In [228…
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import preprocessing
from sklearn import linear_model
import os
import re
```

In [229…
```
path="C:\\Users\\moham\\Documents\\JNTU_DataScience\\JNTU_machine_learning\\ml_assignments\\M

df=pd.read_csv(path)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4712 entries, 0 to 4711
Data columns (total 9 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   name                  4712 non-null   object 
 1   start_time            4712 non-null   object 
 2   usage_time            4712 non-null   object 
 3   IP                    4712 non-null   object 
 4   MAC                   4712 non-null   object 
 5   upload                4712 non-null   object 
 6   download              4712 non-null   object 
 7   total_transfer        4712 non-null   float64
 8   seession_break_reason  4703 non-null   object 
dtypes: float64(1), object(8)
memory usage: 331.4+ KB
```

In [197…
```
df.start_time
```

Out[197]:
```
0       10-05-2022 02:59
1       10-05-2022 18:53
2       10-05-2022 21:20
3       11-05-2022 00:37
4       11-05-2022 02:59
              ...
4707    04-11-2022 01:11
4708    04-11-2022 10:26
4709    04-11-2022 20:41
4710    05-11-2022 00:21
4711    05-11-2022 20:55
Name: start_time, Length: 4712, dtype: object
```

In [198…
```
name=df.name
name.value_counts()
```

Out[198]:
```
user4    727
user1    674
user6    674
user9    572
user7    528
user3    519
user2    457
user5    336
user8    225
Name: name, dtype: int64
```

In [199…
```
t=pd.DataFrame()
t=df['start_time'].str[11:-3]
print(t)
#t.value_counts()
```

```
0       02
1       18
2       21
3       00
4       02
        ..
4707    01
4708    10
4709    20
4710    00
4711    20
Name: start_time, Length: 4712, dtype: object
```

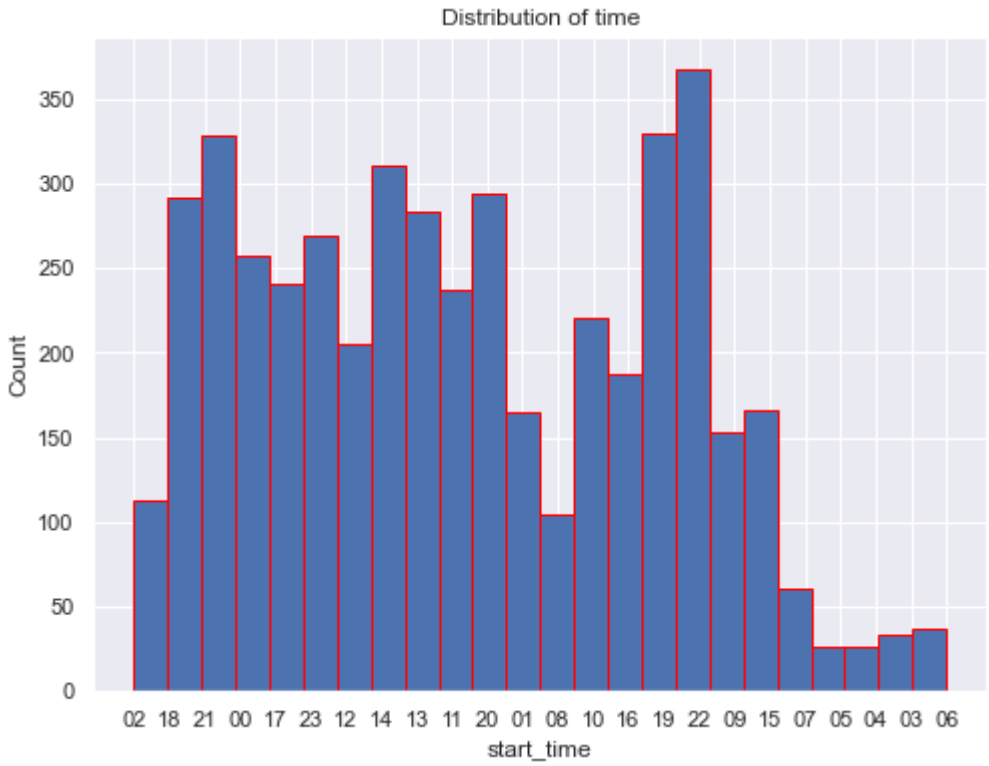
In [200…

```
t.info()
```

```
<class 'pandas.core.series.Series'>
RangeIndex: 4712 entries, 0 to 4711
Series name: start_time
Non-Null Count  Dtype
--------------  -----
4712 non-null   object
dtypes: object(1)
memory usage: 36.9+ KB
```

In [146…

```
fig=plt.figure(figsize=(8,6))
axes=fig.add_subplot(111)
axes.set(title='Distribution of time',ylabel='Count',xlabel='start_time')
axes.hist(t,bins=24,orientation='vertical',edgecolor='red')
plt.show()
```



Frequent time is 22:00,19:00,21:00,14:00

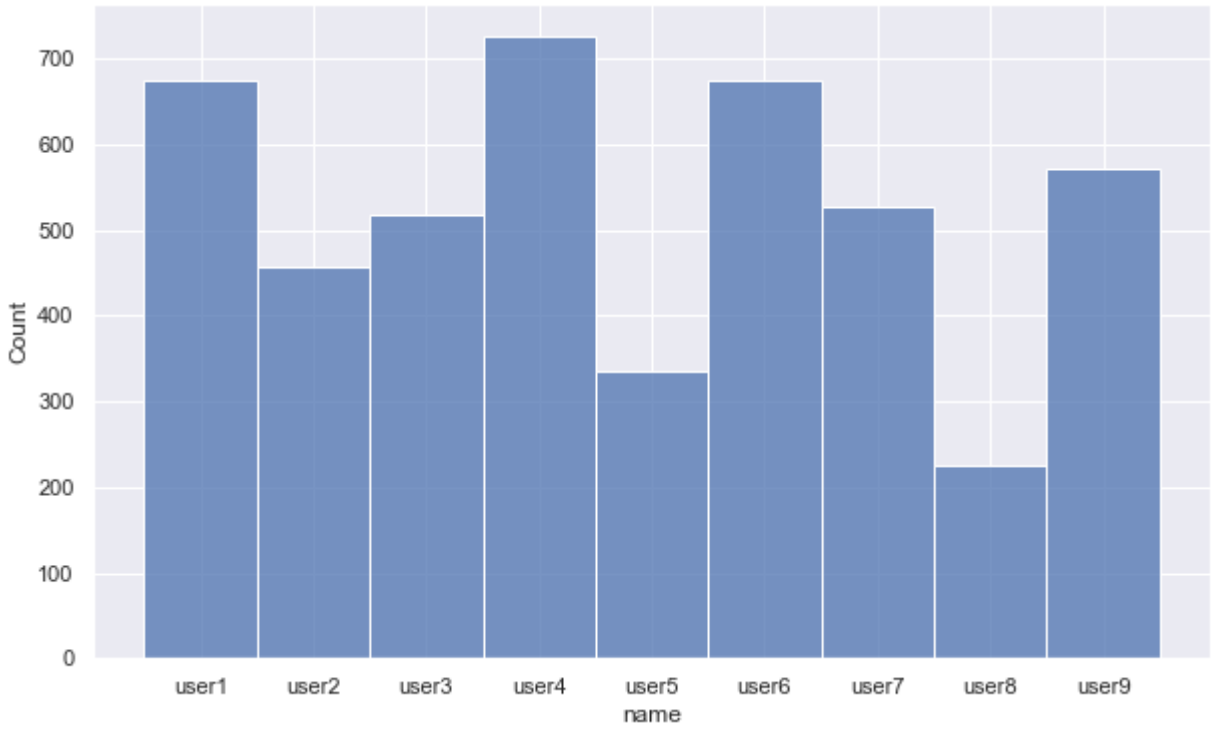
In [147…

```
ip=df[['name','IP']]
ip
ipcount=ip.groupby(['name','IP'])['IP'].count()
ipcount
```

Out[147]:
```
name   IP
user1  10.55.0.202     1
       10.55.0.238     7
       10.55.0.242     7
       10.55.0.245     9
       10.55.0.248    22
                      ..
user9  10.55.9.51      2
       10.55.9.65      1
       10.55.9.73      3
       10.55.9.79      1
       10.55.9.98      1
Name: IP, Length: 1629, dtype: int64
```

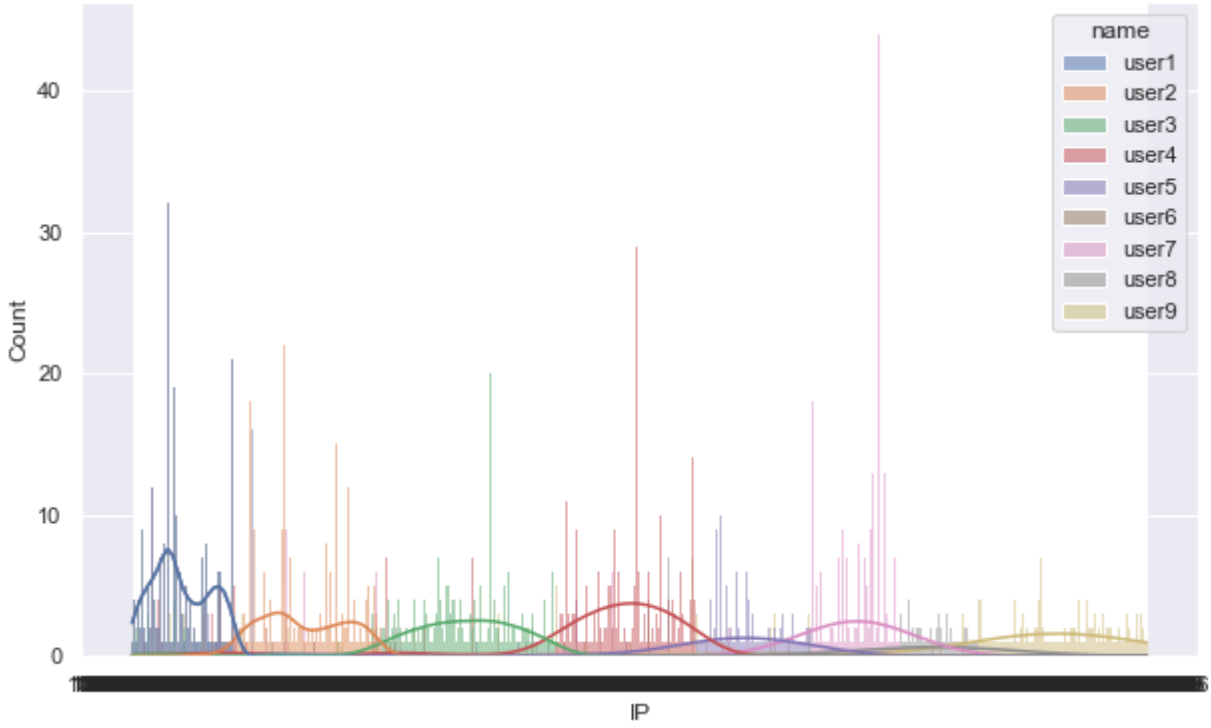
In [148…
```
sns.set(rc={'figure.figsize':(10,6)})
sns.histplot(data=ip,x="name",bins=30)
```

Out[148]:
```
<AxesSubplot:xlabel='name', ylabel='Count'>
```



In [149…
```
sns.histplot(data=ip,x='IP',bins=30,hue='name',kde=True)
```

Out[149]:
```
<AxesSubplot:xlabel='IP', ylabel='Count'>
```



As the distribution is not normal ,we cqn come to a conclusion that uesr1 is fluctuating more with IP

In [150…

```
device=df[['name','MAC']]
device
dcount=device.groupby(['name','MAC'])['MAC'].count()
dcount
```

Out[150]:

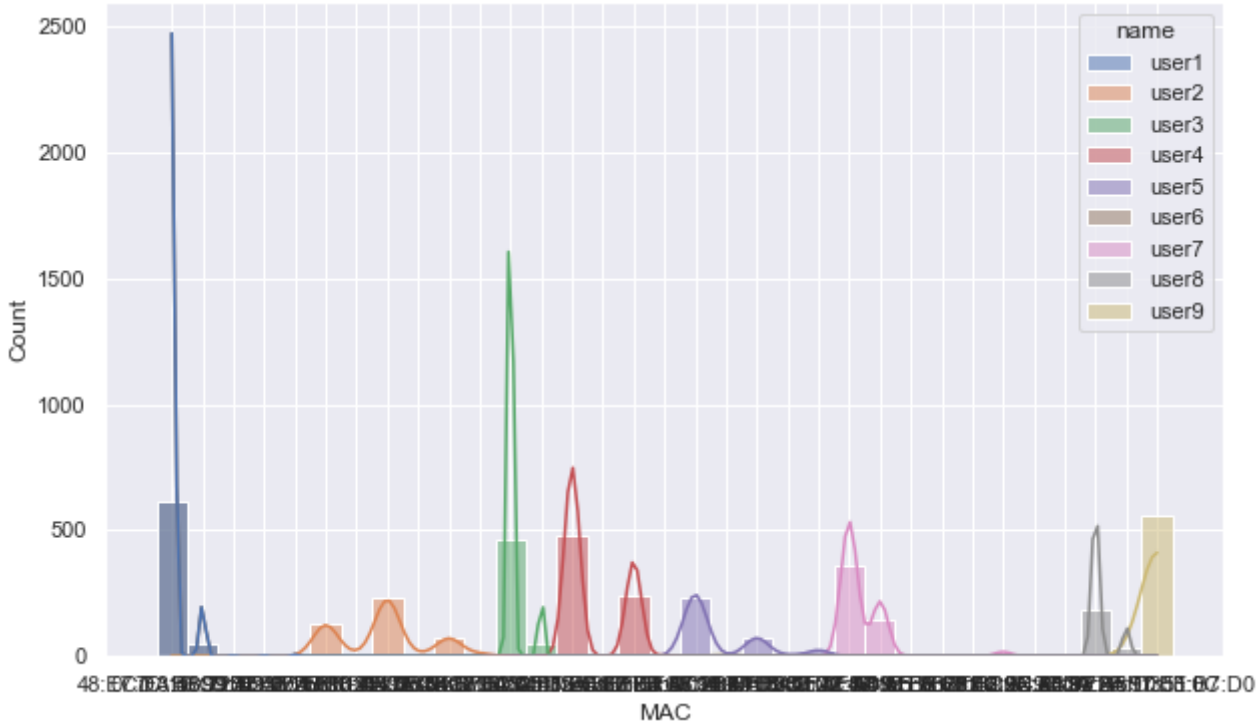
```
name   MAC
user1  0C:F3:46:71:E2:27     52
       48:E7:DA:58:22:E9    618
       7C:10:C9:AD:6E:E6      1
       90:32:4B:3B:10:DB      2
       B6:99:3E:1D:FB:19      1
user2  32:25:C6:7D:5B:54      7
       36:D2:09:C8:FA:0D     73
       A8:93:4A:7E:34:6F    231
       CA:48:87:B2:A5:12     11
       EC:2E:98:CB:B7:8D      7
       FA:B5:D7:B1:A4:6B    128
user3  C0:E4:34:D5:88:0F     54
       C8:3D:DC:CF:16:C6    465
user4  80:F3:EF:36:7D:AD      1
       92:02:4F:EE:EB:3F    481
       AA:E1:02:47:2B:0A      3
       D8:9C:67:BA:DC:B9    242
user5  68:14:01:09:51:71     70
       94:17:00:37:AF:A8      5
       C2:BB:83:2B:FF:5A    236
       E0:D0:45:5E:60:F5     23
       E8:6F:38:A4:F8:2F      2
user6  0C:F3:46:71:E2:27     52
       48:E7:DA:58:22:E9    618
       7C:10:C9:AD:6E:E6      1
       90:32:4B:3B:10:DB      2
       B6:99:3E:1D:FB:19      1
user7  72:47:85:EE:2F:8C      2
       86:12:FC:78:C9:94      2
       90:E8:68:F2:9C:21      1
       B0:68:E6:9E:AB:77     12
       C6:2C:1D:5E:86:70    147
       E2:42:6C:64:6E:48      1
       EC:2E:98:93:B5:D3      1
       F8:89:D2:D4:AE:8B    362
user8  2A:7F:97:6A:10:51    189
       9C:29:76:F1:E3:0C     36
user9  C2:BB:83:2B:FF:5A      1
       DA:2F:97:0E:B7:D0    561
       E8:6F:38:A4:F8:2F     10
Name: MAC, dtype: int64
```

In [151…

```
sns.histplot(data=device,x='MAC',bins=40,hue='name',kde=True)
```

Out[151]:

```
<AxesSubplot:xlabel='MAC', ylabel='Count'>
```



We analyse that user7 is often changing the device

In [152…
```
df.describe()
```

Out[152]:

	total_transfer
count	4.712000e+03
mean	4.303743e+05
std	9.952647e+05
min	1.120000e+00
25%	6.187008e+04
50%	2.029312e+05
75%	4.996915e+05
max	2.855272e+07

In [243…
```
#average_1=pd.DataFrame()
#average_1=df[['total_transfer'],['start_time']]
#average_1.info()

#df.resample('H', on='start_time').total_transfer.sum()

#datetime_series['start_time'] = pd.to_datetime(df['start_time'])
df['start_time'] = pd.to_datetime(df.start_time, format='%d-%m-%Y %H:%M')

#df['start_time'].dt.hour
#df['start_time'].dt.day_name()
#df['start_time'].dt.month_name()

df.groupby(df['start_time'].dt.hour).total_transfer.mean()
```

Out[243]:
```
start_time
0     464530.443023
1     530880.856788
2     431576.112743
3     345303.341176
4     359809.443333
5     275960.910769
6     468959.586757
7     292886.830164
8     366681.918762
9     377480.638954
10    393259.119955
11    309492.445992
12    310137.981415
13    335270.579648
14    472403.712765
15    517005.111506
16    403919.401872
17    525423.692116
18    665414.452500
19    390839.426364
20    355740.055442
21    471461.399116
22    449600.499185
23    407785.083903
Name: total_transfer, dtype: float64
```

In [245…
```
df['start_time'] = pd.to_datetime(df.start_time, format='%d-%m-%Y %H:%M')

#df['start_time'].dt.hour
#df['start_time'].dt.day_name()
#df['start_time'].dt.month_name()

df.groupby(df['start_time'].dt.day_name()).total_transfer.mean()
```

Out[245]:
```
start_time
Friday       399411.493425
Monday       437467.678774
Saturday     468626.719857
Sunday       484650.236789
Thursday     462790.832707
Tuesday      367632.288851
Wednesday    391861.088848
Name: total_transfer, dtype: float64
```

In [246…
```
df['start_time'] = pd.to_datetime(df.start_time, format='%d-%m-%Y %H:%M')

#df['start_time'].dt.hour
#df['start_time'].dt.day_name()
#df['start_time'].dt.month_name()

df.groupby(df['start_time'].dt.month_name()).total_transfer.mean()
```

Out[246]:
```
start_time
August       479042.438202
July         418583.993765
June         338418.082988
May          311177.156960
November     399675.450813
October      549467.626233
September    482955.522841
Name: total_transfer, dtype: float64
```

In [254…
```
df['start_time'] = pd.to_datetime(df.start_time, format='%d-%m-%Y %H:%M')
df.groupby(df['start_time'].dt.date).total_transfer.mean()
```

Out[254]:
```
start_time
2022-05-09    109844.480000
2022-05-10    151600.782667
2022-05-11    411055.589200
2022-05-12    340207.616000
2022-05-13    297072.926250
                  ...
2022-11-01    374462.644706
2022-11-02    463347.552895
2022-11-03    348276.877241
2022-11-04    424498.885517
2022-11-05    373287.936000
Name: total_transfer, Length: 154, dtype: float64
```

In [27]:
```
from sklearn.compose import make_column_selector as selector

categorical_columns_selector = selector(dtype_include=object)
categorical_columns = categorical_columns_selector(df)
categorical_columns
```

Out[27]:
```
['name',
 'start_time',
 'usage_time',
 'IP',
 'MAC',
 'upload',
 'download',
 'seession_break_reason']
```

In [34]:
```
data_categorical = df[categorical_columns]
data_categorical.head()
#data_categorical.info()
```

Out[34]:

	name	start_time	usage_time	IP	MAC	upload	download	seession_break_reason
0	user1	10-05-2022 02:59	00:00:36:28	10.55.14.222	48:E7:DA:58:22:E9	15861.76	333168.64	Idle-Timeout
1	user1	10-05-2022 18:53	00:01:49:56	10.55.2.253	48:E7:DA:58:22:E9	16957.44	212152.32	Idle-Timeout
2	user1	10-05-2022 21:20	00:01:35:00	10.55.2.253	48:E7:DA:58:22:E9	14080	195153.92	Idle-Timeout
3	user1	11-05-2022 00:37	00:00:26:00	10.55.2.253	48:E7:DA:58:22:E9	5242.88	40806.4	Idle-Timeout
4	user1	11-05-2022 02:59	00:00:11:52	10.55.2.253	48:E7:DA:58:22:E9	22067.2	10772.48	Idle-Timeout

In [40]:
```
from sklearn.preprocessing import OrdinalEncoder

start_time_column = data_categorical[["start_time"]]

encoder = OrdinalEncoder()
start_time_encoded = encoder.fit_transform(start_time_column)
start_time_encoded
```

Out[40]:
```
array([[1038.],
       [1054.],
       [1060.],
       ...,
       [ 469.],
       [ 564.],
       [ 585.]])
```

In [ ]:

In [41]:
```
usage_time_column = data_categorical[["usage_time"]]

encoder = OrdinalEncoder()
usage_time_encoded = encoder.fit_transform(usage_time_column)
usage_time_encoded
```

Out[41]:
```
array([[ 868.],
       [2004.],
       [1836.],
       ...,
       [1686.],
       [3418.],
       [1442.]])
```

In [ ]:

In [31]:
```
encoder.categories_
```

Out[31]:
```
[array(['01-06-2022 00:30', '01-06-2022 01:48', '01-06-2022 05:53', ...,
        '31-10-2022 22:12', '31-10-2022 22:40', '31-10-2022 22:46'],
       dtype=object)]
```

In [32]:
```
df_encoded = encoder.fit_transform(data_categorical)
df_encoded[:5]
```

Out[32]:
```
array([[   0., 1038.,  868.,  522.,    4.,  552., 1807.,    0.],
       [   0., 1054., 2004.,  701.,    4.,  641., 1051.,    0.],
       [   0., 1060., 1836.,  701.,    4.,  401.,  938.,    0.],
       [   0., 1178.,  643.,  701.,    4., 2152., 2184.,    0.],
       [   0., 1180.,  325.,  701.,    4.,  991.,  117.,    0.]])
```

In [ ]:

In [ ]:

In [ ]:

In [ ]: