Reading the Data and Observations on Dataset

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

```
# Reading the Excel file
df = pd.read_excel('/content/Sample_Longlist_Data.xlsx')
df.head()
```

	Date Added	category	sellerlink	sellerlink-url	sellerstorefront-url	sellerproductcount	sellerratings	sellerdetails	
0	2020-11-15	Garden	Seller 1	Seller 1-link	Seller 1-storefrontlink	1-16 of over 100,000 results	88% positive in the last 12 months (118 ratings)	Lohas Living Inc James Mazzello US 845 3RD Ave...	Busir
1	2020-11-15	Garden	Seller 2	Seller 2-link	Seller 2-storefrontlink	NaN	90% positive in the last 12 months (566 ratings)	Herzlich Willkommen im Amazon Shop von 1a-Hand...	Busines
2	2020-11-15	Garden	Seller 3	Seller 3-link	Seller 3-storefrontlink	1-16 of over 2,000 results	85% positive in the last 12 months (75 ratings)	67878u6i7 is committed to providing each custo...	Name:nantongao
3	2020-11-15	Garden	Seller 4	Seller 4-link	Seller 4-storefrontlink	1-16 of 123 results	NaN	7 days home life supermarket is committed to p...	Business Name:ZI
4	2020-11-15	Garden	Seller 5	Seller 5-link	Seller 5-storefrontlink	1-16 of over 1,000 results	81% positive in the last 12 months (52 ratings)	78 68yui678 is committed to providing each cus...	Name:shenzhenfei

```
#Checking the datatypes of each column
df.dtypes
```

```
Date Added                                    datetime64[ns]
category                                              object
sellerlink                                            object
sellerlink-url                                        object
sellerstorefront-url                                  object
sellerproductcount                                    object
sellerratings                                         object
sellerdetails                                         object
seller business name                                  object
businessaddress                                       object
Count of seller brands                                 int64
Max % of negative seller ratings - last 30 days        int64
Max % of negative seller ratings - last 90 days        int64
Max % of negative seller ratings - last 12 months      int64
Hero Product 1 #ratings                                int64
Hero Product 2 #ratings                                int64
Sample brand name                                     object
Sample Brand URL                                      object
dtype: object
```

```
#Seeing whether 'Garden' is the unique value or not in the Category value
df.category.value_counts()
```

```
Garden    1839
Name: category, dtype: int64
```

```
# Counting the occurrences of each unique date in the 'Date Added' column
df['Date Added'].value_counts()
```

```
2020-11-15    1839
Name: Date Added, dtype: int64
```

```
# Filtering the dataset to select rows containing at least one null value
df[df.isnull().any(axis=1)]
```

	Date Added	category	sellerlink	sellerlink-url	sellerstorefront-url	sellerproductcount	sellerratings	sellerdetails	
1	2020-11-15	Garden	Seller 2	Seller 2-link	Seller 2-storefrontlink	NaN	90% positive in the last 12 months (566 ratings)	Herzlich Willkommen im Amazon Shop von 1a-Hand...	Busin
3	2020-11-15	Garden	Seller 4	Seller 4-link	Seller 4-storefrontlink	1-16 of 123 results	NaN	7 days home life supermarket is committed to p...	Business Name
5	2020-11-15	Garden	Seller 6	Seller 6-link	Seller 6-storefrontlink	1-16 of 28 results	NaN	7830jeff is committed to providing each custom...	
6	2020-11-15	Garden	Seller 7	Seller 7-link	Seller 7-storefrontlink	1-16 of over 4,000 results	NaN	7hgfeee is committed to providing each custome...	Name:quanzh
7	2020-11-15	Garden	Seller 8	Seller 8-link	Seller 8-storefrontlink	1-16 of over 3,000 results	NaN	7s6fguisdhj is committed to providing each cus...	Name:jinange
...	...	...	...	...	...	...	...	...	
1826	2020-11-15	Garden	Seller 1827	Seller 1827-link	Seller 1827-storefrontlink	1-16 of 210 results	NaN	ZYANUGRT is committed to providing each custom...	Name:baoding
1828	2020-11-15	Garden	Seller 1829	Seller 1829-link	Seller 1829-storefrontlink	NaN	NaN	NaN	
1829	2020-11-15	Garden	Seller 1830	Seller 1830-link	Seller 1830-storefrontlink	1-16 of over 10,000 results	NaN	ZYMBH is committed to providing each customer ...	Name:AoHanQiXi
1832	2020-11-15	Garden	Seller 1833	Seller 1833-link	Seller 1833-storefrontlink	NaN	NaN	ZYWLpp is committed to providing each customer...	Name:sh
1835	2020-11-15	Garden	Seller 1836	Seller 1836-link	Seller 1836-storefrontlink	1-16 of over 5,000 results	NaN	ZZMIARY is committed to providing each custome...	Name:wujix

680 rows × 18 columns

From the problem statement, "**Razor does not acquire Chinese sellers at this point, so you can use this data to exclude sellers in China from your analysis,**" from the problem statement. I have concluded that Chinese sellers are not involved in the analysis.

- Decided to remove the rows that include China (CN).

```
#First value of 'businessaddress'
df.businessaddress.iloc[0]
```

```
'411 THEODORE FREMD AVESTE 206 SOUTHRyeNY10580-1410US'
```

```
df.businessaddress.iloc[7]
```

```
'TianQiaoQuTongYuanJuQianJie11HaoAQu204JiNanShiShanDongSheng250000CN'
```

```
df.businessaddress.iloc[90]
```

```
'Einener Dorfbauerschaft42Warendorf48231DE'
```

I have observed that the **country's code/shortcut** is located in the last two letters of every ***'Business Address'***.

```
#Seeing how many null values are present in 'businessaddress'
df['businessaddress'].isnull().sum()
```

```
75
```

```
df[df['businessaddress'].str.endswith('CN', na=False)]
```

	Date Added	category	sellerlink	sellerlink-url	sellerstorefront-url	sellerproductcount	sellerratings	sellerdetails	
2	2020-11-15	Garden	Seller 3	Seller 3-link	Seller 3-storefrontlink	1-16 of over 2,000 results	85% positive in the last 12 months (75 ratings)	67878u6i7 is committed to providing each custo...	Name:nantongac
3	2020-11-15	Garden	Seller 4	Seller 4-link	Seller 4-storefrontlink	1-16 of 123 results	NaN	7 days home life supermarket is committed to p...	Business Name:Z
4	2020-11-15	Garden	Seller 5	Seller 5-link	Seller 5-storefrontlink	1-16 of over 1,000 results	81% positive in the last 12 months (52 ratings)	78 68yui678 is committed to providing each cus...	Name:shenzhenfe
6	2020-11-15	Garden	Seller 7	Seller 7-link	Seller 7-storefrontlink	1-16 of over 4,000 results	NaN	7hgfeee is committed to providing each custome...	Name:quanzhou
7	2020-11-15	Garden	Seller 8	Seller 8-link	Seller 8-storefrontlink	1-16 of over 3,000 results	NaN	7s6fguisdhj is committed to providing each cus...	Name:jinangesł
...	...	...	...	...	...	...	...	...	
1834	2020-11-15	Garden	Seller 1835	Seller 1835-link	Seller 1835-storefrontlink	1-16 of over 2,000 results	83% positive in the last 12 months (46 ratings)	ZZM Store is committed to providing each custo...	Business Nan
1835	2020-11-15	Garden	Seller 1836	Seller 1836-link	Seller 1836-storefrontlink	1-16 of over 5,000 results	NaN	ZZMIARY is committed to providing each custome...	Name:wujixia
1836	2020-11-15	Garden	Seller 1837	Seller 1837-link	Seller 1837-storefrontlink	1-16 of over 10,000 results	0% positive lifetime (1 total ratings)	zzshd75 is committed to providing each custome...	Name:henanliy
1837	2020-11-15	Garden	Seller 1838	Seller 1838-link	Seller 1838-storefrontlink	1-16 of 473 results	67% positive lifetime (3 total ratings)	ZZY xiaodian is committed to providing each cu...	Name:shanxim
1838	2020-11-15	Garden	Seller 1839	Seller 1839-link	Seller 1839-storefrontlink	1-16 of over 20,000 results	100% positive lifetime (2 total ratings)	zzzswbl is committed to providing each custome...	Name:nanchan

1237 rows × 18 columns

```
#Filtering the data excluding the China(CN)
dfwithout_china = df[~df['businessaddress'].str.endswith('CN', na=False)]
```

```
dfwithout_china.businessaddress[90]
```

```
'Einener Dorfbauerschaft42Warendorf48231DE'
```

```
#Finding count of Null values in each column of non CHINA datatable
dfwithout_china.isnull().sum()
```

```
Date Added                                    0
category                                      0
sellerlink                                    0
sellerlink-url                                0
sellerstorefront-url                          0
sellerproductcount                          121
sellerratings                                43
sellerdetails                                26
seller business name                         56
businessaddress                              75
Count of seller brands                        0
Max % of negative seller ratings - last 30 days    0
Max % of negative seller ratings - last 90 days    0
Max % of negative seller ratings - last 12 months  0
Hero Product 1 #ratings                       0
Hero Product 2 #ratings                       0
Sample brand name                             0
Sample Brand URL                              0
dtype: int64
```

```
#First 5 rows of dataset
dfwithout_china.head()
```

	Date Added	category	sellerlink	sellerlink-url	sellerstorefront-url	sellerproductcount	sellerratings	sellerdetails	seller busine na
0	2020-11-15	Garden	Seller 1	Seller 1-link	Seller 1-storefrontlink	1-16 of over 100,000 results	88% positive in the last 12 months (118 ratings)	Lohas Living Inc James Mazzello US 845 3RD Ave...	Busine Name:Lohas Livi I
1	2020-11-15	Garden	Seller 2	Seller 2-link	Seller 2-storefrontlink	NaN	90% positive in the last 12 months (566 ratings)	Herzlich Willkommen im Amazon Shop von 1a-Hand...	Busine Name:1 Handelsagent
5	2020-11-15	Garden	Seller 6	Seller 6-link	Seller 6-storefrontlink	1-16 of 28 results	NaN	7830jeff is committed to providing each custom...	Na
14	2020-11-15	Garden	Seller 15	Seller 15-link	Seller 15-storefrontlink	1-16 of 919 results	97% positive in the last 12 months (116 ratings)	Impressum\n \n Gesetzliche Anbieterkennung:\n ...	Busine Name:Oliver Mi
19	2020-11-15	Garden	Seller 20	Seller 20-link	Seller 20-storefrontlink	NaN	76% positive in the last 12 months (3721 ratings)	Impressum: ABC-Schnäppchenmarkt GmbH \n Jösser...	Busine Name:AB Schnäppchenma Gmb

- Keeping the Columns that are necessary for Analysis

```
#Columns of the dataset
dfwithout_china.columns
```

```
Index(['Date Added', 'category', 'sellerlink', 'sellerlink-url',
       'sellerstorefront-url', 'sellerproductcount', 'sellerratings',
       'sellerdetails', 'seller business name', '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',
       'Sample brand name', 'Sample Brand URL'],
      dtype='object')
```

```
# Filter the DataFrame 'dfwithout_china' to select specific columns of interest
filtered_data = dfwithout_china.loc[:, ['sellerlink',
                                        'sellerproductcount',
                                        'sellerratings', 'sellerdetails', 'seller business name', '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' ]]
```

```
#First five rows of filtered data
filtered_data.head()
```

	sellerlink	sellerproductcount	sellerratings	sellerdetails	seller business name	businessaddress	Count of seller brands	Max % negat sel rati - 1 30 c
0	Seller 1	1-16 of over 100,000 results	88% positive in the last 12 months (118 ratings)	Lohas Living Inc James Mazzello US 845 3RD Ave...	Business Name:Lohas Living Inc	411 THEODORE FREMD AVESTE 206 SOUTHRyeNY10580-...	16	
1	Seller 2	NaN	90% positive in the last 12 months (566 ratings)	Herzlich Willkommen im Amazon Shop von 1a-Hand...	Business Name:1a-Handelsagentur	1a HandelsagenturLindenallee 2MalchowMecklenbu...	0	
5	Seller 6	1-16 of 28 results	NaN	7830jeff is committed to providing each custom...	NaN	NaN	13	
14	Seller 15	1-16 of 919 results	97% positive in the last 12 months (116 ratings)	Impressum\n \n Gesetzliche Anbieterkennung:\n ...	Business Name:Oliver Mills	Kaiserstr.12Borgholzhausen33829DE	15	
19	Seller 20	NaN	76% positive in the last 12 months (3721 ratings)	Impressum: ABC-Schnäppchenmarkt GmbH \n Jösser...	Business Name:ABC-Schnäppchenmarkt GmbH	Jösser Weg10Petershagen32469DE	0	

Feature Engineering

Feature Engineering is the crucial step while dealing with this Data. To enhance the analysis, the following features can be derived from the existing data:

1. Seller Product Count (From 'sellerproductcount' feature)
- Eg: 1-16 of over 100,000 results -> **100000**
2. Positive Percent & Positive Rating Count (from 'sellerratings' feature)
- Eg: 88% positive in the last 12 months (118 ratings) -> **(88, 118)**
3. Mobile number & Gmail (from 'sellerdetails' feature)
- Eg: Telefon: 0049 - (0)30 - 54 70 15 05 Mail: kerri-keramik@web.de -> **(004903054701505, kerri-keramik@web.de)**
4. Businessname (from sellerbusinessname)
- Eg: Business Name:Lohas Living Inc -> **(Lohas Living Inc)**
5. Country Code (from 'businessaddress')
- Eg: 411 THEODORE FREMD AVESTE 206 SOUTHRyeNY10580-1410US -> **(US)**
6. Deleting unwanted Features.

1. Extracting **'seller_product_count'** from ***'sellerproduct'***

```
#Importing 're' to perform Regular Expressions for Extracting necessary information from Dataset
import re
```

```
#Sample to perform RegEx
example = filtered_data.sellerproductcount[0]
```

```
example
```

```
'1-16 of over 100,000 results'
```

```
#RegEx pattern to extract the text(number) that in the format xxx,xxx or xxxx or any number of continous digit
pattern = '(\d{1,3}(?:,\d{3})*|\d+) results'

re.findall(pattern, example)
```

```
['100,000']
```

```
#Applying RegEx 'pattern' on example to find out Seller Product Count
re.findall(pattern, "1-16 of over 1900 results")
```

```
['1900']
```

```
#Function to extract 'seller_product_count'
def seller_product_count(text=None):
    if text is None:
        return np.nan                          #returns 'NaN' if the value is Null

    pattern = r'(\d{1,3}(?:,\d{3})*|\d+) results'
    match = re.findall(pattern, text)
    if match:
        count = match[0].replace(",", "")
        return int(count)
    else:
        return np.nan                          #returns 'NaN' if no text matching the RegEx pattern
```

```
seller_product_count() #Checking that the function working or not for Null values
```

```
nan
```

```
seller_product_count(example) #Applying the function on Example
```

```
100000
```

```
filtered_data['sellerproductcount'] #'sellerproductcount' column before applying the Function
```

```
0       1-16 of over 100,000 results
1                                NaN
5                1-16 of 28 results
14               1-16 of 919 results
19                               NaN
                    ...
1808             1-16 of 52 results
1813            1-16 of 177 results
1814      1-16 of over 1,000 results
1815      1-16 of over 2,000 results
1828                             NaN
Name: sellerproductcount, Length: 602, dtype: object
```

```
filtered_data['sellerproductcount'].astype(str).apply(seller_product_count) #'sellerproductcount' column after applying the Function
```

```
0       100000.0
1            NaN
5           28.0
14         919.0
19           NaN
           ...
1808        52.0
1813       177.0
1814      1000.0
1815      2000.0
1828         NaN
Name: sellerproductcount, Length: 602, dtype: float64
```

```
#Applying the 'seller_product_count' function on the entire column of 'sellerproductcount'
filtered_data['sellerproductcount'] = filtered_data['sellerproductcount'].astype(str).apply(seller_product_count)
```

```
#Checking the table after applying the 'seller_product_count' function
filtered_data.head()
```

	sellerlink	sellerproductcount	sellerratings	sellerdetails	seller business name	businessaddress	Count of seller brands	Max % negat sel rati - 1 30 c
0	Seller 1	100000.0	88% positive in the last 12 months (118 ratings)	Lohas Living Inc James Mazzello US 845 3RD Ave...	Business Name:Lohas Living Inc	411 THEODORE FREMD AVESTE 206 SOUTHRyeNY10580-...	16	
1	Seller 2	NaN	90% positive in the last 12 months (566 ratings)	Herzlich Willkommen im Amazon Shop von 1a-Hand...	Business Name:1a-Handelsagentur	1a HandelsagenturLindenallee 2MalchowMecklenbu...	0	
5	Seller 6	28.0	NaN	7830jeff is committed to providing each custom...	NaN	NaN	13	
14	Seller 15	919.0	97% positive in the last 12 months (116 ratings)	Impressum\n \n Gesetzliche Anbieterkennung:\n ...	Business Name:Oliver Mills	Kaiserstr.12Borgholzhausen33829DE	15	
19	Seller 20	NaN	76% positive in the last 12	Impressum: ABC-Schnäppchenmarkt	Business Name:ABC-	Jösser Weg10Petershagen32469DE	0	

At completion of this section, I have successfully inserted '**seller_product_count**' column derived from ***'sellerproductcount'***

2. Extracting '**positive_percent**' and '**seller_ratings**' from ***'sellerratings'***

```
#Sample to perform RegEx
example2 = filtered_data.sellerratings[0]
example2
```

```
'88% positive in the last 12 months (118 ratings)'
```

```
#1. RegEx pattern to extract the number in the format "number%"
pattern = '(\d+)%'

re.findall(pattern, example2)
```

```
['88']
```

```
#Function to extract 'positive_percent'
def positive_percent(text=None):
    if text is None:
        return np.nan                   #returns 'NaN' if the value is Null

    pattern = r'(\d+)%'
    match = re.findall(pattern, text)
    if match:
        count = match[0]
        return int(count)
    else:
        return np.nan                   #returns 'NaN' if no text matching the RegEx pattern
```

```
positive_percent(example2)    #Applying the function on Example2
```

```
88
```

```
#2. RegEx pattern to extract the number in the format - (xxxx ratings or xxxx total ratings)
pattern = '(\d+)\s+ratings|(\d+)\s+total ratings'

re.findall(pattern, example2)
```

```
[('118', '')]
```

```
#Function to extract 'seller_ratings'

def seller_ratings(text=None):
    if text is None:
        return np.nan                   #returns 'NaN' if the value is Null

    pattern = r'(\d+)\s+ratings|(\d+)\s+total ratings'
    match = re.findall(pattern, text)
```

```
    if match:
        if match[0][0]:
            count = match[0][0]         #returns the value if pattern "xxxx ratings"
            return int(count)
        elif match[0][1]:
            count = match[0][1]         #returns the value if patterns "xxxx total ratings"
            return int(count)

    return np.nan                       #returns 'NaN' if no text matching the RegEx pattern
```

```
#Inserting the new columns 'positive_percent' & 'seller_ratings' next to 'sellerratings'

column_position = filtered_data.columns.get_loc('sellerratings') + 1

filtered_data.insert(column_position, 'positive_percent', filtered_data['sellerratings'].astype(str).apply(positive_percent))

filtered_data.insert(column_position + 1, 'seller_ratings', filtered_data['sellerratings'].astype(str).apply(seller_ratings))
```

```
#Verifying whether table after inserting the new columns 'positive_percent' & 'seller_ratings'
filtered_data.head(10)
```

	sellerlink	sellerproductcount	sellerratings	positive_percent	seller_ratings	sellerdetails	seller business name	
0	Seller 1	100000.0	88% positive in the last 12 months (118 ratings)	88.0	118.0	Lohas Living Inc James Mazzello US 845 3RD Ave...	Business Name:Lohas Living Inc	411 THE 20
1	Seller 2	NaN	90% positive in the last 12 months (566 ratings)	90.0	566.0	Herzlich Willkommen im Amazon Shop von 1a-Hand...	Business Name:1a-Handelsagentur	1a H
5	Seller 6	28.0	NaN	NaN	NaN	7830jeff is committed to providing each custom...	NaN	
14	Seller 15	919.0	97% positive in the last 12 months (116 ratings)	97.0	116.0	Impressum\n \n Gesetzliche Anbieterkennung:\n ...	Business Name:Oliver Mills	Kaiserstr.12
19	Seller 20	NaN	76% positive in the last 12 months (3721 ratings)	76.0	3721.0	Impressum: ABC-Schnäppchenmarkt GmbH \n Jösser...	Business Name:ABC-Schnäppchenmarkt GmbH	Jösser We
20	Seller 21	123.0	NaN	NaN	NaN	AllSparesEire is committed to providing each c...	Business Name:All Terrain Ireland	W
21	Seller 22	NaN	92% positive in the last 12 months (181 ratings)	92.0	181.0	Firmenwortlaut: GURU2016GmbH\n Geschäftsführun...	Business Name:GURU 2016 GmbH	GmbHWar
22	Seller 23	40000.0	100% positive in the last 12 months (10 ratings)	100.0	10.0	Bestof Floral is committed to providing each c...	Business Name:MADEWELL SUPPLY INC	90 BRC
23	Seller 24	100000.0	88% positive in the last 12 months (3134 ratings)	88.0	3134.0	Blumenbecker Industriebedarf GmbH Sudhoferweg ...	Business Name:Blumenbecker Industriebedarf GmbH	
24	Seller 25	3000.0	97% positive in the last 12 months (610 ratings)	97.0	610.0	Herzlich willkommen auf unserer Internetpräsen...	Business Name:Jörg Müller & Petra Scheerer GbR	

At completion of this section, I have successfully inserted '**positive_percent**' & **seller_ratings** column derived from ***'sellerratings'***

3. Extracting '**Mobile number**' and '**Email**' from the ***'sellerdetails'***

```
filtered_data.sellerdetails
```

```
0       Lohas Living Inc James Mazzello US 845 3RD Ave...
1       Herzlich Willkommen im Amazon Shop von 1a-Hand...
5       7830jeff is committed to providing each custom...
14      Impressum\n \n Gesetzliche Anbieterkennung:\n ...
19      Impressum: ABC-Schnäppchenmarkt GmbH \n Jösser...
                              ...
1808    Zündholz Riesa is committed to providing each ...
1813    Impressum:\n\nStephanie Severt\n\nZUR ROSA KUH...
1814    Reinhard Joormann -FansandTrends- e.K.   Im St...
1815    Zwoofershop is committed to providing each cus...
1828                                                  NaN
Name: sellerdetails, Length: 602, dtype: object
```

```
#Sample to perform RegEx
example3 = filtered_data.sellerdetails[14]
example3
```

```
'Impressum\n \n Gesetzliche Anbieterkennung:\n \n Oliver Mills\n 9:PM\n An der Bundesstrasse 26\n 33829 Borgholzhausen\n Deutschla
nd\n Telefon: 015140008562\n E-Mail: webmaster@9pm-store.de\n USt-IdNr.: DE215752000\n \n Wir sind seit\xa001.08.2015\xa0Mitglied
der Initiative "FairCommerce".\n Nähere Informationen hierzu finden Sie unter .fair-commerce.de. Alternative Streitbeilegung: \nDi
e Europäische Kommission bietet eine Onlineplattform für Streitbeilegung an, die Sie hier finden: https://ec.europa.eu/consumers/o
dr/\n\nUnter diesem L   '
```

```
#Function to extract the text(string) in the format of xxxx@xxxx(mail)
def extract_mail(text=None):
    if text is None:
        return np.nan                          #returns 'NaN' if the value is Null

    pattern = r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+(?!\.[A-Za-z]{2})\b'
    match = re.findall(pattern, text)
    if match:
        count = match[0]
        return str(count)
    else:
        return np.nan                          #returns 'NaN' if no text matching the RegEx pattern
```

```
#Applying the function on Example
extract_mail(example3)
```

```
'webmaster@9pm-store.de'
```

```
#Function to extract the text(number) in the format of Telefon:xxxxx or Tel: xxxxxx
def extract_phonenumbers(text=None):
    if text is None:
        return np.nan

    phone_numbers = re.findall(r'(?:Tel|Telefon)[^A-Za-z\s]*([\d\s\/()+.-]+)', text)
    phone_numbers = [re.sub(r'\D', '', number) for number in phone_numbers] #uses a list comprehension to iterate over each phone number

    if phone_numbers:
        count = phone_numbers[0]
        return str(count)
    else:
        return np.nan
```

```
extract_phonenumbers(example3) #Applying the function on Example3
```

```
'015140008562'
```

```
#Inserting the new columns 'Telephone' & 'Email' next to 'sellerdetails'
column_position = filtered_data.columns.get_loc('sellerdetails') + 1
filtered_data.insert(column_position, 'Email', filtered_data['sellerdetails'].astype(str).apply(extract_mail))
filtered_data.insert(column_position + 1, 'Telephone', filtered_data['sellerdetails'].astype(str).apply(extract_phonenumbers))
```

```
filtered_data.head(3)
```

	sellerlink	sellerproductcount	sellerratings	positive_percent	seller_ratings	sellerdetails	Email	Telephor
0	Seller 1	100000.0	88% positive in the last 12 months (118 ratings)	88.0	118.0	Lohas Living Inc James Mazzello US 845 3RD Ave...	jadgemaello@gmail.com	Na
			90% positive in the last 12			Herzlich Willkommen im	info@1a-	

Note: The unstructured data contains various types and patterns of numbers and mails. While considerable efforts were made to extract as much contact information as possible from the 'sellerdetails' field, it is essential to recognize that not all entries have complete or extractable phone numbers and email addresses. Due to time constraints, the extraction process focused on maximizing available data, resulting in some incomplete entries.

Though incomplete extraction of phone numbers and emails does not significantly impact the determination of '**BEST SELLERS**' from the dataset further.

4. Extracting "**Businessname**" from ***seller business name***

```
#Sample to perform RegEx
example4 = filtered_data['seller business name'][0]
example4
```

```
'Business Name:Lohas Living Inc'
```

```
#Function to extract 'businessname'
def businessname(text=None):
  if text is None:
        return np.nan

  matches = re.findall(r"Business Name:(.*)", text)
  if matches:
    extracted_text = matches[0].strip()
    return extracted_text                  #returns the text if matches 'Business Name: xxxxxxx'
  else:
    return np.nan                          # returns 'NaN' if no text matching the RegEx pattern
```

```
#Applying the 'businessname' on the entire column of 'seller business name'
filtered_data['seller business name'] = filtered_data['seller business name'].astype(str).apply(businessname)
```

```
#Renaming the column 'seller business name' as 'businessname'
filtered_data.rename(columns={'seller business name': 'businessname'}, inplace=True)
```

```
#Checking the table after applying the 'businessname' function
filtered_data.head(2)
```

	sellerlink	sellerproductcount	sellerratings	positive_percent	seller_ratings	sellerdetails	Email	Telephor
0	Seller 1	100000.0	88% positive in the last 12 months (118 ratings)	88.0	118.0	Lohas Living Inc James Mazzello US 845 3RD Ave...	jadgemaello@gmail.com	Na
1	Seller 2	NaN	90% positive in the last 12 months (566 ratings)	90.0	566.0	Herzlich Willkommen im Amazon Shop von 1a-Hand...	info@1a-handelsagentur.de	0399328297

At completion of this section, I have successfully inserted '**businessname**' column derived from *'seller business name'*

5. Extracting '**country_code**' from 'businessaddress'

- The '**country_code**' is extracted from the 'businessaddress' by considering the last two characters of each text value.
- However, a challenge arises when some addresses end with '**GB**', which are not unique within the column (**e.g., NLGB, RLGB, RBGB, etc.**).
- To address this issue, I have made an exception and finalized to extract the '**last four characters**' if the case is '**GB**' as the last characters, ensuring uniqueness among these addresses.

```
#Function to extract the Country code
def extracting_last_two_char(text):
  if text is None:
        return np.nan         #returns 'NaN' if the value is Null


  if len(text) >= 2:
        if text[-2:] == 'GB':
            return text[-4:]    #returns the last 4 characters if the ending characters were 'GB'
        else:
            return text[-2:]    #returns the last 2 characters if the ending characters are other than 'GB'
  else:
    return np.nan             #returns 'NaN' if there's no matching the RegEx pattern
```

```
#Applying the 'extracting_last_two_char' function on the entire column of 'businessaddress'
filtered_data.businessaddress = filtered_data.businessaddress.astype(str).apply(extracting_last_two_char)
```

```
#Checking the table after applying the 'extracting_last_two_char' function
filtered_data.head()
```

	sellerlink	sellerproductcount	sellerratings	positive_percent	seller_ratings	sellerdetails	Email	Tele
0	Seller 1	100000.0	88% positive in the last 12 months (118 ratings)	88.0	118.0	Lohas Living Inc James Mazzello US 845 3RD Ave...	jadgemaello@gmail.com	
1	Seller 2	NaN	90% positive in the last 12 months (566 ratings)	90.0	566.0	Herzlich Willkommen im Amazon Shop von 1a-Hand...	info@1a-handelsagentur.de	039932
5	Seller 6	28.0	NaN	NaN	NaN	7830jeff is committed to providing each custom...	NaN	
14	Seller 15	919.0	97% positive in the last 12 months (116 ratings)	97.0	116.0	Impressum\n \n Gesetzliche Anbieterkennung:\n ...	webmaster@9pm-store.de	015140
19	Seller 20	NaN	76% positive in the last 12 months (3721 ratings)	76.0	3721.0	Impressum: ABC-Schnäppchenmarkt GmbH \n Jösser...	abc-markt@web.de	0570

```
##Seeing businessaddress value counts
filtered_data.businessaddress.value_counts()
```

```
DE      364
an       75
IT       25
ES       15
US       14
       ...
UYGB      1
APGB      1
DPGB      1
GNGB      1
CH        1
Name: businessaddress, Length: 65, dtype: int64
```

6. Deleting **unwanted features**

```
filtered_data.columns
```

```
Index(['sellerlink', 'sellerproductcount', 'sellerratings', 'positive_percent',
       'seller_ratings', 'sellerdetails', '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')
```

```
#Dropping unnecessary columns for further analysis
final_df = filtered_data.drop(['sellerratings','sellerdetails'],axis=1)
```

```
#Checking the dataset after dropping the unwanted columns
final_df.head()
```

	sellerlink	sellerproductcount	positive_percent	seller_ratings	Email	Telephone	businessname	business
0	Seller 1	100000.0	88.0	118.0	jadgemaello@gmail.com	NaN	Lohas Living Inc	
1	Seller 2	NaN	90.0	566.0	info@1a-handelsagentur.de	039932829721	1a-Handelsagentur	
5	Seller 6	28.0	NaN	NaN	NaN	NaN	NaN	
14	Seller 15	919.0	97.0	116.0	webmaster@9pm-store.de	015140008562	Oliver Mills	
19	Seller 20	NaN	76.0	3721.0	abc-markt@web.de	0570591155	ABC-Schnäppchenmarkt GmbH	

2. Data Preprocessing

1. Dealing with **Null Values**

```
#Rectifying the mistake by replacing 'an' by 'NaN' values in 'businessaddres'
final_df['businessaddress'] = final_df['businessaddress'].replace('an', np.nan)
```

```
#Seeing how many null values are present in each individual columns
final_df.isnull().sum()
```

```
sellerlink                                      0
sellerproductcount                            121
positive_percent                               43
seller_ratings                                 43
Email                                         284
Telephone                                     291
businessname                                   77
businessaddress                                75
Count of seller brands                          0
Max % of negative seller ratings - last 30 days 0
Max % of negative seller ratings - last 90 days 0
Max % of negative seller ratings - last 12 months 0
Hero Product 1 #ratings                         0
Hero Product 2 #ratings                         0
dtype: int64
```

```
#Seeing the individual column statistics to identify the best parameters to replace the null values
final_df.describe()
```

	sellerproductcount	positive_percent	seller_ratings	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
count	481.000000	559.000000	559.000000	602.000000	602.000000	602.000000	602.000000	602.00000	602.000000
mean	6538.110187	90.266547	462.515206	9.513289	5.310631	5.735880	7.303987	4110.52990	2271.294020
std	18662.725294	14.258198	1824.115487	6.736593	14.626857	12.610972	13.999249	10315.68045	6414.703705
min	3.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000
25%	257.000000	89.000000	21.000000	1.000000	0.000000	0.000000	0.000000	5.00000	2.000000
50%	898.000000	94.000000	82.000000	14.000000	0.000000	0.000000	3.000000	456.50000	225.500000
75%	3000.000000	98.000000	280.000000	15.000000	3.000000	6.000000	8.000000	2980.25000	1734.000000
max	200000.000000	100.000000	37124.000000	16.000000	100.000000	100.000000	100.000000	86856.00000	70732.000000

```
final_df['positive_percent'].value_counts()
```

```
100.0    82
97.0     52
96.0     44
98.0     40
94.0     38
95.0     35
99.0     26
90.0     24
92.0     24
91.0     22
93.0     20
89.0     18
86.0     15
88.0     11
87.0     11
80.0      9
82.0      8
84.0      7
81.0      6
67.0      6
83.0      5
0.0       5
73.0      5
85.0      4
79.0      4
75.0      4
70.0      4
57.0      3
50.0      3
76.0      3
71.0      2
33.0      2
78.0      2
77.0      2
53.0      1
72.0      1
9.0       1
40.0      1
41.0      1
55.0      1
25.0      1
69.0      1
60.0      1
65.0      1
42.0      1
44.0      1
43.0      1
Name: positive_percent, dtype: int64
```

```
final_df['positive_percent'].median()
```

```
94.0
```

```
#Assigning the 'Median' value of 'sellerproductcount' to the variable 'spc_median_value'
spc_median_value = final_df['sellerproductcount'].median()

#Replacing the 'Null' values in 'sellerproductcount' by 'spc_median_value'
final_df['sellerproductcount'] = final_df['sellerproductcount'].fillna(spc_median_value)
```

```
#Assigning the 'Median' value of 'positive_percent' to the variable 'pp_median_value'
pp_median_value = final_df['positive_percent'].median()
```

```
#Replacing the 'Null' values in 'positive_percent' by 'pp_median_value'
final_df['positive_percent'] = final_df['positive_percent'].fillna(pp_median_value)
```

```
#Assigning the 'Median' value of 'seller_ratings' to the variable 'sr_median_value'
sr_median_value = final_df['seller_ratings'].median()

#Replacing the 'Null' values in 'seller_ratings' by 'sr_median_value'
final_df['seller_ratings'] = final_df['seller_ratings'].fillna(sr_median_value)
```

```
final_df.head()
```

	sellerlink	sellerproductcount	positive_percent	seller_ratings	Email	Telephone	businessname	business
0	Seller 1	100000.0	88.0	118.0	jadgemaello@gmail.com	NaN	Lohas Living Inc	
1	Seller 2	898.0	90.0	566.0	info@1a-handelsagentur.de	039932829721	1a-Handelsagentur	
5	Seller 6	28.0	94.0	82.0	NaN	NaN	NaN	
14	Seller 15	919.0	97.0	116.0	webmaster@9pm-store.de	015140008562	Oliver Mills	
19	Seller 20	898.0	76.0	3721.0	abc-markt@web.de	0570591155	ABC-Schnäppchenmarkt GmbH	

```
final_df.isnull().sum()
```

```
sellerlink                                  0
sellerproductcount                          0
positive_percent                            0
seller_ratings                              0
Email                                     284
Telephone                                 291
businessname                               77
businessaddress                            75
Count of seller brands                      0
Max % of negative seller ratings - last 30 days     0
Max % of negative seller ratings - last 90 days     0
Max % of negative seller ratings - last 12 months   0
Hero Product 1 #ratings                     0
Hero Product 2 #ratings                     0
dtype: int64
```

- There's still some Null values in rows like columns like 'Email', 'Telephone','Businessname' & 'Businessaddress'
- But they are not able to replace with any of the 'Central Tendency'(Except 'Businessaddress').
- So leaving them as it is, so they are not much impactful on further Analysis.

```
#Exporting the final data as 'Data_for_Anaysis' as a separate file for further analysis
final_df.to_excel('Data_for_Analysis.xlsx',index=False)
```

- The preprocessing part of the data has been completed.
- Feature engineering has been performed, and null values have been dealt with by replacing them with central tendencies.
- The processed dataframe has been exported for further analysis to determine the 'Best Sellers'.
- To avoid confusion, the data analysis has been conducted in a separate .ipynb file, separating it from the preprocessing steps.

Submitted by:

B. Datta Sai Srinivas,

8919009721,

dssrinivasbaswa@gmail.com.