import numpy as np import pandas as pd

*# Input data files are available in the "../input/" directory.*

*# For example, running this (by clicking run or pressing Shift+Enter) will list the files in the input directory*

import matplotlib.pyplot as plt

%matplotlib inline import seaborn as sns

df= pd.read\_csv("pseudo\_facebook.csv") df.head()

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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

df.describe()

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  \  count 99001.000000 | | friend\_count  99003.000000 | friendships\_initiated  99003.000000 | | likes  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 |

features = df.columns features

Index(['userid', 'age', 'dob\_day', 'dob\_year', 'dob\_month', 'gender', 'tenure',

'friend\_count', 'friendships\_initiated', 'likes', 'likes\_received',

'mobile\_likes', 'mobile\_likes\_received', 'www\_likes', 'www\_likes\_received'],

dtype='object') df.info()

<class 'pandas.core.frame.DataFrame'> RangeIndex: 99003 entries, 0 to 99002 Data columns (total 15 columns):

# Column Non-Null Count Dtype

1. userid 99003 non-null int64
2. age 99003 non-null int64
3. dob\_day 99003 non-null int64
4. dob\_year 99003 non-null int64
5. dob\_month 99003 non-null int64
6. gender 98828 non-null object
7. tenure 99001 non-null float64
8. friend\_count 99003 non-null int64
9. friendships\_initiated 99003 non-null int64
10. likes 99003 non-null int64
11. likes\_received 99003 non-null int64
12. mobile\_likes 99003 non-null int64
13. mobile\_likes\_received 99003 non-null int64
14. www\_likes 99003 non-null int64
15. www\_likes\_received 99003 non-null int64 dtypes: float64(1), int64(13), object(1)

memory usage: 11.3+ MB df.hist()

array([[<AxesSubplot:title={'center':'userid'}>,

<AxesSubplot:title={'center':'age'}>,

<AxesSubplot:title={'center':'dob\_day'}>,

<AxesSubplot:title={'center':'dob\_year'}>], [<AxesSubplot:title={'center':'dob\_month'}>,

<AxesSubplot:title={'center':'tenure'}>,

<AxesSubplot:title={'center':'friend\_count'}>,

<AxesSubplot:title={'center':'friendships\_initiated'}>], [<AxesSubplot:title={'center':'likes'}>,

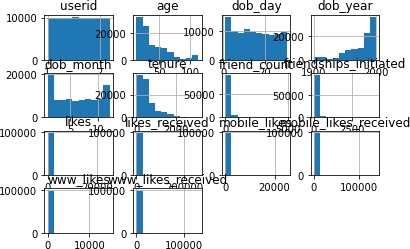
<AxesSubplot:title={'center':'likes\_received'}>,

<AxesSubplot:title={'center':'mobile\_likes'}>,

<AxesSubplot:title={'center':'mobile\_likes\_received'}>], [<AxesSubplot:title={'center':'www\_likes'}>,

<AxesSubplot:title={'center':'www\_likes\_received'}>,

<AxesSubplot:>, <AxesSubplot:>]], dtype=object)



Num\_features = [feature **for** feature **in** features **if** df[feature].dtype !

= object]

Cat\_features = [feature **for** feature **in** features **if** df[feature].dtype

== object] Num\_features

['userid',

'age', 'dob\_day', 'dob\_year', 'dob\_month', 'tenure', 'friend\_count',

'friendships\_initiated', 'likes', 'likes\_received', 'mobile\_likes', 'mobile\_likes\_received',

'www\_likes', 'www\_likes\_received']

Cat\_features ['gender']

df = df.fillna(method="bfill")

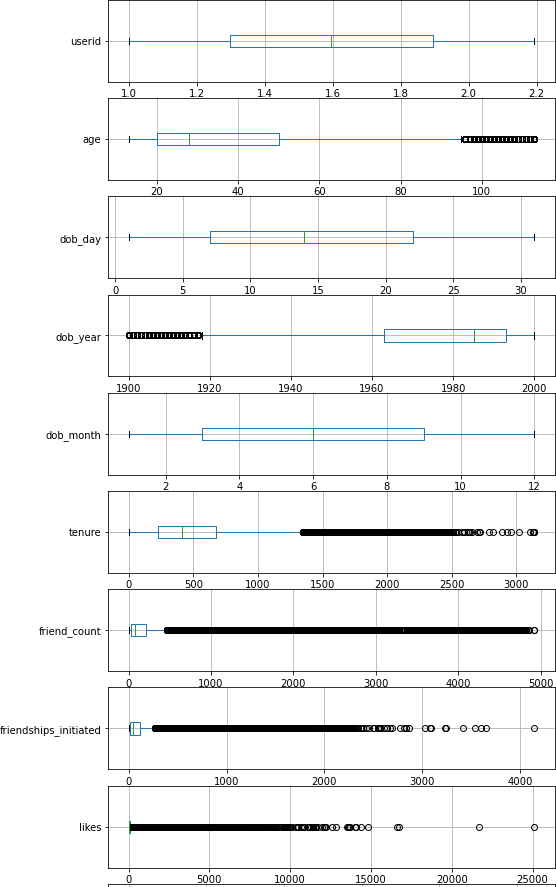
# Data exploration

In data exploration, we'll plot histograms, boxplots , correaltion matrix,subplot of all numerical features and see countplot of the categorical variables

fig, axes = plt.subplots(14,1 ,figsize=(8,25))

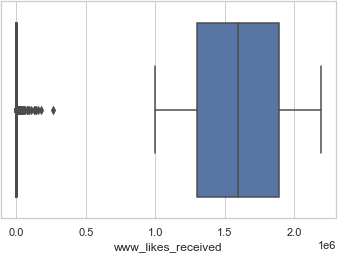
**for** i,c **in** enumerate(Num\_features):

f = df[[c]].boxplot(ax=axes[i], vert=False)



import seaborn as sns sns.set\_theme(style="whitegrid") **for** i, c **in** enumerate(Num\_features):

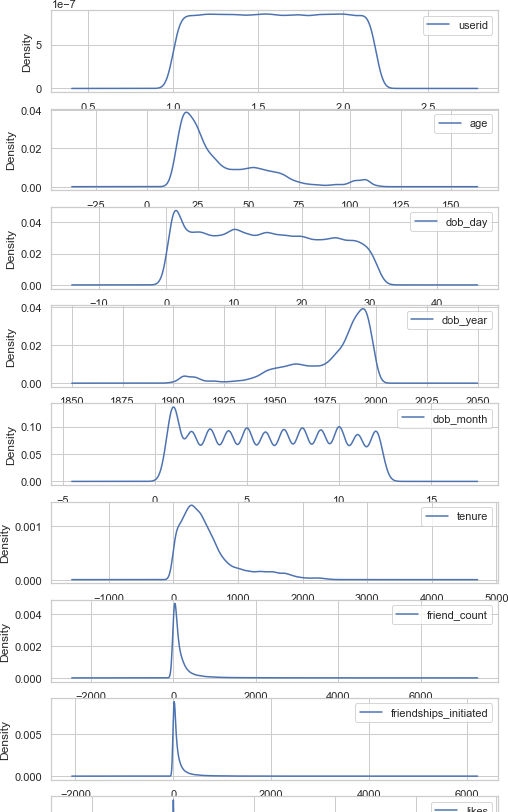
ax = sns.boxplot(x = df[c])



fig, axes = plt.subplots(14,1 ,figsize=(8,25))

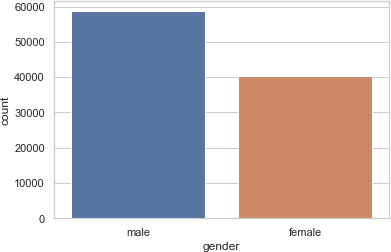
**for** i,c **in** enumerate(Num\_features):

f = df[[c]].plot(kind = 'kde', ax=axes[i])



*# For categorical features*

countplot = sns.countplot(x="gender",data=df)

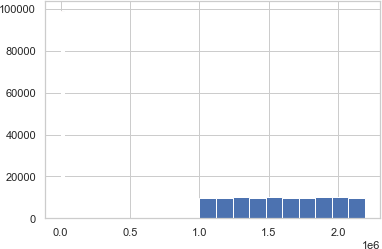


df\_mean = df[Num\_features].mean() df\_mean

|  |  |
| --- | --- |
| userid | 1.597045e+06 |
| age | 3.728022e+01 |
| dob\_day | 1.453041e+01 |
| dob\_year | 1.975720e+03 |
| dob\_month | 6.283365e+00 |
| tenure | 5.378966e+02 |
| friend\_count | 1.963508e+02 |
| friendships\_initiated | 1.074525e+02 |
| likes | 1.560788e+02 |
| likes\_received | 1.426894e+02 |
| mobile\_likes | 1.061163e+02 |
| mobile\_likes\_received | 8.412049e+01 |
| www\_likes | 4.996243e+01 |
| www\_likes\_received | 5.856883e+01 |
| dtype: float64 |  |

df\_n = df.groupby('gender').mean()

**for** r **in** Num\_features: df[r].hist()



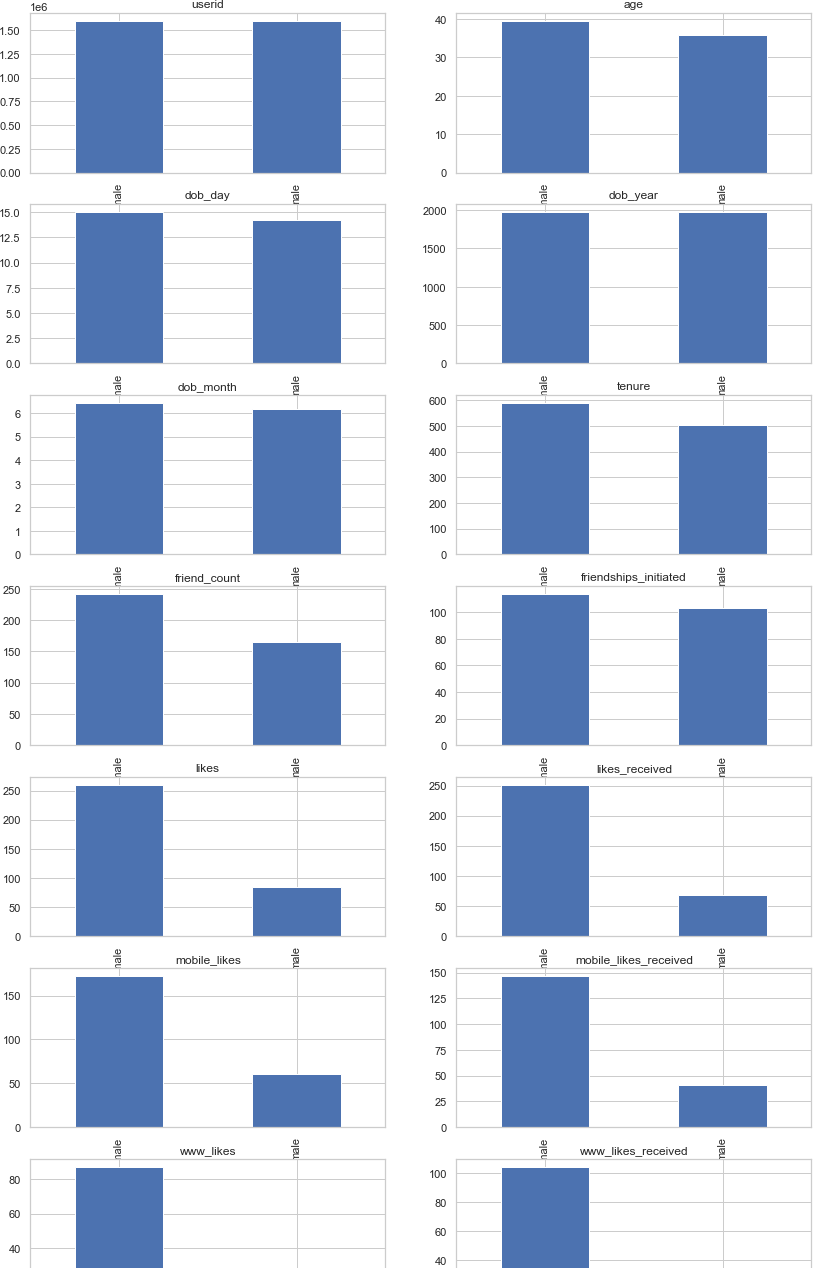
*#df\_col = df\_n.columns*

*# Relationship between all features mean and our targer feature*

fix, axes = plt.subplots(7,2, figsize=(14,24)) axes = [ax **for** axes\_row **in** axes **for** ax **in** axes\_row] **for** i,c **in** enumerate(df[Num\_features]):

df\_n = df.groupby('gender')[c].mean()

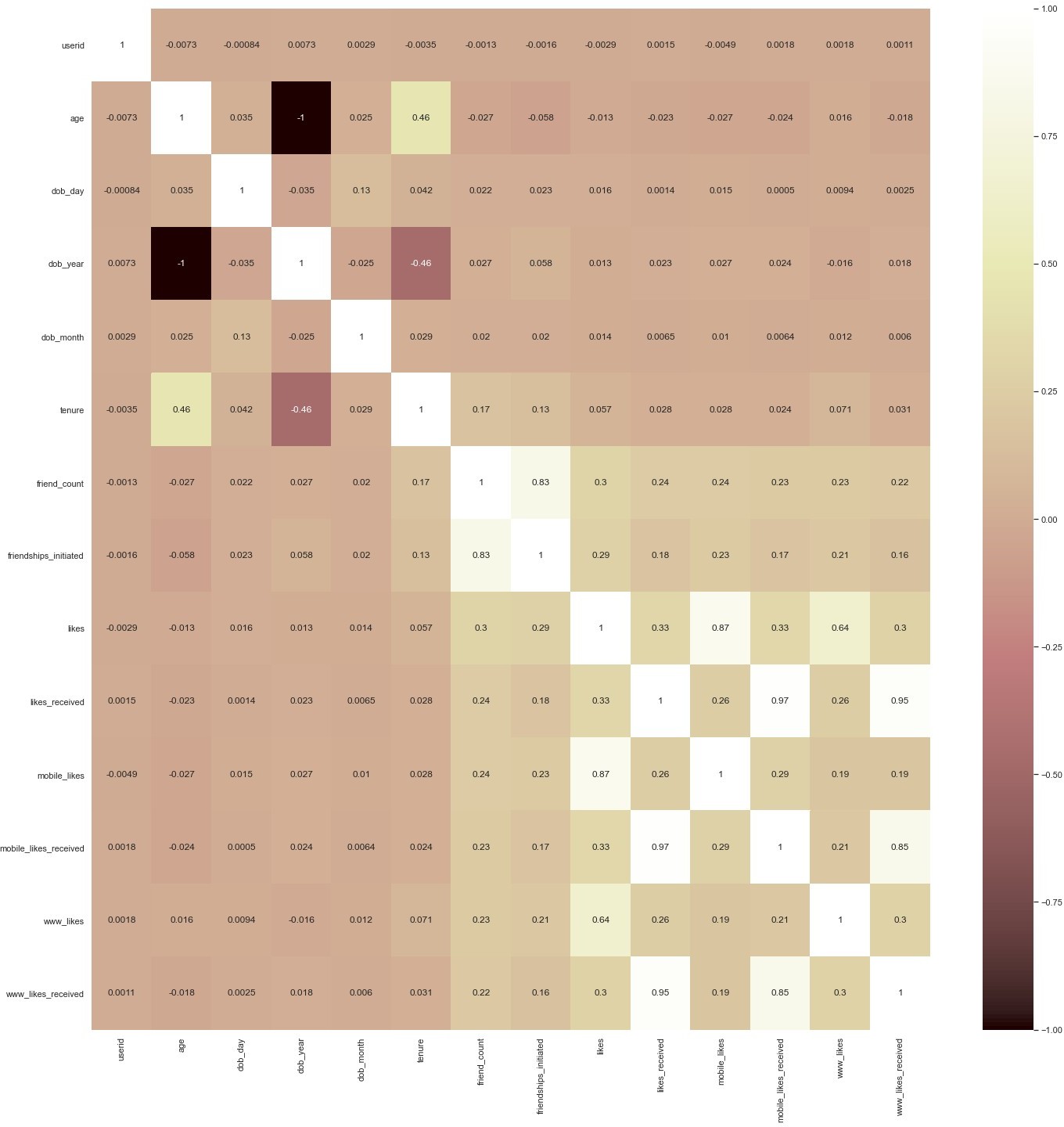
plot = df\_n.plot(kind='bar',title=c,ax=axes[i])



*# Pearson Correlation matrix*

corr\_matrix = df[Num\_features].corr(method='pearson') plt.figure(figsize=(24,24))

correc = sns.heatmap(corr\_matrix, annot=True, cmap = 'pink')



*# Find features with high and low correlation* df['gender'] = df.gender.map({"male":0, "female":1}) df

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | userid | age | dob\_day | dob\_year | dob\_month | gender | tenure | \ |
| 0 | 2094382 | 14 | 19 | 1999 | 11 | 0 | 266.0 |  |
| 1 | 1192601 | 14 | 2 | 1999 | 11 | 1 | 6.0 |  |
| 2 | 2083884 | 14 | 16 | 1999 | 11 | 0 | 13.0 |  |
| 3 | 1203168 | 14 | 25 | 1999 | 12 | 1 | 93.0 |  |
| 4 | 1733186 | 14 | 4 | 1999 | 12 | 0 | 82.0 |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ... | ... | ... | ... | ... | ... | ... | ... | |
| 98998 | 1268299 | 68 | 4 | 1945 | 4 | 1 | 541.0 | |
| 98999 | 1256153 | 18 | 12 | 1995 | 3 | 1 | 21.0 | |
| 99000 | 1195943 | 15 | 10 | 1998 | 5 | 1 | 111.0 | |
| 99001 | 1468023 | 23 | 11 | 1990 | 4 | 1 | 416.0 | |
| 99002 | 1397896 | 39 | 15 | 1974 | 5 | 1 | 397.0 | |
|  | friend\_count | | friendships\_initiated | | likes | likes\_received | | \ |
| 0 | 0 | | 0 | | 0 | 0 | |  |
| 1 | 0 | | 0 | | 0 | 0 | |  |
| 2 | 0 | | 0 | | 0 | 0 | |  |
| 3 | 0 | | 0 | | 0 | 0 | |  |
| 4  ... 98998 | 0  ... 2118 | | 0  ... 341 | | 0  ... 3996 | 0  ... 18089 | |  |
| 98999 | 1968 | | 1720 | | 4401 | 13412 | |  |
| 99000 | 2002 | | 1524 | | 11959 | 12554 | |  |
| 99001 | 2560 | | 185 | | 4506 | 6516 | |  |
| 99002 | 2049 | | 768 | | 9410 | 12443 | |  |
| mobile\_likes | | | mobile\_likes\_received | | www\_likes | | | |
| www\_likes\_received | | | | | | | | |
| 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  ...  ... 98998 | ... 3505 | | ... 11887 | | ... 491 | | | |
| 6202 |  | |  | |  | | | |
| 98999 | 4399 | | 10592 | | 2 | | | |
| 2820 |  | |  | |  | | | |
| 99000 | 11959 | | 11462 | | 0 | | | |
| 1092 |  | |  | |  | | | |
| 99001 | 4506 | | 5760 | | 0 | | | |
| 756 |  | |  | |  | | | |
| 99002 | 9410 | | 9530 | | 0 | | | |
| 2913 |  | |  | |  | | | |

[99003 rows x 15 columns] det = df.corr()

det['gender'].sort\_values(ascending = False)

|  |  |
| --- | --- |
| gender | 1.000000 |
| likes | 0.150567 |
| mobile\_likes | 0.124310 |
| www\_likes | 0.107918 |
| friend\_count | 0.097638 |
| tenure | 0.093523 |
| age | 0.082228 |
| likes\_received | 0.064988 |
| www\_likes\_received | 0.063122 |
| mobile\_likes\_received | 0.062193 |
| dob\_day | 0.046112 |
| dob\_month | 0.035472 |
| friendships\_initiated | 0.028335 |
| userid | 0.001480 |
| dob\_year | -0.082228 |

Name: gender, dtype: float64

# Data cleaning and feature engineering

1) All null values were found and replaced by the before fill method (as the percentage of null was less than 0.5%). 2) Numerical and categorical variables were seperated . 3) After plotting correlation matrix, we dropped least significant features(with relation to the feature'gender') . 4) Robust scalar was used to scale every feature{ also removing outliers}.

**Key Findings and Insights, which synthesizes the results of Exploratory Data Analysis in an insightful and actionable manner** 1) Correlation and significance of all features were found out. 2) distribution of values in all features were seen. 3) Relationship between all features mean and our targer feature was seen

df\_final = df.drop('dob\_day',1) df\_final = df\_final.drop('dob\_month',1)

df\_final = df\_final.drop('friendships\_initiated',1) df\_final = df\_final.drop('userid',1)

df\_final

<ipython-input-32-705c7589e56e>:1: FutureWarning: In a future version of pandas all arguments of DataFrame.drop except for the argument 'labels' will be keyword-only

df\_final = df.drop('dob\_day',1)

<ipython-input-32-705c7589e56e>:2: FutureWarning: In a future version of pandas all arguments of DataFrame.drop except for the argument 'labels' will be keyword-only

df\_final = df\_final.drop('dob\_month',1)

<ipython-input-32-705c7589e56e>:3: FutureWarning: In a future version of pandas all arguments of DataFrame.drop except for the argument 'labels' will be keyword-only

df\_final = df\_final.drop('friendships\_initiated',1)

<ipython-input-32-705c7589e56e>:4: FutureWarning: In a future version of pandas all arguments of DataFrame.drop except for the argument 'labels' will be keyword-only

df\_final = df\_final.drop('userid',1)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| age dob\_year likes\_received \ | | | gender | tenure | friend\_count | likes |
| 0 | 14 | 1999 | 0 | 266.0 | 0 | 0 |
| 0 |  |  |  |  |  |  |
| 1 | 14 | 1999 | 1 | 6.0 | 0 | 0 |
| 0 |  |  |  |  |  |  |
| 2 | 14 | 1999 | 0 | 13.0 | 0 | 0 |
| 0 |  |  |  |  |  |  |
| 3 | 14 | 1999 | 1 | 93.0 | 0 | 0 |
| 0 |  |  |  |  |  |  |
| 4 | 14 | 1999 | 0 | 82.0 | 0 | 0 |
| 0  ... | ... | ... | ... | ... | ... | ... |
| ... 98998 | 68 | 1945 | 1 | 541.0 | 2118 | 3996 |
| 18089 |  |  |  |  |  |  |
| 98999 | 18 | 1995 | 1 | 21.0 | 1968 | 4401 |
| 13412 |  |  |  |  |  |  |
| 99000 | 15 | 1998 | 1 | 111.0 | 2002 | 11959 |
| 12554 |  |  |  |  |  |  |
| 99001 | 23 | 1990 | 1 | 416.0 | 2560 | 4506 |
| 6516 |  |  |  |  |  |  |
| 99002 | 39 | 1974 | 1 | 397.0 | 2049 | 9410 |
| 12443 |  |  |  |  |  |  |

mobile\_likes mobile\_likes\_received www\_likes www\_likes\_received

|  |  |  |  |
| --- | --- | --- | --- |
| 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 |  |  |  |
| ... | ... | ... | ... |
| ... |  |  |  |
| 98998 | 3505 | 11887 | 491 |
| 6202 |  |  |  |
| 98999 | 4399 | 10592 | 2 |
| 2820 |  |  |  |
| 99000 | 11959 | 11462 | 0 |
| 1092 |  |  |  |
| 99001 | 4506 | 5760 | 0 |
| 756 |  |  |  |
| 99002 | 9410 | 9530 | 0 |
| 2913 |  |  |  |

[99003 rows x 11 columns] df\_final.hist()

array([[<AxesSubplot:title={'center':'age'}>,

<AxesSubplot:title={'center':'dob\_year'}>,

<AxesSubplot:title={'center':'gender'}>], [<AxesSubplot:title={'center':'tenure'}>,

<AxesSubplot:title={'center':'friend\_count'}>,

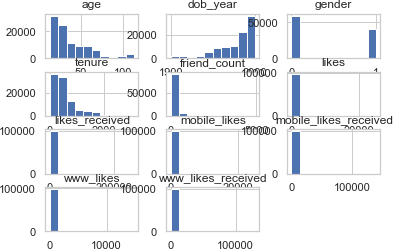
<AxesSubplot:title={'center':'likes'}>], [<AxesSubplot:title={'center':'likes\_received'}>,

<AxesSubplot:title={'center':'mobile\_likes'}>,

<AxesSubplot:title={'center':'mobile\_likes\_received'}>], [<AxesSubplot:title={'center':'www\_likes'}>,

<AxesSubplot:title={'center':'www\_likes\_received'}>,

<AxesSubplot:>]], dtype=object)



from sklearn.preprocessing import RobustScaler from pandas import DataFrame

transformation = RobustScaler()

X = transformation.fit\_transform(df\_final) dataset = DataFrame(X) print(dataset.describe())

dataset.hist()

0 1 2 3

4 \

count 99003.000000 99003.000000 99003.000000 99003.000000

99003.000000

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| mean | 0.309341 | -0.309341 | 0.407341 | 0.280393 |  |
| 0.653433 |  |  |  |  |
| std | 0.752992 | 0.752992 | 0.491342 | 1.019284 |
| 2.213167 |  |  |  |  |
| min | -0.500000 | -2.833333 | 0.000000 | -0.917595 | - |
| 0.468571 |  |  |  |  |  |
| 25% | -0.266667 | -0.733333 | 0.000000 | -0.414254 | - |
| 0.291429 |  |  |  |  |  |
| 50% | 0.000000 | 0.000000 | 0.000000 | 0.000000 |  |
| 0.000000 |  |  |  |  |  |
| 75% | 0.733333 | 0.266667 | 1.000000 | 0.585746 |  |
| 0.708571 |  |  |  |  |  |
| max | 2.833333 | 0.500000 | 1.000000 | 6.073497 |  |
| 27.662857 |  |  |  |  |  |

5 6 7 8

9 \

count 99003.000000 99003.000000 99003.000000 99003.000000

99003.000000

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| mean | 1.813485 | 2.322230 | 2.219920 | 2.427894 |
| 7.137489 |  |  |  |  |
| std | 7.153509 | 23.929648 | 9.679413 | 25.451195 |
| 40.794307 |  |  |  |  |
| min | -0.137500 | -0.137931 | -0.086957 | -0.121212 |
| 0.000000 |  |  |  |  |
| 25% | -0.125000 | -0.120690 | -0.086957 | -0.121212 |
| 0.000000 |  |  |  |  |
| 50% | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 0.000000 |  |  |  |  |
| 75% | 0.875000 | 0.879310 | 0.913043 | 0.878788 |
| 1.000000 |  |  |  |  |
| max | 313.750000 | 4503.258621 | 545.804348 | 4198.696970 |

2123.571429

|  |  |
| --- | --- |
|  | 10 |
| count | 99003.000000 |
| mean | 2.828442 |
| std | 30.070817 |
| min | -0.100000 |
| 25% | -0.100000 |
| 50% | 0.000000 |
| 75% | 0.900000 |
| max | 6497.550000 |

array([[<AxesSubplot:title={'center':'0'}>,

<AxesSubplot:title={'center':'1'}>,

<AxesSubplot:title={'center':'2'}>], [<AxesSubplot:title={'center':'3'}>,

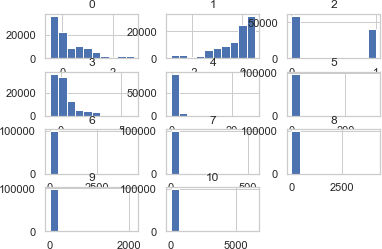
<AxesSubplot:title={'center':'4'}>,

<AxesSubplot:title={'center':'5'}>], [<AxesSubplot:title={'center':'6'}>,

<AxesSubplot:title={'center':'7'}>,

<AxesSubplot:title={'center':'8'}>], [<AxesSubplot:title={'center':'9'}>,

<AxesSubplot:title={'center':'10'}>, <AxesSubplot:>]], dtype=object)



**Findings** 1) likes, mobile\_likes,, www\_likes, friend\_count, tenure. age, likes\_recieved, www\_likes\_recieved, mobile\_likes recieved are positively related to the target feature(Gender). 2) dob\_year is negatively related to gender. 3) The rest features do not have any significant relationship/correlation with gender, thus can be dropped.

Chosen features for next ML/DL algorithm are likes, mobile\_likes,, www\_likes, friend\_count, tenure. age, likes\_recieved, www\_likes\_recieved, mobile\_likes recieved, dob\_year.

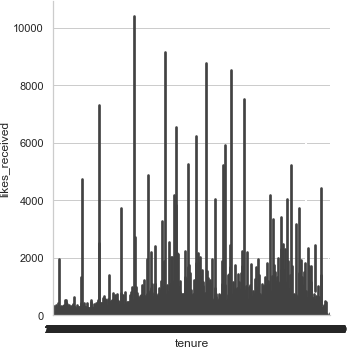
We haven't been asked to apply ML/DL algos in this question, so we'll jump to the Hypothesis part directly.

# Different hypothesis about the dataset

1) The mean of likes recieved by the 2 genders are different. 2) Likes increase with the passage of tenure {likes\_recieved is directly proportional to tenure}. 3) Female recieve more likes than male for the same friendship initiation.

sns.catplot(x='tenure', y='likes\_received', kind="bar", data=df)

<seaborn.axisgrid.FacetGrid at 0x1ca3de5f250>

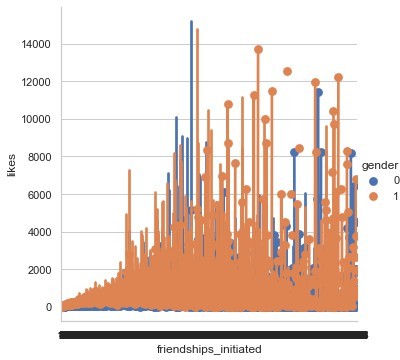


# Conducting a formal significance test for one of the hypotheses and discuss the results

Hypothesis 2 was null hypothesis as we call nullify it with the results obtained from above plot. The likes do not necessarily increase with increase in tenure.

sns.catplot(x="friendships\_initiated", y="likes", hue="gender", kind="point", data=df)

<seaborn.axisgrid.FacetGrid at 0x1ca3bc29f70>



# Suggestions for next steps in analyzing this data

1. Maybe power transformations can be tried to make data more gaussian like distribution.
2. Outlier removal algorithms Density based clustering algorithms can be used to clean the data.

# Summarizes the quality of this data set and a request for additional data if needed

The quality of data was average. There weren't many highly coorelated features to our targer 'gender'. Also, the mean and median of the features was far away, thus indicating that outliers may be presesnt in it.(for that we used robust scaler). The dataset was more inclined towards male {the value of almost all features was more inclined towards males than females(subplots of averages)}.