In [47]:
```
###DONOT DELETE THIS CELL. ALWAYS RUN THIS CELL BEFORE SUBMISSION
eventid = '20210823'
questionid = 'hack-d8d04194-ccfd-4469-8e0a-275e1b86ad72'
```

In [48]:
```
import mathlogic as ml
import pandas as pd
import numpy as np
```

In [49]:
```
### Data on which you need to create features
df = pd.read_csv("/home/data/telecom_train_v2.csv")
df.head()
```

Out[49]:

	MNTH	id	ROAM_INT_REV	INTN_CALLS	LAST_INTN_CALL_MNTH	INTN_REV	CALL_MIN	R
0	202010	825139825	0.0	0	NaN	0.0	168.200000	
1	202102	822439533	0.0	0	NaN	0.0	169.150000	
2	202009	820530656	0.0	0	NaN	0.0	677.800000	
3	202011	828618560	0.0	0	NaN	0.0	48.616667	
4	202012	818400281	0.0	0	NaN	0.0	75.850000	

In [50]:
```
df1=df[df['MNTH']!=202102]
df1=df1.fillna(99)
df1=df1.sort_values(by=['id','MNTH'])
df1['Total_Rev']=df1['ROAM_INT_REV']+df1['INTN_REV']+df1['RMNG_REVN']+df1['VOICE_REV']+
df1.head()
```

Out[50]:

	MNTH	id	ROAM_INT_REV	INTN_CALLS	LAST_INTN_CALL_MNTH	INTN_REV	CALL_M
178918	202008	800000008	0.0	0	99.0	0.0	791.3000
56196	202009	800000008	0.0	0	99.0	0.0	718.0000
461449	202010	800000008	0.0	0	99.0	0.0	641.3000
628767	202011	800000008	10.0	0	99.0	10.0	629.9833
570927	202012	800000008	11.2	1	202012.0	11.2	483.5166

In [51]:
```
df2=df1.groupby('id').agg(#RATIO_HIGH_LOW_SMS_REV=('SMS_REV',lambda y:max(y)/min(y) if
                          RATIO_HIGH_LOW_MONEY_TRN_REV=('MONEY_TRN_REV',lambda y:max(y)
                          RATIO_HIGH_LOW_CNT_ADV_DATA=('CNT_ADV_DATA',lambda y:max(y)/m
                          RATIO_HIGH_LOW_NUM_CHARGES_MNTH=('NUM_CHARGES_MNTH',lambda y:
                          MEAN_ROAM_INT_REV=('ROAM_INT_REV','mean'),MEAN_INTN_CALLS=('I
                          RATIO_NUM_INT_MNTHS=('INTN_CALLS',lambda y: len(y[y>0])/len(y
df2.head()
#RATIO_HIGH_LOW_ROAM_INT_REV=('ROAM_INT_REV',lambda y:max(y)/min(y) if (len(list(y))>=1
```

Out[51]:

id	RATIO_HIGH_LOW_MONEY_TRN_REV	RATIO_HIGH_LOW_CNT_ADV_DATA	RATIO_HIGH_LOW_NU
800000008	8.41925	-1.000000	
800000193	-1.00000	-1.000000	
800001524	-1.00000	-1.000000	
800001559	-1.00000	-1.000000	
800002101	-1.00000	1.916667	

5 rows × 30 columns

In [52]:

```
df2.MAX_LAST_INTN_CALL_MNTH.astype('Int64')
df2[['MAX_LAST_INTN_CALL_MNTH','MAX_LAST_RCHG_MNTH','MAX_LAST_ADV_DATA_MNTH']]=df2[['MA
df2['MAX_MNTHS']=df2['MAX_MNTHS'].replace([202008.0,202009.0,202010.0,202011.0,202012.0
df2.head()
```

Out[52]:

id	RATIO_HIGH_LOW_MONEY_TRN_REV	RATIO_HIGH_LOW_CNT_ADV_DATA	RATIO_HIGH_LOW_NU
800000008	8.41925	-1.000000	
800000193	-1.00000	-1.000000	
800001524	-1.00000	-1.000000	
800001559	-1.00000	-1.000000	
800002101	-1.00000	1.916667	

5 rows × 30 columns

In [53]:

```
JAN=df[df['MNTH']==202101].groupby('id')['VOICE_REV'].agg('max')
JAN=JAN.fillna(0)
DEC=df[df['MNTH']==202012].groupby('id')['VOICE_REV'].agg('max')
DEC=DEC.fillna(0)
df5=pd.merge(JAN,DEC,how='outer',on='id')
df5=df5.fillna(0)
df5['diff']=df5['VOICE_REV_x']-df5['VOICE_REV_y']
df6=pd.DataFrame()
df6['MAX_DIFF_VOICE_REV']=df5['diff'].fillna(0)
df4=pd.concat([df2,df6],axis=1).fillna(0)
```

In [54]:

```
JAN_DIV=df[df['MNTH']==202101].groupby('id')['NUM_CHARGES_MNTH'].agg('max')
JAN_DIV=JAN_DIV.fillna(0)
DEC_DIV=df[df['MNTH']==202012].groupby('id')['NUM_CHARGES_MNTH'].agg('max')
DEC_DIV=DEC_DIV.fillna(0)
df7=pd.merge(JAN_DIV,DEC_DIV,how='outer',on='id')
df7=df7.fillna(0)
```

```
df7['div']=df7['NUM_CHARGES_MNTH_x']/df7['NUM_CHARGES_MNTH_y']
df7=df7.fillna(-1)
df7=df7.replace([np.inf],[-1])
df8=pd.DataFrame()
df8['MAX_RATIO_NUM_CHARGES_MNTH']=df7['div']
df8=df8.fillna(-1)
df4=pd.concat([df4,df8],axis=1).fillna(-1)

df4['MAX_RATIO_NUM_CHARGES_MNTH'].isnull().sum()
```

Out[54]:
```
0
```

In [55]:
```
df1['MNTH']=df1['MNTH'].replace([202008.0,202009.0,202010.0,202011.0,202012.0,202101.0]
```

In [56]:
```
df1=df1.set_index('MNTH')
df1.head(10)
```

Out[56]:

MNTH	id	ROAM_INT_REV	INTN_CALLS	LAST_INTN_CALL_MNTH	INTN_REV	CALL_MIN	RMN
0	800000008	0.0000	0	99.0	0.0000	791.300000	
1	800000008	0.0000	0	99.0	0.0000	718.000000	
2	800000008	0.0000	0	99.0	0.0000	641.300000	
3	800000008	10.0000	0	99.0	10.0000	629.983333	
4	800000008	11.2000	1	202012.0	11.2000	483.516667	
5	800000008	20.0000	0	99.0	20.0000	260.050000	
0	800000193	793.3439	42	202008.0	793.3439	545.466667	
1	800000193	393.7819	34	202009.0	393.7819	263.183333	
2	800000193	369.9392	40	202010.0	369.9392	364.050000	
3	800000193	47.7493	8	202011.0	47.7493	243.583333	

In [57]:
```
def myfunc(y):
    count=0
    rev=list(y.values)
    m=list(y.index)
    j=[0,0,0,0,0,0]
    for i in range(len(m)):
        j[m[i]]=rev[i]
    for i in range(1,len(j)):
        if (j[i]>j[i-1]):
            count+=1
    return count
```

In [58]:
```
df12=pd.DataFrame()
df12['MNTHS_STABLE_REV']=df1.groupby('id').agg(MNTHS_STABLE_REV=('Total_Rev',lambda y:
```

```
df12.head()
```

Out[58]:

id	MNTHS_STABLE_REV
800000008	3
800000193	3
800001524	2
800001559	2
800002101	3

In [59]:

```
ans=pd.DataFrame()
ans=pd.concat([df12,df4],axis=1)
ans.head()
```

Out[59]:

id	MNTHS_STABLE_REV	RATIO_HIGH_LOW_MONEY_TRN_REV	RATIO_HIGH_LOW_CNT_ADV_DATA
800000008	3	8.41925	-1.000000
800000193	3	-1.00000	-1.000000
800001524	2	-1.00000	-1.000000
800001559	2	-1.00000	-1.000000
800002101	3	-1.00000	1.916667

5 rows × 33 columns

In [60]:

```
#ans['RATIO_NEW']=(ans['TOT_INTN_REV']/ans['TOT_INTN_CALLS']).fillna(-1)
#ans['RATIO_NEW']=ans['RATIO_NEW'].replace([np.inf],[-1])
```

In [61]:

```
ans=ans.reset_index()
#ml.eval(questionid, eventid,ans)
```

In [62]:

```
df_feb=pd.DataFrame()
df_feb=df[df['MNTH']==202102]
df_feb=df_feb['id']
df_feb=pd.DataFrame(df_feb)
df_feb["NC"] = 0
df_feb.head()
```

Out[62]:

	id	NC
1	822439533	0
44	808956254	0

	id	NC
58	815981800	0
59	826818859	0
76	814998939	0

In [63]:
```
ans.head()
```

Out[63]:

	id	MNTHS_STABLE_REV	RATIO_HIGH_LOW_MONEY_TRN_REV	RATIO_HIGH_LOW_CNT_ADV_DAT
0	800000008	3	8.41925	-1.0000
1	800000193	3	-1.00000	-1.0000
2	800001524	2	-1.00000	-1.0000
3	800001559	2	-1.00000	-1.0000
4	800002101	3	-1.00000	1.9166

5 rows × 34 columns

In [64]:
```
mer=pd.merge(ans,df_feb,how="left",on="id")
mer["NC"]=mer["NC"].fillna(1)
mer.to_csv(r"C:\Users\rakesh.doddamani\Desktop\Accenture\new_data2.csv")
```

In [65]:
```
#mer.columns
```

In [66]:
```
#mer["NC"].isnull().sum()
```

In [67]:
```
#mer["NC"]=mer["NC"].fillna(1)
```

In [68]:
```
#mer["NC"].value_counts()
```

In [69]:
```
#mer.to_csv(r"C:\Users\rakesh.doddamani\Desktop\Accenture\new_data1.csv")
```

In [ ]:
```

```