouses      2983
Sweaters     1380
Pants        1350
Jeans        1104
Fine gauge   1059
Skirts       903
Jackets      683
Lounge       669
Swim         332
Outerwear    319
Shorts       304
Sleep        214
Legwear      158
Intimates    147
Layering     132
Trend        118
Casual bottoms  1
Chemises     1
Name: SubCategory2, dtype: int64
```

```
1 # Value counts for 'Location'
2 Location_counts = dataset['Location'].value_counts()
```

```
3 print("Value Counts for Location:")
4 print(Location_counts)
```

```
Value Counts for Location:
Gurgaon      8491
Mumbai       6859
Bangalore    5050
Chennai      2242
Name: Location, dtype: int64
```

```
1 # Value counts for 'Channel'
2 Channel_counts = dataset['Channel'].value_counts()
3 print("Value Counts for Channel:")
4 print(Channel_counts)
```

```
Value Counts for Channel:
Web          13096
Mobile       9546
Name: Channel, dtype: int64
```

```
1 # Value counts for 'Customer_Age'
2 Customer_Age_counts = dataset['Customer_Age'].value_counts()
3 print("Value Counts for Customer_Age:")
4 print(Customer_Age_counts)
```

```
Value Counts for Customer_Age:
39    1226
35     851
36     801
34     766
38     751
...
93         2
90         2
86         2
99         2
92         1
Name: Customer_Age, Length: 77, dtype: int64
```

```
1 # Value counts for 'Recommend_Flag'
2 Recommend_Flag_counts = dataset['Recommend_Flag'].value_counts()
3 print("Value Counts for Recommend_Flag:")
4 print(Recommend_Flag_counts)
```

```
Value Counts for Recommend_Flag:
1    18541
0    4101
Name: Recommend_Flag, dtype: int64
```

```
1 # Value counts for 'Rating'
2 Rating_counts = dataset['Rating'].value_counts()
3 print("Value Counts for Rating:")
4 print(Rating_counts)
```

```
Value Counts for Rating:
5    12541
4     4908
3     2823
2     1549
1       821
Name: Rating, dtype: int64
```

```
1 #Check the correlation between numerical features.
2 dataset.corr()
```

```
C:\Users\pholl\AppData\Local\Temp\ipykernel_8740\3955366760.py:2: FutureWarning: The def
dataset.corr()
```

	Product_ID	Customer_Age	Rating	Recommend_Flag
Product_ID	1.000000	0.017646	-0.018425	-0.014855
Customer_Age	0.017646	1.000000	0.029926	0.034184
Rating	-0.018425	0.029926	1.000000	0.792570
Recommend_Flag	-0.014855	0.034184	0.792570	1.000000

```
1 # Save the cleaned dataset to a new CSV file
2 dataset.to_excel('Womens Clothing Reviews Data New.xlsx', index=False)
```

