EDA on Facebook Utilization

October 30, 2022

1 Exploratory Data Analysis (EDA) on Facebook Utilization

1.1 Installing Libraries

[3]:
```
!pip install -q datascience
!pip install -q pandas-profiling
```

1.2 Upgrading Libraries

[4]:
```
!pip install -q --upgrade pandas-profiling
```

1.3 Importing Libraries

[5]:
```
import pandas as pd
from pandas_profiling import ProfileReport
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import scipy as sp
```

1.4 Data Acquisition

[18]:
```
data = pd.read_csv(filepath_or_buffer = 'https://raw.githubusercontent.com/
↪insaid2018/Term-1/master/Data/Projects/facebook_data.csv')
print('Data Shape:', data.shape)
data.head()
```

```
Data Shape: (99003, 15)
```

[18]:

	userid	age	dob_day	dob_year	dob_month	gender	tenure	friend_count	\
0	2094382	14	19	1999	11	male	266.0	0	
1	1192601	14	2	1999	11	female	6.0	0	
2	2083884	14	16	1999	11	male	13.0	0	
3	1203168	14	25	1999	12	female	93.0	0	
4	1733186	14	4	1999	12	male	82.0	0	

	friendships_initiated	likes	likes_received	mobile_likes	\

```
0                     0      0              0             0
1                     0      0              0             0
2                     0      0              0             0
3                     0      0              0             0
4                     0      0              0             0

   mobile_likes_received  www_likes  www_likes_received
0                      0          0                   0
1                      0          0                   0
2                      0          0                   0
3                      0          0                   0
4                      0          0                   0
```

1.5 Data Description

[19]:
```
data.describe()
```

[19]:
```
             userid           age       dob_day      dob_year     dob_month  \
count  9.900300e+04  99003.000000  99003.000000  99003.000000  99003.000000
mean   1.597045e+06     37.280224     14.530408   1975.719776      6.283365
std    3.440592e+05     22.589748      9.015606     22.589748      3.529672
min    1.000008e+06     13.000000      1.000000   1900.000000      1.000000
25%    1.298806e+06     20.000000      7.000000   1963.000000      3.000000
50%    1.596148e+06     28.000000     14.000000   1985.000000      6.000000
75%    1.895744e+06     50.000000     22.000000   1993.000000      9.000000
max    2.193542e+06    113.000000     31.000000   2000.000000     12.000000

             tenure  friend_count  friendships_initiated         likes  \
count  99001.000000  99003.000000           99003.000000  99003.000000
mean     537.887375    196.350787             107.452471    156.078785
std      457.649874    387.304229             188.786951    572.280681
min        0.000000      0.000000               0.000000      0.000000
25%      226.000000     31.000000              17.000000      1.000000
50%      412.000000     82.000000              46.000000     11.000000
75%      675.000000    206.000000             117.000000     81.000000
max     3139.000000   4923.000000            4144.000000  25111.000000

       likes_received  mobile_likes  mobile_likes_received     www_likes  \
count    99003.000000  99003.000000           99003.000000  99003.000000
mean       142.689363    106.116300              84.120491     49.962425
std       1387.919613    445.252985             839.889444    285.560152
min          0.000000      0.000000               0.000000      0.000000
25%          1.000000      0.000000               0.000000      0.000000
50%          8.000000      4.000000               4.000000      0.000000
75%         59.000000     46.000000              33.000000      7.000000
max     261197.000000  25111.000000          138561.000000  14865.000000
```

```
       www_likes_received
count        99003.000000
mean            58.568831
std            601.416348
min              0.000000
25%              0.000000
50%              2.000000
75%             20.000000
max         129953.000000
```

1.6 Data Information

[20]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99003 entries, 0 to 99002
Data columns (total 15 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   userid                 99003 non-null  int64
 1   age                    99003 non-null  int64
 2   dob_day                99003 non-null  int64
 3   dob_year               99003 non-null  int64
 4   dob_month              99003 non-null  int64
 5   gender                 98828 non-null  object
 6   tenure                 99001 non-null  float64
 7   friend_count           99003 non-null  int64
 8   friendships_initiated  99003 non-null  int64
 9   likes                  99003 non-null  int64
 10  likes_received         99003 non-null  int64
 11  mobile_likes           99003 non-null  int64
 12  mobile_likes_received  99003 non-null  int64
 13  www_likes              99003 non-null  int64
 14  www_likes_received     99003 non-null  int64
dtypes: float64(1), int64(13), object(1)
memory usage: 11.3+ MB
```

1.7 Data Pre-Profiling

[21]:
```
profile = data.profile_report(title='Pre Profiling Report')
profile.to_file(output_file="pre_profiling.html")
```

```
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]

Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]

Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]

Export report to file:   0%|          | 0/1 [00:00<?, ?it/s]
```

1.8 Data Pre-Processing

1.8.1 Checking for missing data

[22]:
```
data.isnull().sum()
```

[22]:
```
userid                     0
age                        0
dob_day                    0
dob_year                   0
dob_month                  0
gender                   175
tenure                     2
friend_count               0
friendships_initiated      0
likes                      0
likes_received             0
mobile_likes               0
mobile_likes_received      0
www_likes                  0
www_likes_received         0
dtype: int64
```

1.8.2 Eliminating rows having null value in tenure column

[23]:
```
data = data[~data.tenure.isnull()].copy()
data.isnull().sum()
```

[23]:
```
userid                     0
age                        0
dob_day                    0
dob_year                   0
dob_month                  0
gender                   175
tenure                     0
friend_count               0
friendships_initiated      0
likes                      0
likes_received             0
mobile_likes               0
mobile_likes_received      0
www_likes                  0
www_likes_received         0
dtype: int64
```

1.8.3 Filling the missing gender values with the mode of the gender column

[24]:
```
gender_mode = data.gender.mode()[0]
data.gender.fillna(gender_mode, inplace = True)
data.gender.isnull().sum()
```

[24]:
```
0
```

1.8.4 Again checking for missing data

[25]:
```
data.isnull().sum()
```

[25]:
```
userid                   0
age                      0
dob_day                  0
dob_year                 0
dob_month                0
gender                   0
tenure                   0
friend_count             0
friendships_initiated    0
likes                    0
likes_received           0
mobile_likes             0
mobile_likes_received    0
www_likes                0
www_likes_received       0
dtype: int64
```

1.9 Data Post-Profiling

[26]:
```
profile = data.profile_report(title='Post Profiling Report')
profile.to_file(output_file="post_profiling.html")
```

```
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]

Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]

Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]

Export report to file:   0%|          | 0/1 [00:00<?, ?it/s]
```

1.10 Exploratory Data Analysis (EDA)

1.10.1 Counting number of users based on birth year

[27]:
```
data['dob_year'].value_counts()
```

[27]:
```
1995    5196
1990    4404
```

```
1994    4391
1993    3769
1992    3671
        …
1926      42
1901      18
1902      18
1903      15
1904       9
Name: dob_year, Length: 101, dtype: int64
```

1.10.2 Counting number of users based on gender including NaN

[28]:
```
data['gender'].value_counts(dropna=False)
```

[28]:
```
male      58749
female    40252
Name: gender, dtype: int64
```

1.10.3 Dividing the dob_year into groups of 10

[29]:
```
labels=['1900-1910','1911-1920','1921-1930','1931-1940','1941-1950','1951-1960','1961-1970','1
data['dob_year_group'] = pd.cut(data.dob_year,bins=np.
 ↪arange(1900,2010,10),labels=labels,right=True)
data.head()
```

[29]:

	userid	age	dob_day	dob_year	dob_month	gender	tenure	friend_count	\
0	2094382	14	19	1999	11	male	266.0	0	
1	1192601	14	2	1999	11	female	6.0	0	
2	2083884	14	16	1999	11	male	13.0	0	
3	1203168	14	25	1999	12	female	93.0	0	
4	1733186	14	4	1999	12	male	82.0	0	

	friendships_initiated	likes	likes_received	mobile_likes	\
0	0	0	0	0	
1	0	0	0	0	
2	0	0	0	0	
3	0	0	0	0	
4	0	0	0	0	

	mobile_likes_received	www_likes	www_likes_received	dob_year_group
0	0	0	0	1991-2000
1	0	0	0	1991-2000
2	0	0	0	1991-2000
3	0	0	0	1991-2000
4	0	0	0	1991-2000

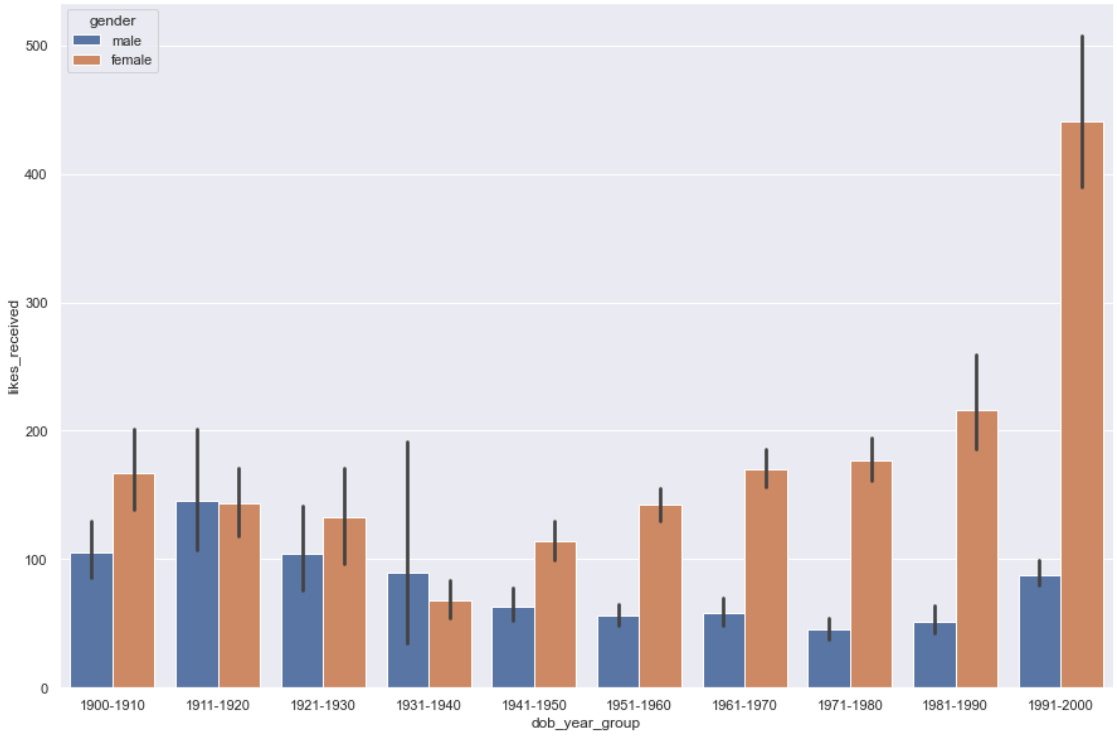
1.10.4 Counting values in dob_year_group

[30]:
```
data.dob_year_group.value_counts()
```

[30]:
```
1991-2000    31455
1981-1990    25080
1971-1980    10990
1961-1970     9298
1951-1960     8921
1941-1950     5935
1900-1910     3140
1931-1940     1787
1911-1920     1435
1921-1930      758
Name: dob_year_group, dtype: int64
```

1.10.5 Plotting barplot with dob_year_group as x-axis and likes_received as y-axis showing females got more likes than males

[82]:
```
sns.barplot(x=data['dob_year_group'],y=data['likes_received'],hue=data.gender)
sns.set(rc = {'figure.figsize':(15,10)})
```



1.10.6 Number of users having some likes

[33]:
```
np.count_nonzero(data.likes_received)
```

[33]:
```
74573
```

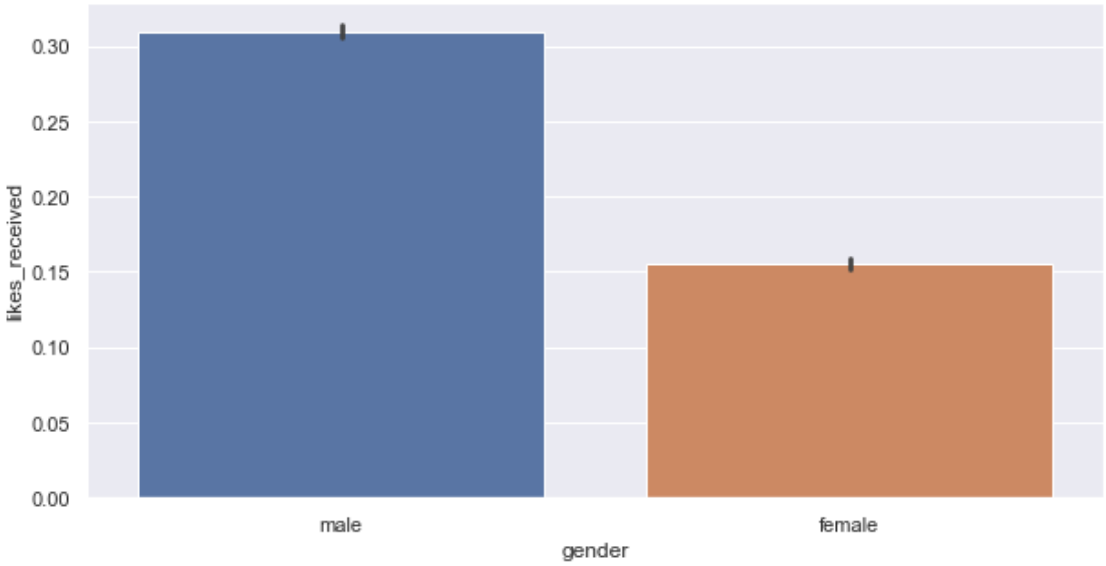
1.10.7 Number of users having zero likes

[34]:
```
fc=data.likes_received==0
fc.value_counts()
```

[34]:
```
False    74573
True     24428
Name: likes_received, dtype: int64
```

1.10.8 Plotting the gender vs zero likes count users

[36]:
```
sns.barplot(y=data.likes_received==0,x=data.gender)
sns.set(rc = {'figure.figsize':(10,5)})
```



1.10.9 Top 5 users having highest friend count

[42]:
```
data.sort_values(by='friend_count',ascending=False)[:5]
```

[42]:
```
        userid  age  dob_day  dob_year  dob_month  gender  tenure  \
98159  2090699  103        1      1910         10  female   783.0
98026  1660276   66        1      1947          7    male   716.0
98184  1926655   19        1      1994          8  female   469.0
```

```
98797  1685573   38       20      1975          2    male  1930.0
98087  1386477   61       30      1952          3    male  1210.0

       friend_count  friendships_initiated  likes  likes_received  \
98159          4923                     96     26              80
98026          4917                    431      0             114
98184          4863                    241     37             166
98797          4845                    942   1768            4886
98087          4844                    561      7             247

       mobile_likes  mobile_likes_received  www_likes  www_likes_received  \
98159            26                     71          0                   9
98026             0                     69          0                  45
98184            36                    117          1                  49
98797          1208                   1619        560                3267
98087             5                     96          2                 151

      dob_year_group
98159      1900-1910
98026      1941-1950
98184      1991-2000
98797      1971-1980
98087      1951-1960
```

1.10.10 Top 10 users getting highest mobile likes

[49]:
```
temp = data.sort_values(by='mobile_likes_received',ascending=False)[:10]
temp
```

[49]:
```
        userid  age  dob_day  dob_year  dob_month  gender  tenure  \
77121  1441676   20        5      1993          8  female   253.0
94906  1674584   17       14      1996          8  female   401.0
98822  1715925   23        4      1990          9  female   705.0
98994  2063006   20        4      1993          1  female   402.0
98878  1053087   23        6      1990          6    male   596.0
98937  1559908   20        4      1993         12  female  1334.0
49230  1432020   20       12      1993          1    male   245.0
98936  1781243   17        1      1996          5  female   976.0
98973  1836366   26        3      1987          8  female  1669.0
98773  2042824   18       25      1995          1    male    51.0

       friend_count  friendships_initiated  likes  likes_received  \
77121           230                     73   2078          178166
94906           818                    395   1016          261197
98822          4077                    793   1877          152014
98994          1988                    332   7351          106025
98878          4320                    836   2996           82623
```

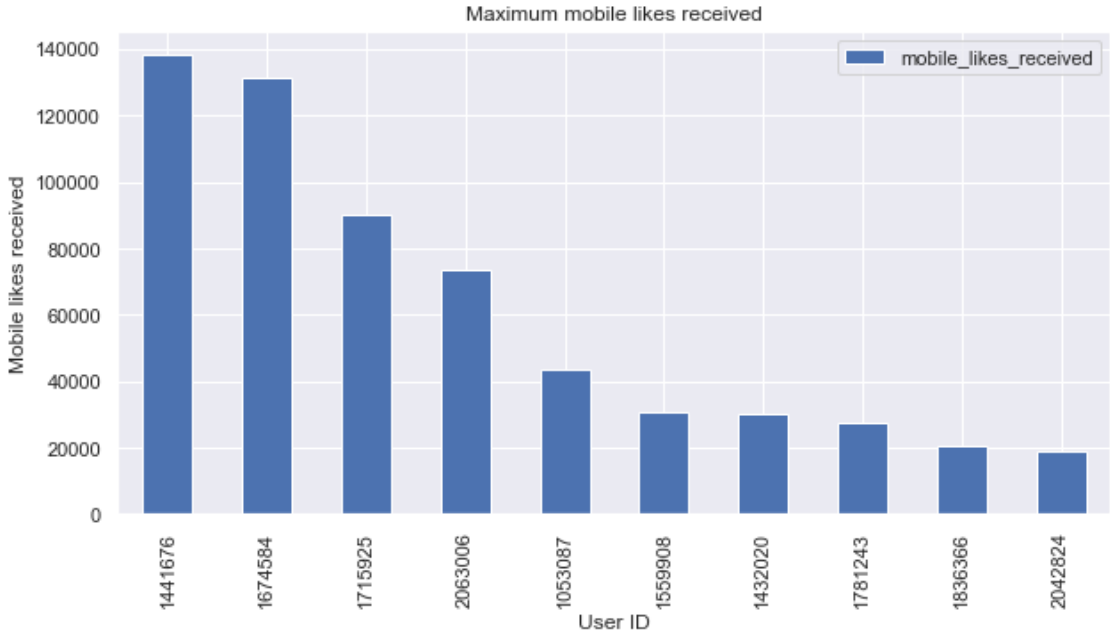
```
98937          4622                    1819   4280           45633
49230            79                      50    477           53534
98936          3683                     755  10478           42449
98973          4240                     857   4794           28778
98773          4817                      32   1346           52964

       mobile_likes  mobile_likes_received  www_likes  www_likes_received  \
77121          1982                 138561         96               39605
94906           659                 131244        357              129953
98822            80                  89911       1797               62103
98994          7248                  73333        103               32692
98878           179                  43410       2817               39213
98937           472                  30754       3808               14879
49230            78                  30387        399               23147
98936           246                  27353      10232               15096
98973          1153                  20770       3641                8008
98773          1342                  18925          4               34039

      dob_year_group
77121      1991-2000
94906      1991-2000
98822      1981-1990
98994      1991-2000
98878      1981-1990
98937      1991-2000
49230      1991-2000
98936      1991-2000
98973      1981-1990
98773      1991-2000
```

1.10.11 Plotting barplot showing users getting highest mobile likes and the corresponding mobile likes received count

[52]:
```
temp.plot(x='userid',y='mobile_likes_received',kind='bar')
plt.ylabel("Mobile likes received")
plt.xlabel("User ID")
plt.title("Maximum mobile likes received")
plt.show()
```


Maximum mobile likes received
mobile_likes_received
Mobile likes received
140000
120000
100000
80000
60000
40000
20000
0
1441676
1674584
1715925
2063006
1053087
1559908
1432020
1781243
1836366
2042824
User ID