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

df =  pd.read_csv('Customer-Churn-Records.csv')

df.head()
```

```
   RowNumber  CustomerId   Surname  CreditScore Geography  Gender  Age
\
0          1    15634602  Hargrave          619    France  Female   42

1          2    15647311      Hill          608     Spain  Female   41

2          3    15619304      Onio          502    France  Female   42

3          4    15701354      Boni          699    France  Female   39

4          5    15737888  Mitchell          850     Spain  Female   43


   Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \
0       2       0.00              1          1               1
1       1   83807.86              1          0               1
2       8  159660.80              3          1               0
3       1       0.00              2          0               0
4       2  125510.82              1          1               1

   EstimatedSalary  Exited  Complain  Satisfaction Score Card Type  \
0        101348.88       1         1                   2   DIAMOND
1        112542.58       0         1                   3   DIAMOND
2        113931.57       1         1                   3   DIAMOND
3         93826.63       0         0                   5      GOLD
4         79084.10       0         0                   5      GOLD

   Point Earned
0           464
1           456
2           377
3           350
4           425
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 18 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   RowNumber          10000 non-null  int64
```

```
 1   CustomerId           10000 non-null  int64
 2   Surname              10000 non-null  object
 3   CreditScore          10000 non-null  int64
 4   Geography            10000 non-null  object
 5   Gender               10000 non-null  object
 6   Age                  10000 non-null  int64
 7   Tenure               10000 non-null  int64
 8   Balance              10000 non-null  float64
 9   NumOfProducts        10000 non-null  int64
 10  HasCrCard            10000 non-null  int64
 11  IsActiveMember       10000 non-null  int64
 12  EstimatedSalary      10000 non-null  float64
 13  Exited               10000 non-null  int64
 14  Complain             10000 non-null  int64
 15  Satisfaction Score   10000 non-null  int64
 16  Card Type            10000 non-null  object
 17  Point Earned         10000 non-null  int64
dtypes: float64(2), int64(12), object(4)
memory usage: 1.4+ MB
```

```
# To check the null values in the data
df.isnull().sum()
```

```
RowNumber              0
CustomerId             0
Surname                0
CreditScore            0
Geography              0
Gender                 0
Age                    0
Tenure                 0
Balance                0
NumOfProducts          0
HasCrCard              0
IsActiveMember         0
EstimatedSalary        0
Exited                 0
Complain               0
Satisfaction Score     0
Card Type              0
Point Earned           0
dtype: int64
```

```
df.describe()
```

```
         RowNumber    CustomerId   CreditScore           Age
Tenure  \
count  10000.00000  1.000000e+04  10000.000000  10000.000000
10000.000000
mean    5000.50000  1.569094e+07    650.528800     38.921800
```

```
5.012800
std     2886.89568  7.193619e+04     96.653299     10.487806
2.892174
min        1.00000  1.556570e+07    350.000000     18.000000
0.000000
25%     2500.75000  1.562853e+07    584.000000     32.000000
3.000000
50%     5000.50000  1.569074e+07    652.000000     37.000000
5.000000
75%     7500.25000  1.575323e+07    718.000000     44.000000
7.000000
max    10000.00000  1.581569e+07    850.000000     92.000000
10.000000

             Balance  NumOfProducts    HasCrCard  IsActiveMember  \
count   10000.000000   10000.000000  10000.00000    10000.000000
mean    76485.889288       1.530200      0.70550        0.515100
std     62397.405202       0.581654      0.45584        0.499797
min         0.000000       1.000000      0.00000        0.000000
25%         0.000000       1.000000      0.00000        0.000000
50%     97198.540000       1.000000      1.00000        1.000000
75%    127644.240000       2.000000      1.00000        1.000000
max    250898.090000       4.000000      1.00000        1.000000

       EstimatedSalary        Exited      Complain  Satisfaction Score
\
count     10000.000000  10000.000000  10000.000000        10000.000000

mean     100090.239881      0.203800      0.204400            3.013800

std       57510.492818      0.402842      0.403283            1.405919

min          11.580000      0.000000      0.000000            1.000000

25%       51002.110000      0.000000      0.000000            2.000000

50%      100193.915000      0.000000      0.000000            3.000000

75%      149388.247500      0.000000      0.000000            4.000000

max      199992.480000      1.000000      1.000000            5.000000


       Point Earned
count  10000.000000
mean     606.515100
std      225.924839
min      119.000000
25%      410.000000
50%      605.000000
```

```
75%      801.000000
max     1000.000000
```

```
df.duplicated().sum()
```

```
np.int64(0)
```

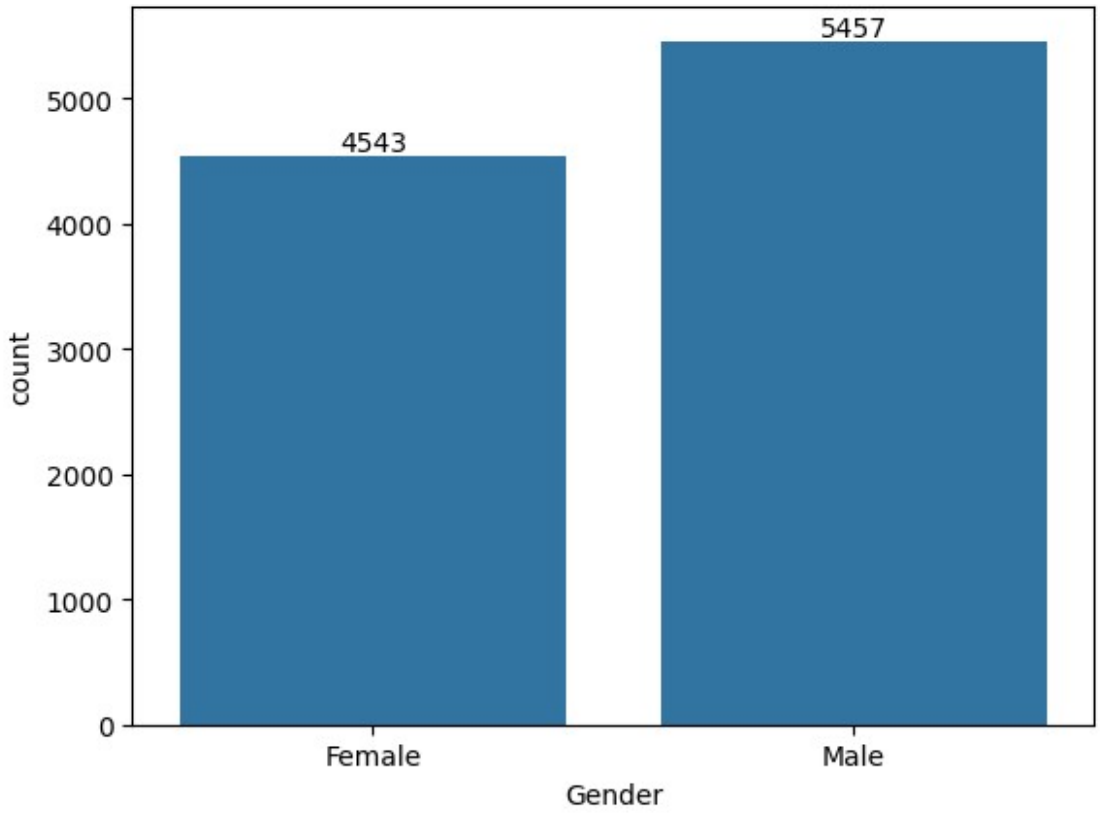
```
df["CustomerId"].duplicated().sum()
```

```
np.int64(0)
```

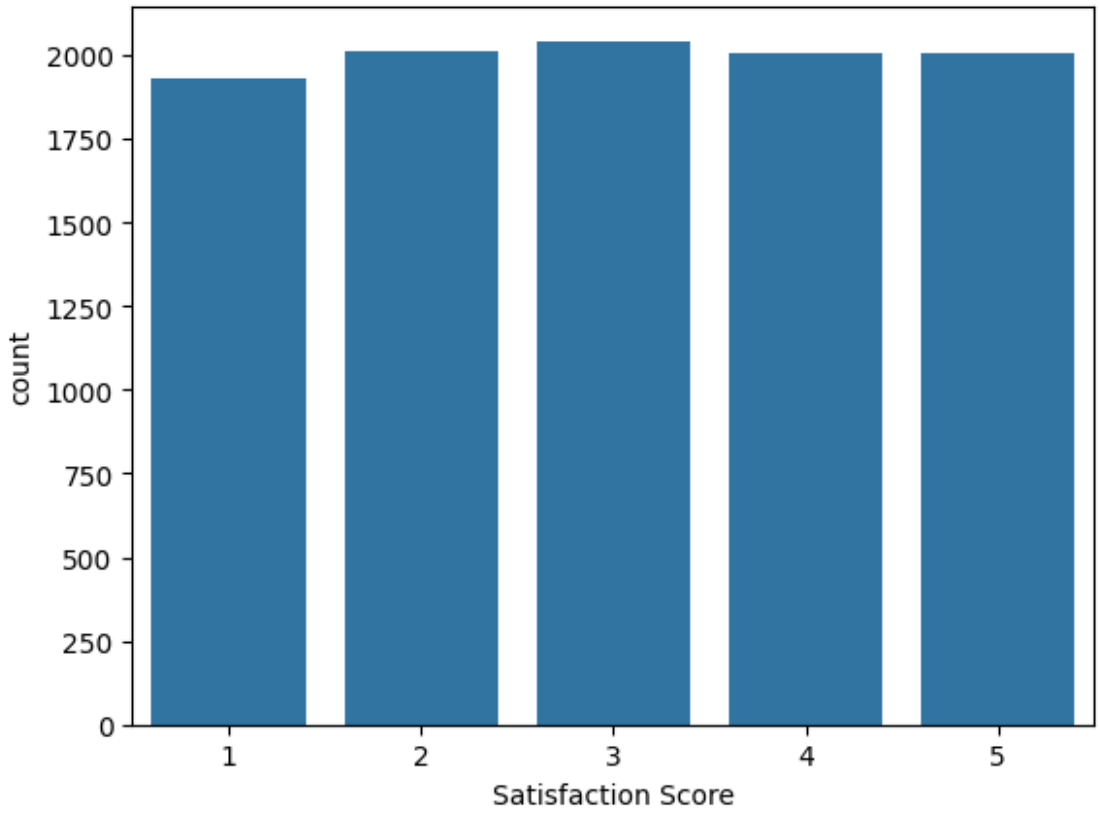
```
df.head()
```

```
   RowNumber  CustomerId   Surname  CreditScore Geography  Gender  Age
\
0          1    15634602  Hargrave          619    France  Female   42

1          2    15647311      Hill          608     Spain  Female   41

2          3    15619304      Onio          502    France  Female   42

3          4    15701354      Boni          699    France  Female   39

4          5    15737888  Mitchell          850     Spain  Female   43


   Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \
0       2       0.00              1          1               1
1       1   83807.86              1          0               1
2       8  159660.80              3          1               0
3       1       0.00              2          0               0
4       2  125510.82              1          1               1

   EstimatedSalary  Exited  Complain  Satisfaction Score Card Type  \
0        101348.88       1         1                   2   DIAMOND
1        112542.58       0         1                   3   DIAMOND
2        113931.57       1         1                   3   DIAMOND
3         93826.63       0         0                   5      GOLD
4         79084.10       0         0                   5      GOLD

   Point Earned
0           464
1           456
2           377
3           350
4           425
```

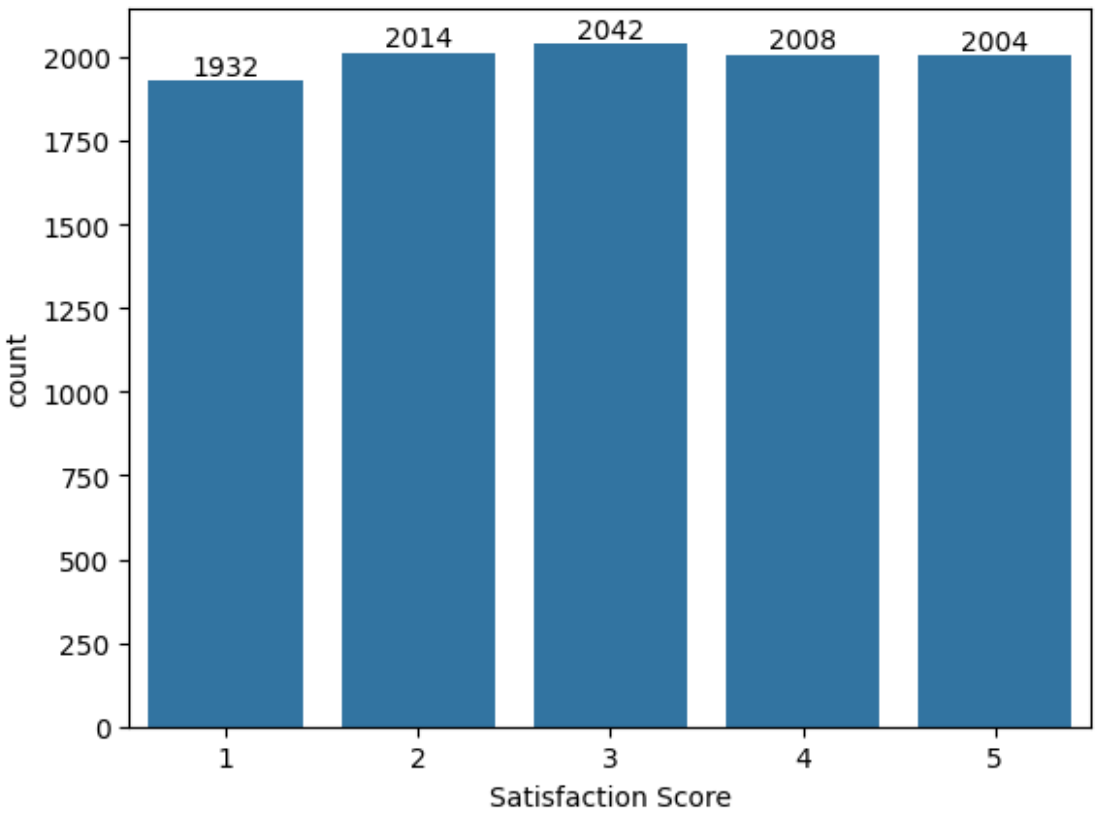
```
# count of customers by Gender
ax = sns.countplot(x = 'Gender', data = df)
ax.bar_label(ax.containers[0])
plt.show()
```




```
# rating for satisfaction score
sns.countplot(x = 'Satisfaction Score', data = df)
plt.show()
```




```
# count of customers rated for every satisfaction score
ax = sns.countplot(x = 'Satisfaction Score', data = df)
ax.bar_label(ax.containers[0])
plt.show()
```




```
df.head()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age
0	1	15634602	Hargrave	619	France	Female	42
1	2	15647311	Hill	608	Spain	Female	41
2	3	15619304	Onio	502	France	Female	42
3	4	15701354	Boni	699	France	Female	39
4	5	15737888	Mitchell	850	Spain	Female	43

	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember
0	2	0.00	1	1	1
1	1	83807.86	1	0	1
2	8	159660.80	3	1	0
3	1	0.00	2	0	0
4	2	125510.82	1	1	1

	EstimatedSalary	Exited	Complain	Satisfaction Score	Card Type
0	101348.88	1	1	2	DIAMOND
1	112542.58	0	1	3	DIAMOND

```
2        113931.57       1         1                   3   DIAMOND
3         93826.63       0         0                   5      GOLD
4         79084.10       0         0                   5      GOLD

   Point Earned
0           464
1           456
2           377
3           350
4           425
```

```
# Conversion of 0 and 1 values to Yes and No

def conv(value):
    if value == 1:
        return "yes"
    else:
        return "no"

df['IsActiveMember'] = df['IsActiveMember'].apply(conv)

df.head()
```

```
   RowNumber  CustomerId   Surname  CreditScore Geography  Gender  Age
\
0          1    15634602  Hargrave          619    France  Female   42

1          2    15647311      Hill          608     Spain  Female   41

2          3    15619304      Onio          502    France  Female   42

3          4    15701354      Boni          699    France  Female   39

4          5    15737888  Mitchell          850     Spain  Female   43


   Tenure    Balance  NumOfProducts  HasCrCard IsActiveMember  \
0       2       0.00              1          1            yes
1       1   83807.86              1          0            yes
2       8  159660.80              3          1             no
3       1       0.00              2          0             no
4       2  125510.82              1          1            yes

   EstimatedSalary  Exited  Complain  Satisfaction Score Card Type  \
0        101348.88       1         1                   2   DIAMOND
1        112542.58       0         1                   3   DIAMOND
2        113931.57       1         1                   3   DIAMOND
3         93826.63       0         0                   5      GOLD
4         79084.10       0         0                   5      GOLD

   Point Earned
```

```
0           464
1           456
2           377
3           350
4           425
```

```
# Conversion of 0 and 1 values to Yes and No

def conv(value):
    if value == 1:
        return "yes"
    else:
        return "no"

df['HasCrCard'] = df['HasCrCard'].apply(conv)

df.head()
```

```
   RowNumber  CustomerId   Surname  CreditScore Geography  Gender  Age
\
0          1    15634602  Hargrave          619    France  Female   42

1          2    15647311      Hill          608     Spain  Female   41

2          3    15619304      Onio          502    France  Female   42

3          4    15701354      Boni          699    France  Female   39

4          5    15737888  Mitchell          850     Spain  Female   43


   Tenure    Balance  NumOfProducts HasCrCard IsActiveMember
EstimatedSalary  \
0       2       0.00              1       yes            yes
101348.88
1       1   83807.86              1        no            yes
112542.58
2       8  159660.80              3       yes             no
113931.57
3       1       0.00              2        no             no
93826.63
4       2  125510.82              1       yes            yes
79084.10

   Exited  Complain  Satisfaction Score Card Type  Point Earned
0       1         1                   2   DIAMOND           464
1       0         1                   3   DIAMOND           456
2       1         1                   3   DIAMOND           377
3       0         0                   5      GOLD           350
4       0         0                   5      GOLD           425
```

```
# Conversion of 0 and 1 values to Yes and No

def conv(value):
    if value == 1:
        return "yes"
    else:
        return "no"

df['Exited'] = df['Exited'].apply(conv)

df.head()
```

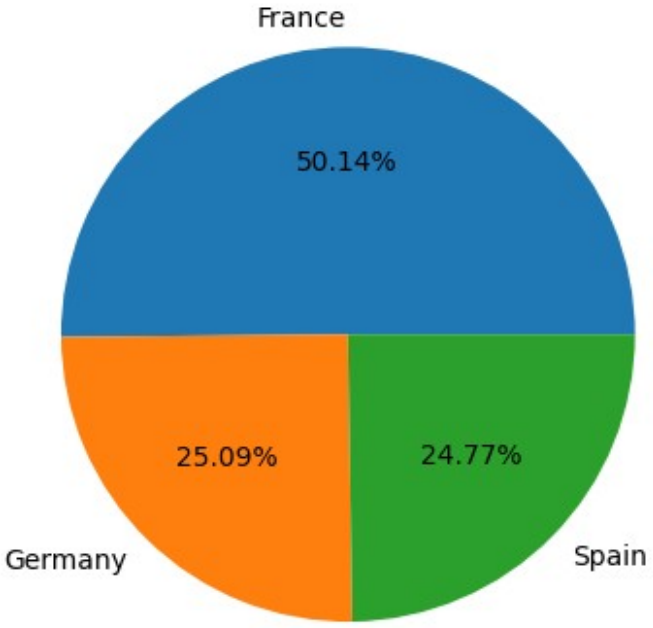
```
   RowNumber  CustomerId   Surname  CreditScore Geography  Gender  Age
\
0          1    15634602  Hargrave          619    France  Female   42

1          2    15647311      Hill          608     Spain  Female   41

2          3    15619304      Onio          502    France  Female   42

3          4    15701354      Boni          699    France  Female   39

4          5    15737888  Mitchell          850     Spain  Female   43


   Tenure    Balance  NumOfProducts HasCrCard IsActiveMember
EstimatedSalary  \
0       2       0.00              1       yes            yes
101348.88
1       1   83807.86              1        no            yes
112542.58
2       8  159660.80              3       yes             no
113931.57
3       1       0.00              2        no             no
93826.63
4       2  125510.82              1       yes            yes
79084.10

  Exited  Complain  Satisfaction Score Card Type  Point Earned
0    yes         1                   2   DIAMOND           464
1     no         1                   3   DIAMOND           456
2    yes         1                   3   DIAMOND           377
3     no         0                   5      GOLD           350
4     no         0                   5      GOLD           425
```

```
# Conversion of 0 and 1 values to Yes and No

def conv(value):
    if value == 1:
        return "yes"
    else:
```

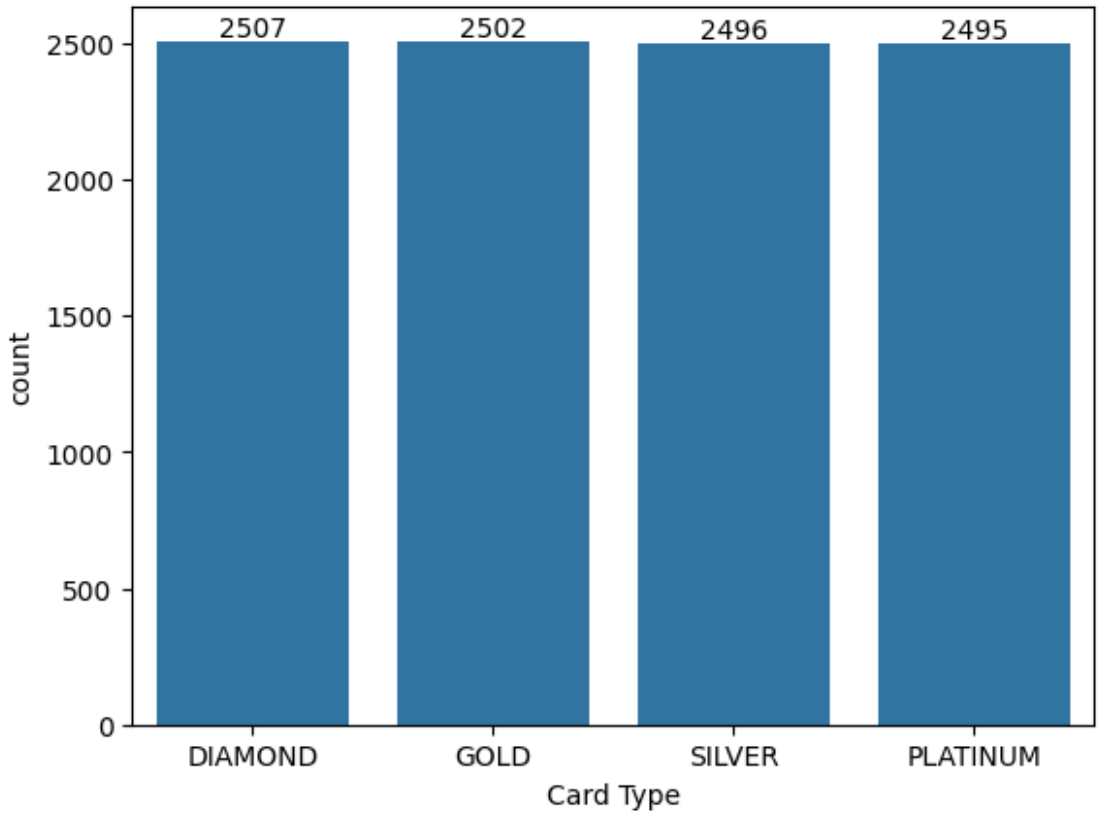
```
        return "no"

df['Complain'] = df['Complain'].apply(conv)

df.head()
```

```
   RowNumber  CustomerId   Surname  CreditScore Geography  Gender  Age
\
0          1    15634602  Hargrave          619    France  Female   42

1          2    15647311      Hill          608     Spain  Female   41

2          3    15619304      Onio          502    France  Female   42

3          4    15701354      Boni          699    France  Female   39

4          5    15737888  Mitchell          850     Spain  Female   43


   Tenure    Balance  NumOfProducts HasCrCard IsActiveMember
EstimatedSalary  \
0       2       0.00              1       yes             yes
101348.88
1       1   83807.86              1        no             yes
112542.58
2       8  159660.80              3       yes              no
113931.57
3       1       0.00              2        no              no
93826.63
4       2  125510.82              1       yes             yes
79084.10

  Exited Complain  Satisfaction Score Card Type  Point Earned
0    yes      yes                   2   DIAMOND           464
1     no      yes                   3   DIAMOND           456
2    yes      yes                   3   DIAMOND           377
3     no       no                   5      GOLD           350
4     no       no                   5      GOLD           425
```

```
# region wise percentage of customers

gb = df.groupby("Geography").agg({'Geography':'count'})
plt.pie(gb["Geography"], labels = gb.index, autopct = "%1.2f%%")
plt.show()
```




```
# count of customers rated for every satisfaction score
ax = sns.countplot(x = 'Card Type', data = df)
ax.bar_label(ax.containers[0])
plt.show()
```



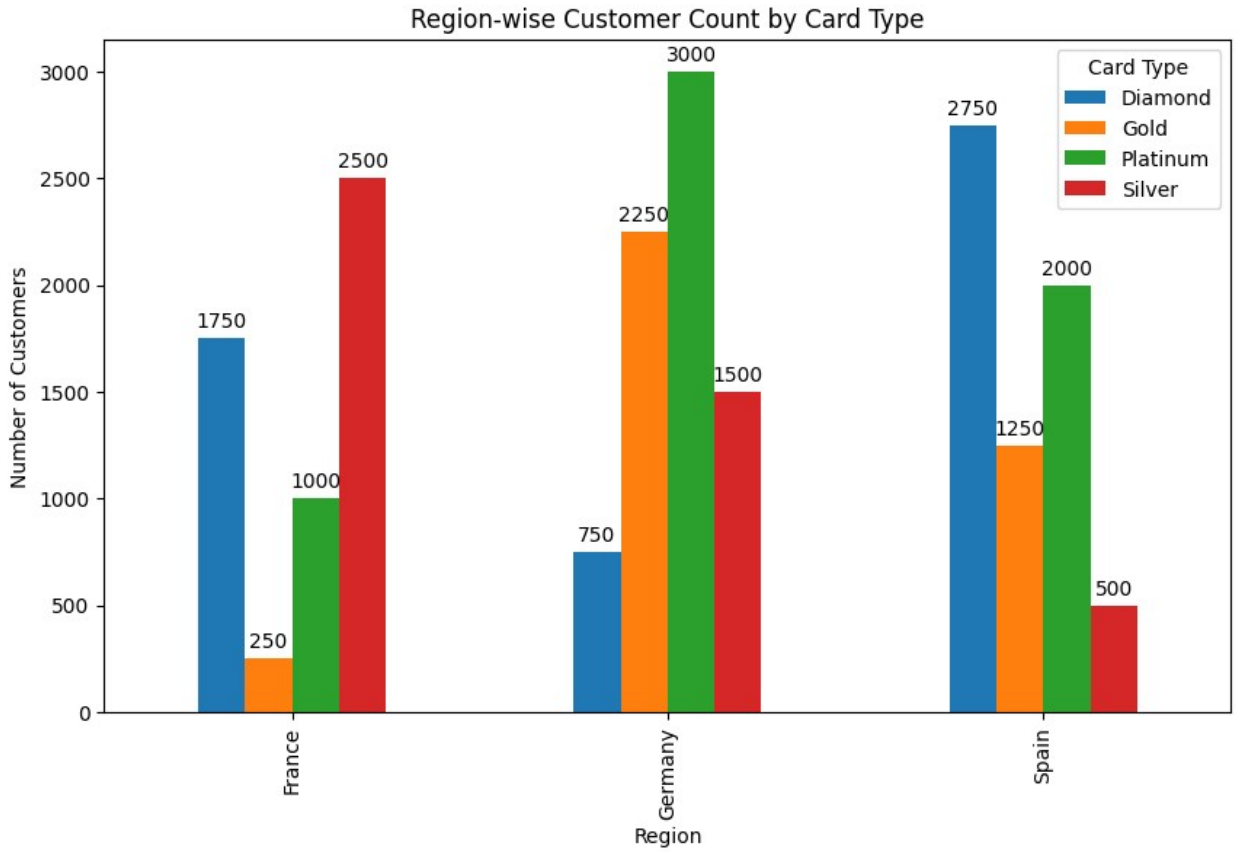

```
df.head()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	yes	yes	101348.88
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	no	yes	112542.58
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	yes	no	113931.57
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	no	no	93826.63
4	5	15737888	Mitchell	850	Spain	Female	43						

```
4       2  125510.82              1       yes            yes
79084.10

  Exited Complain  Satisfaction Score Card Type  Point Earned
0    yes      yes                   2   DIAMOND           464
1     no      yes                   3   DIAMOND           456
2    yes      yes                   3   DIAMOND           377
3     no       no                   5      GOLD           350
4     no       no                   5      GOLD           425
```

```
data = {
    'Region':
['France','Spain','Germany','France','Spain','Germany','France','Spain
','Germany','France','Spain','Germany'],
    'Card_Type': ['Gold', 'Silver', 'Diamond', 'Platinum','Gold',
'Silver', 'Diamond', 'Platinum','Gold', 'Silver', 'Diamond',
'Platinum'],
    'Customer_Count':
[250,500,750,1000,1250,1500,1750,2000,2250,2500,2750,3000]
}

df = pd.DataFrame(data)

# Pivot the data for better visualization
pivot_df = df.pivot_table(index='Region', columns='Card_Type',
values='Customer_Count', aggfunc='sum')

# print("Region-wise Customer Count by Card Type:")
# print(pivot_df)


# Plot a bar chart
ax = pivot_df.plot(kind='bar', figsize=(10, 6))
plt.title('Region-wise Customer Count by Card Type')
plt.xlabel('Region')
plt.ylabel('Number of Customers')
plt.legend(title='Card Type')

# Add value labels on bars
for container in ax.containers:
    ax.bar_label(container, fmt='%d', label_type='edge', fontsize=10,
padding=3)

# # Plot a bar chart
# pivot_df.plot(kind='bar', figsize=(10, 6))
# plt.title('Region-wise Customer Count by Card Type')
# plt.xlabel('Region')
# plt.ylabel('Number of Customers')
# plt.legend(title='Card Type')
```

```
# plt.xticks(rotation=45)
# plt.show()
```



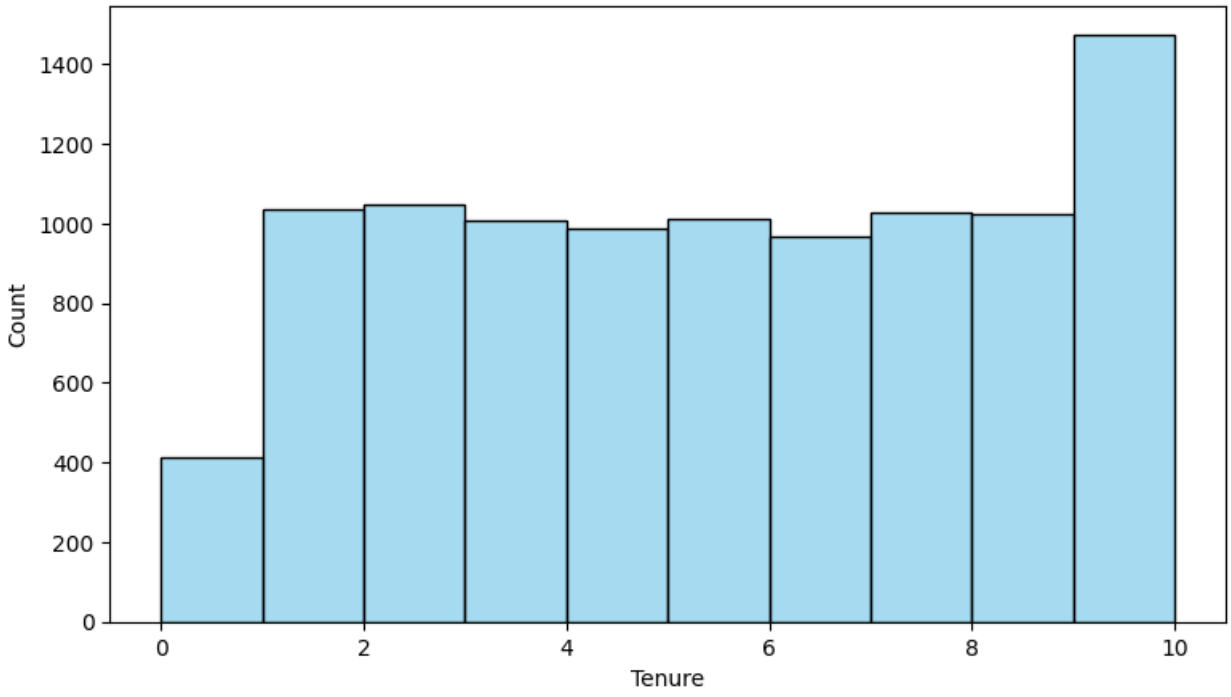

```
df =  pd.read_csv('Customer-Churn-Records.csv')

df.head()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age
0	1	15634602	Hargrave	619	France	Female	42
1	2	15647311	Hill	608	Spain	Female	41
2	3	15619304	Onio	502	France	Female	42
3	4	15701354	Boni	699	France	Female	39
4	5	15737888	Mitchell	850	Spain	Female	43

	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember
0	2	0.00	1	1	1
1	1	83807.86	1	0	1
2	8	159660.80	3	1	0

```
3       1       0.00              2          0               0
4       2  125510.82              1          1               1

   EstimatedSalary  Exited  Complain  Satisfaction Score Card Type  \
0        101348.88       1         1                   2   DIAMOND
1        112542.58       0         1                   3   DIAMOND
2        113931.57       1         1                   3   DIAMOND
3         93826.63       0         0                   5      GOLD
4         79084.10       0         0                   5      GOLD

   Point Earned
0           464
1           456
2           377
3           350
4           425
```

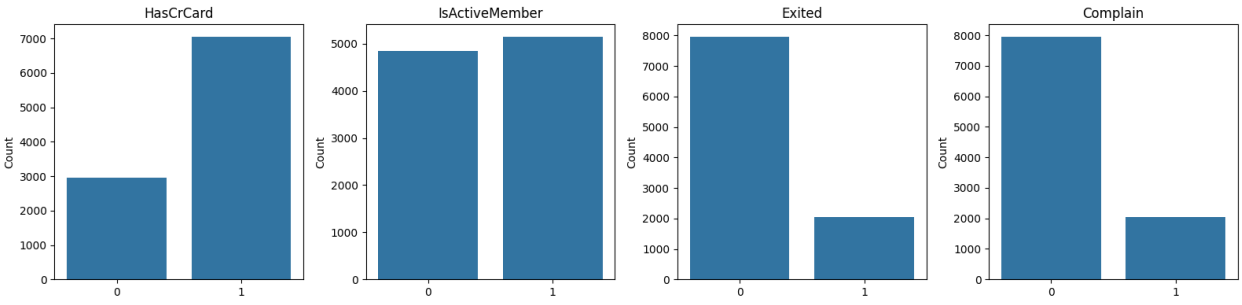
```
df_copy = df.copy()

plt.figure(figsize = (9,5))
sns.histplot(x = 'Tenure', data=df, bins=10, color= 'skyblue')
plt.show()
```




```
df.columns.values

array(['RowNumber', 'CustomerId', 'Surname', 'CreditScore',
'Geography',
       'Gender', 'Age', 'Tenure', 'Balance', 'NumOfProducts',
```

```
'HasCrCard',
       'IsActiveMember', 'EstimatedSalary', 'Exited', 'Complain',
       'Satisfaction Score', 'Card Type', 'Point Earned'],
dtype=object)
```

```
columns = ['HasCrCard', 'IsActiveMember', 'Exited', 'Complain']

# Create subplots
fig, axes = plt.subplots(1, 4, figsize=(16, 4))  # 1 row, 4 columns

# Generate countplots for each column
for i, col in enumerate(columns):
    sns.countplot(x=df[col], ax=axes[i])
    axes[i].set_title(col)  # Set title
    axes[i].set_xlabel('')  # Remove x-label text
    axes[i].set_ylabel('Count')

# Adjust layout for better spacing
plt.tight_layout()
plt.show()
```



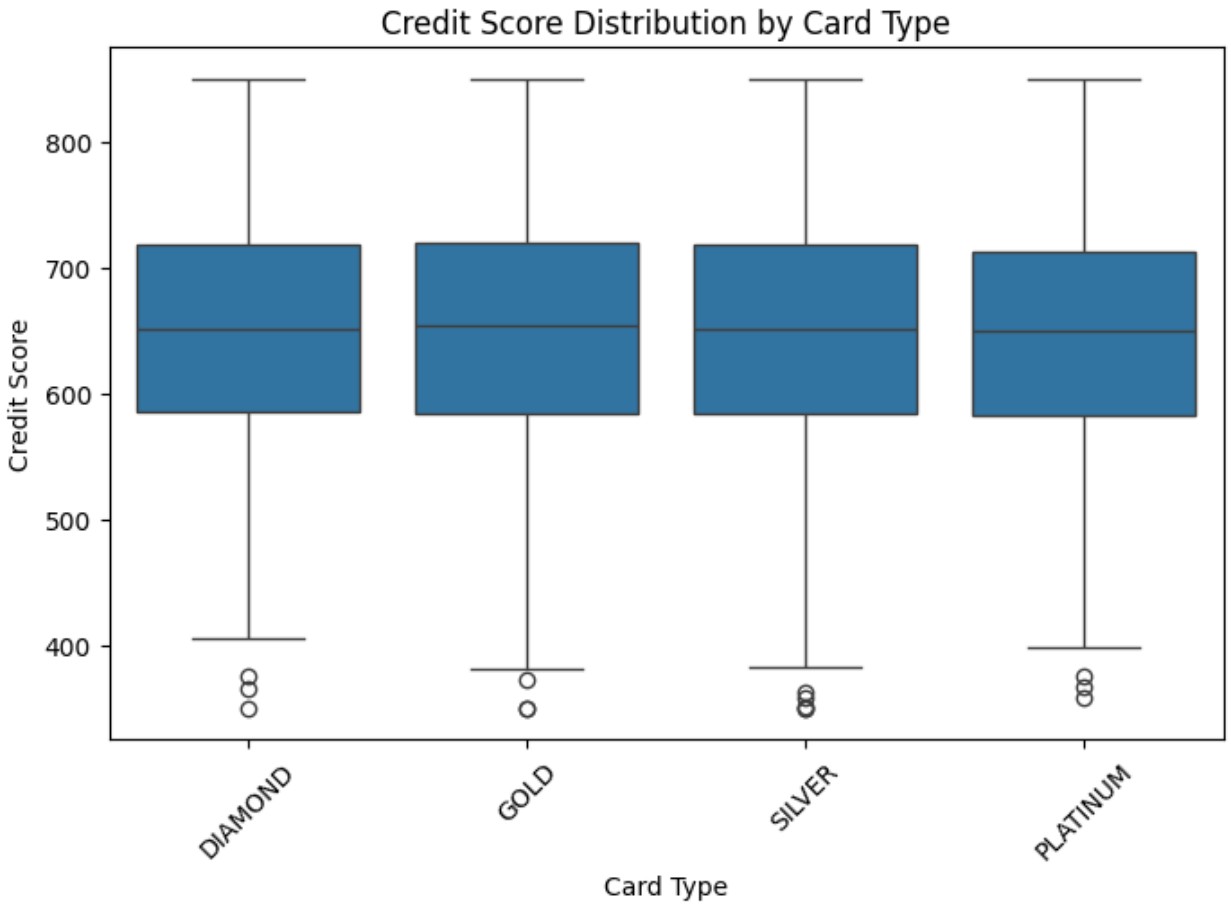
Here 1 represents "YES" and 0 represents "NO".

```
df.head()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age
0	1	15634602	Hargrave	619	France	Female	42
1	2	15647311	Hill	608	Spain	Female	41
2	3	15619304	Onio	502	France	Female	42
3	4	15701354	Boni	699	France	Female	39
4	5	15737888	Mitchell	850	Spain	Female	43

```
   Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \
0       2       0.00              1          1               1
1       1   83807.86              1          0               1
2       8  159660.80              3          1               0
3       1       0.00              2          0               0
4       2  125510.82              1          1               1

   EstimatedSalary  Exited  Complain  Satisfaction Score Card Type  \
0        101348.88       1         1                   2   DIAMOND
1        112542.58       0         1                   3   DIAMOND
2        113931.57       1         1                   3   DIAMOND
3         93826.63       0         0                   5      GOLD
4         79084.10       0         0                   5      GOLD

   Point Earned
0           464
1           456
2           377
3           350
4           425
```

```
plt.figure(figsize=(8, 5))
sns.boxplot(x=df['Card Type'], y=df['CreditScore'])

plt.title('Credit Score Distribution by Card Type')
plt.xlabel('Card Type')
plt.ylabel('Credit Score')
plt.xticks(rotation=45)
plt.show()
```




```
df.head()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age
0	1	15634602	Hargrave	619	France	Female	42
1	2	15647311	Hill	608	Spain	Female	41
2	3	15619304	Onio	502	France	Female	42
3	4	15701354	Boni	699	France	Female	39
4	5	15737888	Mitchell	850	Spain	Female	43

	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember
0	2	0.00	1	1	1
1	1	83807.86	1	0	1
2	8	159660.80	3	1	0
3	1	0.00	2	0	0
4	2	125510.82	1	1	1

	EstimatedSalary	Exited	Complain	Satisfaction Score	Card Type	\
0	101348.88	1	1	2	DIAMOND	
1	112542.58	0	1	3	DIAMOND	
2	113931.57	1	1	3	DIAMOND	
3	93826.63	0	0	5	GOLD	
4	79084.10	0	0	5	GOLD	

	Point Earned
0	464
1	456
2	377
3	350
4	425