Rating Prediction-Project

January 14, 2023

1 DESCRIPTION

1.1 Objective: Make a model to predict the app rating, with other information about the app provided.

1.1.1 Problem Statement:

Google Play Store team is about to launch a new feature wherein, certain apps that are promising, are boosted in visibility. The boost will manifest in multiple ways including higher priority in recommendations sections ("Similar apps", "You might also like", "New and updated games"). These will also get a boost in search results visibility. This feature will help bring more attention to newer apps that have the potential.

1.1.2 Domain: General

Analysis to be done: The problem is to identify the apps that are going to be good for Google to promote. App ratings, which are provided by the customers, is always a great indicator of the goodness of the app. The problem reduces to: predict which apps will have high ratings.

1.1.3 Fields in the data –

`App: Application name`

`Category: Category to which the app belongs`

`Rating: Overall user rating of the app`

`Reviews: Number of user reviews for the app`

`Size: Size of the app`

`Installs: Number of user downloads/installs for the app`

`Type: Paid or Free`

`Price: Price of the app`

Content Rating: Age group the app is targeted at - Children / Mature 21+ / Adult

Genres: An app can belong to multiple genres (apart from its main category). For example, a mus

Last Updated: Date when the app was last updated on Play Store

Current Ver: Current version of the app available on Play Store

Android Ver: Minimum required Android version

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt, seaborn as sns
%matplotlib inline

import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
```

2 Steps to perform:

2.1 1. Load the data file using pandas.

[2]:
```
df = pd.read_csv("googleplaystore.csv")
```

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

[3]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Everyone	Art & Design	January 7, 2018	1.0.0
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Everyone	Art & Design;Pretend Play	January 15, 2018	2.0.0
2	U Launcher Lite - FREE Live Cool Themes, Hide …	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free	0	Everyone	Art & Design	August 1, 2018	1.2.4
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0	Teen			
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0	Everyone			

```
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]:
```
df.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        10840 non-null  object
 2   Rating          9367 non-null   float64
 3   Reviews         10841 non-null  int64
 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  10841 non-null  object
 9   Genres          10840 non-null  object
 10  Last Updated    10841 non-null  object
 11  Current Ver     10833 non-null  object
 12  Android Ver     10839 non-null  object
dtypes: float64(1), int64(1), object(11)
memory usage: 1.1+ MB
```

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

Dropping the records with null ratings - this is done because ratings is our target variable

[5]:
```
df.isnull().sum()
```

[5]:
```
App                0
Category           1
Rating          1474
Reviews            0
Size               0
Installs           0
Type               1
```

```
Price              0
Content Rating     0
Genres             1
Last Updated       0
Current Ver        8
Android Ver        2
dtype: int64
```

2.3 3. Drop records with nulls in any of the columns.

[6]:
```
df.dropna(how = 'any', inplace = True)
```

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

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

Confirming that the null records have been dropped

Change variable to correct types

[8]:
```
df.dtypes
```

[8]:
```
App               object
Category          object
Rating           float64
Reviews            int64
Size              object
Installs          object
Type              object
Price             object
Content Rating    object
Genres            object
Last Updated      object
```

```
Current Ver         object
Android Ver         object
dtype: object
```

df.head()

2.4 4. Variables seem to have incorrect type and inconsistent formatting. You need to fix them:

1. Size column has sizes in Kb as well as Mb. To analyze, you'll need to convert these to numeric. a. Extract the numeric value from the column b. Multiply the value by 1,000, if size is mentioned in Mb 2. Reviews is a numeric field that is loaded as a string field. Convert it to numeric (int/float). 3. Installs field is currently stored as string and has values like 1,000,000+. a. Treat 1,000,000+ as 1,000,000 b. remove '+', ',' from the field, convert it to integer 4. Price field is a string and has *symbol.Remove*'' sign, and convert it to numeric.

4.4 Price column needs to be cleaned

[9]:
```
df.Price.value_counts()[:5]
```

[9]:
```
0        8715
$2.99     114
$0.99     106
$4.99      70
$1.99      59
Name: Price, dtype: int64
```

Some have dollars, some have 0 - we need to conditionally handle this - first, let's modify the column to take 0 if value is 0, else take the first letter onwards

[10]:
```
# Write a function named 'clean_price' if price is 0 it remains 0 otherwise␣
 ↪delete the $

# delete the $ = removing the element at index 0

'$2.99'[1:] # example
float('$2.99'[1:])

# use map to apply the function to the column as shown in the last line no. 9

# inp0['Price'] = inp0.Price.map(clean_price)

def clean_price(x):
    if x.startswith('$'):
        x = x[1:]
        return x
    else:
```

```
        return x

df['Price'] = df.Price.map(clean_price)
```

[11]:
```
df.Price.value_counts()[:5]
```

[11]:
```
0       8715
2.99     114
0.99     106
4.99      70
1.99      59
Name: Price, dtype: int64
```

[12]:
```
df['Price'] = df['Price'].astype('float')
```

4.2 Converting reviews to numeric

[13]:
```
# use astype("int32")
```

[14]:
```
df['Reviews'] = df['Reviews'].astype('float')
```

[15]:
```
df.Reviews.describe()
```

[15]:
```
count    9.360000e+03
mean     5.143767e+05
std      3.145023e+06
min      1.000000e+00
25%      1.867500e+02
50%      5.955000e+03
75%      8.162750e+04
max      7.815831e+07
Name: Reviews, dtype: float64
```

4.3 Now, handling the installs column

[16]:
```
df.Installs.value_counts()
```

[16]:
```
1,000,000+        1576
10,000,000+       1252
100,000+          1150
10,000+           1009
5,000,000+         752
1,000+             712
500,000+           537
50,000+            466
5,000+             431
```

```
100,000,000+       409
100+               309
50,000,000+        289
500+               201
500,000,000+        72
10+                 69
1,000,000,000+      58
50+                 56
5+                   9
1+                   3
Name: Installs, dtype: int64
```

We'll need to remove the commas and the plus signs Defining function for the same

[17]:
```
# define a function 'clean_installs' where replace(",","") and replace("+","")

def clean_installs(val):
    return int(val.replace(",","").replace("+",""))
```

[18]:
```
# use map to apply the function to the column as shown earlier
df.Installs = df.Installs.map(clean_installs)
```

[19]:
```
df.Installs.describe()
```

[19]:
```
count    9.360000e+03
mean     1.790875e+07
std      9.126637e+07
min      1.000000e+00
25%      1.000000e+04
50%      5.000000e+05
75%      5.000000e+06
max      1.000000e+09
Name: Installs, dtype: float64
```

2.4.1 4.1 Handling the app size field

[20]:
```
# write a function 'change_size',
# if there is M which is size in MB, delete the last element, mutiply it with␣
 ↪1000 and convert it to float
# if there is k which is size in kB, delete the last element and convert it to␣
 ↪float
# otherwise return None

def change_size(size):
    if "M" in size:
```

```
        x = size[:-1] # start : stop - 1
        x = float(x)*1000
        return x
    elif 'k' in size[-1]:
        x = size[:-1]
        x = float(x)
        return x
    else:
        return None
```

[21]:
```
change_size("19k")
```

[21]:
```
19.0
```

[22]:
```
"19k"[-1]
```

[22]:
```
'k'
```

[23]:
```
# use map to apply the function to the column as shown earlier
df["Size"] = df["Size"].map(change_size)
```

[24]:
```
df.Size.describe()
```

[24]:
```
count      7723.000000
mean      22970.456105
std       23449.628935
min           8.500000
25%        5300.000000
50%       14000.000000
75%       33000.000000
max      100000.000000
Name: Size, dtype: float64
```

[25]:
```
df["Size"].isnull().sum()
```

[25]:
```
1637
```

[26]:
```
# filling Size which had NA
df.Size.fillna(method = 'ffill', inplace = True) # the missing values are␣
↪introduced when we are working on the data type
```

[27]:
```
df.dtypes
```

[27]:
```
App                object
Category           object
Rating            float64
Reviews           float64
```

```
Size              float64
Installs            int64
Type               object
Price             float64
Content Rating     object
Genres             object
Last Updated       object
Current Ver        object
Android Ver        object
dtype: object
```

2.5 5. Some sanity checks

1. 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.
2. Reviews should not be more than installs as only those who installed can review the app. If there are any such records, drop them.
3. For free apps (type = "Free"), the price should not be >0. Drop any such rows.

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

[28]:
```
df.Rating.describe()
```

[28]:
```
count    9360.000000
mean        4.191838
std         0.515263
min         1.000000
25%         4.000000
50%         4.300000
75%         4.500000
max         5.000000
Name: Rating, dtype: float64
```

Min is 1 and max is 5. Looks good.

[29]:
```
df.loc[df.Rating > 5]
```

[29]:
```
Empty DataFrame
Columns: [App, Category, Rating, Reviews, Size, Installs, Type, Price, Content
Rating, Genres, Last Updated, Current Ver, Android Ver]
Index: []
```

5.2. Reviews should not be more than installs as only those who installed can review the app. Checking if reviews are more than installs. Counting total rows like this.

[30]:
```
len(df[df['Reviews'] > df['Installs']])
```

[30]: 7

[31]:
```
df[df['Reviews'] > df['Installs']]
```

[31]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
2454	KBA-EZ Health Guide	MEDICAL	5.0	4.0	25000.0	1	Free	0.00	Everyone	Medical	August 2, 2018	1.0.72	4.0.3 and up
4663	Alarmy (Sleep If U Can) - Pro	LIFESTYLE	4.8	10249.0	30000.0	10000	Paid	2.49	Everyone	Lifestyle	July 30, 2018	Varies with device	Varies with device
5917	Ra Ga Ba	GAME	5.0	2.0	20000.0	1	Paid	1.49	Everyone	Arcade	February 8, 2017	1.0.4	2.3 and up
6700	Brick Breaker BR	GAME	5.0	7.0	19000.0	5	Free	0.00	Everyone	Arcade	July 23, 2018	1	4.1 and up
7402	Trovami se ci riesci	GAME	5.0	11.0	6100.0	10	Free	0.00	Everyone	Arcade	March 11, 2017	0.1	2.3 and up
8591	DN Blog	SOCIAL	5.0	20.0	4200.0	10	Free	0.00	Teen	Social	July 23, 2018	1	4.0 and up
10697	Mu.F.O.	GAME	5.0	2.0	16000.0	1	Paid	0.99	Everyone	Arcade	March 3, 2017	1	2.3 and up

[32]:
```
# work with the code used to drop the outliers in Missing Value and Outlier Treatment file
```

[33]:
```
df = df[df['Reviews'] <= df['Installs']]
```

[34]:
```
df.shape
```

[34]: (9353, 13)

2.5.1 5.3 For free apps (type = "Free"), the price should not be > 0. Drop any such rows.

[35]:
```
len(df[(df.Type == "Free") & (df.Price>0)])
```

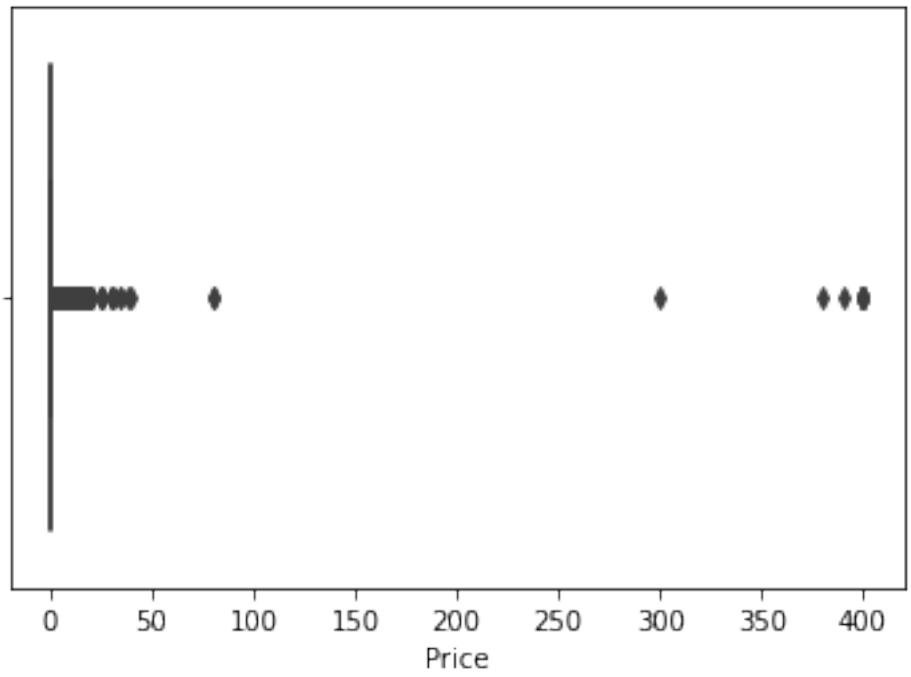
[35]: 0

2.6 5.A. Performing univariate analysis:

5.A. Performing univariate analysis: - Boxplot for Price o Are there any outliers? Think about the price of usual apps on Play Store. - Boxplot for Reviews o Are there any apps with very high number of reviews? Do the values seem right? - Histogram for Rating o How are the ratings distributed? Is it more toward higher ratings? - Histogram for Size ### Note down your observations for the plots made above. Which of these seem to have outliers?

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

[36]:
```
sns.boxplot(df.Price);
```

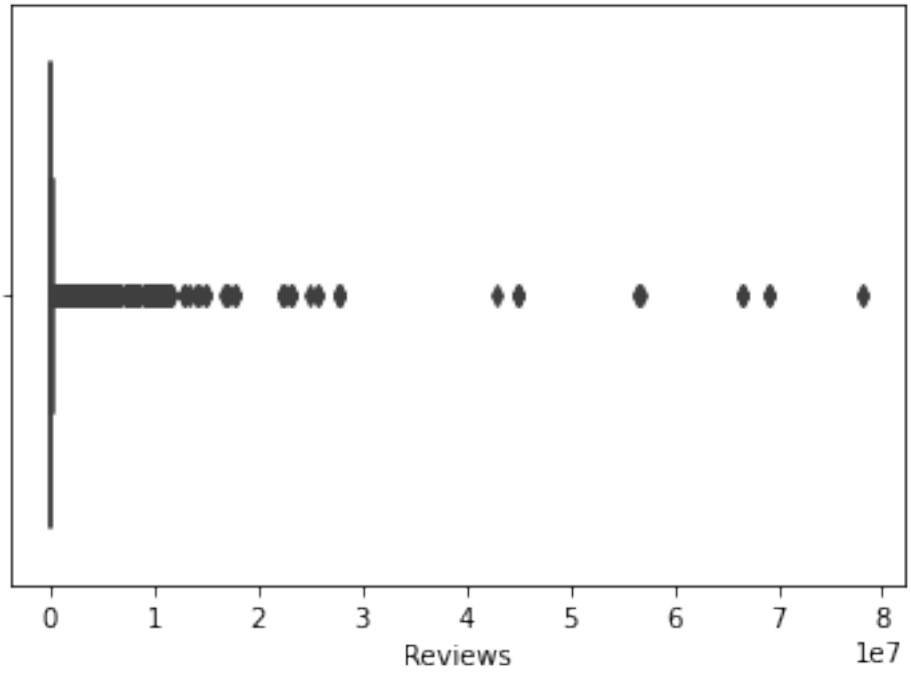


2.7 Analysis:

Yes, there are outliers. The price of apps are usually lesser than $50. Hence the price of other apps seem to be outliers.

Box plot for Reviews o Are there any apps with very high number of reviews? Do the values seem right?

[37]:
```
sns.boxplot(df.Reviews);
```

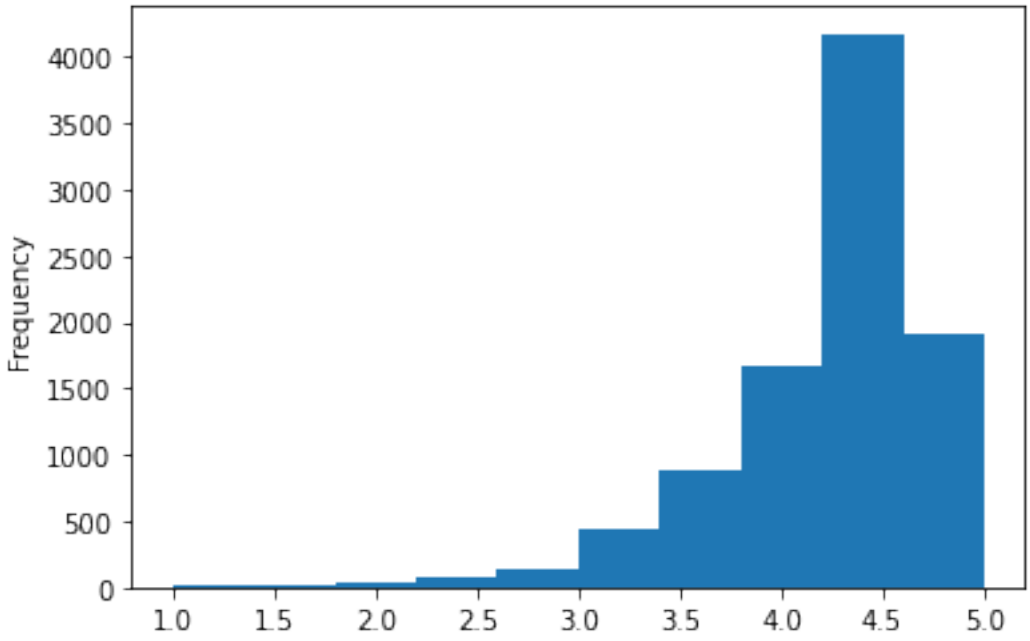


2.8 Analysis:

Yes, based on the above Box Plot, there seem to be some apps with higher number of reviews.

Histogram for Rating o How are the ratings distributed? Is it more toward higher ratings?

[38]:
```
df.Rating.plot.hist();
```

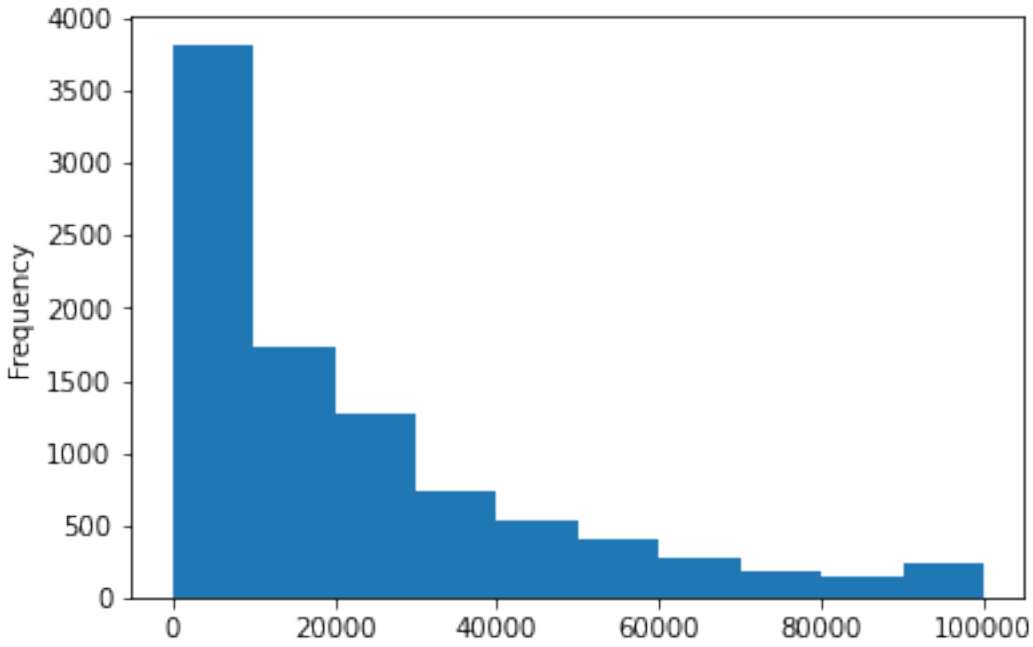


2.9 Analysis:

Yes, based on the above histogram, the Ratings are above 4.0 for a lot of apps.

Histogram of Size

[39]:
```
df.Size.plot.hist();
```



2.10 6. Outlier treatment:

1. 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!
   a. Check out the records with very high price
      i. Is 200 indeed a high price?
   b. Drop these as most seem to be junk apps
2. 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.
3. Installs: There seems to be some outliers in this field too. Apps having very high number of installs should be dropped from the analysis. a. Find out the different percentiles – 10, 25, 50, 70, 90, 95, 99 b. Decide a threshold as cutoff for outlier and drop records having values more than that

6.1. 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!

```
a.  Check out the records with very high price
    i.  Is 200 indeed a high price?
b.  Drop these as most seem to be junk apps
```

[40]:
```
len(df[df['Price'] >= 200])
```

[40]: 15

[41]:
```
df = df[df['Price'] < 200]
```

[42]:
```
df.shape
```

[42]: (9338, 13)

6.2 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.

[43]:
```
df = df[df['Reviews'] < 2000000]
```

[44]:
```
df.shape
```

[44]: (8885, 13)

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

a. Find out the different percentiles - 10, 25, 50, 70, 90, 95, 99
b. Decide a threshold as cutoff for outlier and drop records having values more than that

Dropping very high Installs values

[45]:
```
df.Installs.quantile([0.1, 0.25, 0.5, 0.70, 0.9, 0.95, 0.99])
```

[45]:
```
0.10         1000.0
0.25        10000.0
0.50       500000.0
0.70      1000000.0
0.90     10000000.0
0.95     10000000.0
0.99    100000000.0
Name: Installs, dtype: float64
```

[46]:
```
Q1 = df.Installs.quantile(0.25)
Q1
```

[46]: 10000.0

[47]:
```
Q3 = df.Installs.quantile(0.75)
Q3
```

[47]: 5000000.0

[48]:
```
IQR = Q3 - Q1
IQR
```

[48]: 4990000.0

[49]:
```
lower_limit = Q1 - 1.5*IQR
upper_limit = Q3 + 1.5*IQR

print(lower_limit)
print(upper_limit)
```

```
-7475000.0
12485000.0
```

[50]:
```
len(df[df['Installs'] > 12485000.0])
```

[50]: 389

[51]:
```
len(df[df['Installs'] >= 100000000])
```

[51]: 142

[52]:
```
df = df[df['Installs'] < 100000000]
```

[53]:
```
df.shape
```

[53]: (8743, 13)

Looks like there are just 1% apps having more than 100M installs. These apps might be genuine, but will definitely skew our analysis.
We need to drop these.

2.11 7. Bivariate analysis: Let's look at how the available predictors relate to the variable of interest, i.e., our target variable rating. Make scatter plots (for numeric features) and box plots (for character features) to assess the relations between rating and the other features.

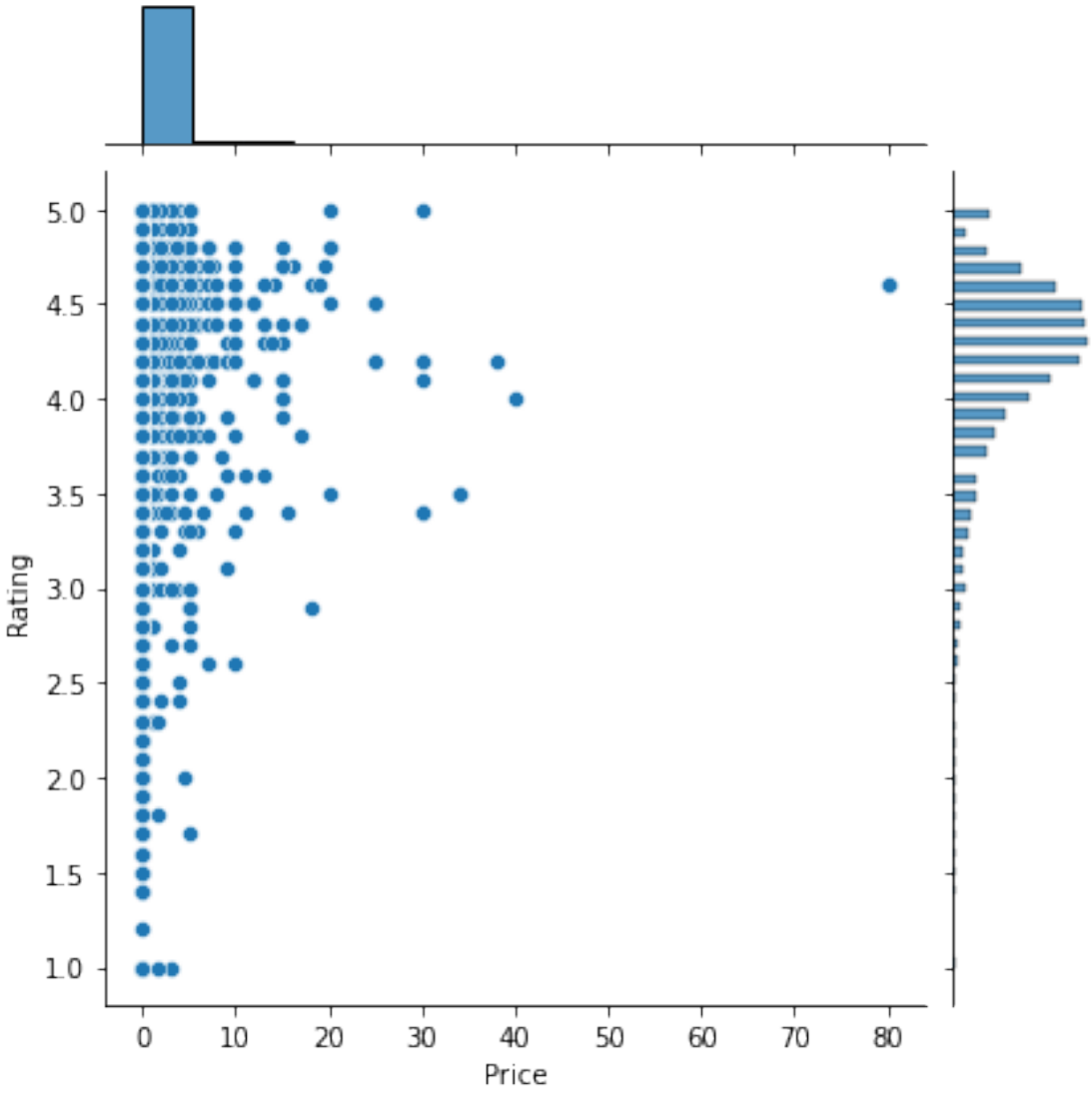
1. Make scatter plot/joinplot for Rating vs. Price
    a. What pattern do you observe? Does rating increase with price?
2. Make scatter plot/joinplot for Rating vs. Size
    a. Are heavier apps rated better?
3. Make scatter plot/joinplot for Rating vs. Reviews
    a. Does more review mean a better rating always?
4. Make boxplot for Rating vs. Content Rating
    a. Is there any difference in the ratings? Are some types liked better?
5. Make boxplot for Ratings vs. Category
    a. Which genre has the best ratings?

2.11.1 For each of the plots above, note down your observation.

7.1. Make scatter plot/joinplot for Rating vs Price

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

[54]:
```
sns.jointplot(df.Price, df.Rating);
```



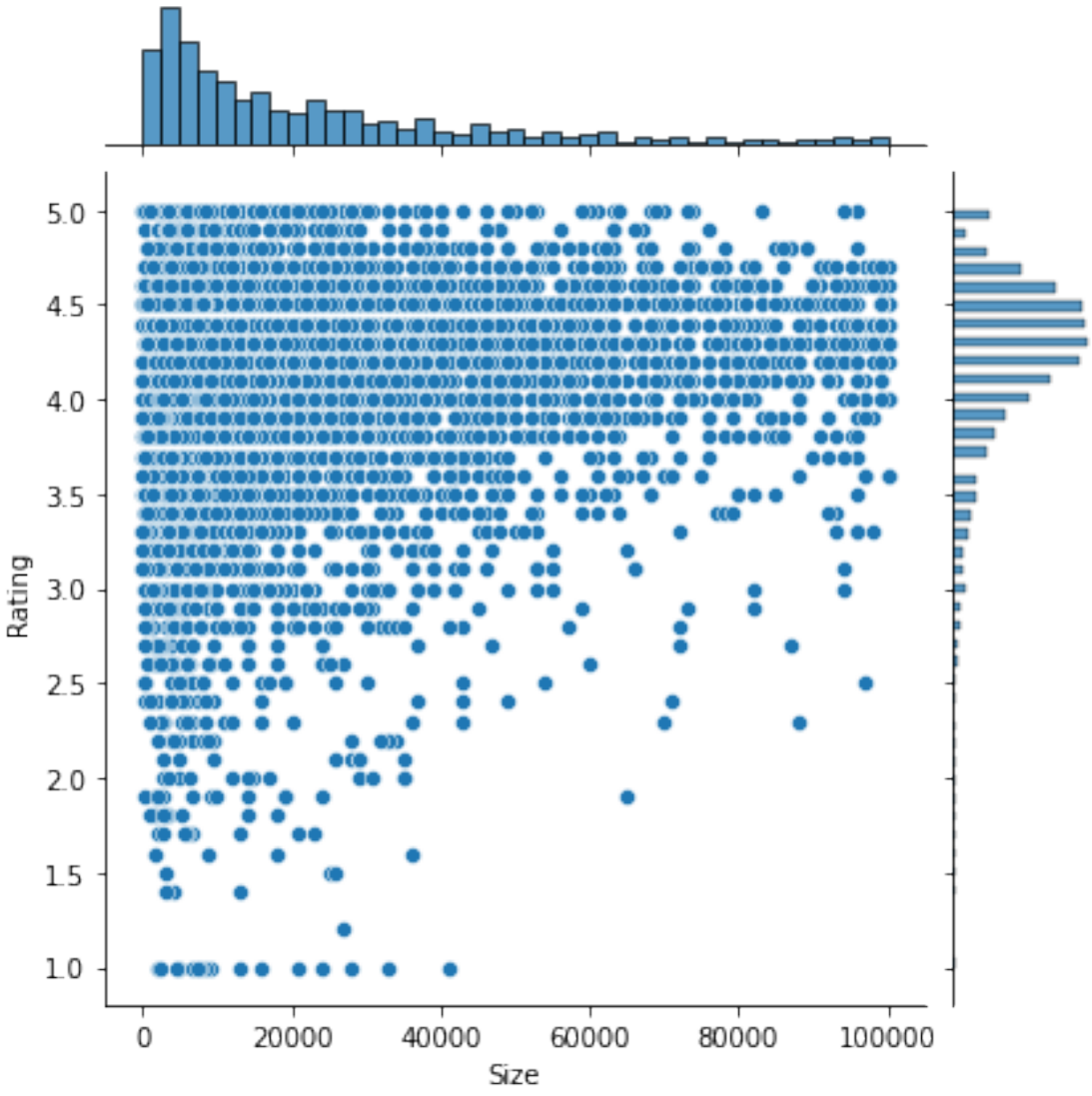
2.12 Analysis:

No, from the above plot, Rating does not increase with Price.

7.2 Make scatter plot/joinplot for Rating vs Size

a. Are heavier apps rated better?

[55]:
```
sns.jointplot(df.Size,df.Rating);
```



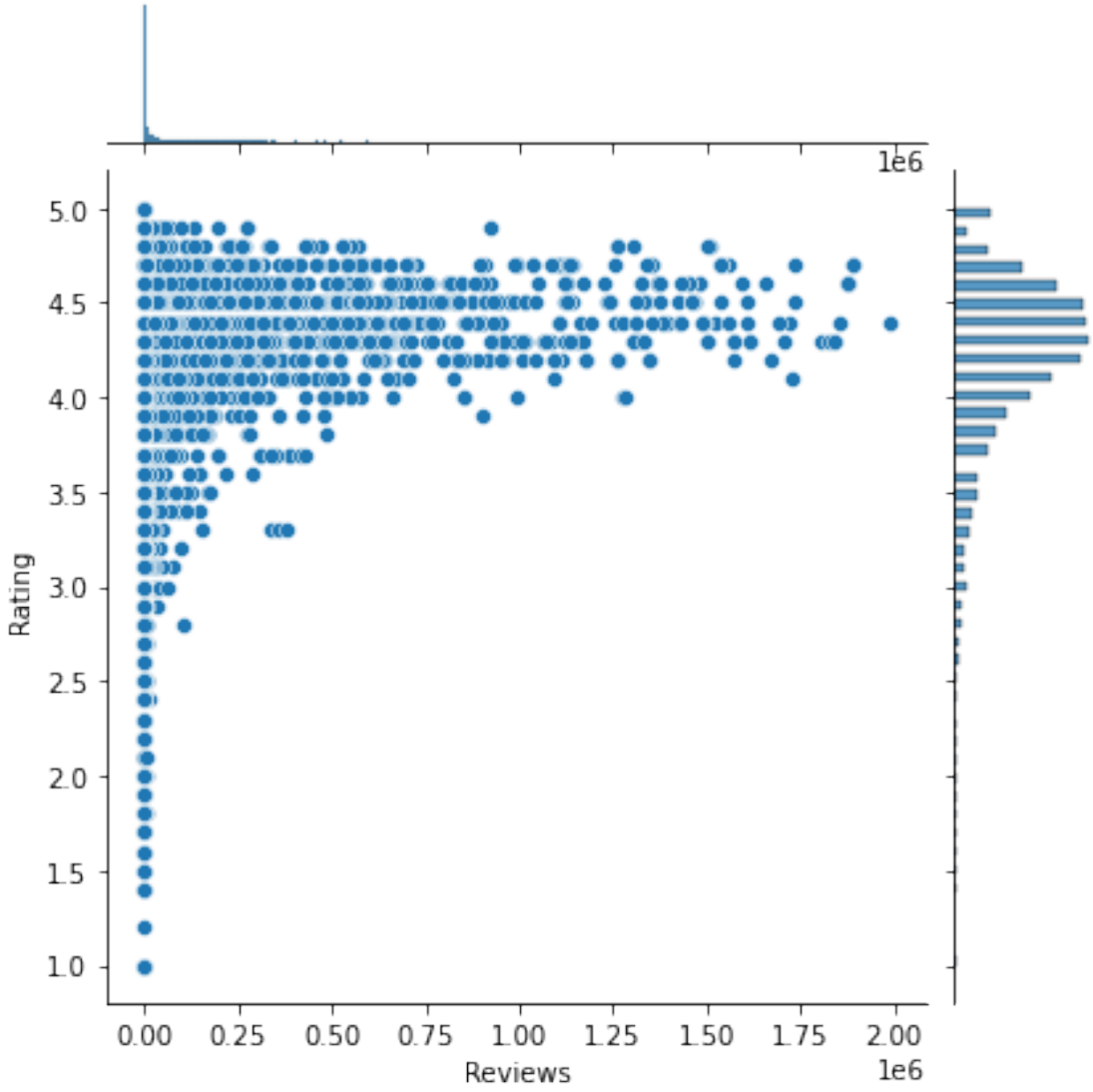
2.13 Analysis:

Yes, the above plot shows that the heavier apps are rated better.

7.3 Make scatter plot/joinplot for Rating vs Reviews

a. Does more review mean a better rating always?

[56]:
```
sns.jointplot(df.Reviews, df.Rating);
```



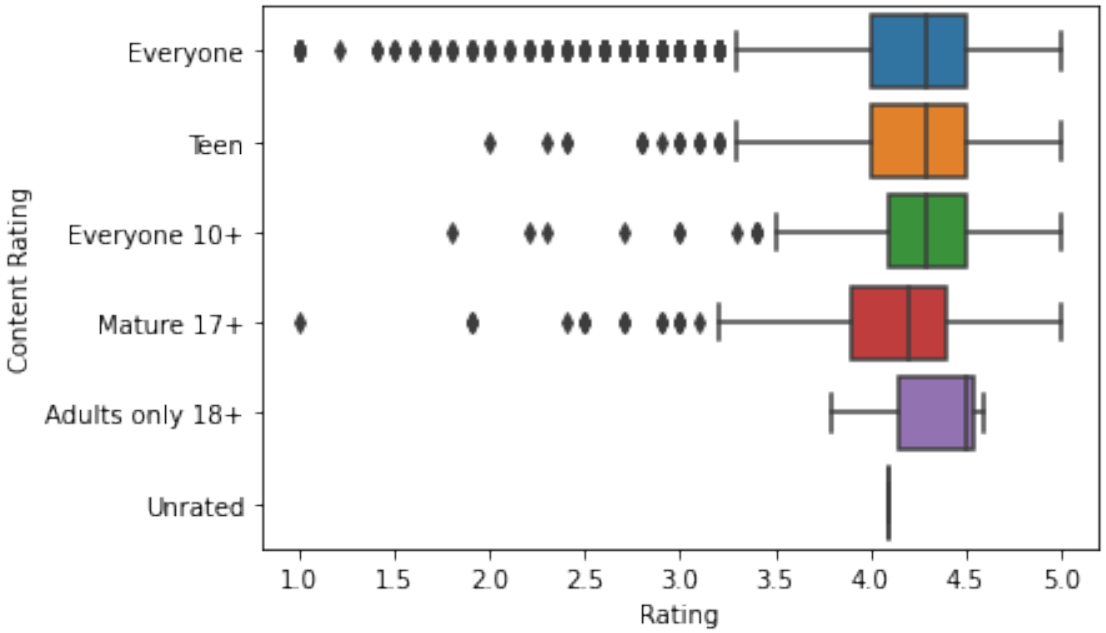
2.14 Analysis:

Yes, more the Reviews, higher the Ratings.

7.4 Make boxplot for Rating vs Content Rating

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

[57]:
```
sns.boxplot(df.Rating, df['Content Rating']);
```



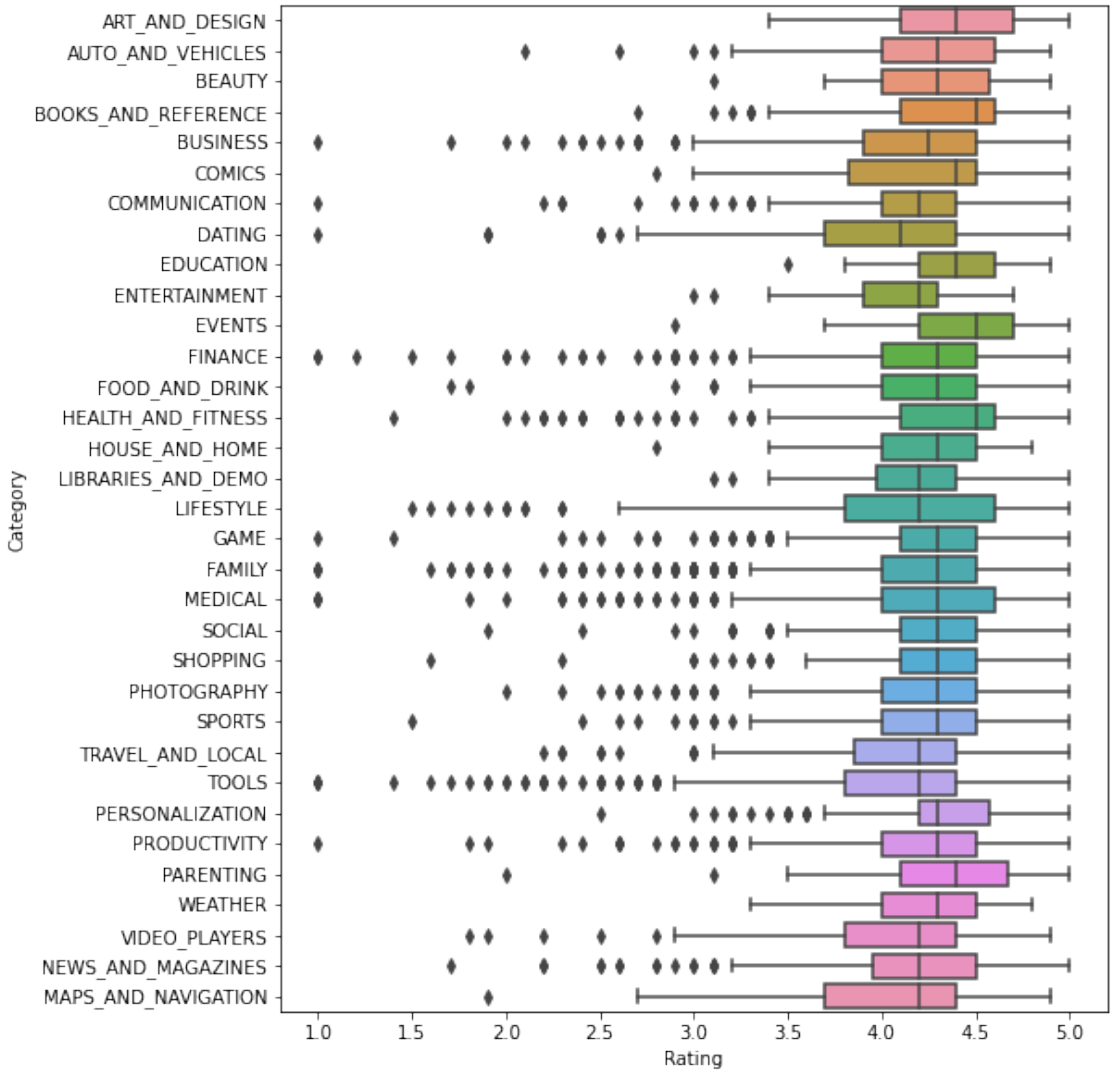
2.15 Analysis:

Apps with Content Rating as 'Everyone', 'Teen', 'Everyone 10+' and 'Adults only 18+' are liked better than 'Mature 17+' and 'Unrated'.

7.5 Make boxplot for Ratings vs. Category

a. Which genre has the best ratings?

[58]:
```
plt.figure(figsize=(8,10))
sns.boxplot(x=df['Rating'],y=df['Category']);
plt.show()
```



[59]:
```
df2 = pd.DataFrame(df.groupby(['Category'])['Rating'].agg(np.median))
df2.loc[(df2['Rating'] == df2['Rating'].max())]
```

[59]:

Category	Rating
BOOKS_AND_REFERENCE	4.5
EVENTS	4.5
HEALTH_AND_FITNESS	4.5

2.16 Analysis:

Apps under 'Books and Reference', 'Events' and 'Health & Fitness' have higher ratings than others.

2.17 8 Data preprocessing

For the steps below, create a copy of the dataframe to make all the edits. Name it inp1. 1. 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. 2. Drop columns App, Last Updated, Current Ver, and Android Ver. These variables are not useful for our task. 3. 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.

Making a copy of the dataset

[60]:
```
inp1 = df.copy()
```

8.1 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.

[61]:
```
# describe
inp1.Installs.describe()
```

[61]:
```
count    8.743000e+03
mean     3.486865e+06
std      8.659419e+06
min      5.000000e+00
25%      1.000000e+04
50%      1.000000e+05
75%      5.000000e+06
max      5.000000e+07
Name: Installs, dtype: float64
```

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

[63]:
```
# Do the same thing for Reviews
inp1.Reviews.describe()
```

[63]:
```
count    8.743000e+03
mean     8.957859e+04
std      2.320521e+05
min      1.000000e+00
25%      1.490000e+02
50%      3.878000e+03
75%      5.023650e+04
max      1.986068e+06
Name: Reviews, dtype: float64
```

[64]:
```
inp1.Reviews = inp1.Reviews.apply(np.log1p)
```

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

[65]:
```
inp1.shape
```

[65]:
```
(8743, 13)
```

[66]:
```
# ["App", "Last Updated", "Current Ver", "Android Ver"]
drop_columns = ['App', 'Last Updated', 'Current Ver', 'Android Ver']
inp1.drop(drop_columns, axis = 1, inplace = True)
```

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

[67]:
```
(8743, 9)
```

8.3 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. Getting dummy variables for Category, Genres, Content Rating

[68]:
```
inp1.dtypes
```

[68]:
```
Category           object
Rating            float64
Reviews           float64
Size              float64
Installs          float64
Type               object
Price             float64
Content Rating     object
Genres             object
dtype: object
```

[69]:
```
pd.get_dummies(inp1['Category'], prefix = 'Category', drop_first = True)
```

[69]:
```
       Category_AUTO_AND_VEHICLES  Category_BEAUTY  \
0                               0                0
1                               0                0
2                               0                0
3                               0                0
4                               0                0
…                             …                …
10834                           0                0
10836                           0                0
10837                           0                0
10839                           0                0
10840                           0                0
```

```
       Category_BOOKS_AND_REFERENCE  Category_BUSINESS  Category_COMICS  \
0                                 0                  0                0
1                                 0                  0                0
2                                 0                  0                0
3                                 0                  0                0
4                                 0                  0                0
…                                 …                  …                …
10834                             0                  0                0
10836                             0                  0                0
10837                             0                  0                0
10839                             1                  0                0
10840                             0                  0                0

       Category_COMMUNICATION  Category_DATING  Category_EDUCATION  \
0                           0                0                   0
1                           0                0                   0
2                           0                0                   0
3                           0                0                   0
4                           0                0                   0
…                           …                …                   …
10834                       0                0                   0
10836                       0                0                   0
10837                       0                0                   0
10839                       0                0                   0
10840                       0                0                   0

       Category_ENTERTAINMENT  Category_EVENTS  …  Category_PERSONALIZATION  \
0                           0                0  …                         0
1                           0                0  …                         0
2                           0                0  …                         0
3                           0                0  …                         0
4                           0                0  …                         0
…                           …                …  …                         …
10834                       0                0  …                         0
10836                       0                0  …                         0
10837                       0                0  …                         0
10839                       0                0  …                         0
10840                       0                0  …                         0

       Category_PHOTOGRAPHY  Category_PRODUCTIVITY  Category_SHOPPING  \
0                         0                      0                  0
1                         0                      0                  0
2                         0                      0                  0
3                         0                      0                  0
4                         0                      0                  0
…                         …                      …                  …
```

```
10834                    0                    0                    0
10836                    0                    0                    0
10837                    0                    0                    0
10839                    0                    0                    0
10840                    0                    0                    0

       Category_SOCIAL  Category_SPORTS  Category_TOOLS  \
0                    0                0               0
1                    0                0               0
2                    0                0               0
3                    0                0               0
4                    0                0               0
…                  …                …               …
10834                0                0               0
10836                0                0               0
10837                0                0               0
10839                0                0               0
10840                0                0               0

       Category_TRAVEL_AND_LOCAL  Category_VIDEO_PLAYERS  Category_WEATHER
0                              0                       0                 0
1                              0                       0                 0
2                              0                       0                 0
3                              0                       0                 0
4                              0                       0                 0
…                            …                       …               …
10834                          0                       0                 0
10836                          0                       0                 0
10837                          0                       0                 0
10839                          0                       0                 0
10840                          0                       0                 0

[8743 rows x 32 columns]
```

[70]:
```
inp1 = inp1.join(pd.get_dummies(inp1['Category'], prefix = 'Category',␣
↪drop_first = True))
inp1.head(2)
```

[70]:
```
          Category  Rating   Reviews     Size   Installs  Type  Price  \
0  ART_AND_DESIGN     4.1  5.075174  19000.0   9.210440  Free    0.0
1  ART_AND_DESIGN     3.9  6.875232  14000.0  13.122365  Free    0.0

  Content Rating                     Genres  Category_AUTO_AND_VEHICLES  …  \
0       Everyone               Art & Design                           0  …
1       Everyone  Art & Design;Pretend Play                           0  …

   Category_PERSONALIZATION  Category_PHOTOGRAPHY  Category_PRODUCTIVITY  \
```

```
0                         0                     0                      0
1                         0                     0                      0

   Category_SHOPPING  Category_SOCIAL  Category_SPORTS  Category_TOOLS  \
0                  0                0                0               0
1                  0                0                0               0

   Category_TRAVEL_AND_LOCAL  Category_VIDEO_PLAYERS  Category_WEATHER
0                          0                       0                 0
1                          0                       0                 0

[2 rows x 41 columns]
```

[71]:
```
pd.get_dummies(inp1['Genres'], prefix = 'Genres', drop_first = True)
```

[71]:
```
       Genres_Action;Action & Adventure  Genres_Adventure  \
0                                     0                 0
1                                     0                 0
2                                     0                 0
3                                     0                 0
4                                     0                 0
…                                   …                 …
10834                                 0                 0
10836                                 0                 0
10837                                 0                 0
10839                                 0                 0
10840                                 0                 0

       Genres_Adventure;Action & Adventure  Genres_Adventure;Brain Games  \
0                                        0                             0
1                                        0                             0
2                                        0                             0
3                                        0                             0
4                                        0                             0
…                                      …                             …
10834                                    0                             0
10836                                    0                             0
10837                                    0                             0
10839                                    0                             0
10840                                    0                             0

       Genres_Adventure;Education  Genres_Arcade  \
0                               0              0
1                               0              0
2                               0              0
3                               0              0
4                               0              0
```

```
…                              …             …
10834                              0             0
10836                              0             0
10837                              0             0
10839                              0             0
10840                              0             0

       Genres_Arcade;Action & Adventure  Genres_Arcade;Pretend Play  \
0                                     0                           0
1                                     0                           0
2                                     0                           0
3                                     0                           0
4                                     0                           0
…                                   …                           …
10834                                 0                           0
10836                                 0                           0
10837                                 0                           0
10839                                 0                           0
10840                                 0                           0

       Genres_Art & Design  Genres_Art & Design;Creativity  …  Genres_Tools  \
0                        1                               0  …             0
1                        0                               0  …             0
2                        1                               0  …             0
3                        1                               0  …             0
4                        0                               1  …             0
…                      …                               …  …           …
10834                    0                               0  …             0
10836                    0                               0  …             0
10837                    0                               0  …             0
10839                    0                               0  …             0
10840                    0                               0  …             0

       Genres_Tools;Education  Genres_Travel & Local  \
0                           0                      0
1                           0                      0
2                           0                      0
3                           0                      0
4                           0                      0
…                         …                      …
10834                       0                      0
10836                       0                      0
10837                       0                      0
10839                       0                      0
10840                       0                      0

       Genres_Travel & Local;Action & Adventure  Genres_Trivia  \
```

```
0                                                  0              0
1                                                  0              0
2                                                  0              0
3                                                  0              0
4                                                  0              0
…                                                  …              …
10834                                              0              0
10836                                              0              0
10837                                              0              0
10839                                              0              0
10840                                              0              0

       Genres_Video Players & Editors  \
0                                    0
1                                    0
2                                    0
3                                    0
4                                    0
…                                    …
10834                                0
10836                                0
10837                                0
10839                                0
10840                                0

       Genres_Video Players & Editors;Creativity  \
0                                                0
1                                                0
2                                                0
3                                                0
4                                                0
…                                                …
10834                                            0
10836                                            0
10837                                            0
10839                                            0
10840                                            0

       Genres_Video Players & Editors;Music & Video  Genres_Weather  \
0                                                 0               0
1                                                 0               0
2                                                 0               0
3                                                 0               0
4                                                 0               0
…                                                 …               …
10834                                             0               0
10836                                             0               0
```

```
10837                                                 0               0
10839                                                 0               0
10840                                                 0               0

       Genres_Word
0                0
1                0
2                0
3                0
4                0
…              …
10834            0
10836            0
10837            0
10839            0
10840            0

[8743 rows x 114 columns]
```

[72]:
```
inp1 = inp1.join(pd.get_dummies(inp1['Genres'], prefix = 'Genres', drop_first =␣
↪True))
inp1.head(2)
```

[72]:
```
          Category  Rating   Reviews     Size   Installs  Type  Price  \
0  ART_AND_DESIGN     4.1  5.075174  19000.0   9.210440  Free    0.0
1  ART_AND_DESIGN     3.9  6.875232  14000.0  13.122365  Free    0.0

  Content Rating                     Genres  Category_AUTO_AND_VEHICLES  …  \
0       Everyone               Art & Design                           0  …
1       Everyone  Art & Design;Pretend Play                           0  …

   Genres_Tools  Genres_Tools;Education  Genres_Travel & Local  \
0             0                       0                      0
1             0                       0                      0

   Genres_Travel & Local;Action & Adventure  Genres_Trivia  \
0                                         0              0
1                                         0              0

   Genres_Video Players & Editors  Genres_Video Players & Editors;Creativity  \
0                               0                                           0
1                               0                                           0

   Genres_Video Players & Editors;Music & Video  Genres_Weather  Genres_Word
0                                             0              0            0
1                                             0              0            0
```

```
[2 rows x 155 columns]
```

[73]:
```
pd.get_dummies(inp1['Content Rating'], prefix = 'Content Rating', drop_first =␣
↪True)
```

[73]:
```
       Content Rating_Everyone  Content Rating_Everyone 10+  \
0                            1                            0
1                            1                            0
2                            1                            0
3                            0                            0
4                            1                            0
…                          …                            …
10834                        1                            0
10836                        1                            0
10837                        1                            0
10839                        0                            0
10840                        1                            0

       Content Rating_Mature 17+  Content Rating_Teen  Content Rating_Unrated
0                              0                    0                       0
1                              0                    0                       0
2                              0                    0                       0
3                              0                    1                       0
4                              0                    0                       0
…                            …                    …                     …
10834                          0                    0                       0
10836                          0                    0                       0
10837                          0                    0                       0
10839                          1                    0                       0
10840                          0                    0                       0

[8743 rows x 5 columns]
```

[74]:
```
inp1 = inp1.join(pd.get_dummies(inp1['Content Rating'], prefix = 'Content␣
↪Rating', drop_first = True))
inp1.head(2)
```

[74]:
```
          Category  Rating   Reviews     Size   Installs  Type  Price  \
0  ART_AND_DESIGN     4.1  5.075174  19000.0   9.210440  Free    0.0
1  ART_AND_DESIGN     3.9  6.875232  14000.0  13.122365  Free    0.0

  Content Rating                     Genres  Category_AUTO_AND_VEHICLES  …  \
0       Everyone               Art & Design                           0  …
1       Everyone  Art & Design;Pretend Play                           0  …

   Genres_Video Players & Editors  Genres_Video Players & Editors;Creativity  \
0                               0                                          0
```

```
1                                0                                                0

   Genres_Video Players & Editors;Music & Video  Genres_Weather  Genres_Word  \
0                                             0               0            0
1                                             0               0            0

   Content Rating_Everyone  Content Rating_Everyone 10+  \
0                        1                            0
1                        1                            0

   Content Rating_Mature 17+  Content Rating_Teen  Content Rating_Unrated
0                          0                    0                       0
1                          0                    0                       0

[2 rows x 160 columns]
```

[75]:
```
inp1 = inp1.join(pd.get_dummies(inp1.Type, prefix = 'Type', drop_first = True))
inp1.head(2)
```

[75]:
```
          Category  Rating   Reviews     Size   Installs  Type  Price  \
0  ART_AND_DESIGN     4.1  5.075174  19000.0   9.210440  Free    0.0
1  ART_AND_DESIGN     3.9  6.875232  14000.0  13.122365  Free    0.0

  Content Rating                     Genres  Category_AUTO_AND_VEHICLES  …  \
0       Everyone               Art & Design                           0  …
1       Everyone  Art & Design;Pretend Play                           0  …

   Genres_Video Players & Editors;Creativity  \
0                                          0
1                                          0

   Genres_Video Players & Editors;Music & Video  Genres_Weather  Genres_Word  \
0                                             0               0            0
1                                             0               0            0

   Content Rating_Everyone  Content Rating_Everyone 10+  \
0                        1                            0
1                        1                            0

   Content Rating_Mature 17+  Content Rating_Teen  Content Rating_Unrated  \
0                          0                    0                       0
1                          0                    0                       0

   Type_Paid
0          0
1          0
```

```
[2 rows x 161 columns]
```

[76]:
```
drop_columns2 = ['Category','Type','Genres','Content Rating']
inp2 = inp1.drop(drop_columns2, axis =1)
```

[77]:
```
inp2.columns
```

[77]:
```
Index(['Rating', 'Reviews', 'Size', 'Installs', 'Price',
       'Category_AUTO_AND_VEHICLES', 'Category_BEAUTY',
       'Category_BOOKS_AND_REFERENCE', 'Category_BUSINESS', 'Category_COMICS',
       …
       'Genres_Video Players & Editors;Creativity',
       'Genres_Video Players & Editors;Music & Video', 'Genres_Weather',
       'Genres_Word', 'Content Rating_Everyone', 'Content Rating_Everyone 10+',
       'Content Rating_Mature 17+', 'Content Rating_Teen',
       'Content Rating_Unrated', 'Type_Paid'],
      dtype='object', length=157)
```

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

[78]:
```
(8743, 157)
```

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

Train - test split

[79]:
```
from sklearn.model_selection import train_test_split
```

[80]:
```
df_train, df_test = train_test_split(inp2, test_size = 0.3, random_state =1)
```

[81]:
```
display(df_train.shape)
display(df_test.shape)
```

```
(6120, 157)
```

```
(2623, 157)
```

[82]:
```
inp2.head()
```

[82]:

	Rating	Reviews	Size	Installs	Price	Category_AUTO_AND_VEHICLES	\
0	4.1	5.075174	19000.0	9.210440	0.0	0	
1	3.9	6.875232	14000.0	13.122365	0.0	0	
2	4.7	11.379520	8700.0	15.424949	0.0	0	
3	4.5	12.281389	25000.0	17.727534	0.0	0	

```
4     4.3   6.875232   2800.0  11.512935    0.0                          0

   Category_BEAUTY  Category_BOOKS_AND_REFERENCE  Category_BUSINESS  \
0                0                             0                  0
1                0                             0                  0
2                0                             0                  0
3                0                             0                  0
4                0                             0                  0

   Category_COMICS  …  Genres_Video Players & Editors;Creativity  \
0                0  …                                         0
1                0  …                                         0
2                0  …                                         0
3                0  …                                         0
4                0  …                                         0

   Genres_Video Players & Editors;Music & Video  Genres_Weather  Genres_Word  \
0                                             0               0            0
1                                             0               0            0
2                                             0               0            0
3                                             0               0            0
4                                             0               0            0

   Content Rating_Everyone  Content Rating_Everyone 10+  \
0                        1                            0
1                        1                            0
2                        1                            0
3                        0                            0
4                        1                            0

   Content Rating_Mature 17+  Content Rating_Teen  Content Rating_Unrated  \
0                          0                    0                       0
1                          0                    0                       0
2                          0                    0                       0
3                          0                    1                       0
4                          0                    0                       0

   Type_Paid
0          0
1          0
2          0
3          0
4          0

[5 rows x 157 columns]
```

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

[83]:
```
X_train = df_train.drop(['Rating'], axis = 1)
```

[84]:
```
X_test = df_test.drop(['Rating'], axis = 1)
```

[85]:
```
y_train = df_train['Rating']
```

[86]:
```
y_test = df_test['Rating']
```

[87]:
```
display(X_train.shape)
display(y_train.shape)
display(X_test.shape)
display(y_test.shape)
```

```
(6120, 156)
```

```
(6120,)
```

```
(2623, 156)
```

```
(2623,)
```

2.20 11 . Model building

- Use linear regression as the technique
- Report the R2 on the train set

[88]:
```
from sklearn.linear_model import LinearRegression
linreg = LinearRegression()

linreg.fit(X_train,y_train)
```

[88]:
```
LinearRegression()
```

[89]:
```
y_train_pred = linreg.predict(X_train)
```

[90]:
```
from sklearn.metrics import r2_score

print(r2_score(y_train, y_train_pred))
```

```
0.1656451695716198
```

2.21 12. Make predictions on test set and report R2.

[91]:
```
y_test_pred = linreg.predict(X_test)
```

[92]:
```
print(r2_score(y_test, y_test_pred))
```

```
0.13823167336238884
```

2.22 Mean Squared Error:

[93]:
```
from sklearn.metrics import mean_squared_error

mean_squared_error(y_test_pred, y_test, squared = False)
```

[93]:
```
0.4985860933341173
```

3 Analysis:

From the R squared value obtained above, looks like Linear Regression is not the apt Model Building Algorithm for the above specified Data Analysis.

[ ]:
```

```