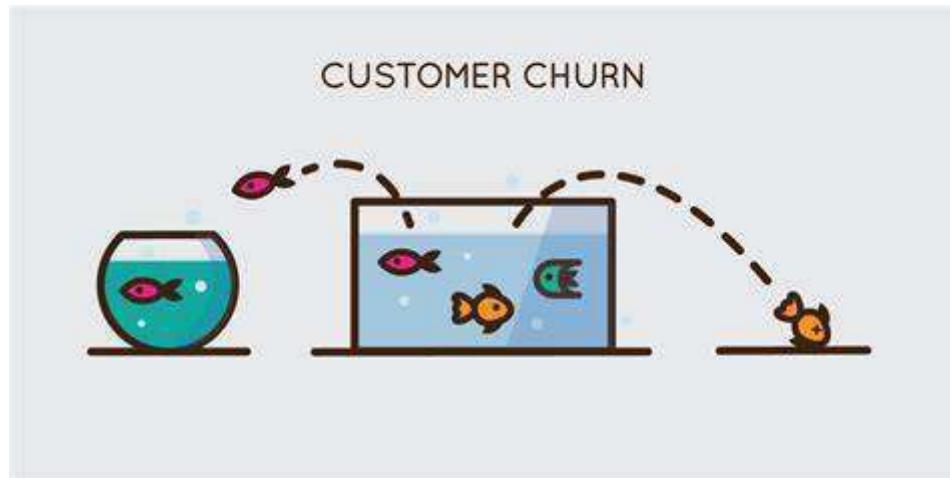


```
In [353]: from IPython.display import Image  
Image(filename='churn_rate.jpg')
```

Out[353]:



```
In [1]: import pandas as pd  
import numpy as np  
import seaborn as sns  
import matplotlib.pyplot as plt  
from sklearn.feature_selection import VarianceThreshold  
from sklearn.feature_selection import mutual_info_classif  
from sklearn.feature_selection import RFE  
from scipy.stats import chi2_contingency
```

```
In [2]: churn=pd.read_csv('churn.csv')
```

```
In [53]: df=churn.copy()
```

```
In [6]: df.head(2)
```

Out[6]:

	Unnamed: 0	age	gender	security_no	region_category	membership_category	joining_date	joine
0	0	18	F	XW0DQ7H	Village	Platinum Membership	2017-08-17	
1	1	32	F	5K0N3X1	City	Premium Membership	2017-08-28	

2 rows × 24 columns

```
In [7]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 36992 entries, 0 to 36991
Data columns (total 24 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Unnamed: 0        36992 non-null   int64  
 1   age              36992 non-null   int64  
 2   gender            36992 non-null   object  
 3   security_no       36992 non-null   object  
 4   region_category  31564 non-null   object  
 5   membership_category 36992 non-null   object  
 6   joining_date     36992 non-null   object  
 7   joined_through_referral 36992 non-null   object  
 8   referral_id      36992 non-null   object  
 9   preferred_offer_types 36704 non-null   object  
 10  medium_of_operation 36992 non-null   object  
 11  internet_option  36992 non-null   object  
 12  last_visit_time  36992 non-null   object  
 13  days_since_last_login 36992 non-null   int64  
 14  avg_time_spent   36992 non-null   float64 
 15  avg_transaction_value 36992 non-null   float64 
 16  avg_frequency_login_days 36992 non-null   object  
 17  points_in_wallet  33549 non-null   float64 
 18  used_special_discount 36992 non-null   object  
 19  offer_application_preference 36992 non-null   object  
 20  past_complaint    36992 non-null   object  
 21  complaint_status  36992 non-null   object  
 22  feedback           36992 non-null   object  
 23  churn_risk_score  36992 non-null   int64  
dtypes: float64(3), int64(4), object(17)
memory usage: 6.8+ MB
```

```
In [54]: df.drop(columns=['Unnamed: 0','security_no','referral_id'],inplace=True)
```

```
In [55]: df.shape
```

```
Out[55]: (36992, 21)
```

```
In [10]: df.head(2)
```

Out[10]:

	age	gender	region_category	membership_category	joining_date	joined_through_referral	prefe
0	18	F	Village	Platinum Membership	2017-08-17	No	Gift V
1	32	F	City	Premium Membership	2017-08-28	?	Gift V

2 rows × 21 columns



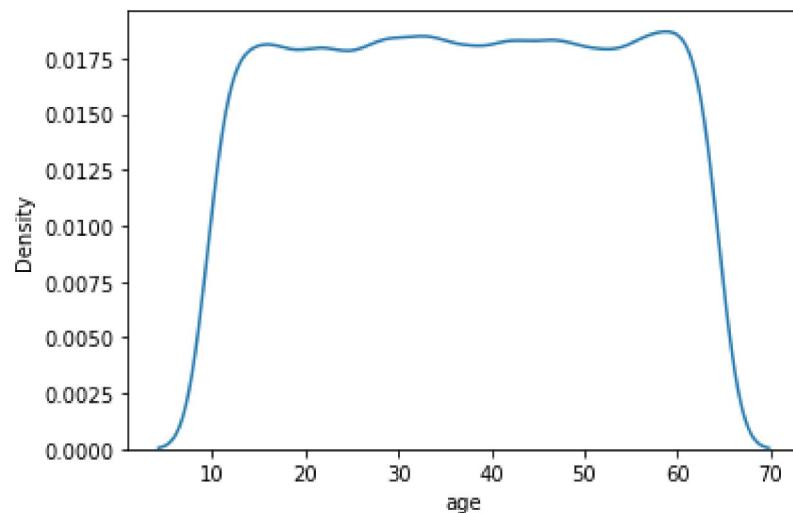
```
In [14]: df.isna().mean()
```

```
Out[14]: age                  0.000000
gender                0.000000
region_category      0.146734
membership_category  0.000000
joining_date         0.000000
joined_through_referral  0.000000
preferred_offer_types  0.007785
medium_of_operation   0.000000
internet_option      0.000000
last_visit_time       0.000000
days_since_last_login 0.000000
avg_time_spent        0.000000
avg_transaction_value 0.000000
avg_frequency_login_days 0.000000
points_in_wallet      0.093074
used_special_discount 0.000000
offer_application_preference 0.000000
past_complaint        0.000000
complaint_status      0.000000
feedback              0.000000
```

```
In [23]: def correlation(df):
    return df.corr(method='spearman')['churn_risk_score']
```

```
In [16]: sns.kdeplot(df['age'])
```

```
Out[16]: <AxesSubplot:xlabel='age', ylabel='Density'>
```



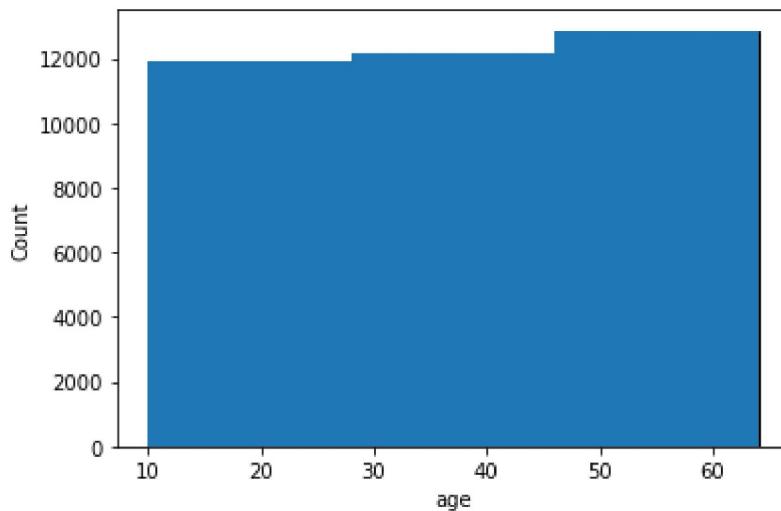
```
In [17]: ### normal distributed but variance is high.the data are distributed far from the mean  
df['age'].std()
```

```
Out[17]: 15.86741220457268
```

In [19]:

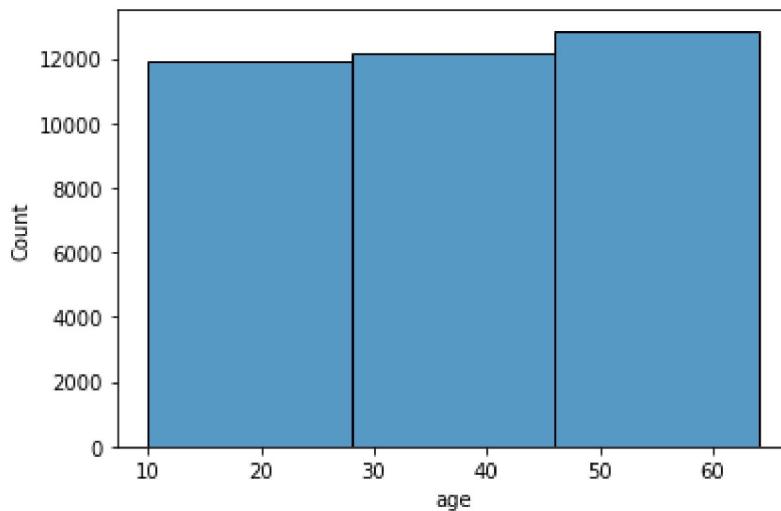
```
plt.hist(df['age'], bins=3)[1]
```

Out[19]: array([10., 28., 46., 64.])



In [20]: sns.histplot(df['age'], bins=3)

Out[20]: <AxesSubplot:xlabel='age', ylabel='Count'>



In [24]: correlation(df)

```
age           0.006102
days_since_last_login 0.012238
avg_time_spent      -0.020681
avg_transaction_value -0.150094
points_in_wallet     -0.556742
churn_risk_score      1.000000
Name: churn_risk_score, dtype: float64
```

```
In [56]: def make_age_bins(df,col):
    ctpt=[9, 28., 46., 64.]
    labels=['young','mid','high']
    df[col+'_range']=pd.cut(df[col],ctpt,labels=labels)
    return df
```

```
In [57]: df=make_age_bins(df,'age')
chi2_contingency(pd.crosstab(df['age_range'],df['churn_risk_score']))
```

```
Out[57]: (3.1740133105826525,
0.20453694589553115,
2,
array([[5797.39943772, 6832.60056228],
       [5589.92322664, 6588.07677336],
       [5592.67733564, 6591.32266436]]))
```

since the value of p is greater than the significant value that is 0.05 therefore we are going to accept the null hypothesis

```
In [31]: df.head()
```

```
Out[31]:
```

	age	gender	region_category	membership_category	joining_date	joined_through_referral	prefe
0	18	F	Village	Platinum Membership	2017-08-17	No	Gift V
1	32	F	City	Premium Membership	2017-08-28	?	Gift V
2	44	F	Town	No Membership	2016-11-11	Yes	Gift V
3	37	M	City	No Membership	2016-10-29	Yes	Gift V
4	31	F	City	No Membership	2017-09-12	No	

5 rows × 22 columns

```
In [58]: df['age_range']=df['age_range'].replace({'young':0,'mid':1,'high':2}).astype(int)
```

```
In [59]: correlation(df)
```

```
Out[59]: age                  0.006102
days_since_last_login      0.012238
avg_time_spent            -0.020681
avg_transaction_value    -0.150094
points_in_wallet          -0.556742
churn_risk_score          1.000000
age_range                 0.006742
Name: churn_risk_score, dtype: float64
```

```
In [29]: chi2_contingency(pd.crosstab(df['gender'],df['churn_risk_score']))
```

```
Out[29]: (1.4568560193678342,
          0.4826671421035432,
          2,
          array([[ 8487.245891 , 10002.754109 ],
                 [ 8465.6720372, 9977.3279628],
                 [ 27.0820718,   31.9179282]]))
```

```
In [36]: ##### Here also the p value is greater than 0.05 so we have to accept the null hypothesis
```

```
In [38]: df.groupby(['gender'])['churn_risk_score'].mean().to_dict()
```

```
Out[38]: {'F': 0.5434829637641969,
          'M': 0.5386325435124437,
          'Unknown': 0.4915254237288136}
```

```
In [60]: df['gender']=df['gender'].replace(df.groupby(['gender'])['churn_risk_score'].mean())
```

```
In [41]: correlation(df)
```

```
Out[41]: age                  0.006102
gender                0.005164
days_since_last_login      0.012238
avg_time_spent            -0.020681
avg_transaction_value    -0.150094
points_in_wallet          -0.556742
churn_risk_score          1.000000
age_range                 0.006742
Name: churn_risk_score, dtype: float64
```

```
In [42]: df.head()
```

Out[42]:

category	membership_category	joining_date	joined_through_referral	preferred_offer_types	medium_of_operation	internet_option	last_visit_time	days_since_last_login	avg_time_spent	avg_transaction_value	avg_frequency_login_days	points_in_wallet	used_special_discount	offer_application_preference	past_complaint	complaint_status	feedback	churn_risk_score	age_range	dtype
Village	Platinum Membership	2017-08-17	No	Gift Vouchers/Coupons																
City	Premium Membership	2017-08-28	?	Gift Vouchers/Coupons																
Town	No Membership	2016-11-11	Yes	Gift Vouchers/Coupons																
City	No Membership	2016-10-29	Yes	Gift Vouchers/Coupons																
City	No Membership	2017-09-12	No	Credit/Debit Card Offers															S	



```
In [61]: df.isna().sum()
```

```
Out[61]: age                      0
gender                     0
region_category           5428
membership_category        0
joining_date                 0
joined_through_referral      0
preferred_offer_types       288
medium_of_operation          0
internet_option              0
last_visit_time                0
days_since_last_login          0
avg_time_spent                  0
avg_transaction_value          0
avg_frequency_login_days         0
points_in_wallet            3443
used_special_discount          0
offer_application_preference      0
past_complaint                  0
complaint_status                  0
feedback                         0
churn_risk_score                  0
age_range                         0
dtype: int64
```

```
In [50]: p_val=[]
for i in df.columns:
    if(i!='region_category'):
        p_val.append(chi2_contingency(pd.crosstab(df['region_category'],df[i]))[1])
```

```
In [62]: df1=pd.DataFrame(p_val,index=[i for i in df.columns if i!='region_category'])
```

```
In [67]: df[df1[df1[0]<0.05].index[:-1]]
```

Out[67]:

	membership_category	feedback
0	Platinum Membership	Products always in Stock
1	Premium Membership	Quality Customer Care
2	No Membership	Poor Website
3	No Membership	Poor Website
4	No Membership	Poor Website
...
36987	Basic Membership	No reason specified
36988	Basic Membership	Poor Customer Service
36989	Basic Membership	Poor Website
36990	Platinum Membership	No reason specified
36991	Silver Membership	Quality Customer Care

36992 rows × 2 columns

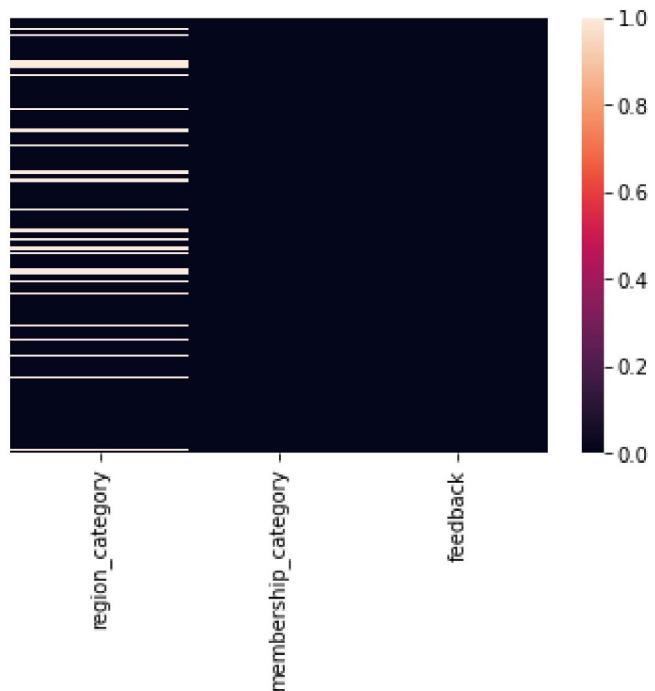
```
In [72]: df1[df1[0]<0.05]
```

Out[72]:

	0
membership_category	4.761268e-03
feedback	1.541975e-14
churn_risk_score	1.475413e-04

```
In [70]: sns.heatmap(df[['region_category','membership_category','feedback']].isna(),ytick
```

```
Out[70]: <AxesSubplot:>
```



```
In [79]: df.groupby(['region_category'])['feedback'].value_counts()
```

```
Out[79]: region_category feedback
City          Poor Product Quality    2213
              Too many ads           2205
              No reason specified   2202
              Poor Customer Service  2170
              Poor Website           2142
              Reasonable Price        464
              Quality Customer Care  459
              Products always in Stock 451
              User Friendly Website  431
Town          Poor Website            2473
              Poor Customer Service  2446
              Too many ads           2409
              No reason specified   2394
              Poor Product Quality   2381
              User Friendly Website  530
              Quality Customer Care  500
              Products always in Stock 498
              Reasonable Price        497
Village       Poor Product Quality   793
              No reason specified   759
              Poor Website            752
              Too many ads           737
              Poor Customer Service  734
              User Friendly Website  240
              Products always in Stock 233
              Reasonable Price        232
              Quality Customer Care  219
Name: feedback, dtype: int64
```

```
In [80]: reg_feed={'City':'Poor Product Quality','Town':'Poor Website','Village':'Poor Pro
```

```
In [81]: null=df[df['region_category'].isna()]
```

```
In [84]: null['region_category']=np.where(null['region_category']=='City',reg_feed['City'],
                                         np.where(null['region_category']=='Town',reg_feed['Town'],reg_feed['Villag
```

...

```
In [86]: df.update(null)
```

```
In [87]: df.isna().sum()
```

```
Out[87]: age 0  
gender 0  
region_category 0  
membership_category 0  
joining_date 0  
joined_through_referral 0  
preferred_offer_types 288  
medium_of_operation 0  
internet_option 0  
last_visit_time 0  
days_since_last_login 0  
avg_time_spent 0  
avg_transaction_value 0  
avg_frequency_login_days 0  
points_in_wallet 3443  
used_special_discount 0  
offer_application_preference 0  
past_complaint 0  
complaint_status 0
```

```
In [93]: chi2_contingency(pd.crosstab(df['region_category'],df['churn_risk_score']))
```

```
Out[93]: (19.224780862358422,  
 0.0002456447289590135,  
 3,  
 array([[2156.92636246, 2542.07363754],  
        [6485.00865052, 7642.99134948],  
        [5846.51438149, 6890.48561851],  
        [2491.55060554, 2936.44939446]]))
```

```
In [92]: df['region_category']=df['region_category'].replace(df.groupby(['region_category'])
```

```
In [94]: correlation(df)
```

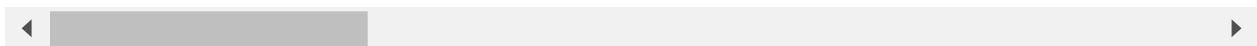
```
Out[94]: age 0.006102  
gender 0.005164  
region_category 0.017654  
days_since_last_login 0.012238  
avg_time_spent -0.020681  
avg_transaction_value -0.150094  
points_in_wallet -0.556742  
churn_risk_score 1.000000  
age_range 0.006742  
Name: churn_risk_score, dtype: float64
```

```
In [95]: df.head()
```

Out[95]:

	age	gender	region_category	membership_category	joining_date	joined_through_referral	pre
0	18.0	0.543483		0.512024	Platinum Membership	2017-08-17	No Gift
1	32.0	0.543483		0.547146	Premium Membership	2017-08-28	? Gift
2	44.0	0.543483		0.542044	No Membership	2016-11-11	Yes Gift
3	37.0	0.538633		0.547146	No Membership	2016-10-29	Yes Gift
4	31.0	0.543483		0.547146	No Membership	2017-09-12	No

5 rows × 22 columns



```
In [96]: chi2_contingency(pd.crosstab(df['membership_category'],df['churn_risk_score']))
```

```
Out[96]: (22851.551634762145,
          0.0,
          5,
          array([[3545.45631488, 4178.54368512],
                 [3119.02843858, 3675.97156142],
                 [3530.76773356, 4161.23226644],
                 [1991.2208045 , 2346.7791955 ],
                 [2044.92592993, 2410.07407007],
                 [2748.60077855, 3239.39922145]]))
```

```
In [100]: df['membership_category'].replace(df.groupby(['membership_category'])['churn_risk_score'].mean(), 0)
```

```
In [101]: correlation(df)
```

```
Out[101]: age                  0.006102
           gender                0.005164
           region_category        0.017654
           membership_category    0.755832
           days_since_last_login  0.012238
           avg_time_spent         -0.020681
           avg_transaction_value -0.150094
           points_in_wallet       -0.556742
           churn_risk_score        1.000000
           age_range                0.006742
Name: churn_risk_score, dtype: float64
```

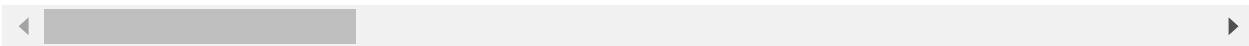
```
In [103]: df.drop(columns=['joining_date'], inplace=True)
```

```
In [104]: df.head()
```

Out[104]:

	age	gender	region_category	membership_category	joined_through_referral	preferred_offer_type
0	18.0	0.543483	0.512024	0.000000	No	Gift Vouchers/Coupons
1	32.0	0.543483	0.547146	0.000000	?	Gift Vouchers/Coupons
2	44.0	0.543483	0.542044	0.970619	Yes	Gift Vouchers/Coupons
3	37.0	0.538633	0.547146	0.970619	Yes	Gift Vouchers/Coupons
4	31.0	0.543483	0.547146	0.970619	No	Credit/Debit Card

5 rows × 21 columns



```
In [105]: df['joined_through_referral'].unique()
```

```
Out[105]: array(['No', '?', 'Yes'], dtype=object)
```

```
In [106]: chi2_contingency(pd.crosstab(df['joined_through_referral'],df['churn_risk_score']))
```

```
Out[106]: (42.708674504171356,
5.320214796682642e-10,
2,
array([[2496.1407872, 2941.8592128],
       [7270.3887327, 8568.6112673],
       [7213.4704801, 8501.5295199]]))
```

```
In [109]: df['joined_through_referral'].replace(df.groupby(['joined_through_referral'])['churn_risk_score'].mean(), inplace=True)
```

```
In [110]: correlation(df)
```

```
Out[110]: age           0.006102
gender          0.005164
region_category 0.017654
membership_category 0.755832
joined_through_referral 0.033316
days_since_last_login 0.012238
avg_time_spent      -0.020681
avg_transaction_value -0.150094
points_in_wallet     -0.556742
churn_risk_score      1.000000
age_range           0.006742
Name: churn_risk_score, dtype: float64
```

```
In [129]: df.isna().sum()
```

```
Out[129]: age           0
gender          0
region_category 0
membership_category 0
joined_through_referral 0
preferred_offer_types 288
medium_of_operation 0
internet_option    0
last_visit_time     0
days_since_last_login 0
avg_time_spent      0
avg_transaction_value 0
avg_frequency_login_days 0
points_in_wallet     3443
used_special_discount 0
offer_application_preference 0
past_complaint      0
complaint_status     0
feedback            0
churn_risk_score      0
age_range           0
dtype: int64
```

```
In [112]: df['preferred_offer_types']
```

```
Out[112]: 0           Gift Vouchers/Coupons
1           Gift Vouchers/Coupons
2           Gift Vouchers/Coupons
3           Gift Vouchers/Coupons
4           Credit/Debit Card Offers
...
36987       Credit/Debit Card Offers
36988       Without Offers
36989       Gift Vouchers/Coupons
36990       Gift Vouchers/Coupons
36991       Gift Vouchers/Coupons
Name: preferred_offer_types, Length: 36992, dtype: object
```

```
In [122]: p_val=[]
for i in df.columns:
    if(i!='preferred_offer_types'):
        p_val.append(chi2_contingency(pd.crosstab(df['preferred_offer_types'],df[i]))[1])
```

```
In [123]: df1=pd.DataFrame(p_val,index=[i for i in df.columns if i!='preferred_offer_types'])
```

```
In [124]: df[df1[df1[0]<0.05].index[:-1]]
```

Out[124]:

	membership_category	feedback
0	0.000000	Products always in Stock
1	0.000000	Quality Customer Care
2	0.970619	Poor Website
3	0.970619	Poor Website
4	0.970619	Poor Website
...
36987	0.967504	No reason specified
36988	0.967504	Poor Customer Service
36989	0.967504	Poor Website
36990	0.000000	No reason specified
36991	0.427522	Quality Customer Care

36992 rows × 2 columns

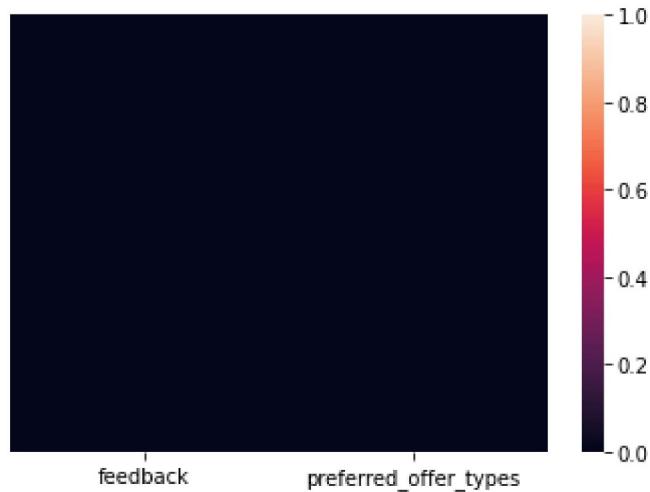
```
In [126]: df1[df1[0]<0.05]
```

Out[126]:

	0
membership_category	8.457888e-09
feedback	9.116817e-60
churn_risk_score	3.684451e-16

```
In [131]: sns.heatmap(df[['feedback','preferred_offer_types']].isna(),yticklabels=False)
```

```
Out[131]: <AxesSubplot:>
```



```
In [133]: df.groupby(['feedback'])['preferred_offer_types'].value_counts()
```

```
Out[133]: feedback      preferred_offer_types
No reason specified    Without Offers          2225
                           Credit/Debit Card Offers  2039
                           Gift Vouchers/Coupons   1970
Poor Customer Service   Without Offers          2143
                           Credit/Debit Card Offers  2043
                           Gift Vouchers/Coupons   2009
Poor Product Quality    Without Offers          2156
                           Credit/Debit Card Offers  2111
                           Gift Vouchers/Coupons   2037
Poor Website             Without Offers          2096
                           Credit/Debit Card Offers  2093
                           Gift Vouchers/Coupons   2037
Products always in Stock  Gift Vouchers/Coupons   567
                           Credit/Debit Card Offers  494
                           Without Offers          310
Quality Customer Care    Gift Vouchers/Coupons   551
                           Credit/Debit Card Offers  466
                           Without Offers          337
Reasonable Price          Gift Vouchers/Coupons   614
                           Credit/Debit Card Offers  497
                           Without Offers          297
Too many ads              Without Offers          2173
                           Credit/Debit Card Offers  2068
                           Gift Vouchers/Coupons   1989
User Friendly Website     Gift Vouchers/Coupons   575
                           Credit/Debit Card Offers  460
                           Without Offers          347
Name: preferred_offer_types, dtype: int64
```

```
In [134]: null=df[df['preferred_offer_types'].isna()]
```

```
In [137]: null['preferred_offer_types']=np.where((null['feedback']=='Too many ads') | (null
```

```
C:\Users\sanja\AppData\Local\Temp\ipykernel_37204\1847588174.py:1: SettingWithCopyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame.
```

```
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
null['preferred_offer_types']=np.where((null['feedback']=='Too many ads') |  
(null['feedback']=='Poor Product Quality') | (null['feedback']=='No reason specified'), 'Without Offers', 'Gift Vouchers/Coupons')
```

```
In [139]: df.update(null)
```

```
In [140]: chi2_contingency(pd.crosstab(df['preferred_offer_types'],df['churn_risk_score']))
```

```
Out[140]: (78.01867326446548,
 1.1440904433054263e-17,
 2,
 array([[5633.98897059, 6640.01102941],
 [5731.3008218 , 6754.6991782 ],
 [5614.71020761, 6617.28979239]]))
```

```
In [144]: df['preferred_offer_types'].replace(df.groupby(['preferred_offer_types'])['churn_
```

```
In [145]: correlation(df)
```

```
Out[145]: age           0.006102
gender         0.005164
region_category 0.017654
membership_category 0.755832
joined_through_referral 0.033316
preferred_offer_types 0.045742
days_since_last_login 0.012238
avg_time_spent      -0.020681
avg_transaction_value -0.150094
points_in_wallet     -0.556742
churn_risk_score      1.000000
age_range          0.006742
Name: churn_risk_score, dtype: float64
```

```
In [149]: df['medium_of_operation'].unique()
```

```
Out[149]: array(['?', 'Desktop', 'Smartphone', 'Both'], dtype=object)
```

```
In [150]: chi2_contingency(pd.crosstab(df['medium_of_operation'],df['churn_risk_score']))
```

```
Out[150]: (17.880759777403362,
 0.00046547097969026777,
 3,
 array([[2475.48496972, 2917.51503028],
 [1748.8592128 , 2061.1407872 ],
 [6386.31974481, 7526.68025519],
 [6369.33607266, 7506.66392734]]))
```

```
In [151]: df['medium_of_operation'].replace(df.groupby(['medium_of_operation'])['churn_risk_
```

```
In [153]: chi2_contingency(pd.crosstab(df['internet_option'],df['churn_risk_score']))
```

```
Out[153]: (2.225230877844357,
 0.3286981459366513,
 2,
 array([[5616.54628028, 6619.45371972],
 [5665.66122405, 6677.33877595],
 [5697.79249567, 6715.20750433]]))
```

```
In [154]: df['internet_option'].replace(df.groupby(['internet_option'])['churn_risk_score']
```

```
In [155]: correlation(df)
```

```
Out[155]: age           0.006102  
gender          0.005164  
region_category 0.017654  
membership_category 0.755832  
joined_through_referral 0.033316  
preferred_offer_types 0.045742  
medium_of_operation 0.021657  
internet_option    0.007498  
days_since_last_login 0.012238  
avg_time_spent     -0.020681  
avg_transaction_value -0.150094  
points_in_wallet    -0.556742  
churn_risk_score     1.000000  
age_range          0.006742  
Name: churn_risk_score, dtype: float64
```

```
In [157]: df.columns
```

```
Out[157]: Index(['age', 'gender', 'region_category', 'membership_category',  
                 'joined_through_referral', 'preferred_offer_types',  
                 'medium_of_operation', 'internet_option', 'last_visit_time',  
                 'days_since_last_login', 'avg_time_spent', 'avg_transaction_value',  
                 'avg_frequency_login_days', 'points_in_wallet', 'used_special_discount',  
                 'offer_application_preference', 'past_complaint', 'complaint_status',  
                 'feedback', 'churn_risk_score', 'age_range'],  
                 dtype='object')
```

```
In [159]: df.drop(columns=['last_visit_time'], inplace=True)
```

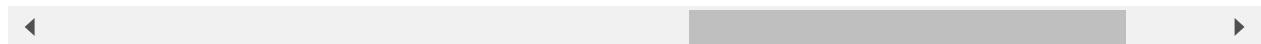
```
In [166]: df['days_since_last_login'].describe()
```

```
Out[166]: count    36992.000000  
mean      -41.915576  
std       228.819900  
min      -999.000000  
25%        8.000000  
50%       12.000000  
75%       16.000000  
max       26.000000  
Name: days_since_last_login, dtype: float64
```

```
In [168]: df[df['days_since_last_login']<0].describe()
```

Out[168]:

net_option	days_since_last_login	avg_time_spent	avg_transaction_value	points_in_wallet	churn_risk
999.000000	1999.0	1999.000000	1999.000000	1832.000000	1999.
0.540920	-999.0	243.945305	28919.117614	679.628380	0.
0.003871	0.0	383.407799	19500.144677	196.728108	0
0.535729	-999.0	-2034.801880	840.340000	-301.998780	0.
0.535729	-999.0	60.570000	13749.385000	612.060000	0.
0.542413	-999.0	166.170000	27170.460000	689.020000	1.
0.544868	-999.0	347.800000	40378.940000	759.475000	1.
0.544868	-999.0	2341.451934	99696.230000	1755.094693	1.



```
In [170]: df['days_since_last_login'].replace({-999:0},inplace=True)
```

```
In [171]: df['days_since_last_login'].describe()
```

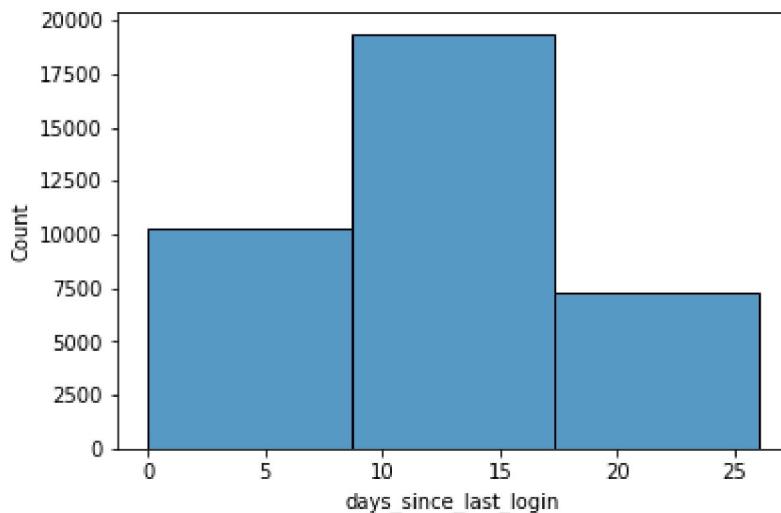
```
Out[171]: count    36992.000000
mean      12.069096
std       6.139792
min      0.000000
25%      8.000000
50%      12.000000
75%      16.000000
max      26.000000
Name: days_since_last_login, dtype: float64
```

```
In [172]: correlation(df)
```

```
Out[172]: age                  0.006102
gender                0.005164
region_category        0.017654
membership_category   0.755832
joined_through_referral 0.033316
preferred_offer_types 0.045742
medium_of_operation    0.021657
internet_option        0.007498
days_since_last_login   0.012238
avg_time_spent         -0.020681
avg_transaction_value -0.150094
points_in_wallet        -0.556742
churn_risk_score        1.000000
age_range               0.006742
Name: churn_risk_score, dtype: float64
```

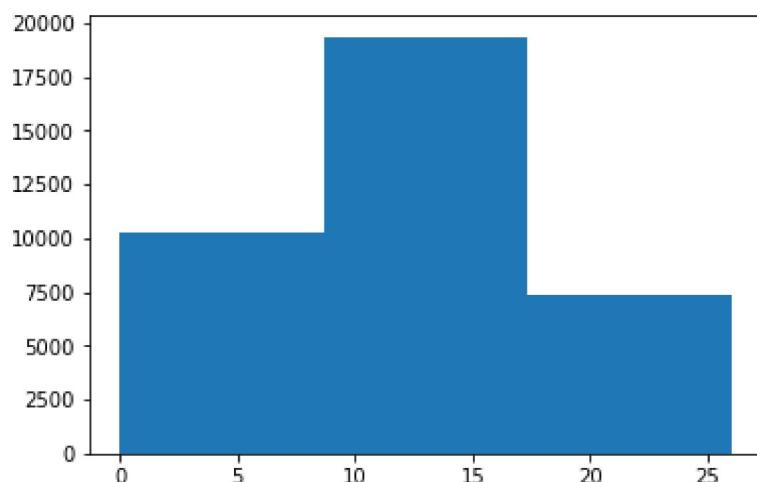
```
In [174]: sns.histplot(df['days_since_last_login'],bins=3)
```

```
Out[174]: <AxesSubplot:xlabel='days_since_last_login', ylabel='Count'>
```



```
In [176]: plt.hist(df['days_since_last_login'],bins=3)[1]
```

```
Out[176]: array([ 0.          ,  8.66666667, 17.33333333, 26.        ])
```



```
In [177]: def make_bins(df,col):
    ctpt=[-1          ,  8.66666667, 17.33333333, 26.        ]
    labels=['freq','less freq','very less']
    df[col+'_range']=pd.cut(df[col],ctpt,labels=labels)
    return df
```

```
In [180]: df=make_bins(df,'days_since_last_login')
chi2_contingency(pd.crosstab(df['days_since_last_login_range'],df['churn_risk_soc
```

```
Out[180]: (6.459091863707832,
0.039575464667095224,
2,
array([[ 4725.13300173,  5568.86699827],
[ 8890.722859 , 10478.277141 ],
[ 3364.14413927,  3964.85586073]]))
```

```
In [183]: df['days_since_last_login_range']=df['days_since_last_login_range'].replace({'fre
```

```
In [184]: correlation(df)
```

```
Out[184]: age                      0.006102
gender                     0.005164
region_category            0.017654
membership_category        0.755832
joined_through_referral    0.033316
preferred_offer_types      0.045742
medium_of_operation        0.021657
internet_option            0.007498
days_since_last_login       0.012238
avg_time_spent             -0.020681
avg_transaction_value     -0.150094
points_in_wallet            -0.556742
churn_risk_score           1.000000
age_range                   0.006742
days_since_last_login_range 0.012932
Name: churn_risk_score, dtype: float64
```

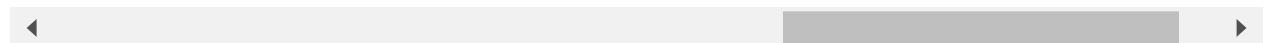
```
In [186]: df.columns
```

```
Out[186]: Index(['age', 'gender', 'region_category', 'membership_category',
'joined_through_referral', 'preferred_offer_types',
'medium_of_operation', 'internet_option', 'days_since_last_login',
'avg_time_spent', 'avg_transaction_value', 'avg_frequency_login_days',
'points_in_wallet', 'used_special_discount',
'offer_application_preference', 'past_complaint', 'complaint_status',
'feedback', 'churn_risk_score', 'age_range',
'days_since_last_login_range'],
dtype='object')
```

```
In [190]: df[df['avg_time_spent']<0].describe()
```

Out[190]:

time_spent	avg_transaction_value	points_in_wallet	churn_risk_score	age_range	days_since_last_lo
-19.000000	1719.000000	1568.000000	1719.000000	1719.000000	17
-67.710232	29076.914444	684.302135	0.560209	0.979058	
-283.577884	19101.741521	203.000076	0.496506	0.827090	
-314.109110	829.410000	-315.897438	0.000000	0.000000	
-324.388276	13857.330000	616.630000	0.000000	0.000000	
-778.915502	27707.280000	690.870000	1.000000	1.000000	
-331.340527	41248.575000	765.657500	1.000000	2.000000	
-12.338460	99353.370000	1637.676500	1.000000	2.000000	



```
In [197]: df[df['avg_time_spent']<0]['churn_risk_score'].mean()
```

Out[197]: 0.5602094240837696

```
In [198]: df[df['avg_time_spent']>0]['churn_risk_score'].mean()
```

Out[198]: 0.5400447934680918

```
In [204]: df['avg_time_spent_range']=np.where(df['avg_time_spent']<0,0.5602094240837696,0.5400447934680918)
```

```
In [205]: chi2_contingency(pd.crosstab(df['avg_time_spent_range'],df['churn_risk_score']))
```

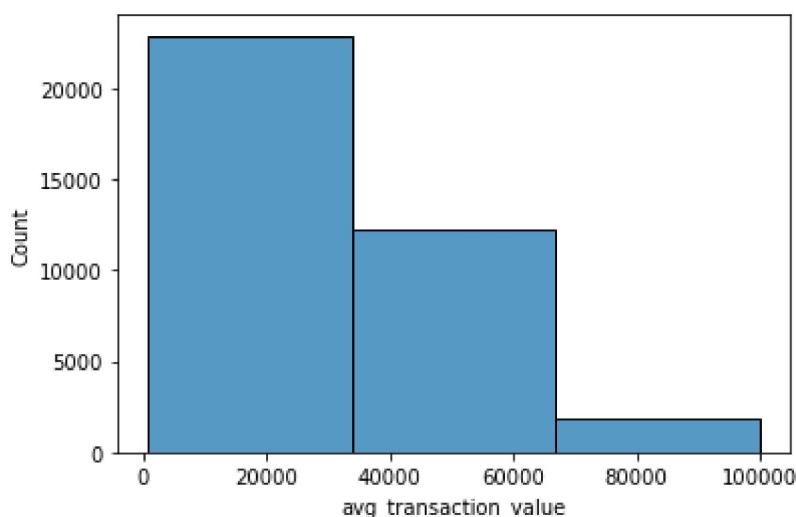
Out[205]: (2.6033850486410643,
 0.1066357349898201,
 1,
 array([[16190.94777249, 19082.05222751],
 [789.05222751, 929.94777249]]))

```
In [207]: correlation(df)
```

```
Out[207]: age                      0.006102
gender                   0.005164
region_category          0.017654
membership_category       0.755832
joined_through_referral   0.033316
preferred_offer_types     0.045742
medium_of_operation        0.021657
internet_option           0.007498
days_since_last_login      0.012238
avg_time_spent            -0.020681
avg_transaction_value     -0.150094
points_in_wallet           -0.556742
churn_risk_score           1.000000
age_range                  0.006742
days_since_last_login_range 0.012932
avg_time_spent_range       0.008518
Name: churn_risk_score, dtype: float64
```

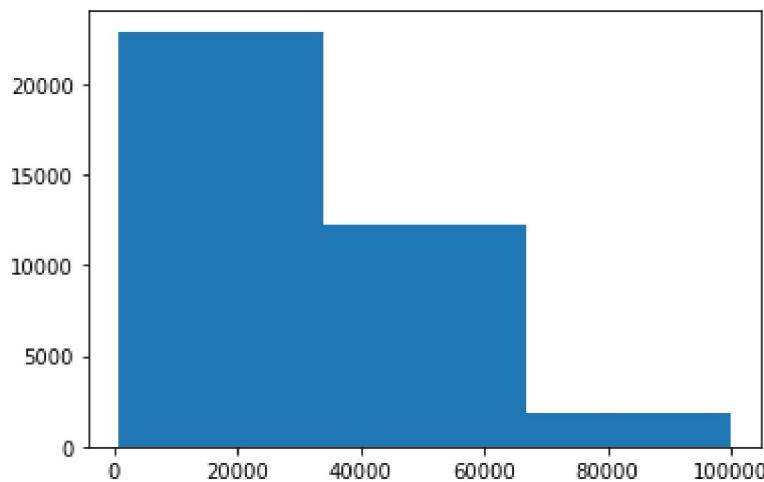
```
In [209]: sns.histplot(df['avg_transaction_value'], bins=3)
```

```
Out[209]: <AxesSubplot:xlabel='avg_transaction_value', ylabel='Count'>
```



```
In [210]: plt.hist(df['avg_transaction_value'], bins=3)[1]
```

```
Out[210]: array([ 800.46 , 33838.3233333, 66876.18666667, 99914.05 ])
```



```
In [211]: def make_bins(df,col):
    ctpt=[ 800 , 33838.3233333, 66876.18666667, 99914.05 ]
    labels=['low','medium','high']
    df[col+'_range']=pd.cut(df[col],ctpt,labels=labels)
    return df
```

```
In [216]: df=make_bins(df,'avg_transaction_value')
```

```
In [220]: df['avg_transaction_value_range']=df['avg_transaction_value_range'].fillna(value=
```

```
In [222]: chi2_contingency(pd.crosstab(df['avg_transaction_value_range'],df['churn_risk_sc
```

```
Out[222]: (2419.7094657724247,
0.0,
2,
array([[10509.6799308 , 12386.3200692 ],
[ 5606.90689879, 6608.09310121],
[ 863.41317042, 1017.58682958]]))
```

```
In [237]: df['avg_frequency_login_days']=df['avg_frequency_login_days'].replace({'Error':0})
```

```
In [238]: df['avg_frequency_login_days'].describe()
```

```
Out[238]: count    36992.000000
mean      14.455576
std       9.941579
min     -43.652702
25%      7.000000
50%      14.000000
75%      22.000000
max     73.061995
Name: avg_frequency_login_days, dtype: float64
```

```
In [240]: df[df['avg_frequency_login_days']<0]['churn_risk_score'].mean()
```

```
Out[240]: 0.5314787701317716
```

```
In [241]: df[df['avg_frequency_login_days']>0]['churn_risk_score'].mean()
```

```
Out[241]: 0.5409156067953762
```

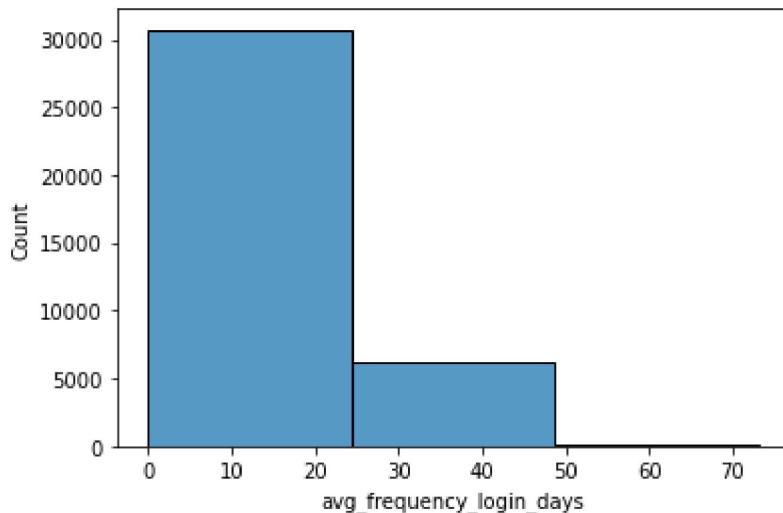
```
In [246]: correlation(df)
```

```
Out[246]: age                  0.006102
gender                0.005164
region_category        0.017654
membership_category    0.755832
joined_through_referral 0.033316
preferred_offer_types  0.045742
medium_of_operation    0.021657
internet_option        0.007498
days_since_last_login   0.012238
avg_time_spent         -0.020681
avg_transaction_value -0.150094
avg_frequency_login_days 0.118436
points_in_wallet        -0.556742
churn_risk_score        1.000000
age_range               0.006742
days_since_last_login_range 0.012932
avg_time_spent_range    0.008518
avg_transaction_value_range -0.155855
Name: churn_risk_score, dtype: float64
```

```
In [245]: df['avg_frequency_login_days']=np.where(df['avg_frequency_login_days']<0,0,df['av
```

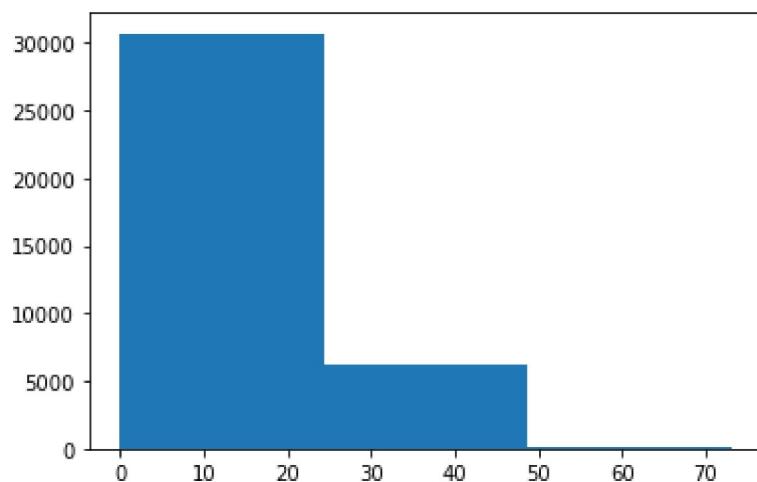
```
In [248]: sns.histplot(df['avg_frequency_login_days'],bins=3)
```

```
Out[248]: <AxesSubplot:xlabel='avg_frequency_login_days', ylabel='Count'>
```



```
In [249]: plt.hist(df['avg_frequency_login_days'],bins=3)[1]
```

```
Out[249]: array([ 0.          , 24.3539982 , 48.7079964 , 73.06199459])
```



```
In [250]: def make_bins(df,col):
    ctpt=[ -1      , 24.3539982 , 48.7079964 , 73.06199459]
    labels=['low','mid','high']
    df[col+'_range']=pd.cut(df[col],ctpt,labels=labels)
    return df
```

```
In [253]: df=make_bins(df,'avg_frequency_login_days')
```

```
In [254]: chi2_contingency(pd.crosstab(df['avg_frequency_login_days_range'],df['churn_risk_
```

```
Out[254]: (259.7704014920197,
            3.9045773900269883e-57,
            2,
            array([[14092.69768322, 16608.30231678],
                   [ 2826.2512503 , 3330.7487497 ],
                   [ 61.05106648, 71.94893352]]))
```

```
In [257]: df['avg_frequency_login_days_range']=df['avg_frequency_login_days_range'].fillna(
```

```
In [258]: correlation(df)
```

```
Out[258]: age                      0.006102
           gender                  0.005164
           region_category          0.017654
           membership_category      0.755832
           joined_through_referral  0.033316
           preferred_offer_types    0.045742
           medium_of_operation       0.021657
           internet_option          0.007498
           days_since_last_login     0.012238
           avg_time_spent            -0.020681
           avg_transaction_value     -0.150094
           avg_frequency_login_days   0.118436
           points_in_wallet          -0.556742
           churn_risk_score          1.000000
           age_range                 0.006742
           days_since_last_login_range 0.012932
           avg_time_spent_range      0.008518
           avg_transaction_value_range -0.155855
           avg_frequency_login_days_range 0.083822
Name: churn_risk_score, dtype: float64
```

```
In [259]: df.isna().sum()
```

```
Out[259]: age                      0  
gender                     0  
region_category            0  
membership_category         0  
joined_through_referral    0  
preferred_offer_types      0  
medium_of_operation        0  
internet_option            0  
days_since_last_login      0  
avg_time_spent             0  
avg_transaction_value      0  
avg_frequency_login_days   0  
points_in_wallet           3443  
used_special_discount       0  
offer_application_preference 0  
past_complaint              0  
complaint_status            0  
feedback                     0  
churn_risk_score            0  
age_range                    0  
days_since_last_login_range 0  
avg_time_spent_range        0  
avg_transaction_value_range 0  
avg_frequency_login_days_range 0  
dtype: int64
```

```
In [263]: df['points_in_wallet'].fillna(value=0,inplace=True)
```

```
In [262]: df['points_in_wallet'].describe()
```

```
Out[262]: count    33549.000000  
mean     686.882199  
std      194.063624  
min     -760.661236  
25%     616.150000  
50%     697.620000  
75%     763.950000  
max     2069.069761  
Name: points_in_wallet, dtype: float64
```

```
In [264]: df[df['points_in_wallet']<0]['churn_risk_score'].mean()
```

```
Out[264]: 0.6838235294117647
```

```
In [265]: df[df['points_in_wallet']>0]['churn_risk_score'].mean()
```

```
Out[265]: 0.5401789722563074
```

```
In [267]: df['points_in_wallet_range']=np.where(df['points_in_wallet']<0,1,0)
```

```
In [268]: correlation(df)
```

```
Out[268]: age                      0.006102
gender                   0.005164
region_category          0.017654
membership_category       0.755832
joined_through_referral   0.033316
preferred_offer_types     0.045742
medium_of_operation        0.021657
internet_option           0.007498
days_since_last_login      0.012238
avg_time_spent            -0.020681
avg_transaction_value     -0.150094
avg_frequency_login_days   0.118436
points_in_wallet           -0.459160
churn_risk_score           1.000000
age_range                  0.006742
days_since_last_login_range 0.012932
avg_time_spent_range       0.008518
avg_transaction_value_range -0.155855
avg_frequency_login_days_range 0.083822
points_in_wallet_range      0.017413
Name: churn_risk_score, dtype: float64
```

```
In [269]: chi2_contingency(pd.crosstab(df['points_in_wallet_range'],df['churn_risk_score']))
```

```
Out[269]: (10.646023435708432,
0.0011030782371494355,
1,
array([[16917.57352941, 19938.42647059],
[ 62.42647059,  73.57352941]]))
```

```
In [271]: df.columns
```

```
Out[271]: Index(['age', 'gender', 'region_category', 'membership_category',
'joined_through_referral', 'preferred_offer_types',
'medium_of_operation', 'internet_option', 'days_since_last_login',
'avg_time_spent', 'avg_transaction_value', 'avg_frequency_login_days',
'points_in_wallet', 'used_special_discount',
'offer_application_preference', 'past_complaint', 'complaint_status',
'feedback', 'churn_risk_score', 'age_range',
'days_since_last_login_range', 'avg_time_spent_range',
'avg_transaction_value_range', 'avg_frequency_login_days_range',
'points_in_wallet_range'],
dtype='object')
```

```
In [278]: df['past_complaint'].replace({'Yes':1,'No':0},inplace=True)
```

```
In [282]: df['complaint_status'].replace(df.groupby(['complaint_status'])['churn_risk_score'])
```

```
In [283]: correlation(df)
```

```
...
```

```
In [284]: chi2_contingency(pd.crosstab(df['complaint_status'],df['churn_risk_score']))
```

```
Out[284]: (5.067955657603337,  
 0.2803956244521792,  
 4,  
 array([[ 2131.68036332,  2512.31963668],  
       [ 8538.65592561, 10063.34407439],  
       [ 2120.20490917, 2498.79509083],  
       [ 2088.53265571, 2461.46734429],  
       [ 2100.92614619, 2476.07385381]]))
```

```
In [286]: df['feedback'].unique()
```

```
Out[286]: array(['Products always in Stock', 'Quality Customer Care',  
 'Poor Website', 'No reason specified', 'Poor Product Quality',  
 'Poor Customer Service', 'Too many ads', 'User Friendly Website',  
 'Reasonable Price'], dtype=object)
```

```
In [289]: df['feedback'].replace(df.groupby(['feedback'])['churn_risk_score'].mean().to_dict(), inplace=True)
```

```
In [291]: chi2_contingency(pd.crosstab(df['feedback'],df['churn_risk_score']))
```

```
Out[291]: (7703.219643595683,  
 0.0,  
 5,  
 array([[2547.5508218 , 3002.4491782 ],  
       [2882.17506488, 3396.82493512],  
       [2878.50291955, 3392.49708045],  
       [2887.22426471, 3402.77573529],  
       [2869.78157439, 3382.21842561],  
       [2914.76535467, 3435.23464533]]))
```

```
In [292]: df.info()
```

```
0    internet_option      36992 non-null   float64
1    days_since_last_login 36992 non-null   float64
2    avg_time_spent        36992 non-null   float64
3    avg_transaction_value 36992 non-null   float64
4    avg_frequency_login_days 36992 non-null   float64
5    points_in_wallet      36992 non-null   float64
6    used_special_discount 36992 non-null   int64
7    offer_application_preference 36992 non-null   int64
8    past_complaint         36992 non-null   int64
9    complaint_status       36992 non-null   float64
10   feedback               36992 non-null   float64
11   churn_risk_score       36992 non-null   float64
12   age_range               36992 non-null   float64
13   days_since_last_login_range 36992 non-null   int32
14   avg_time_spent_range   36992 non-null   float64
15   avg_transaction_value_range 36992 non-null   int32
16   avg_frequency_login_days_range 36992 non-null   int32
17   points_in_wallet_range 36992 non-null   int32
dtypes: float64(18), int32(4), int64(3)
memory usage: 6.5 MB
```

```
In [293]: from sklearn.tree import DecisionTreeClassifier
```

```
In [294]: from sklearn.model_selection import train_test_split, GridSearchCV, cross_val_score
X_train, X_test, y_train, y_test = train_test_split(df.drop(columns=['churn_risk_score']),
                                                    test_size=0.2, random_state=42, stratify=df['churn_risk_score'])
```

```
In [295]: from sklearn.feature_selection import VarianceThreshold
var = VarianceThreshold()
```

```
In [296]: var.fit(X_train, y_train)
```

```
Out[296]: VarianceThreshold()
          VarianceThreshold()
```

```
In [297]: var.get_support()
```

```
Out[297]: array([ True,  True,  True,  True,  True,  True,  True,  True,  True,  True,
       True,  True,  True,  True,  True,  True,  True,  True,  True])
```

```
In [298]: var = VarianceThreshold(threshold=0.01)
```

```
In [300]: var.fit(X_train, y_train)
rem_col = X_train.columns[~var.get_support()]
```

```
In [301]: X_train.drop(columns=rem_col,inplace=True)
X_test.drop(columns=rem_col,inplace=True)
```

```
In [303]: X_train.shape,X_test.shape
```

```
Out[303]: ((29593, 15), (7399, 15))
```

```
In [304]: X_train.T.duplicated()
```

```
Out[304]: age                         False
membership_category                 False
days_since_last_login               False
avg_time_spent                     False
avg_transaction_value              False
avg_frequency_login_days           False
points_in_wallet                   False
used_special_discount              False
offer_application_preference       False
past_complaint                     False
feedback                          False
age_range                          False
days_since_last_login_range        False
avg_transaction_value_range        False
avg_frequency_login_days_range     False
dtype: bool
```

```
In [305]: X_train.shape
```

```
Out[305]: (29593, 15)
```

```
In [306]: def correlation(df,thres):
            corr=df.corr(method='spearman')
            rem_col=set()
            for i in range(len(corr.columns)):
                for j in range(i):
                    if(abs(corr.iloc[i,j])>thres):
                        rem_col.add(corr.columns[i])

            return rem_col
```

```
In [308]: rem_col=correlation(X_train,0.85)
```

```
In [309]: X_train.drop(columns=rem_col,inplace=True)
X_test.drop(columns=rem_col,inplace=True)
```

```
In [310]: X_train.shape
```

```
Out[310]: (29593, 12)
```

```
In [311]: check_train=X_train.sample(50)
check_test=X_test.sample(50)
check_train['tag']=1
check_test['tag']=0
check=check_train.append(check_test)
dt=DecisionTreeClassifier(random_state=42)
dt.fit(check.drop(columns=['tag']),check['tag'])
```

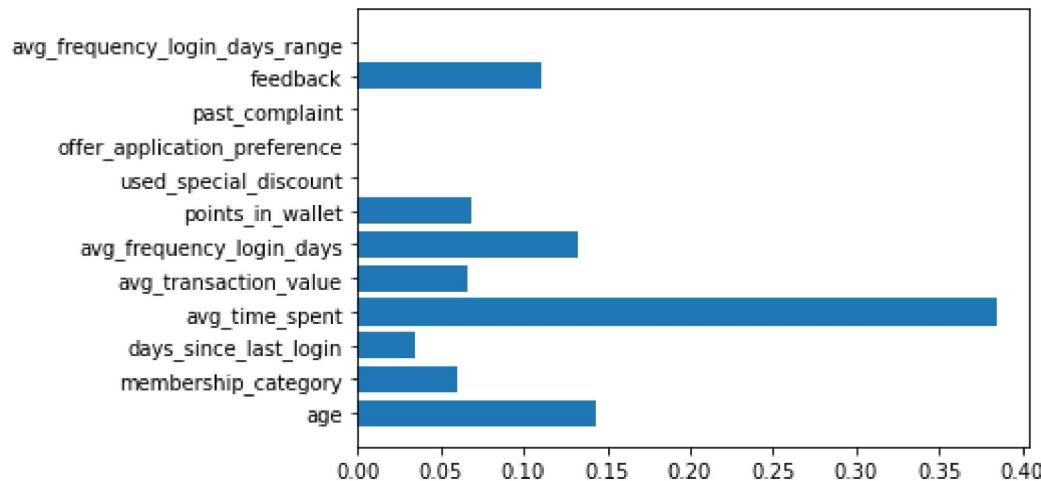
C:\Users\sanja\AppData\Local\Temp\ipykernel_37204\1486015105.py:5: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.
check=check_train.append(check_test)

```
Out[311]:
```

```
▼      DecisionTreeClassifier
DecisionTreeClassifier(random_state=42)
```

```
In [316]: plt.barh(dt.feature_names_in_,dt.feature_importances_)
```

```
Out[316]: <BarContainer object of 12 artists>
```



```
In [317]: X_train.drop(columns=['avg_time_spent'],inplace=True)
X_test.drop(columns=['avg_time_spent'],inplace=True)
```

```
In [325]: X_train.shape
```

```
Out[325]: (29593, 10)
```

```
In [319]: from sklearn.feature_selection import mutual_info_classif
```

```
In [320]: clss=[]
for i in X_train.columns:
    clss.append(mutual_info_classif(X_train[i].values.reshape(-1,1),y_train)[0])
```

```
In [321]: clss
```

```
Out[321]: [0.002055778815072351,  
 0.39859013891956674,  
 0.001207936160344314,  
 0.06646831801840469,  
 0.029399509037111304,  
 0.3905300243563341,  
 0.0018406771013750323,  
 0,  
 0.006085043372634269,  
 0.12822451823088987,  
 0.007013815787707989]
```

```
In [322]: pd.DataFrame(clss,index=X_train.columns)
```

```
Out[322]:
```

	0
age	0.002056
membership_category	0.398590
days_since_last_login	0.001208
avg_transaction_value	0.066468
avg_frequency_login_days	0.029400
points_in_wallet	0.390530
used_special_discount	0.001841
offer_application_preference	0.000000
past_complaint	0.006085
feedback	0.128225
avg_frequency_login_days_range	0.007014

```
In [323]: rem_col=['offer_application_preference']
```

```
In [324]: X_train.drop(columns=rem_col,inplace=True)  
X_test.drop(columns=rem_col,inplace=True)
```

```
In [326]: from sklearn.feature_selection import RFE
```

```
In [327]: from sklearn.ensemble import RandomForestClassifier
```

```
In [328]: from sklearn.metrics import log_loss
```

```
In [329]: for i in range(1,X_train.shape[1]+1):
    rfe=RFE(RandomForestClassifier(random_state=42),n_features_to_select=i)
    rfe.fit(X_train,y_train)
    X_train_rfe=rfe.transform(X_train)
    X_test_rfe=rfe.transform(X_test)
    dt=DecisionTreeClassifier(random_state=42)
    dt.fit(X_train_rfe,y_train)
    print(i, ' ',log_loss(y_test,dt.predict_proba(X_test_rfe)))
```

```
1  4.351388451758847
2  2.447970642504771
3  3.1695944279151718
4  3.0342214700218872
5  2.8755083469745886
6  3.001545238806268
7  3.006213271837071
8  3.1042419654839315
9  3.220942791254004
10 3.2722911545928355
```

```
In [330]: rfe=RFE(RandomForestClassifier(random_state=42),n_features_to_select=5)
rfe.fit(X_train,y_train)
X_train_rfe=X_train[X_train.columns[rfe.get_support()]]
X_test_rfe=X_test[X_train_rfe.columns]
```

```
In [333]: score=cross_val_score(dt,X_test_rfe,y_test,cv=10,scoring='neg_log_loss')
score.mean()
```

```
Out[333]: -2.9828887827057047
```

```
In [332]: score=cross_val_score(dt,X_train_rfe,y_train,cv=10,scoring='neg_log_loss')
score.mean()
```

```
Out[332]: -3.0240486818175234
```

```
In [336]: met=pd.DataFrame([score.mean(),],columns=['log_loss'],index=['Decision tree'])
```

```
In [337]: met
```

```
Out[337]:
```

	log_loss
Decision tree	-2.982889

```
In [338]: for i in range(1,X_train.shape[1]+1):
    rfe=RFE(RandomForestClassifier(random_state=42),n_features_to_select=i)
    rfe.fit(X_train,y_train)
    X_train_rfe=rfe.transform(X_train)
    X_test_rfe=rfe.transform(X_test)
    dt=RandomForestClassifier(random_state=42)
    dt.fit(X_train_rfe,y_train)
    print(i, ' ',log_loss(y_test,dt.predict_proba(X_test_rfe)))
```

```
1  1.0918310325673983
2  0.6545541219136775
3  0.41610986609502165
4  0.35122373103038845
5  0.28123039226584584
6  0.23491592642213271
7  0.21798737007934865
8  0.2114347695481794
9  0.21497256932176292
10 0.23264873418485496
```

```
In [339]: rfc=RandomForestClassifier(random_state=42)
```

```
In [340]: rfe=RFE(RandomForestClassifier(random_state=42),n_features_to_select=8)
rfe.fit(X_train,y_train)
X_train_rfe=X_train[X_train.columns[rfe.get_support()]]
X_test_rfe=X_test[X_train_rfe.columns]
```

```
In [343]: score=cross_val_score(rfc,X_test_rfe,y_test,cv=10,scoring='neg_log_loss')
score.mean()
```

```
Out[343]: -0.21272487974934737
```

```
In [342]: score=cross_val_score(rfc,X_train_rfe,y_train,cv=10,scoring='neg_log_loss')
score.mean()
```

```
Out[342]: -0.19771783976112797
```

```
In [344]: met.loc['random forest','log_loss']=score.mean()
```

```
In [345]: met
```

```
Out[345]:
```

	log_loss
Decision tree	-2.982889
random forest	-0.212725

```
In [346]: param_rfc={'n_estimators':[int(c) for c in np.linspace(12,125,4)],  
                      'max_depth':[int(x) for x in np.linspace(12,125,5)],  
                      'max_samples':np.linspace(0.5,0.75,3),  
                      'max_features':np.linspace(0.3,0.45,2)}
```

```
In [347]: grid_rfc=GridSearchCV(rfc,param_rfc,cv=5,scoring='neg_log_loss',verbose=3)
```

```
In [348]: grid_rfc.fit(X_train_rfe,y_train)  
  
=12;, score=-0.472 total time= 0.6s  
[CV 5/5] END max_depth=68, max_features=0.45, max_samples=0.625, n_estimators  
=12;, score=-0.500 total time= 0.5s  
[CV 1/5] END max_depth=68, max_features=0.45, max_samples=0.625, n_estimators  
=49;, score=-0.290 total time= 2.3s  
[CV 2/5] END max_depth=68, max_features=0.45, max_samples=0.625, n_estimators  
=49;, score=-0.273 total time= 3.1s  
[CV 3/5] END max_depth=68, max_features=0.45, max_samples=0.625, n_estimators  
=49;, score=-0.247 total time= 2.3s  
[CV 4/5] END max_depth=68, max_features=0.45, max_samples=0.625, n_estimators  
=49;, score=-0.300 total time= 2.3s  
[CV 5/5] END max_depth=68, max_features=0.45, max_samples=0.625, n_estimators  
=49;, score=-0.306 total time= 2.4s  
[CV 1/5] END max_depth=68, max_features=0.45, max_samples=0.625, n_estimators  
=87;, score=-0.239 total time= 4.9s  
[CV 2/5] END max_depth=68, max_features=0.45, max_samples=0.625, n_estimators  
=87;, score=-0.206 total time= 4.8s  
[CV 3/5] END max_depth=68, max_features=0.45, max_samples=0.625, n_estimators  
=87;, score=-0.186 total time= 4.6s  
[CV 4/5] END max_depth=68, max_features=0.45, max_samples=0.625, n_estimators
```

```
In [349]: score=cross_val_score(grid_rfc.best_estimator_,X_test_rfe,y_test,cv=10,scoring='r  
score.mean()
```

```
Out[349]: -0.16226084343665778
```

```
In [350]: score=cross_val_score(grid_rfc.best_estimator_,X_train_rfe,y_train,cv=10,scoring=  
score.mean())
```

```
Out[350]: -0.15264219397130768
```

```
In [351]: met.loc['random forest hyper parameter tuned','log_loss']=-0.16226084343665778
```

```
In [352]: met
```

```
Out[352]:
```

	log_loss
Decision tree	-2.982889
random forest	-0.212725
random forest hyper parameter tuned	-0.162261

In []: