In [1]:

```
import numpy as np # linear algebra
from numpy import *
import numpy
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import os
import sys
import ast
from sklearn.metrics import accuracy_score,roc_auc_score
from sklearn  import metrics

#from catboost import Pool
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

import catboost
from catboost import CatBoostClassifier
print(catboost.__version__)

import seaborn as sns
import statsmodels.formula.api as smfrmla
import statsmodels.api as sm
import warnings
```

```
0.26
```

Data reading, informatin checking and visualization

In [2]:

```
tokens_df = pd.read_csv('C:/Users/nasrin/Desktop/AGE PREDICTION/archive/bundles_desc_tokens
tokens_df.info()
tokens_df.head()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 293392 entries, 492765 to 80282
Data columns (total 3 columns):
 #   Column  Non-Null Count   Dtype 
---  ------  --------------   ----- 
 0   tokens  293392 non-null  object
 1   genre   293392 non-null  object
 2   genres  293392 non-null  object
dtypes: object(3)
memory usage: 9.0+ MB
```

Out[2]:

id	tokens	genre	genres
492765	['king', 'deliveri', 'game', 'awad', 'abu', 's...	Games	['Games', 'Racing', 'Casual', 'Entertainment']
687458	['guid', 'jurass', 'winner', 'world', 'tip', '...	Books & Reference	[]
876577	['car', 'photo', 'frame', 'car', 'photo', 'fra...	Photography	[]
1405997	['short', 'tale', 'toy', 'size', 'room', 'esca...	Puzzle	[]
64074	['super', 'hero', 'citi', 'rescu', 'crime', 'f...	Games	['Games', 'Role Playing', 'Action']

In [3]:

```
desc_df = pd.read_csv('C:/Users/nasrin/Desktop/AGE PREDICTION/archive/bundles_desc.csv', in
desc_df.info()
desc_df.head()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 293392 entries, 492765 to 80282
Data columns (total 1 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   description  293392 non-null  object
dtypes: object(1)
memory usage: 4.5+ MB
```

Out[3]:

id	description
492765	لعبة سباق سيارات لت\nملك التوصيل - عوض أبو شفة...
687458	Kiat untuk panduan dan trik Lego Jurassic Worl...
876577	Car photo editor-photo frames is an elite pho...
1405997	A Short Tale is a first person point and click...
64074	This crime fighter rescue is all about a super...

In [4]:

```
prop_df = pd.read_csv('C:/Users/nasrin/Desktop/AGE PREDICTION/archive/bundles_prop.csv', in
prop_df.head()
```

Out[4]:

id	store_os	bundle_released_at	bundle_updated_at	updated_at
492765	ios	2014-06-18 07:00:00.000	2016-07-13T12:03:29.000+00:00	2021-01-14 06:41:41.706
687458	android	2020-02-15 00:00:00.000	2020-02-16T07:20:30.000+00:00	2021-01-14 06:41:46.056
876577	android	2017-12-13 00:00:00.000	2020-12-18T04:19:20.000+00:00	2021-01-14 11:35:25.963
1405997	android	2016-02-10 00:00:00.000	2020-08-14T15:05:48.000+00:00	2021-01-14 19:28:52.177
64074	ios	2017-10-30 14:18:03.000	2020-05-28T19:21:48.000+00:00	2021-01-14 06:41:51.201

In [5]:

```
summary_df = pd.read_csv('C:/Users/nasrin/Desktop/AGE PREDICTION/archive/bundles_summary.cs
summary_df.info()
summary_df.head()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 293392 entries, 492765 to 80282
Data columns (total 6 columns):
 #   Column     Non-Null Count   Dtype  
---  ------     --------------   -----  
 0   summary    232372 non-null  object 
 1   rating     293392 non-null  float64
 2   reviews    293392 non-null  int64  
 3   score      293392 non-null  float64
 4   languages  293392 non-null  object 
 5   is_free    293392 non-null  bool   
dtypes: bool(1), float64(2), int64(1), object(2)
memory usage: 13.7+ MB
```

Out[5]:

id	summary	rating	reviews	score	languages	is_free
492765	NaN	0.0	2707	4.48356	['EN']	False
687458	Tips for guides and tricks for Lego Jurassic W...	407.0	280	2.79000	[]	False
876577	Car Photo Editor-Photo Frames with Background...	161.0	49	3.66000	[]	False
1405997	Decipher clues, solve puzzles, and escape from...	171.0	80	4.71000	[]	True
64074	NaN	0.0	32	4.03125	['EN']	False

In [6]:

```
bundles_gender= pd.read_csv('C:/Users/nasrin/Desktop/AGE PREDICTION/archive/bundles_gender.
df = tokens_df.join(desc_df).join(prop_df).join(summary_df).join(bundles_gender)
df.info()
df.head()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 293392 entries, 492765 to 80282
Data columns (total 17 columns):
 #   Column              Non-Null Count   Dtype  
---  ------              --------------   -----  
 0   tokens              293392 non-null  object 
 1   genre               293392 non-null  object 
 2   genres              293392 non-null  object 
 3   description         293392 non-null  object 
 4   store_os            293392 non-null  object 
 5   bundle_released_at  285312 non-null  object 
 6   bundle_updated_at   293392 non-null  object 
 7   updated_at          293392 non-null  object 
 8   summary             232372 non-null  object 
 9   rating              293392 non-null  float64
 10  reviews             293392 non-null  int64  
 11  score               293392 non-null  float64
 12  languages           293392 non-null  object 
 13  is_free             293392 non-null  bool   
 14  cnt                 171109 non-null  float64
 15  M                   171109 non-null  float64
 16  F                   171109 non-null  float64
dtypes: bool(1), float64(5), int64(1), object(10)
memory usage: 48.3+ MB
```

Out[6]:

id	tokens	genre	genres	description	store_os	bundle_released_at	
492765	['king', 'deliveri', 'game', 'awad', 'abu', 's...	Games	['Games', 'Racing', 'Casual', 'Entertainment']	ملك التوصيل - عوض أبو لعبة سباق\nشفة سيارات لت...	ios	2014-06-18 07:00:00.000	1
687458	['guid', 'jurass', 'winner', 'world', 'tip', '...	Books & Reference	[]	Kiat untuk panduan dan trik Lego Jurassic Worl...	android	2020-02-15 00:00:00.000	1
876577	['car', 'photo', 'frame', 'car', 'photo', 'fra...	Photography	[]	Car photo editor-photo frames is an elite pho...	android	2017-12-13 00:00:00.000	1

id	tokens	genre	genres	description	store_os	bundle_released_at	
1405997	['short', 'tale', 'toy', 'size', 'room', 'esca...	Puzzle	[]	A Short Tale is a first person point and click...	android	2016-02-10 00:00:00.000	1
64074	['super', 'hero', 'citi', 'rescu', 'crime', 'f...	Games	['Games', 'Role Playing', 'Action']	This crime fighter rescue is all about a super...	ios	2017-10-30 14:18:03.000	2

In [ ]:

```
```

In [7]:

```
df1=df.drop(columns=['tokens','genres','description','bundle_released_at','bundle_updated_a
print(df1.head())
```

```
                  genre store_os  rating  reviews    score  is_free
cnt  \
id
492765            Games      ios     0.0     2707  4.48356    False
NaN
687458   Books & Reference  android   407.0      280  2.79000    False
4.0
876577       Photography  android   161.0       49  3.66000    False
1.0
1405997           Puzzle  android   171.0       80  4.71000     True
NaN
64074             Games      ios     0.0       32  4.03125    False  2
1.0

                M         F
id
492765        NaN       NaN
687458   0.000000  1.000000
876577   1.000000  0.000000
1405997       NaN       NaN
64074    0.571429  0.428571
```

In [8]:

```
print("Shape-" ,df1.shape)
```

```
Shape- (293392, 9)
```

In [9]:

```
print(df1.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 293392 entries, 492765 to 80282
Data columns (total 9 columns):
 #   Column    Non-Null Count   Dtype  
---  ------    --------------   -----  
 0   genre     293392 non-null  object 
 1   store_os  293392 non-null  object 
 2   rating    293392 non-null  float64
 3   reviews   293392 non-null  int64  
 4   score     293392 non-null  float64
 5   is_free   293392 non-null  bool   
 6   cnt       171109 non-null  float64
 7   M         171109 non-null  float64
 8   F         171109 non-null  float64
dtypes: bool(1), float64(5), int64(1), object(2)
memory usage: 30.4+ MB
None
```

In [10]:

```
print(df1.genre.value_counts())
```

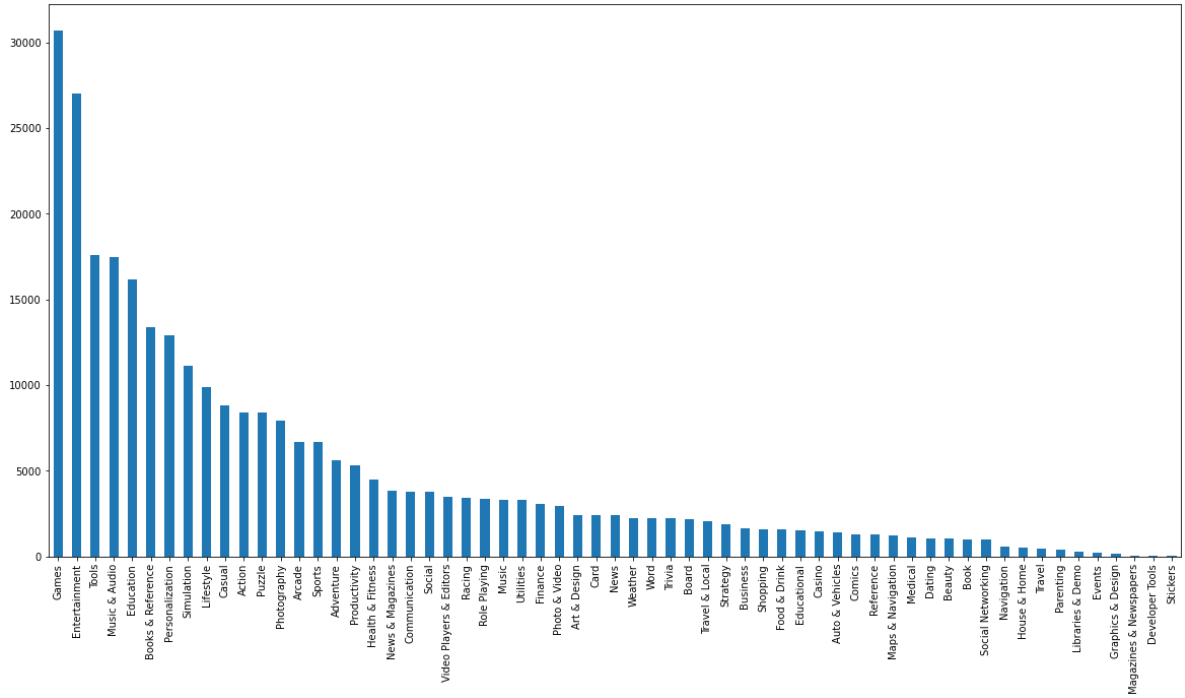
```
Games                    30716
Entertainment            27020
Tools                    17582
Music & Audio            17489
Education                16176
                         ...  
Events                     199
Graphics & Design          126
Magazines & Newspapers       8
Developer Tools              4
Stickers                     4
Name: genre, Length: 61, dtype: int64
```

Bar graph for no of different geners app:

In [11]:

```
df.head()
print(df.genre.value_counts().plot(kind="bar",figsize=(20, 10)))
```

```
AxesSubplot(0.125,0.125;0.775x0.755)
```



Histogram of number of particular gender user for a given range of probability

In [12]:

```
print(bundles_gender)
```

```
         cnt         M         F
id
26550     79  0.430380  0.569620
22488    236  0.525424  0.474576
203745     6  0.500000  0.500000
101327     5  0.200000  0.800000
354773     1  0.000000  1.000000
...      ...       ...       ...
13899    365  0.430137  0.569863
370068    11  0.636364  0.363636
1385065    2  0.000000  1.000000
1096415   29  0.034483  0.965517
295193    48  0.416667  0.583333

[171272 rows x 3 columns]
```

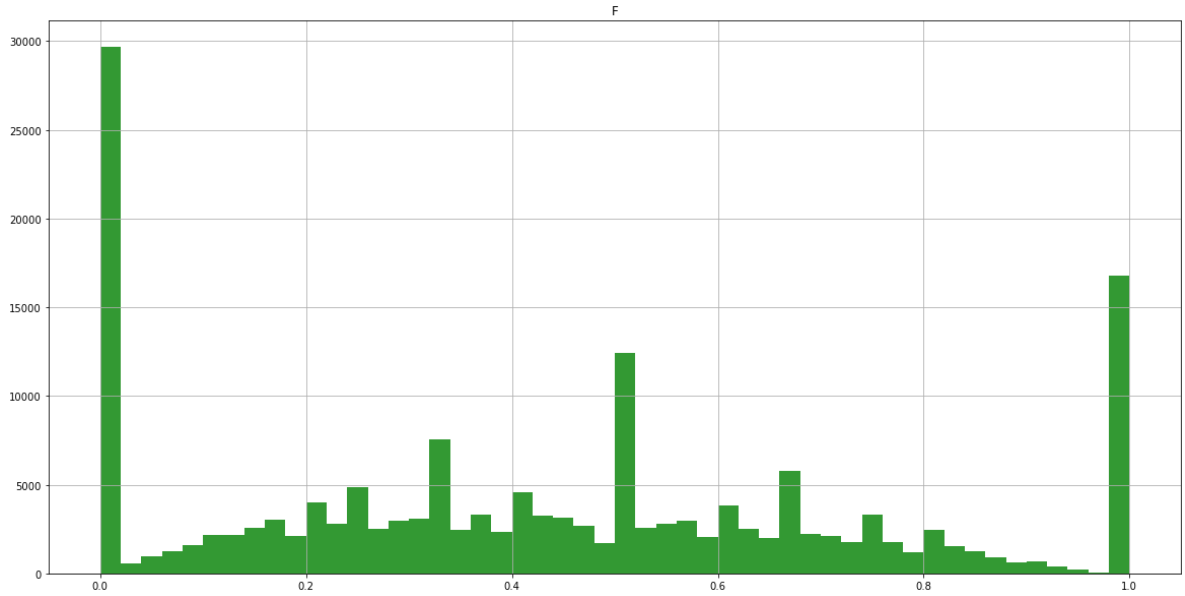
In [13]:

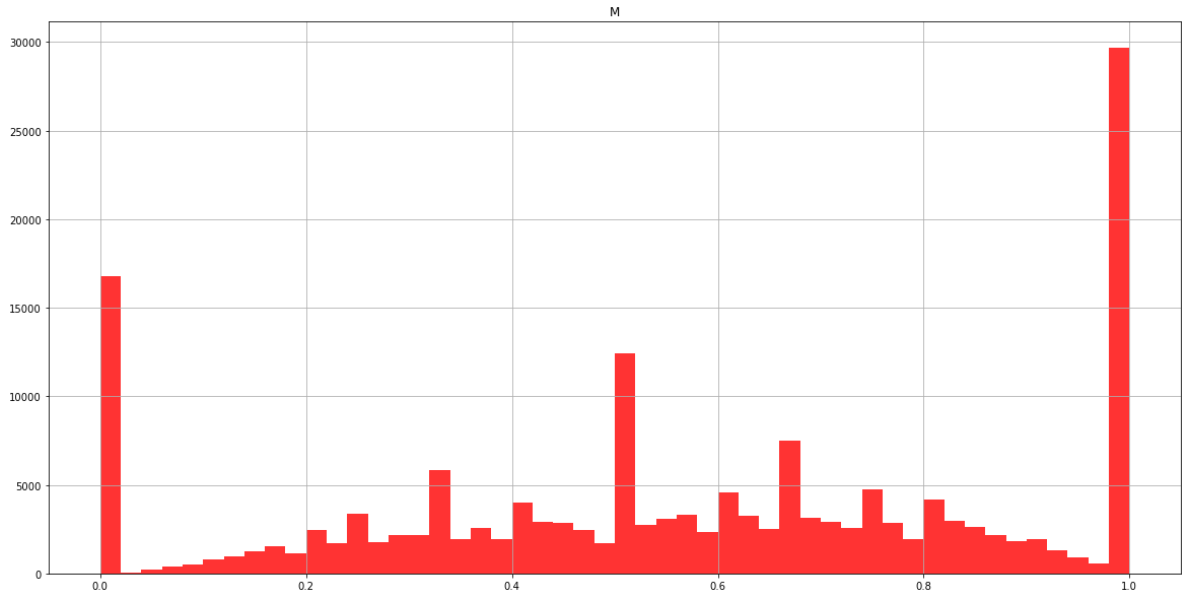
```
bundles_gender.hist(column='F',alpha = .8, color= 'g',bins=50,figsize=(20, 10))

bundles_gender.hist(column='M',alpha = 0.8, color= 'r',bins=50,figsize=(20, 10))
```

Out[13]:

```
array([[<AxesSubplot:title={'center':'M'}>]], dtype=object)
```





Users data reading, informatin checking and visualization

In [14]:

```
users_df = pd.read_csv('C:/Users/nasrin/Desktop/AGE PREDICTION/archive/users.csv', index_co
print(users_df.head())
```

```
                                                                ids gend
uid
1288147374830629126l              [1550,112062,2838,54980,64759,993066]    M
587146916961704617l             [40391,2190,1371978,2023,39200,3516,1634]    M
13595464671590588595  [9728,1314190,979199,2552,1479,1449,976774,131...    M
1265021993296607235l                      [4564,284734,16370,3044,10801]    M
1423826778407581255l                                 [6834,4149,1408540]    M
```

In [15]:

```
print("Shape-" ,users_df.shape,"\n\n\n INFO-" )
print(users_df.info())
```

```
Shape- (627761, 2)


 INFO-
<class 'pandas.core.frame.DataFrame'>
UInt64Index: 627761 entries, 1288147374830629126l to 7574564252383497711
Data columns (total 2 columns):
 #   Column  Non-Null Count   Dtype 
---  ------  --------------   ----- 
 0   ids     627761 non-null  object
 1   gend    627761 non-null  object
dtypes: object(2)
memory usage: 14.4+ MB
None
```

In [16]:

```
i=len(str.split(users_df.iloc[0,0],','))
print(i)
```

```
6
```

In [17]:

```
print(users_df.gend.value_counts())
```

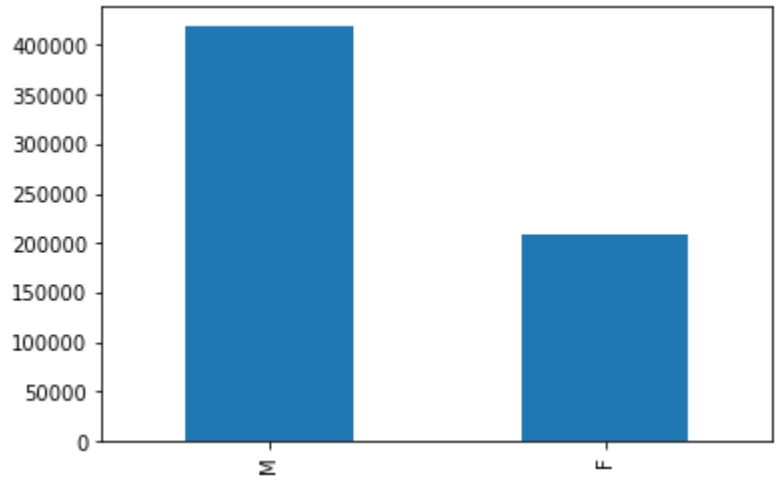
```
M    418845
F    208916
Name: gend, dtype: int64
```

So, This is a unbalanced dataset.

In [18]:

```
print(users_df.gend.value_counts().plot(kind="bar"))
```

```
AxesSubplot(0.125,0.125;0.775x0.755)
```



In [19]:

```
print(users_df.ids.value_counts())
```

```
[1569,2594,5510]
85
[1487,1569,1482]
53
[1468,11166,4149]
39
[1450,1900,1471]
35
[2594,1569,5510]
33

..
[58801,1449,1583,49560,5435,3086,4356,1371991]
1
[8906,1041829,1370225,1408949]
1
[4825,1476,2661,1498,1450,1779,8626,67154,1637,14085,9386,1375247,1372211]
1
[1453,2487,33658,5925]
1
[2371,4356,1450,1533]
1
Name: ids, Length: 619397, dtype: int64
```

In [20]:

```
users_df['apps_count'] = users_df['ids'].str.split(',').apply(len)
users_df.groupby('gend')['apps_count'].describe()
```

Out[20]:

gend	count	mean	std	min	25%	50%	75%	max
F	208916.0	15.166067	21.847228	3.0	5.0	8.0	16.0	685.0
M	418845.0	12.445571	17.637542	3.0	5.0	7.0	13.0	772.0

So, average no of apps used by women more than the male.

In [21]:

```
bundles_gender[(bundles_gender['F']>=0.3325) & (bundles_gender['F']<=0.3375)].describe()
```

Out[21]:

	cnt	M	F
count	6151.000000	6151.000000	6151.000000
mean	171.156235	0.666571	0.333429
std	2760.011403	0.000529	0.000529
min	3.000000	0.662500	0.332512
25%	3.000000	0.666667	0.333333
50%	3.000000	0.666667	0.333333
75%	9.000000	0.666667	0.333333
max	113392.000000	0.667488	0.337500

Using mean of probability of each installed app

#Accuracy and AUC based on probability value given

```
Not working code:
g_dict = bundles_gender['F'].to_dict()
users_df['F_prob'] = users_df['ids'].apply(
    lambda x: np.mean(
        list(filter(None.__ne__, list(map(g_dict.get, x))))
    )
)

Output:RED ALART
    C:\Users\nasrin\anaconda3\lib\site-packages\numpy\core\fromnumeric.py:3372:
RuntimeWarning: Mean of empty slice.
  return _methods._mean(a, axis=axis, dtype=dtype,
```

Type *Markdown* and LaTeX: α^2

In [22]:

```
g_dict = bundles_gender['F'].to_dict()
```

In [23]:

```
def meanprob(a):
    summ =0
    mean=0
    i=0
    count=0
    count2=0
    for i in range (len(a)):
        try :
            summ = summ + g_dict.get(int(str(a[i])))
        except (RuntimeError, TypeError, NameError ,ZeroDivisionError,IndexError ):
            count= count+1
            pass
    try :
        mean = summ / (len(a)-count)
        return mean

    except (RuntimeError, TypeError, NameError ,ZeroDivisionError,IndexError ):
        count2=count2 +1
        return ("None")
        pass

    if count2>0:
        print("No of missing value",count2)
```

In [24]:

```
F_prob=[]
for k in range (0,6):
    p=k*100000
    q=(k+1)*100000

    for j in range (p,q):
        b=list((users_df.iloc[j][0])[1:-1].split(","))
        F_prob.append(meanprob(b))

for j in range (600000,627761):
    b=list((users_df.iloc[j][0])[1:-1].split(","))
    F_prob.append(meanprob(b))
users_df['F_prob'] = np.array(F_prob)
```

In [25]:

```
print(users_df)
```

```
                                                                ids gend
\
uid
1288147374830629126l             [1550,112062,2838,54980,64759,993066]    M
5871496169617046171          [40391,2190,1371978,2023,39200,3516,1634]    M
13595464671590588595  [9728,1314190,979199,2552,1479,1449,976774,131...    M
12650219932966072351                  [4564,284734,16370,3044,10801]    M
14238267784075812558                            [6834,4149,1408540]    M
...                                                             ...   ...
2105809415949843792                        [1528,35250,15078,2439]    F
12814429133630927125                             [17776,1550,6562]    M
13688574997457355644                          [1449,165261,10768]    M
5830984436669696438   [1900,1374218,17248,6841,1860,6787,1412486,147...    F
7574564252383497711   [1511,42869,1833,10183,152191,16036,27304,1077...    M

                     apps_count    F_prob
uid
1288147374830629126l          6  0.238108
5871496169617046171           7  0.392768
13595464671590588595         39  0.581965
12650219932966072351          5  0.435303
14238267784075812558          3  0.345937
...                         ...       ...
2105809415949843792           4  0.614698
12814429133630927125          3  0.141977
13688574997457355644          3  0.186081
5830984436669696438          13  0.755751
7574564252383497711          39  0.244649

[627761 rows x 4 columns]
```

In [26]:

```
print(f"Accuracy: {accuracy_score(users_df['gend'].astype('category').cat.codes, users_df['
```

```
Accuracy: 0.740925288445762
```

In [27]:

```
print(f"AUC: {1 - roc_auc_score(users_df['gend'].astype('category').cat.codes, users_df['F_
```

```
AUC: 0.7793767184317941
```

In [28]:

```
np.corrcoef(users_df['F_prob'],users_df['gend'].astype('category').cat.codes)[0,1]
```

Out[28]:

```
-0.46602945129982837
```

==========

.....

Logistic Regression:

In [ ]:

```

```

In [ ]:

```

```