NYP Data Science Foundation 2022S1 ITI102 Assignment 1

Smartphone Applications Dataset Analysis

Instructions:

- Complete sections with label TODO with the correct code
- Add the code under the label #Add code
- Fill your Name and Admin in the below section

```
Name : Ng Jarebb
Admin no: 22A208Z
```

In this assignment, you are going to analyse the dataset for Smartphone Applications

Dataset Information

You are given a csv file **dataset.csv** with all the data.
There are 13 columns in the csv file.
The first row is the header.

Following are the description for each header for the columns:

1.App : Name of the Application

2.Category : the catergory where the apllication belong example Art and design, beauty etc..

3.Rating : The rating of the app between 0 to 5 with 1 decimal point

4.Reviews :Number of customers reviewed app

5.Size :The size of the software(app)

6.Installs : Number of customers installed the app

7.Type: refer to the purchase mode of the app-either free or paid

8.Price : the price of the app

9.Content Rating : refer the content suitability for different audience eg. everyone,erveryone 17+,teen, Mature 17+ ,..etc

10.Genres : denoting the type of app. eg. Art & Design, Auto & Vehicles,..etc

11.Last Updated : Date where the app last updated

12.Current Ver : App current version

13.Android Ver : Android version that support the app

Complete all the codes in the following sections:

> 1. Importing Packages
> 2. Reading Data
> 3. Data Preprocessing
> - 3.1 Handling NULL Values
> - 3.2 Handling Data Types and Values
> 4. Analyzing Features
> 5. Furthur Analysis.

Read the requirement at TODO
Then fill the code in the cell with tag #Add code

1. Importing the required packages

TODO: (0.5 mark)

- Add all the modules that you require for the subsequence steps.
- All the required modules must be added in this section.

```
#Add code

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

TODO: (0.5 mark)

- Mount drive in colab
- Create directory in google drive '/content/drive/My Drive/Data/DSA1/'
- Copy file 'dataset.csv' into directory '/content/drive/My Drive/Data/DSA1/'

```
#Add Code

from google.colab import drive
```

```
drive.mount('/content/drive', force_remount=True)
datadir="/content/drive/My Drive/Data/DSA1/"
```

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

2. Reading Data

TODO: (1 mark)

- Use pandas function to read in the given dataset 'dataset.csv' that you have copied in the directory '/content/drive/My Drive/Data/DSA1/'.
- Assign the read dataframe into a variable. Use this dataframe variable for the rest of the part in the notebook.
- Display the first 5 rows of data that you read using the pandas function.

```
#Add code

df = pd.read_csv(datadir + 'dataset.csv')
```

```
#Add code
df.head(5)
```

	App	Category	Rating	Reviews	Size	In
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	5
2	U Launcher Lite – FREE Live Cool Themes, Hide	ART_AND_DESIGN	4.7	87510	8.7M	5,0

3. Data Preprocessing

3.1 Handling NULL/Duplication Values

This is a very crucial step in every analysis and model, which help to improves the accuracy of insights and predictions.

TODO: (1 mark)

Use the dataframe variable that you read from the previous section using pandas function.

- Count number of Null in each columns. Print the result.
- Drop all rows with Null values. Print the total rows after remove the Null.

```
#Check the number of rows before the data cleaning
df.shape
```

```
(10841, 13)
```

```
#Add code

df.isnull().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
```

```
#Add code

df = df.dropna()
print(len(df))
```

```
9360
```

There also duplications in the dataset

TODO: (1 mark)

- Remove the duplications rows in the dataset

```
#Add code
df = df.drop_duplicates()
```

Print the shape of the data read

- Display the size of the dataset read

```
df.shape
```

```
(8886, 13)
```

3.2 Handling Data Types of each Feature

The data types of each feature must be changed to a proper format that can be used for analysis.

This is the tasks for this section.

TODO: (2 marks)

The Column *Reviews* must be of correct numerical type

- Check the Column Reviews type. Display the result.
- Change the Reviews column type to int64. Display the result.

```
#Add code

df['Reviews'].dtypes
```

```
dtype('O')
```

```
#Add code

df['Reviews'] = df['Reviews'].astype(np.int64)
df['Reviews']
```

```
0           159
1           967
2         87510
3        215644
4           967
          ...
10834         7
10836        38
10837         4
10839       114
10840    398307
Name: Reviews, Length: 8886, dtype: int64
```

TODO: (2 marks)

Changing the Feature : Installs

- The installs values must be changed to a proper format so that we can use them for analysis and plots.
  Example : Change '10,000+' to 10000
- After format the data in installs column, convert all items in the column to float.

- Update the new data into your dataframe where the installs column is.

```
#Add code
df['Installs'] = df['Installs'].map(lambda x: x.replace('+', '').replace(',', '')).
```

TODO: (2 marks)

Changing the feature : Size

- The column Size values are not able to use for data analysis, it not a value.
- Change it into number format example 15M to 15.0 and other format set to 0.0. Do for all the rest in this column.
- After format the data in Size column, convert all items in the column to float.
- Update the new data into your dataframe where the Size column is.

```
#Add code

df['Size'] = df['Size'].map(lambda x: x.replace('M', '000000').replace('k', '000').
```

TODO: (2 marks)

Changing the feature Price

- Most value in the Price column is 0 but some have value example

```
$2.44
```

- You need to remove '$' in the data.
- After format the data in Price column, convert all items in the column to float.
- Update the new data into your dataframe where the Price column is.

```
#Add code

df['Price'] = df['Price'].map(lambda x: x.replace('$', '')).astype(float)
```

TODO: (2 marks)

Changing the feature, Android Ver

The column Android Ver store data example 4.0.3 and up

- We are only interest on the major version not the minor version
- We need to change format, for example **4.0.3 and up** to **4.0**, do for the rest of items in this column.
- After format the data in Android Ver column, convert all items in the column to float.

- Update the new data into your dataframe where the Android Ver column is.

```
#Add code

df['Android Ver'] = df['Android Ver'].map(lambda x: x.replace('Varies with device',
```

4. Analyzing Features :

4.1 Categories

TODO: (2 marks)

Find the Total count for each of the Category

- In the Category column display the different App category. You need to count the total number for each Category.

Example

```
Example only. Not the answer.

FAMILY                1717
GAME                  1074
TOOLS                  733
PRODUCTIVITY           334
FINANCE                317
.....
.....
```

```
#Add code
df["Category"].value_counts()
```

```
FAMILY                 1717
GAME                   1074
TOOLS                   733
PRODUCTIVITY            334
FINANCE                 317
PERSONALIZATION         308
COMMUNICATION           307
LIFESTYLE               305
PHOTOGRAPHY             304
MEDICAL                 302
SPORTS                  286
BUSINESS                270
HEALTH_AND_FITNESS      262
SOCIAL                  244
```

```
NEWS_AND_MAGAZINES      214
TRAVEL_AND_LOCAL        205
SHOPPING                202
BOOKS_AND_REFERENCE     177
VIDEO_PLAYERS           160
DATING                  159
EDUCATION               129
MAPS_AND_NAVIGATION     124
ENTERTAINMENT           111
FOOD_AND_DRINK          106
WEATHER                  75
AUTO_AND_VEHICLES        73
HOUSE_AND_HOME           68
LIBRARIES_AND_DEMO       64
ART_AND_DESIGN           61
COMICS                   58
PARENTING                50
EVENTS                   45
BEAUTY                   42
Name: Category, dtype: int64
```

TODO: (3 marks)

- Group using the column 'Type', 'Category'and aggregate Rating with the mean
- Which 'Type' and 'Category' has the highest mean Rating? Write your answer in the text cell below.

```
Example only. Not the answer.

                            Rating
                           mean
Type    Category
Free    ART_AND_DESIGN       4.358621
       AUTO_AND_VEHICLES     4.184722
       BEAUTY                4.278571
       BOOKS_AND_REFERENCE   4.350888
       BUSINESS              4.103448
...    ...    ...
Paid     SPORTS              4.254545
        TOOLS                4.169841
        TRAVEL_AND_LOCAL     4.100000
        VIDEO_PLAYERS        4.100000
        WEATHER              4.371429
```

```
#Add code
df.groupby(['Type', 'Category']).agg(Mean=('Rating', np.mean)).sort_values(['Mean']
```

		Mean
Type	**Category**	
Paid	**NEWS_AND_MAGAZINES**	4.800000
	EDUCATION	4.750000
	ART_AND_DESIGN	4.733333
	ENTERTAINMENT	4.600000
	AUTO_AND_VEHICLES	4.600000
	...	...
	MAPS_AND_NAVIGATION	3.860000
	FINANCE	3.830769
	SOCIAL	3.700000
	DATING	3.625000
	PARENTING	3.350000

Which 'Type' and 'Category' has the highest mean Rating?

- **Paid** & **NEWS_AND_MAGAZINES**

4.2 Price

TODO: (1 mark)

- Find the row where the column Price is max and store as a dataframe

```
#Add code
df_max_price = df[df['Price'] == df['Price'].max()]
```

5. Furthur Analysis

TODO: (2 marks)

- Find all Apps with 5.0 ratings:
- Then use the result to Group using the column 'Type', 'Category'and aggregate Rating with the number of count
- Which of the Type and Group have the maximum Rating count?
- Place your answers in the Text Cell.

```
Example only. Not the answer.
```

```
                          Rating count
 Type    Category
 Free    ART_AND_DESIGN    1
       BOOKS_AND_REFERENCE    4
       BUSINESS    18
       COMICS    2
       COMMUNICATION    5
       DATING    6
       EVENTS    6
       FAMILY    59
       FINANCE    8
       FOOD_AND_DRINK    2
       GAME    8
       HEALTH_AND_FITNESS    12
       LIBRARIES_AND_DEMO    2
       LIFESTYLE    27
       MEDICAL    23
       NEWS_AND_MAGAZINES    7
       PARENTING    1
       PERSONALIZATION    3
       PHOTOGRAPHY    6
       PRODUCTIVITY    7
       SHOPPING    6
       SOCIAL    8
       SPORTS    4
       TOOLS    15
       TRAVEL_AND_LOCAL    3
 Paid    BOOKS_AND_REFERENCE    2
       FAMILY    8
       GAME    4
       LIFESTYLE    2
       MEDICAL    2
       PERSONALIZATION    7
       PRODUCTIVITY    1
       TOOLS    2
```

```
#Add code
df_app_five_rating = df[df['Rating'] == 5]
```

```
#Add code
df_top_count = df_app_five_rating.groupby(['Type', 'Category']).agg(Count=('Rating'
```

```
#Add code
df_top_count
```

		Count
Type	**Category**	
Free	**FAMILY**	59

Which of the Type and Group have the maximum Rating count?

- **Free** & **FAMILY**

TODO: (2 marks)

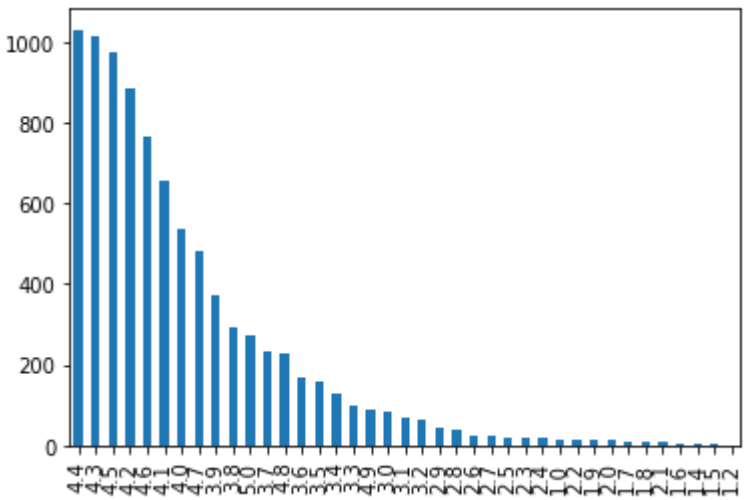
- Show the descriptive statistics for the column Rating
- Plot a frequencey plot for distribution of the column Rating
- Get the skew value of the column Rating

```
#Add code
df['Rating'].describe()
```

```
count    8886.000000
mean        4.187959
std         0.522428
min         1.000000
25%         4.000000
50%         4.300000
75%         4.500000
max         5.000000
Name: Rating, dtype: float64
```

```
#Add code
fig, ax = plt.subplots()
df['Rating'].value_counts().plot(ax=ax, kind='bar')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f16c9fe1d50>
```




```
#Add code
df['Rating'].skew()
```

```
-1.8239465880060681
```

TODO: (6 marks)

Category and Reviews:

- In each unique Category get average of review and rating
- Sort the rating in the descending order as show result

```
Example only. Not the answer.

                      Reviews    Rating
Category
EVENTS                3.568667e+03    4.435556
ART_AND_DESIGN        2.810356e+04    4.377049
EDUCATION              1.795715e+05    4.375969
BOOKS_AND_REFERENCE    1.235752e+05    4.347458
PERSONALIZATION        2.441312e+05    4.333117
PARENTING              1.907218e+04    4.300000
GAME                  1.318002e+06    4.281285
BEAUTY                9.407929e+03    4.278571
HEALTH_AND_FITNESS    1.177293e+05    4.261450
SOCIAL                2.186789e+06    4.254918
.......
.......
.......
```

```
#Add code
gb_rv_rating = df.groupby(['Category']).agg(Reviews=('Reviews', 'mean'), Rating=('R
```

```
#Add code
gb_rv_rating.sort_values(['Rating'], ascending=False)
```

Category	Reviews	Rating
EVENTS	3.568667e+03	4.435556
ART_AND_DESIGN	2.810356e+04	4.377049
EDUCATION	1.795715e+05	4.375969
BOOKS_AND_REFERENCE	1.235752e+05	4.347458
PERSONALIZATION	2.441312e+05	4.333117
PARENTING	1.907218e+04	4.300000
GAME	1.318002e+06	4.281285
BEAUTY	9.407929e+03	4.278571
HEALTH_AND_FITNESS	1.177293e+05	4.261450
SOCIAL	2.186789e+06	4.254918
SHOPPING	4.699553e+05	4.251485
WEATHER	1.947293e+05	4.244000
SPORTS	2.283990e+05	4.225175
PRODUCTIVITY	3.070486e+05	4.201796
FAMILY	2.310824e+05	4.191264
AUTO_AND_VEHICLES	1.594014e+04	4.190411
PHOTOGRAPHY	6.720308e+05	4.182895
MEDICAL	4.623930e+03	4.182450
LIBRARIES_AND_DEMO	1.583422e+04	4.179688
HOUSE_AND_HOME	4.109399e+04	4.164706
FOOD_AND_DRINK	7.237033e+04	4.164151
COMICS	5.830940e+04	4.155172
COMMUNICATION	1.958544e+06	4.151466
ENTERTAINMENT	4.285650e+05	4.136036
NEWS_AND_MAGAZINES	1.787145e+05	4.128505
FINANCE	5.362640e+04	4.127445
BUSINESS	4.576928e+04	4.102593
LIFESTYLE	4.203134e+04	4.096066
TRAVEL_AND_LOCAL	2.710488e+05	4.094146
VIDEO_PLAYERS	6.898731e+05	4.063750
MAPS_AND_NAVIGATION	2.472505e+05	4.051613
TOOLS	3.726878e+05	4.047203

TOOLS	3.726378e+03	4.047203
DATING	3.487525e+04	3.971698