

eda-1

June 8, 2025

```
[4]: import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
```

```
[5]: df = pd.read_csv('/content/Customer Churn.csv')
df
```

```
[5]:
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	\
0	7590-VHVEG	Female	0	Yes	No	1	
1	5575-GNVDE	Male	0	No	No	34	
2	3668-QPYBK	Male	0	No	No	2	
3	7795-CFOCW	Male	0	No	No	45	
4	9237-HQITU	Female	0	No	No	2	
...	
7038	6840-RESVB	Male	0	Yes	Yes	24	
7039	2234-XADUH	Female	0	Yes	Yes	72	
7040	4801-JZAZL	Female	0	Yes	Yes	11	
7041	8361-LTMKD	Male	1	Yes	No	4	
7042	3186-AJIEK	Male	0	No	No	66	

	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	\
0	No	No phone service	DSL	No	...	
1	Yes	No	DSL	Yes	...	
2	Yes	No	DSL	Yes	...	
3	No	No phone service	DSL	Yes	...	
4	Yes	No	Fiber optic	No	...	
...	
7038	Yes	Yes	DSL	Yes	...	
7039	Yes	Yes	Fiber optic	No	...	
7040	No	No phone service	DSL	Yes	...	
7041	Yes	Yes	Fiber optic	No	...	
7042	Yes	No	Fiber optic	Yes	...	

	DeviceProtection	TechSupport	StreamingTV	StreamingMovies	Contract	\
0	No	No	No	No	Month-to-month	
1	Yes	No	No	No	One year	

2	No	No	No	No	Month-to-month
3	Yes	Yes	No	No	One year
4	No	No	No	No	Month-to-month
...
7038	Yes	Yes	Yes	Yes	One year
7039	Yes	No	Yes	Yes	One year
7040	No	No	No	No	Month-to-month
7041	No	No	No	No	Month-to-month
7042	Yes	Yes	Yes	Yes	Two year

	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	\
0	Yes	Electronic check	29.85	29.85	
1	No	Mailed check	56.95	1889.5	
2	Yes	Mailed check	53.85	108.15	
3	No	Bank transfer (automatic)	42.30	1840.75	
4	Yes	Electronic check	70.70	151.65	
...	
7038	Yes	Mailed check	84.80	1990.5	
7039	Yes	Credit card (automatic)	103.20	7362.9	
7040	Yes	Electronic check	29.60	346.45	
7041	Yes	Mailed check	74.40	306.6	
7042	Yes	Bank transfer (automatic)	105.65	6844.5	

Churn	
0	No
1	No
2	Yes
3	No
4	Yes
...	...
7038	No
7039	No
7040	No
7041	Yes
7042	No

[7043 rows x 21 columns]

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

```
[6]:  customerID  gender  SeniorCitizen  Partner  Dependents  tenure  PhoneService  \
0  7590-VHVEG  Female              0      Yes           No         1           No
1  5575-GNVDE   Male              0      No            No        34           Yes
2  3668-QPYBK   Male              0      No            No         2           Yes
3  7795-CFOCW   Male              0      No            No        45           No
4  9237-HQITU   Female            0      No            No         2           Yes
```

	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	\
0	No phone service	DSL	No	...	No	
1	No	DSL	Yes	...	Yes	
2	No	DSL	Yes	...	No	
3	No phone service	DSL	Yes	...	Yes	
4	No	Fiber optic	No	...	No	

	TechSupport	StreamingTV	StreamingMovies	Contract	PaperlessBilling	\
0	No	No	No	Month-to-month	Yes	
1	No	No	No	One year	No	
2	No	No	No	Month-to-month	Yes	
3	Yes	No	No	One year	No	
4	No	No	No	Month-to-month	Yes	

	PaymentMethod	MonthlyCharges	TotalCharges	Churn
0	Electronic check	29.85	29.85	No
1	Mailed check	56.95	1889.5	No
2	Mailed check	53.85	108.15	Yes
3	Bank transfer (automatic)	42.30	1840.75	No
4	Electronic check	70.70	151.65	Yes

[5 rows x 21 columns]

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
#   Column                Non-Null Count  Dtype
---  -
0   customerID            7043 non-null   object
1   gender                 7043 non-null   object
2   SeniorCitizen          7043 non-null   int64
3   Partner                7043 non-null   object
4   Dependents             7043 non-null   object
5   tenure                 7043 non-null   int64
6   PhoneService           7043 non-null   object
7   MultipleLines          7043 non-null   object
8   InternetService        7043 non-null   object
9   OnlineSecurity         7043 non-null   object
10  OnlineBackup           7043 non-null   object
11  DeviceProtection       7043 non-null   object
12  TechSupport            7043 non-null   object
13  StreamingTV            7043 non-null   object
14  StreamingMovies        7043 non-null   object
15  Contract               7043 non-null   object
16  PaperlessBilling       7043 non-null   object
```

```

17 PaymentMethod      7043 non-null  object
18 MonthlyCharges     7043 non-null  float64
19 TotalCharges       7043 non-null  object
20 Churn              7043 non-null  object
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB

```

```
[8]: df['TotalCharges'] = df['TotalCharges'].replace(" ", "0")
df['TotalCharges'] = df['TotalCharges'].astype(float)
```

```
[9]: df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
#   Column                Non-Null Count  Dtype
---  -
0   customerID            7043 non-null  object
1   gender                7043 non-null  object
2   SeniorCitizen         7043 non-null  int64
3   Partner               7043 non-null  object
4   Dependents            7043 non-null  object
5   tenure                7043 non-null  int64
6   PhoneService          7043 non-null  object
7   MultipleLines         7043 non-null  object
8   InternetService       7043 non-null  object
9   OnlineSecurity        7043 non-null  object
10  OnlineBackup          7043 non-null  object
11  DeviceProtection      7043 non-null  object
12  TechSupport           7043 non-null  object
13  StreamingTV           7043 non-null  object
14  StreamingMovies       7043 non-null  object
15  Contract              7043 non-null  object
16  PaperlessBilling      7043 non-null  object
17  PaymentMethod         7043 non-null  object
18  MonthlyCharges        7043 non-null  float64
19  TotalCharges          7043 non-null  float64
20  Churn                 7043 non-null  object
dtypes: float64(2), int64(2), object(17)
memory usage: 1.1+ MB

```

```
[10]: df.isnull().sum().sum()
```

```
[10]: np.int64(0)
```

```
[11]: df.describe()
```

```
[11]:      SeniorCitizen      tenure  MonthlyCharges  TotalCharges
count      7043.000000  7043.000000      7043.000000      7043.000000
mean         0.162147    32.371149         64.761692    2279.734304
std          0.368612    24.559481         30.090047    2266.794470
min          0.000000     0.000000         18.250000     0.000000
25%          0.000000     9.000000         35.500000    398.550000
50%          0.000000    29.000000         70.350000   1394.550000
75%          0.000000    55.000000         89.850000   3786.600000
max          1.000000    72.000000        118.750000   8684.800000
```

```
[12]: df["customerID"].duplicated().sum()
```

```
[12]: np.int64(0)
```

```
[13]: def conv(value):
      if value == 1:
          return "Yes"
      else:
          return "No"

      df["SeniorCitizen"] = df["SeniorCitizen"].apply(conv)
```

```
[14]: df
```

```
[14]:      customerID  gender SeniorCitizen Partner Dependents  tenure \
0      7590-VHVEG  Female             No      Yes          No        1
1      5575-GNVDE   Male             No      No          No       34
2      3668-QPYBK   Male             No      No          No        2
3      7795-CFOWC   Male             No      No          No       45
4      9237-HQITU  Female             No      No          No        2
...      ...      ...      ...      ...      ...      ...
7038   6840-RESVB   Male             No      Yes          Yes       24
7039   2234-XADUH  Female             No      Yes          Yes       72
7040   4801-JAZL   Female             No      Yes          Yes       11
7041   8361-LTMKD   Male             Yes      Yes          No        4
7042   3186-AJIEK   Male             No      No          No       66

      PhoneService  MultipleLines  InternetService  OnlineSecurity  ... \
0              No  No phone service              DSL              No  ...
1              Yes              No              DSL              Yes  ...
2              Yes              No              DSL              Yes  ...
3              No  No phone service              DSL              Yes  ...
4              Yes              No  Fiber optic              No  ...
...      ...      ...      ...      ...      ...
7038          Yes              Yes              DSL              Yes  ...
7039          Yes              Yes  Fiber optic              No  ...
7040          No  No phone service              DSL              Yes  ...
```

7041	Yes	Yes	Fiber optic	No	...
7042	Yes	No	Fiber optic	Yes	...

	DeviceProtection	TechSupport	StreamingTV	StreamingMovies	Contract	\
0	No	No	No	No	Month-to-month	
1	Yes	No	No	No	One year	
2	No	No	No	No	Month-to-month	
3	Yes	Yes	No	No	One year	
4	No	No	No	No	Month-to-month	
...	
7038	Yes	Yes	Yes	Yes	One year	
7039	Yes	No	Yes	Yes	One year	
7040	No	No	No	No	Month-to-month	
7041	No	No	No	No	Month-to-month	
7042	Yes	Yes	Yes	Yes	Two year	

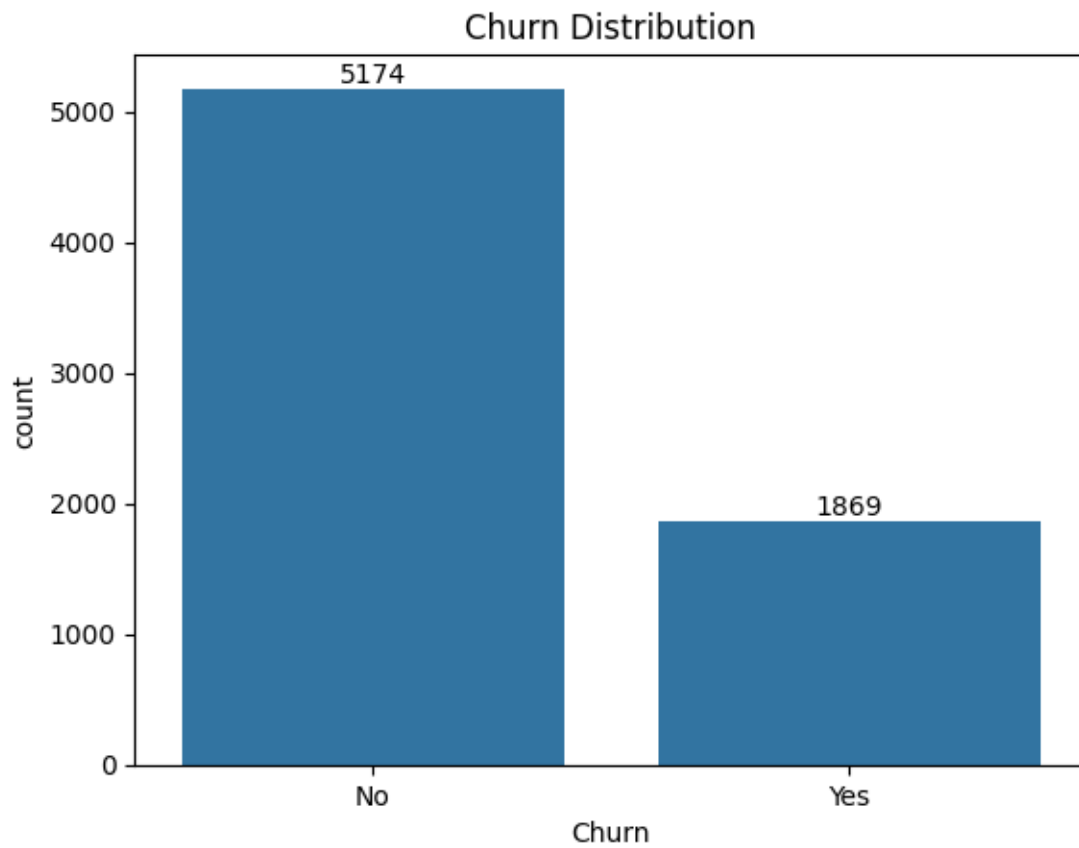
	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	\
0	Yes	Electronic check	29.85	29.85	
1	No	Mailed check	56.95	1889.50	
2	Yes	Mailed check	53.85	108.15	
3	No	Bank transfer (automatic)	42.30	1840.75	
4	Yes	Electronic check	70.70	151.65	
...	
7038	Yes	Mailed check	84.80	1990.50	
7039	Yes	Credit card (automatic)	103.20	7362.90	
7040	Yes	Electronic check	29.60	346.45	
7041	Yes	Mailed check	74.40	306.60	
7042	Yes	Bank transfer (automatic)	105.65	6844.50	

	Churn
0	No
1	No
2	Yes
3	No
4	Yes
...	...
7038	No
7039	No
7040	No
7041	Yes
7042	No

[7043 rows x 21 columns]

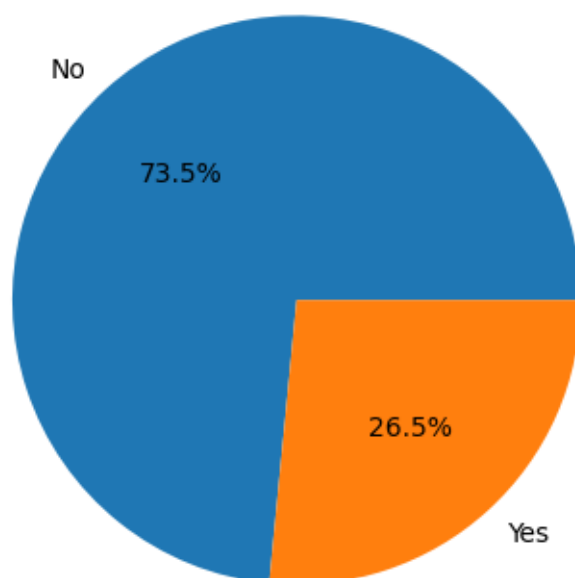
```
[15]: ax = sns.countplot(x="Churn", data=df)
plt.title("Churn Distribution")
ax.bar_label(ax.containers[0])
```

```
plt.show()
```

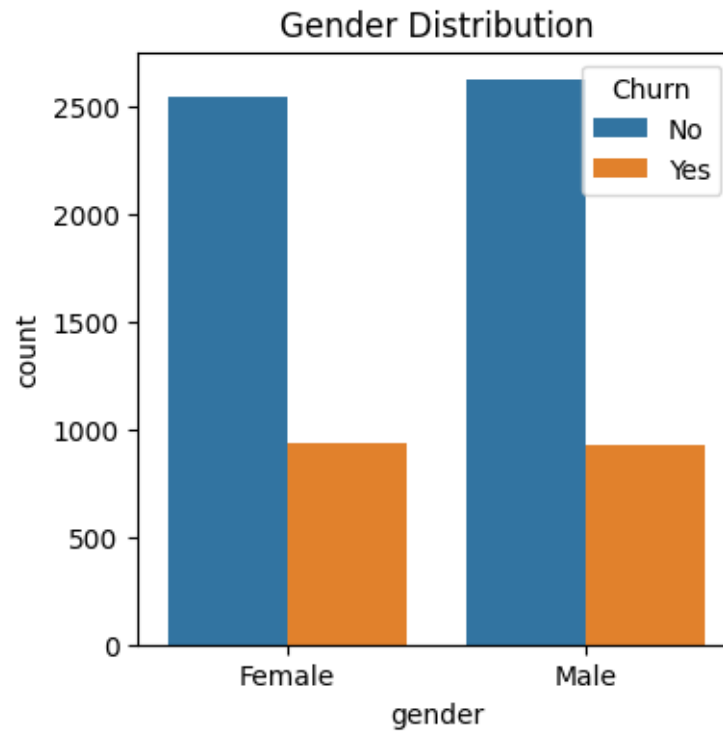


```
[16]: gb = df.groupby("Churn").agg({"Churn": "count"})
plt.pie(gb['Churn'], labels=gb.index, autopct='%1.1f%%')
plt.title("Churn Distribution")
plt.show()
```

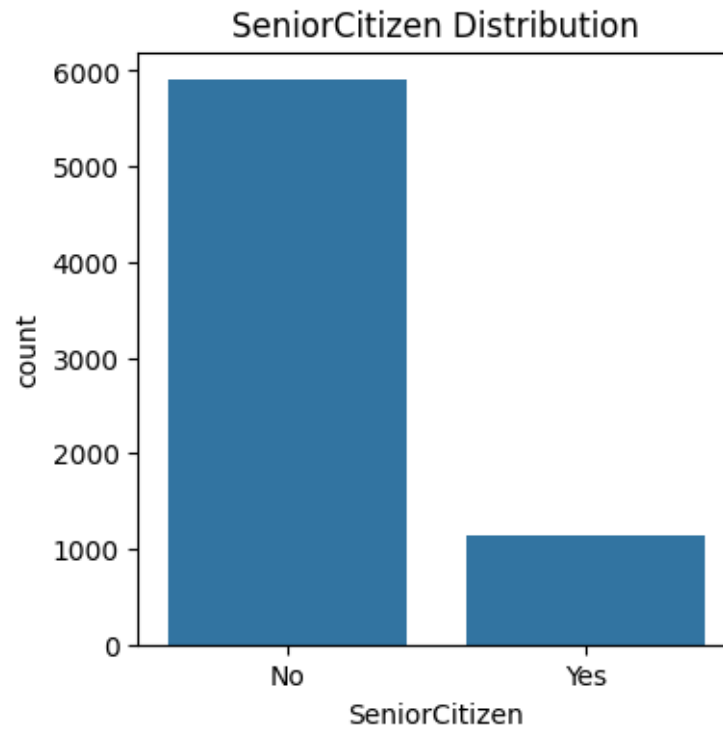
Churn Distribution



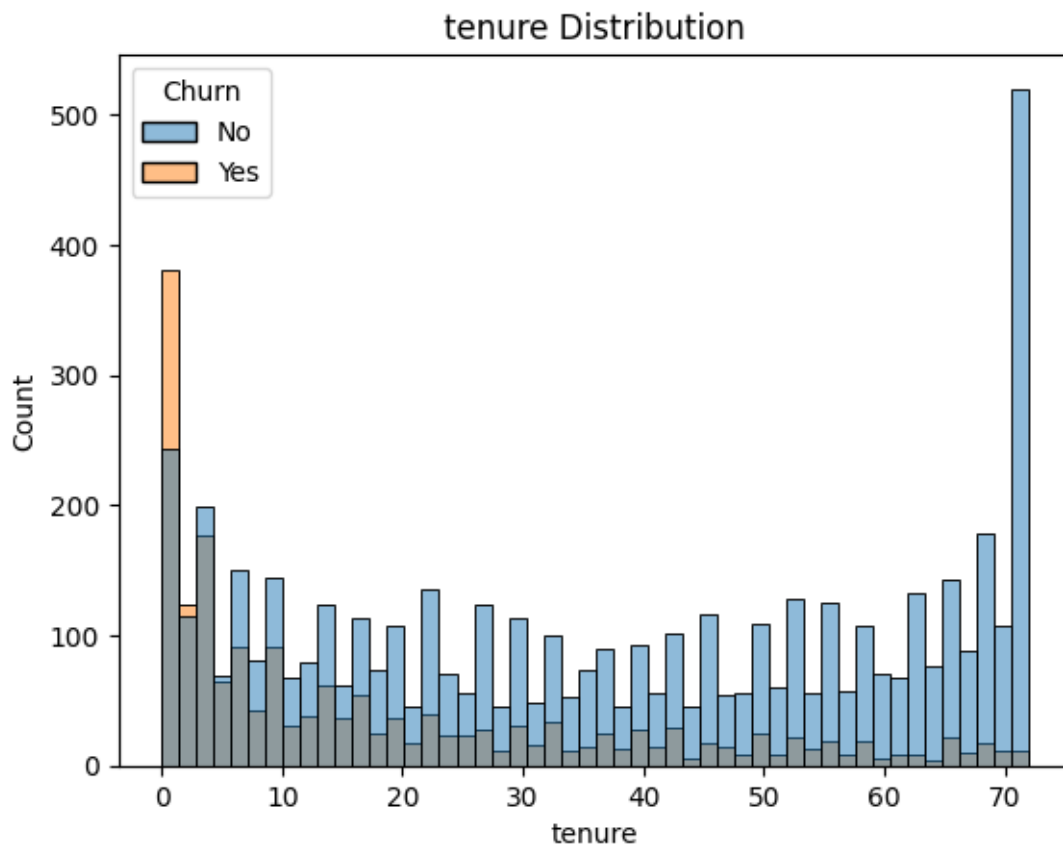
```
[19]: plt.figure(figsize = (4,4))  
sns.countplot(x="gender", hue="Churn", data=df)  
plt.title("Gender Distribution")  
plt.show()
```

```
[21]: plt.figure(figsize = (4,4))
sns.countplot(x="SeniorCitizen", data=df)
plt.title("SeniorCitizen Distribution")
plt.show()
```

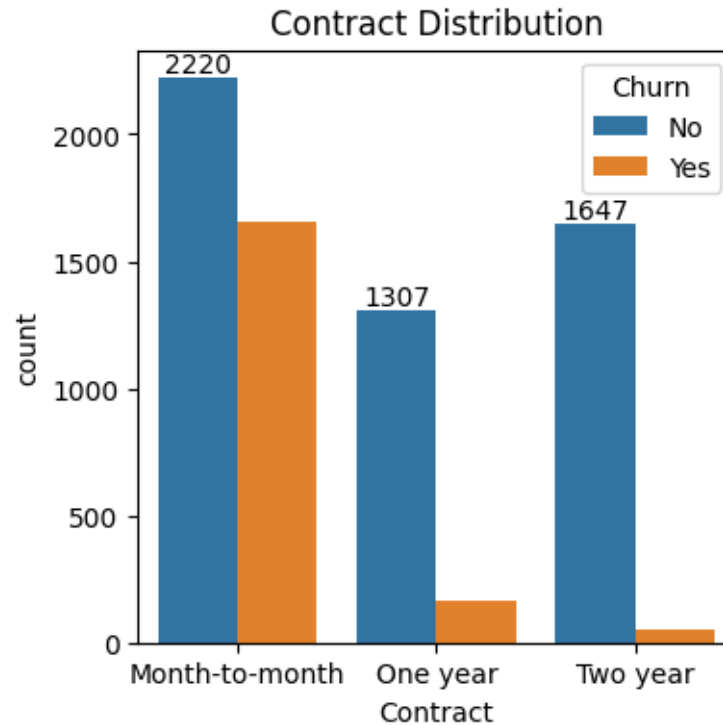


```
[31]: sns.histplot(x="tenure",hue="Churn", data=df, bins = 50)  
plt.title("tenure Distribution")  
plt.show()
```



[]:

```
[37]: plt.figure(figsize = (4,4))
      ax = sns.countplot(x="Contract",hue= "Churn", data=df)
      ax.bar_label(ax.containers[0])
      plt.title("Contract Distribution")
      plt.show()
```



```
[38]: df.columns.values
```

```
[38]: array(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',
        'tenure', 'PhoneService', 'MultipleLines', 'InternetService',
        'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
        'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract',
        'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges',
        'TotalCharges', 'Churn'], dtype=object)
```

```
[42]: # Ensure 'TotalCharges' is numeric
df['TotalCharges'] = pd.to_numeric(df['TotalCharges'], errors='coerce')

# Bin numerical features
df['tenure_binned'] = pd.cut(df['tenure'], bins=[0, 12, 24, 48, 60, 72],
    labels=['0-12', '13-24', '25-48', '49-60', '61-72'])
df['MonthlyCharges_binned'] = pd.cut(df['MonthlyCharges'], bins=5)
df['TotalCharges_binned'] = pd.cut(df['TotalCharges'], bins=5)

# Use these binned versions
features = [
    'gender', 'SeniorCitizen', 'Partner', 'Dependents', 'PhoneService',
    'MultipleLines', 'InternetService', 'OnlineSecurity', 'OnlineBackup',
    'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies',
```

```

    'Contract', 'PaperlessBilling', 'PaymentMethod', 'Churn',
    'tenure_binned', 'MonthlyCharges_binned', 'TotalCharges_binned'
]

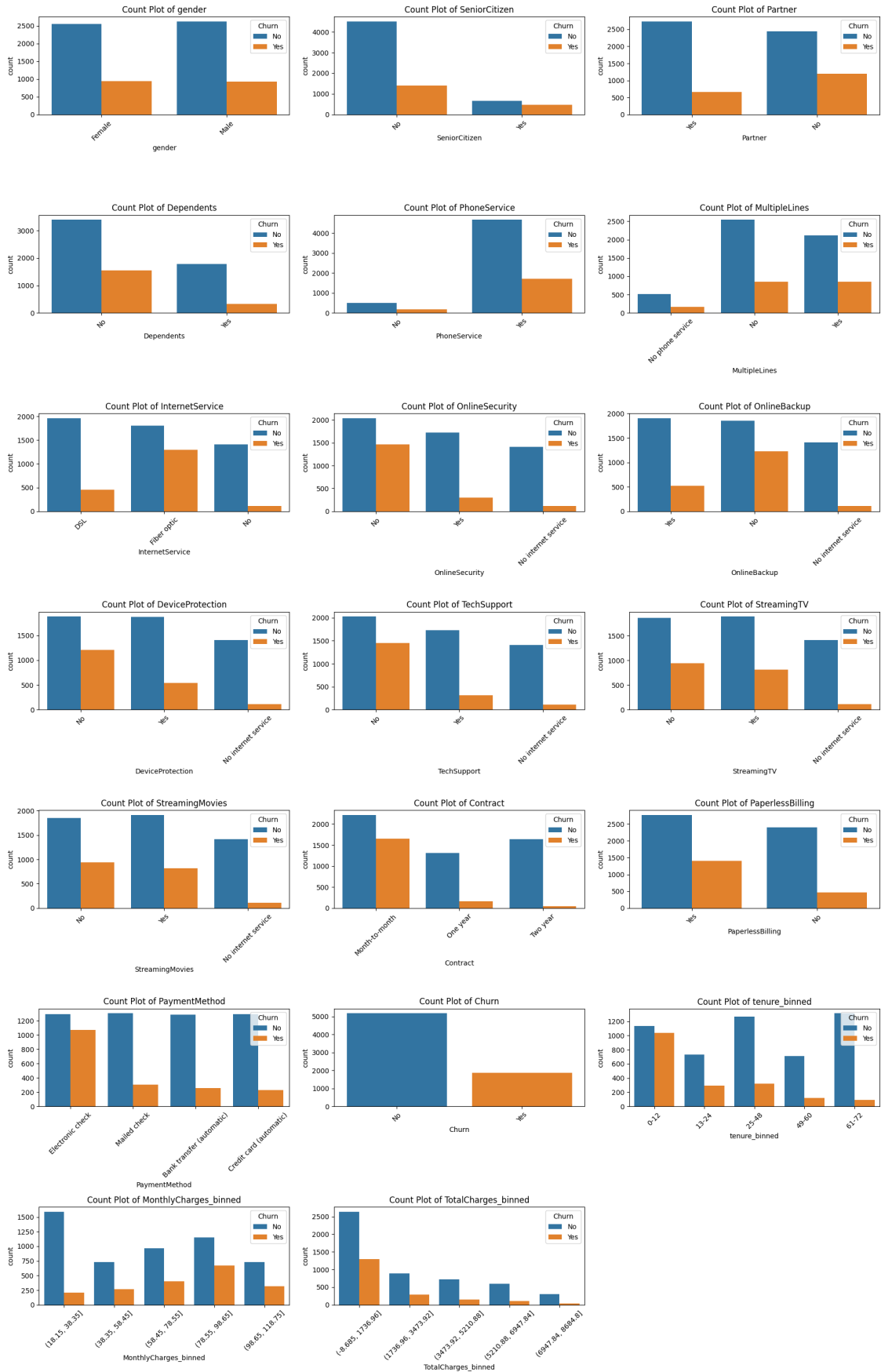
# Set up subplots
n = len(features)
rows = (n // 3) + int(n % 3 != 0)
fig, axes = plt.subplots(nrows=rows, ncols=3, figsize=(18, rows * 4))
axes = axes.flatten()

# Plot count plots
for i, col in enumerate(features):
    sns.countplot(data=df, x=col, ax=axes[i], hue="Churn")
    axes[i].set_title(f'Count Plot of {col}')
    axes[i].tick_params(axis='x', rotation=45)

# Remove unused axes
for j in range(i + 1, len(axes)):
    fig.delaxes(axes[j])

plt.tight_layout()
plt.show()

```



```
[48]: plt.figure(figsize = (10,10))
ax = sns.countplot(x="PaymentMethod",hue= "Churn", data=df)
ax.bar_label(ax.containers[0])
plt.title("PaymentMethod Distribution")
plt.show()
```

