In [167]:

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

Read Data & Checking NA values

In [168]:

```
data = pd.read_csv(r"C:\Users\Rahul\Downloads\1569582940_googleplaystore.zip"); data.head()
```

Out[168]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Andro V
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	4.0 and
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	4.0 and
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	4.0 and
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0	Teen	Art & Design	June 8, 2018	Varies with device	4.2 a
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0	Everyone	Art & Design;Creativity	June 20, 2018	1.1	4.4 a

In [169]:

```
print(data.dtypes,data.isnull().sum())
data.shape
```

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

Out[169]:

```
(10841, 13)
```

In [170]:

```
data.dropna(inplace=True);print(data.shape)
```

```
(9360, 13)
```

Data Cleaning

Variables seem to have incorrect type and inconsistent formatting,also Size column has sizes in Kb as well as Mb,Multiplying the value by 1,000, whose size is mentioned in Mb

In [171]:

```
print(data.Size.value_counts())

def change(Size):
    if 'M'in Size:
        x=Size[:-1]
        x=float(x)*1000
        return x

    elif 'k'in Size:
        x=Size[:-1]
        x=float(x)
        return x

    else: return None
```

```
Varies with device    1637
14M                    165
12M                    161
11M                    159
15M                    159
                      ...
556k                     1
818k                     1
121k                     1
376k                     1
246k                     1
Name: Size, Length: 413, dtype: int64
```

In [172]:

```
data.Size=data.Size.map(change);data.Size.value_counts()
```

Out[172]:

```
14000.0    165
12000.0    161
11000.0    159
15000.0    159
13000.0    157
          ...
241.0        1
837.0        1
930.0        1
812.0        1
143.0        1
Name: Size, Length: 411, dtype: int64
```

In [173]:

```
print(data.Size.isnull().sum())
data.Size.fillna(method='pad',inplace=True)
print(data.Size.isnull().sum())
```

```
1637
0
```

Reviews is a numeric field that is loaded as a string field.Convert it to numeric.

Installs field is currently stored as string and has values like 1,000,000+, remove ‘+’, ‘,’ from the field, convert it to integer.

Price field is a string and has symbol. Remove‘ ’ sign, and convert it to numeric.

In [174]:

```
data.Reviews=data.Reviews.astype('float')

print(data.Installs.value_counts()[:5])
data.Installs=data.Installs.map(lambda x:x.replace(',','').replace('+',''))
print(data.Installs.value_counts()[:5])
data.Installs=data.Installs.astype('float')

print(data.Price.value_counts()[:5])
data.Price=data.Price.map(lambda x:x.replace('$',''))
print(data.Price.value_counts()[:5])
data.Price=data.Price.astype('float')

print(data.dtypes)
```

```
1,000,000+     1576
10,000,000+    1252
100,000+       1150
10,000+        1009
5,000,000+      752
Name: Installs, dtype: int64
1000000     1576
10000000    1252
100000      1150
10000       1009
5000000      752
Name: Installs, dtype: int64
0        8715
$2.99     114
$0.99     106
$4.99      70
$1.99      59
Name: Price, dtype: int64
0       8715
2.99     114
0.99     106
4.99      70
1.99      59
Name: Price, dtype: int64
App                object
Category           object
Rating            float64
Reviews           float64
Size              float64
Installs          float64
Type               object
Price             float64
Content Rating     object
Genres             object
Last Updated       object
Current Ver        object
Android Ver        object
dtype: object
```

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.

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

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

In [175]:

```
print(len(data[data.Rating>5]))
print(len(data[data.Reviews>data.Installs]))
print(len(data[(data.Type=='free')&(data.Price>0)]))

data=data[data.Reviews<data.Installs].copy();print(data.shape)

print(len(data[data.Price>200]))
data=data[data.Price<200].copy();print(data.shape)

print(len(data[data.Reviews>=2000000]))
data=data[data.Reviews<=2000000].copy();print(data.shape)

print(data.Installs.quantile([.25,.50,.75,.90,.99]))

print(len(data[data.Installs>= 10000000]))

data=data[data.Installs<=10000000].copy();print(data.shape)
```

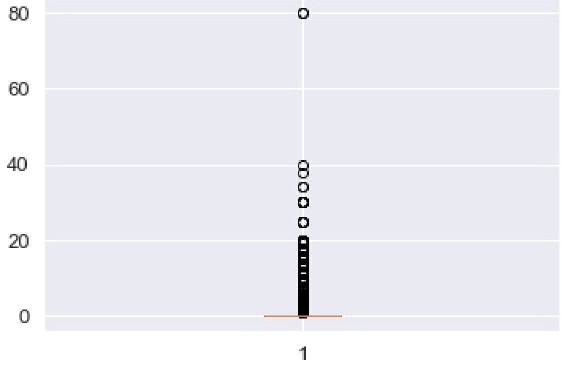
```
0
7
0
(9351, 13)
15
(9336, 13)
453
(8883, 13)
0.25        10000.0
0.50       500000.0
0.75      5000000.0
0.90     10000000.0
0.99    100000000.0
Name: Installs, dtype: float64
1627
(8494, 13)
```

Univariate Analysis

Boxplot for Price

In [176]:

```
plt.boxplot(data['Price'])
plt.show()
```

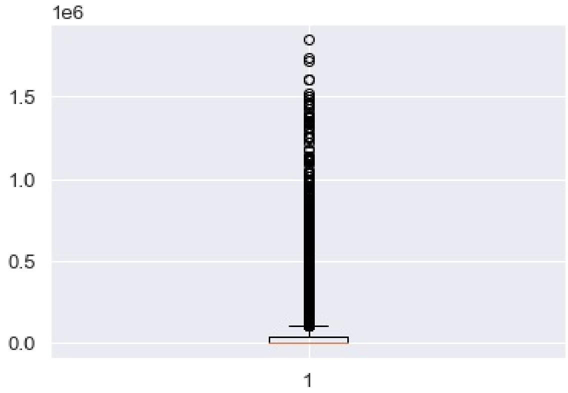


Price Boxplot shows that majority of the apps has usual price but rare apps have unusual price which oulies of the price range

Boxplot for Reviews

In [177]:

```
plt.boxplot(data['Reviews'])
plt.show()
```



Boxplot representation of Reviews shows that there is no app with higher Reviews, so values seems to be right.

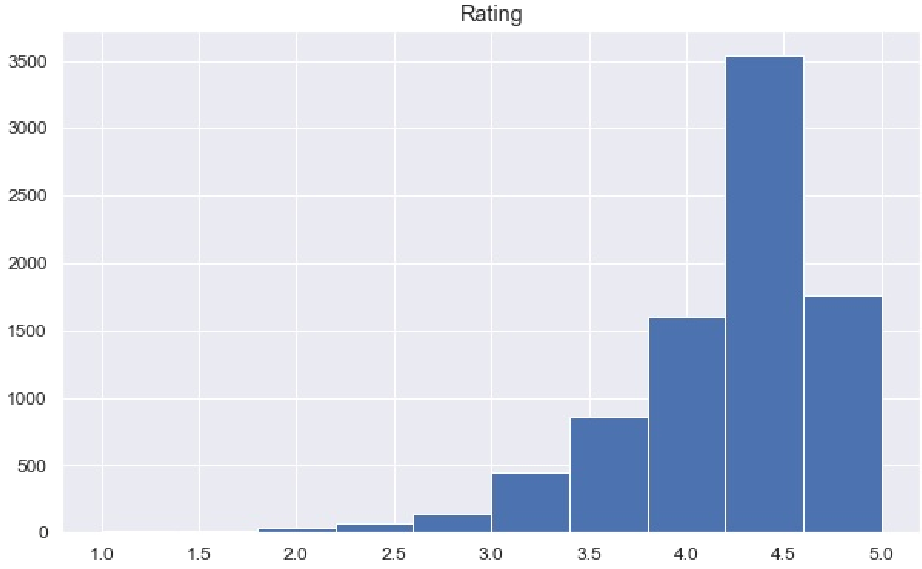
Histogram for Rating

In [178]:

```
data.hist(['Rating'], figsize = (10,6), xlabelsize=12, ylabelsize=12)
```

Out[178]:

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000009B5FB43490>]],
      dtype=object)
```



The rating shown on above histogram is more towards high ratings.

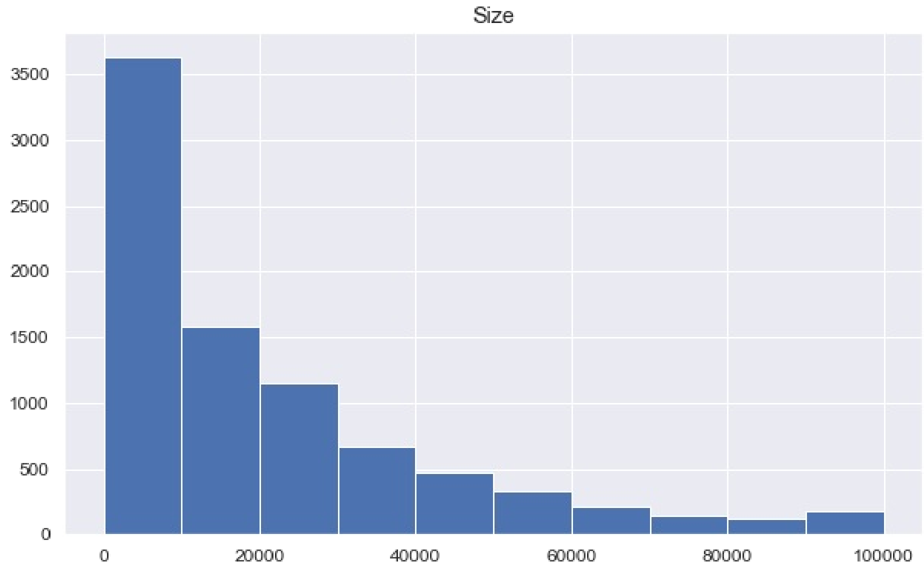
Histogram for Size

In [179]:

```
data.hist(['Size'], figsize = (10,6), xlabelsize=12, ylabelsize=12)
```

Out[179]:

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000009B5FC17820>]],
      dtype=object)
```



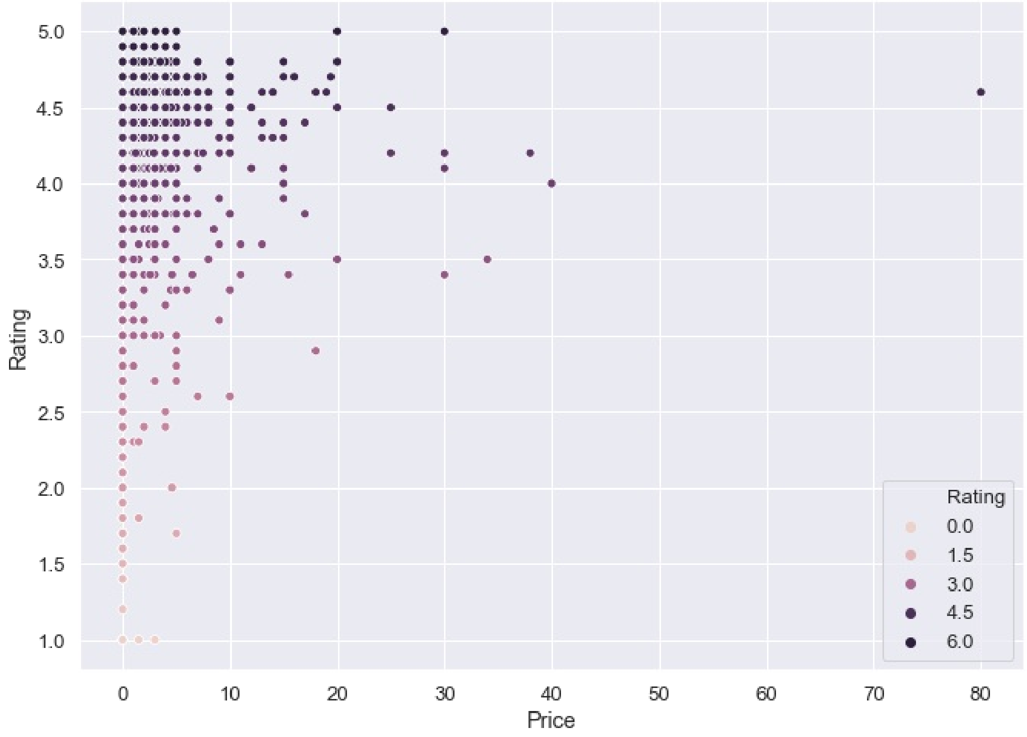
Bivariate Analysis

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.

In [180]:

```
# Scatterplot for Rating vs Price
plt.figure(figsize=(11,8))
sns.set_style(style='whitegrid',)
sns.set(font_scale=1.2)
sns.scatterplot(data.Price,data.Rating,hue=data.Rating)

plt.show()
```



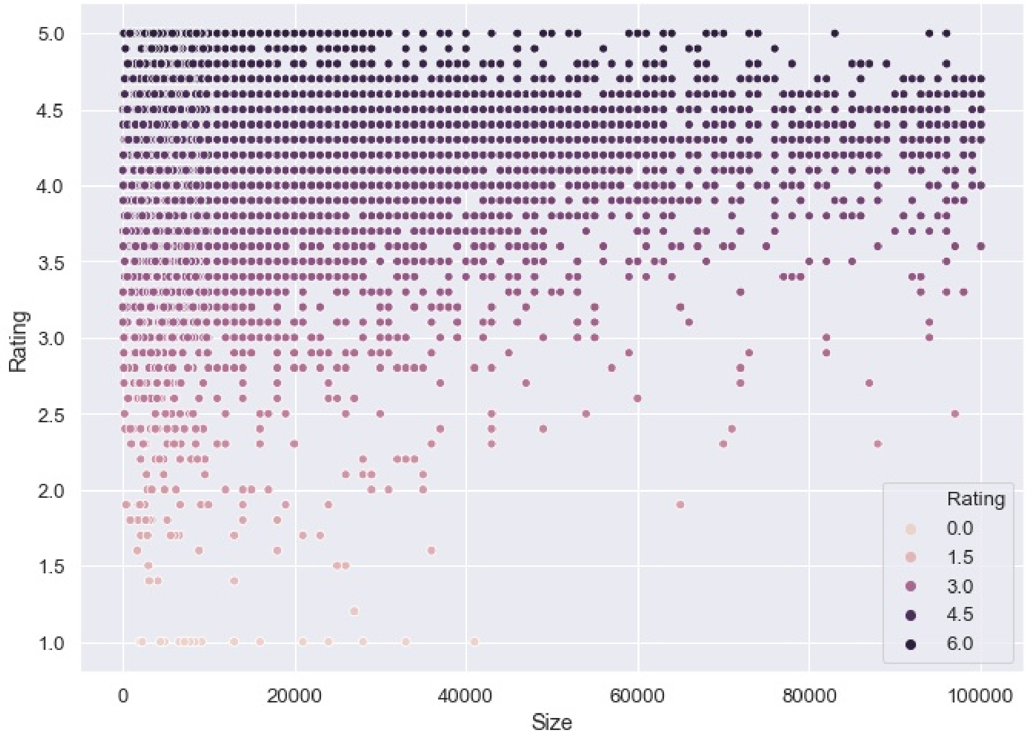
While there is not a very clean pattern, it does look that the higher priced apps have better rating. Although, there are not a lot of apps which are high priced, but the pattern is apparent.

In [181]:

```
# Scatterplot for Rating vs Size
plt.figure(figsize=(11,8))
sns.scatterplot(data.Size,data.Rating,hue=data.Rating)
```

Out[181]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x9b5fb4a0a0>
```



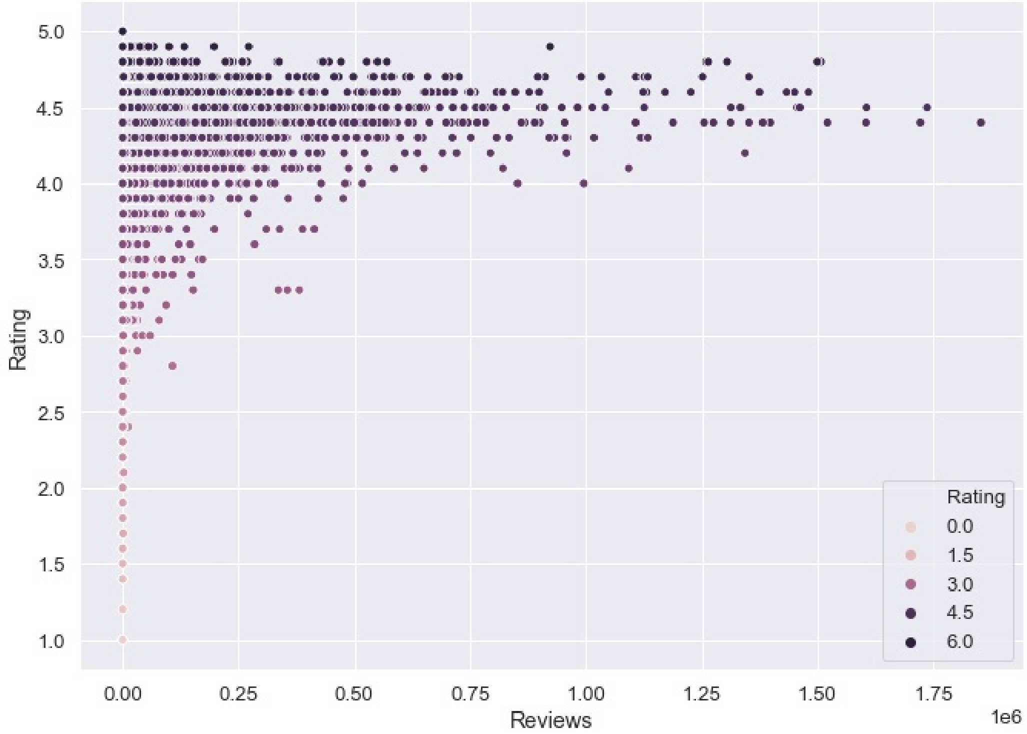
Again, not a very clean pattern, but it does look like heavier apps are better rated.

In [182]:

```
# Scatterplot for Rating vs Reviews
plt.figure(figsize=(11,8))
sns.scatterplot(data.Reviews,data.Rating,hue=data.Rating)
```

Out[182]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x9b5df77940>
```



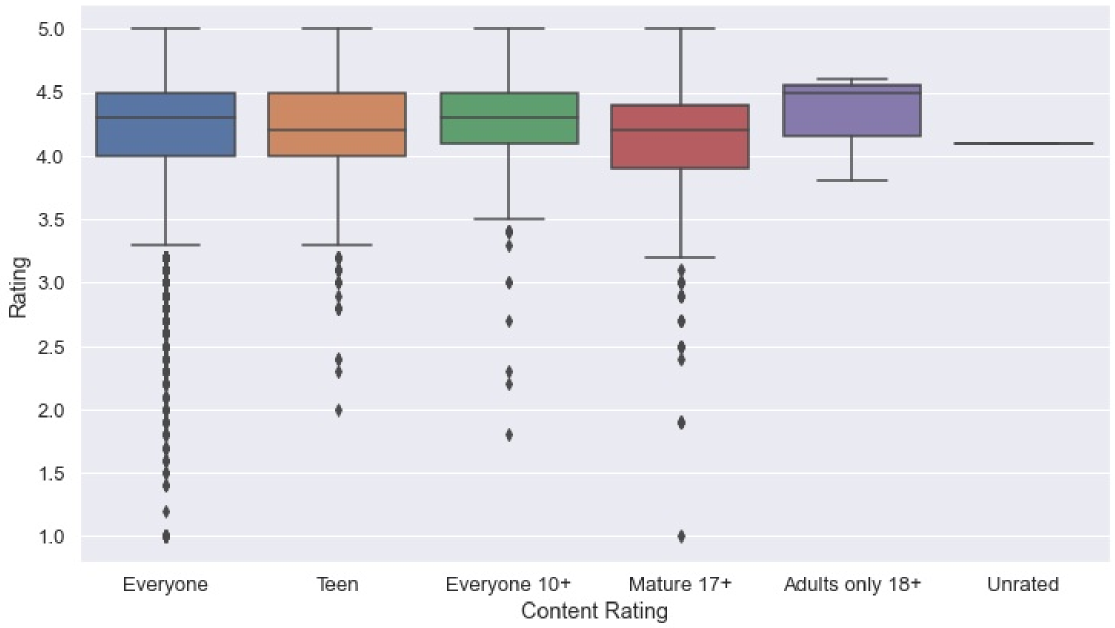
No clear pattern. There are fewer low rated apps among the popular ones (maybe poor ones won’t get so popular), after a certain point, the rating does not depend on the popularity.

In [183]:

```
# Rating vs Content Rating
plt.figure(figsize=(12,6.68))
sns.boxplot(data['Content Rating'],data.Rating)
```

Out[183]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x9b5fb4a9a0>
```



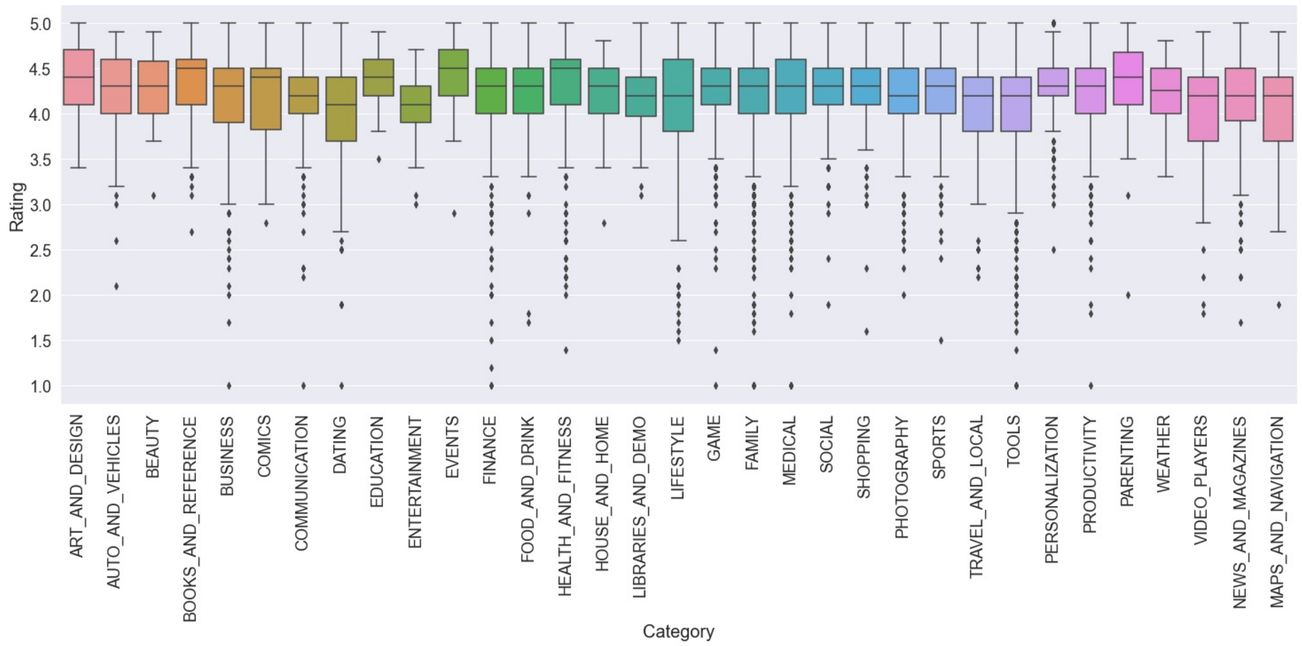
While the median rating for most others is similar, the rating for “Adults Only 18+” is the highest.

In [184]:

```
# Rating vs Category
plt.figure(figsize=(25,8.27))
sns.boxplot(data.Category,data.Rating)
plt.xticks(fontsize=18,rotation='vertical')
plt.yticks(fontsize=18)
plt.xlabel("Category",fontsize=20)
plt.ylabel("Rating",fontsize=20)
```

Out[184]:

```
Text(0, 0.5, 'Rating')
```



Apps around Health & Fitness, Books and Reference, Events seem to have the highest median ratings.

Data Preprocessing

In [186]:

```
inp1 = data.Reviews=data.Reviews.apply(func=np.log1p)
inp1
```

Out[186]:

```
0        1.804211
1        2.063723
2        2.516043
4        2.063723
5        1.812210
           ...   
10834    1.124748
10836    1.539779
10837    0.959135
10839    1.748318
10840    2.631528
Name: Reviews, Length: 8494, dtype: float64
```

In [188]:

```
inp1 = data.Installs=data.Installs.apply(func=np.log1p)
inp1
```

Out[188]:

```
0        2.323411
1        2.647760
2        2.798801
4        2.526763
5        2.469776
           ...   
10834    1.976385
10836    2.253121
10837    1.725463
10839    2.067970
10840    2.840136
Name: Installs, Length: 8494, dtype: float64
```

Deleting Unnecessary Variables

In [189]:

```
inp2 = data.drop(["App", "Last Updated", "Current Ver", "Android Ver"], axis=1, inplace=True);print(data.shape)
inp2 = data = pd.get_dummies(data,drop_first=True);print(data.columns)
```

```
(8494, 9)
Index(['Rating', 'Reviews', 'Size', 'Installs', 'Price',
       'Category_AUTO_AND_VEHICLES', 'Category_BEAUTY',
       'Category_BOOKS_AND_REFERENCE', 'Category_BUSINESS', 'Category_COMICS',
       ...
       'Genres_Tools', 'Genres_Tools;Education', 'Genres_Travel & Local',
       'Genres_Travel & Local;Action & Adventure', 'Genres_Trivia',
       'Genres_Video Players & Editors',
       'Genres_Video Players & Editors;Creativity',
       'Genres_Video Players & Editors;Music & Video', 'Genres_Weather',
       'Genres_Word'],
      dtype='object', length=157)
```

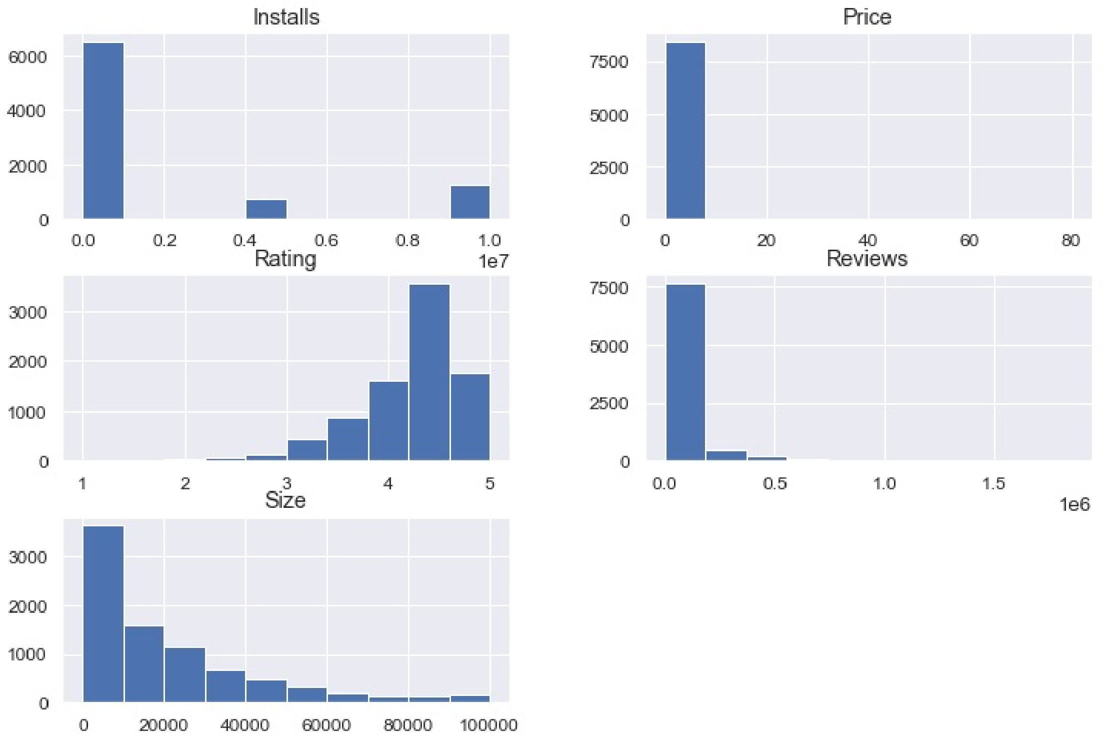
Outlier Correction

It seems from the histogram(below) the variables has some skewness and from boxplot it is evident that it has outliers too, Lets correct it by applying log.

In [137]:

```
# Histogram for the mention columns
print(data.hist(['Rating','Reviews','Size','Installs','Price'],figsize=(12,8),xlabelsize=12,ylabelsize=12))
```

```
[[<matplotlib.axes._subplots.AxesSubplot object at 0x000000D4B9D1B910>
  <matplotlib.axes._subplots.AxesSubplot object at 0x000000D4BBA7F2E0>]
 [<matplotlib.axes._subplots.AxesSubplot object at 0x000000D4BB6389A0>
  <matplotlib.axes._subplots.AxesSubplot object at 0x000000D4BB66E160>]
 [<matplotlib.axes._subplots.AxesSubplot object at 0x000000D4BB6968E0>
  <matplotlib.axes._subplots.AxesSubplot object at 0x000000D4BB6CC130>]]
```

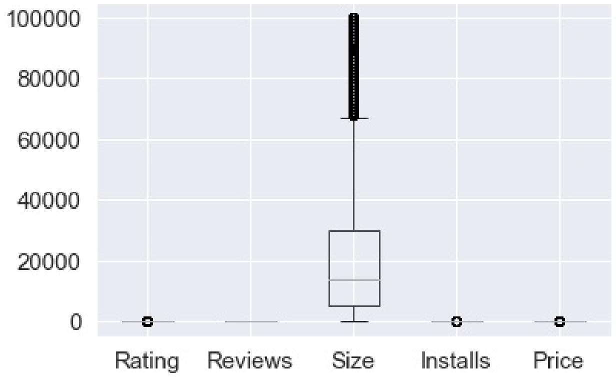


In [22]:

```
data.boxplot(fontsize=15)
```

Out[22]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x9b58cebd60>
```



In [94]:

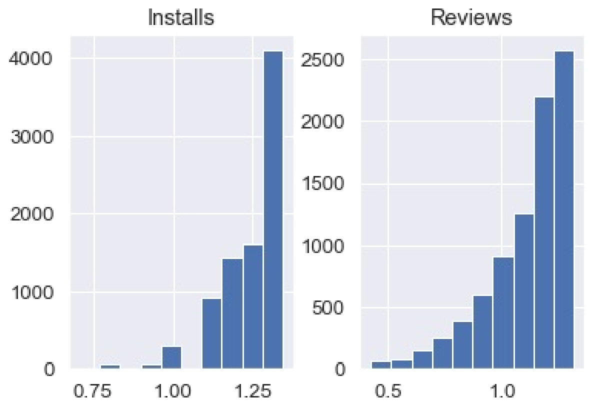
```
data.Reviews=data.Reviews.apply(func=np.log1p)
data.Installs=data.Installs.apply(func=np.log1p)
```

In [95]:

```
# Histogram for Installs & Reviews
data.hist(column=['Reviews','Installs'])
```

Out[95]:

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000009B5BA74CA0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000009B5BA8CEE0>]],
      dtype=object)
```



Linear Regression Model

In [110]:

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
linreg=LinearRegression()
from statsmodels.api import OLS
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error as ms
```

Next, we split 70% of the data to the training set while 30% of the data to test set using below code.

In [38]:

```
X=data.iloc[:,1:]
y=data.iloc[:,:1]
X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=0.30, random_state=1)
X_train.shape,X_test.shape
```

Out[38]:

```
((5945, 156), (2549, 156))
```

Building Model & Predicting the Ratings, also checking the difference between the actual value and predicted value.

In [39]:

```
# Linear Regression values for Actual & Predicted.
Model=linreg.fit(X_train, y_train)
predict=linreg.predict(X_test)

y_test=np.array(y_test)
predict=np.array(predict)

a=pd.DataFrame({'Actual':y_test.flatten(),'Predicted':predict.flatten()});a.head(10)
```

Out[39]:

	Actual	Predicted
0	3.7	4.292656
1	3.0	3.751739
2	3.2	3.999185
3	4.0	4.136414
4	4.2	4.103574
5	5.0	3.804942
6	3.9	4.015669
7	4.7	4.469564
8	4.5	4.279194
9	4.2	4.226456

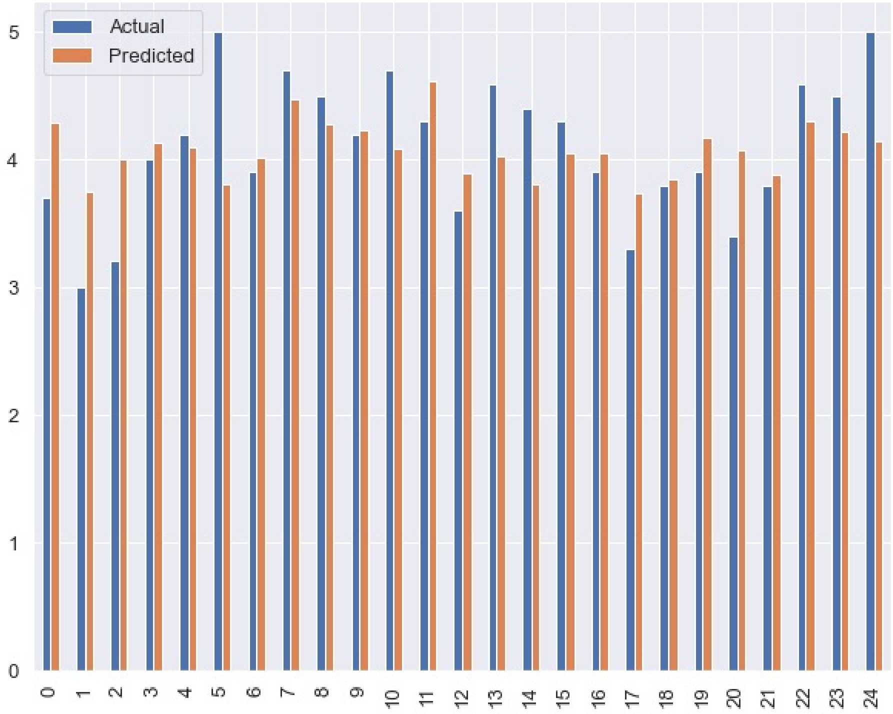
In Below figure we can observe here that the model has returned shows pretty good prediction results.

In [40]:

```
# Bar chart for Actual & Predicted Values
fig=a.head(25)
fig.plot(kind='bar',figsize=(10,8))
```

Out[40]:

```
<matplotlib.axes._subplots.AxesSubplot at 0xd4a9ef8b80>
```



Model Summary

In [41]:

```
results=OLS( y_train,X_train).fit()
results.summary()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\statsmodels\base\model.py:1362: RuntimeWarning: invalid v
alue encountered in true_divide
  return self.params / self.bse
C:\ProgramData\Anaconda3\lib\site-packages\scipy\stats\_distn_infrastructure.py:1932: RuntimeWarning
: invalid value encountered in less_equal
  cond2 = cond0 & (x <= _a)
```

Out[41]:

OLS Regression Results

Dep. Variable:	Rating	**R-squared (uncentered):**	0.987
Model:	OLS	**Adj. R-squared (uncentered):**	0.986
Method:	Least Squares	**F-statistic:**	3545.
Date:	Sun, 01 Nov 2020	**Prob (F-statistic):**	0.00
Time:	19:08:25	**Log-Likelihood:**	-4135.3
No. Observations:	5945	**AIC:**	8515.
Df Residuals:	5823	**BIC:**	9331.
Df Model:	122		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Reviews	0.1711	0.006	26.534	0.000	0.158	0.184
Size	-3.174e-07	3.42e-07	-0.929	0.353	-9.87e-07	3.53e-07
Installs	-0.1477	0.006	-22.845	0.000	-0.160	-0.135
Price	-1.006e-05	0.004	-0.003	0.998	-0.007	0.007
Category_AUTO_AND_VEHICLES	1.5142	0.119	12.713	0.000	1.281	1.748
Category_BEAUTY	1.5928	0.123	12.914	0.000	1.351	1.835
Category_BOOKS_AND_REFERENCE	1.5621	0.117	13.379	0.000	1.333	1.791
Category_BUSINESS	1.4594	0.116	12.596	0.000	1.232	1.687
Category_COMICS	1.5040	0.116	13.009	0.000	1.277	1.731
Category_COMMUNICATION	1.4334	0.116	12.344	0.000	1.206	1.661
Category_DATING	1.3599	0.117	11.603	0.000	1.130	1.590
Category_EDUCATION	2.5812	0.239	10.792	0.000	2.112	3.050
Category_ENTERTAINMENT	2.5334	0.240	10.567	0.000	2.063	3.003
Category_EVENTS	1.6074	0.122	13.221	0.000	1.369	1.846
Category_FAMILY	2.6287	0.232	11.344	0.000	2.174	3.083
Category_FINANCE	1.4554	0.116	12.586	0.000	1.229	1.682
Category_FOOD_AND_DRINK	1.4573	0.118	12.358	0.000	1.226	1.689
Category_GAME	2.9137	0.230	12.667	0.000	2.463	3.365
Category_HEALTH_AND_FITNESS	1.5082	0.116	13.009	0.000	1.281	1.735
Category_HOUSE_AND_HOME	1.4893	0.120	12.410	0.000	1.254	1.725
Category_LIBRARIES_AND_DEMO	1.5360	0.120	12.780	0.000	1.300	1.772
Category_LIFESTYLE	1.8885	0.224	8.418	0.000	1.449	2.328
Category_MAPS_AND_NAVIGATION	1.4218	0.117	12.107	0.000	1.192	1.652
Category_MEDICAL	1.4981	0.116	12.959	0.000	1.271	1.725
Category_NEWS_AND_MAGAZINES	1.4520	0.116	12.497	0.000	1.224	1.680
Category_PARENTING	2.4520	0.221	11.117	0.000	2.020	2.884
Category_PERSONALIZATION	1.5435	0.116	13.328	0.000	1.316	1.771
Category_PHOTOGRAPHY	1.4473	0.116	12.424	0.000	1.219	1.676
Category_PRODUCTIVITY	1.4641	0.116	12.628	0.000	1.237	1.691
Category_SHOPPING	1.4916	0.116	12.825	0.000	1.264	1.720
Category_SOCIAL	1.4937	0.116	12.846	0.000	1.266	1.722
Category_SPORTS	2.8604	0.417	6.862	0.000	2.043	3.677
Category_TOOLS	1.4295	0.115	12.410	0.000	1.204	1.655
Category_TRAVEL_AND_LOCAL	1.4473	0.117	12.410	0.000	1.219	1.676
Category_VIDEO_PLAYERS	2.6080	0.510	5.115	0.000	1.608	3.607
Category_WEATHER	1.4733	0.120	12.252	0.000	1.238	1.709
Type_Paid	-0.0398	0.033	-1.218	0.223	-0.104	0.024

Content Rating_Everyone	1.6381	0.228	7.177	0.000	1.191	2.086
Content Rating_Everyone 10+	1.6448	0.230	7.148	0.000	1.194	2.096
Content Rating_Mature 17+	1.6316	0.230	7.079	0.000	1.180	2.083
Content Rating_Teen	1.6360	0.229	7.157	0.000	1.188	2.084
Content Rating_Unrated	-1.269e-11	2e-11	-0.633	0.527	-5.2e-11	2.66e-11
Genres_Action;Action & Adventure	0.3606	0.166	2.175	0.030	0.036	0.686
Genres_Adventure	-0.0901	0.082	-1.105	0.269	-0.250	0.070
Genres_Adventure;Action & Adventure	0.2234	0.498	0.449	0.654	-0.752	1.199
Genres_Adventure;Brain Games	0.5573	0.497	1.121	0.263	-0.418	1.532
Genres_Adventure;Education	-7.242e-12	1.13e-11	-0.644	0.520	-2.93e-11	1.48e-11
Genres_Arcade	0.0509	0.061	0.838	0.402	-0.068	0.170
Genres_Arcade;Action & Adventure	0.2551	0.182	1.401	0.161	-0.102	0.612
Genres_Arcade;Pretend Play	0.5790	0.497	1.164	0.244	-0.396	1.554
Genres_Art & Design	3.2654	0.243	13.455	0.000	2.790	3.741
Genres_Art & Design;Creativity	2.2902	0.256	8.958	0.000	1.789	2.791
Genres_Art & Design;Pretend Play	1.7128	0.370	4.632	0.000	0.988	2.438
Genres_Auto & Vehicles	1.5142	0.119	12.713	0.000	1.281	1.748
Genres_Beauty	1.5928	0.123	12.914	0.000	1.351	1.835
Genres_Board	0.0379	0.098	0.388	0.698	-0.154	0.229
Genres_Board;Action & Adventure	0.2427	0.356	0.682	0.495	-0.455	0.941
Genres_Board;Brain Games	0.4140	0.163	2.538	0.011	0.094	0.734
Genres_Board;Pretend Play	0.9925	0.497	1.995	0.046	0.017	1.968
Genres_Books & Reference	1.5621	0.117	13.379	0.000	1.333	1.791
Genres_Books & Reference;Education	0.3093	0.356	0.869	0.385	-0.389	1.007
Genres_Business	1.4594	0.116	12.596	0.000	1.232	1.687
Genres_Card	-0.0346	0.096	-0.362	0.717	-0.222	0.153
Genres_Card;Action & Adventure	0.0829	0.357	0.232	0.816	-0.617	0.782
Genres_Card;Brain Games	-4.437e-12	7.6e-12	-0.584	0.559	-1.93e-11	1.05e-11
Genres_Casino	0.0572	0.100	0.570	0.569	-0.139	0.254
Genres_Casual	0.1527	0.086	1.765	0.078	-0.017	0.322
Genres_Casual;Action & Adventure	0.2486	0.168	1.476	0.140	-0.082	0.579
Genres_Casual;Brain Games	0.6370	0.172	3.700	0.000	0.299	0.975
Genres_Casual;Creativity	0.4805	0.212	2.262	0.024	0.064	0.897
Genres_Casual;Education	0.3466	0.356	0.974	0.330	-0.351	1.045
Genres_Casual;Music & Video	0.4150	0.356	1.165	0.244	-0.284	1.114
Genres_Casual;Pretend Play	0.2881	0.136	2.112	0.035	0.021	0.556
Genres_Comics	1.5040	0.116	13.009	0.000	1.277	1.731
Genres_Comics;Creativity	1.324e-11	2.08e-11	0.637	0.524	-2.75e-11	5.4e-11
Genres_Communication	1.4334	0.116	12.344	0.000	1.206	1.661
Genres_Communication;Creativity	0.4780	0.497	0.962	0.336	-0.496	1.452
Genres_Dating	1.3599	0.117	11.603	0.000	1.130	1.590
Genres_Education	0.5115	0.087	5.907	0.000	0.342	0.681
Genres_Education;Action & Adventure	0.7892	0.259	3.048	0.002	0.282	1.297
Genres_Education;Brain Games	0.3987	0.297	1.343	0.179	-0.183	0.981
Genres_Education;Creativity	0.7943	0.217	3.655	0.000	0.368	1.220
Genres_Education;Education	0.5934	0.124	4.793	0.000	0.351	0.836
Genres_Education;Music & Video	0.4442	0.294	1.509	0.131	-0.133	1.021
Genres_Education;Pretend Play	0.6480	0.159	4.068	0.000	0.336	0.960
Genres_Educational	0.0282	0.125	0.226	0.821	-0.217	0.273
Genres_Educational;Action & Adventure	0.6007	0.497	1.209	0.227	-0.374	1.575
Genres_Educational;Brain Games	0.5465	0.234	2.334	0.020	0.088	1.005
Genres_Educational;Creativity	0.3518	0.259	1.360	0.174	-0.155	0.859
Genres_Educational;Education	0.4869	0.119	4.097	0.000	0.254	0.720
Genres_Educational;Pretend Play	0.3942	0.183	2.158	0.031	0.036	0.752
Genres_Entertainment	0.2674	0.086	3.126	0.002	0.100	0.435

Genres_Entertainment;Action & Adventure	0.3935	0.295	1.336	0.182	-0.184	0.971
Genres_Entertainment;Brain Games	0.4464	0.203	2.198	0.028	0.048	0.845
Genres_Entertainment;Creativity	0.8228	0.497	1.656	0.098	-0.151	1.797
Genres_Entertainment;Education	0.6540	0.497	1.316	0.188	-0.320	1.628
Genres_Entertainment;Music & Video	0.3056	0.148	2.059	0.040	0.015	0.597
Genres_Entertainment;Pretend Play	0.0354	0.356	0.099	0.921	-0.663	0.733
Genres_Events	1.6074	0.122	13.221	0.000	1.369	1.846
Genres_Finance	1.4554	0.116	12.586	0.000	1.229	1.682
Genres_Food & Drink	1.4573	0.118	12.358	0.000	1.226	1.689
Genres_Health & Fitness	1.5082	0.116	13.009	0.000	1.281	1.735
Genres_Health & Fitness;Action & Adventure	1.073e-12	1.86e-12	0.578	0.563	-2.56e-12	4.71e-12
Genres_Health & Fitness;Education	0.5774	0.497	1.162	0.245	-0.397	1.552
Genres_House & Home	1.4893	0.120	12.410	0.000	1.254	1.725
Genres_Libraries & Demo	1.5360	0.120	12.780	0.000	1.300	1.772
Genres_Lifestyle	1.0318	0.182	5.674	0.000	0.675	1.388
Genres_Lifestyle;Education	1.602e-12	2.53e-12	0.633	0.527	-3.36e-12	6.56e-12
Genres_Lifestyle;Pretend Play	0.8567	0.336	2.546	0.011	0.197	1.516
Genres_Maps & Navigation	1.4218	0.117	12.107	0.000	1.192	1.652
Genres_Medical	1.4981	0.116	12.959	0.000	1.271	1.725
Genres_Music	-0.2098	0.152	-1.377	0.169	-0.508	0.089
Genres_Music & Audio;Music & Video	0.7383	0.497	1.485	0.138	-0.236	1.713
Genres_Music;Music & Video	0.6443	0.497	1.296	0.195	-0.331	1.619
Genres_News & Magazines	1.4520	0.116	12.497	0.000	1.224	1.680
Genres_Parenting	0.8196	0.150	5.446	0.000	0.525	1.115
Genres_Parenting;Brain Games	0.3492	0.401	0.870	0.384	-0.438	1.136
Genres_Parenting;Education	0.4108	0.255	1.611	0.107	-0.089	0.911
Genres_Parenting;Music & Video	0.8725	0.202	4.312	0.000	0.476	1.269
Genres_Personalization	1.5435	0.116	13.328	0.000	1.316	1.771
Genres_Photography	1.4473	0.116	12.424	0.000	1.219	1.676
Genres_Productivity	1.4641	0.116	12.628	0.000	1.237	1.691
Genres_Puzzle	0.3607	0.088	4.090	0.000	0.188	0.534
Genres_Puzzle;Action & Adventure	0.4291	0.497	0.864	0.388	-0.545	1.403
Genres_Puzzle;Brain Games	0.4750	0.150	3.159	0.002	0.180	0.770
Genres_Puzzle;Creativity	0.4356	0.356	1.224	0.221	-0.262	1.134
Genres_Puzzle;Education	0.8892	0.497	1.790	0.074	-0.085	1.863
Genres_Racing	-0.1142	0.079	-1.449	0.148	-0.269	0.040
Genres_Racing;Action & Adventure	0.4249	0.175	2.425	0.015	0.081	0.768
Genres_Racing;Pretend Play	0.9796	0.497	1.970	0.049	0.005	1.954
Genres_Role Playing	0.2013	0.090	2.232	0.026	0.024	0.378
Genres_Role Playing;Action & Adventure	0.3129	0.258	1.213	0.225	-0.193	0.819
Genres_Role Playing;Brain Games	0.3898	0.497	0.784	0.433	-0.584	1.364
Genres_Role Playing;Pretend Play	0.2192	0.234	0.937	0.349	-0.240	0.678
Genres_Shopping	1.4916	0.116	12.825	0.000	1.264	1.720
Genres_Simulation	0.2402	0.089	2.684	0.007	0.065	0.416
Genres_Simulation;Action & Adventure	0.4048	0.216	1.872	0.061	-0.019	0.829
Genres_Simulation;Education	0.2441	0.290	0.842	0.400	-0.324	0.812
Genres_Simulation;Pretend Play	0.3376	0.294	1.147	0.251	-0.239	0.914
Genres_Social	1.4937	0.116	12.846	0.000	1.266	1.722
Genres_Sports	0.0569	0.349	0.163	0.870	-0.627	0.741
Genres_Sports;Action & Adventure	0.3056	0.295	1.037	0.300	-0.272	0.883
Genres_Strategy	0.1287	0.097	1.326	0.185	-0.062	0.319
Genres_Strategy;Action & Adventure	0.5422	0.356	1.523	0.128	-0.156	1.240
Genres_Strategy;Creativity	0.1793	0.497	0.361	0.718	-0.795	1.154
Genres_Strategy;Education	0.8785	0.497	1.768	0.077	-0.096	1.853
Genres_Tools	1.4295	0.115	12.410	0.000	1.204	1.655

Genres_Tools;Education	0	0	nan	nan	0	0
Genres_Travel & Local	1.4473	0.117	12.410	0.000	1.219	1.676
Genres_Travel & Local;Action & Adventure	0	0	nan	nan	0	0
Genres_Trivia	-0.0460	0.114	-0.404	0.687	-0.269	0.177
Genres_Video Players & Editors	0.2151	0.464	0.463	0.643	-0.695	1.125
Genres_Video Players & Editors;Creativity	0.0789	0.422	0.187	0.852	-0.749	0.907
Genres_Video Players & Editors;Music & Video	0.1538	0.331	0.464	0.642	-0.495	0.803
Genres_Weather	1.4733	0.120	12.252	0.000	1.238	1.709
Genres_Word	0.1840	0.111	1.653	0.098	-0.034	0.402

Omnibus:	1965.043	**Durbin-Watson:**	1.966
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	9494.515
Skew:	-1.525	**Prob(JB):**	0.00
Kurtosis:	8.388	**Cond. No.**	4.06e+20

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 3.59e-29. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

In [42]:

```
print('R2_Score=',r2_score(y_test,predict))
print('Root Mean Squared Error=',np.sqrt(ms(y_test,predict)))
print('Prediction Error Percentage is',round((0.50/np.mean(y_test))*100))
```

```
R2_Score= 0.14132279149648597
Root Mean Squared Error= 0.5074075540316355
Prediction Error Percentage is 12.0
```

Summary Interpretation

1. A large F-statistic will corresponds to a statistically significant p-value ($p < 0.05$). In the data, the F-statistic equals 3412. that leads to less P_Value which says that at least one of the predictor variables is significantly related to the outcome variable. 2. From the output above, the adjusted R2 is 0.986, meaning that the observed and the predicted outcome values are highly correlated, which is very good.3. The prediction error RMSE (Root Mean Squared Error), representing the average difference between the observed known outcome values in the test data and the predicted outcome seems to be 0.50 which is very good thus represents the error rate of 12%.