In [2]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from dateutil import parser
```

In [3]:

```
df = pd.read_csv("FineTech_appData.csv")
```

In [4]:

```
df.shape
```

Out[4]:

```
(50000, 12)
```

In [5]:

```
missing_value = df.isnull().sum()/df.shape[0]*100
```

In [6]:

```
missing_value
```

Out[6]:

```
user                     0.000
first_open               0.000
dayofweek                0.000
hour                     0.000
age                      0.000
screen_list              0.000
numscreens               0.000
minigame                 0.000
used_premium_feature     0.000
enrolled                 0.000
enrolled_date           37.852
liked                    0.000
dtype: float64
```

In [7]:

```
drop_col = missing_value[missing_value>20].keys()
drop_col
```

Out[7]:

```
Index(['enrolled_date'], dtype='object')
```

In [8]:

```
Df_not_null = df.drop('enrolled_date',axis =1)
```

In [9]:

```
Df_not_null
```

Out[9]:

	user	first_open	dayofweek	hour	age	s
0	235136	2012-12-27 02:14:51.273	3	02:00:00	23	idscreen,joinscreen,Cycle,product_revie
1	333588	2012-12-02 01:16:00.905	6	01:00:00	24	joinscreen,product_review,product_revie
2	254414	2013-03-19 19:19:09.157	1	19:00:00	23	Splash,(
3	234192	2013-07-05 16:08:46.354	4	16:00:00	28	product_review,Home,product_review,Loa
4	51549	2013-02-26 18:50:48.661	1	18:00:00	31	idscreen,joinscreen,Cycle,Credit3Conta
...	...	...	...	...	...	
49995	222774	2013-05-09 13:46:17.871	3	13:00:00	32	Splash,Home,ScanPreview,VerifyPhone,Ve
49996	169179	2013-04-09 00:05:17.823	1	00:00:00	35	Cycle,Splash,Home,Reward
49997	302367	2013-02-20 22:41:51.165	2	22:00:00	39	joinscreen,product_review,product_revie
49998	324905	2013-04-28 12:33:04.288	6	12:00:00	27	Cycle,Home,product_review,product_revi
49999	27047	2012-12-14 01:22:44.638	4	01:00:00	25	product_review,ScanPreview,VerifyDate

50000 rows × 11 columns

In [10]:

```
df
```

Out[10]:

	user	first_open	dayofweek	hour	age	s
0	235136	2012-12-27 02:14:51.273	3	02:00:00	23	idscreen,joinscreen,Cycle,product_revie
1	333588	2012-12-02 01:16:00.905	6	01:00:00	24	joinscreen,product_review,product_revie
2	254414	2013-03-19 19:19:09.157	1	19:00:00	23	Splash,(
3	234192	2013-07-05 16:08:46.354	4	16:00:00	28	product_review,Home,product_review,Loa
4	51549	2013-02-26 18:50:48.661	1	18:00:00	31	idscreen,joinscreen,Cycle,Credit3Conta
...	...	...	...	...	...	
49995	222774	2013-05-09 13:46:17.871	3	13:00:00	32	Splash,Home,ScanPreview,VerifyPhone,Ve
49996	169179	2013-04-09 00:05:17.823	1	00:00:00	35	Cycle,Splash,Home,Reward
49997	302367	2013-02-20 22:41:51.165	2	22:00:00	39	joinscreen,product_review,product_revie
49998	324905	2013-04-28 12:33:04.288	6	12:00:00	27	Cycle,Home,product_review,product_revi
49999	27047	2012-12-14 01:22:44.638	4	01:00:00	25	product_review,ScanPreview,VerifyDate

50000 rows × 12 columns

In [11]:

```
df_int2 = df.drop(["user","first_open","screen_list","enrolled_date"],axis=1)
```

In [12]:

```
df_int2
```

Out[12]:

	dayofweek	hour	age	numscreens	minigame	used_premium_feature	enrolled	li
0	3	02:00:00	23	15	0	0	0	
1	6	01:00:00	24	13	0	0	0	
2	1	19:00:00	23	3	0	1	0	
3	4	16:00:00	28	40	0	0	1	
4	1	18:00:00	31	32	0	0	1	
...	...	...	...	...	...	...	...	
49995	3	13:00:00	32	13	0	0	1	
49996	1	00:00:00	35	4	0	1	0	
49997	2	22:00:00	39	25	0	0	0	
49998	6	12:00:00	27	26	0	0	1	
49999	4	01:00:00	25	26	0	0	0	

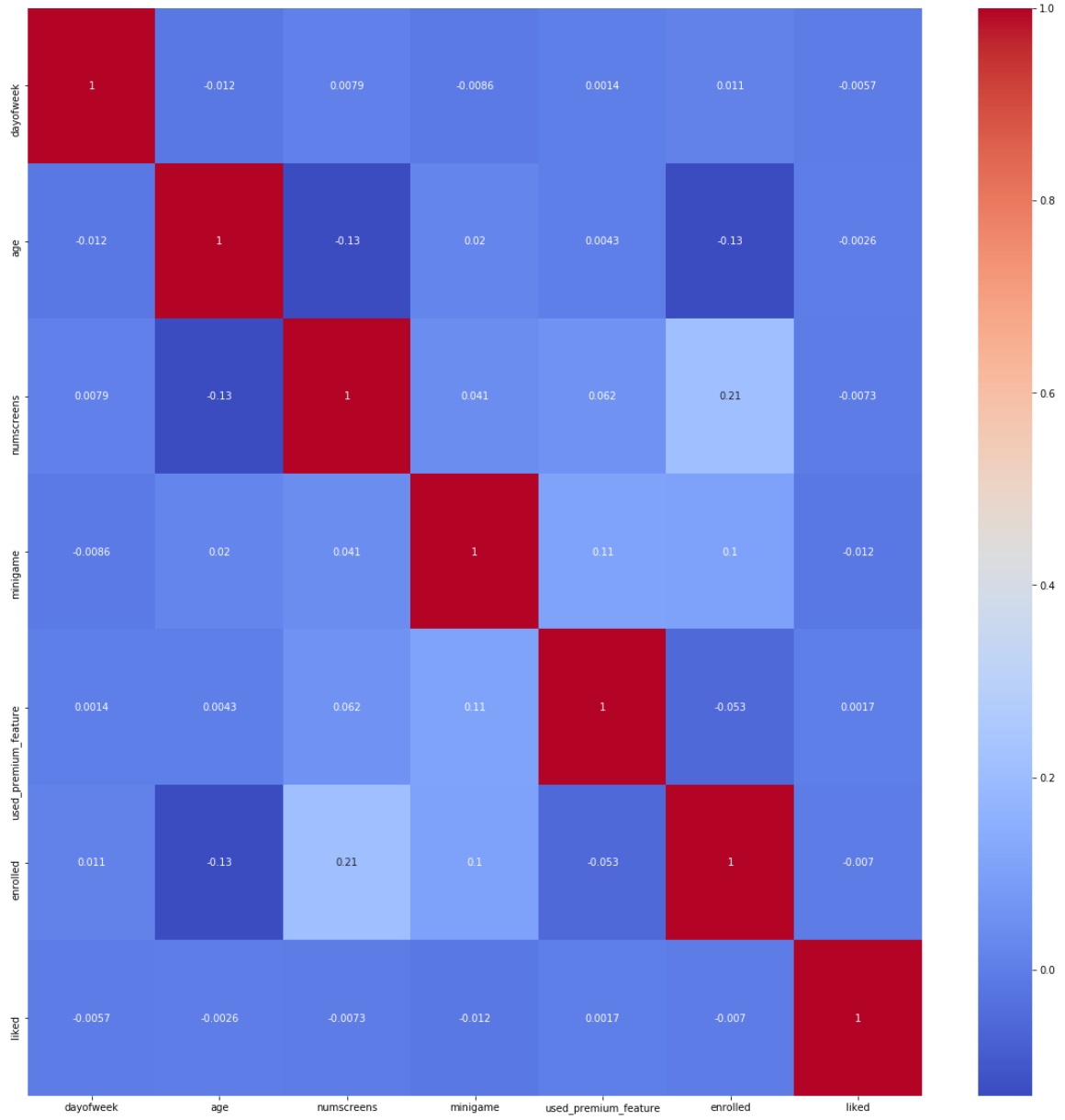
50000 rows × 8 columns

In [13]:

```
plt.figure(figsize=(20,20))
sns.heatmap(df_int2.corr(),annot = True,cmap="coolwarm")
```

Out[13]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x122ff437c48>
```


dayofweek
age
numscreens
minigame
used_premium_feature
enrolled
liked
1 -0.012 0.0079 -0.0086 0.0014 0.011 -0.0057
-0.012 1 -0.13 0.02 0.0043 -0.13 -0.0026
0.0079 -0.13 1 0.041 0.062 0.21 -0.0073
-0.0086 0.02 0.041 1 0.11 0.1 -0.012
0.0014 0.0043 0.062 0.11 1 -0.053 0.0017
0.011 -0.13 0.21 0.1 -0.053 1 -0.007
-0.0057 -0.0026 -0.0073 -0.012 0.0017 -0.007 1
1.0
0.8
0.6
0.4
0.2
0.0

In [14]:

```
sns.pairplot(df_int2, hue = "enrolled")
```

```
---------------------------------------------------------------------------
-
ValueError                                Traceback (most recent call las
t)
~\anaconda3\lib\site-packages\statsmodels\nonparametric\kde.py in kdensity
fft(X, kernel, bw, weights, gridsize, adjust, clip, cut, retgrid)
    450     try:
--> 451         bw = float(bw)
    452     except:

ValueError: could not convert string to float: 'scott'

During handling of the above exception, another exception occurred:

RuntimeError                              Traceback (most recent call las
t)
<ipython-input-14-074c480aa8f6> in <module>
----> 1 sns.pairplot(df_int2, hue = "enrolled")

~\anaconda3\lib\site-packages\seaborn\axisgrid.py in pairplot(data, hue, h
ue_order, palette, vars, x_vars, y_vars, kind, diag_kind, markers, height,
aspect, corner, dropna, plot_kws, diag_kws, grid_kws, size)
   2119             diag_kws.setdefault("shade", True)
   2120             diag_kws["legend"] = False
-> 2121             grid.map_diag(kdeplot, **diag_kws)
   2122
   2123     # Maybe plot on the off-diagonals

~\anaconda3\lib\site-packages\seaborn\axisgrid.py in map_diag(self, func,
 **kwargs)
   1488                     data_k = utils.remove_na(data_k)
   1489
-> 1490                 func(data_k, label=label_k, color=color, **kwargs)
   1491
   1492             self._clean_axis(ax)

~\anaconda3\lib\site-packages\seaborn\distributions.py in kdeplot(data, da
ta2, shade, vertical, kernel, bw, gridsize, cut, clip, legend, cumulative,
shade_lowest, cbar, cbar_ax, cbar_kws, ax, **kwargs)
    703         ax = _univariate_kdeplot(data, shade, vertical, kernel, b
w,
    704                                  gridsize, cut, clip, legend, ax,
--> 705                                  cumulative=cumulative, **kwargs)
    706
    707     return ax

~\anaconda3\lib\site-packages\seaborn\distributions.py in _univariate_kdep
lot(data, shade, vertical, kernel, bw, gridsize, cut, clip, legend, ax, cu
mulative, **kwargs)
    293         x, y = _statsmodels_univariate_kde(data, kernel, bw,
    294                                            gridsize, cut, clip,
--> 295                                            cumulative=cumulative)
    296     else:
    297         # Fall back to scipy if missing statsmodels

~\anaconda3\lib\site-packages\seaborn\distributions.py in _statsmodels_uni
variate_kde(data, kernel, bw, gridsize, cut, clip, cumulative)
    365     fft = kernel == "gau"
    366     kde = smnp.KDEUnivariate(data)
--> 367     kde.fit(kernel, bw, fft, gridsize=gridsize, cut=cut, clip=clip
)
```

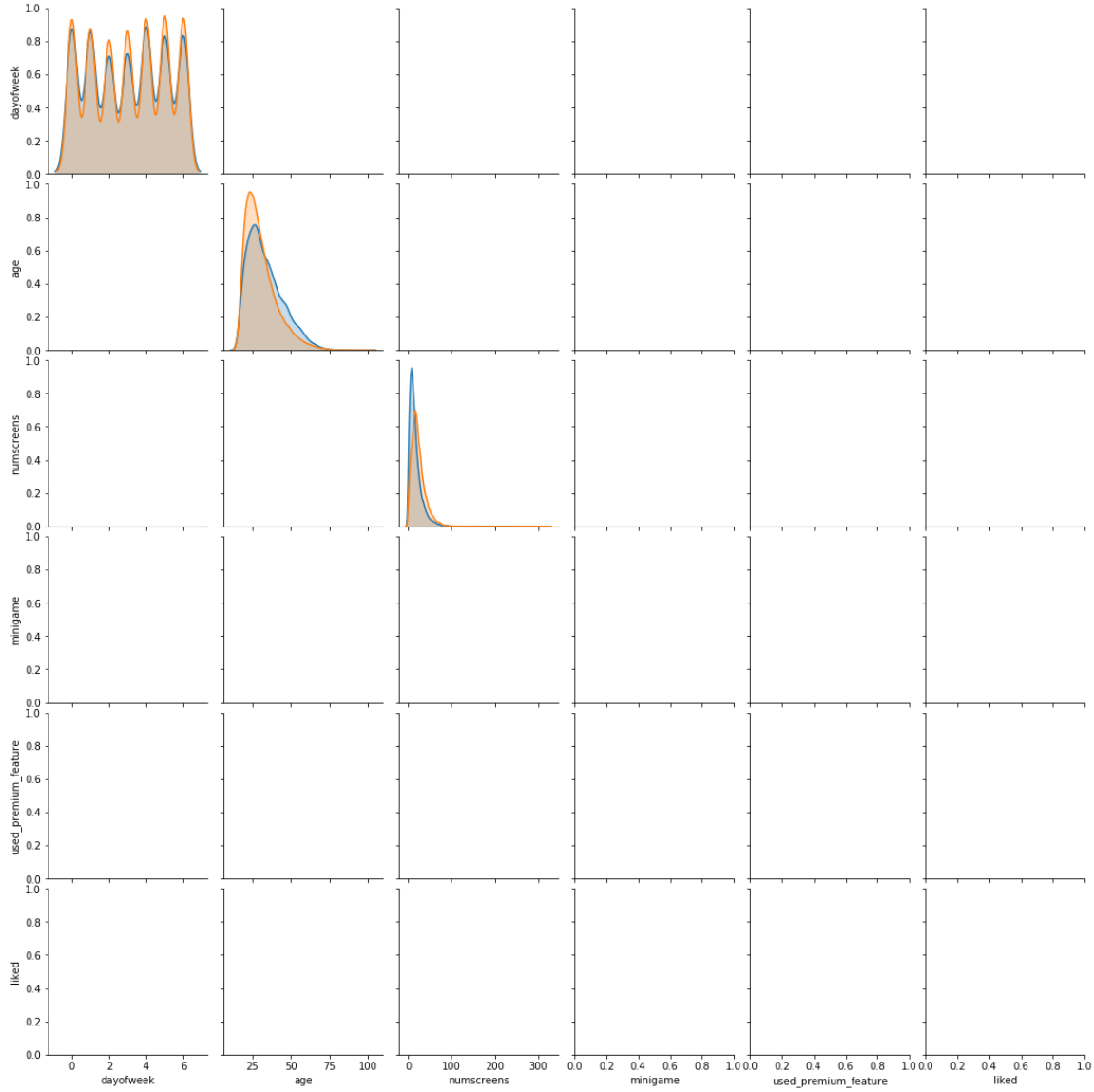
```
    368     if cumulative:
    369         grid, y = kde.support, kde.cdf

~\anaconda3\lib\site-packages\statsmodels\nonparametric\kde.py in fit(sel
f, kernel, bw, fft, weights, gridsize, adjust, cut, clip)
    138             density, grid, bw = kdensityfft(endog, kernel=kernel,
 bw=bw,
    139                     adjust=adjust, weights=weights, gridsize=grids
ize,
--> 140                     clip=clip, cut=cut)
    141         else:
    142             density, grid, bw = kdensity(endog, kernel=kernel, bw=
bw,

~\anaconda3\lib\site-packages\statsmodels\nonparametric\kde.py in kdensity
fft(X, kernel, bw, weights, gridsize, adjust, clip, cut, retgrid)
    451         bw = float(bw)
    452     except:
--> 453         bw = bandwidths.select_bandwidth(X, bw, kern) # will cross
-val fit this pattern?
    454     bw *= adjust
    455

~\anaconda3\lib\site-packages\statsmodels\nonparametric\bandwidths.py in s
elect_bandwidth(x, bw, kernel)
    172         # eventually this can fall back on another selection crite
rion.
    173         err = "Selected KDE bandwidth is 0. Cannot estiamte densit
y."
--> 174         raise RuntimeError(err)
    175     else:
    176         return bandwidth

RuntimeError: Selected KDE bandwidth is 0. Cannot estiamte density.
```


dayofweek
age
numscreens
minigame
used_premium_feature
liked
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
25
50
75
100
0
100
200
300

In [ ]:

```
sns.countplot(df_int2.enrolled)
```

In [ ]:

```
(df_int2.enrolled < 1).sum()
```

In [ ]:

```
(df_int2.enrolled == 1).sum()
```

In [ ]:

```
len(df_int2.columns)
```

In [ ]:

```
plt.figure(figsize= (20,20))
features = df_int2.columns
for i,j in enumerate(features):
    plt.subplot(3,3,i+1)
    plt.title("Histogram of {}".format(j),fontsize=15,color="white")

    bin = len(df_int2[j].unique())
    plt.hist(df_int2[j],bins=bin,rwidth=0.8,edgecolor="y")
```

In [ ]:

```
sns.set()
plt.figure(figsize= (16,9))
plt.title("correlation of all features with enrolled",fontsize=20)
df_int_enr = df_int2.drop(['enrolled'],axis =1)
ax =sns.barplot(df_int_enr.columns,df_int_enr.corrwith(df_int2.enrolled))
ax.tick_params(labelsize=15, labelrotation = 20, color ="k")
```

In [ ]:

```
sns.set() # set background dark grid
plt.figure(figsize = (14,5))
plt.title("Correlation all features with 'enrolled' ", fontsize = 20)
fineTech_appData3 = df_int2.drop(['enrolled'], axis = 1) # drop 'enrolled' feature
ax =sns.barplot(fineTech_appData3.columns,fineTech_appData3.corrwith(df_int2.enrolled))
# plot barplot
ax.tick_params(labelsize=15, labelrotation = 20, color ="k") # decorate x &amp;amp; y ticks
font
```

In [ ]:

```
df_int2
```

In [ ]:

```
df['first_open'] =[parser.parse(i) for i in df['first_open']]

df['enrolled_date'] =[parser.parse(i) if isinstance(i, str) else i for i in df['enrolle
d_date']]

df.dtypes
```

In [15]:

```
df.dtypes
```

Out[15]:

```
user                     int64
first_open              object
dayofweek                int64
hour                    object
age                      int64
screen_list             object
numscreens               int64
minigame                 int64
used_premium_feature     int64
enrolled                 int64
enrolled_date           object
liked                    int64
dtype: object
```

In [16]:

```
df["time_enrolled"] = (df.enrolled_date - df.first_open).astype('timedelta64[h]')
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last
t)
~\anaconda3\lib\site-packages\pandas\core\ops\array_ops.py in na_arithmeti
c_op(left, right, op, str_rep)
    148     try:
--> 149         result = expressions.evaluate(op, str_rep, left, right)
    150     except TypeError:

~\anaconda3\lib\site-packages\pandas\core\computation\expressions.py in ev
aluate(op, op_str, a, b, use_numexpr)
    207     if use_numexpr:
--> 208         return _evaluate(op, op_str, a, b)
    209     return _evaluate_standard(op, op_str, a, b)

~\anaconda3\lib\site-packages\pandas\core\computation\expressions.py in _e
valuate_numexpr(op, op_str, a, b)
    120     if result is None:
--> 121         result = _evaluate_standard(op, op_str, a, b)
    122

~\anaconda3\lib\site-packages\pandas\core\computation\expressions.py in _e
valuate_standard(op, op_str, a, b)
     69     with np.errstate(all="ignore"):
---> 70         return op(a, b)
     71

TypeError: unsupported operand type(s) for -: 'float' and 'str'

During handling of the above exception, another exception occurred:

TypeError                                 Traceback (most recent call last
t)
<ipython-input-16-d71322155d6e> in <module>
----> 1 df["time_enrolled"] = (df.enrolled_date - df.first_open).astype('t
imedelta64[h]')

~\anaconda3\lib\site-packages\pandas\core\ops\common.py in new_method(sel
f, other)
     62         other = item_from_zerodim(other)
     63
---> 64         return method(self, other)
     65
     66     return new_method

~\anaconda3\lib\site-packages\pandas\core\ops\__init__.py in wrapper(left,
right)
    498         lvalues = extract_array(left, extract_numpy=True)
    499         rvalues = extract_array(right, extract_numpy=True)
--> 500         result = arithmetic_op(lvalues, rvalues, op, str_rep)
    501
    502         return _construct_result(left, result, index=left.index, n
ame=res_name)

~\anaconda3\lib\site-packages\pandas\core\ops\array_ops.py in arithmetic_o
p(left, right, op, str_rep)
    195     else:
    196         with np.errstate(all="ignore"):
--> 197             res_values = na_arithmetic_op(lvalues, rvalues, op, st
r_rep)
```

```
    198
    199     return res_values

~\anaconda3\lib\site-packages\pandas\core\ops\array_ops.py in na_arithmeti
c_op(left, right, op, str_rep)
    149         result = expressions.evaluate(op, str_rep, left, right)
    150     except TypeError:
--> 151         result = masked_arith_op(left, right, op)
    152
    153     return missing.dispatch_fill_zeros(op, left, right, result)

~\anaconda3\lib\site-packages\pandas\core\ops\array_ops.py in masked_arith
_op(x, y, op)
     92         if mask.any():
     93             with np.errstate(all="ignore"):
---> 94                 result[mask] = op(xrav[mask], yrav[mask])
     95
     96     else:

TypeError: unsupported operand type(s) for -: 'str' and 'str'
```

In [ ]:

```
df["time_enrolled"].isnull().sum()
```

In [17]:

```
plt.figure(figsize=(12,6))
plt.hist(df["time_enrolled"].dropna(),range=(0,100))
```

```
---------------------------------------------------------------------------
-
KeyError                                  Traceback (most recent call las
t)
~\anaconda3\lib\site-packages\pandas\core\indexes\base.py in get_loc(self,
key, method, tolerance)
   2645             try:
-> 2646                 return self._engine.get_loc(key)
   2647             except KeyError:

pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObject
HashTable.get_item()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObject
HashTable.get_item()

KeyError: 'time_enrolled'

During handling of the above exception, another exception occurred:

KeyError                                  Traceback (most recent call las
t)
<ipython-input-17-3158c67c39a8> in <module>
      1 plt.figure(figsize=(12,6))
----> 2 plt.hist(df["time_enrolled"].dropna(),range=(0,100))

~\anaconda3\lib\site-packages\pandas\core\frame.py in __getitem__(self, ke
y)
   2798             if self.columns.nlevels > 1:
   2799                 return self._getitem_multilevel(key)
-> 2800             indexer = self.columns.get_loc(key)
   2801             if is_integer(indexer):
   2802                 indexer = [indexer]

~\anaconda3\lib\site-packages\pandas\core\indexes\base.py in get_loc(self,
key, method, tolerance)
   2646                 return self._engine.get_loc(key)
   2647             except KeyError:
-> 2648                 return self._engine.get_loc(self._maybe_cast_index
er(key))
   2649         indexer = self.get_indexer([key], method=method, tolerance
=tolerance)
   2650         if indexer.ndim > 1 or indexer.size > 1:

pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObject
HashTable.get_item()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObject
HashTable.get_item()

KeyError: 'time_enrolled'
```

```
<Figure size 864x432 with 0 Axes>
```

In [ ]:

```
df['hour'] = df['hour'].str.slice(1,3).astype(int)
```

In [18]:

```
df.dtypes
```

Out[18]:

```
user                     int64
first_open              object
dayofweek                int64
hour                    object
age                      int64
screen_list             object
numscreens               int64
minigame                 int64
used_premium_feature     int64
enrolled                 int64
enrolled_date           object
liked                    int64
dtype: object
```

In [24]:

```
df["screen_list"].nunique()
```

Out[24]:

```
38799
```

In [40]:

```
ts = pd.read_csv("top_screens.csv").top_screens.values
```

In [41]:

```
ts
```

Out[41]:

```
array(['Loan2', 'location', 'Institutions', 'Credit3Container',
       'VerifyPhone', 'BankVerification', 'VerifyDateOfBirth',
       'ProfilePage', 'VerifyCountry', 'Cycle', 'idscreen',
       'Credit3Dashboard', 'Loan3', 'CC1Category', 'Splash', 'Loan',
       'CC1', 'RewardsContainer', 'Credit3', 'Credit1', 'EditProfile',
       'Credit2', 'Finances', 'CC3', 'Saving9', 'Saving1', 'Alerts',
       'Saving8', 'Saving10', 'Leaderboard', 'Saving4', 'VerifyMobile',
       'VerifyHousing', 'RewardDetail', 'VerifyHousingAmount',
       'ProfileMaritalStatus', 'ProfileChildren ', 'ProfileEducation',
       'Saving7', 'ProfileEducationMajor', 'Rewards', 'AccountView',
       'VerifyAnnualIncome', 'VerifyIncomeType', 'Saving2', 'Saving6',
       'Saving2Amount', 'Saving5', 'ProfileJobTitle', 'Login',
       'ProfileEmploymentLength', 'WebView', 'SecurityModal', 'Loan4',
       'ResendToken', 'TransactionList', 'NetworkFailure', 'ListPicker'],
      dtype=object)
```

In [42]:

```
ts
```

Out[42]:

```
array(['Loan2', 'location', 'Institutions', 'Credit3Container',
       'VerifyPhone', 'BankVerification', 'VerifyDateOfBirth',
       'ProfilePage', 'VerifyCountry', 'Cycle', 'idscreen',
       'Credit3Dashboard', 'Loan3', 'CC1Category', 'Splash', 'Loan',
       'CC1', 'RewardsContainer', 'Credit3', 'Credit1', 'EditProfile',
       'Credit2', 'Finances', 'CC3', 'Saving9', 'Saving1', 'Alerts',
       'Saving8', 'Saving10', 'Leaderboard', 'Saving4', 'VerifyMobile',
       'VerifyHousing', 'RewardDetail', 'VerifyHousingAmount',
       'ProfileMaritalStatus', 'ProfileChildren ', 'ProfileEducation',
       'Saving7', 'ProfileEducationMajor', 'Rewards', 'AccountView',
       'VerifyAnnualIncome', 'VerifyIncomeType', 'Saving2', 'Saving6',
       'Saving2Amount', 'Saving5', 'ProfileJobTitle', 'Login',
       'ProfileEmploymentLength', 'WebView', 'SecurityModal', 'Loan4',
       'ResendToken', 'TransactionList', 'NetworkFailure', 'ListPicker'],
      dtype=object)
```

In [43]:

```
df["Screen_list"] = df.screen_list.astype(str)+","
```

In [44]:

```
df["Screen_list"]
```

Out[44]:

```
0        idscreen,joinscreen,Cycle,product_review,ScanP...
1        joinscreen,product_review,product_review2,Scan...
2                                     Splash,Cycle,Loan,
3        product_review,Home,product_review,Loan3,Finan...
4        idscreen,joinscreen,Cycle,Credit3Container,Sca...
                               ...
49995    Splash,Home,ScanPreview,VerifyPhone,VerifySSN,...
49996                  Cycle,Splash,Home,RewardsContainer,
49997    joinscreen,product_review,product_review2,Scan...
49998    Cycle,Home,product_review,product_review,produ...
49999    product_review,ScanPreview,VerifyDateOfBirth,V...
Name: Screen_list, Length: 50000, dtype: object
```

In [45]:

```
for screen_name in ts:
    df[screen_name] = df.screen_list.str.contains(screen_name).astype(int)
    df['screen_list'] = df.screen_list.str.replace(screen_name+",", "")
```

In [58]:

```
df3 =df2.drop(['screen_list'],axis=1)
```

In [59]:

```
df3.shape
```

Out[59]:

```
(50000, 69)
```

In [67]:

```
df3.columns
```

Out[67]:

```
Index(['user', 'first_open', 'dayofweek', 'hour', 'age', 'numscreens',
       'minigame', 'used_premium_feature', 'enrolled', 'enrolled_date',
       'liked', 'Loan2', 'location', 'Institutions', 'Credit3Container',
       'VerifyPhone', 'BankVerification', 'VerifyDateOfBirth', 'ProfilePag
e',
       'VerifyCountry', 'Cycle', 'idscreen', 'Credit3Dashboard', 'Loan3',
       'CC1Category', 'Splash', 'Loan', 'CC1', 'RewardsContainer', 'Credit
3',
       'Credit1', 'EditProfile', 'Credit2', 'Finances', 'CC3', 'Saving9',
       'Saving1', 'Alerts', 'Saving8', 'Saving10', 'Leaderboard', 'Saving
4',
       'VerifyMobile', 'VerifyHousing', 'RewardDetail', 'VerifyHousingAmou
nt',
       'ProfileMaritalStatus', 'ProfileChildren ', 'ProfileEducation',
       'Saving7', 'ProfileEducationMajor', 'Rewards', 'AccountView',
       'VerifyAnnualIncome', 'VerifyIncomeType', 'Saving2', 'Saving6',
       'Saving2Amount', 'Saving5', 'ProfileJobTitle', 'Login',
       'ProfileEmploymentLength', 'WebView', 'SecurityModal', 'Loan4',
       'ResendToken', 'TransactionList', 'NetworkFailure', 'ListPicker'],
      dtype='object')
```

In [68]:

```
saving_screens = ['Saving1',
                  'Saving2',
                  'Saving2Amount',
                  'Saving4',
                  'Saving5',
                  'Saving6',
                  'Saving7',
                  'Saving8',
                  'Saving9',
                  'Saving10',
                 ]
df3['saving_screens_count'] = df3[saving_screens].sum(axis = 1)
df3.drop(columns = saving_screens, inplace = True)
```

In [69]:

```
credit_screens = ['Credit1',
                  'Credit2',
                  'Credit3',
                  'Credit3Container',
                  'Credit3Dashboard',
                 ]
df3['credit_screens_count'] = df3[credit_screens].sum(axis = 1)
df3.drop(columns = credit_screens, axis = 1, inplace = True)
```

In [70]:

```
cc_screens = ['CC1',
              'CC1Category',
              'CC3',
             ]
df3["cc_screen_count"]=df3[cc_screens].sum(axis=1)
df3.drop(columns=cc_screens,axis=1,inplace=True)
```

In [73]:

```
df3.dtypes
```

Out[73]:

```
user                          int64
first_open                   object
dayofweek                     int64
hour                          int32
age                           int64
numscreens                    int64
minigame                      int64
used_premium_feature          int64
enrolled                      int64
enrolled_date                object
liked                         int64
Loan2                         int32
location                      int32
Institutions                  int32
VerifyPhone                   int32
BankVerification              int32
VerifyDateOfBirth             int32
ProfilePage                   int32
VerifyCountry                 int32
Cycle                         int32
idscreen                      int32
Loan3                         int32
Splash                        int32
Loan                          int32
RewardsContainer              int32
EditProfile                   int32
Finances                      int32
Alerts                        int32
Leaderboard                   int32
VerifyMobile                  int32
VerifyHousing                 int32
RewardDetail                  int32
VerifyHousingAmount           int32
ProfileMaritalStatus          int32
ProfileChildren               int32
ProfileEducation              int32
ProfileEducationMajor         int32
Rewards                       int32
AccountView                   int32
VerifyAnnualIncome            int32
VerifyIncomeType              int32
ProfileJobTitle               int32
Login                         int32
ProfileEmploymentLength       int32
WebView                       int32
SecurityModal                 int32
Loan4                         int32
ResendToken                   int32
TransactionList               int32
NetworkFailure                int32
ListPicker                    int32
saving_screens_count          int64
credit_screens_count          int64
cc_screen_count               int64
dtype: object
```

In [72]:

```
df3['hour'] = df3['hour'].str.slice(1,3).astype(int)
```

In [76]:

```
df3['first_open'] =[parser.parse(i) for i in df3['first_open']]

df3['enrolled_date'] =[parser.parse(i) if isinstance(i, str) else i for i in df3['enrol
led_date']]
df3["time_enrolled"] = (df3.enrolled_date - df3.first_open).astype('timedelta64[h]')

df3.dtypes
```

Out[76]:

```
user                              int64
first_open                datetime64[ns]
dayofweek                         int64
hour                              int32
age                               int64
numscreens                        int64
minigame                          int64
used_premium_feature              int64
enrolled                          int64
enrolled_date             datetime64[ns]
liked                             int64
Loan2                             int32
location                          int32
Institutions                      int32
VerifyPhone                       int32
BankVerification                  int32
VerifyDateOfBirth                 int32
ProfilePage                       int32
VerifyCountry                     int32
Cycle                             int32
idscreen                          int32
Loan3                             int32
Splash                            int32
Loan                              int32
RewardsContainer                  int32
EditProfile                       int32
Finances                          int32
Alerts                            int32
Leaderboard                       int32
VerifyMobile                      int32
VerifyHousing                     int32
RewardDetail                      int32
VerifyHousingAmount               int32
ProfileMaritalStatus              int32
ProfileChildren                   int32
ProfileEducation                  int32
ProfileEducationMajor             int32
Rewards                           int32
AccountView                       int32
VerifyAnnualIncome                int32
VerifyIncomeType                  int32
ProfileJobTitle                   int32
Login                             int32
ProfileEmploymentLength           int32
WebView                           int32
SecurityModal                     int32
Loan4                             int32
ResendToken                       int32
TransactionList                   int32
NetworkFailure                    int32
ListPicker                        int32
saving_screens_count              int64
credit_screens_count              int64
cc_screen_count                   int64
time_enrolled                   float64
dtype: object
```

In [77]:

```
df3["time_enrolled"] = (df3.enrolled_date - df3.first_open).astype('timedelta64[h]')
```

In [78]:

```
df3.dtypes
```

Out[78]:

```
user                              int64
first_open                datetime64[ns]
dayofweek                         int64
hour                              int32
age                               int64
numscreens                        int64
minigame                          int64
used_premium_feature              int64
enrolled                          int64
enrolled_date             datetime64[ns]
liked                             int64
Loan2                             int32
location                          int32
Institutions                      int32
VerifyPhone                       int32
BankVerification                  int32
VerifyDateOfBirth                 int32
ProfilePage                       int32
VerifyCountry                     int32
Cycle                             int32
idscreen                          int32
Loan3                             int32
Splash                            int32
Loan                              int32
RewardsContainer                  int32
EditProfile                       int32
Finances                          int32
Alerts                            int32
Leaderboard                       int32
VerifyMobile                      int32
VerifyHousing                     int32
RewardDetail                      int32
VerifyHousingAmount               int32
ProfileMaritalStatus              int32
ProfileChildren                   int32
ProfileEducation                  int32
ProfileEducationMajor             int32
Rewards                           int32
AccountView                       int32
VerifyAnnualIncome                int32
VerifyIncomeType                  int32
ProfileJobTitle                   int32
Login                             int32
ProfileEmploymentLength           int32
WebView                           int32
SecurityModal                     int32
Loan4                             int32
ResendToken                       int32
TransactionList                   int32
NetworkFailure                    int32
ListPicker                        int32
saving_screens_count              int64
credit_screens_count              int64
cc_screen_count                   int64
time_enrolled                   float64
dtype: object
```

In [80]:

```
df3.drop(columns = ['time_enrolled', 'enrolled_date', 'first_open'], inplace=True)
```

In [82]:

```
df3.shape
```

Out[82]:

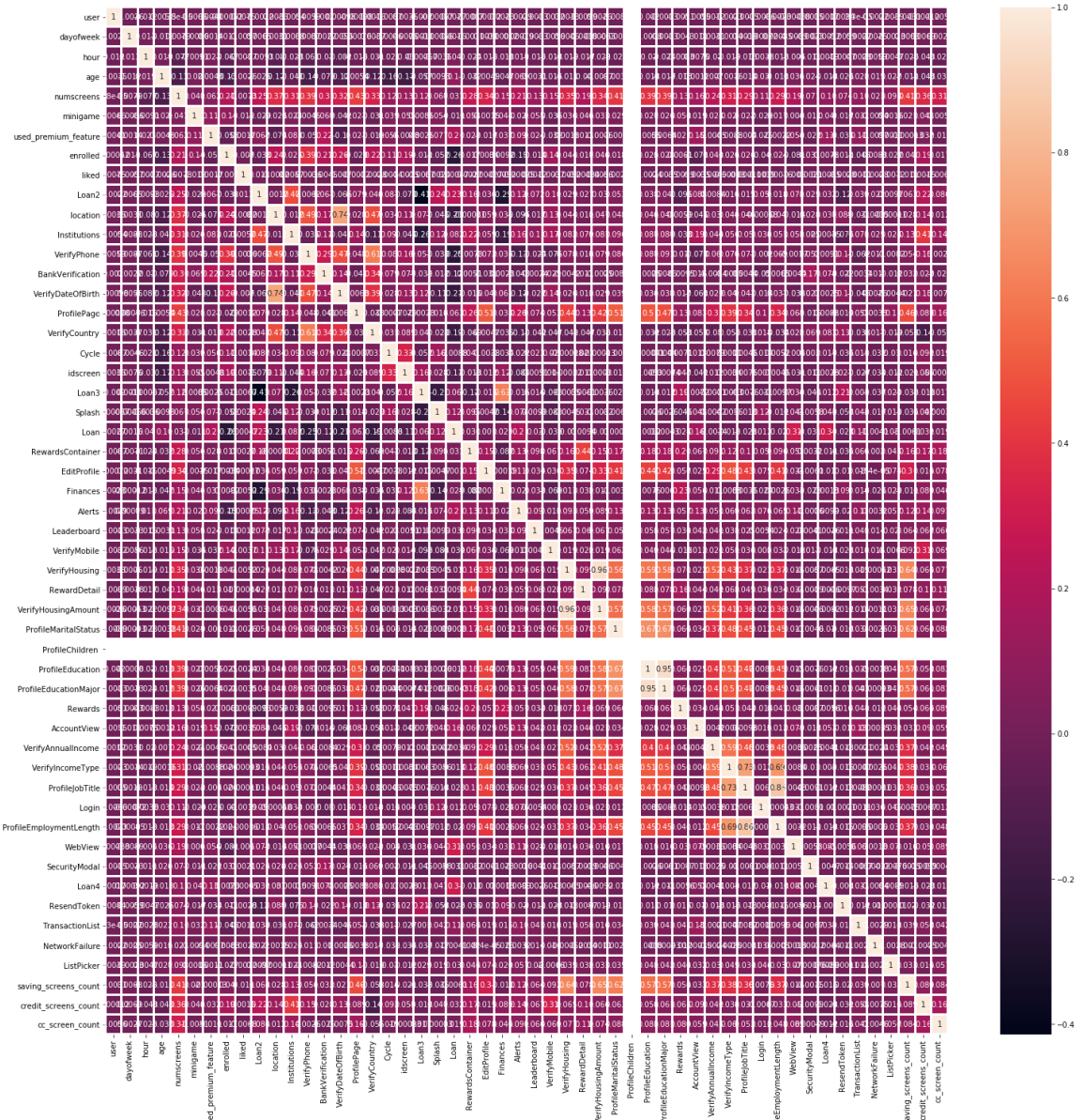
```
(50000, 52)
```

In [85]:

```
plt.figure(figsize=(25,25))
sns.heatmap(df3.corr(),annot=True,linewidth=2)
```

Out[85]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x122834eeb48>
```



In [87]:

```
x = df3.drop(['enrolled'],axis=1)
```

In [88]:

```
y = df.enrolled
```

In [90]:

```
from sklearn.model_selection import train_test_split
```

In [91]:

```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3)
```

In [92]:

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train_sc = sc.fit_transform(x_train)
x_test_sc = sc.fit_transform(x_test)
```

In [96]:

```
from sklearn.tree import DecisionTreeClassifier
dt = DecisionTreeClassifier()
dt.fit(x_train,y_train)
dt.score(x_test,y_test)
```

Out[96]:

```
0.6905333333333333
```

In [101]:

```
from sklearn.neighbors import KNeighborsClassifier
```

In [103]:

```
kn = KNeighborsClassifier()
kn.fit(x_train,y_train)
kn.score(x_test,y_test)
```

Out[103]:

```
0.5617333333333333
```

In [106]:

```
from sklearn.naive_bayes import GaussianNB
nb = GaussianNB()
nb.fit(x_train,y_train)
nb.score(x_test,y_test)
```

Out[106]:

```
0.6434
```

In [112]:

```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression(random_state = 0, penalty = 'l2')
lr.fit(x_train,y_train)
lr.score(x_test,y_test)
```

Out[112]:

```
0.6136
```

In [114]:

```
from sklearn.svm import SVC
s = SVC()
s.fit(x_train,y_train)
s.score(x_test,y_test)
```

Out[114]:

```
0.6136
```

In [122]:

```
from xgboost import XGBClassifier
xb = XGBClassifier()
xb.fit(x_train,y_train)
y_predict = xb.predict(x_test)
xb.score(x_test,y_test)
```

Out[122]:

```
0.766
```

In [119]:

```
y_predict
```

Out[119]:

```
array([1, 1, 1, ..., 0, 1, 1], dtype=int64)
```

In [123]:

```
from sklearn.metrics import confusion_matrix
cnf_mat = confusion_matrix(x_train,y_train,y_predict)
sns.heatmap(cnf_mat,annot = True, fmt ="g")
plt.title("Confussion Matrix", fontsize = 20)
```

```
---------------------------------------------------------------------------
-
ValueError                                Traceback (most recent call las
t)
<ipython-input-123-abae672ec9d1> in <module>
      1 from sklearn.metrics import confusion_matrix
----> 2 cnf_mat = confusion_matrix(x_train,y_train,y_predict)
      3 sns.heatmap(cnf_mat,annot = True, fmt ="g")
      4 plt.title("Confussion Matrix", fontsize = 20)

~\anaconda3\lib\site-packages\sklearn\metrics\_classification.py in confus
ion_matrix(y_true, y_pred, labels, sample_weight, normalize)
    266
    267     """
--> 268     y_type, y_true, y_pred = _check_targets(y_true, y_pred)
    269     if y_type not in ("binary", "multiclass"):
    270         raise ValueError("%s is not supported" % y_type)

~\anaconda3\lib\site-packages\sklearn\metrics\_classification.py in _check
_targets(y_true, y_pred)
     88     if len(y_type) > 1:
     89         raise ValueError("Classification metrics can't handle a mi
x of {0} "
---> 90                          "and {1} targets".format(type_true, type_
pred))
     91
     92     # We can't have more than one value on y_type => The set is no
more needed

ValueError: Classification metrics can't handle a mix of multiclass-multio
utput and binary targets
```

In [126]:

```
from sklearn.model_selection import cross_val_score
cross_validation = cross_val_score(estimator = xb, X = x_train, y = y_train, cv = 10)
```

In [ ]:

```

```