

Project1_App Rating Prediction

October 8, 2022

```
[1]: #App Rating Prediction
import pandas as pd
import numpy as np
import seaborn as sns
```

Load the data file using pandas.

```
[2]: data = pd.read_csv('googleplaystore.csv')
```

```
[3]: data.head()
```

```
[3]:
```

	App	Category	Rating \
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1
1	Coloring book moana	ART_AND_DESIGN	3.9
2	U Launcher Lite - FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3

	Reviews	Size	Installs	Type	Price	Content	Rating \
0	159	19M	10,000+	Free	0	Everyone	
1	967	14M	500,000+	Free	0	Everyone	
2	87510	8.7M	5,000,000+	Free	0	Everyone	
3	215644	25M	50,000,000+	Free	0	Teen	
4	967	2.8M	100,000+	Free	0	Everyone	

	Genres	Last Updated	Current Ver \
0	Art & Design	January 7, 2018	1.0.0
1	Art & Design;Pretend Play	January 15, 2018	2.0.0
2	Art & Design	August 1, 2018	1.2.4
3	Art & Design	June 8, 2018	Varies with device
4	Art & Design;Creativity	June 20, 2018	1.1

	Android Ver
0	4.0.3 and up
1	4.0.3 and up
2	4.0.3 and up
3	4.2 and up
4	4.4 and up

```
[4]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10841 entries, 0 to 10840
Data columns (total 13 columns):
#   Column                Non-Null Count  Dtype
---  -
0   App                    10841 non-null  object
1   Category               10841 non-null  object
2   Rating                 9367 non-null   float64
3   Reviews                10841 non-null  object
4   Size                   10841 non-null  object
5   Installs               10841 non-null  object
6   Type                   10840 non-null  object
7   Price                  10841 non-null  object
8   Content Rating         10840 non-null  object
9   Genres                 10841 non-null  object
10  Last Updated           10841 non-null  object
11  Current Ver            10833 non-null  object
12  Android Ver            10838 non-null  object
dtypes: float64(1), object(12)
memory usage: 1.1+ MB
```

```
[5]: data.shape
```

```
[5]: (10841, 13)
```

2 Check for null values in the data. Get the number of null values for each column.

```
[6]: data.isnull().any()
```

```
[6]: App                    False
Category                 False
Rating                   True
Reviews                  False
Size                     False
Installs                 False
Type                     True
Price                    False
Content Rating           True
Genres                   False
Last Updated             False
Current Ver              True
Android Ver              True
dtype: bool
```

```
[7]: data.isnull().sum()
```

```
[7]: App          0
      Category     0
      Rating      1474
      Reviews      0
      Size         0
      Installs     0
      Type         1
      Price        0
      Content Rating 1
      Genres       0
      Last Updated  0
      Current Ver   8
      Android Ver   3
      dtype: int64
```

3 Drop records with nulls in any of the columns.

```
[8]: data = data.dropna()
```

```
[9]: data.isnull().any()
```

```
[9]: App          False
      Category     False
      Rating      False
      Reviews      False
      Size         False
      Installs     False
      Type         False
      Price        False
      Content Rating False
      Genres       False
      Last Updated  False
      Current Ver   False
      Android Ver   False
      dtype: bool
```

```
[10]: data.shape
```

```
[10]: (9360, 13)
```

4.I Size column has sizes in Kb as well as Mb. To analyze, you'll need to convert these to numeric.

Extract the numeric value from the column

Multiply the value by 1,000, if size is mentioned in Mb

```
[11]: data["Size"] = [ float(i.split('M')[0]) if 'M' in i else float(0) for i in data["Size"] ]
```

```
[12]: data.head()
```

```
[12]:
```

	App	Category	Rating \
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1
1	Coloring book moana	ART_AND_DESIGN	3.9
2	U Launcher Lite - FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3

	Reviews	Size	Installs	Type	Price	Content Rating	\
0	159	19.0	10,000+	Free	0	Everyone	
1	967	14.0	500,000+	Free	0	Everyone	
2	87510	8.7	5,000,000+	Free	0	Everyone	
3	215644	25.0	50,000,000+	Free	0	Teen	
4	967	2.8	100,000+	Free	0	Everyone	

	Genres	Last Updated	Current Ver \
0	Art & Design	January 7, 2018	1.0.0
1	Art & Design;Pretend Play	January 15, 2018	2.0.0
2	Art & Design	August 1, 2018	1.2.4
3	Art & Design	June 8, 2018	Varies with device
4	Art & Design;Creativity	June 20, 2018	1.1

	Android Ver
0	4.0.3 and up
1	4.0.3 and up
2	4.0.3 and up
3	4.2 and up
4	4.4 and up

```
[13]: data["Size"] = 1000 * data["Size"]
```

```
[14]: data
```

```
[14]:
```

	App	Category \
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN
1	Coloring book moana	ART_AND_DESIGN
2	U Launcher Lite - FREE Live Cool Themes, Hide ...	ART_AND_DESIGN
3	Sketch - Draw & Paint	ART_AND_DESIGN
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN
...
10834	FR Calculator	FAMILY
10836	Sya9a Maroc - FR	FAMILY
10837	Fr. Mike Schmitz Audio Teachings	FAMILY
10839	The SCP Foundation DB fr nn5n	BOOKS_AND_REFERENCE
10840	iHoroscope - 2018 Daily Horoscope & Astrology	LIFESTYLE

	Rating	Reviews	Size	Installs	Type	Price	Content	Rating \
0	4.1	159	19000.0	10,000+	Free	0	Everyone	
1	3.9	967	14000.0	500,000+	Free	0	Everyone	
2	4.7	87510	8700.0	5,000,000+	Free	0	Everyone	
3	4.5	215644	25000.0	50,000,000+	Free	0	Teen	
4	4.3	967	2800.0	100,000+	Free	0	Everyone	
...	
10834	4.0	7	2600.0	500+	Free	0	Everyone	
10836	4.5	38	53000.0	5,000+	Free	0	Everyone	
10837	5.0	4	3600.0	100+	Free	0	Everyone	
10839	4.5	114	0.0	1,000+	Free	0	Mature 17+	
10840	4.5	398307	19000.0	10,000,000+	Free	0	Everyone	

	Genres	Last Updated	Current Ver \
0	Art & Design	January 7, 2018	1.0.0
1	Art & Design;Pretend Play	January 15, 2018	2.0.0
2	Art & Design	August 1, 2018	1.2.4
3	Art & Design	June 8, 2018	Varies with device
4	Art & Design;Creativity	June 20, 2018	1.1
...
10834	Education	June 18, 2017	1.0.0
10836	Education	July 25, 2017	1.48
10837	Education	July 6, 2018	1.0
10839	Books & Reference	January 19, 2015	Varies with device
10840	Lifestyle	July 25, 2018	Varies with device

	Android Ver
0	4.0.3 and up
1	4.0.3 and up
2	4.0.3 and up
3	4.2 and up
4	4.4 and up
...	...
10834	4.1 and up
10836	4.1 and up
10837	4.1 and up
10839	Varies with device
10840	Varies with device

[9360 rows x 13 columns]

4.II Reviews is a numeric field that is loaded as a string field. Convert it to numeric (int/float).

```
[15]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9360 entries, 0 to 10840
Data columns (total 13 columns):
```

#	Column	Non-Null Count	Dtype
0	App	9360 non-null	object
1	Category	9360 non-null	object
2	Rating	9360 non-null	float64
3	Reviews	9360 non-null	object
4	Size	9360 non-null	float64
5	Installs	9360 non-null	object
6	Type	9360 non-null	object
7	Price	9360 non-null	object
8	Content Rating	9360 non-null	object
9	Genres	9360 non-null	object
10	Last Updated	9360 non-null	object
11	Current Ver	9360 non-null	object
12	Android Ver	9360 non-null	object

dtypes: float64(2), object(11)
memory usage: 1023.8+ KB

```
[16]: data["Reviews"] = data["Reviews"].astype(float)
```

```
[17]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9360 entries, 0 to 10840
Data columns (total 13 columns):
#   Column          Non-Null Count  Dtype
---  -
0   App              9360 non-null   object
1   Category         9360 non-null   object
2   Rating           9360 non-null   float64
3   Reviews          9360 non-null   float64
4   Size             9360 non-null   float64
5   Installs         9360 non-null   object
6   Type             9360 non-null   object
7   Price            9360 non-null   object
8   Content Rating   9360 non-null   object
9   Genres           9360 non-null   object
10  Last Updated     9360 non-null   object
11  Current Ver      9360 non-null   object
12  Android Ver      9360 non-null   object
dtypes: float64(3), object(10)
memory usage: 1023.8+ KB
```

4.III Installs field is currently stored as string and has values like 1,000,000+.

```
[18]: data["Installs"] = [ float(i.replace('+','').replace(',',' ')) if '+' in i or
    ↪ ',' in i else float(0) for i in data["Installs"] ]
```

```
[19]: data.head()
```

```
[19]:
```

	App	Category	Rating \
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1
1	Coloring book moana	ART_AND_DESIGN	3.9
2	U Launcher Lite - FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3

	Reviews	Size	Installs	Type	Price	Content Rating	\
0	159.0	19000.0	10000.0	Free	0	Everyone	
1	967.0	14000.0	500000.0	Free	0	Everyone	
2	87510.0	8700.0	5000000.0	Free	0	Everyone	
3	215644.0	25000.0	50000000.0	Free	0	Teen	
4	967.0	2800.0	100000.0	Free	0	Everyone	

	Genres	Last Updated	Current Ver \
0	Art & Design	January 7, 2018	1.0.0
1	Art & Design;Pretend Play	January 15, 2018	2.0.0
2	Art & Design	August 1, 2018	1.2.4
3	Art & Design	June 8, 2018	Varies with device
4	Art & Design;Creativity	June 20, 2018	1.1

	Android Ver
0	4.0.3 and up
1	4.0.3 and up
2	4.0.3 and up
3	4.2 and up
4	4.4 and up

```
[20]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9360 entries, 0 to 10840
Data columns (total 13 columns):
#   Column          Non-Null Count  Dtype
---  -
0   App              9360 non-null   object
1   Category         9360 non-null   object
2   Rating           9360 non-null   float64
3   Reviews          9360 non-null   float64
4   Size             9360 non-null   float64
5   Installs         9360 non-null   float64
6   Type             9360 non-null   object
7   Price            9360 non-null   object
8   Content Rating   9360 non-null   object
9   Genres           9360 non-null   object
```

```

10 Last Updated      9360 non-null  object
11 Current Ver       9360 non-null  object
12 Android Ver       9360 non-null  object
dtypes: float64(4), object(9)
memory usage: 1023.8+ KB

```

```
[21]: data["Installs"] = data["Installs"].astype(int)
```

```
[22]: data.info()
```

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 9360 entries, 0 to 10840
Data columns (total 13 columns):
#   Column          Non-Null Count  Dtype
---  -
0   App              9360 non-null  object
1   Category         9360 non-null  object
2   Rating           9360 non-null  float64
3   Reviews          9360 non-null  float64
4   Size             9360 non-null  float64
5   Installs         9360 non-null  int64
6   Type             9360 non-null  object
7   Price            9360 non-null  object
8   Content Rating   9360 non-null  object
9   Genres           9360 non-null  object
10  Last Updated     9360 non-null  object
11  Current Ver      9360 non-null  object
12  Android Ver      9360 non-null  object
dtypes: float64(3), int64(1), object(9)
memory usage: 1023.8+ KB

```

4.IV Price field is a string and has \$ symbol. Remove '\$' sign, and convert it to numeric.

```
[23]: data['Price'] = [ float(i.split('$')[1]) if '$' in i else float(0) for i in
    ↪data['Price'] ]
```

```
[24]: data.head()
```

```

[24]:
           App          Category  Rating \
0  Photo Editor & Candy Camera & Grid & ScrapBook  ART_AND_DESIGN    4.1
1              Coloring book moana  ART_AND_DESIGN    3.9
2  U Launcher Lite - FREE Live Cool Themes, Hide ...  ART_AND_DESIGN    4.7
3              Sketch - Draw & Paint  ART_AND_DESIGN    4.5
4      Pixel Draw - Number Art Coloring Book  ART_AND_DESIGN    4.3

           Reviews      Size  Installs  Type  Price  Content Rating \
0          159.0  19000.0    10000  Free    0.0      Everyone
1          967.0  14000.0   500000  Free    0.0      Everyone

```


2	87510.0	8700.0	5000000	Free	0.0	Everyone
3	215644.0	25000.0	50000000	Free	0.0	Teen
4	967.0	2800.0	100000	Free	0.0	Everyone

	Genres	Last Updated	Current Ver \
0	Art & Design	January 7, 2018	1.0.0
1	Art & Design;Pretend Play	January 15, 2018	2.0.0
2	Art & Design	August 1, 2018	1.2.4
3	Art & Design	June 8, 2018	Varies with device
4	Art & Design;Creativity	June 20, 2018	1.1

	Android Ver
0	4.0.3 and up
1	4.0.3 and up
2	4.0.3 and up
3	4.2 and up
4	4.4 and up

```
[25]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9360 entries, 0 to 10840
Data columns (total 13 columns):
#   Column          Non-Null Count  Dtype
---  -
0   App              9360 non-null   object
1   Category         9360 non-null   object
2   Rating           9360 non-null   float64
3   Reviews          9360 non-null   float64
4   Size             9360 non-null   float64
5   Installs         9360 non-null   int64
6   Type             9360 non-null   object
7   Price            9360 non-null   float64
8   Content Rating   9360 non-null   object
9   Genres           9360 non-null   object
10  Last Updated     9360 non-null   object
11  Current Ver      9360 non-null   object
12  Android Ver      9360 non-null   object
dtypes: float64(4), int64(1), object(8)
memory usage: 1023.8+ KB
```

```
[26]: data["Price"] = data["Price"].astype(int)
```

```
[27]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9360 entries, 0 to 10840
```

Data columns (total 13 columns):

#	Column	Non-Null Count	Dtype
0	App	9360 non-null	object
1	Category	9360 non-null	object
2	Rating	9360 non-null	float64
3	Reviews	9360 non-null	float64
4	Size	9360 non-null	float64
5	Installs	9360 non-null	int64
6	Type	9360 non-null	object
7	Price	9360 non-null	int64
8	Content Rating	9360 non-null	object
9	Genres	9360 non-null	object
10	Last Updated	9360 non-null	object
11	Current Ver	9360 non-null	object
12	Android Ver	9360 non-null	object

dtypes: float64(3), int64(2), object(8)

memory usage: 1023.8+ KB

4.V.a Sanity checks:

Average rating should be between 1 and 5 as only these values are allowed on the play store. Drop the rows that have a value outside this range.

```
[28]: data.shape
```

```
[28]: (9360, 13)
```

```
[29]: data.drop(data[(data['Reviews'] < 1) & (data['Reviews'] > 5)].index, inplace =  
→ True)
```

```
[30]: data.shape
```

```
[30]: (9360, 13)
```

4.V.b Reviews should not be more than installs as only those who installed can review the app. If there are any such records, drop them.

```
[31]: data.shape
```

```
[31]: (9360, 13)
```

```
[32]: data.drop(data[data['Installs'] < data['Reviews']].index, inplace = True)
```

```
[33]: data.shape
```

```
[33]: (9353, 13)
```

4.V.c For free apps (type = “Free”), the price should not be >0. Drop any such rows.

```
[34]: data.shape
```

```
[34]: (9353, 13)
```

```
[35]: data.drop(data[(data['Type'] == 'Free') & (data['Price'] > 0)].index, inplace =  
↳ True)
```

```
[36]: data.shape
```

```
[36]: (9353, 13)
```

5.I Performing univariate analysis:

Boxplot for Price

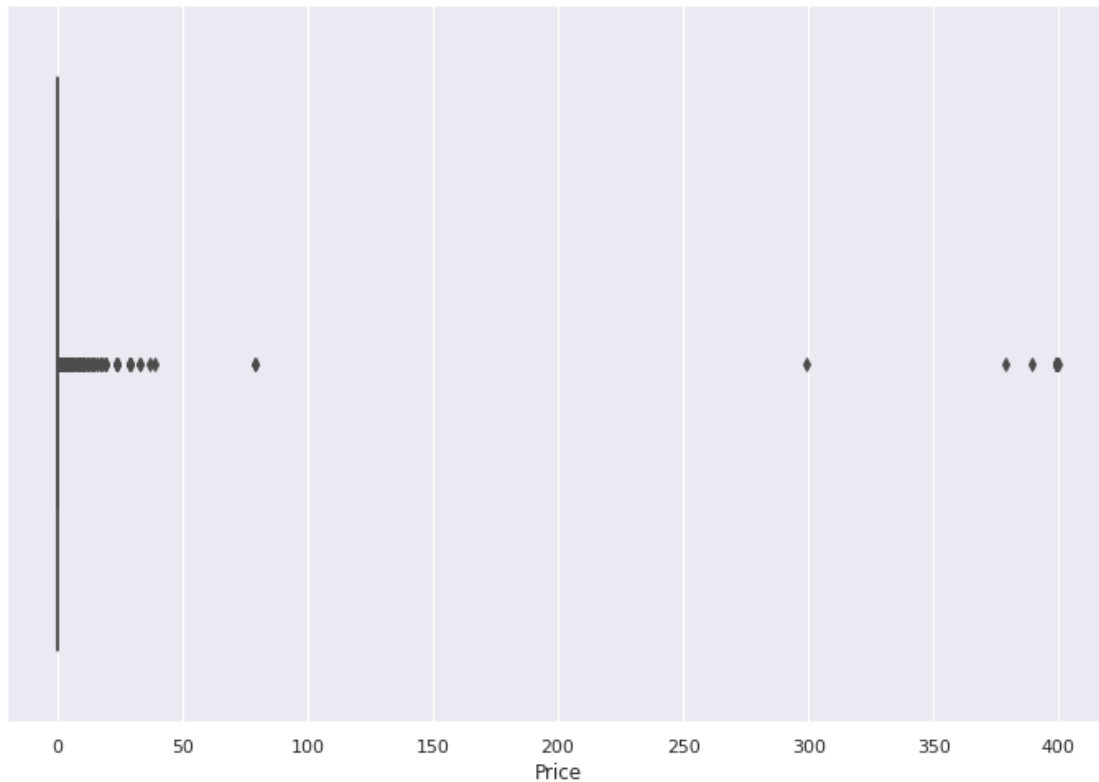
Are there any outliers? Think about the price of usual apps on Play Store.

```
[37]: sns.set(rc={'figure.figsize':(12,8)})
```

```
[38]: sns.boxplot(data['Price'])
```

```
/usr/local/lib/python3.7/site-packages/seaborn/_decorators.py:43: FutureWarning:  
Pass the following variable as a keyword arg: x. From version 0.12, the only  
valid positional argument will be `data`, and passing other arguments without an  
explicit keyword will result in an error or misinterpretation.  
FutureWarning
```

```
[38]: <AxesSubplot:xlabel='Price'>
```



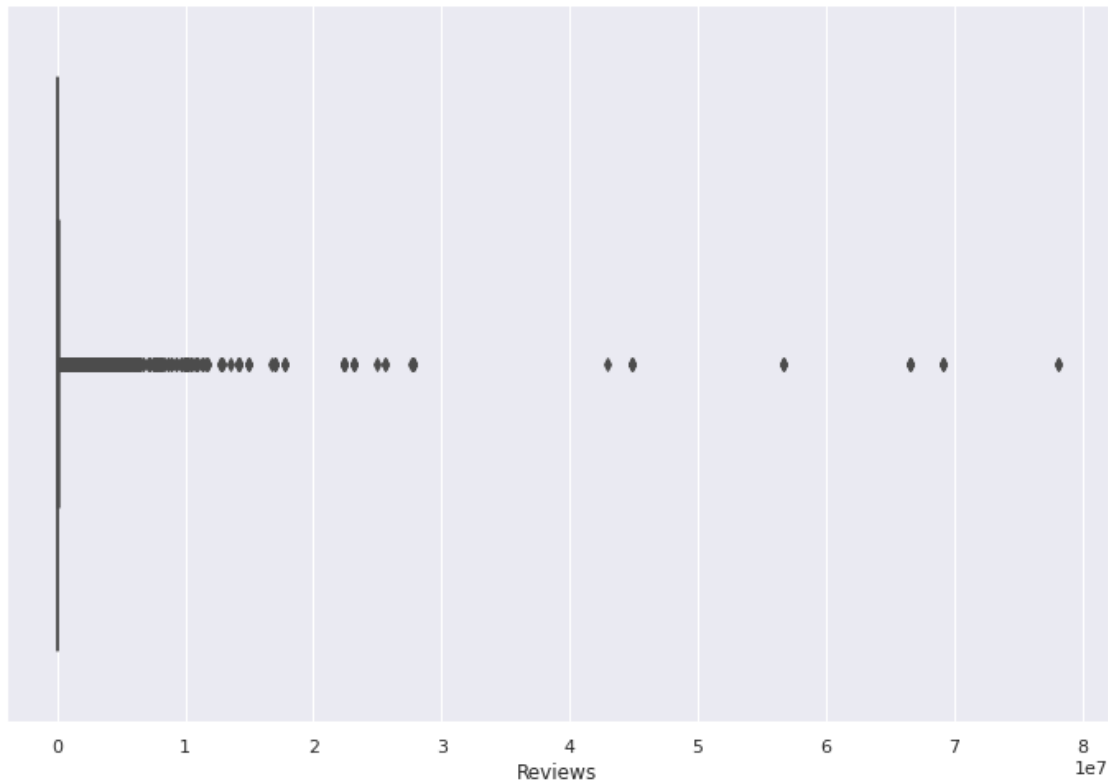
5.II Boxplot for Reviews

Are there any apps with very high number of reviews? Do the values seem right?

```
[39]: sns.boxplot(data['Reviews'])
```

```
/usr/local/lib/python3.7/site-packages/seaborn/_decorators.py:43: FutureWarning:  
Pass the following variable as a keyword arg: x. From version 0.12, the only  
valid positional argument will be `data`, and passing other arguments without an  
explicit keyword will result in an error or misinterpretation.  
FutureWarning
```

```
[39]: <AxesSubplot:xlabel='Reviews'>
```



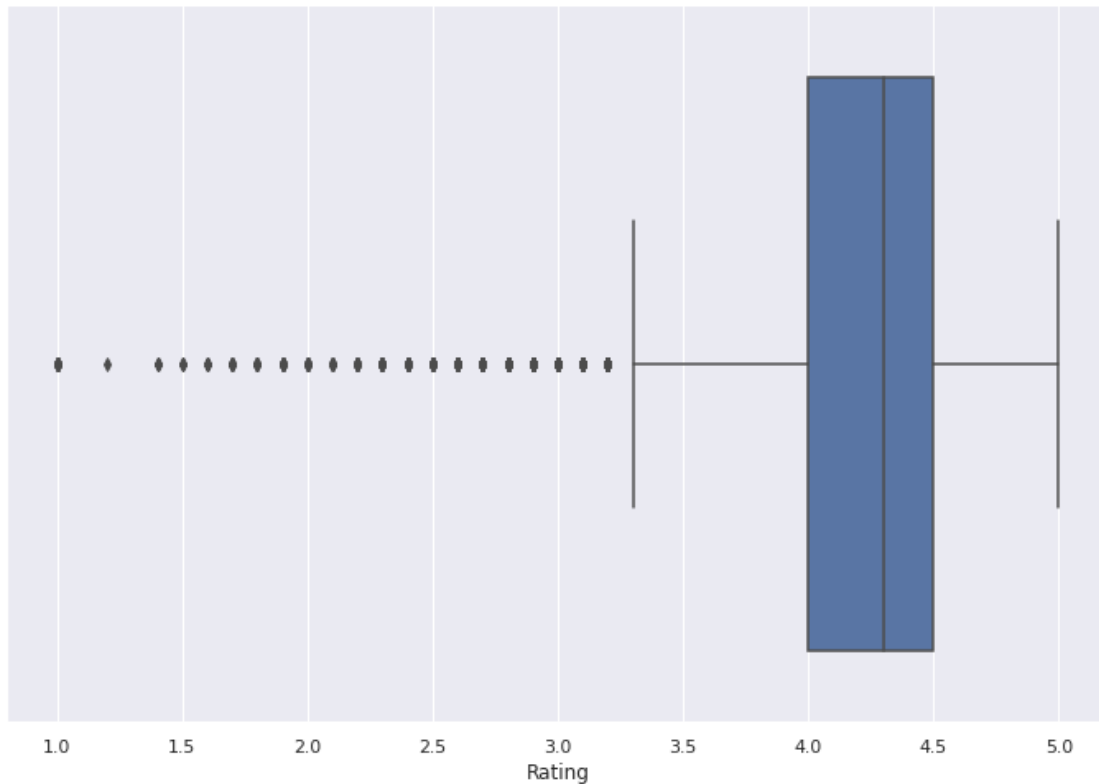
5.III Histogram for Rating

How are the ratings distributed? Is it more toward higher ratings?

```
[40]: sns.boxplot(data['Rating'])
```

```
/usr/local/lib/python3.7/site-packages/seaborn/_decorators.py:43: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
FutureWarning
```

```
[40]: <AxesSubplot:xlabel='Rating'>
```



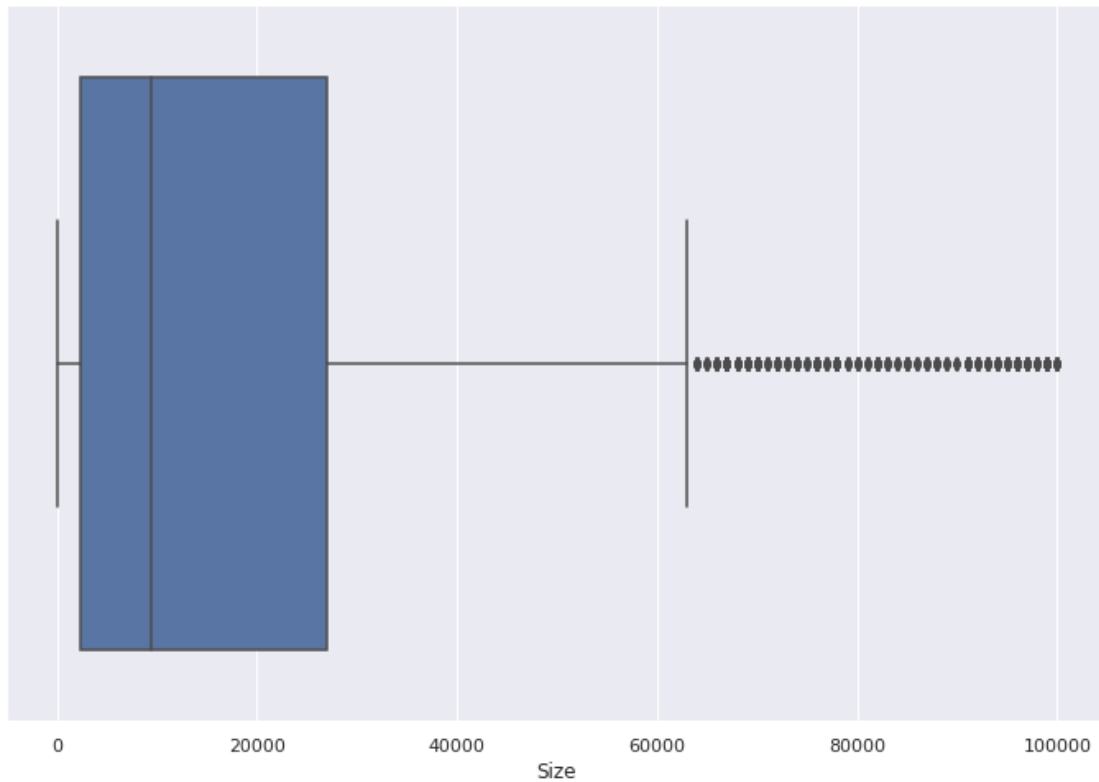
5.IV Histogram for Size

```
[41]: sns.boxplot(data['Size'])
```

/usr/local/lib/python3.7/site-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning

```
[41]: <AxesSubplot:xlabel='Size'>
```



6.I Outlier treatment:

Price: From the box plot, it seems like there are some apps with very high price. A price of \$200 for an application on the Play Store is very high and suspicious!

Check out the records with very high price

Is 200 indeed a high price?

```
[42]: more = data.apply(lambda x : True
                        if x['Price'] > 200 else False, axis = 1)
```

```
[43]: more_count = len(more[more == True].index)
```

```
[44]: data.shape
```

```
[44]: (9353, 13)
```

```
[45]: data.drop(data[data['Price'] > 200].index, inplace = True)
```

```
[46]: data.shape
```

```
[46]: (9338, 13)
```

6.II Reviews: Very few apps have very high number of reviews. These are all star apps that don't help with the analysis and, in fact, will skew it. Drop records having more than 2 million reviews.

```
[47]: data.drop(data[data['Reviews'] > 2000000].index, inplace = True)
```

```
[48]: data.shape
```

```
[48]: (8885, 13)
```

6.III Installs: There seems to be some outliers in this field too. Apps having very high number of installs should be dropped from the analysis.

Find out the different percentiles – 10, 25, 50, 70, 90, 95, 99

Decide a threshold as cutoff for outlier and drop records having values more than that

```
[49]: data.quantile([.1, .25, .5, .70, .90, .95, .99], axis = 0)
```

```
[49]:
```

	Rating	Reviews	Size	Installs	Price
0.10	3.5	18.00	0.0	1000.0	0.0
0.25	4.0	159.00	2600.0	10000.0	0.0
0.50	4.3	4290.00	9500.0	500000.0	0.0
0.70	4.5	35930.40	23000.0	1000000.0	0.0
0.90	4.7	296771.00	50000.0	10000000.0	0.0
0.95	4.8	637298.00	68000.0	10000000.0	1.0
0.99	5.0	1462800.88	95000.0	100000000.0	7.0

```
[50]: # dropping more than 10000000 Installs value
data.drop(data[data['Installs'] > 10000000].index, inplace = True)
```

```
[51]: data.shape
```

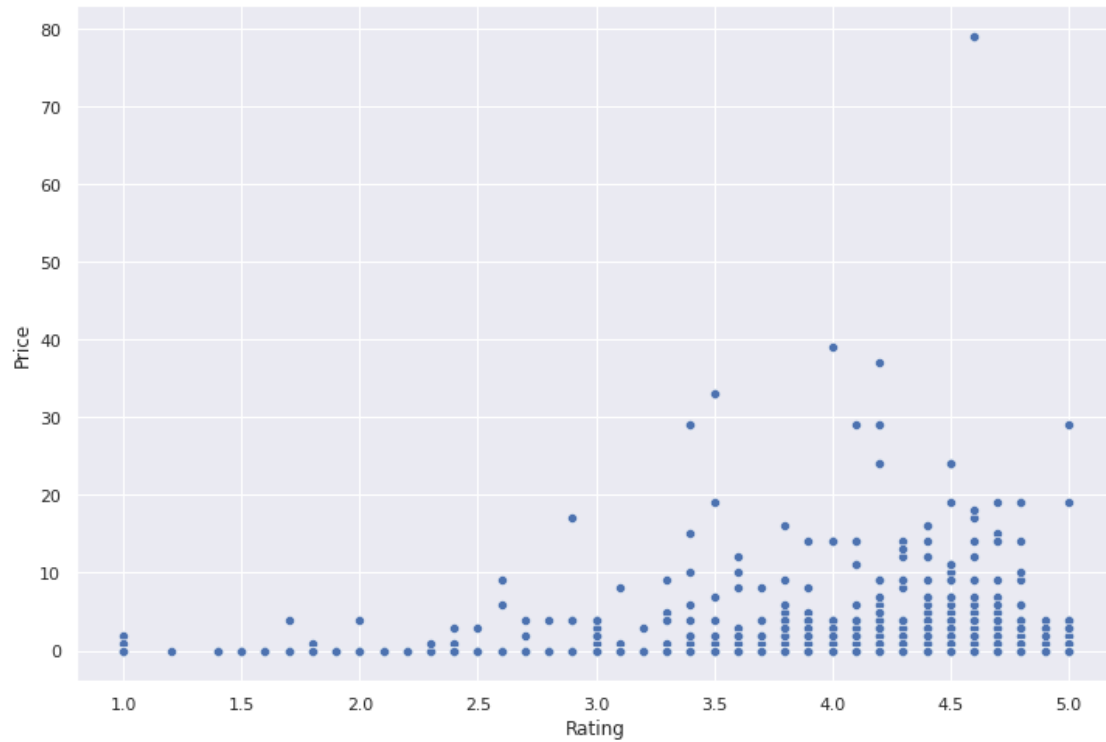
```
[51]: (8496, 13)
```

7.I Bivariate analysis Make scatter plot/joinplot for Rating vs. Price

What pattern do you observe? Does rating increase with price?

```
[52]: sns.scatterplot(x='Rating',y='Price',data=data)
```

```
[52]: <AxesSubplot:xlabel='Rating', ylabel='Price'>
```

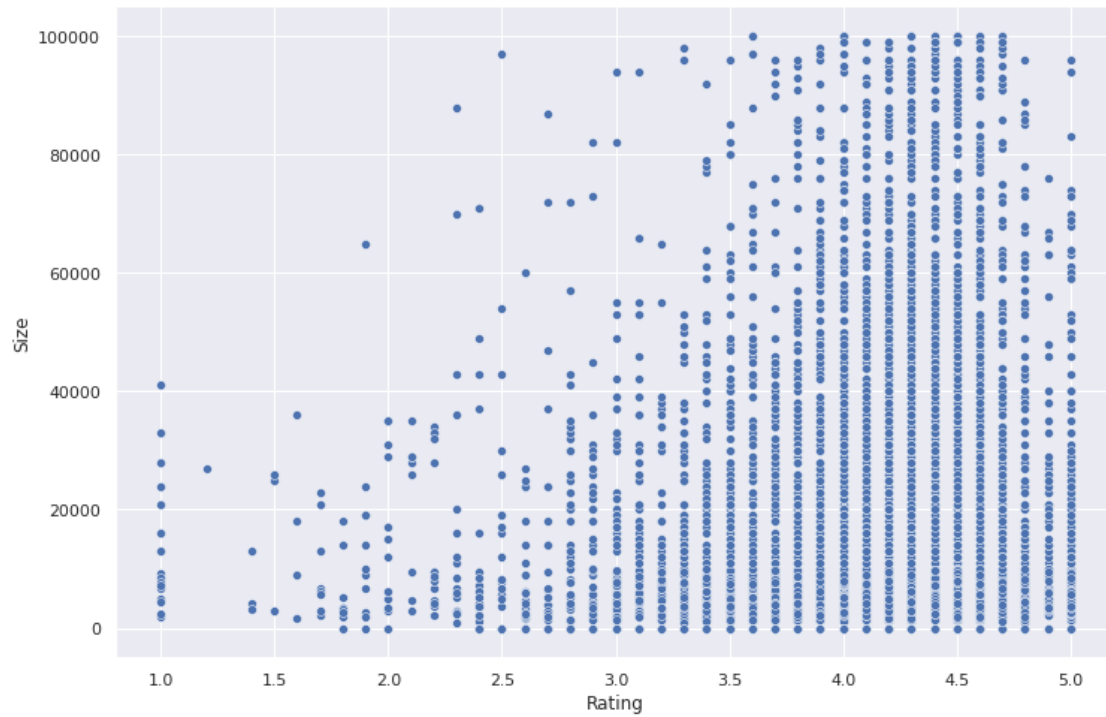
Yes, Paid apps are higher ratings compared to free apps.

7.II Make scatter plot/joinplot for Rating vs. Size

Are heavier apps rated better?

```
[53]: sns.scatterplot(x='Rating',y='Size',data=data)
```

```
[53]: <AxesSubplot:xlabel='Rating', ylabel='Size'>
```



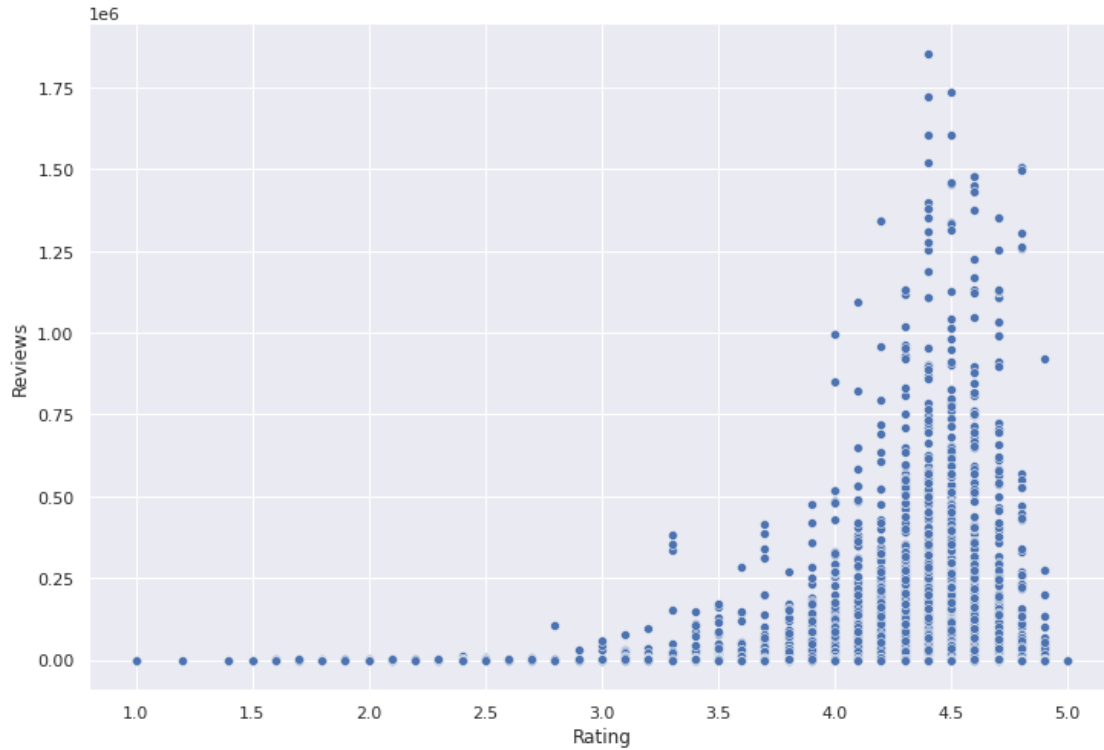
Yes it is clear that heavier apps are rated better.

7.III Make scatter plot/joinplot for Rating vs. Reviews

Does more review mean a better rating always?

```
[54]: sns.scatterplot(x='Rating',y='Reviews',data=data)
```

```
[54]: <AxesSubplot:xlabel='Rating', ylabel='Reviews'>
```



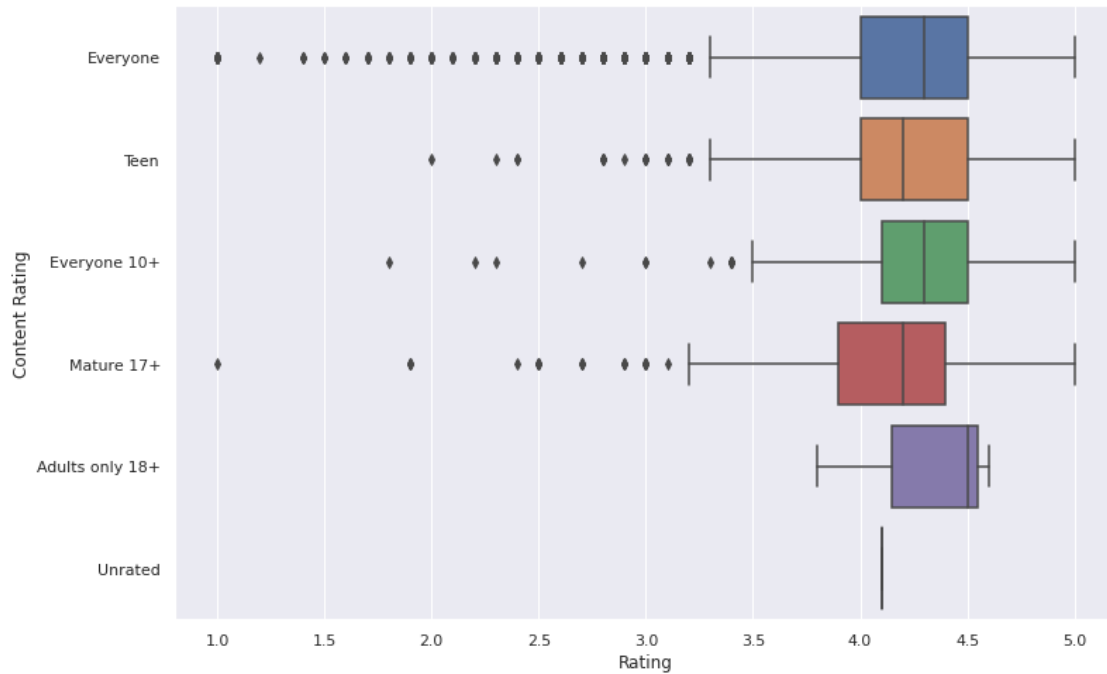
It is cristal clear that more reviews makes app rating better.

7.IV Make boxplot for Rating vs. Content Rating

Is there any difference in the ratings? Are some types liked better?

```
[55]: sns.boxplot(x="Rating", y="Content Rating", data=data)
```

```
[55]: <AxesSubplot:xlabel='Rating', ylabel='Content Rating'>
```



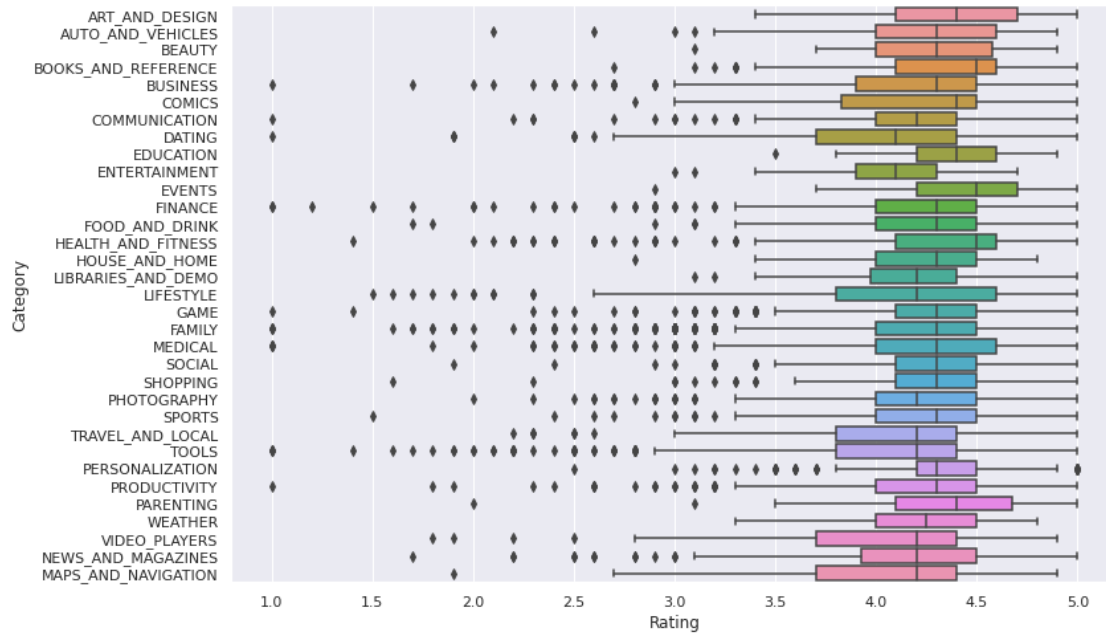
Apps which are for everyone has more bad ratings compare to other sections as it has so much outliers value, while 18+ apps have better ratings.

7.V Make boxplot for Ratings vs. Category

Which genre has the best ratings

```
[56]: sns.boxplot(x="Rating", y="Category", data=data)
```

```
[56]: <AxesSubplot:xlabel='Rating', ylabel='Category'>
```



Events category has best ratings compare to others.

8.I Data preprocessing Reviews and Install have some values that are still relatively very high. Before building a linear regression model, you need to reduce the skew. Apply log transformation (np.log1p) to Reviews and Installs.

```
[57]: inp1 = data
```

```
[58]: inp1.head()
```

```
[58]:
```

	App	Category	Rating \
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1
1	Coloring book moana	ART_AND_DESIGN	3.9
2	U Launcher Lite - FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3
5	Paper flowers instructions	ART_AND_DESIGN	4.4

	Reviews	Size	Installs	Type	Price	Content	Rating \
0	159.0	19000.0	10000	Free	0	Everyone	
1	967.0	14000.0	500000	Free	0	Everyone	
2	87510.0	8700.0	5000000	Free	0	Everyone	
4	967.0	2800.0	100000	Free	0	Everyone	
5	167.0	5600.0	50000	Free	0	Everyone	

	Genres	Last Updated	Current Ver	Android Ver
0	Art & Design	January 7, 2018	1.0.0	4.0.3 and up
1	Art & Design;Pretend Play	January 15, 2018	2.0.0	4.0.3 and up

2	Art & Design	August 1, 2018	1.2.4	4.0.3 and up
4	Art & Design;Creativity	June 20, 2018	1.1	4.4 and up
5	Art & Design	March 26, 2017	1.0	2.3 and up

```
[59]: inp1.skew()
```

```
[59]: Rating      -1.749753
Reviews      4.576494
Size         1.655917
Installs     1.543697
Price       18.074542
dtype: float64
```

```
[62]: installsskew = np.log1p(inp1['Installs'])
inp1['Installs']
```

```
[62]: 0          10000
1          500000
2         5000000
4          100000
5           50000
...
10834         500
10836        5000
10837         100
10839        1000
10840       10000000
Name: Installs, Length: 8496, dtype: int64
```

```
[63]: installsskew.skew()
```

```
[63]: -0.5097286542754812
```

```
[64]: inp1.head()
```

```
[64]:
```

	App	Category	Rating	\
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	
1	Coloring book moana	ART_AND_DESIGN	3.9	
2	U Launcher Lite - FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	
5	Paper flowers instructions	ART_AND_DESIGN	4.4	

	Reviews	Size	Installs	Type	Price	Content	Rating	\
0	159.0	19000.0	10000	Free	0		Everyone	
1	967.0	14000.0	500000	Free	0		Everyone	
2	87510.0	8700.0	5000000	Free	0		Everyone	
4	967.0	2800.0	100000	Free	0		Everyone	

5	167.0	5600.0	50000	Free	0	Everyone
---	-------	--------	-------	------	---	----------

	Genres	Last Updated	Current Ver	Android Ver
0	Art & Design	January 7, 2018	1.0.0	4.0.3 and up
1	Art & Design;Pretend Play	January 15, 2018	2.0.0	4.0.3 and up
2	Art & Design	August 1, 2018	1.2.4	4.0.3 and up
4	Art & Design;Creativity	June 20, 2018	1.1	4.4 and up
5	Art & Design	March 26, 2017	1.0	2.3 and up

8.II Drop columns App, Last Updated, Current Ver, and Android Ver. These variables are not useful for our task.

```
[65]: inp1.drop(["Last Updated", "Current Ver", "Android_Ver", "App", "Type"], axis=1, inplace=True)
```

```
[66]: inp1.head()
```

```
[66]:
```

	Category	Rating	Reviews	Size	Installs	Price	Content Rating	\
0	ART_AND_DESIGN	4.1	159.0	19000.0	10000	0	Everyone	
1	ART_AND_DESIGN	3.9	967.0	14000.0	500000	0	Everyone	
2	ART_AND_DESIGN	4.7	87510.0	8700.0	5000000	0	Everyone	
4	ART_AND_DESIGN	4.3	967.0	2800.0	100000	0	Everyone	
5	ART_AND_DESIGN	4.4	167.0	5600.0	50000	0	Everyone	

	Genres
0	Art & Design
1	Art & Design;Pretend Play
2	Art & Design
4	Art & Design;Creativity
5	Art & Design

```
[67]: inp1.shape
```

```
[67]: (8496, 8)
```

8.III Get dummy columns for Category, Genres, and Content Rating. This needs to be done as the models do not understand categorical data, and all data should be numeric. Dummy encoding is one way to convert character fields to numeric. Name of dataframe should be inp2.

```
[68]: inp2 = inp1
```

```
[69]: inp2.head()
```

```
[69]:
```

	Category	Rating	Reviews	Size	Installs	Price	Content Rating	\
0	ART_AND_DESIGN	4.1	159.0	19000.0	10000	0	Everyone	
1	ART_AND_DESIGN	3.9	967.0	14000.0	500000	0	Everyone	
2	ART_AND_DESIGN	4.7	87510.0	8700.0	5000000	0	Everyone	
4	ART_AND_DESIGN	4.3	967.0	2800.0	100000	0	Everyone	

5	ART_AND_DESIGN	4.4	167.0	5600.0	50000	0	Everyone
---	----------------	-----	-------	--------	-------	---	----------

	Genres
0	Art & Design
1	Art & Design;Pretend Play
2	Art & Design
4	Art & Design;Creativity
5	Art & Design

Let's apply Dummy EnCoding on Column "Category"

```
[70]: #get unique values in Column "Category"
inp2.Category.unique()
```

```
[70]: array(['ART_AND_DESIGN', 'AUTO_AND_VEHICLES', 'BEAUTY',
        'BOOKS_AND_REFERENCE', 'BUSINESS', 'COMICS', 'COMMUNICATION',
        'DATING', 'EDUCATION', 'ENTERTAINMENT', 'EVENTS', 'FINANCE',
        'FOOD_AND_DRINK', 'HEALTH_AND_FITNESS', 'HOUSE_AND_HOME',
        'LIBRARIES_AND_DEMO', 'LIFESTYLE', 'GAME', 'FAMILY', 'MEDICAL',
        'SOCIAL', 'SHOPPING', 'PHOTOGRAPHY', 'SPORTS', 'TRAVEL_AND_LOCAL',
        'TOOLS', 'PERSONALIZATION', 'PRODUCTIVITY', 'PARENTING', 'WEATHER',
        'VIDEO_PLAYERS', 'NEWS_AND_MAGAZINES', 'MAPS_AND_NAVIGATION'],
        dtype=object)
```

```
[71]: inp2.Category = pd.Categorical(inp2.Category)

x = inp2[['Category']]
del inp2['Category']

dummies = pd.get_dummies(x, prefix = 'Category')
inp2 = pd.concat([inp2,dummies], axis=1)
inp2.head()
```

	Rating	Reviews	Size	Installs	Price	Content Rating	\
0	4.1	159.0	19000.0	10000	0	Everyone	
1	3.9	967.0	14000.0	500000	0	Everyone	
2	4.7	87510.0	8700.0	5000000	0	Everyone	
4	4.3	967.0	2800.0	100000	0	Everyone	
5	4.4	167.0	5600.0	50000	0	Everyone	

	Genres	Category_ART_AND_DESIGN	\
0	Art & Design		1
1	Art & Design;Pretend Play		1
2	Art & Design		1
4	Art & Design;Creativity		1
5	Art & Design		1

	Category_AUTO_AND_VEHICLES	Category_BEAUTY	...	Category_PERSONALIZATION	\
0	0	0	...	0	
1	0	0	...	0	
2	0	0	...	0	
4	0	0	...	0	
5	0	0	...	0	

	Category_PHOTOGRAPHY	Category_PRODUCTIVITY	Category_SHOPPING	\
0	0	0	0	
1	0	0	0	
2	0	0	0	
4	0	0	0	
5	0	0	0	

	Category_SOCIAL	Category_SPORTS	Category_TOOLS	\
0	0	0	0	
1	0	0	0	
2	0	0	0	
4	0	0	0	
5	0	0	0	

	Category_TRAVEL_AND_LOCAL	Category_VIDEO_PLAYERS	Category_WEATHER
0	0	0	0
1	0	0	0
2	0	0	0
4	0	0	0
5	0	0	0

[5 rows x 40 columns]

```
[72]: inp2.shape
```

```
[72]: (8496, 40)
```

Let's apply Dummy EnCoding on Column "Genres"

```
[73]: #get unique values in Column "Genres"
inp2["Genres"].unique()
```

```
[73]: array(['Art & Design', 'Art & Design;Pretend Play',
        'Art & Design;Creativity', 'Auto & Vehicles', 'Beauty',
        'Books & Reference', 'Business', 'Comics', 'Comics;Creativity',
        'Communication', 'Dating', 'Education', 'Education;Creativity',
        'Education;Education', 'Education;Music & Video',
        'Education;Action & Adventure', 'Education;Pretend Play',
        'Education;Brain Games', 'Entertainment',
        'Entertainment;Brain Games', 'Entertainment;Creativity',
```

```

'Entertainment;Music & Video', 'Events', 'Finance', 'Food & Drink',
'Health & Fitness', 'House & Home', 'Libraries & Demo',
'Lifestyle', 'Lifestyle;Pretend Play', 'Card', 'Casual', 'Puzzle',
'Action', 'Arcade', 'Word', 'Racing', 'Casual;Creativity',
'Sports', 'Board', 'Simulation', 'Role Playing', 'Adventure',
'Strategy', 'Simulation;Education', 'Action;Action & Adventure',
'Trivia', 'Casual;Brain Games', 'Simulation;Action & Adventure',
'Educational;Creativity', 'Puzzle;Brain Games',
'Educational;Education', 'Card;Brain Games',
'Educational;Brain Games', 'Educational;Pretend Play',
'Casual;Action & Adventure', 'Entertainment;Education',
'Casual;Education', 'Casual;Pretend Play', 'Music;Music & Video',
'Racing;Action & Adventure', 'Arcade;Pretend Play',
'Adventure;Action & Adventure', 'Role Playing;Action & Adventure',
'Simulation;Pretend Play', 'Puzzle;Creativity',
'Sports;Action & Adventure', 'Educational;Action & Adventure',
'Arcade;Action & Adventure', 'Entertainment;Action & Adventure',
'Puzzle;Action & Adventure', 'Strategy;Action & Adventure',
'Music & Audio;Music & Video', 'Health & Fitness;Education',
'Adventure;Education', 'Board;Brain Games',
'Board;Action & Adventure', 'Board;Pretend Play',
'Casual;Music & Video', 'Role Playing;Pretend Play',
'Entertainment;Pretend Play', 'Video Players & Editors;Creativity',
'Card;Action & Adventure', 'Medical', 'Social', 'Shopping',
'Photography', 'Travel & Local',
'Travel & Local;Action & Adventure', 'Tools', 'Tools;Education',
'Personalization', 'Productivity', 'Parenting',
'Parenting;Music & Video', 'Parenting;Brain Games',
'Parenting;Education', 'Weather', 'Video Players & Editors',
'Video Players & Editors;Music & Video', 'News & Magazines',
'Maps & Navigation', 'Health & Fitness;Action & Adventure',
'Music', 'Educational', 'Casino', 'Adventure;Brain Games',
'Lifestyle;Education', 'Books & Reference;Education',
'Puzzle;Education', 'Role Playing;Brain Games',
'Strategy;Education', 'Racing;Pretend Play',
'Communication;Creativity', 'Strategy;Creativity'], dtype=object)

```

Since, There are too many categories under Genres. Hence, we will try to reduce some categories which have very few samples under them and put them under one new common category i.e. "Other".

```

[74]: lists = []
      for i in inp2.Genres.value_counts().index:
          if inp2.Genres.value_counts()[i]<20:
              lists.append(i)
      inp2.Genres = ['Other' if i in lists else i for i in inp2.Genres]

```

```
[75]: inp2["Genres"].unique()
```

```
[75]: array(['Art & Design', 'Other', 'Auto & Vehicles', 'Beauty',
        'Books & Reference', 'Business', 'Comics', 'Communication',
        'Dating', 'Education', 'Education;Education',
        'Education;Pretend Play', 'Entertainment',
        'Entertainment;Music & Video', 'Events', 'Finance', 'Food & Drink',
        'Health & Fitness', 'House & Home', 'Libraries & Demo',
        'Lifestyle', 'Card', 'Casual', 'Puzzle', 'Action', 'Arcade',
        'Word', 'Racing', 'Sports', 'Board', 'Simulation', 'Role Playing',
        'Adventure', 'Strategy', 'Trivia', 'Educational;Education',
        'Casual;Pretend Play', 'Medical', 'Social', 'Shopping',
        'Photography', 'Travel & Local', 'Tools', 'Personalization',
        'Productivity', 'Parenting', 'Weather', 'Video Players & Editors',
        'News & Magazines', 'Maps & Navigation', 'Educational', 'Casino'],
        dtype=object)
```

```
[76]: inp2.Genres = pd.Categorical(inp2['Genres'])
x = inp2[["Genres"]]
del inp2['Genres']
dummies = pd.get_dummies(x, prefix = 'Genres')
inp2 = pd.concat([inp2,dummies], axis=1)
```

```
[77]: inp2.head()
```

```
[77]:
```

	Rating	Reviews	Size	Installs	Price	Content Rating	\
0	4.1	159.0	19000.0	10000	0	Everyone	
1	3.9	967.0	14000.0	500000	0	Everyone	
2	4.7	87510.0	8700.0	5000000	0	Everyone	
4	4.3	967.0	2800.0	100000	0	Everyone	
5	4.4	167.0	5600.0	50000	0	Everyone	

	Category_ART_AND_DESIGN	Category_AUTO_AND_VEHICLES	Category_BEAUTY	\
0	1		0	0
1	1		0	0
2	1		0	0
4	1		0	0
5	1		0	0

	Category_BOOKS_AND_REFERENCE	...	Genres_Simulation	Genres_Social	\
0	0	...	0	0	
1	0	...	0	0	
2	0	...	0	0	
4	0	...	0	0	
5	0	...	0	0	

	Genres_Sports	Genres_Strategy	Genres_Tools	Genres_Travel & Local	\
--	---------------	-----------------	--------------	-----------------------	---

0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
4	0	0	0	0
5	0	0	0	0

	Genres_Trivia	Genres_Video	Players & Editors	Genres_Weather	Genres_Word
0	0		0	0	0
1	0		0	0	0
2	0		0	0	0
4	0		0	0	0
5	0		0	0	0

[5 rows x 91 columns]

```
[78]: inp2.shape
```

```
[78]: (8496, 91)
```

Let's apply Dummy EnCoding on Column "Content Rating"

```
[79]: #get unique values in Column "Content Rating"
inp2["Content Rating"].unique()
```

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

```
[80]: inp2['Content Rating'] = pd.Categorical(inp2['Content Rating'])

x = inp2[['Content Rating']]
del inp2['Content Rating']

dummies = pd.get_dummies(x, prefix = 'Content Rating')
inp2 = pd.concat([inp2,dummies], axis=1)
inp2.head()
```

```
[80]:
```

	Rating	Reviews	Size	Installs	Price	Category_ART_AND_DESIGN \
0	4.1	159.0	19000.0	10000	0	1
1	3.9	967.0	14000.0	500000	0	1
2	4.7	87510.0	8700.0	5000000	0	1
4	4.3	967.0	2800.0	100000	0	1
5	4.4	167.0	5600.0	50000	0	1

	Category_AUTO_AND_VEHICLES	Category_BEAUTY	Category_BOOKS_AND_REFERENCE \
0	0	0	0
1	0	0	0
2	0	0	0

4		0	0	0
5		0	0	0

	Category_BUSINESS	...	Genres_Trivia	Genres_Video	Players & Editors	\
0	0	...	0			0
1	0	...	0			0
2	0	...	0			0
4	0	...	0			0
5	0	...	0			0

	Genres_Weather	Genres_Word	Content Rating_Adults	only 18+	\
0	0	0		0	
1	0	0		0	
2	0	0		0	
4	0	0		0	
5	0	0		0	

	Content Rating_Everyone	Content Rating_Everyone	10+	\
0		1	0	
1		1	0	
2		1	0	
4		1	0	
5		1	0	

	Content Rating_Mature	17+	Content Rating_Teen	Content Rating_Unrated
0		0	0	0
1		0	0	0
2		0	0	0
4		0	0	0
5		0	0	0

[5 rows x 96 columns]

```
[81]: inp2.shape
```

```
[81]: (8496, 96)
```

9. Train test split and apply 70-30 split. Name the new dataframes df_train and df_test.

10. Separate the dataframes into X_train, y_train, X_test, and y_test.

```
[82]: from sklearn.model_selection import train_test_split as tts
from sklearn.linear_model import LinearRegression as LR
from sklearn.metrics import mean_squared_error as mse
```

```
[83]: d1 = inp2
X = d1.drop('Rating',axis=1)
y = d1['Rating']
```

```
Xtrain, Xtest, ytrain, ytest = tts(X,y, test_size=0.3, random_state=5)
```

11 . Model building

Use linear regression as the technique

Report the R2 on the train set

```
[84]: reg_all = LR()  
      reg_all.fit(Xtrain,ytrain)
```

```
[84]: LinearRegression()
```

```
[85]: R2_train = round(reg_all.score(Xtrain,ytrain),3)  
      print("The R2 value of the Training Set is : {}".format(R2_train))
```

The R2 value of the Training Set is : 0.066

12. Make predictions on test set and report R2.

```
[86]: R2_test = round(reg_all.score(Xtest,ytest),3)  
      print("The R2 value of the Testing Set is : {}".format(R2_test))
```

The R2 value of the Testing Set is : 0.054