In [32]:
```
#import all required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import plotly.express as px
import plotly.graph_objects as go
from sklearn.naive_bayes import GaussianNB
```

In [2]:
```
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.preprocessing import LabelBinarizer
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.cluster import KMeans
```

In [3]:
```
#read data
df = pd.read_csv("customer")
```

In [4]:
```
df.head()
```

Out[4]:

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_Category	C
0	768805383	Existing Customer	45	M	3	High School	Married	$60K-80K$	
1	818770008	Existing Customer	49	F	5	Graduate	Single	Less than $40K	
2	713982108	Existing Customer	51	M	3	Graduate	Married	$80K-120K$	
3	769911858	Existing Customer	40	F	4	High School	Unknown	Less than $40K	
4	709106358	Existing Customer	40	M	3	Uneducated	Married	$60K-80K$	

5 rows × 23 columns

In [5]:
```
df.tail()
```

Out[5]:

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_Catego
10122	772366833	Existing Customer	50	M	2	Graduate	Single	$40K-6$
10123	710638233	Attrited Customer	41	M	2	Unknown	Divorced	$40K-6$
10124	716506083	Attrited Customer	44	F	1	High School	Married	Less than $4
10125	717406983	Attrited Customer	30	M	2	Graduate	Unknown	$40K-6$
10126	714337233	Attrited Customer	43	F	2	Graduate	Married	Less than $4

5 rows × 23 columns

In [6]:
```
df.shape
```

Out[6]:
```
(10127, 23)
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10127 entries, 0 to 10126
Data columns (total 23 columns):
 #   Column
Non-Null Count  Dtype
---  ------
--------------  -----
 0   CLIENTNUM
10127 non-null  int64
 1   Attrition_Flag
10127 non-null  object
 2   Customer_Age
10127 non-null  int64
 3   Gender
10127 non-null  object
 4   Dependent_count
10127 non-null  int64
 5   Education_Level
10127 non-null  object
 6   Marital_Status
10127 non-null  object
 7   Income_Category
10127 non-null  object
 8   Card_Category
10127 non-null  object
 9   Months_on_book
10127 non-null  int64
 10  Total_Relationship_Count
10127 non-null  int64
 11  Months_Inactive_12_mon
10127 non-null  int64
 12  Contacts_Count_12_mon
10127 non-null  int64
 13  Credit_Limit
10127 non-null  float64
 14  Total_Revolving_Bal
10127 non-null  int64
 15  Avg_Open_To_Buy
10127 non-null  float64
 16  Total_Amt_Chng_Q4_Q1
10127 non-null  float64
 17  Total_Trans_Amt
10127 non-null  int64
 18  Total_Trans_Ct
10127 non-null  int64
 19  Total_Ct_Chng_Q4_Q1
10127 non-null  float64
 20  Avg_Utilization_Ratio
10127 non-null  float64
 21  Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Educa
tion_Level_Months_Inactive_12_mon_1  10127 non-null  float64
 22  Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Educa
tion_Level_Months_Inactive_12_mon_2  10127 non-null  float64
dtypes: float64(7), int64(10), object(6)
memory usage: 1.8+ MB
```

In [8]:
```
df.describe()
```

Out[8]:

	CLIENTNUM	Customer_Age	Dependent_count	Months_on_book	Total_Relationship_Count	Months_Inactive_12_mon
count	1.012700e+04	10127.000000	10127.000000	10127.000000	10127.000000	10127.000000
mean	7.391776e+08	46.325960	2.346203	35.928409	3.812580	2.341167
std	3.690378e+07	8.016814	1.298908	7.986416	1.554408	1.010622
min	7.080821e+08	26.000000	0.000000	13.000000	1.000000	0.000000
25%	7.130368e+08	41.000000	1.000000	31.000000	3.000000	2.000000
50%	7.179264e+08	46.000000	2.000000	36.000000	4.000000	2.000000
75%	7.731435e+08	52.000000	3.000000	40.000000	5.000000	3.000000
max	8.283431e+08	73.000000	5.000000	56.000000	6.000000	6.000000

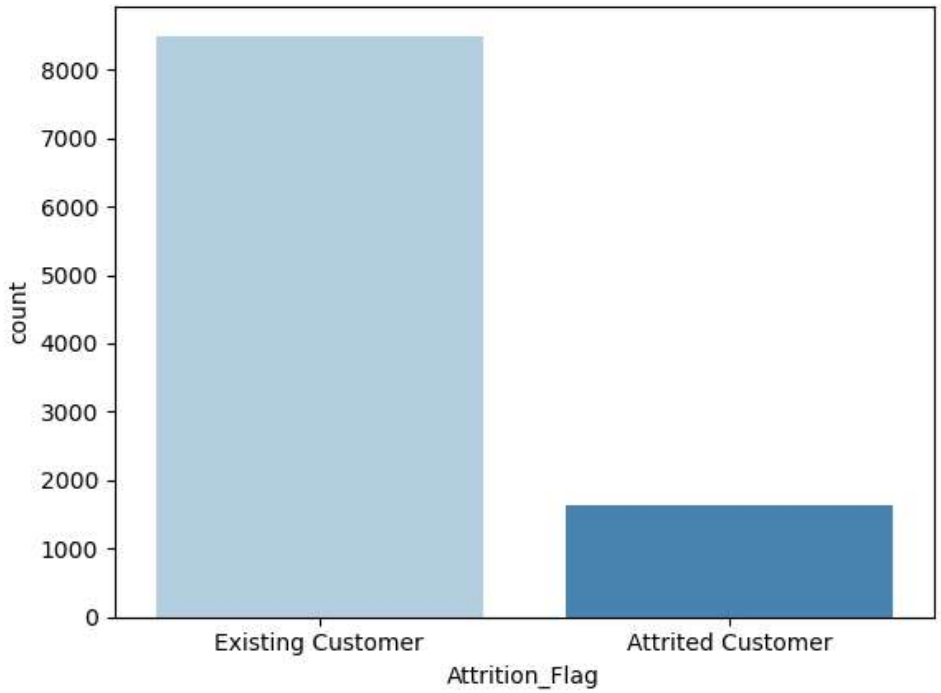
In [9]:
```
df.nunique()
```

Out[9]:
```
CLIENTNUM
10127
Attrition_Flag
2
Customer_Age
45
Gender
2
Dependent_count
6
Education_Level
7
Marital_Status
4
Income_Category
6
Card_Category
4
Months_on_book
44
Total_Relationship_Count
6
Months_Inactive_12_mon
7
Contacts_Count_12_mon
7
Credit_Limit
6205
Total_Revolving_Bal
1974
Avg_Open_To_Buy
6813
Total_Amt_Chng_Q4_Q1
1158
Total_Trans_Amt
5033
Total_Trans_Ct
126
Total_Ct_Chng_Q4_Q1
830
Avg_Utilization_Ratio
964
Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_
Level_Months_Inactive_12_mon_1     1704
Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_
Level_Months_Inactive_12_mon_2      640
dtype: int64
```

EDA

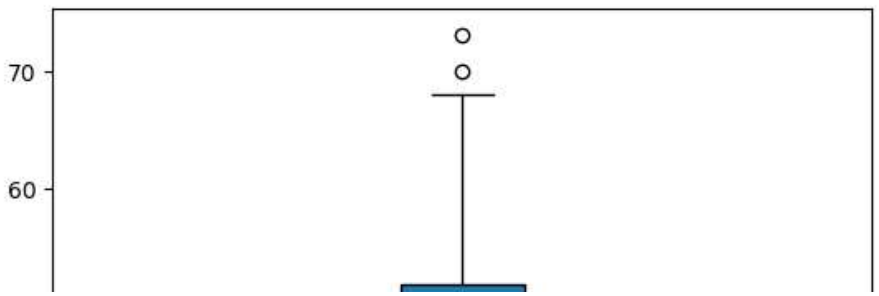
In [10]:
```
sns.countplot(x="Attrition_Flag" , data=df, palette="Blues")
```

Out[10]:
```
<Axes: xlabel='Attrition_Flag', ylabel='count'>
```



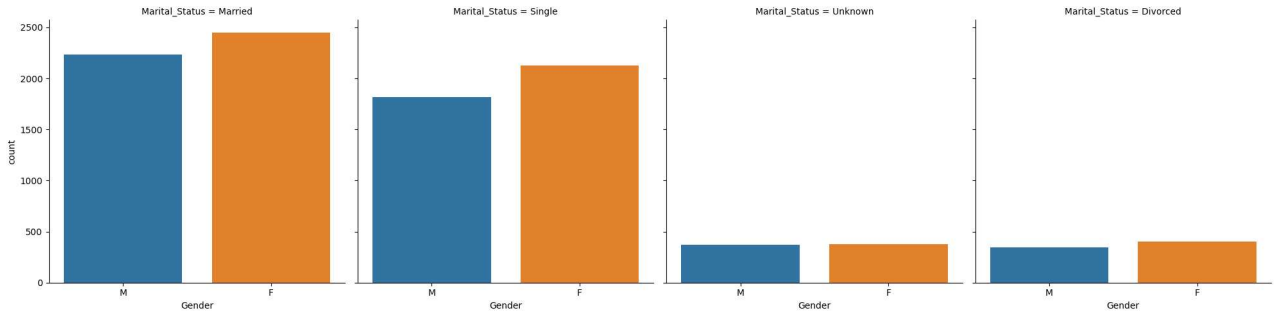
In [11]:
```
#for analyze age of customer
plt.boxplot(df["Customer_Age"], patch_artist=True)
```

Out[11]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x1b60acf8310>,
  <matplotlib.lines.Line2D at 0x1b60acf8e50>],
 'caps': [<matplotlib.lines.Line2D at 0x1b60ac67690>,
  <matplotlib.lines.Line2D at 0x1b60acfa350>],
 'boxes': [<matplotlib.patches.PathPatch at 0x1b60acebdd0>],
 'medians': [<matplotlib.lines.Line2D at 0x1b60acfaed0>],
 'fliers': [<matplotlib.lines.Line2D at 0x1b60acfb9d0>],
 'means': []}
```



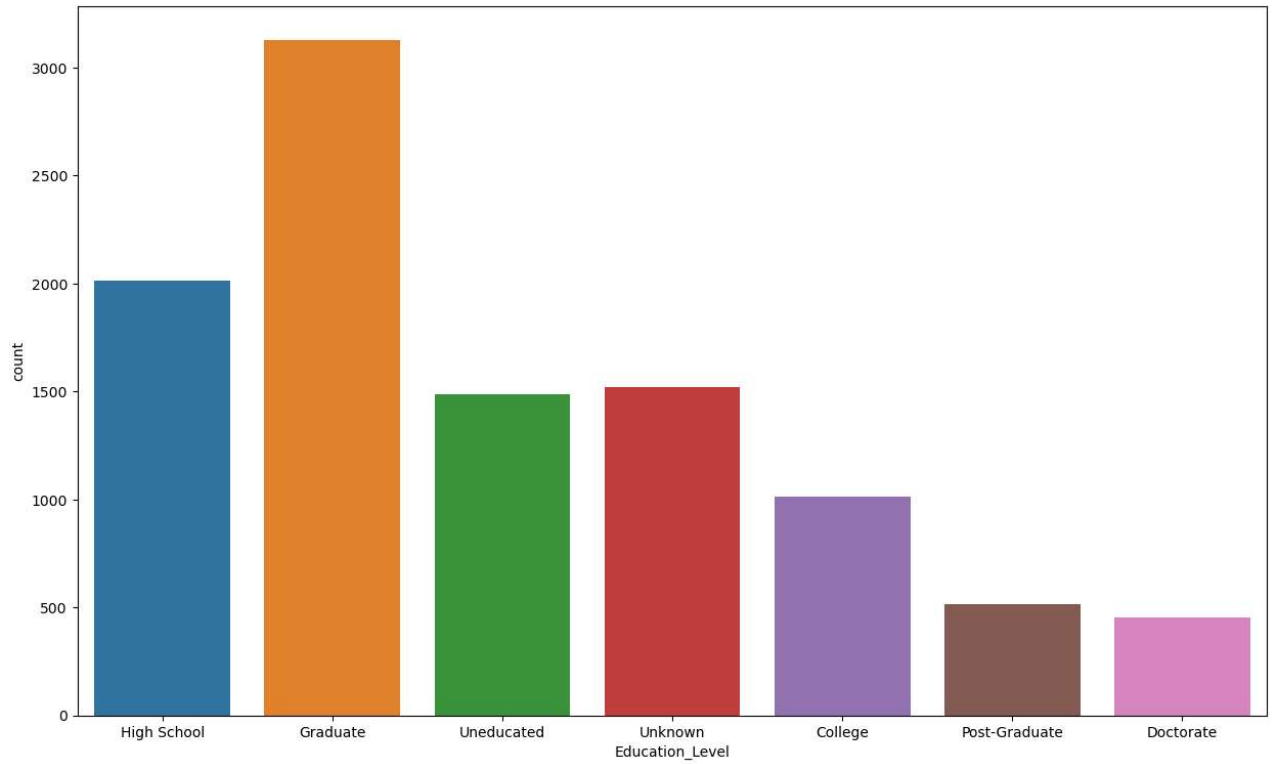
In [12]:
```
#customer gender and merital Status
sns.catplot(x="Gender", data=df, kind="count", col="Marital_Status")
```

Out[12]:
```
<seaborn.axisgrid.FacetGrid at 0x1b60ac67b50>
```



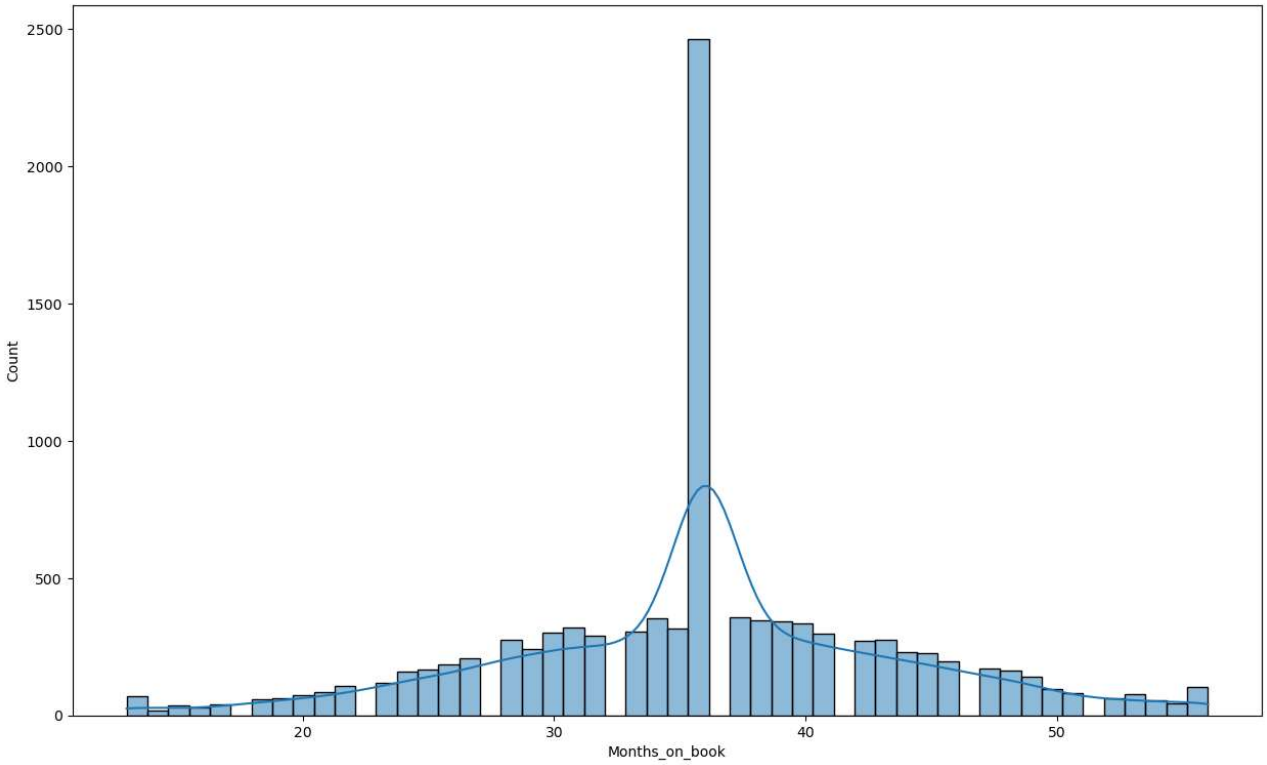
In [13]:
```
#analyze education
plt.figure(figsize=(15,9))
sns.countplot(x="Education_Level", data=df)
```

Out[13]:
```
<Axes: xlabel='Education_Level', ylabel='count'>
```



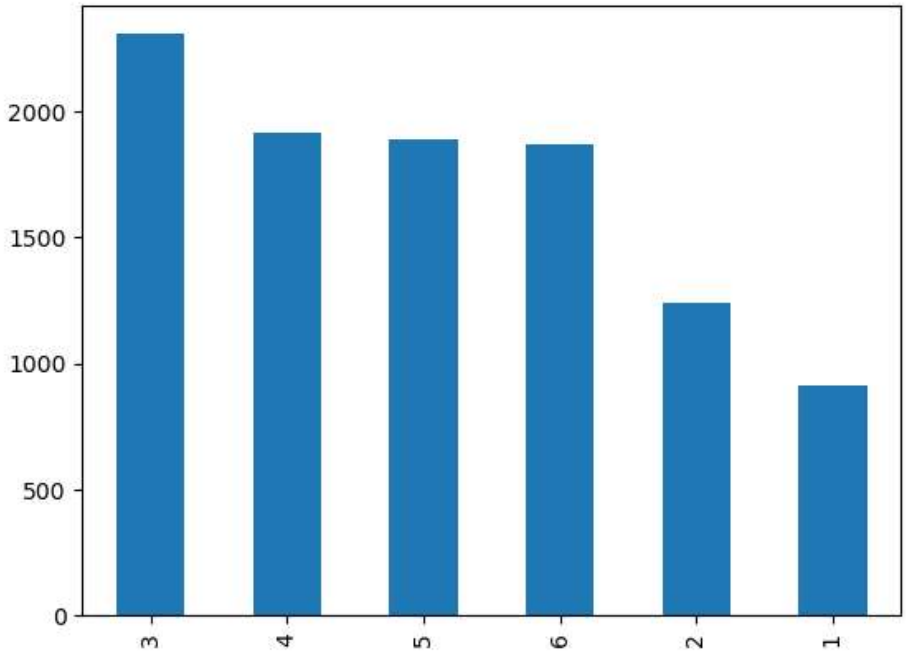
In [14]:
```
#analyze months span
plt.figure(figsize=(15,9))
sns.histplot(x="Months_on_book", data=df, kde=True)
```

Out[14]: <Axes: xlabel='Months_on_book', ylabel='Count'>



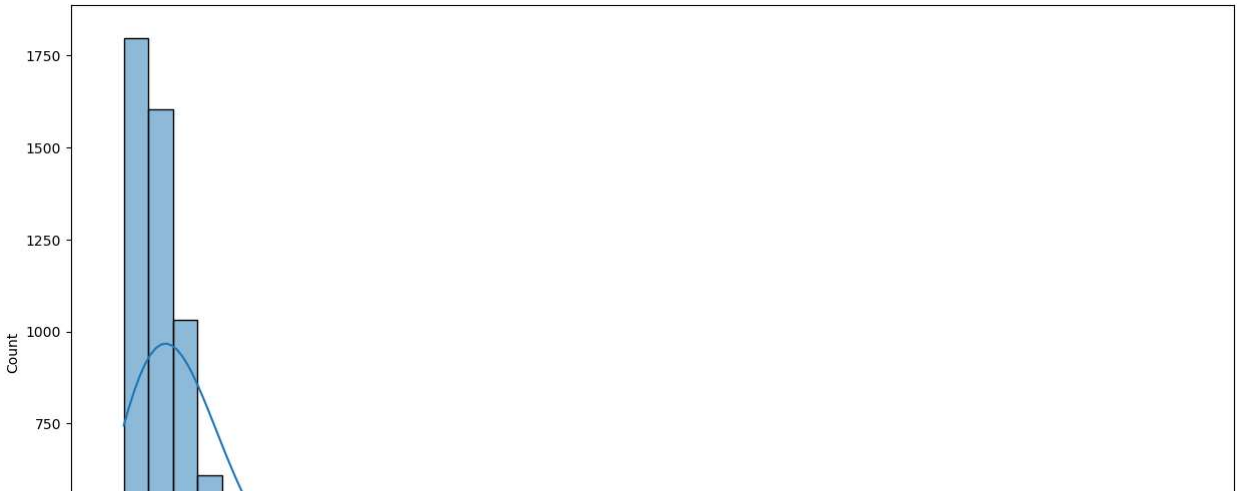
In [15]:
```
df.Total_Relationship_Count.value_counts().plot(kind="bar")
```

Out[15]: <Axes: >



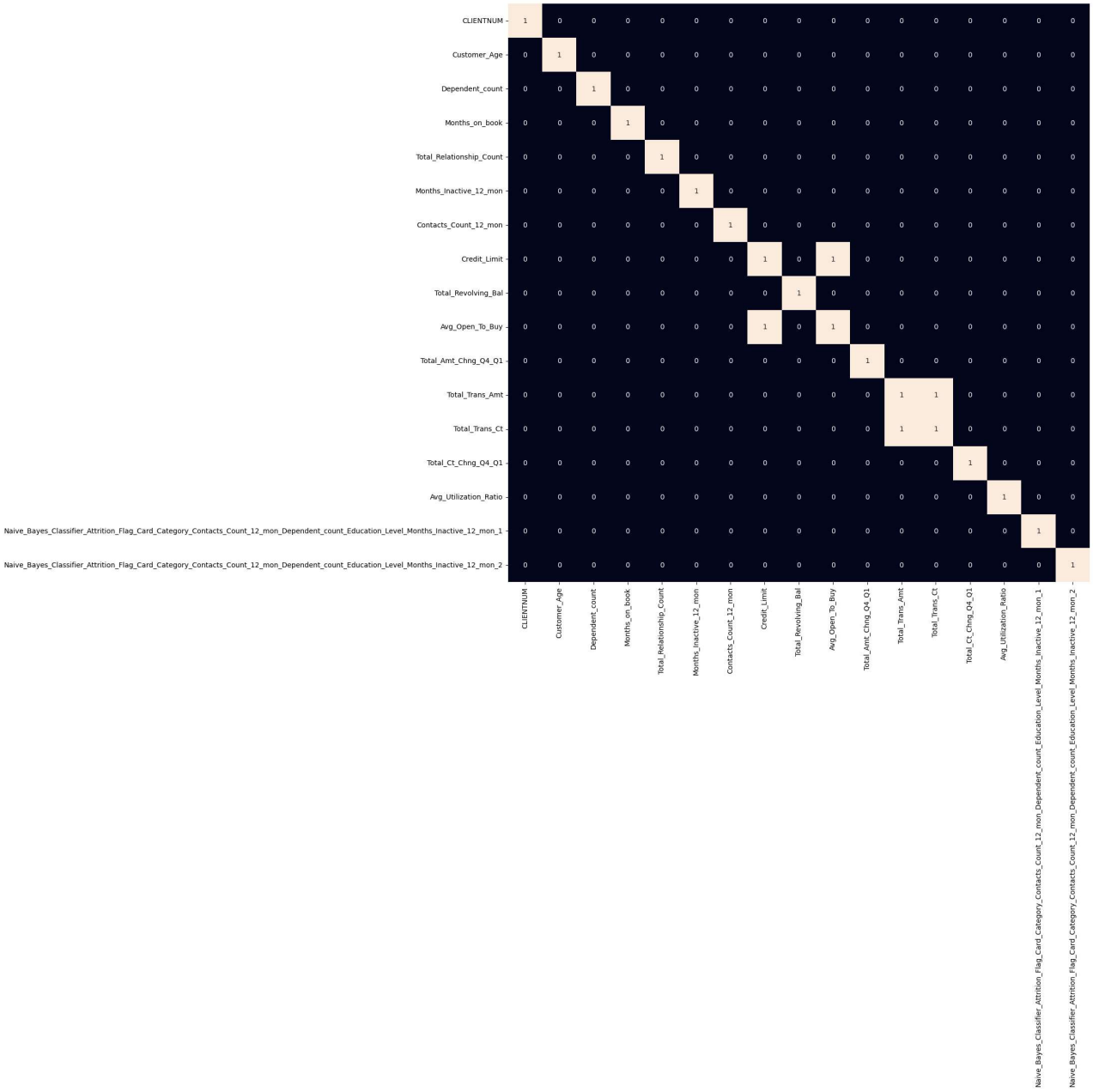
In [16]:
```
#credit limit of customers
plt.figure(figsize=(15,9))
sns.histplot(x="Credit_Limit" , data=df, kde=True)
```

Out[16]: <Axes: xlabel='Credit_Limit', ylabel='Count'>



In [17]:
```
plt.figure(figsize=(15, 15))
sns.heatmap(df.corr() > 0.8, annot=True, cbar=False)
plt.show()
```

```
C:\Users\ABC\AppData\Local\Temp\ipykernel_11140\2690130172.py:2: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select
only valid columns or specify the value of numeric_only to silence this warning.
  sns.heatmap(df.corr() > 0.8, annot=True, cbar=False)
```



In [18]:
```
df.columns
```

Out[18]:
```
Index(['CLIENTNUM', 'Attrition_Flag', 'Customer_Age', 'Gender',
       'Dependent_count', 'Education_Level', 'Marital_Status',
       'Income_Category', 'Card_Category', 'Months_on_book',
       'Total_Relationship_Count', 'Months_Inactive_12_mon',
       'Contacts_Count_12_mon', 'Credit_Limit', 'Total_Revolving_Bal',
       'Avg_Open_To_Buy', 'Total_Amt_Chng_Q4_Q1', 'Total_Trans_Amt',
       'Total_Trans_Ct', 'Total_Ct_Chng_Q4_Q1', 'Avg_Utilization_Ratio',
       'Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Ed
ucation_Level_Months_Inactive_12_mon_1',
       'Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Ed
ucation_Level_Months_Inactive_12_mon_2'],
      dtype='object')
```

In [20]:
```
lis = ["Attrition_Flag", "Gender", "Marital_Status" ,"Education_Level", "Income_Category", "Card_Cate
```

In [21]:
```
le = LabelBinarizer()
```

In [22]:
```
for columns in lis:
    df[columns] = le.fit_transform(df[columns])
```

In [23]:
```
x= df.drop(columns=["Attrition_Flag"])
y = df["Attrition_Flag"]
```

In [24]:
```
x.shape
```

Out[24]:
```
(10127, 22)
```

In [25]:
```
y.shape
```

Out[25]:
```
(10127,)
```

In [26]:
```
x = StandardScaler().fit_transform(x)
```

In [27]:
```
x[:3]
```

Out[27]:
```
array([[ 8.02878101e-01, -1.65405580e-01,  1.05995565e+00,
         5.03368127e-01, -3.33388189e-01, -2.82405097e-01,
        -2.78101466e-01,  2.70610758e-01,  3.84620878e-01,
         7.63942609e-01, -1.32713603e+00,  4.92403766e-01,
         4.46621903e-01, -4.73422218e-01,  4.88970818e-01,
         2.62349444e+00, -9.59706574e-01, -9.73895182e-01,
         3.83400260e+00, -7.75882235e-01, -4.37753814e-01,
         4.37763128e-01],
       [ 2.15686101e+00,  3.33570383e-01, -9.43435701e-01,
         2.04319867e+00, -3.33388189e-01, -2.82405097e-01,
        -2.78101466e-01,  2.70610758e-01,  1.01071482e+00,
         1.40730617e+00, -1.32713603e+00, -4.11615984e-01,
        -4.13666521e-02, -3.66666822e-01, -8.48598788e-03,
         3.56329284e+00, -9.16432607e-01, -1.35734038e+00,
         1.26085729e+01, -6.16275655e-01, -4.37853975e-01,
         4.37845257e-01],
       [-6.82768542e-01,  5.83058365e-01,  1.05995565e+00,
         5.03368127e-01, -3.33388189e-01, -2.82405097e-01,
        -2.78101466e-01,  2.70610758e-01,  8.96451285e-03,
         1.20579050e-01, -1.32713603e+00, -2.21965548e+00,
        -5.73697797e-01, -1.42685834e+00, -4.45658333e-01,
         8.36721381e+00, -7.40981694e-01, -1.91120566e+00,
         6.80786367e+00, -9.97154993e-01, -4.37951926e-01,
         4.37954761e-01]])
```

In [28]:
```
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size=0.30, random_state=42, shuffle=Tru
```

In [29]:
```
x_train.shape, x_test.shape, y_train.shape, y_test.shape
```

Out[29]:
```
((7088, 22), (3039, 22), (7088,), (3039,))
```

In [36]:
```
NV = GaussianNB()
NV.fit(x_train, y_train)
NV.score(x_train, y_train)*100, NV.score(x_test, y_test)*100
```

Out[36]:
```
(100.0, 100.0)
```

In [38]:
```
y_predict = NV.predict(x_test)
y_predict[:10], y_test[:10]
```

Out[38]:
```
(array([1, 1, 1, 1, 1, 1, 1, 1, 1, 0]),
 3781    1
 2922    1
 5070    1
 7246    1
 623     1
 3931    1
 4767    1
 7094    1
 3282    1
 3994    0
 Name: Attrition_Flag, dtype: int32)
```

In [ ]:

In [ ]: