Data Analysis

The **idea behind the data analysis part** to obtain the '**Best Sellers**' from the preprocessed dataset is as follows:

-

1. Consider the relevant features that are useful for analyzing the data. Sort and organize these features based on convenience and their significance in determining the 'Best Sellers'.
2. Sort each individual feature in the dataset and create separate dataframes for each of them. This allows for a focused analysis of each feature.
3. Concatenate all the individual feature dataframes, bringing them together into a consolidated dataframe. This helps identify the common sellers that appear across multiple features, providing a more comprehensive view.
4. Apply filtering techniques to extract the top rows that are most relevant in determining the best sellers. This step helps isolate the key sellers based on predetermined criteria or rankings.
5. Generate insights from the derived filtered dataset and understand the attributes to be a "**Best Seller**".

By following these steps, I will be analyze effectivey the preprocessed dataset and extract the best sellers based on various feature aspects, contributing to a more informed decision-making process.

```
#Importing necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# Reading the Excel file
data_for_analysis = pd.read_excel('/content/Data_for_Analysis.xlsx')
```

```
#Displaying the first 10 samples of the Analysis Data
data_for_analysis.head(10)
```

	sellerlink	sellerproductcount	positive_percent	seller_ratings	Email	Telephone	businessname
0	Seller 1	100000	88	118	jadgemaello@gmail.com	NaN	Lohas Living Inc
1	Seller 2	898	90	566	info@1a-handelsagentur.de	3.993283e+10	1a-Handelsagentur
2	Seller 6	28	94	82	NaN	NaN	NaN
3	Seller 15	919	97	116	webmaster@9pm-store.de	1.514001e+10	Oliver Mills
4	Seller 20	898	76	3721	abc-markt@web.de	5.705912e+08	ABC-Schnäppchenmarkt GmbH
5	Seller 21	123	94	82	NaN	NaN	All Terrain Ireland
6	Seller 22	898	92	181	support@bauguru.at	NaN	GURU 2016 GmbH
7	Seller 23	40000	100	10	NaN	NaN	MADEWELL SUPPLY INC
8	Seller 24	100000	88	3134	bib.beckum@blumenbecker.com	2.521841e+08	Blumenbecker Industriebedarf GmbH
9	Seller 25	3000	97	610	NaN	NaN	Jörg Müller & Petra Scheerer GbR

```
#Shape of the Data for Analysis
data_for_analysis.shape
```

```
(602, 14)
```

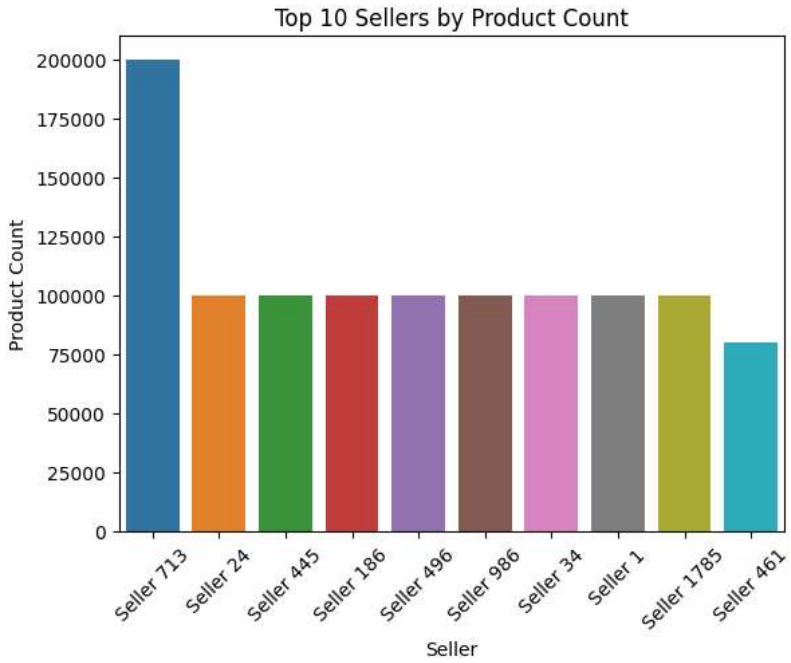
1. Sorting sellers based upon **Product Count**

```
#Sorting the 'sellerproductcount' in Descending order(High to Low) and storing a 'product_count_sort' variable
product_count_sort = data_for_analysis.sort_values('sellerproductcount', ascending=False)
product_count_sort
```

	sellerlink	sellerproductcount	positive_percent	seller_ratings	Email	Telephone	
347	Seller 713	200000	76	2251	NaN	NaN	T
8	Seller 24	100000	88	3134	bib.beckum@blumenbecker.com	2.521841e+08	
209	Seller 445	100000	100	7	NaN	NaN	
111	Seller 186	100000	86	29	NaN	NaN	
228	Seller 496	100000	91	288	NaN	NaN	(
...	...	...	...	...	...	...	
453	Seller 881	9	100	6	NaN	NaN	(ha
400	Seller 798	8	76	17	NaN	NaN	V
513	Seller 999	5	79	14	Esloheinfo@weihnachtsbaumland.deUmsatzsteuer-I...	NaN	We
155	Seller 261	4	100	19	info@natinos.de	1.511495e+10	
127	Seller 216	3	75	12	NaN	NaN	

602 rows × 14 columns

```
# Plotting the bar chart of 'sellerproductcount' for first 10 values using seaborn
sns.barplot(x='sellerlink', y='sellerproductcount', data=product_count_sort[:10])
plt.xlabel('Seller')
plt.ylabel('Product Count')
plt.title('Top 10 Sellers by Product Count')
plt.xticks(rotation=45)
plt.show()
```



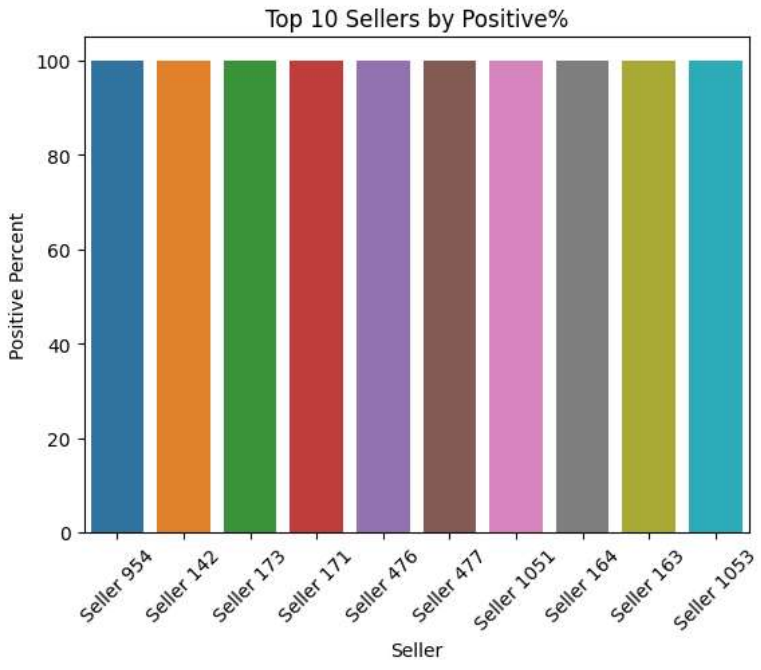
2. Sorting sellers based upon **positive_percent**

```
#Sorting the 'positive_percent' in Descending order(High to Low) and storing a 'positive_percent_sort' variable

positive_percent_sort = data_for_analysis.sort_values('positive_percent', ascending=False)
positive_percent_sort
```

	sellerlink	sellerproductcount	positive_percent	seller_ratings	Email	Telephone	businessname
489	Seller 954	4000	100	13	305574WARSTOREUK@SKY.COMVAT	NaN	NaN
76	Seller 142	443	100	2	office@haberl.at	2.749863e+07	Klaus Haberl GmbH
102	Seller 173	444	100	18	NaN	NaN	KOOK TIME PRODUCTS S.L.
101	Seller 171	479	100	121	info@kontur-pur.de	6.051889e+10	Peter Kaientz
218	Seller 476	69	100	16	NaN	NaN	Yuriy Lynnyk
...	...	...	...	...	...	...	...
576	Seller 1744	3000	0	1	NaN	NaN	NaN
451	Seller 879	898	0	1	NaN	NaN	JACOB STYLE SL
535	Seller 1065	1000	0	1	NaN	NaN	NaN
174	Seller 307	561	0	1	NaN	NaN	NaN
541	Seller 1104	5000	0	1	NaN	NaN	NaN

602 rows × 14 columns

```
# Plotting the bar chart of 'positive_percent' for first 10 values using seaborn
sns.barplot(x='sellerlink', y='positive_percent', data=positive_percent_sort[:10])
plt.xlabel('Seller')
plt.ylabel('Positive Percent')
plt.title('Top 10 Sellers by Positive%')
plt.xticks(rotation=45)
plt.show()
```



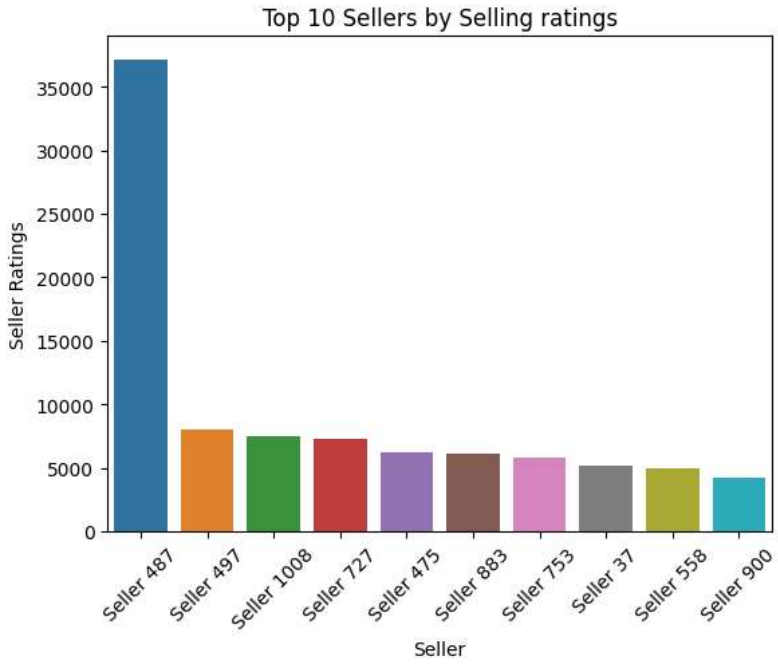
3. Sorting sellers based upon **seller_ratings**

```
#Sorting the 'seller_ratings' in Descending order(High to Low) and storing a 'seller_ratings_sort' variable

seller_ratings_sort = data_for_analysis.sort_values('seller_ratings', ascending=False)
seller_ratings_sort
```

	sellerlink	sellerproductcount	positive_percent	seller_ratings	Email	Telephone	businessname	busine
222	Seller 487	898	95	37124	NaN	NaN	PEARL. GmbH	
229	Seller 497	898	90	8063	info@shop-ar.de	4.121261e+09	AR E-Commerce Trading UG (haftungsbeschränkt)	
515	Seller 1008	60000	94	7465	NaN	NaN	Asia Pacific Elite Ltd	
359	Seller 727	40000	89	7242	support@trendmile.com	6.206000e+03	trendmile GmbH	
217	Seller 475	754	96	6204	info@lyra-pet.de	7.431603e+10	Lyra Pet GmbH	
...	...	...	...	...	...	...	...	
42	Seller 79	495	100	1	NaN	NaN	sarl jack bout	
451	Seller 879	898	0	1	NaN	NaN	JACOB STYLE SL	
514	Seller 1000	59	100	1	NaN	NaN	Alexander Kuhnigk	
541	Seller 1104	5000	0	1	NaN	NaN	NaN	
315	Seller 662	898	100	1	NaN	NaN	DCL SPEDITION LTD	

602 rows × 14 columns

```
# Plotting the bar chart of 'seller_ratings' of top 10 values using seaborn

sns.barplot(x='sellerlink', y='seller_ratings', data=seller_ratings_sort[:10])
plt.xlabel('Seller')
plt.ylabel('Seller Ratings')
plt.title('Top 10 Sellers by Selling ratings')
plt.xticks(rotation=45)
plt.show()
```



4. Sorting sellers based upon **Count of seller brands**

```
#Sorting the 'seller_ratings' in Descending order(High to Low) and storing a 'count_of_seller_sort' variable
count_of_seller_sort = data_for_analysis.sort_values('Count of seller brands', ascending=False)
count_of_seller_sort.head(6)
```

	sellerlink	sellerproductcount	positive_percent	seller_ratings	Email	Telephone	businessname	businessadd
0	Seller 1	100000	88	118	jadgemaello@gmail.com	NaN	Lohas Living Inc	
586	Seller 1773	710	97	140	info@zoo-bue-pet.de	6.206952e+09	NaN	
181	Seller 323	30000	92	699	info@limuno.com	4.959718e+12	Milovan Stojkovic	
183	Seller 331	368	95	19	shop@linofant.de	6.659987e+09	linofant GmbH	
187	Seller 364	3000	82	1438	service@livindo.de	4.900000e+01	mk Objektmöbel GmbH	
197	Seller 391	83	96	255	NaN	NaN	Zdzislaw Lorek	

```
data_for_analysis.columns
```

```
Index(['sellerlink', 'sellerproductcount', 'positive_percent',
       'seller_ratings', 'Email', 'Telephone', 'businessname',
       'businessaddress', 'Count of seller brands',
       'Max % of negative seller ratings - last 30 days',
       'Max % of negative seller ratings - last 90 days',
       'Max % of negative seller ratings - last 12 months',
       'Hero Product 1 #ratings', 'Hero Product 2 #ratings'],
      dtype='object')
```

5. Sorting sellers based upon **Max% of negative seller ratings**

```
#Sorting the 'seller_ratings' in Ascending order(Low to High) and storing a 'seller_ratings_sort' variable
max_negative_sort = data_for_analysis.sort_values(['Max % of negative seller ratings - last 30 days',
       'Max % of negative seller ratings - last 90 days',
       'Max % of negative seller ratings - last 12 months'], ascending=True)
max_negative_sort.head(6)
```

	sellerlink	sellerproductcount	positive_percent	seller_ratings	Email	Telephone	businessname	businessaddr
2	Seller 6	28	94	82	NaN	NaN	NaN	
5	Seller 21	123	94	82	NaN	NaN	All Terrain Ireland	
7	Seller 23	40000	100	10	NaN	NaN	MADEWELL SUPPLY INC	
17	Seller 35	42	90	10	amazon@hockdesign.de	2.262708e+10	HOCK Design e.K.	
25	Seller 56	60	87	15	NaN	NaN	KAMAI.PL GmbH	
27	Seller 58	21	100	1	NaN	NaN	NaN	

6. Sorting sellers based upon **Best Selling Product**(Hero Product 1 & Hero Product 2)

```
#Sorting the 'Hero Product 1 #ratings', 'Hero Product 2 #ratings'
#in Descending order(High to Low) and storing a 'hero_product_sort' variable
hero_product_sort = data_for_analysis.sort_values(['Hero Product 1 #ratings', 'Hero Product 2 #ratings'], ascending=False)
hero_product_sort
```

	sellerlink	sellerproductcount	positive_percent	seller_ratings	Email	Telephone	businessname	businessaddress
485	Seller 948	10000	92	159	NaN	NaN	VIP Plaza Japan Inc	JP
228	Seller 496	100000	91	288	NaN	NaN	C. Y. MARKETING INC.	US
181	Seller 323	30000	92	699	info@limuno.com	4.959718e+12	Milovan Stojkovic	DE
429	Seller 843	8000	97	38	NaN	NaN	Vertecchi S.r.l.	IT
455	Seller 883	4000	94	6086	service@vitamed-quinger.com	4.962369e+12	Vitamed Matthias Quinger e.K.	DE
...	...	...	...	...	...	...	...	...
546	Seller 1347	898	100	4	NaN	NaN	NaN	NaN
547	Seller 1366	898	94	82	NaN	NaN	NaN	NaN
570	Seller 1683	898	94	82	NaN	NaN	NaN	NaN
576	Seller 1744	3000	0	1	NaN	NaN	NaN	NaN
601	Seller 1829	898	94	82	NaN	NaN	NaN	NaN

602 rows × 14 columns

Concatenating all Dataframes

```
#Concating the All the derived dataframes upto top '300' to further analysis.

concatenated_df = pd.concat([product_count_sort[:300],
                             positive_percent_sort[:300],
                             seller_ratings_sort[:300],
                             count_of_seller_sort[:300],
                             max_negative_sort[:300],
                             hero_product_sort[:300]])
```

```
concatenated_df.head(3) #Displaying the Concatenated Dataframe
```

	sellerlink	sellerproductcount	positive_percent	seller_ratings	Email	Telephone	businessname	busin
347	Seller 713	200000	76	2251	NaN	NaN	TRADEINN RETAIL SERVICES SL	
8	Seller 24	100000	88	3134	bib.beckum@blumenbecker.com	252184060.0	Blumenbecker Industriebedarf GmbH	
209	Seller 445	100000	100	7	NaN	NaN	NaN	

```
#Value Counts of 'sellerlink' in Concatenated DataFrame
concatenated_df.sellerlink.value_counts()
```

```
Seller 833     6
Seller 541     6
Seller 1032    6
Seller 196     6
Seller 265     6
               ..
Seller 652     1
Seller 692     1
Seller 22      1
Seller 159     1
Seller 1404    1
Name: sellerlink, Length: 582, dtype: int64
```

```
#Computing Value Counts of 'sellerlink' in Concatenated DataFrame
value_counts = concatenated_df.sellerlink.value_counts()

#Filtering Values with Frequency 6 in Value Counts(Filtering the values that present in all the 6 tables)
filtered_values = value_counts[value_counts == 6]

#Length of the values that are present common in all 6 tables(concatenated as 'concatenated_df')
len(filtered_values)
```

```
19
```

```
#Converting Filtered Values to DataFrame and List
filtered_values_df = filtered_values.reset_index()
filtered_values_list = list(filtered_values_df.values)
filtered_values_list
```

```
[array(['Seller 833', 6], dtype=object),
 array(['Seller 541', 6], dtype=object),
 array(['Seller 1032', 6], dtype=object),
 array(['Seller 196', 6], dtype=object),
 array(['Seller 265', 6], dtype=object),
 array(['Seller 831', 6], dtype=object),
 array(['Seller 935', 6], dtype=object),
 array(['Seller 221', 6], dtype=object),
 array(['Seller 52', 6], dtype=object),
 array(['Seller 584', 6], dtype=object),
 array(['Seller 158', 6], dtype=object),
 array(['Seller 128', 6], dtype=object),
 array(['Seller 1774', 6], dtype=object),
 array(['Seller 153', 6], dtype=object),
 array(['Seller 437', 6], dtype=object),
 array(['Seller 15', 6], dtype=object),
 array(['Seller 863', 6], dtype=object),
 array(['Seller 705', 6], dtype=object),
 array(['Seller 1724', 6], dtype=object)]
```

```
#Appending the values of 'Best Sellers' into a List
seller_list = []

for seller,count in filtered_values_list:
  seller_list.append(seller)

print(seller_list)
```

```
['Seller 833', 'Seller 541', 'Seller 1032', 'Seller 196', 'Seller 265', 'Seller 831', 'Seller 935', 'Seller 221', 'Seller 52', 'Sel
```

Best Seller's List

```
#Filtering Best Sellers from Data for Analysis table
best_sellers = data_for_analysis[data_for_analysis['sellerlink'].isin(seller_list)]
```

```
#Converting 'Telephone' Column to Numeric Type
best_sellers['Telephone'] = pd.to_numeric(best_sellers['Telephone'], errors='coerce').astype(pd.Int64Dtype())
```

```
<ipython-input-23-3bb116c5cebc>:2: 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
  best_sellers['Telephone'] = pd.to_numeric(best_sellers['Telephone'], errors='coerce').astype(pd.Int64Dtype())
```

- The task to obtain the **'Best Sellers'** has been successfully completed, and we have obtained the final results.
- The filtered data for the best sellers is stored in the **'best_sellers'** DataFrame.

```
#Finally the 'Results' of the Task- To obtain the 'Best Sellers' has been completed.
best_sellers
```

	sellerlink	sellerproductcount	positive_percent	seller_ratings	Email	Telephone	businessname	bu
3	Seller 15	919	97	116	webmaster@9pm-store.de	15140008562	Oliver Mills	
22	Seller 52	1000	95	221	info@kaiser24.shop	20260937540	Georg Kaiser	
68	Seller 128	2000	98	256	info@kirchner24.de	938297910	Kirchner GmbH	
86	Seller 153	2000	99	156	mail@knobelbox.com	4084605742	Friedel und Wernecke UG	
90	Seller 158	2000	100	576	kontakt@kochmesser-shop.com	51197819650	Horn, Eckard	
118	Seller 196	20000	97	197	Service@KS-Germany.com	49201899240	KS Licht- u. Elektrotechnik GmbH	
131	Seller 221	1000	98	449	kvm-janke@web.de	3423755622	Großhandel Karin Janke	
157	Seller 265	3000	98	124	Info@LederMilz.deRegistergericht	7141902067	Michael Milz, Inhaber Irene Koser	
207	Seller 437	912	99	308	NaN	78529354105	Nelcilene Maia Costa	
246	Seller 541	921	97	117	TBcompany.de@gmail.com	57314969091	Tom Brinkmann-Clasberg	
272	Seller 584	952	98	344	eitner@te-importe.de	9723937160	Thomas Erfurth	
339	Seller 705	2000	98	135	mario.klunker@gmx.de	17697563029	TOYMATIC	
417	Seller 831	1000	97	105	info@fairmarkt24.de	71312798611	Demir Versandhandel	
419	Seller 833	5000	95	215	info@fliegende-pillen.de	8002402242	Birken-Apotheke	
440	Seller 863	6000	100	669	info@vinfiz.de	52419977601	Vinfiz GmbH	
479	Seller 935	1000	99	304	info@automatenwagner.de	7343919260	Wagner Automaten	
523	Seller 1032	1000	97	222	NaN	<NA>	Wiemann Lehrmittel	
571	Seller 1724	2000	97	910	zierfischtreff@aol.comUmsatzsteuer-Identifikat...	9834975133	Hubert Schmalzbauer	
587	Seller 1774	4000	96	173	info@zooheld.de	3475714957	Moritz-Consumer-Service	

```
#Printing the Business Names of the Best Sellers
print("These were the Business names of Best sellers:")
print(" ")
for name in best_sellers.businessname.values:
  print(name)
```

```
These were the Business names of Best sellers:

Oliver Mills
Georg Kaiser
Kirchner GmbH
Friedel und Wernecke UG
Horn, Eckard
KS Licht- u. Elektrotechnik GmbH
Großhandel Karin Janke
```

```
Michael Milz, Inhaber Irene Koser
Nelcilene Maia Costa
Tom Brinkmann-Clasberg
Thomas Erfurth
TOYMATIC
Demir Versandhandel
Birken-Apotheke
Vinfiz GmbH
Wagner Automaten
Wiemann Lehrmittel
Hubert Schmalzbauer
Moritz-Consumer-Service
```

Insights:

From the final table of '**best_sellers**' we can conclude that:

1. There are total **19 best sellers** based upon all the factors i.e, Product count, positive percentage- ratings, brand magnitude, less negative reviews and Top(Hero) products.
2. All of the Best sellers are from **Germany(DE)**.
3. These best sellers sale the products minimum of **900** to maximum of **20,000**.
4. Positive percent of the sellers are in between **95%-100%** which was a very good Percentage.
5. Count of the seller brands are in between **14(50% percentile),15(75% percentile)** and **16(Maximum)** i.e, maximum number of top sellers selling their maximum products to us.
6. Among these sellers, **none of them have had any complaints** in the past 30-90 days. This criterion has proven to be successful in selecting reliable sellers.
7. And in most of the cases(**15 in 19**), best sellers has a **least(1-4) negative reviews** complaints. And rest of the sellers(128,153,158 and 863) dont even having a single negative reviews from past one year.
8. These were the **Best or Most promising sellers in this dataset that the Acquisitions team at Razor should reach out to, and acquire.**
9. They may vary in count if we set the "Top" number to another.(For Eg: Top 200, Top 100, Top 400 etc.)
10. As there were a total of **19 best sellers**, which is a good number, and all of them are from the same country (Germany), it would be easier to establish stronger connections and increase revenue through enhanced collaboration.

```
best_sellers.to_excel("Best_Sellers.xlsx", index=False)
```

THE END

- Hope I have completed my task cleanly and precisely. Although some troubles have arisen, I am confident that they would not impact the final result in identifying the best sellers from the dataset.
- I have diligently followed a structured and step-by-step methodology to preprocess and analyze the data
- I hope to be shortlisted in this round and proceed to further recruiting stages. In the event that I am not selected, it would be greatly beneficial for my future if you could mention the drawbacks I encountered in this task.
- I am confident in my abilities and believe that I will be selected for the next phase of the recruitment process. This task has truly excited me, and I would like to learn more about the job and responsibilities of a Data Scientist Intern at your company.
- **Thank you once again for providing me with this opportunity.**

Submitted by:

B. Datta Sai Srinivas,

8919009721,

dssrinivasbaswa@gmail.com.