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

df = pd.read_csv("F:/AI Projects/11.WA_Fn-UseC_-Telco-Customer-
Churn_Project/WA_Fn-UseC_-Telco-Customer-Churn_new_ad.csv")
df
```

```
      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
...           ...     ...            ...     ...        ...     ...
10318  7549-MYXPK  Female              0      No         No      16
10319  3948-BLXYF  Female              0      No        Yes      66
10320  3149-NPXCN    Male              1      No         No      45
10321  5140-PTXKA    Male              0      No        Yes      69
10322  4641-NJXUX    Male              1      No         No      15

      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
...             ...               ...             ...
...
10318           Yes               Yes     Fiber optic  No internet
service
10319           Yes                No              No
Yes
10320           Yes               Yes     Fiber optic  No internet
service
10321           Yes  No phone service              No
No
10322           Yes                No     Fiber optic
No

       ...     DeviceProtection          TechSupport
StreamingTV  \
0      ...                   No                   No
```

```
No
1      ...                 Yes                  No
No
2      ...                  No                  No
No
3      ...                 Yes                 Yes
No
4      ...                  No                  No
No
...    ...                 ...                 ...
...
10318  ...  No internet service  No internet service  No internet
service
10319  ...                  No                  No
Yes
10320  ...  No internet service  No internet service  No internet
service
10321  ...                  No                 Yes
Yes
10322  ...                  No                  No
Yes

           StreamingMovies        Contract PaperlessBilling  \
0                       No  Month-to-month              Yes
1                       No        One year               No
2                       No  Month-to-month              Yes
3                       No        One year               No
4                       No  Month-to-month              Yes
...                    ...             ...              ...
10318  No internet service        Two year               No
10319                  Yes        One year               No
10320  No internet service        One year               No
10321                  Yes  Month-to-month              Yes
10322                  Yes  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
...                          ...            ...           ...   ...
10318               Mailed check          20.40         292.4    No
10319               Mailed check          74.80        1821.2    No
10320    Credit card (automatic)          24.80       1600.95    No
10321               Mailed check         100.85        399.25    No
10322           Electronic check         101.35       1218.55   Yes

[10323 rows x 21 columns]
```

```
df.head(2)
```

```
   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

      MultipleLines InternetService OnlineSecurity  ...
DeviceProtection  \
0  No phone service             DSL             No  ...
No
1                No             DSL            Yes  ...
Yes

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

      PaymentMethod MonthlyCharges  TotalCharges Churn
0  Electronic check          29.85         29.85    No
1      Mailed check          56.95        1889.5    No

[2 rows x 21 columns]
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10323 entries, 0 to 10322
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   customerID        10323 non-null  object
 1   gender            10323 non-null  object
 2   SeniorCitizen     10323 non-null  int64
 3   Partner           10323 non-null  object
 4   Dependents        10323 non-null  object
 5   tenure            10323 non-null  int64
 6   PhoneService      10323 non-null  object
 7   MultipleLines     10323 non-null  object
 8   InternetService   10323 non-null  object
 9   OnlineSecurity    10323 non-null  object
 10  OnlineBackup      10323 non-null  object
 11  DeviceProtection  10323 non-null  object
 12  TechSupport       10323 non-null  object
 13  StreamingTV       10323 non-null  object
 14  StreamingMovies   10323 non-null  object
 15  Contract          10323 non-null  object
 16  PaperlessBilling  10323 non-null  object
 17  PaymentMethod     10323 non-null  object
 18  MonthlyCharges    10323 non-null  float64
 19  TotalCharges      10295 non-null  object
 20  Churn             10323 non-null  object
dtypes: float64(1), int64(2), object(18)
memory usage: 1.7+ MB
```

```
df.columns
```

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

```
# Replacing blanks with 0 as tenure is 0 and no totoal charges are
record

df["TotalCharges"] = df["TotalCharges"].replace(" ", "0")
df["TotalCharges"] = df["TotalCharges"].astype("float")

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10323 entries, 0 to 10322
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   customerID        10323 non-null  object
 1   gender            10323 non-null  object
 2   SeniorCitizen     10323 non-null  int64
 3   Partner           10323 non-null  object
 4   Dependents        10323 non-null  object
 5   tenure            10323 non-null  int64
 6   PhoneService      10323 non-null  object
 7   MultipleLines     10323 non-null  object
 8   InternetService   10323 non-null  object
 9   OnlineSecurity    10323 non-null  object
 10  OnlineBackup      10323 non-null  object
 11  DeviceProtection  10323 non-null  object
 12  TechSupport       10323 non-null  object
 13  StreamingTV       10323 non-null  object
 14  StreamingMovies   10323 non-null  object
 15  Contract          10323 non-null  object
 16  PaperlessBilling  10323 non-null  object 
 17  PaymentMethod     10323 non-null  object 
 18  MonthlyCharges    10323 non-null  float64
 19  TotalCharges      10295 non-null  float64
 20  Churn             10323 non-null  object 
dtypes: float64(2), int64(2), object(17)
memory usage: 1.7+ MB
```

```
df.isnull()
```

```
       customerID  gender  SeniorCitizen  Partner  Dependents  tenure 
\
0           False   False          False    False       False   False

1           False   False          False    False       False   False

2           False   False          False    False       False   False

3           False   False          False    False       False   False

4           False   False          False    False       False   False

...           ...     ...            ...      ...         ...     ...

10318       False   False          False    False       False   False

10319       False   False          False    False       False   False

10320       False   False          False    False       False   False

10321       False   False          False    False       False   False

10322       False   False          False    False       False   False


       PhoneService  MultipleLines  InternetService 
OnlineSecurity  ...  \
0             False          False            False 
False  ...   
1             False          False            False 
False  ...   
2             False          False            False 
False  ...   
3             False          False            False 
False  ...   
4             False          False            False 
False  ...   
...             ...            ...              ...             ...  .
..   
10318         False          False            False 
False  ...   
```

```
10319        False        False          False
False  ...
10320        False        False          False
False  ...
10321        False        False          False
False  ...
10322        False        False          False
False  ...

       DeviceProtection  TechSupport  StreamingTV  StreamingMovies
Contract  \
0                 False        False        False            False
False
1                 False        False        False            False
False
2                 False        False        False            False
False
3                 False        False        False            False
False
4                 False        False        False            False
False
...                 ...          ...          ...              ...
...
10318             False        False        False            False
False
10319             False        False        False            False
False
10320             False        False        False            False
False
10321             False        False        False            False
False
10322             False        False        False            False
False

       PaperlessBilling  PaymentMethod  MonthlyCharges  TotalCharges
Churn
0                 False          False           False         False
False
1                 False          False           False         False
False
2                 False          False           False         False
False
3                 False          False           False         False
False
4                 False          False           False         False
False
...                 ...            ...             ...           ...
...
10318             False          False           False         False
```

```
False
10319             False         False          False         False
False
10320             False         False          False         False
False
10321             False         False          False         False
False
10322             False         False          False         False
False

[10323 rows x 21 columns]
```

```
df.isnull().sum()
```

```
customerID           0
gender               0
SeniorCitizen        0
Partner              0
Dependents           0
tenure               0
PhoneService         0
MultipleLines        0
InternetService      0
OnlineSecurity       0
OnlineBackup         0
DeviceProtection     0
TechSupport          0
StreamingTV          0
StreamingMovies      0
Contract             0
PaperlessBilling     0
PaymentMethod        0
MonthlyCharges       0
TotalCharges        28
Churn                0
dtype: int64
```

```
df.describe()
```

```
       SeniorCitizen        tenure  MonthlyCharges  TotalCharges
count   10323.000000  10323.000000    10323.000000  10295.000000
mean        0.203139     32.284317       64.818493   2275.535624
std         0.402354     24.574950       30.113270   2264.457186
min         0.000000      0.000000       18.250000      0.000000
25%         0.000000      9.000000       35.450000    398.900000
50%         0.000000     29.000000       70.450000   1389.200000
75%         0.000000     55.000000       89.900000   3780.625000
max         1.000000     72.000000      118.750000   8684.800000
```

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

```
13
```

```
df = df.drop_duplicates(subset="customerID")
```

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

```
0
```

```
duplicate_columns = [col for col in df.columns if 
df[col].duplicated().any()]
print(f"Columns with duplicate values: {duplicate_columns}")
print(f"Number of columns with duplicate values: 
{len(duplicate_columns)}")

df_cleaned = df.drop_duplicates(subset=duplicate_columns)
```

```
Columns with duplicate values: ['gender', 'SeniorCitizen', 'Partner', 
'Dependents', 'tenure', 'PhoneService', 'MultipleLines', 
'InternetService', 'OnlineSecurity', 'OnlineBackup', 
'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies', 
'Contract', 'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges', 
'TotalCharges', 'Churn']
Number of columns with duplicate values: 20
```

```
def conv(value):
    if value ==1:
        return "yes"
    else:
        return "no"
df["SeniorCitizen"] = df["SeniorCitizen"].apply(conv)
```

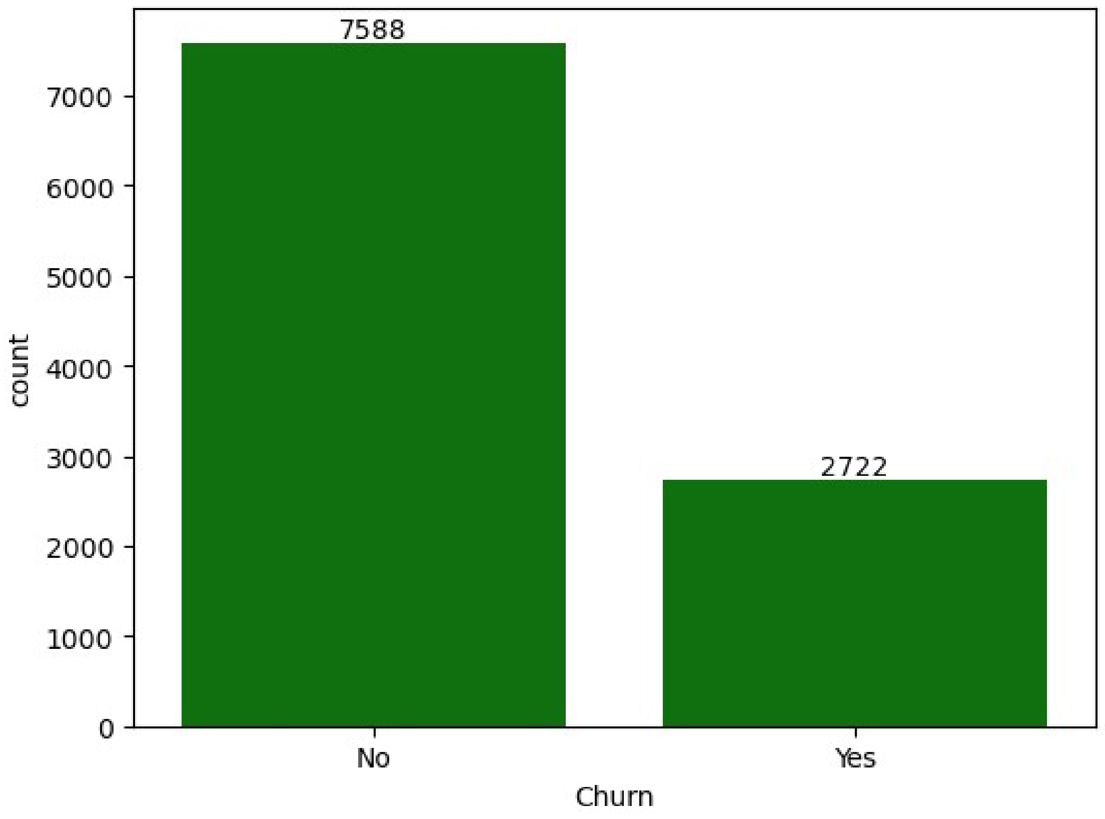
```
C:\Users\user\AppData\Local\Temp\ipykernel_12504\2155067224.py:6: 
SettingWithCopyWarning: 
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: 
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df["SeniorCitizen"] = df["SeniorCitizen"].apply(conv)
```

```
# converting 0 and 1 values of senior citizen to yes/no make it easier 
to undrstand
```

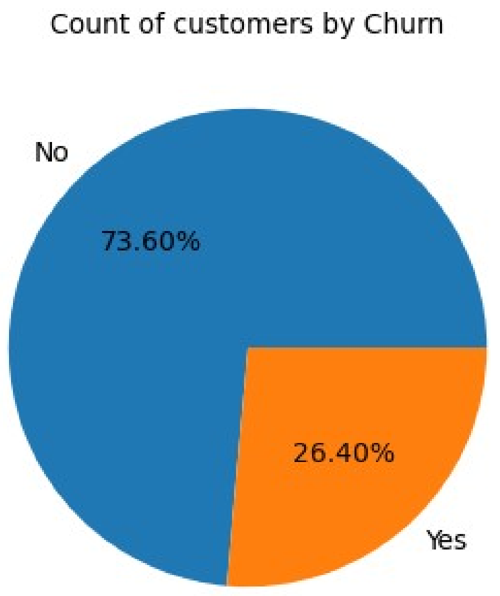
```
df.head()
```

```
   customerID  gender SeniorCitizen Partner Dependents  tenure 
PhoneService  \
0  7590-VHVEG  Female            no     Yes         No       1 
No
```

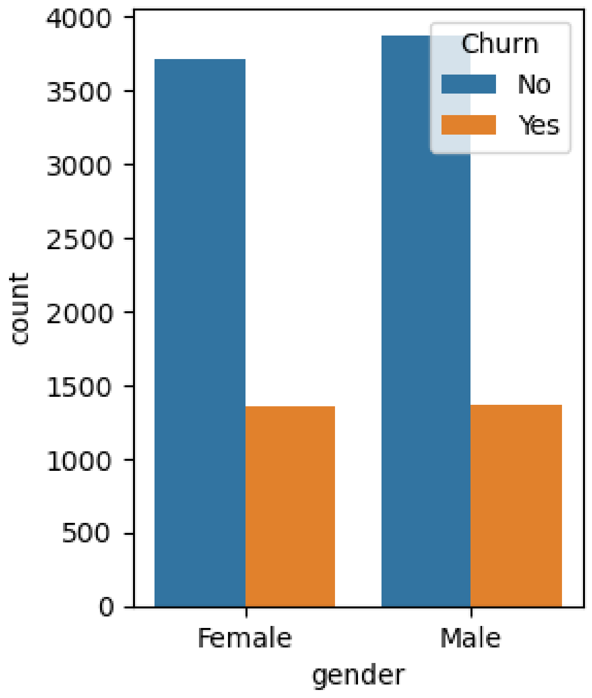
```
1  5575-GNVDE    Male               no       No          No       34
Yes
2  3668-QPYBK    Male               no       No          No        2
Yes
3  7795-CFOCW    Male               no       No          No       45
No
4  9237-HQITU  Female               no       No          No        2
Yes

      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       1889.50     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]

ax = sns.countplot(x = df["Churn"], data = df, color = "green")
ax.bar_label(ax.containers[0])
plt.show()
```



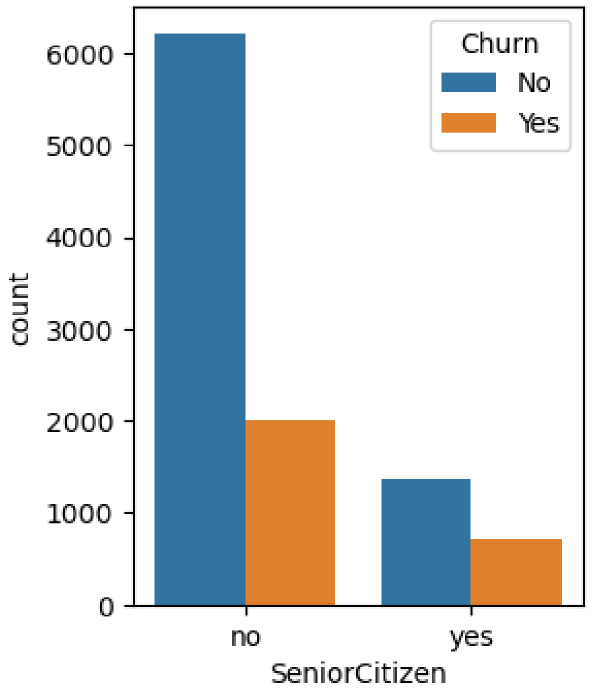
```python
plt.figure(figsize = (4,4))
gb = df.groupby("Churn").agg({"Churn":"count"})
plt.pie(gb["Churn"], labels = gb.index, autopct = "%1.2f%%")
plt.title("Count of customers by Churn", fontsize = 10)
plt.show()
```



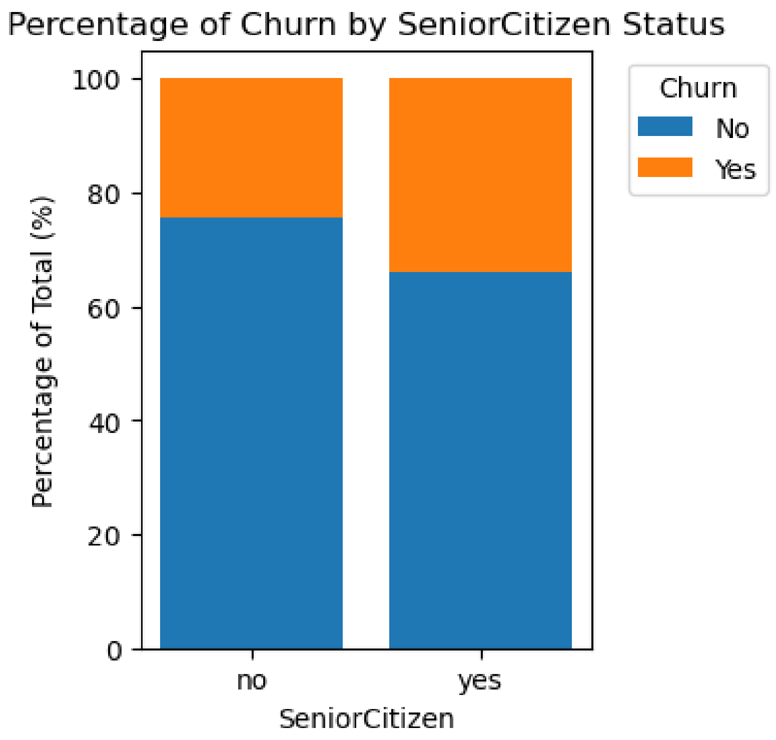
```
# From the given pie chart we can conclude that 25.54% of our
customers have churned
# out not let's explore the reason behind it.

plt.figure(figsize = (3,4))
sns.countplot( x= "gender", data=df, hue = "Churn")
plt.show()
```



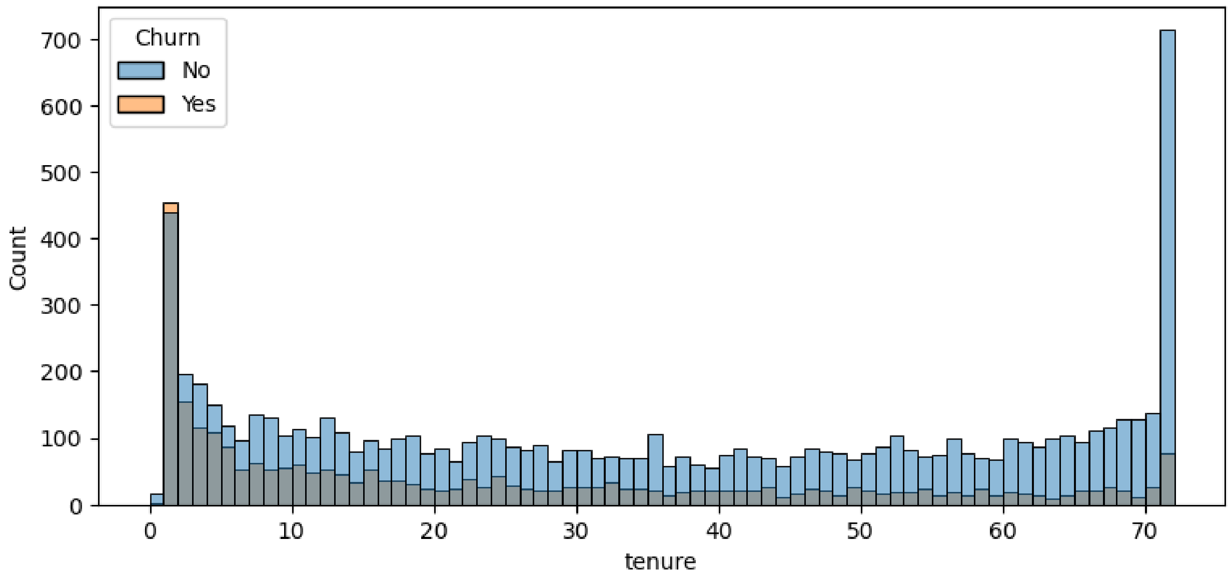
```
plt.figure(figsize = (3,4))
sns.countplot( x= "SeniorCitizen", data=df, hue = "Churn")
plt.show()
```



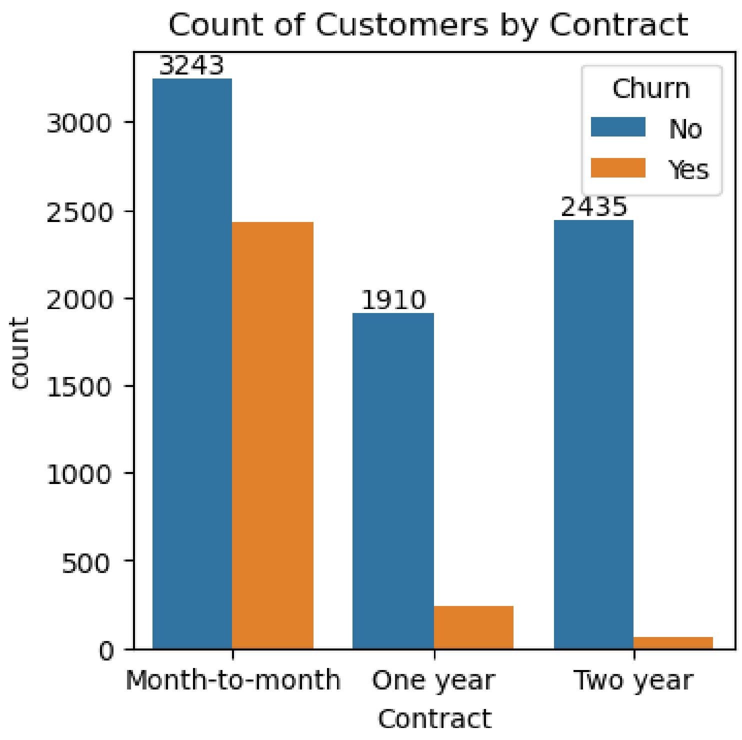
```
# Step 1: Calculate the counts and percentage of total for each
category
# Create a crosstab to count occurrences for each combination of
SeniorCitizen and Churn
count_data = pd.crosstab(df['SeniorCitizen'], df['Churn'])

# Calculate the percentage by dividing each cell by the total counts
per 'SeniorCitizen' row
percentage_data = count_data.div(count_data.sum(axis=1), axis=0) * 100

# Step 2: Plot the stacked bar chart
plt.figure(figsize=(3, 4))

# Plot each section of the stack
bottom = None
for churn_status in percentage_data.columns:
    plt.bar(percentage_data.index, percentage_data[churn_status],
bottom=bottom, label=churn_status)
    bottom = percentage_data[churn_status] if bottom is None else
bottom + percentage_data[churn_status]

# Adding labels and title
plt.xlabel("SeniorCitizen")
plt.ylabel("Percentage of Total (%)")
plt.title("Percentage of Churn by SeniorCitizen Status")
plt.legend(title="Churn", bbox_to_anchor=(1.05, 1), loc='upper left')

# Display the plot
plt.show()
```



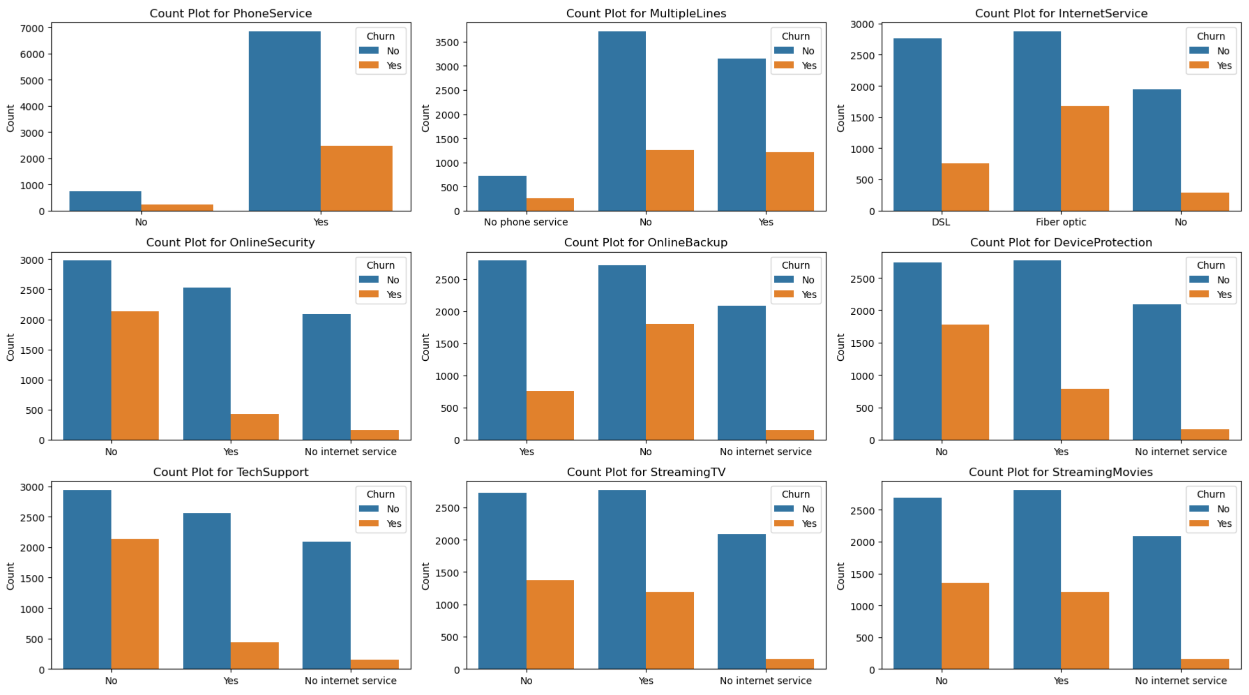
```
plt.figure(figsize = (9, 4))
sns.histplot(x = "tenure", data = df, bins = 72, hue = "Churn")
plt.show()
```



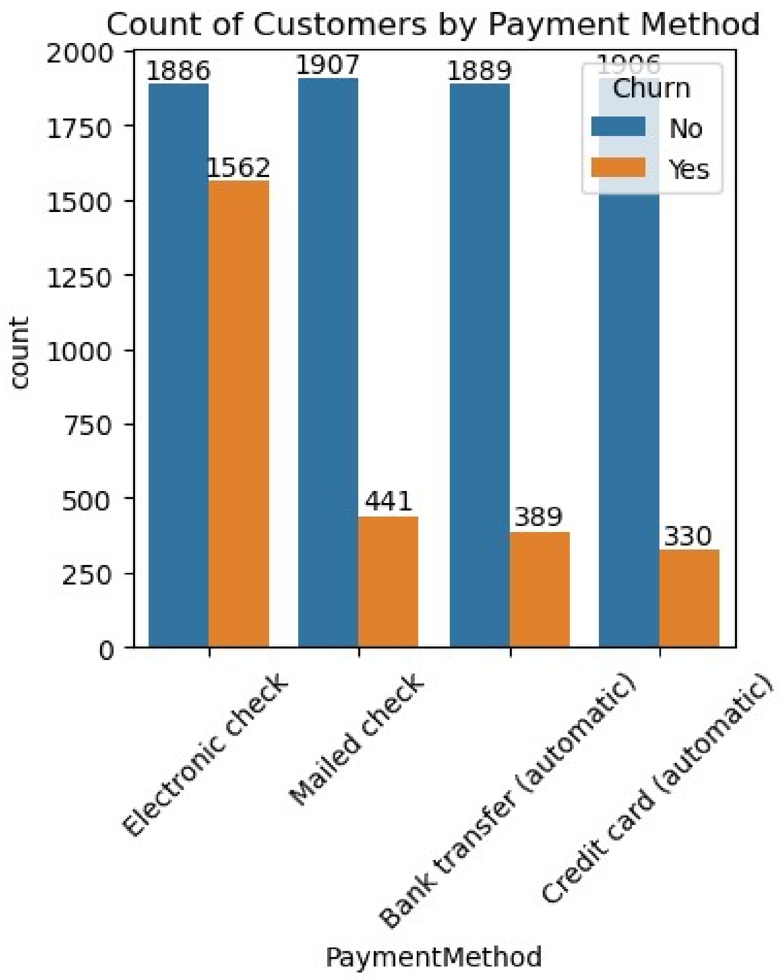
```
# People who have used our services for a long time have statyed and
pople who have used our services # 1 or 2 months have churned

plt.figure(figsize = (4,4))
ax = sns.countplot(x = "Contract", data = df, hue = "Churn")
ax.bar_label(ax.containers[0])
plt.title("Count of Customers by Contract ")
plt.show()
```



```
# Popole who have month to month contract are likely to churn then
form those  who have 1 or 2 years ar contract

df.columns.values

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

# Define the list of columns you want to plot
columns = ['PhoneService', 'MultipleLines', 'InternetService',
'OnlineSecurity',
           'OnlineBackup', 'DeviceProtection', 'TechSupport',
'StreamingTV', 'StreamingMovies']

# Calculate the number of rows and columns for subplots grid
n_cols = 3  # Number of plots per row
n_rows = (len(columns) + n_cols - 1) // n_cols  # Calculate required
number of rows

# Set up the matplotlib figure
fig, axes = plt.subplots(n_rows, n_cols, figsize=(18, 10))
axes = axes.flatten()  # Flatten to easily iterate over

# Create a count plot for each column
for i, column in enumerate(columns):
    sns.countplot(data=df, x=column, ax=axes[i], hue = df["Churn"])
    axes[i].set_title(f'Count Plot for {column}')
    axes[i].set_xlabel('')
    axes[i].set_ylabel('Count')

# Remove any empty subplots if the grid isn't perfectly filled
for j in range(i + 1, len(axes)):
    fig.delaxes(axes[j])

plt.tight_layout()
plt.show()
```



```
# The plots show the distribution of various telecom services among
customers, segmented by churn status. Key observations include that
most customers have phone service, with a significant portion not
opting for multiple lines. Internet services are split, with fiber
optic users showing a higher churn rate. Features like online
security, backup, device protection, tech support, and streaming
services display lower churn for customers who opted out, suggesting
that these additional services might influence churn behavior.

 plt.figure(figsize = (4,4))
ax = sns.countplot(x = "PaymentMethod", data = df, hue = "Churn")
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
plt.title("Count of Customers by Payment Method")
plt.xticks(rotation =45)
plt.show()
```



```
# Customer is likely to churn when he
```