Exploratory Data Analysis :

Exploratory data analysis (EDA) is used to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods. It helps determine how best to manipulate data sources to get the answers you need, making it easier for data scientists to discover patterns, spot anomalies, test a hypothesis, or check assumptions.

Here we are using the Google Playstore dataset, which contains details about the Apps in playstore, there are more than 10,0000+ Apps in the playstore.The size of the dataset is 210Mb.

The main objective of this project is to deliver insights to understand customer demands better and thus help developers to popularize the product.

The dataset we are using is taken from the Kaggle, the link of the dataset is given below

↪ Exploratory Data Analysis on Google playstore dataset

↪ Dataset link 🔗 kaggle link

Importing the dependecies

In [2]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

Data Preparation and Cleaing

↪ Load the csv file with the pandas

↪ creating the dataframe and understanding the data present in the dataset

↪ Dealing with the missing data and the incorrect records

In [4]:

```
df=pd.read_csv("Google-Playstore.csv")
```

In [5]:

```
df.head(5)
```

Out[5]:

	App Name	App Id	Category	Rating	Rating Count	Installs	Minimum Installs	Maximum Installs	Free	Price	...	Developer Website
0	Gakondo	com.ishakwe.gakondo	Adventure	0.0	0.0	10+	10.0	15	True	0.0	...	https://beniyizibyose.tk/#/
1	Ampere Battery Info	com.webserveis.batteryinfo	Tools	4.4	64.0	5,000+	5000.0	7662	True	0.0	...	https://webserveis.netlify.app/
2	Vibook	com.doantiepvien.crm	Productivity	0.0	0.0	50+	50.0	58	True	0.0	...	NaN
3	Smart City Trichy Public Service Vehicles 17UC...	cst.stJoseph.ug17ucs548	Communication	5.0	5.0	10+	10.0	19	True	0.0	...	http://www.climatesmarttech.com/
4	GROW.me	com.horodyski.grower	Tools	0.0	0.0	100+	100.0	478	True	0.0	...	http://www.horodyski.com.pl

5 rows × 24 columns

In [6]:

```
1 df.columns
```

Out[6]:

```
Index(['App Name', 'App Id', 'Category', 'Rating', 'Rating Count', 'Installs',
       'Minimum Installs', 'Maximum Installs', 'Free', 'Price', 'Currency',
       'Size', 'Minimum Android', 'Developer Id', 'Developer Website',
       'Developer Email', 'Released', 'Last Updated', 'Content Rating',
       'Privacy Policy', 'Ad Supported', 'In App Purchases', 'Editors Choice',
       'Scraped Time'],
      dtype='object')
```

In [7]:

```
1 df.shape
```

Out[7]:

```
(2312944, 24)
```

In [8]:

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2312944 entries, 0 to 2312943
Data columns (total 24 columns):
 #   Column             Dtype  
---  ------             -----  
 0   App Name           object 
 1   App Id             object 
 2   Category           object 
 3   Rating             float64
 4   Rating Count       float64
 5   Installs           object 
 6   Minimum Installs   float64
 7   Maximum Installs   int64  
 8   Free               bool   
 9   Price              float64
 10  Currency           object 
 11  Size               object 
 12  Minimum Android    object 
 13  Developer Id       object 
 14  Developer Website  object 
 15  Developer Email    object 
 16  Released           object 
 17  Last Updated       object 
 18  Content Rating     object 
 19  Privacy Policy     object 
 20  Ad Supported       bool   
 21  In App Purchases   bool   
 22  Editors Choice     bool   
 23  Scraped Time       object 
dtypes: bool(4), float64(4), int64(1), object(15)
memory usage: 361.8+ MB
```

Observation:

The dataset having 2312944 rows and 24 columns

The columns in the dataset are:

'App Name', 'App Id', 'Category', 'Rating', 'Rating Count, Installs','Minimum Installs', 'Maximum Installs', 'Free', 'Price', 'Currency','Size', 'Minimum Android', 'Developer Id', 'Developer Website','Developer Email', 'Released', 'Last Updated', 'Content Rating','Privacy Policy', 'Ad Supported', 'In App Purchases', 'Editors Choice','Scraped Time'

Descriptive statistics

In [9]:

```
1 df.describe()
```

Out[9]:

	Rating	Rating Count	Minimum Installs	Maximum Installs	Price
count	2.290061e+06	2.290061e+06	2.312837e+06	2.312944e+06	2.312944e+06
mean	2.203152e+00	2.864839e+03	1.834452e+05	3.202017e+05	1.034992e-01
std	2.106223e+00	2.121626e+05	1.513144e+07	2.355495e+07	2.633127e+00
min	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	0.000000e+00	0.000000e+00	5.000000e+01	8.400000e+01	0.000000e+00
50%	2.900000e+00	6.000000e+00	5.000000e+02	6.950000e+02	0.000000e+00
75%	4.300000e+00	4.200000e+01	5.000000e+03	7.354000e+03	0.000000e+00
max	5.000000e+00	1.385576e+08	1.000000e+10	1.205763e+10	4.000000e+02

Observation :

we have 5 numerical columns in our dataset To look the entire columns we can use pandas functon pd.set_option()

In [10]:

```
1 pd.set_option('display.max_columns', None)
2
```

In [11]:

```
1
2 df.head()
```

Out[11]:

	App Name	App Id	Category	Rating	Rating Count	Installs	Minimum Installs	Maximum Installs	Free	Price	Currency	Size	Minimum Android	Develo
0	Gakondo	com.ishakwe.gakondo	Adventure	0.0	0.0	10+	10.0	15	True	0.0	USD	10M	7.1 and up	Cor NIYIZIB
1	Ampere Battery Info	com.webserveis.batteryinfo	Tools	4.4	64.0	5,000+	5000.0	7662	True	0.0	USD	2.9M	5.0 and up	Webs
2	Vibook	com.doantiepvien.crm	Productivity	0.0	0.0	50+	50.0	58	True	0.0	USD	3.7M	4.0.3 and up	Cabir
3	Smart City Trichy Public Service Vehicles 17UC...	cst.stJoseph.ug17ucs548	Communication	5.0	5.0	10+	10.0	19	True	0.0	USD	1.8M	4.0.3 and up	C Smart
4	GROW.me	com.horodyski.grower	Tools	0.0	0.0	100+	100.0	478	True	0.0	USD	6.2M	4.1 and up	Rafal Hor

Missing values in the data

In [13]:

```
df.isnull().sum().sort_values(ascending = False)
```

Out[13]:

```
Developer Website    760835
Privacy Policy       420953
Released              71053
Rating                22883
Rating Count          22883
Minimum Android        6530
Size                    196
Currency                135
Installs                107
Minimum Installs        107
Developer Id             33
Developer Email          31
App Name                  2
App Id                    0
Price                     0
Free                      0
Maximum Installs          0
Last Updated              0
Content Rating            0
Category                  0
Ad Supported              0
In App Purchases          0
Editors Choice            0
Scraped Time              0
dtype: int64
```

Exploratory Analysis and Visualization

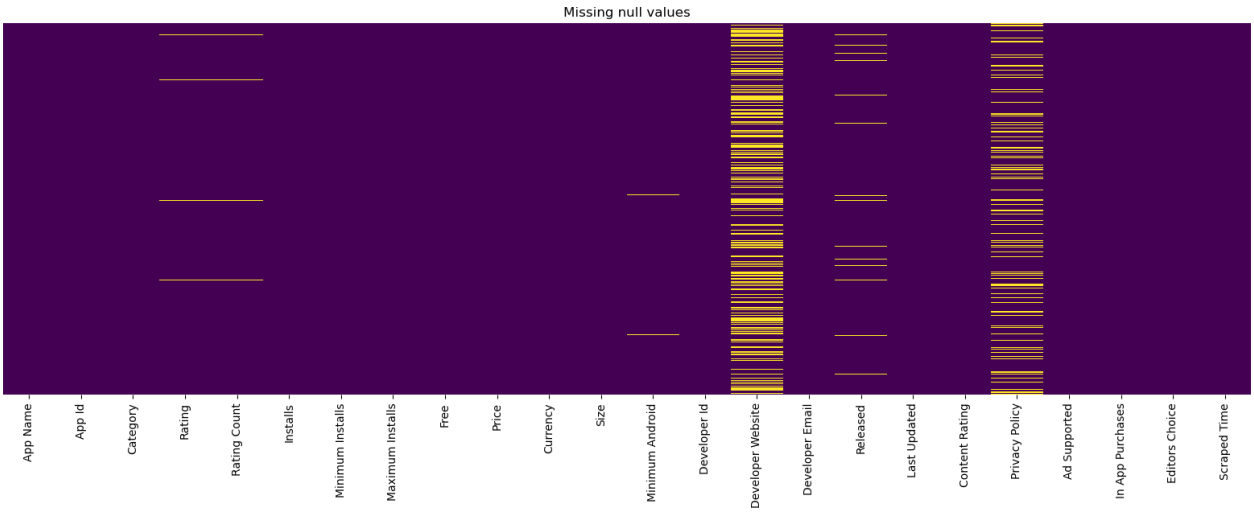
Let's plot the missing null values

In [15]:

```
import matplotlib
matplotlib.rcParams['figure.figsize'] = (20,6)
sns.heatmap(df.isnull(),yticklabels = False, cbar = False , cmap = 'viridis')
plt.title("Missing null values")
```

Out[15]:

```
Text(0.5, 1.0, 'Missing null values')
```




```
Let's see the missing percentage in data
```

In [16]:

```
missing_percentage = df.isnull().sum().sort_values(ascending = False)/len(df)
missing_percentage
```

Out[16]:

```
Developer Website    3.289466e-01
Privacy Policy       1.819988e-01
Released             3.071972e-02
Rating               9.893452e-03
Rating Count         9.893452e-03
Minimum Android      2.823242e-03
Size                 8.474049e-05
Currency             5.836717e-05
Installs             4.626139e-05
Minimum Installs     4.626139e-05
Developer Id         1.426753e-05
Developer Email      1.340283e-05
App Name             8.646988e-07
App Id               0.000000e+00
Price                0.000000e+00
Free                 0.000000e+00
Maximum Installs     0.000000e+00
Last Updated         0.000000e+00
Content Rating       0.000000e+00
Category             0.000000e+00
Ad Supported         0.000000e+00
In App Purchases     0.000000e+00
Editors Choice       0.000000e+00
Scraped Time         0.000000e+00
dtype: float64
```

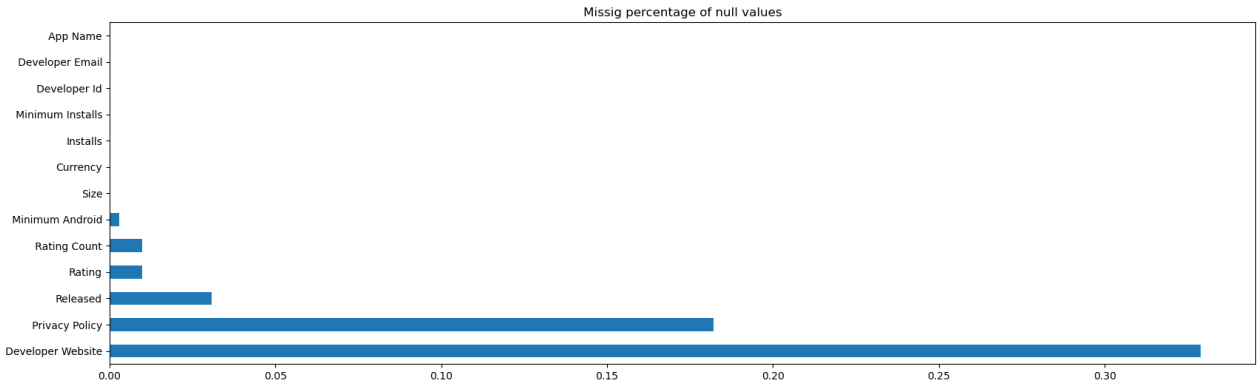
Let's plot the missing percentage of null values

In [17]:

```
missing_percentage = missing_percentage[missing_percentage != 0] # Only the missing data
import matplotlib
matplotlib.rcParams['figure.figsize'] = (20,6)
missing_percentage.plot(kind = 'barh')
plt.title("Missig percentage of null values")
```

Out[17]:

```
Text(0.5, 1.0, 'Missig percentage of null values')
```



Observation :

Dealing with the null values

The columns having highest percentage of null values are : Developer Website Privacy Policy these two are not that much helpful if we want we can drop these columns and in the further steps we drop and add some features in the dataset after the Analysis.

we can drop the small null values for the columns Size Currency Installs Minimum Installs Developer Id Developer Email Rating,Rating Count,Released are important column so it is good to fill the null values. Rating Rating count Minimum Andriod Released

```
from observation 2 droping the null values of the columns got selected

```

In [18]:

```
df.dropna(subset = ['App Name','Size','Currency','Installs','Minimum Installs','Developer Id','Developer Email'], inplace = True)
```

In [19]:

```
df.isnull().sum()
```

Out[19]:

```
App Name                  0
App Id                    0
Category                  0
Rating                22572
Rating Count          22572
Installs                  0
Minimum Installs          0
Maximum Installs          0
Free                      0
Price                     0
Currency                  0
Size                      0
Minimum Android        6530
Developer Id              0
Developer Website    760716
Developer Email           0
Released              70748
Last Updated              0
Content Rating            0
Privacy Policy       420845
Ad Supported              0
In App Purchases          0
Editors Choice            0
Scraped Time              0
dtype: int64
```

Now let's start cleaning the each row

```
checking any duplicates in the App Name
```

In [20]:

```
boolean = df['App Name'].duplicated().any()
boolean
```

Out[20]:

```
True
```

In [21]:

```
df['App Name'].value_counts()
```

Out[21]:

```
Tic Tac Toe                     382
Calculator                      261
Flashlight                      256
BMI Calculator                  201
Age Calculator                  190
                               ...
Foodiz Resto                      1
Astra OtoReward                   1
MedTerm Dictionary                1
Urban garden and sowing.          1
Biliyor Musun - Sonsuz Yarış      1
Name: App Name, Length: 2177593, dtype: int64
```

```
 from the above cell output we can see that App Name column has multiple rows. Let's check out that data in those rows is identical or not
```

In [22]:

```
df[df['App Name'] == 'Tic Tac Toe']
```

Out[22]:

Rating Count	Installs	Minimum Installs	Maximum Installs	Free	Price	Currency	Size	Minimum Android	Developer Id	Developer Website	
0.0	10+	10.0	30	True	0.0	USD	3.4M	4.0 and up	ATTE AKA FRANCOIS	https://ne2ad32ee.app-ads-txt.com	dev.atte.fra
0.0	10+	10.0	31	True	0.0	USD	34M	4.3 and up	Smart Digital Tools	NaN	msms.z
11.0	5,000+	5000.0	6379	True	0.0	USD	7.4M	7.0 and up	Sundeep Reddy Thirumuru	NaN	sundeep.thiru
0.0	500+	500.0	922	True	0.0	USD	8.6M	4.1 and up	SteadFast Games	NaN	devme
0.0	10+	10.0	38	True	0.0	USD	927k	4.0 and up	SnatchDreams	https://www.snatchdreams.net	snatchdr
...	...	...	...	...	...	...	...	...	...	...	
9269.0	500,000+	500000.0	800662	True	0.0	USD	2.3M	2.3 and up	Mobile Apps Pro	NaN	sma
0.0	50+	50.0	50	True	0.0	USD	3.5M	5.0 and up	Jainam Jhaveri	https://erjainam.com	engineerja
7.0	10+	10.0	34	True	0.0	USD	5.3M	4.1 and up	Cybertron's Hogwarts	http://prudhvikchirunomula.pythonanywhere.com/	prudhvik
20.0	100+	100.0	462	True	0.0	USD	2.1M	2.2 and up	Infuse Apps	http://www.infuseapps.com	infuseandroic
10.0	100+	100.0	173	True	0.0	USD	1.7M	4.4 and up	RentMyTent	NaN	justma

In [23]:

```
df['App Id'].duplicated().any()
```

Out[23]:

```
False
```

In [24]:

```
df['App Id'].value_counts()
```

Out[24]:

```
com.ishakwe.gakondo                    1
com.avai.amp.dewtour                   1
com.myhomebuy                          1
com.apocalipseescatologia.app          1
com.puzzlegame.wordconnect             1
                                      ..
id.compro.virtualcompetition           1
com.fitivity.basketball_point_guard    1
smartgr.gardenshower.ideas             1
com.euroland.mobiletools.ae_nbad       1
com.yyazilim.biliyormusun              1
Name: App Id, Length: 2312548, dtype: int64
```

```
1. we have the Apps with the same names but with the unique App IDs so the Apps are differed based on the App IDs
```

let's Explore the numerical columns

In [25]:

```
df['Installs'].unique()
```

Out[25]:

```
array(['10+', '5,000+', '50+', '100+', '1,000+', '500+', '50,000+',
       '10,000+', '1+', '500,000+', '100,000+', '5+', '10,000,000+',
       '1,000,000+', '5,000,000+', '0+', '100,000,000+', '50,000,000+',
       '1,000,000,000+', '500,000,000+', '5,000,000,000+',
       '10,000,000,000+'], dtype=object)
```

```
It is in the object type, we need to change it into the int type
```

In [27]:

```
df['Installs'] = df['Installs'].str.split('+').str[0]    # reomoves the + symbol

```

In [28]:

```
df["Installs"].replace(',','',regex=True , inplace=True)
```

In [30]:

```
df["Installs"].dtype
```

Out[30]:

```
dtype('O')
```

In [32]:

```
df["Installs"]=pd.to_numeric(df["Installs"])
```

In [34]:

```
df["Installs"].dtype
```

Out[34]:

```
dtype('int64')
```

In [35]:

```
df['Currency'].unique()
```

Out[35]:

```
array(['USD', 'XXX', 'CAD', 'EUR', 'INR', 'VND', 'GBP', 'BRL', 'KRW',
       'TRY', 'RUB', 'SGD', 'AUD', 'PKR', 'ZAR'], dtype=object)
```

In [36]:

```
df['Size'].unique()
```

Out[36]:

```
array(['10M', '2.9M', '3.7M', ..., '405M', '3.2k', '512M'], dtype=object)
```

```
The Size of data can be in GB, MB and KB let's convert the data into the size in MB
```

In [37]:

```
df['Size'] = df['Size'].apply(lambda x: str(x).replace('M', '') if 'M' in str(x) else x)

```

```
here we get a missmatched value with the data
we got the value 1,018 we can drop it or we can assume as it may be a '.' (dot) the would incorrectly added to the dataset. so
let assume it as dot for now and repalce the ',' with the dot '.'
```

In [41]:

```
df['Size'] = df['Size'].apply(lambda x: str(x).replace(',', '.') if ',' in str(x) else x)

```

```
1
```

In [47]:

```
1 df['Minimum Android']
```

Out[47]:

```
0            7.1 and up
1            5.0 and up
2          4.0.3 and up
3          4.0.3 and up
4            4.1 and up
               ...
2312939      4.1 and up
2312940      4.1 and up
2312941      5.0 and up
2312942      5.0 and up
2312943      5.0 and up
Name: Minimum Android, Length: 2312548, dtype: object
```

In [48]:

```
1 df['Content Rating']
```

Out[48]:

```
0          Everyone
1          Everyone
2          Everyone
3          Everyone
4          Everyone
             ...
2312939        Teen
2312940    Everyone
2312941    Everyone
2312942    Everyone
2312943    Everyone
Name: Content Rating, Length: 2312548, dtype: object
```

In [49]:

```
1 df['Released']
```

Out[49]:

```
0          Feb 26, 2020
1          May 21, 2020
2           Aug 9, 2019
3          Sep 10, 2018
4          Feb 21, 2020
               ...
2312939             NaN
2312940    Jan 17, 2018
2312941    Aug 19, 2018
2312942     Aug 1, 2016
2312943     Aug 9, 2019
Name: Released, Length: 2312548, dtype: object
```

In [50]:

```
1 df['Last Updated']
```

Out[50]:

```
0          Feb 26, 2020
1          May 06, 2021
2          Aug 19, 2019
3          Oct 13, 2018
4          Nov 12, 2018
               ...
2312939    Jun 01, 2021
2312940    Feb 02, 2018
2312941    Aug 19, 2018
2312942    May 05, 2021
2312943    Aug 19, 2019
Name: Last Updated, Length: 2312548, dtype: object
```

In [51]:

```
df['Privacy Policy']
```

Out[51]:

```
0                       https://beniyizibyose.tk/projects/ (https://beniyizibyose.tk/projects/)
1          https://dev4phones.wordpress.com/licencia-de-uso/ (https://dev4phones.wordpress.com/licencia-de-uso/)
2          https://www.vietnamairlines.com/vn/en/terms-an... (https://www.vietnamairlines.com/vn/en/terms-an...)
3                                                        NaN
4                              http://www.horodyski.com.pl (http://www.horodyski.com.pl)
                                                  ...
2312939              http://a.4399sy.com.hk/user/aggreement (http://a.4399sy.com.hk/user/aggreement)
2312940      http://www.oru.edu/about-oru/privacy-policy.php (http://www.oru.edu/about-oru/privacy-policy.php)
2312941    https://appoworld.000webhostapp.com/datastruct... (https://appoworld.000webhostapp.com/datastruct...)
2312942    https://docs.google.com/document/d/1x-9reZuLRX... (https://docs.google.com/document/d/1x-9reZuLRX...)
2312943    https://biliyor-musun-sons.flycricket.io/priva... (https://biliyor-musun-sons.flycricket.io/priva...)
Name: Privacy Policy, Length: 2312548, dtype: object
```

In [52]:

```
df['Scraped Time']
```

Out[52]:

```
0          2021-06-15 20:19:35
1          2021-06-15 20:19:35
2          2021-06-15 20:19:35
3          2021-06-15 20:19:35
4          2021-06-15 20:19:35
                  ...
2312939    2021-06-16 12:59:18
2312940    2021-06-16 12:59:19
2312941    2021-06-16 12:59:19
2312942    2021-06-16 12:59:19
2312943    2021-06-16 12:59:19
Name: Scraped Time, Length: 2312548, dtype: object
```

In [53]:

```
df['Free']
```

Out[53]:

```
0          True
1          True
2          True
3          True
4          True
           ...
2312939    True
2312940    True
2312941    True
2312942    True
2312943    True
Name: Free, Length: 2312548, dtype: bool
```

```
Creataing the column type for free and paid Apps by using the Free column, it's helpfull while dealing with the paid and Free Apps


```

In [54]:

```
df['Type'] = np.where(df['Free'] == True,'Free','Paid')
df.drop(['Free'],axis=1, inplace= True )
```

```
let's clean the Content Rating column
```

In [55]:

```
df['Content Rating'].unique()
```

Out[55]:

```
array(['Everyone', 'Teen', 'Mature 17+', 'Everyone 10+',
       'Adults only 18+', 'Unrated'], dtype=object)
```

In [56]:

```
df['Content Rating'].value_counts()
```

Out[56]:

```
Everyone           2021788
Teen                196311
Mature 17+           60278
Everyone 10+         33881
Unrated                154
Adults only 18+        136
Name: Content Rating, dtype: int64
```

Observation :

we have varies Categories in the content Rating column : Everyone Teen Mature 17+ Everyone 10+ Unrated Adults only 18+ Now, we makes this Categories to a simple 3 Categories for better understanding : Everyone, Teen, Adults

Mature 17+ ----> to Adults Everyone 10+ ----> to Teen Unrated ----> to Everyone Adults only 18+ -> to Adults

In [57]:

```
df["Content Rating"]=df["Content Rating"].replace("Unrated","Everyone")
df["Content Rating"]=df["Content Rating"].replace("Everyone 10+","Teen")
df["Content Rating"]=df["Content Rating"].replace("Mature 17+","Adults")
df["Content Rating"]=df["Content Rating"].replace("Adults only 18+","Adults")
```

In [58]:

```
df['Content Rating'].unique()
```

Out[58]:

```
array(['Everyone', 'Teen', 'Adults'], dtype=object)
```

In [59]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2312548 entries, 0 to 2312943
Data columns (total 24 columns):
 #   Column             Dtype
---  ------             -----
 0   App Name           object
 1   App Id             object
 2   Category           object
 3   Rating             float64
 4   Rating Count       float64
 5   Installs           int64
 6   Minimum Installs   float64
 7   Maximum Installs   int64
 8   Price              float64
 9   Currency           object
 10  Size               object
 11  Minimum Android    object
 12  Developer Id       object
 13  Developer Website  object
 14  Developer Email    object
 15  Released           object
 16  Last Updated       object
 17  Content Rating     object
 18  Privacy Policy     object
 19  Ad Supported       bool
 20  In App Purchases   bool
 21  Editors Choice     bool
 22  Scraped Time       object
 23  Type               object
dtypes: bool(3), float64(4), int64(2), object(15)
memory usage: 394.8+ MB
```

In [60]:

```
df['Rating'].unique()
```

Out[60]:

```
array([0. , 4.4, 5. , 4.5, 2. , 4.7, 4.9, 3.9, 3.7, 4.2, 3.4, 3.8, 4.6,
       4.1, 2.5, 2.3, 2.1, 2.7, 4.3, 3.1, 4.8, 3.3, 4. , 2.4, 3.2, 3.6,
       2.6, nan, 1.5, 3.5, 2.9, 1.3, 2.8, 3. , 2.2, 1.8, 1.4, 1.2, 1.9,
       1.7, 1.6, 1. , 1.1])
```

In [61]:

```
df['Rating Count'].unique()
```

Out[61]:

```
array([0.0000e+00, 6.4000e+01, 5.0000e+00, ..., 8.7553e+04, 7.5960e+04,
       7.8351e+04])
```

In [62]:

```
df['Rating Count'].max()
```

Out[62]:

```
138557570.0
```

In [63]:

```
df['Rating Type'] = 'NoRatingProvided'
df.loc[(df['Rating Count'] > 0) & (df['Rating Count'] <= 10000.0),'Rating Type'] = 'Less than 10K'
df.loc[(df['Rating Count'] > 10000) & (df['Rating Count'] <= 500000.0),'Rating Type'] = 'Between 10K and 500K'
df.loc[(df['Rating Count'] > 500000) & (df['Rating Count'] <= 138557570.0),'Rating Type'] = 'More than 500K'
df['Rating Type'].value_counts()
```

Out[63]:

```
Less than 10K           1192801
NoRatingProvided        1082303
Between 10K and 500K      35779
More than 500K             1665
Name: Rating Type, dtype: int64
```

In [64]:

```
df['Rating Type']
```

Out[64]:

```
0              NoRatingProvided
1                 Less than 10K
2              NoRatingProvided
3                 Less than 10K
4              NoRatingProvided
                    ...
2312939    Between 10K and 500K
2312940        NoRatingProvided
2312941        NoRatingProvided
2312942           Less than 10K
2312943           Less than 10K
Name: Rating Type, Length: 2312548, dtype: object
```

In [65]:

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2312548 entries, 0 to 2312943
Data columns (total 25 columns):
 #   Column             Dtype
---  ------             -----
 0   App Name           object
 1   App Id             object
 2   Category           object
 3   Rating             float64
 4   Rating Count       float64
 5   Installs           int64
 6   Minimum Installs   float64
 7   Maximum Installs   int64
 8   Price              float64
 9   Currency           object
 10  Size               object
 11  Minimum Android    object
 12  Developer Id       object
 13  Developer Website  object
 14  Developer Email    object
 15  Released           object
 16  Last Updated       object
 17  Content Rating     object
 18  Privacy Policy     object
 19  Ad Supported       bool
 20  In App Purchases   bool
 21  Editors Choice     bool
 22  Scraped Time       object
 23  Type               object
 24  Rating Type        object
dtypes: bool(3), float64(4), int64(2), object(16)
memory usage: 412.4+ MB
```

In [ ]:

```
1
```

In [ ]:

```
1
```

In [ ]:

```
1
```

In [ ]:

```
1
```

In [ ]:

```
1
```

In [ ]:

```
1
```