Problem Statement

Perform Exploratory Data Analysis(EDA) on a given sample Dataset of Facebook Data.

Installing, Importing and Upgrading Libraries

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
!pip install -q datascience
# For pandas profiling
!pip install -q pandas-profiling
# Library to generate basic statistics about data
!pip install -q --upgrade pandas-profiling
# Upgrading pandas profiling to the latest version
from pandas_profiling import ProfileReport
# Import Pandas Profiling (To generate Univariate Analysis)
```

Data Collection using Pandas

In [2]:
```
data=pd.read_csv("https://raw.githubusercontent.com/insaid2018/Term-1/master/Data/Projects
```

Data Information

In [3]:
```
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
```

Data Description

In [5]:
```
data.describe()
```

Out[5]:

	userid	age	dob_day	dob_year	dob_month	tenure	friend_count	friendship
count	9.900300e+04	99003.000000	99003.000000	99003.000000	99003.000000	99001.000000	99003.000000	99
mean	1.597045e+06	37.280224	14.530408	1975.719776	6.283365	537.887375	196.350787	
std	3.440592e+05	22.589748	9.015606	22.589748	3.529672	457.649874	387.304229	
min	1.000008e+06	13.000000	1.000000	1900.000000	1.000000	0.000000	0.000000	
25%	1.298806e+06	20.000000	7.000000	1963.000000	3.000000	226.000000	31.000000	
50%	1.596148e+06	28.000000	14.000000	1985.000000	6.000000	412.000000	82.000000	
75%	1.895744e+06	50.000000	22.000000	1993.000000	9.000000	675.000000	206.000000	
max	2.193542e+06	113.000000	31.000000	2000.000000	12.000000	3139.000000	4923.000000	4

We display first few tuples of the data.

In [6]:
```
data.head()
```

Out[6]:

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

We find the Correlation between different features

In [6]:
```
data.corr()
```

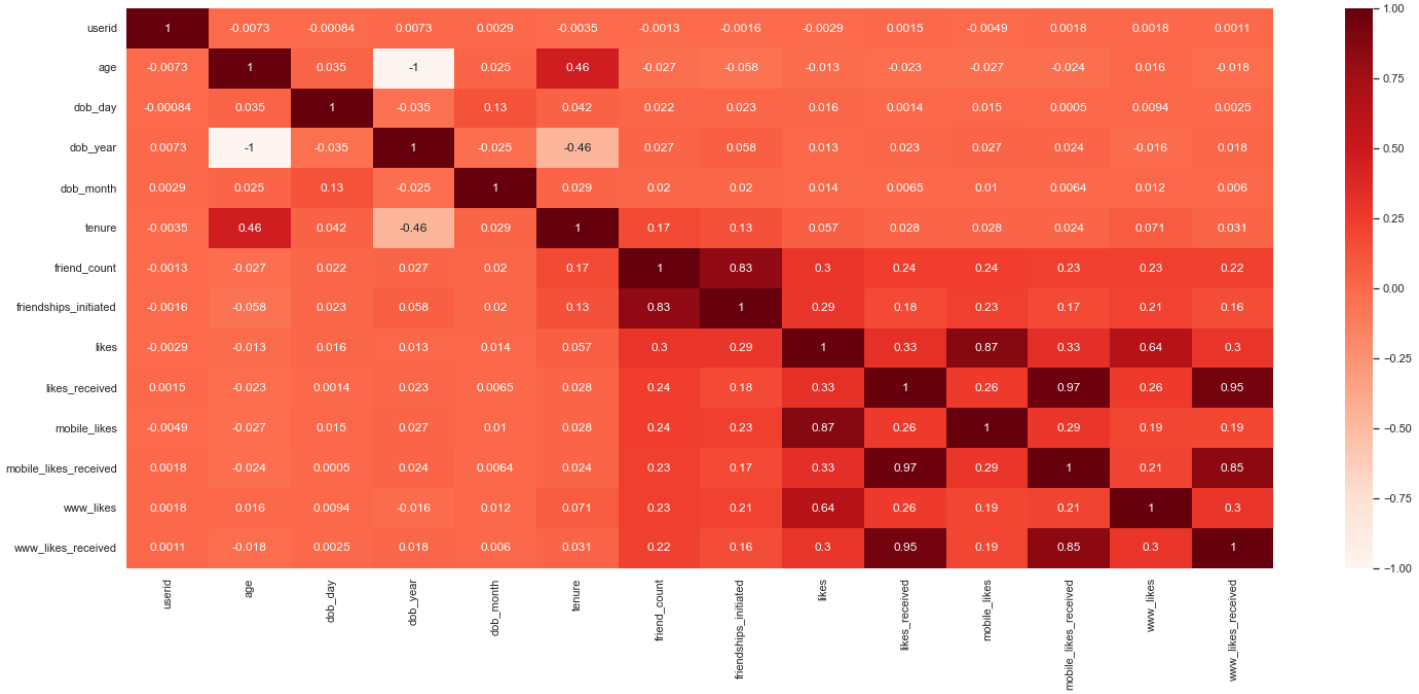
Out[6]:

	userid	age	dob_day	dob_year	dob_month	tenure	friend_count	friendships_init
userid	1.000000	-0.007265	-0.000839	0.007265	0.002924	-0.003446	-0.001314	-0.0
age	-0.007265	1.000000	0.035035	-1.000000	0.025167	0.462742	-0.027407	-0.0
dob_day	-0.000839	0.035035	1.000000	-0.035035	0.129443	0.041855	0.021961	0.0
dob_year	0.007265	-1.000000	-0.035035	1.000000	-0.025167	-0.462742	0.027407	0.0
dob_month	0.002924	0.025167	0.129443	-0.025167	1.000000	0.029446	0.019804	0.0
tenure	-0.003446	0.462742	0.041855	-0.462742	0.029446	1.000000	0.166256	0.1
friend_count	-0.001314	-0.027407	0.021961	0.027407	0.019804	0.166256	1.000000	0.8
friendships_initiated	-0.001591	-0.058059	0.022999	0.058059	0.020075	0.133505	0.825850	1.0
likes	-0.002875	-0.013009	0.015980	0.013009	0.014147	0.057132	0.298017	0.2
likes_received	0.001526	-0.022570	0.001367	0.022570	0.006495	0.027745	0.236463	0.1

	userid	age	dob_day	dob_year	dob_month	tenure	friend_count	friendships_init
mobile_likes	-0.004868	-0.026715	0.014541	0.026715	0.010400	0.028052	0.235656	0.2
mobile_likes_received	0.001753	-0.024248	0.000497	0.024248	0.006435	0.023971	0.232701	0.1
www_likes	0.001828	0.015585	0.009353	-0.015585	0.012136	0.070757	0.229803	0.2
www_likes_received	0.001074	-0.018224	0.002460	0.018224	0.006003	0.030553	0.220727	0.1

We form the correlation heat map using Seaborns Library for better understanding.

In [27]:
```
dataplot = sns.heatmap(data.corr(), cmap="Reds", annot=True)
sns.set(rc = {'figure.figsize':(35,20)})
```



We use Pandas Profiling to derive some observations.

In [9]:
```
profile=data.profile_report(title="Pandas Profiling Report")
profile.to_file(output_file="pandas_profiling.html")
```

We do a pre-processing of data.

We analyse the missing values.

In [10]:
```
data.isnull().sum() # Check for missing values in the dataset
```

Out[10]:
```
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
```

Gender column has missing values, so we fill mode of the preset data in place of those values.

In [10]:
```
Gen=data.gender.mode()[0] # Find mode of the data
data.gender.fillna(Gen,inplace=True) # Fill missig values
data.gender.isnull().sum() # Find the records with null values now
```

Out[10]:
```
0
```

We ignore the tenure tuples, as their no is negligible

WE ANALYSE THE AGE

In [12]:
```
#Here, we divide the age into groups of 10, starting from 6.
groups=["6-15","16-25","26-35","36-45","46-55","56-65","66-75","76-85","86-95","96-105","1
data["age_range"]=pd.cut(data.age,bins=np.arange(6,126,10),labels=groups,right=True)
data.head()
```

Out[12]:

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

In [13]:
```
data.age_range.value_counts()
```

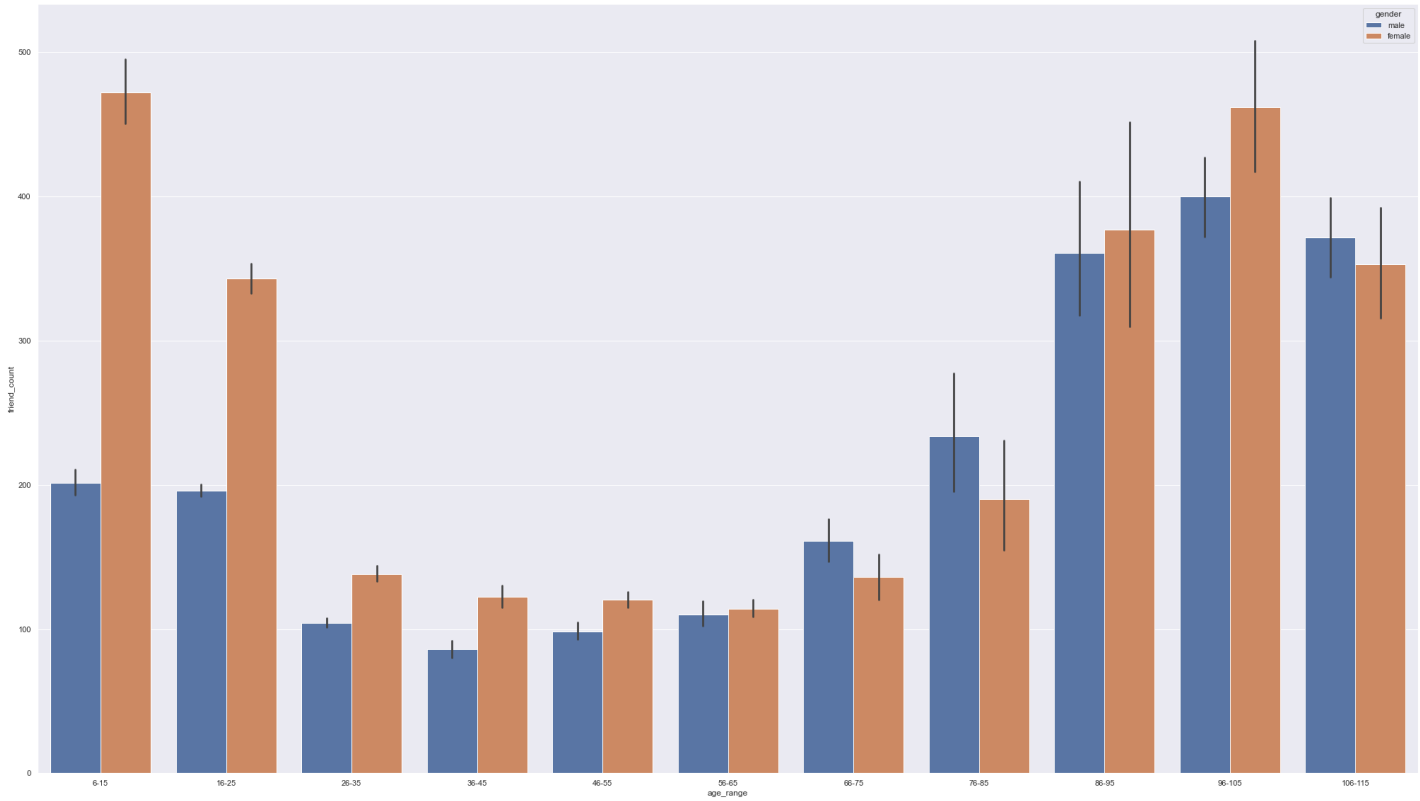
Out[13]:
```
16-25      37029
26-35      16942
46-55       9627
36-45       9058
6-15        8113
56-65       7987
66-75       3945
96-105      2223
```

```
106-115     2021
76-85       1162
86-95        896
Name: age_range, dtype: int64
```

Hence, people in the age group 16-25 have the most users.

We now find which gender and age group has more friends

In [14]:
```
# We compare to find which gender has more friends
sns.barplot(x=data["age_range"],y=data["friend_count"],hue=data.gender)
sns.set(rc = {'figure.figsize':(25,10)})
```



We arrive at the conclusion that females have more friends than males.

Are there any people with no friends at all?

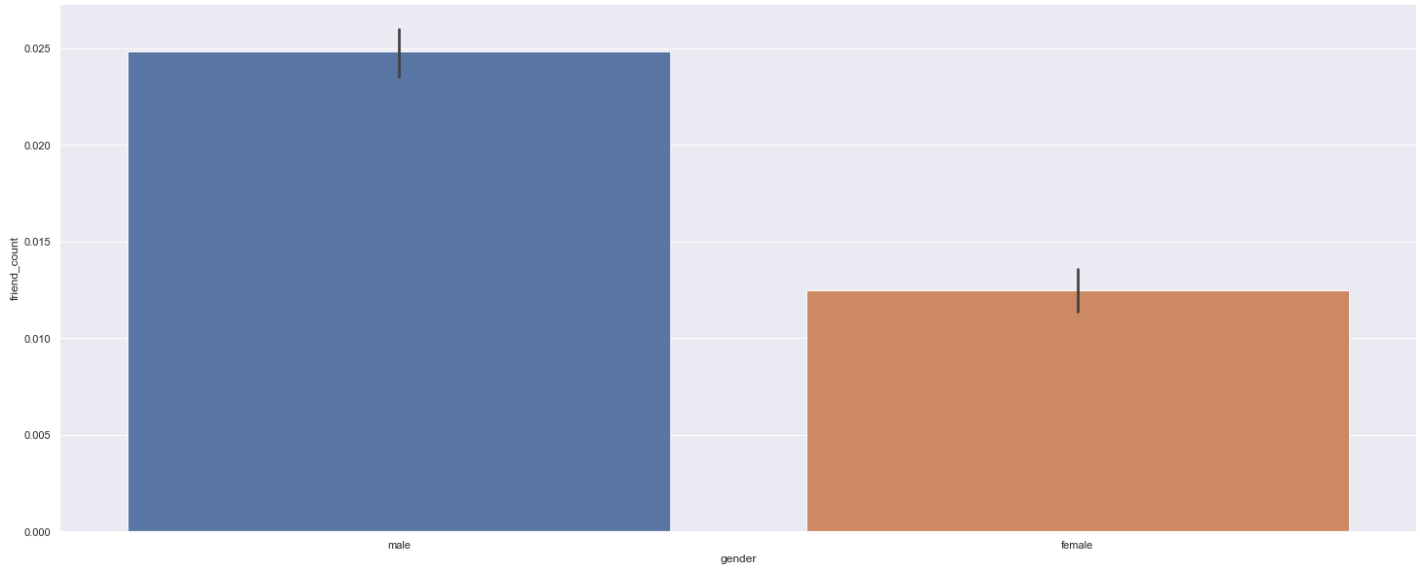
In [15]:
```
x=data.friend_count==0
x.value_counts()
```

Out[15]:
```
False    97041
True      1962
Name: friend_count, dtype: int64
```

1962 people have no friends at all

We now analyse which gender has more 0 friends

In [15]:
```
sns.barplot(x=data.gender,y=data.friend_count==0)
sns.set(rc = {'figure.figsize':(25,10)})
```



Hence, more males have 0 friends, than females.

We find people with most likes recieved.

(Top 25)

In [17]:
```
data.sort_values(by='likes_received',ascending=False)[:25]
```

Out[17]:

	userid	age	dob_day	dob_year	dob_month	gender	tenure	friend_count	friendships_initiated	likes	li
94906	1674584	17	14	1996	8	female	401.0	818	395	1016	
77121	1441676	20	5	1993	8	female	253.0	230	73	2078	
98822	1715925	23	4	1990	9	female	705.0	4077	793	1877	
98994	2063006	20	4	1993	1	female	402.0	1988	332	7351	
98878	1053087	23	6	1990	6	male	596.0	4320	836	2996	
49230	1432020	20	12	1993	1	male	245.0	79	50	477	
98773	2042824	18	25	1995	1	male	51.0	4817	32	1346	
98937	1559908	20	4	1993	12	female	1334.0	4622	1819	4280	
98936	1781243	17	1	1996	5	female	976.0	3683	755	10478	
98686	1015907	74	27	1939	11	male	832.0	4630	831	966	
98939	1914977	20	15	1993	1	female	712.0	3131	935	11142	
97990	1554413	18	19	1995	3	female	1075.0	1848	850	12172	
98973	1836366	26	3	1987	8	female	1669.0	4240	857	4794	

	userid	age	dob_day	dob_year	dob_month	gender	tenure	friend_count	friendships_initiated	likes	li
97694	1314104	19	7	1994	4	female	1552.0	1401	319	714	
98938	1298331	19	7	1994	9	female	1059.0	3776	1068	6912	
97942	1568305	28	14	1985	6	female	484.0	1539	954	2914	
98941	1771772	23	4	1990	2	female	1179.0	3840	1830	9734	
98968	2185422	20	18	1993	8	female	1486.0	4132	1780	3781	
96999	1267229	24	14	1989	9	female	484.0	1171	695	14799	
97958	1386285	18	15	1995	9	female	443.0	1567	663	5620	
98828	1535797	16	27	1997	7	female	1251.0	2768	1042	2053	
98975	1749991	107	29	1906	12	male	328.0	3748	396	6203	
98959	1292537	16	19	1997	1	female	631.0	3763	498	6369	
97991	2126022	39	29	1974	11	male	947.0	1729	589	10115	
98998	1268299	68	4	1945	4	female	541.0	2118	341	3996	

We analyse the gender ratio of the datatset.

In [18]:
```
data["gender"].value_counts()
```

Out[18]:
```
male      58749
female    40254
Name: gender, dtype: int64
```

Now we analyse the tenures(in terms of number of days) of the samples of the dataset.

In [18]:
```
data.tenure.interpolate(inplace=True)
```

In [19]:
```
# Now we create a new column of data, with the range of years a person ha sbeen a part of
temp=["0-1 year","1-2 years","2-3 years","3-4 years","4-5 years","5-6 years","6-7 years",'
data["year_range"]=pd.cut(data.tenure,bins=np.arange(0,3700,365),labels=temp,right=True)
```

In [20]:
```
data.head()
```

Out[20]:

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

In [21]:
```
data.year_range.fillna(value="0-1 year", inplace=True)
```

We filled 0-1 year for people with missing or corrupted values in the tenure column.

We find the tenure of people on the platform, in terms of year range.

In [24]:
```
data.year_range.value_counts(dropna=False)
```

Out[24]:
```
0-1 year      43659
1-2 years     33366
2-3 years      9861
3-4 years      5448
4-5 years      4557
5-6 years      1507
6-7 years       581
7-8 years        15
8-9 years         9
9-10 years        0
Name: year_range, dtype: int64
```

Now we sort people by the no of friendships intitated by them.

In [22]:
```
data.sort_values(by="friendships_initiated",ascending=False)[:25]
```

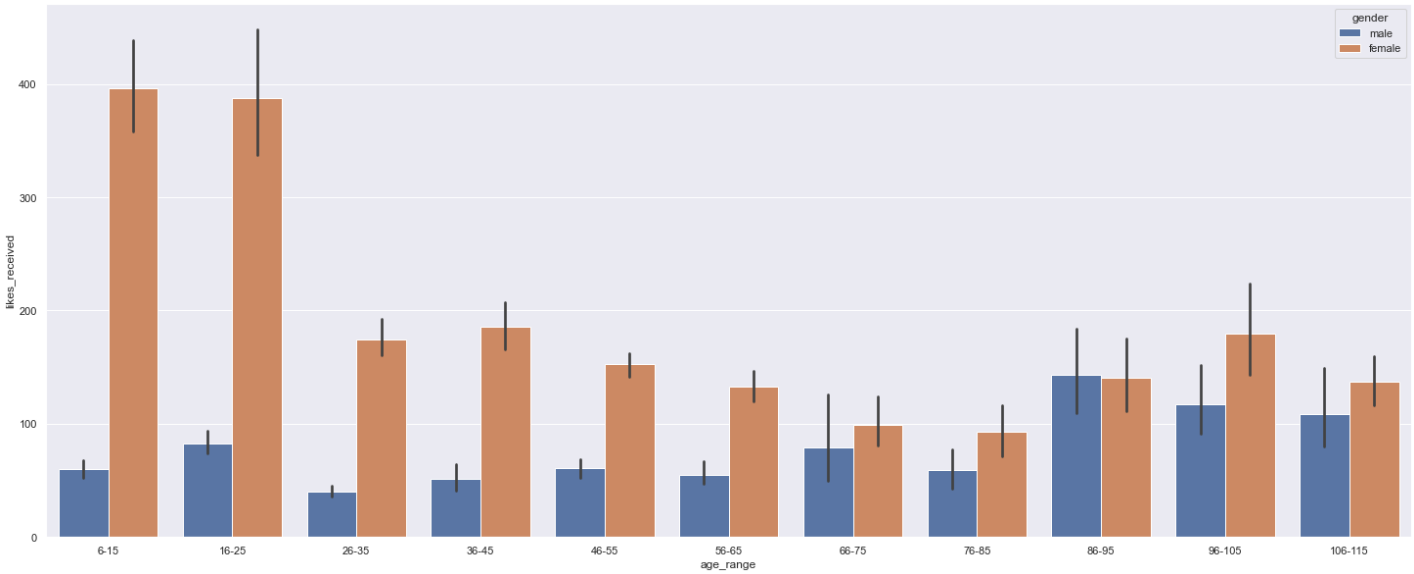
Out[22]:

	userid	age	dob_day	dob_year	dob_month	gender	tenure	friend_count	friendships_initiated	likes	lik
98993	1654565	19	15	1994	8	male	394.0	4538	4144	4501	
98842	1052695	22	23	1991	9	female	874.0	4297	3654	1968	
98675	1949247	19	9	1994	11	female	434.0	4189	3594	927	
98567	1205425	60	17	1953	6	female	1562.0	4794	3538	586	
98347	1403953	19	11	1994	11	male	519.0	3693	3415	170	
98960	1745067	17	1	1996	1	female	947.0	4290	3238	3780	
98898	2010847	18	10	1995	2	female	1084.0	4509	3233	2672	
98949	1103175	15	24	1998	8	female	487.0	3661	3086	6815	
98685	1934087	19	19	1994	5	male	575.0	4516	3078	954	
98835	1075221	22	23	1991	5	male	907.0	4693	3024	2028	
98129	1235124	97	4	1916	6	male	1355.0	4112	2868	16	
98437	1667033	21	8	1992	7	male	1127.0	3450	2837	327	
98636	1405803	25	11	1988	8	male	667.0	3820	2836	835	
98056	1531202	23	17	1990	5	male	917.0	3328	2817	2	
98399	2141603	20	11	1993	5	male	517.0	2983	2772	253	

	userid	age	dob_day	dob_year	dob_month	gender	tenure	friend_count	friendships_initiated	likes	lik
98771	2177920	21	17	1992	9	female	968.0	3279	2693	1402	
98075	2154556	100	19	1913	3	male	2473.0	4765	2668	5	
98375	1112584	27	11	1986	11	male	1349.0	2936	2637	222	
98532	1114537	17	29	1996	8	male	1088.0	4524	2610	505	
98946	1269167	31	30	1982	4	male	513.0	3696	2579	6448	
98887	1090279	17	17	1996	6	female	1110.0	4564	2565	2711	
98672	1256914	15	27	1998	10	female	1189.0	3438	2559	951	
98892	1401489	18	22	1995	3	female	404.0	3769	2550	3162	
98706	1086856	100	23	1913	8	male	1189.0	4632	2548	1064	
98782	1078916	20	13	1993	1	male	959.0	4499	2501	1720	

Now we compare which gender and age group recieves more likes on Facebook.

In [26]:
```
sns.barplot(x=data["age_range"],y=data["likes_received"],hue=data.gender)
sns.set(rc = {'figure.figsize':(25,10)})
```



Hence, Females recieve way more likes on Facebook as compared to Males, with the highest diffence occuring in the adolescents and young adults age groups.

We make a pivot table

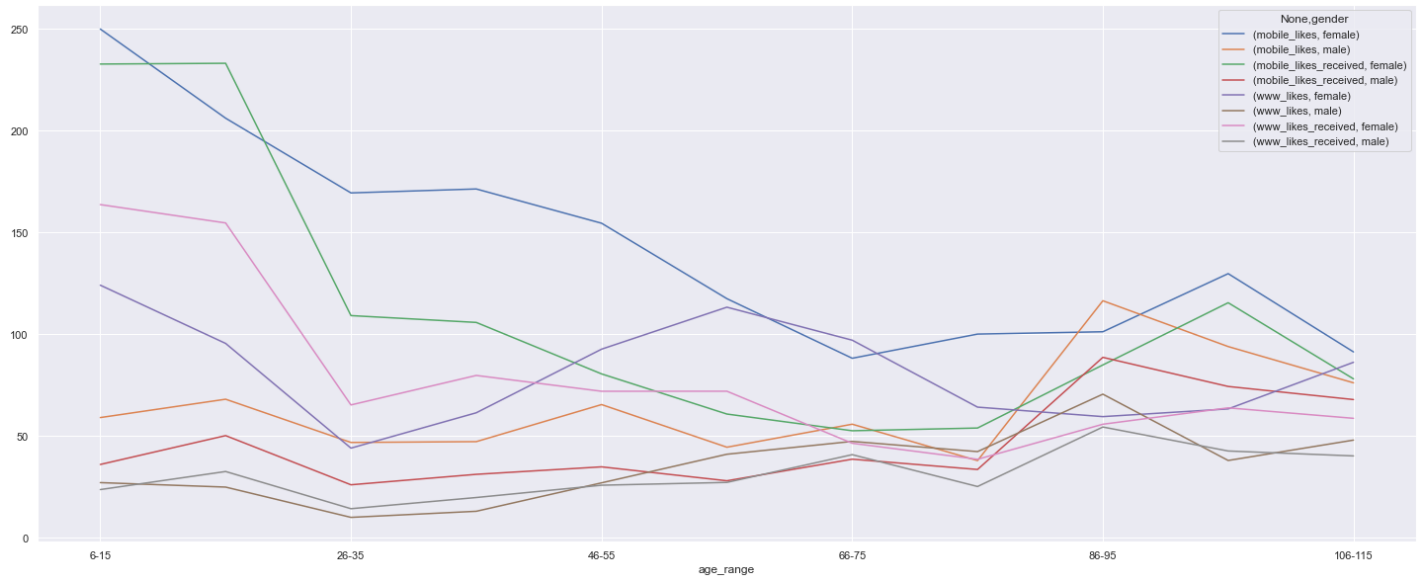
In [27]:
```
data.pivot_table(values=["mobile_likes_received","mobile_likes","www_likes_received","www_
```

Out[27]:

	mobile_likes		mobile_likes_received		www_likes		www_likes_received	
gender	female	male	female	male	female	male	female	male
age_range								
6-15	249.918349	59.077087	232.715540	36.040247	124.100673	27.122232	163.683641	23.702726
16-25	206.127241	68.084572	233.093114	50.215576	95.496685	24.924269	154.666190	32.581393
26-35	169.426254	46.852582	109.162556	26.055591	44.066380	10.056233	65.281079	14.313549
36-45	171.362246	47.253250	105.806505	31.194470	61.428691	13.047427	79.769530	19.803516
46-55	154.530123	65.457973	80.521107	34.868970	92.679713	27.014746	71.991803	25.846851
56-65	117.478201	44.514191	60.806104	28.026454	113.291189	41.064481	72.014915	27.225958
66-75	88.210010	55.820661	52.602552	38.634504	97.083415	47.369691	46.457802	40.858941
76-85	100.081731	37.934944	53.943910	33.574349	64.198718	42.356877	38.581731	25.208178
86-95	101.228650	116.454034	84.925620	88.662289	59.542700	70.568480	55.804408	54.440901
96-105	129.775744	93.956264	115.485126	74.355078	63.327231	37.995552	63.845538	42.658265
106-115	91.331218	76.141930	78.086294	67.922952	86.228426	48.008921	58.710660	40.236821

We plot it vs Age Range

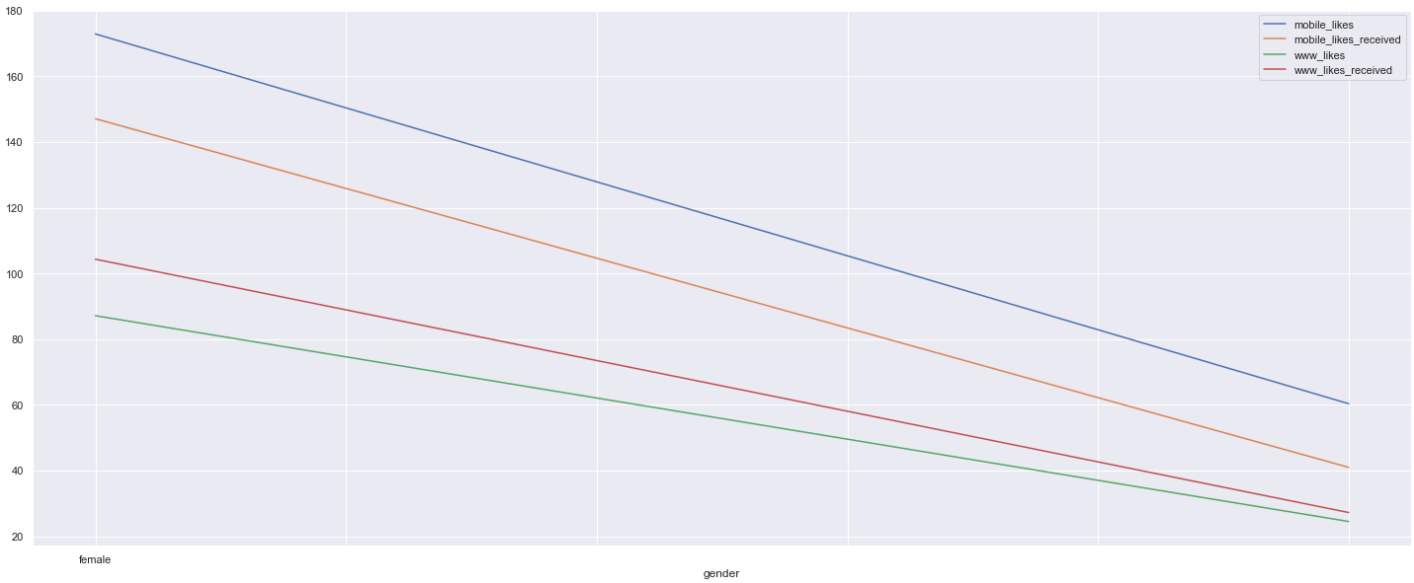
In [28]:
```
data.pivot_table(values=["mobile_likes_received","mobile_likes","www_likes_received","www_
sns.set(rc = {'figure.figsize':(25,10)})
```



We can't derive any reasonable info here, except that all the curves have a similar shape.

We plot it vs Gender

In [29]:
```
data.pivot_table(values=['mobile_likes_received','mobile_likes','www_likes_received','www_
sns.set(rc = {'figure.figsize':(25,10)})
```



All the parameters have a higher value for females than for males.

We find which people are most interested in sending friend requests

In [24]:
```
data.sort_values(by="friendships_initiated",ascending=False)[:10]
```

Out[24]:

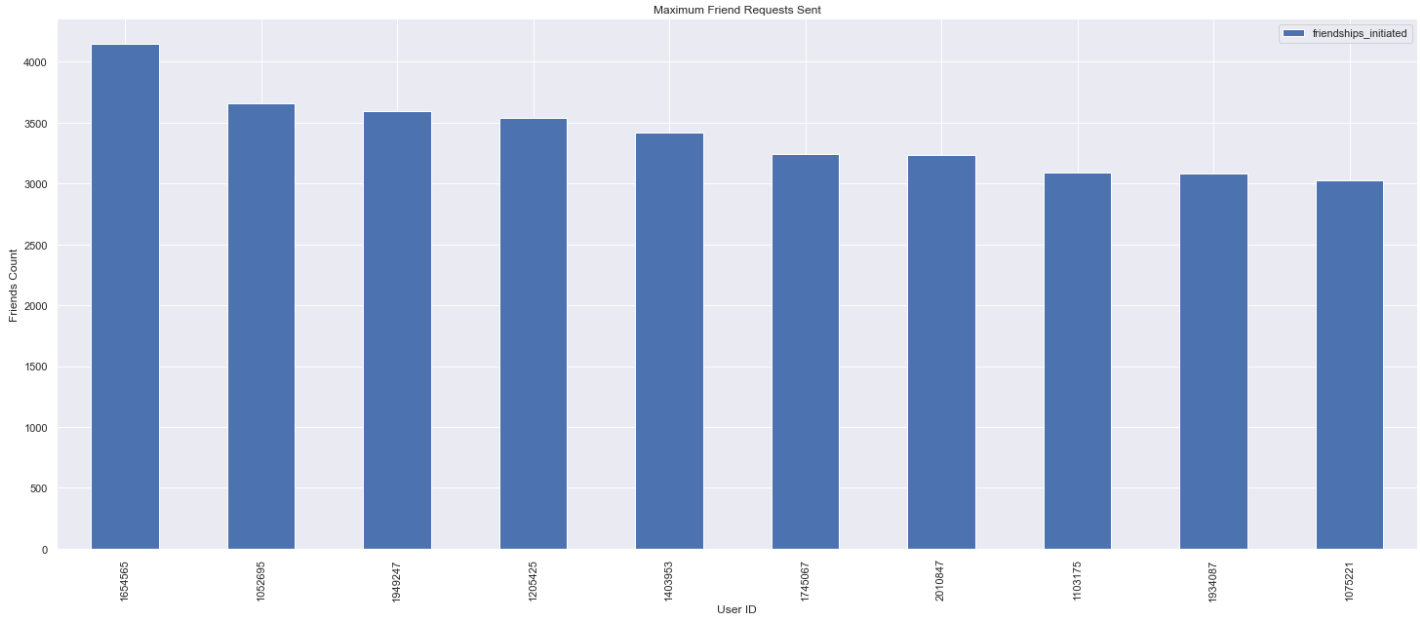
	userid	age	dob_day	dob_year	dob_month	gender	tenure	friend_count	friendships_initiated	likes	lik
98993	1654565	19	15	1994	8	male	394.0	4538	4144	4501	
98842	1052695	22	23	1991	9	female	874.0	4297	3654	1968	
98675	1949247	19	9	1994	11	female	434.0	4189	3594	927	
98567	1205425	60	17	1953	6	female	1562.0	4794	3538	586	
98347	1403953	19	11	1994	11	male	519.0	3693	3415	170	
98960	1745067	17	1	1996	1	female	947.0	4290	3238	3780	
98898	2010847	18	10	1995	2	female	1084.0	4509	3233	2672	
98949	1103175	15	24	1998	8	female	487.0	3661	3086	6815	
98685	1934087	19	19	1994	5	male	575.0	4516	3078	954	
98835	1075221	22	23	1991	5	male	907.0	4693	3024	2028	

In [25]:
```
tempa=data.sort_values(by="friendships_initiated",ascending=False)[:10]
```

We try to plot it by a bar graph for better visualisation.

In [26]:
```
tempa.plot(x="userid",y="friendships_initiated",kind="bar")
plt.xlabel("User ID")
plt.ylabel("Friends Count")
plt.title("Maximum Friend Requests Sent")
```

```
plt.show()
sns.set(rc = {'figure.figsize':(25,10)})
```



Submitted by Vyom Kaushik, 2020UCD2106, CSDS, 5th Semester