Dataset - Google Play Store

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 datafile googleplaystore_user_reviews contains datafields 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

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!

.

Questions on GooglePlayStore Data

1. Which Category has the maximum number of Apps ?
2. List all the apps which got maximum and minimum ratings and above 100 and below 10 reviews respectively.
3. List the top 10 Apps whose size is maximum and minimum respectively along with their corresponding installs.
4. What Percentage of Apps comes under free and paid installation?
5. List the top 10 expensive apps along with their corresponding installs.
6. List Percentage of Apps that come under the different sections of content rating?
7. Which Genre has the most and least popular among the people? [compare the number of installs in each genre].
8. Name the Category whose average rating is best and worst respectively.
9. Name the Category whose average no. of reviews is highest and lowest ?
10. What percentage of Apps can run on the Android Version of 4 and above ?
11. What percentage of Apps whose Rating is below 4 ?
12. What are the 5 top expensive apps that have a rate of 5 ?
13. What is the average price of the paid apps for each genre ?
14. What is the max and min size for free and paid apps ?
15. Is there a correlation between rating, Reviews, and Size with the price of the app?

Data Cleaning Steps

1) Analysis of the data using head, describe, shape functions on the dataset.

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

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

	App	Category	Rating	Reviews	Size	Installs	Type
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free
	Coloring						

```
# 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()
```

	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

```
# 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
```

2) Removed missing values from the dataset.
3) Removed duplicated and irrelevant observations
4) Renamed the columns
5) Removed inconsistencies in the dataset.

```
import pandas as pd
```

```
dataset = pd.read_csv("app.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', '4.6M', '9.2M', '5.2M', '11M',
       '24M', 'Varies with device', '9.4M', '15M', '10M', '1.2M', '26M',
       '8.0M', '7.9M', '56M', '57M', '35M', '54M', '201k', '3.6M', '5.7M',
       '8.6M', '2.4M', '27M', '2.7M', '2.5M', '7.0M', '16M', '3.4M',
       '8.9M', '3.9M', '2.9M', '38M', '32M', '5.4M', '18M', '1.1M',
       '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', '51M',
       '41M', '48M', '8.5M', '46M', '8.3M', '4.3M', '4.7M', '3.3M', '40M',
       '7.8M', '8.8M', '6.6M', '5.1M', '61M', '66M', '79k', '8.4M',
       '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', '77M',
       '4.4M', '70M', '9.3M', '8.1M', '36M', '6.9M', '7.4M', '84M', '97M',
       '2.0M', '1.9M', '1.8M', '5.3M', '47M', '556k', '526k', '76M',
       '7.6M', '59M', '9.7M', '78M', '72M', '43M', '7.7M', '6.3M', '334k',
       '93M', '65M', '79M', '100M', '58M', '50M', '68M', '64M', '34M',
       '67M', '60M', '94M', '9.9M', '232k', '99M', '624k', '95M', '8.5k',
       '41k', '292k', '80M', '1.7M', '10.0M', '74M', '62M', '69M', '75M',
       '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', '169k',
       '45k', '965k', '90M', '545k', '61k', '283k', '655k', '714k', '93k',
       '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',
       '551k', '29k', '103k', '116k', '153k', '209k', '499k', '173k',
       '597k', '809k', '122k', '411k', '400k', '801k', '787k', '50k',
       '643k', '986k', '516k', '837k', '780k', '20k', '498k', '600k',
       '656k', '221k', '228k', '176k', '34k', '259k', '164k', '458k',
       '629k', '28k', '288k', '775k', '785k', '636k', '916k', '994k',
       '309k', '485k', '914k', '903k', '608k', '500k', '54k', '562k',
       '847k', '948k', '811k', '270k', '48k', '523k', '784k', '280k',
       '24k', '892k', '154k', '18k', '33k', '860k', '364k', '387k',
       '626k', '161k', '879k', '39k', '170k', '141k', '160k', '144k',
```

```
       '143k', '190k', '376k', '193k', '473k', '246k', '73k', '253k',
       '957k', '420k', '72k', '404k', '470k', '226k', '240k', '89k',
       '234k', '257k', '861k', '467k', '676k', '552k', '582k', '619k'],
      dtype=object)
```

```
cleanDataset.shape
```

```
(9360, 13)
```

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

```
cleanDataset.shape
```

```
(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',
       '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', '2.2.6.2', '1.1.3', '1.5', '1.0.8',
       '1.03', '6', '6.7.12.2018', '1.2', '2.2', '1.1.0', '1.6', '2.1',
       '1.0.9', '1.3', '2.0.1', '1.46', '1.6.1', '11', '3', '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', '1.5.18', '2.3.4', '2.17', '6.10.1', '2.3.0', '1.0.6',
       '1.9', '3.0.1', '3.3.9', '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',
       '1.91180527', '9.1.363', '1.1.6', '2.3.18', '15', '18.05.31+530',
       '5.0.6', '3.12', '2', '1.28', '6.0.8', '14', '3.05', '2.5.3',
       '7.0.4.6', '1.15', '3.1.7.9', '3.9.1', '3.4.2', '9.7.14188',
       '4.9.10'], dtype=object)
```

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

```
# 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['Android Ver']=='Varies wit
```

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

```
cleanDataset.shape
```

```
(7637, 12)
```

```
from google.colab import drive

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

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

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