# -\*- coding: utf-8 -\*-

"""playstore Analysis Using ML

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/10HNAvDGZv77-Jq\_\_wnPJoSzzDi5T\_fk5

"""

# Commented out IPython magic to ensure Python compatibility.

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt, seaborn as sns

# %matplotlib inline

pla=pd.read\_csv("/content/playstore-analysis (1).csv")

pla.head()

pla.info()

pla.isnull().sum()

"""# Task:1 Data clean up – Missing value treatment

a. Drop records where rating is missing since rating is our target/study variable

"""

pla.dropna(how='any', subset=['Rating'], axis=0, inplace = True)

pla.Rating.isnull().sum()

"""b. Check the null values for the Android Ver column.

i. Are all 3 records having the same problem?

"""

pla.loc[pla['Android Ver'].isnull()]

"""i.Drop the 3rd record i.e. record for “Life Made WIFI …”"""

pla.drop([10472], inplace = True)

pla.loc[pla['Android Ver'].isnull()]

"""iii. Replace remaining missing values with the mode¶"""

pla['Android Ver'].fillna(pla['Android Ver'].mode()[0], inplace=True)

"""c. Current ver – replace with most common value¶"""

pla['Current Ver'].fillna(pla['Current Ver'].mode()[0], inplace=True)

"""# Task: 2. Data clean up – correcting the data types

a. Which all variables need to be brought to numeric types?

Reviews and installs need to be brought to numeric types.

b.. Price variable – remove $ sign and convert to float

"""

price = []

for i in pla['Price']:

if i[0]=='$':

price.append(i[1:])

else:

price.append(i)

pla.drop(labels=pla[pla['Price']=='Everyone'].index, inplace = True)

pla['Price']= price

pla['Price']= pla['Price'].astype('float')

"""c. Installs – remove ‘,’ and ‘+’ sign, convert to integer"""

install = []

for j in pla['Installs']:

install.append(j.replace(',','').replace('+','').strip())

pla['Installs']= install

pla['Installs']= pla['Installs'].astype('int')

"""d. Convert all other identified columns to numeric"""

pla['Reviews']= pla['Reviews'].astype('int')

"""# Task 3. Sanity checks – check for the following and handle accordingly

#### a. Avg. rating should be between 1 and 5, as only these values are allowed on the play store.

i. Are there any such records? Drop if so.

"""

pla.loc[pla.Rating < 1] & pla.loc[pla.Rating > 5]

"""There are no such records with rating less than 1 or greater than 5.

##### b. Reviews should not be more than installs as only those who installed can review the app.

\*\*i. Are there any such records? Drop if so.\*\*

Yes, there are 7 records where Review is greater than Installs.

"""

pla.loc[pla['Reviews'] > pla['Installs']]

temp = pla[pla['Reviews']>pla['Installs']].index

pla.drop(labels=temp, inplace=True)

pla.loc[pla['Reviews'] > pla['Installs']]

"""# Task 4. Identify and handle outliers –

\*\*a. Price column\*\*

i. Make suitable plot to identify outliers in price

"""

plt.boxplot(pla['Price'])

plt.show()

"""ii.Do you expect apps on the play store to cost $200? Check out these cases"""

print('Yes we can expect apps on the play store to cost $200')

pla.loc[pla['Price'] > 200]

"""iv. Limit data to records with price < $30"""

gt\_30 = pla[pla['Price'] > 30].index

pla.drop(labels=gt\_30, inplace=True)

count = pla.loc[pla['Price'] > 30].index

count.value\_counts().sum()

"""iii. After dropping the useless records, make the suitable plot again to identify outliers"""

plt.boxplot(pla['Price'])

plt.show()

"""b. Reviews column

i. Make suitable plot

"""

sns.kdeplot(pla['Reviews'])

plt.show()

"""ii. Limit data to apps with < 1 Million reviews"""

gt\_1m = pla[pla['Reviews'] > 1000000 ].index

pla.drop(labels = gt\_1m, inplace=True)

print(gt\_1m.value\_counts().sum(),'cols dropped')

"""\*\*c. Installs\*\*

i. What is the 95th percentile of the installs?

"""

percentile = pla.Installs.quantile(0.95) #95th Percentile of Installs

print(percentile,"is 95th percentile of Installs")

"""ii.Drop records having a value more than the 95th percentile"""

for i in range(0,101,1):

print(' the {} percentile of installs is {} '.format(i,np.percentile(pla['Installs'],i)))

temp1 = pla[pla["Installs"] > percentile].index

pla.drop(labels = temp1, inplace = True)

print(temp1.value\_counts().sum())#,'cols dropped')

"""# Data analysis to answer business questions

### Task 5. What is the distribution of ratings like? (use Seaborn) More skewed towards higher/lower values?¶

a. How do you explain this?

"""

sns.histplot(pla['Rating'])

plt.show()

print('The skewness of this distribution is',pla['Rating'].skew())

print('The Median of this distribution {} is greater than mean {} of this distribution'.format(pla.Rating.median(),pla.Rating.mean()))

"""b. What is the implication of this on your analysis?\*¶\*"""

pla['Rating'].mode()

"""Since mode>= median > mean, the distribution of Rating is Negatively Skewed.

Therefore distribution of Rating is more Skewed towards lower values.

#### 6. What are the top Content Rating values?

a. Are there any values with very few records?

"""

pla['Content Rating'].value\_counts()

"""Adults only 18+ and Unrated are values with very few records so we drop them.

b. If yes, drop those as they won’t help in the analysis

"""

#Replacing unwanted values with NaN

cr = []

for k in pla['Content Rating']:

cr.append(k.replace('Adults only 18+','NaN').replace('Unrated','NaN'))

pla['Content Rating']=cr

# Droping the NaN values.

temp2 = pla[pla["Content Rating"] == 'NaN'].index

pla.drop(labels=temp2, inplace=True)

print('droped cols',temp2)

pla['Content Rating'].value\_counts() # Just Check

"""# Task 7. Effect of size on rating

a. Make a joinplot to understand the effect of size on rating¶

"""

sns.jointplot(y ='Size', x ='Rating', data = pla, kind ='hex')

plt.show()

"""\*\*b. Do you see any patterns?\*\*

Yes, patterns can be observed between Size and Rating ie. their is correlation between Size and Rating.

\*\*c. How do you explain the pattern?\*\*

Generally on increasing Rating, Size of App also increases. But this is not always true ie. for higher Rating, their is constant Size. Thus we can conclude that their is positive correlation between Size and Rating.

# Task 8. Effect of price on rating

a. Make a jointplot (with regression line)

"""

sns.jointplot(x='Price', y='Rating', data=pla, kind='reg')

plt.show()

"""\*\*b. What pattern do you see?\*\*

Generally on increasing the Price, Rating remains almost constant greater than 4.

\*\*c. How do you explain the pattern?\*\*

Since on increasing the Price, Rating remains almost constant greater than 4. Thus it can be concluded that their is very weak Positive correlation between Rating and Price.

"""

pla.corr()

"""d. Replot the data, this time with only records with price > 0"""

ps1=pla.loc[pla.Price>0]

sns.jointplot(x='Price', y='Rating', data=ps1, kind='reg')

plt.show()

"""\*\*e. Does the pattern change?\*\*

Yes, On limiting the record with Price > 0, the overall pattern changed a slight ie their is very weakly Negative Correlation between Price and Rating.

"""

ps1.corr()

"""\*\*f. What is your overall inference on the effect of price on the rating\*\*

Generally increasing the Prices, doesn't have signifcant effect on Higher Rating. For Higher Price, Rating is High and almost constant ie greater than 4

\*\*9. Look at all the numeric interactions together –\*\*

a. Make a pairplort with the colulmns - 'Reviews', 'Size', 'Rating', 'Price'

"""

sns.pairplot(pla, vars=['Reviews', 'Size', 'Rating', 'Price'], kind='reg')

plt.show()

"""# Task 10. Rating vs. content rating

a. Make a bar plot displaying the rating for each content rating

"""

pla.groupby(['Content Rating'])['Rating'].count().plot.bar(color="darkgreen")

plt.show()

"""\*\*b. Which metric would you use? Mean? Median? Some other quantile?\*\*

We must use Median in this case as we are having Outliers in Rating. Because in case of Outliers , median is the best measure of central tendency.

"""

plt.boxplot(pla['Rating'])

plt.show()

"""c. Choose the right metric and plot"""

pla.groupby(['Content Rating'])['Rating'].median().plot.barh(color="darkgreen")

plt.show()

"""\*\*Task 11. Content rating vs. size vs. rating – 3 variables at a time\*\*

a. Create 5 buckets (20% records in each) based on Size

"""

bins=[0, 20000, 40000, 60000, 80000, 100000]

pla['Bucket Size'] = pd.cut(pla['Size'], bins, labels=['0-20k','20k-40k','40k-60k','60k-80k','80k-100k'])

pd.pivot\_table(pla, values='Rating', index='Bucket Size', columns='Content Rating')

"""b. By Content Rating vs. Size buckets, get the rating (20th percentile) for each combination"""

temp3=pd.pivot\_table(pla, values='Rating', index='Bucket Size', columns='Content Rating', aggfunc=lambda x:np.quantile(x,0.2))

temp3

"""\*\*c. Make a heatmap of this\*\*

i. Annotated

"""

f,ax = plt.subplots(figsize=(5, 5))

sns.heatmap(temp3, annot=True, linewidths=.5, fmt='.1f',ax=ax)

plt.show()

"""ii.Greens color map"""

f,ax = plt.subplots(figsize=(5, 5))

sns.heatmap(temp3, annot=True, linewidths=.5, cmap='Greens',fmt='.1f',ax=ax)

plt.show()

"""\*\*d. What’s your inference? Are lighter apps preferred in all categories? Heavier? Some?\*\*

Based on analysis, its not true that lighter apps are preferred in all categories. Because apps with size 40k-60k and 80k-100k have got the highest rating in all cateegories. So, in general we can conclude that heavier apps are preferred in all categories.

"""