Google Play Store Apps

Data Analysis and Visualization And Data Mining Project Report

Department of Computer Science
Shaheed Sukhdev College of Business Studies
University of Delhi

Submitted by :
Ayush Gupta (20512)
Jatin Paul (20524)
Saurav Ganguly(20545)
Vatsal Singh (20555)

Supervisor :
Mrs. Anamika Gupta

Contents

ACKNOWLEDGEMENT

We express our sincere indebtedness towards our guide Mrs. Anamika Gupta, Department of Computer Science, Shaheed Sukhdev College of Business Studies for her invaluable guidance, suggestions, and supervision throughout the work. Without her kind patronage and guidance the project would not have taken shape. We would also like to express our gratitude and sincere regards for her kind approval of the project, time to time counseling and advice.

Project Description

Mobile apps are everywhere. They are easy to create and can be lucrative. Because of these two factors, more and more apps are being developed. In this project, We will do a comprehensive analysis of the Android app market by comparing over ten thousand apps in Google Play across different categories. We'll look for insights in the data to devise strategies to drive growth and retention. The data for this project was scraped from the Google Play website. While there are many popular datasets for Apple App Store, there aren't many for Google Play apps, which is partially due to the increased difficulty in scraping the latter as compared to the former. The data files are as follows:

- **googleplaystore.csv** :contains all the details of the apps on Google Play. These are the features that describe an app like App name, Category, Rating, Reviews, Size, Installs, Type, Price(if any), Content Rating, Genres, Last updated, Current Ver, and Android Ver.
- **googleplaystore_user_reviews.csv** :contains 100 reviews for each app, most helpful first. The text in each review has been pre-processed, passed through a sentiment analyzer engine and tagged with its sentiment score. The data file googleplaystore_user_reviews contains data fields like App name and their respective translated reviews, Sentiment, sentiment_polarity and sentiment_subjectivity. This datafile is ideal for Sentiment Analysis of the user reviews on various apps listed on Play store.

Source: Kaggle
Link: https://www.kaggle.com/datasets/lava18/google-play-store-apps

Inspiration

The Play Store apps data has enormous potential to drive app-making businesses to success. Actionable insights can be drawn for developers to work on and capture the Android market!

Data Description

The dataset include the following features:

- **App:** Application name
- **Category:** Category the app belongs to
- **Rating:** Overall user rating of the app (as when scraped)
- **Reviews:** Number of user reviews for the app (as when scraped)
- **Size:** Size of the app (as when scraped)
- Installs: Number of user downloads/installs for the app (as when scraped)
- **Type:** Paid or Free
- **Price:** Price of the app (as when scraped)
- **Content Rating:** Age group the app is targeted at - Children / Mature 21+ / Adult
- **Genre:** An app can belong to multiple genres (apart from its main category). For eg, a musical family game will belong to
- **Current Ver:** Current version of the app (as when scraped)
- **Android Ver:** Latest compatible android version

Packages Required:

Python Libraries for Data Analysis and Visualization

PACKAGES

The packages used in this analysis are:

- pandas
- numpy
- matplotlib

Load and Describe Dataset:


```
# Importing dependencies
import pandas as pd
import numpy as np
```

```
# Importing Dataset and displaying data
df = pd.read_csv(r'app.csv')
df
```

Displaying top 5 rows of the dataset:

```
# Displaying top 5 rows in the dataset
df.head()
```

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Cor Ra
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Eve
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Eve
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free	0	Eve
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0	
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0	Eve

- 2018

```
# Making a copy of the dataset
df_copy = df.copy()
df_copy.head()
```

Analyzing the data:

```
# Displaying all the columns/attributes of the dataset
df_copy.columns
```

```
Index(['App', 'Category', 'Rating', 'Reviews', 'Size', 'Installs', 'Type',
       'Price', 'Content Rating', 'Genres', 'Last Updated', 'Current Ver',
       'Android Ver'],
      dtype='object')
```

```
# Checking the index/number of rows in the dataset
df_copy.index
```

```
RangeIndex(start=0, stop=10841, step=1)
```

```
# Displaying values
df_copy.values
```

```
array([['Photo Editor & Candy Camera & Grid & ScrapBook',
        'ART_AND_DESIGN', 4.1, ..., 'January 7, 2018', '1.0.0',
        '4.0.3 and up'],
       ['Coloring book moana', 'ART_AND_DESIGN', 3.9, ...,
        'January 15, 2018', '2.0.0', '4.0.3 and up'],
       ['U Launcher Lite – FREE Live Cool Themes, Hide Apps',
        'ART_AND_DESIGN', 4.7, ..., 'August 1, 2018', '1.2.4',
        '4.0.3 and up'],
       ...,
       ['Parkinson Exercices FR', 'MEDICAL', nan, ...,
```

```
       'January 20, 2017', '1', '2.2 and up'],
      ['The SCP Foundation DB fr nn5n', 'BOOKS_AND_REFERENCE', 4.5, ...,
       'January 19, 2015', 'Varies with device', 'Varies with device'],
      ['iHoroscope - 2018 Daily Horoscope & Astrology', 'LIFESTYLE',
       4.5, ..., 'July 25, 2018', 'Varies with device',
       'Varies with device']], dtype=object)
```

```
# Displaying a particular column in the dataset
App_category = df_copy["Category"]
App_category
```

```
0             ART_AND_DESIGN
1             ART_AND_DESIGN
2             ART_AND_DESIGN
3             ART_AND_DESIGN
4             ART_AND_DESIGN
                ...
10836                FAMILY
10837                FAMILY
10838               MEDICAL
10839    BOOKS_AND_REFERENCE
10840             LIFESTYLE
Name: Category, Length: 10841, dtype: object
```

```
# Displaying the unique values from the "Category" column
App_category_unique = App_category.unique()
App_category_unique
```

```
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',
       '1.9'], dtype=object)
```

```
# Length of the unique values in the "Category" column
len(App_category_unique)
```

```
34
```

Data Cleaning

```
# install the pandas library
pip install pandas;

# import pandas library
import pandas as pd;

# read the apps data in a variable
dataset = pd.read_csv("apps.csv")

# No. of rows with na values
dataset.isna().sum()
```

```
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
```

```
# Removing rows with na values in Rating
cleanDataset = dataset.dropna()

# No. of rows removed
dataset.shape[0] - cleanDataset.shape[0]
```

```
1481
```

```
# No. of rows with na values
cleanDataset.isna().sum()
```

```
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
```

```
# Getting the Unique values for the Category Column
cleanDataset.Category.unique()
```

```
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)
```

```
# Getting the Unique values for the Size Column
cleanDataset.Size.unique()
```

```
array(['19M', '14M', '8.7M', '25M', '2.8M', '5.6M', '29M', '33M',
'3.1M',
                                    '28M', '12M', '20M', '21M', '37M',
                                    '5.5M', '17M', '39M', '31M',
                                    '4.2M', '23M', '6.0M', '6.1M',
'26M',                              '4.6M', '9.2M', '5.2M', '11M',
                                    '24M', 'Varies with device',
'5.7M',                             '9.4M', '15M', '10M', '1.2M',

                                    '8.0M', '7.9M', '56M', '57M',
                                    '35M', '54M', '201k', '3.6M',

                                    '8.6M', '2.4M', '27M', '2.7M',
                                    '2.5M', '7.0M', '16M', '3.4M',
'51M',                              '8.9M', '3.9M', '2.9M', '38M',
                                    '32M', '5.4M', '18M', '1.1M',
'40M',                              '2.2M', '4.5M', '9.8M', '52M',
                                    '9.0M', '6.7M', '30M', '2.6M',
                                    '7.1M', '22M', '6.4M', '3.2M',
                                    '8.2M', '4.9M', '9.5M', '5.0M',
                                    '5.9M', '13M', '73M', '6.8M',
```

'3.5M', '4.0M', '2.3M', '2.1M',
'42M', '9.1M', '55M', '23k',
'7.3M', '6.5M', '1.5M', '7.5M',

'41M', '48M', '8.5M', '46M',
'8.3M', '4.3M', '4.7M', '3.3M',

'7.8M', '8.8M', '6.6M', '5.1M',
'61M', '66M', '79k', '8.4M',

'77M',

'97M',

'334k',

'8.5k',

'75M',

'169k',

'93k',

'3.7M', '118k', '44M', '695k',
'1.6M', '6.2M', '53M', '1.4M',
'3.0M', '7.2M', '5.8M', '3.8M',
'9.6M', '45M', '63M', '49M',

'4.4M', '70M', '9.3M', '8.1M',
'36M', '6.9M', '7.4M', '84M',

'2.0M', '1.9M', '1.8M', '5.3M',
'47M', '556k', '526k', '76M',
'7.6M', '59M', '9.7M', '78M',
'72M', '43M', '7.7M', '6.3M',

'93M', '65M', '79M', '100M', '58M',
'50M', '68M', '64M', '34M',
'67M', '60M', '94M', '9.9M',
'232k', '99M', '624k', '95M',

'41k', '292k', '80M', '1.7M',
'10.0M', '74M', '62M', '69M',

'98M', '85M', '82M', '96M', '87M',
'71M', '86M', '91M', '81M',
'92M', '83M', '88M', '704k',
'862k', '899k', '378k', '4.8M',
'266k', '375k', '1.3M', '975k',
'980k', '4.1M', '89M', '696k',
'544k', '525k', '920k', '779k',
'853k', '720k', '713k', '772k',
'318k', '58k', '241k', '196k',
'857k', '51k', '953k', '865k',
'251k', '930k', '540k', '313k',
'746k', '203k', '26k', '314k',
'239k', '371k', '220k', '730k',
'756k', '91k', '293k', '17k',
'74k', '14k', '317k', '78k',
'924k', '818k', '81k', '939k',

'45k', '965k', '90M', '545k',
'61k', '283k', '655k', '714k',

'872k', '121k', '322k', '976k',
'206k', '954k', '444k', '717k',
'210k', '609k', '308k', '306k',
'175k', '350k', '383k', '454k',
'1.0M', '70k', '812k', '442k',
'842k', '417k', '412k', '459k',
'478k', '335k', '782k', '721k',
'430k', '429k', '192k', '460k',
'728k', '496k', '816k', '414k',
'506k', '887k', '613k', '778k',
'683k', '592k', '186k', '840k',
'647k', '373k', '437k', '598k',
'716k', '585k', '982k', '219k',
'55k', '323k', '691k', '511k',
'951k', '963k', '25k', '554k',

```
'351k', '27k', '82k', '208k',       '847k', '948k', '811k', '270k',
'551k', '29k', '103k', '116k',      '48k', '523k', '784k', '280k',
'153k', '209k', '499k', '173k',     '24k', '892k', '154k', '18k',
'597k', '809k', '122k', '411k',     '33k', '860k', '364k', '387k',
'400k', '801k', '787k', '50k',      '626k', '161k', '879k', '39k',
'643k', '986k', '516k', '837k',     '170k', '141k', '160k', '144k',
'780k', '20k', '498k', '600k',      '143k', '190k', '376k', '193k',
'656k', '221k', '228k', '176k',     '473k', '246k', '73k', '253k',
'34k', '259k', '164k', '458k',      '957k', '420k', '72k', '404k',
'629k', '28k', '288k', '775k',      '470k', '226k', '240k', '89k',
'785k', '636k', '916k', '994k',     '234k', '257k', '861k', '467k',
'309k', '485k', '914k', '903k',     '676k', '552k', '582k',
'608k', '500k', '54k', '562k',
 '619k'],
      dtype=object)
```

```
# getting the shape of the cleanDataset
cleanDataset.shap
```

```
e (9360, 13)
```

```
# Dropping Rows having the value Varies with device in Size Columns
cleanDataset=cleanDataset.drop(cleanDataset.index[cleanDataset['Size'
]
=='Varies with device'])

# getting the shape of the cleanDataset to see the changes
cleanDataset.shap

e (7723, 13)

# Getting the Unique values for the Installs Column
cleanDataset.Installs.unique()

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

# Getting the Unique values for the Type Column
cleanDataset.Type.unique()

array(['Free', 'Paid'],

dtype=object)

# Getting the Unique values for the Price Column
cleanDataset.Price.unique()

array(['0', '$4.99', '$6.99', '$7.99', '$3.99', '$5.99', '$2.99',
'$1.99',
       '$9.99', '$0.99', '$9.00', '$5.49', '$10.00', '$24.99',
'$11.99',
       '$79.99', '$16.99', '$14.99', '$29.99', '$12.99', '$3.49',
       '$10.99', '$7.49', '$1.50', '$19.99', '$15.99', '$33.99',
'$39.99',
       '$2.49', '$4.49', '$1.70', '$1.49', '$3.88', '$399.99',
'$17.99',
       '$400.00', '$3.02', '$1.76', '$4.84', '$4.77', '$1.61',
'$1.59',
       '$6.49', '$1.29', '$299.99', '$379.99', '$37.99', '$18.99',
       '$389.99', '$8.49', '$1.75', '$14.00', '$2.00', '$3.08',
'$2.59',
       '$19.40', '$15.46', '$8.99', '$3.04', '$13.99', '$4.29',
'$3.28',
       '$4.60', '$1.00', '$2.90', '$1.97', '$2.56', '$1.20'],
dtype=object)

# Getting the Unique values for the Content Rating Column
cleanDataset['Content Rating'].unique()

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

```
# Getting the Unique values for the Current Ver Column
cleanDataset['Current Ver'].unique()[:100]
```

```
      array(['1.0.0', '2.0.0', '1.2.4', 'Varies with device', '1.1',
                                                                '1.0',
      '6.1.61.1', '2.9.2', '2.8', '1.0.4', '1.0.15', '3.8', '1.2.3',
       '3.1', '2.2.5', '5.5.4', '4.0', '2.2.6.2', '1.1.3', '1.5',
'1.0.8',
       '1.03', '6.0', '6.7.12.2018', '1.2', '2.20', '1.1.0', '1.6',
'2.1',                                '1.0.9', '1.3', '1', '2.0.1',
                                      '1.46', '1.6.1', '11.0', '3.0',
                                      '1.7.1', '2.5.1', '1.0.1',
                                      '2.493', '1.9.1', '1.7',
                                      '2.20 Build 02', '1.37', '0.2.1',
                                      '4.47.3', '1.9.7', '2.2.21',
                                      '1.79', '2.3.5.1', '8.31',
                                      '1.1.5.0', '10.0.2', '1.10.3',
'3.20.1',
       '1.0.3', '1.4', '2.8.2', '4.0.3', '1.40', '1.5.18', '2.3.4',
       '2.17', '6.10.1', '2.3.0', '1.0.6', '1.9', '3.0.1', '3.3.9',
       '1.20', '2.3.09', '1.4.2', '18.5', '1.2.13', '1.0.2.0',
'3.1.89',
       '2.2.0', '1.9.2', '1.3.2', '3.2.1', '2.0.075', '1.911805270',
       '9.1.363', '1.1.6', '2.3.18', '15.0', '18.05.31+530', '5.0.6',
       '3.12', '2.0', '1.28', '6.0.8', '14.0', '3.05', '2.5.3',
'7.0.4.6',
       '1.15', '3.1.7.9', '3.9.1'], dtype=object)
```

```
# Dropping Rows having the value Varies with device in Size Columns
cleanDataset=cleanDataset.drop(cleanDataset.index[cleanDataset['Curre
n t Ver']=='Varies with device'])
```

```
# Getting the Unique values for the Android Ver Column
cleanDataset['Android Ver'].unique()
```

```
array(['4.0.3 and up', '4.4 and up', '2.3 and up', '4.2 and up',
       '3.0 and up', '4.1 and up', '4.0 and up', '2.2 and up',
       '6.0 and up', '5.0 and up', '1.6 and up', '2.1 and up',
       '1.5 and up', '7.0 and up', '4.3 and up', '4.0.3 - 7.1.1',
       '2.0 and up', '2.3.3 and up', '3.2 and up', '4.4W and up',
       '5.1 and up', '7.1 and up', '7.0 - 7.1.1', 'Varies with
device',
       '8.0 and up', '5.0 - 8.0', '3.1 and up', '2.0.1 and up',
       '4.1 - 7.1.1', '5.0 - 6.0', '1.0 and up'], dtype=object)
```

```
# Dropping Rows having the value Varies with device in Size Columns
cleanDataset=cleanDataset.drop(cleanDataset.index[cleanDataset['Andro
i d Ver']=='Varies with device'])
```

```
# Dropping the Last Updated Column
cleanDataset=cleanDataset.drop(['Last Updated'],axis=1)
```

```
# again getting the shape of cleanDataset to view the changes applied
in this set
cleanDataset.shape
```

```
(7637, 12)
```

```
# we take a copy of cleanDataset into a cleanData variable
cleanData=cleanDataset
```

```
# view the cleanData set
cleanData
```

```
                                                 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
4     Pixel Draw - Number Art Coloring Book
ART_AND_DESIGN
5                            Paper flowers instructions
ART_AND_DESIGN
...                                                ...
...
10832                                        FR Tides
WEATHER
10833                                    Chemin (fr)
BOOKS_AND_REFERENCE
10834                                  FR Calculator
FAMILY
10836                                 Sya9a Maroc - FR
FAMILY
10837                   Fr. Mike Schmitz Audio
Teachings FAMILY

       Rating Reviews Size     Installs Type Price Content Rating \
0         4.1     159   19M     10,000+  Free     0       Everyone
1         3.9     967   14M    500,000+  Free     0       Everyone
2         4.7   87510  8.7M  5,000,000+  Free     0       Everyone
4         4.3     967  2.8M    100,000+  Free     0       Everyone
5         4.4     167  5.6M     50,000+  Free     0       Everyone
...       ...     ...   ...         ...   ...   ...           ...
10832     3.8    1195  582k    100,000+  Free     0       Everyone
10833     4.8      44  619k      1,000+  Free     0       Everyone
10834     4.0       7  2.6M        500+  Free     0       Everyone
10836     4.5      38   53M      5,000+  Free     0       Everyone
10837     5.0       4  3.6M        100+  Free     0       Everyone

                        Genres Current Ver   Android Ver
0                   Art & Design       1.0.0  4.0.3 and up
1      Art & Design;Pretend Play       2.0.0  4.0.3 and up
2                   Art & Design       1.2.4  4.0.3 and up
4        Art & Design;Creativity         1.1    4.4 and up
```

```
5                   Art & Design         1.0    2.3 and up
...                          ...         ...           ...
10832                    Weather         6.0    2.1 and up
10833          Books & Reference         0.8    2.2 and up
10834                  Education       1.0.0    4.1 and up
10836                  Education        1.48    4.1 and up
10837                  Education         1.0    4.1 and up

[7637 rows x 12 columns]
```

```
# getting the datatype of the columns of cleanData
cleanData.dtypes
```

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

```
# we change the type of columns of some columns to string
cleanData['Size']=cleanData['Size'].astype('string');
cleanData['Installs']=cleanData['Installs'].astype('string');
cleanData['Price']=cleanData['Price'].astype('string');
cleanData['Android Ver']=cleanData['Android Ver'].astype('string');
```

```
# we can see the changes in the type of the columns of the dataset
cleanData.dtypes
```

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

```
# we replace the dolloar sign with the empty value in the price column
cleanData['Price']=cleanData['Price'].str.replace('$',"");
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1:
FutureWarning: The default value of regex will change from True to
False in a future version. In addition, single character regular
expressions will *not* be treated as literal strings when
regex=True.
  """Entry point for launching an IPython kernel.
```

```
# we then covert the type of price column to float dtype and also
renamed the column to Price in Dollars
cleanData['Price']=cleanData['Price'].astype('float');
cleanData.rename(columns={'Price':"Price in Dollars"},inplace=True);
```

```
# view the unique values of Price in Dollars Column
cleanData['Price in Dollars'].unique()
```

```
array([ 0.   ,   4.99,   6.99,   7.99,   3.99,   5.99,   2.99,   1.99,
         9.99,   0.99,    9. ,   5.49,   10. ,  24.99,  11.99,  79.99,
        16.99,  14.99,  29.99,  12.99,   3.49,  10.99,   7.49,   1.5 ,
        19.99,  15.99,  33.99,  39.99,   2.49,   4.49,   1.7 ,   1.49,
         3.88, 399.99,  17.99, 400. ,    3.02,   1.76,   4.84,   4.77,
         1.61,   1.59,   6.49,   1.29,  299.99 379.99,  37.99,  18.99,
                                              ,
       389.99,   8.49,   1.75,  14. ,     2. ,   3.08,   2.59,  19.4 ,
        15.46,   8.99,   3.04,  13.99,   4.29,   3.28,   4.6 ,    1. ,
         2.9 ,   1.97,   2.56,   1.2
                                 ])
```

```
# Replace "+" , "," with empty values in installs column and then
change the type to int
cleanData['Installs']=cleanData['Installs'].str.replace('+',"");
cleanData['Installs']=cleanData['Installs'].str.replace(",","");
cleanData['Installs']=cleanData['Installs'].astype('int')
```

```
# view the unique values of intalls column
cleanData['Installs'].unique()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1:
FutureWarning: The default value of regex will change from True to
False in a future version. In addition, single character regular
expressions will *not* be treated as literal strings when
regex=True.
  """Entry point for launching an IPython kernel.
```

```
array(      10000,     500000,    5000000,     100000,      50000,
[
          1000000,   10000000,       5000,  100000000,   50000000,
             1000,  500000000,        100,        500,         10,
        1000000000          5,         50,          1])
                 ,
```

```
# we replace the " and up" with empty values in the Android Ver column
cleanData['Android Ver']=cleanData['Android Ver'].str.replace(" and
up","");
```

```
# and then rename the column to Min Sup Android Ver
cleanData.rename(columns={'Android Ver':"Min Sup Android
Ver"},inplace=True);

# see the cleanData
cleanData
```

```
                                                    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
4     Pixel Draw - Number Art Coloring Book
ART_AND_DESIGN
5                            Paper flowers instructions
ART_AND_DESIGN
...                                                 ...
...
10832                                        FR Tides
WEATHER
10833                                    Chemin (fr)
BOOKS_AND_REFERENCE
10834                                  FR Calculator
FAMILY
10836                                 Sya9a Maroc - FR
FAMILY
10837                  Fr. Mike Schmitz Audio
Teachings FAMILY

       Rating Reviews  Size Installs Type  Price in Dollars Content
Rating \
```

```
10834     4.        7  2.6M       500  Free                0.0
Everyone  0
10836     4.       38   53M      5000  Free                0.0
Everyon   5
e 10837              4  3.6M       100  Free                0.0
          5.
          0
Everyone
```

		Genre	Current Ver	Min Sup	Android Ver
0		s Art & Design	1.0.0		4.0.3
1	Art & Design;Pretend Play		2.0.0		4.0.3
2		Art & Design	1.2.4		4.0.3
4	Art & Design;Creativity		1.1		4.4
5		Art & Design	1.0		2.3
...		...	..		..
10832		Weather	.		.
			6.		2.
			0		1
10833		Books & Reference	0.8		2.2
10834		Education	1.0.0		4.1
10836		Education	1.48		4.1
10837		Education	1.0		4.1

[7637 rows x 12 columns]

```
# Now we export the CleanData file to csv file in google drive
from google.colab import drive

drive.mount('/content/drive',force_remount=True)
path = '/content/drive/My Drive/Data Analysis
Project/cleanedAppsData.csv'

with open(path, 'w', encoding = 'utf-8-sig') as f:
  cleanData.to_csv(f)
```

```
Mounted at /content/drive
```

Data Visualization and Exploratory Data Analysis

Click here to view the python file in github

```
# Install pandas library
 pip install pandas;

 # Import pandas
 import pandas as pd;

 # we will read the csv file in a variable cleanData
 cleanData=pd.read_csv("E:\Folder/5th_Semester/Data_Analysis/Practical
 s
 /Github Saurav Repo/Data/cleanedAppsData.csv");

 # we describe the cleanData variable
 cleanData.describe()
```

```
         Unnamed: 0       Rating       Reviews      Installs Price in
Dollars
count   7637.000000  7637.000000  7.637000e+03  7.637000e+03
7637.000000
mean    5441.113264     4.172502  2.944081e+05  8.165624e+06
1.136429
std     3115.145916     0.546138  1.873040e+06  4.940853e+07
17.504658
min        0.000000     1.000000  1.000000e+00  1.000000e+00
0.00000
0
25%     2695.000000     4.000000  1.050000e+02  1.000000e+04
0.00000
0
50%     5441.000000     4.300000  2.221000e+03  1.000000e+05
0.00000
0
75%     8163.000000     4.500000  3.788200e+04  1.000000e+06
0.00000
0
```

```
max    10837.000000     5.000000 4.489389e+07  1.000000e+09
400.000000

# getting the info of cleanData
cleanData.info()

<class
'pandas.core.frame.DataFrame'>
RangeIndex: 7637 entries, 0 to 7636
Data columns (total 13 columns):
 #   Column               Non-Null Count Dtype
---  ------               -------------- -----
 0   Unnamed: 0           7637 non-null   int64
 1   App                  7637 non-null   object
 2   Category             7637 non-null   object
 3   Rating               7637 non-null   float64
 4   Reviews              7637 non-null   int64
 5   Size                 7637 non-null   object
 6   Installs             7637 non-null   int64
 7   Type                 7637 non-null   object
 8   Price in Dollars     7637 non-null   float64
 9   Content Rating       7637 non-null   object
 10 Genres                7637 non-null   object
 11 Current Ver           7637 non-null   object
 12 Min Sup Android Ver   7637 non-null   object
dtypes: float64(2), int64(3),
object(8) memory usage: 775.8+ KB

# getting the median of the installs column
cleanData['Installs'].median(

) 100000.0

# getting the sum of the data according to category
cleanData.groupby('Category').sum()

                    Unnamed: 0  Rating     Reviews     Installs  \
Category
ART_AND_DESIGN           77158   249.6      871576     49228100
AUTO_AND_VEHICLES       168486   257.1      720366     33769800
BEAUTY                   57987   158.8      185749     13416200
BOOKS_AND_REFERENCE     811037   622.1     4054019    139784155
BUSINESS               1378345  1009.2     5385913    435932920
COMICS                  113959   199.4      586416     16536100
COMMUNICATION          1046590   857.1   117006519   4939915530
DATING                  145862   684.7     3900266    141865110
EDUCATION                84532   478.8     7076206    278202000
ENTERTAINMENT            80221   356.5    13088246    650860000
EVENTS                   91719   156.6       77869      4648300
FAMILY                10684191  6728.7   290020637   6776172480
FINANCE                1474843  1082.0    10051738    306886300
FOOD_AND_DRINK          289744   344.2     4287186    177567750
```

GAME	5246629	4076.5	1342975035	29730752667
HEALTH_AND_FITNESS	825291	927.4	9150424	847406220
HOUSE_AND_HOME	132471	233.1	1644159	74982000
LIBRARIES_AND_DEMO	250320	256.5	995739	59983000
LIFESTYLE	1554800	1137.9	8287747	422739120
MAPS_AND_NAVIGATION	606908	376.4	3665840	175014560
MEDICAL	1240240	1341.9	1315643	44483076
NEWS_AND_MAGAZINES	1084497	695.8	9795612	4241900550
PARENTING	182308	191.3	879978	23566010
PERSONALIZATION	1589160	1188.3	33764858	943031930
PHOTOGRAPHY	1223009	970.1	76439163	2546893130
PRODUCTIVITY	1501441	956.5	37989367	1232302080
SHOPPING	810680	752.3	46945840	1503731540
SOCIAL	1033959	723.7	26080450	674240475
SPORTS	1567330	1038.6	52571750	1138911465
TOOLS	4328946	2526.5	105108107	3518553500
TRAVEL_AND_LOCAL	887074	624.6	6974566	316638300
VIDEO_PLAYERS	696489	455.3	22980256	781662200
WEATHER	287556	207.9	3517382	119296500

Category	Price in Dollars
ART_AND_DESIGN	5.97
AUTO_AND_VEHICLES	0.00
BEAUTY	0.00
BOOKS_AND_REFERENCE	20.89
BUSINESS	61.40
COMICS	0.00
COMMUNICATION	41.73
DATING	14.98
EDUCATION	17.96
ENTERTAINMENT	2.99
EVENTS	0.00
FAMILY	2217.41
FINANCE	2439.87
FOOD_AND_DRINK	4.99
GAME	269.39
HEALTH_AND_FITNESS	34.42
HOUSE_AND_HOME	0.00
LIBRARIES_AND_DEMO	0.00
LIFESTYLE	1953.40
MAPS_AND_NAVIGATION	14.96
MEDICAL	997.24
NEWS_AND_MAGAZINES	3.98
PARENTING	4.99
PERSONALIZATION	118.80
PHOTOGRAPHY	78.35
PRODUCTIVITY	52.96
SHOPPING	5.48
SOCIAL	1.98

```
SPORTS                          80.23
TOOLS                          183.60
TRAVEL_AND_LOCAL                26.51
VIDEO_PLAYERS                    0.99
WEATHER                         23.44
```

```
# getting the mean of the data according to category
cleanData.groupby('Category').mean()
```

Category	Unnamed: 0	Rating	Reviews	Installs
ART_AND_DESIGN	1353.649123	4.378947	1.529081e+04	8.636509e+05
AUTO_AND_VEHICLES	2717.516129	4.146774	1.161881e+04	5.446742e+05
BEAUTY	1567.216216	4.291892	5.020243e+03	3.626000e+05
BOOKS_AND_REFERENCE	5632.201389	4.320139	2.815291e+04	9.707233e+05
BUSINESS	5625.897959	4.119184	2.198332e+04	1.779318e+06
COMICS	2374.145833	4.154167	1.221700e+04	3.445021e+05
COMMUNICATION	5007.607656	4.100957	5.598398e+05	2.363596e+07
DATING	843.132948	3.957803	2.254489e+04	8.200295e+05
EDUCATION	775.522936	4.392661	6.491932e+04	2.552312e+06
ENTERTAINMENT	932.802326	4.145349	1.521889e+05	7.568140e+06
EVENTS	2620.542857	4.474286	2.224829e+03	1.328086e+05
FAMILY	6652.671856	4.189726	1.805857e+05	4.219285e+06
FINANCE	5607.768061	4.114068	3.821954e+04	1.166868e+06
FOOD_AND_DRINK	3449.333333	4.097619	5.103793e+04	2.113902e+06
GAME	5493.852356	4.268586	1.406257e+06	3.113168e+07
HEALTH_AND_FITNESS	3751.322727	4.215455	4.159284e+04	3.851846e+06
HOUSE_AND_HOME	2365.553571	4.162500	2.935998e+04	1.338964e+06
LIBRARIES_AND_DEMO	4103.606557	4.204918	1.632359e+04	9.833279e+05

LIFESTYLE	5592.805755	4.093165	2.981204e+04	1.520644e+06
MAPS_AND_NAVIGATION	6456.468085	4.004255	3.899830e+04	1.861857e+06
MEDICAL	3863.676012	4.180374	4.098576e+03	1.385766e+05
NEWS_AND_MAGAZINES	6455.339286	4.141667	5.830721e+04	2.524941e+07
PARENTING	4143.363636	4.347727	1.999950e+04	5.355911e+05
PERSONALIZATION	5778.763636	4.321091	1.227813e+05	3.429207e+06
PHOTOGRAPHY	5226.534188	4.145726	3.266631e+05	1.088416e+07
PRODUCTIVITY	6499.744589	4.140693	1.644561e+05	5.334641e+06
SHOPPING	4554.382022	4.226404	2.637407e+05	8.447930e+06
SOCIAL	6082.111765	4.257059	1.534144e+05	3.966120e+06
SPORTS	6345.465587	4.204858	2.128411e+05	4.610978e+06
TOOLS	6871.342857	4.010317	1.668383e+05	5.585006e+06
TRAVEL_AND_LOCAL	5723.058065	4.029677	4.499720e+04	2.042828e+06
VIDEO_PLAYERS	6163.619469	4.029204	2.033651e+05	6.917365e+06
WEATHER	5868.489796	4.242857	7.178331e+04	2.434622e+06

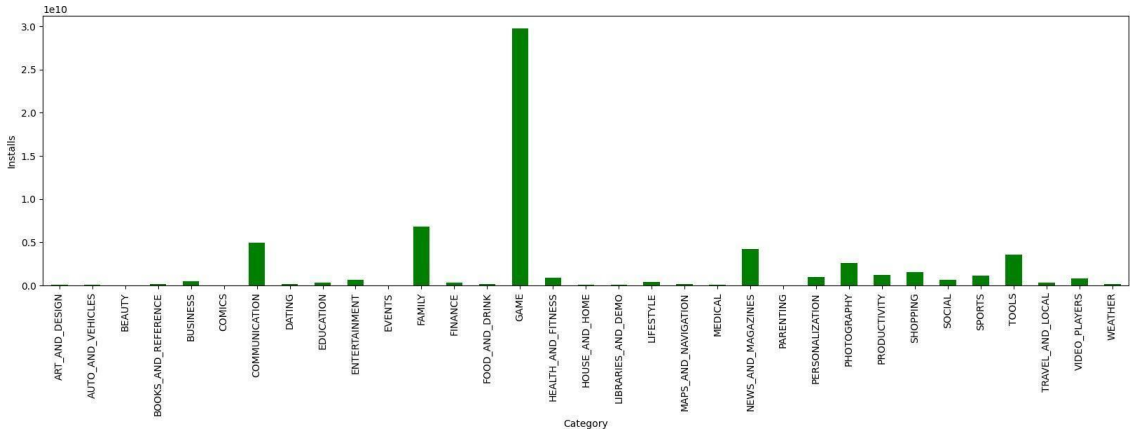
Category	Price in Dollars
ART_AND_DESIGN	0.104737
AUTO_AND_VEHICLES	0.000000
BEAUTY	0.000000
BOOKS_AND_REFERENCE	0.145069
BUSINESS	0.250612
COMICS	0.000000
COMMUNICATION	0.199665
DATING	0.086590
EDUCATION	0.164771
ENTERTAINMENT	0.034767
EVENTS	0.000000

FAMILY	1.380704
FINANCE	9.277072
FOOD_AND_DRINK	0.059405
GAME	0.282084
HEALTH_AND_FITNESS	0.156455

```
HOUSE_AND_HOME                0.000000
LIBRARIES_AND_DEMO            0.000000
LIFESTYLE                     7.026619
MAPS_AND_NAVIGATION           0.159149
MEDICAL                       3.106667
NEWS_AND_MAGAZINES            0.023690
PARENTING                     0.113409
PERSONALIZATION               0.432000
PHOTOGRAPHY                   0.334829
PRODUCTIVITY                  0.229264
SHOPPING                      0.030787
SOCIAL                        0.011647
SPORTS                        0.324818
TOOLS                         0.291429
TRAVEL_AND_LOCAL              0.171032
VIDEO_PLAYERS                 0.008761
WEATHER                       0.478367
```

```
# plotting the bar graph of category wise total installs
cat_ins_sum = cleanData.groupby(['Category'])['Installs'].sum()
cat_ins_sum.plot(kind='bar',ylabel="Installs",color="green",figsize(
2 0,5))
```

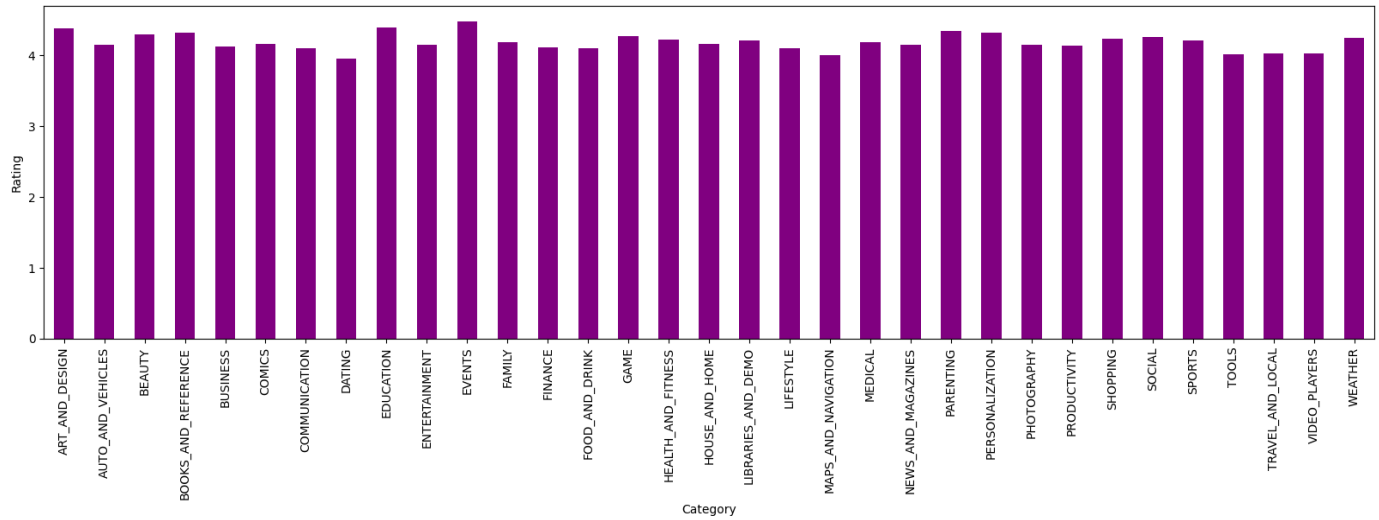
```
<AxesSubplot:xlabel='Category', ylabel='Installs'>
```




```
0          4.1     159   19M     10000   Free                0.0
Everyone
1          3.9     967   14M    500000   Free                0.0
Everyone
2          4.7   87510  8.7M   5000000   Free                0.0
Everyone
4          4.3     967  2.8M    100000   Free                0.0
Everyone
5          4.4     167  5.6M     50000   Free                0.0
Everyone
...        ...     ...   ...       ...   ...                 ...
...
10832      3.8    1195  582k    100000   Free                0.0
Everyone
10833      4.8      44  619k      1000   Free                0.0
Everyone
```

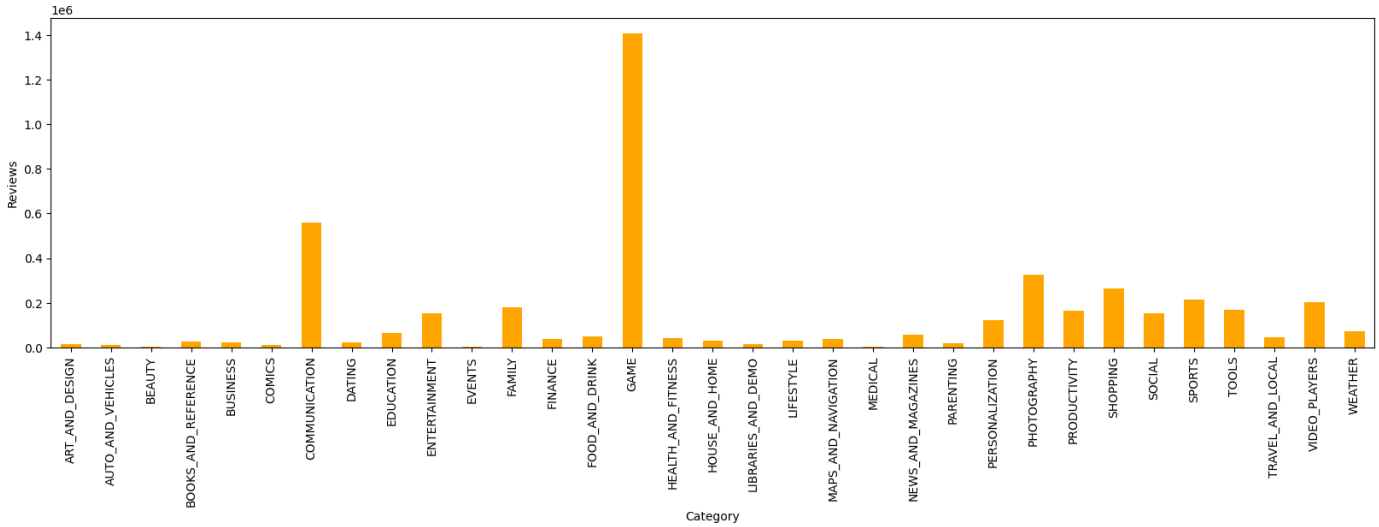
```
# plotting the bar graph of category wise mean rating
cat_ins_sum = cleanData.groupby(['Category'])['Rating'].mean()
cat_ins_sum.plot(kind='bar',ylabel="Rating",color="purple",figsize=(2
0
,5))
```

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



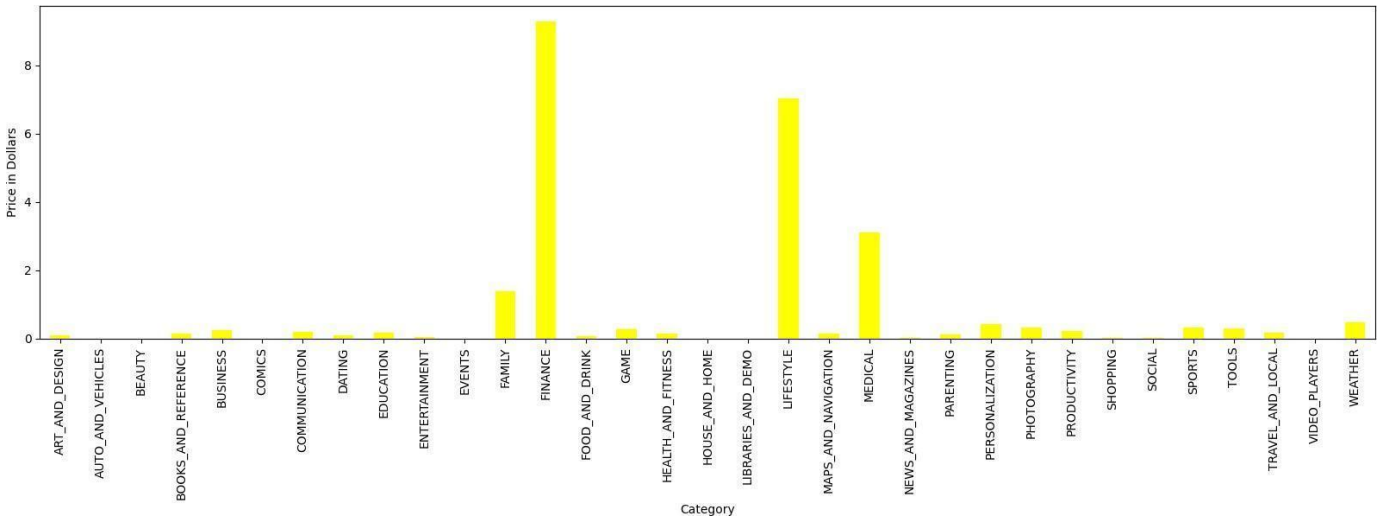

```
# plotting the bar graph of category wise mean reviews
cat_ins_sum = cleanData.groupby(['Category'])['Reviews'].mean()
cat_ins_sum.plot(kind='bar',ylabel="Reviews",color="orange",figsize=(
2 0,5))
```

```
<AxesSubplot:xlabel='Category', ylabel='Reviews'>
```



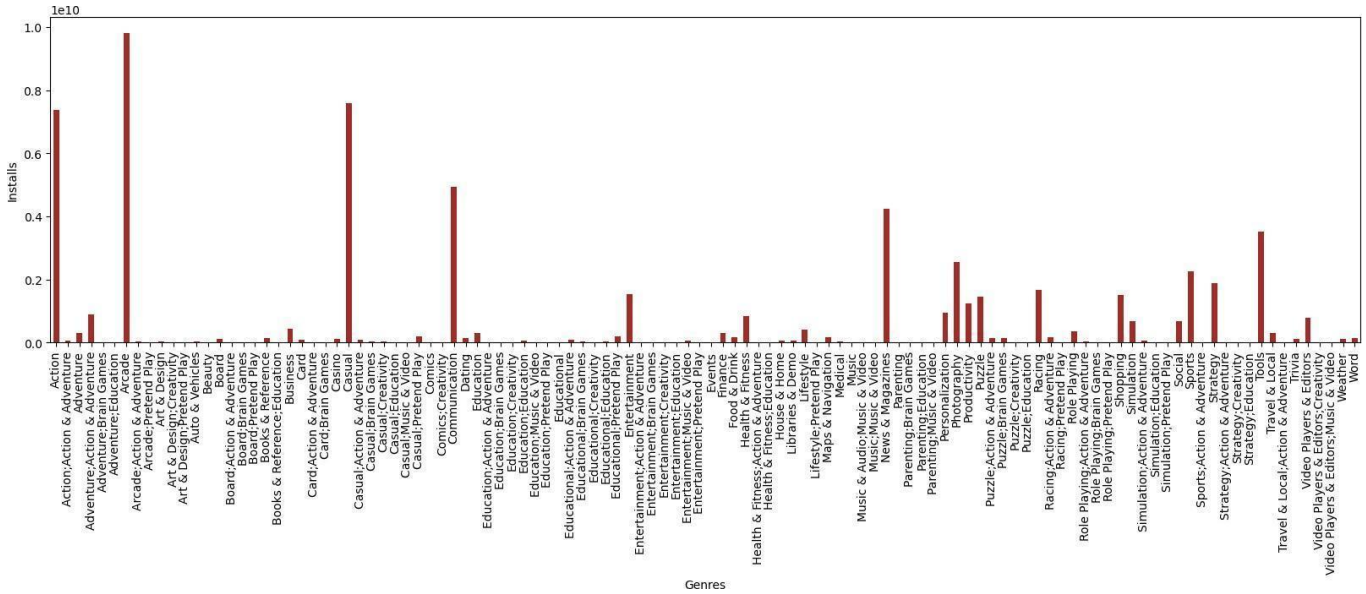

```
# plotting the bar graph of category wise mean price
cat_ins_sum = cleanData.groupby(['Category'])['Price
in Dollars'].mean()
cat_ins_sum.plot(kind='bar',ylabel="Price in
Dollars",color="yellow",figsize=(20,5))
```

```
<AxesSubplot:xlabel='Category', ylabel='Price in Dollars'>
```




```
# plotting the bar graph of genres wise total installs
cat_ins_sum = cleanData.groupby(['Genres'])['Installs'].sum()
cat_ins_sum.plot(kind='bar',ylabel="Installs",color="brown",figsize=(
2 0,5))
```

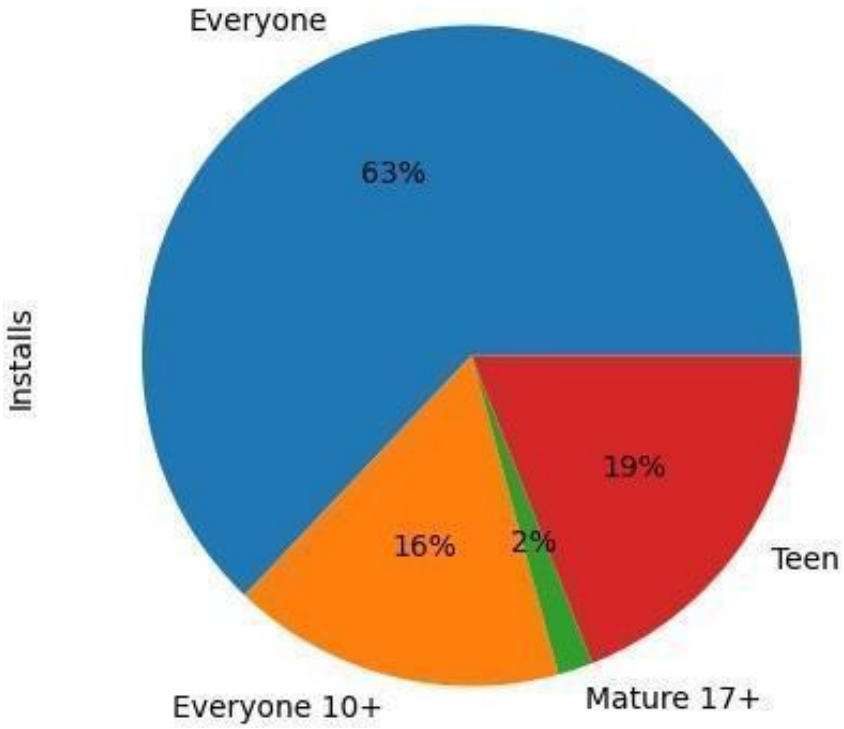
```
<AxesSubplot:xlabel='Genres', ylabel='Installs'>
```




```
# plotting the pie chart of content rating wise total installs
cat_ins_sum = cleanData.groupby(['Content
Rating'])['Installs'].sum() cat_ins_sum.drop(labels=['Unrated',
'Adults only 18+'], inplace=True) #
cat_ins_sum.drop(labels=-['Unrated'], inplace=True);

cat_ins_sum.plot(kind='pie',y="Installs",
figsize=(20,5),autopct='%1.0f%%')
```

```
<AxesSubplot:ylabel='Installs'>
```


Everyone
63%
Installs
19%
16%
2%
Teen
Everyone 10+
Mature 17+

Questions

Click here to view the python file in github

```
import pandas as
pd; cleanAppData =
pd.read_csv("E:/Folder/5th_Semester/Data_Analysis/Practicals/Github
Saurav Repo\data/cleanedAppsData.csv");
cleanAppData
```

```
      Unnamed: 0                                                 Ap   \
0              0     p Photo Editor & Candy Camera & Grid &
                     ScrapBook
1              1                                Coloring book moana
2              2  U    Launcher Lite – FREE Live Cool Themes, Hide
                                                                ...
3              4             Pixel Draw - Number Art Coloring Book
4              5                        Paper flowers instructions
...          ...                                                ...
7632       10832                                          FR Tides
7633       10833                                       Chemin (fr)
7634       10834                                     FR Calculator
7635       10836                                   Sya9a Maroc - FR
7636       10837                 Fr. Mike Schmitz Audio Teachings
```

```
1.0.0
```

```
7634               0.0       Everyone                    Education
1.0.
0
7635               0.0       Everyone                    Education
1.48
7636               0.0       Everyone                    Education
1.0

        Min Sup Android
0                     Ver
                    4.0.3
1                   4.0.3
2                   4.0.3
3                     4.4
4                     2.3
...                   ...
7632                  2.1
7633                  2.2
7634                  4.1
7635                  4.1
7636                  4.1

[7637 rows x 13 columns]
```

#1 Which Category has the maximum number of Apps ?

```
print(cleanAppData.groupby('Category').count()['App'].idxmax()," :
",cleanAppData.groupby('Category').count()['App'].max())
```

```
FAMILY :  1606
```

#2 List all the apps which got maximum ratings

```
cleanAppData[cleanAppData['Rating']==cleanAppData['Rating'].max()]
```

```
      Unnamed: 0                                              App
Category \
244          329                Hojiboy Tojiboyev Life
Hacks COMICS
434          612                American Girls Mobile
Numbers DATING
436          615                                   Awake Dating
DATING
442          633                          Spine- The dating
app DATING
444          636 Girls Live Talk - Free Text and Video Chat
DATING
...          ...                                             ...        ..
.
7557       10721                             Mad Dash Fo'
Cash GAME
7574       10742                          GKPB FP Online
Church LIFESTYLE
```

```
7593       10776                                  Monster Ride
GAME                                                       Pro
7627       10820                               Fr. Daoud Lamei
FAMILY
7636       10837            Fr. Mike         Audio Teachings
                            Schmitz
FAMILY

      Rating Reviews Size Installs Type Price in Dollars Content
Rating \
244      5.0       15   37M      1000  Free                0.0
Everyone
434      5.0        5  4.4M      1000  Free                0.0
Mature 17+
436      5.0        2   70M       100  Free                0.0
Mature 17+
442      5.0        5  9.3M       500  Free                0.0
Teen
444      5.0        6  5.0M       100  Free                0.0
Mature 17+
...      ...      ...   ...       ...   ...                ...
...
7557     5.0       14   16M       100  Free                0.0
Everyon
e 7574   5.0       32  7.9M      1000  Free                0.0
Everyone
7593     5.0        1   24M        10  Free                0.0
Everyone
7627     5.0       22  8.6M      1000  Free                0.0
Teen
7636     5.0        4  3.6M       100  Free                0.0
Everyone

         Genres Current Ver Min Sup Android Ver
244      Comics         2.0               4.0.3
434      Dating         3.0               4.0.3
436      Dating       2.2.9                 4.4
442      Dating         4.0               4.0.3
444      Dating         8.2               4.0.3
...         ...         ...                 ...
7557     Arcade        2.5a                 4.1
7574  Lifestyle       0.7.1                 4.4
7593     Racing         2.0                 2.3
7627  Education       3.8.0                 4.1
7636  Education         1.0                 4.1

[268 rows x 13 columns]
```

```
#3 List all the apps which got minimum ratings
cleanAppData[cleanAppData['Rating']==cleanAppData['Rating'].min()]
```

	Unnamed: 0	App
441	625	House party - live chat
2814	4127	Speech Therapy: F
3607	5151	Clarksburg AH
4212	5978	Truck Driving Test Class 3 BC
4457	6319	BJ Bridge Standard American 2018
4569	6490	MbH BM
5012	7144	CB Mobile Biz
5140	7383	Thistletown CI
5168	7427	CJ DVD Rentals
5460	7806	CR Magazine
5559	7926	Tech CU Card Manager
6226	8820	DS Creator 2.0
6264	8875	DT future1 cam
7311	10324	FE Mechanical Engineering Prep
7354	10400	Familial Hypercholesterolaemia
Handbook 7475	10591	Lottery Ticket Checker - Florida Results & Lotto

	Category	Rating	Reviews	Size	Installs	Type	Price in Dollars
441	DATING	1.0	1	9.2M	10	Free	0.00
2814	FAMILY	1.0	1	16M	10	Paid	2.99
3607	MEDICAL	1.0	1	28M	50	Free	0.00
4212	FAMILY	1.0	1	2.0M	50	Paid	1.49
4457	GAME	1.0	1	4.9M	1000	Free	0.00
4569	MEDICAL	1.0	1	2.3M	100	Free	0.00
5012	FINANCE	1.0	3	8.4M	500	Free	0.00
5140	PRODUCTIVITY	1.0	1	6.6M	100	Free	0.00
5168	COMMUNICATION	1.0	5	13M	100	Free	0.00
5460	BUSINESS	1.0	1	7.8M	100	Free	0.00
5559	FINANCE	1.0	2	7.2M	1000	Free	0.00
6226	TOOLS	1.0	2	4.4M	500	Free	0.00
6264	TOOLS	1.0	1	24M	50	Free	0.00
7311	FAMILY	1.0	2	21M	1000	Free	0.00
7354	MEDICAL	1.0	2	33M	100	Free	0.00

```
7475          TOOLS     1.0        3    41M       500  Free
0.00

     Content Rating         Genres   Current Ver Min Sup Android Ver
441      Mature 17+         Dating          3.52               4.0.3
2814       Everyone      Education           1.0               2.3.3
3607       Everyone        Medical   300000.0.81               4.0.3
4212       Everyone      Education           1.0                 2.1
4457       Everyone           Card      6.2-sayc                 4.0
4569       Everyone        Medical         1.1.3                 4.3
5012       Everyone        Finance      4.4.1255                 4.0
5140       Everyone   Productivity          41.9                 4.1
5168       Everyone  Communication           1.0                 4.1
5460       Everyone       Business         2.4.2               2.3.3
5559       Everyone        Finance         1.0.1                 4.0
6226       Everyone          Tools  2.0.180226.1                 4.0
6264       Everyone          Tools           3.1                 2.2
7311       Everyone      Education     5.33.3669                 5.0
7354       Everyone        Medical         2.0.1                 4.1
7475       Everyone          Tools           1.0                 4.2
```

```
#4 List all the apps which got above 1M reviews
cleanAppData[cleanAppData['Reviews']>1000000]
```

```
      Unnamed: 0                                              App
Category \
147          194       OfficeSuite : Free Office + PDF
Editor BUSINESS
214          293       OfficeSuite : Free Office + PDF
Editor BUSINESS
251          345                  Yahoo Mail – Stay
Organized COMMUNICATION
253          347                imo free video calls and
chat COMMUNICATION
262          366 UC Browser Mini -Tiny Fast Private &
Secure COMMUNICATION
...          ...                                           ...
...
6981        9860                   Voice changer with effects
FAMILY
7194       10186                              Farm Heroes Saga
FAMILY
7197       10190                               Fallout Shelter
FAMILY
7314       10327                              Garena Free Fire
GAME
7510       10636                           FRONTLINE COMMANDO
GAME

      Rating Reviews Size     Installs  Type Price in Dollars Content
```

```
Rating  \
147      4.3   100286    35M  100000000  Free                0.0
                   1
Everyon
e 214    4.3   100285    35M  100000000  Free                0.0
                   9
Everyon
e 251    4.3   418799    16M  100000000  Free                0.0
                   8
Everyon
e 253    4.3   478589    11M  500000000  Free                0.0
                   2
Everyon
e 262    4.4   364812   3.3M  100000000  Free                0.0
                   0
Teen     ..      ..      ...        ..                       ..
...      .     .        8.7     .          ...               .
...      4.    126090   M       5000000                      0.
6981     2     3                0          Free              0
Everyon
e 7194   4.4   761564    71M  100000000  Free                0.0
                   6
Everyon
e 7197   4.6   272192    25M   10000000  Free                0.0
                   3
Teen
7314     4.5   553411    53M  100000000  Free                0.0
                   4
Teen
7510     4.4   135183    12M   10000000  Free                0.0
                   3
Teen

             Genres       Current Ver Min Sup Android Ver


[328 rows x 13 columns]
```

```
#5 Show percentage of apps which got below 100 reviews
percent = (len(cleanAppData[cleanAppData['Reviews']<100])/
len(cleanAppData) )*100
percent = round(percent,
2) percent
```

```
24.59
```

#6 List the top 10 Apps whose size is maximum and minimum respectively along with their corresponding installs.

```
print(cleanAppData.sort_values(by="Size",ascending=False)
[['App','Size','Installs']][:10]);
print(cleanAppData.sort_values(by="Size")[['App','Size','Installs']]
[:10])
```

```
                                                  App Size
Installs
3354                                      Gangster Town 99M
5000000
1342 Miraculous Ladybug & Cat Noir - The Official Game 99M
10000000
6478 League of Stickman 2018- Ninja Arena PVP(Dream... 99M
1000000
6608                  L.A. Crime      Stories Mad City  99M
                                                Crime
1000000
1341                                   Earn to Die 2  99M
50000000
5916                                     Idle Heroes  99M
10000000
3393                                Angry Birds Blast  99M
10000000
1317                                My Talking Angela  99M
100000000
1248                                     Hero Hunters  99M
5000000
3791                    Arena    Valor: 5v5 Arena Game  99M
                        of
10000000
                                                  App  Size  Installs
4545                             BL PowerPoint Remote  1.0M       500
7090                             Remote EX for NISSAN  1.0M      5000
5680                                            go41cx  1.0M      1000
7048                        German Vocabulary Trainer  1.0M    100000
4750                      VOLT DROP CALCULATOR BS 7671  1.1M     10000
7204                             Mini for fb lite  1.1M    100000
3743                                    I AM RICH  1.1M     10000
5860   DG - Digital Coupons - Free Coupon and Discount 1.1M     10000
4107                             camera zoom moon  1.1M    500000
3486  Vpn Hosts (ad blocker & no root & support ipv6 ) 1.1M     10000
```

```
#7 What Percentage of Apps comes under free and paid installation?
freeApps = cleanAppData.groupby('Type').count()[['App']][0:1]
['App'].values[0]
PaidApps = cleanAppData.groupby('Type').count()[['App']][1:2]
['App'].values[0]
totalApps = freeApps + PaidApps;
print("Free Apps percentage : " , (freeApps/totalApps)*100, "%");
print("Paid Apps percentage : " , (PaidApps/totalApps)*100, "%");
```

```
Free Apps percentage : 92.52324211077648 %
Paid Apps percentage : 7.476757889223517 %
```

```
#8 List the top 10 expensive apps along with their corresponding
installs.
print(cleanAppData.sort_values(by="Price in Dollars",ascending=False)
[['App','Price in Dollars','Installs']][:10]);
```

```
                                Ap   Price in Dollar   Installs
3002        I'm p Rich - Trump                  s         10000
                Edition                     400.0
                                            0
3747                I Am Rich Premium       399.99      50000
3755  I am rich (Most expensive app)        399.99       1000
3753                    I Am Rich Pro       399.99       5000
3750               I am rich(premium)       399.99       5000
3749                       I am Rich!       399.99       1000
3760                        I am Rich       399.99       5000
2869          most expensive app (H)        399.99        100
3764               I AM RICH PRO PLUS       399.99       1000
3745                   I am Rich Plus       399.99      10000
```

```
#9 List Percentage of Apps that come under the different sections of
content rating?
cleanAppData.groupby('Content Rating').count()["App"].transform(lambda
x: (x/x.sum())*100)
```

```
Content Rating
Adults only 18+     0.026188
Everyone           80.005238
Everyone 10+        4.085374
Mature 17+          4.726987
Teen               11.143119
Unrated             0.013094
Name: App, dtype: float64
```

```
#10 Which Genre has the most and least popular among the people?
[compare the number of installs in each genre].
print(cleanAppData.groupby('Genres').sum()['Installs'].idxmax()," :
",cleanAppData.groupby('Genres').sum()['Installs'].max())
print(cleanAppData.groupby('Genres').sum()['Installs'].idxmin()," :
",cleanAppData.groupby('Genres').sum()['Installs'].min())
```

```
Arcade : 9820077677
Board;Pretend Play :  100
```

```
#11 Name the Category whose average rating is best and worst
respectively.
print(cleanAppData.groupby('Category').mean()['Rating'].idxmax()," :
",cleanAppData.groupby('Category').mean()['Rating'].max())
print(cleanAppData.groupby('Category').mean()['Rating'].idxmin()," :
",cleanAppData.groupby('Category').mean()['Rating'].min())
```

```
EVENTS : 4.474285714285714
DATING : 3.957803468208093
```

```
#12 Name the Category whose average no. of reviews is highest and
lowest ?
print(cleanAppData.groupby('Category').mean()['Reviews'].idxmax()," :
",cleanAppData.groupby('Category').mean()['Reviews'].max())
print(cleanAppData.groupby('Category').mean()['Reviews'].idxmin()," :
",cleanAppData.groupby('Category').mean()['Reviews'].min())
```

```
GAME : 1406256.5811518324
EVENTS : 2224.8285714285716
```

```
#13 What percentage of Apps whose Rating is below 4 ?
len(cleanAppData[cleanAppData['Rating']<4])/len(cleanAppData['Rating'
]
)*100
```

```
23.43852298022783
```

```
7
```

```
#14 What are the 10 top expensive apps that have a rate of 5 ?
cleanAppData[cleanAppData['Rating']==5].sort_values(by="Price in
Dollars",ascending=False)[['Category','App','Rating','Price in
Dollars','Installs']][:10]
```

```
                Category                                              App
Ratin   \
g
3861              FAMILY                  AP Art History
                                          Flashcards
5.0
5203              FAMILY                 USMLE Step 2 CK
                                         Flashcards
5.0
3669              FAMILY                      Hey AJ! It's
                                              Bedtime!
5.0
5041              FAMILY  TI-84 CE Graphing Calculator Manual TI
                                       84
5.0
3856              FAMILY                 meStudying: AP English
                                         Lit
5.0
3662  BOOKS_AND_REFERENCE                    Hey AJ! It's Saturday!
5.0
5821           LIFESTYLE                       AC DC Power Monitor
5.0
1622             MEDICAL   Supe  Hearing    Secret Voices Recorder
                           r                                   PRO
5.0
1630             MEDICAL                              FHR 5-Tier 2.0
5.0
5635              FAMILY                                Morse Player
5.0

      Price in Dollars  Installs
3861             29.99        10
5203             19.99        10
```

3669	4.99	10
5041	4.99	100
3856	4.99	10
3662	3.99	100

```
5821              3.04        10
1622              2.99       100
1630              2.99       500
5635              1.99       100
```

```
#15 What is the max and min size for free apps ?

print(cleanAppData[cleanAppData["Price in
Dollars"]==0].sort_values(by="Size",ascending=False)
[['App','Size','Price in Dollars']][0:1],"\n")
print(cleanAppData[cleanAppData["Price in
Dollars"]==0].sort_values(by="Size")[['App','Size','Price in
Dollars']][0:1],"\n")
```

```
                                                    App Size Price in
Dollars
6614 Miraculous Ladybug & Cat Noir - The Official Game 99M
0.0

                            App Size Price in Dollars
7048 German Vocabulary Trainer 1.0M
0.0
```

```
#16 What is the max and min size for paid apps ?
print(cleanAppData[cleanAppData["Price in
Dollars"]>0].sort_values(by="Size",ascending=False)
[['App','Size','Price in Dollars']][0:1],"\n")
print(cleanAppData[cleanAppData["Price in
Dollars"]>0].sort_values(by="Size")[['App','Size','Price in
Dollars']] [0:1],"\n")
```

```
                               App Size Price in Dollars
3969 Five Nights at Freddy's: SL 99M
2.99

                        App Size Price in Dollars
4545 BL PowerPoint Remote 1.0M
3.99
```

```
#17 Is there a correlation between rating, Reviews, and Size with the
price of the app?
corr_Ra = cleanAppData["Price in
Dollars"].corr(cleanAppData["Rating"])
corr_Re= cleanAppData["Price in
Dollars"].corr(cleanAppData["Reviews"])
# corr_Si= cleanAppData["Price in Dollars"].corr(cleanAppData["Size"])
print ("Correlation between Price in Dollars and Rating is: ",
corr_Ra)
print("Correlation between Price in Dollars and Rating is: ",
corr_Re) # print("Correlation between Price in Dollars and Rating is:
", corr_Si)
```

```
Correlation between Price in Dollars and Rating is: -
0.021268175833431043
Correlation between Price in Dollars and Rating is: -
0.010135457393102664
```

Machine learning Models

Click here to view the python file in github

```
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestClassifier,
VotingClassifier, GradientBoostingClassifier
from sklearn.tree import DecisionTreeClassifier,
DecisionTreeRegressor
from sklearn.model_selection import train_test_split,
cross_val_score, GridSearchCV
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score


#Import Cleaned Data
df = pd.read_csv('cleanedAppsData.csv')


### Feature Engineering ###

#Filtering for relevant variables - Remove App Name
print(df.columns)
df = df.drop(['App Name'], axis = 1)
print(df.columns)


#Reduce number of target classes
df.loc[df['Installs'].isin(['0 - 100', '100 - 500', '500 - 1,000']),
'Installs'] = '0 - 1,000'
df.loc[df['Installs'].isin(['1,000 - 5,000', '5,000 - 10,000']),
'Installs'] = '1,000 - 10,000'
df.loc[df['Installs'].isin(['10,000 - 50,000', '50,000 - 100,000']),
'Installs']  = '10,000 - 100,000'
df.loc[df['Installs'].isin(['100,000 - 500,000', '500,000 -
1,000,000']), 'Installs']  = '100,000 - 1,000,000'
df.loc[df['Installs'].isin(['1,000,000 - 5,000,000', '5,000,000 -
10,000,000']), 'Installs']  = '1,000,000 - 10,000,000'
df.loc[df['Installs'].isin(['10,000,000 - 50,000,000', '50,000,000 -
100,000,000']), 'Installs']  = '10,000,000 - 100,000,000'
df.loc[df['Installs'].isin(['100,000,000 - 500,000,000', '500,000,000
- 1,000,000,000']), 'Installs']  = '100,000,000 - 1,000,000,000'
df.loc[df['Installs'].isin(['1,000,000,000 - 5,000,000,000',
'5,000,000,000+']), 'Installs']  = '1,000,000,000+'
```

```
df.Installs = pd.Categorical(df.Installs, ['0 - 1,000','1,000 -
10,000', '10,000 - 100,000', '100,000 - 1,000,000', '1,000,000 -
10,000,000', '10,000,000 - 100,000,000', '100,000,000 -
1,000,000,000', '1,000,000,000+'])
print(df.Installs.value_counts().sort_index())
print(df.shape)


#One hot encoding due to Sklearn categorical variable limitation
(Sklearn Decision trees treat categorical variable as continuous)
strat = df.Category.values
df = pd.get_dummies(df, columns=['Content Rating', 'Category',
'Game_genre'], drop_first=True)
print(df.columns)

#Train Test Split - Simple Random Sampling (with Stratification)
X = df.drop(['Installs'], axis = 1).values
y = df.Installs.values
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.3, stratify = strat, random_state = 42)
X.shape[0] == y.shape[0]


#Decision Tree Classifier Train
dt = DecisionTreeClassifier(max_depth = 8, max_features = 'sqrt')
dt.fit(X_train, y_train)
y_pred = dt.predict(X_train)
acc = accuracy_score(y_train, y_pred)
print("Decision Tree train data accuracy: {:.2f}".format(acc))



#Decision Tree Classifier
dt.fit(X_train, y_train)
y_pred = dt.predict(X_test)

acc = accuracy_score(y_test, y_pred)
print("Decision Tree Test data accuracy: {:.2f}".format(acc))


# Decision Tree Cross Val
a = np.mean(cross_val_score(dt, X, y, scoring = 'accuracy', cv = 10))
print("Decision Tree cross validation accuracy: {:.2f}".format(a))

#The Decision Tree gave an accuracy of around 0.63


#K Nearest Neighbors
knn = KNeighborsClassifier()
knn.fit(X_train,y_train)
print('knn train data accuracy',knn.score(X_train, y_train))
print('knn test data accuracy', knn.score(X_test,y_test))
print('knn cross validation accuracy', np.mean(cross_val_score(knn,X,
y, cv = 5)))
```

```
#KNN gave an accuracy of 0.66
```

References

1. https://www.kaggle.com/datasets/lava18/google-play-store-apps
2. Python for Data analysis: data wrangling with pandas, numpy, and ipython