

Import Libraries

```
[1]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Load the dataset

```
[2]: df = pd.read_csv("Customer Churn.csv")
df.head(5)
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechSupport	St
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No	No	
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	Yes	No	
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	No	No	
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	Yes	Yes	
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	No	No	

5 rows × 21 columns

```
[3]: 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
```

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

	tenure	MonthlyCharges	TotalCharges
count	7043.000000	7043.000000	7043.000000
mean	32.371149	64.761692	2279.734304
std	24.559481	30.090047	2266.794470
min	0.000000	18.250000	0.000000
25%	9.000000	35.500000	398.550000
50%	29.000000	70.350000	1394.550000
75%	55.000000	89.850000	3786.600000
max	72.000000	118.750000	8684.800000

```
[7]: # Replace blank with 0 in TotalCharges because 0 Tenure so no Total Charge recorded.
```

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

```
[20]: # Check the Null Values
```

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

```
[20]: 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    0
      Churn           0
      dtype: int64
```

```
[16]: # Check the Duplicated Values

print(df.duplicated().sum())
print(df['customerID'].duplicated().sum())

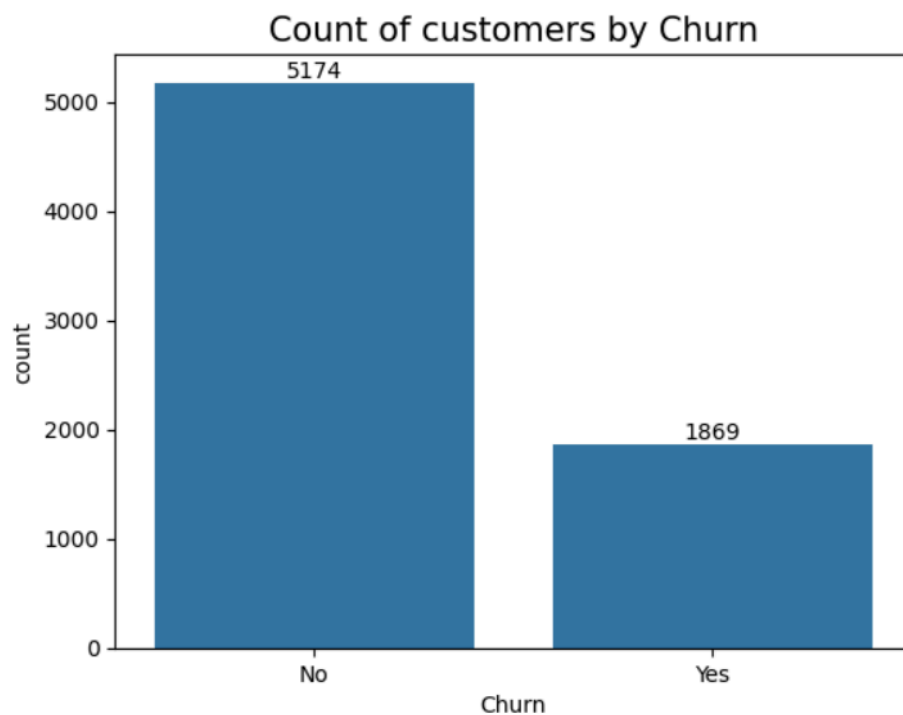
0
0
```

Convert 0 and 1 with yes and no in SeniorCitizen for better understanding

```
[17]: def con(a):
      if a == 1:
          return "yes"
      else:
          return "no"

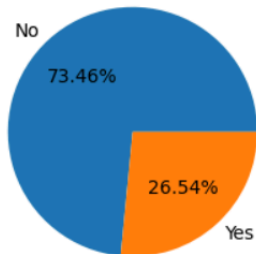
df["SeniorCitizen"] = df["SeniorCitizen"].apply(con)
```

```
[25]: ax = sns.countplot(x = "Churn", data=df)
      plt.bar_label(ax.containers[0])
      plt.title("Count of customers by Churn",size=15)
      plt.show()
```



```
[39]: plt.figure(figsize = (3,4))
ab = df.groupby("Churn").agg({"Churn": "count"})
plt.title("Percentage of Churned Customers", fontsize = 15)
plt.pie(ab["Churn"], labels = ab.index, autopct = "%1.2f%%")
plt.show()
```

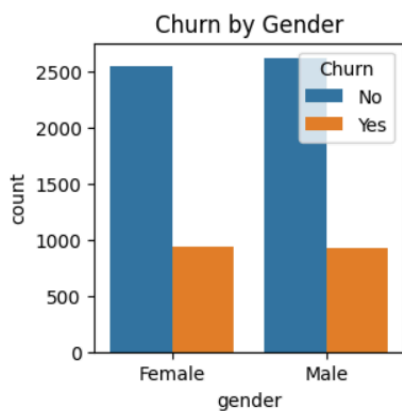
Percentage of Churned Customers



From the above pie chart we can conclude that our 26.54% of customers have Churned out.

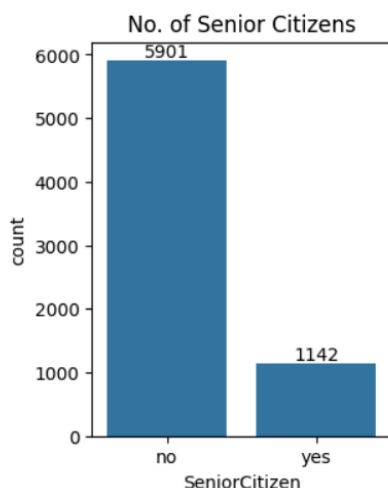
Now find the reason behind it.

```
[44]: plt.figure(figsize = (3,3))
sns.countplot(x = "gender", data=df, hue="Churn")
plt.title("Churn by Gender")
plt.show()
```



Gender does not appear to be a strong factor in predicting customer churn.

```
[49]: plt.figure(figsize = (3,4))
ax = sns.countplot(x = "SeniorCitizen", data=df)
ax.bar_label(ax.containers[0])
plt.title("No. of Senior Citizens")
plt.show()
```



```
[52]: total_counts = df.groupby('SeniorCitizen')['Churn'].value_counts(normalize=True).unstack() * 100

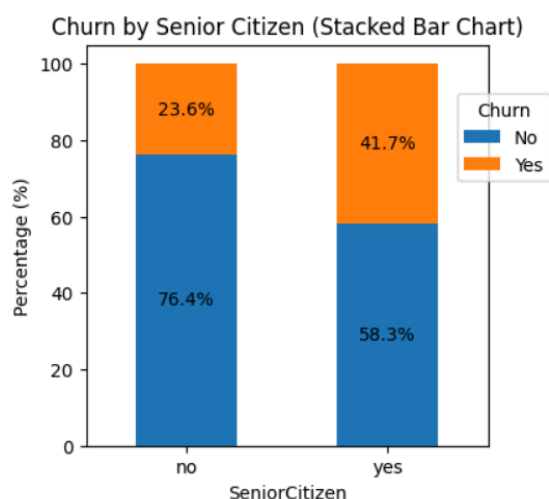
# Plot
fig, ax = plt.subplots(figsize=(4, 4)) # Adjust figsize for better visualization

# Plot the bars
total_counts.plot(kind='bar', stacked=True, ax=ax, color=['#1f77b4', '#ff7f0e']) # Customize colors if desired

# Add percentage labels on the bars
for p in ax.patches:
    width, height = p.get_width(), p.get_height()
    x, y = p.get_xy()
    ax.text(x + width / 2, y + height / 2, f'{height:.1f}%', ha='center', va='center')

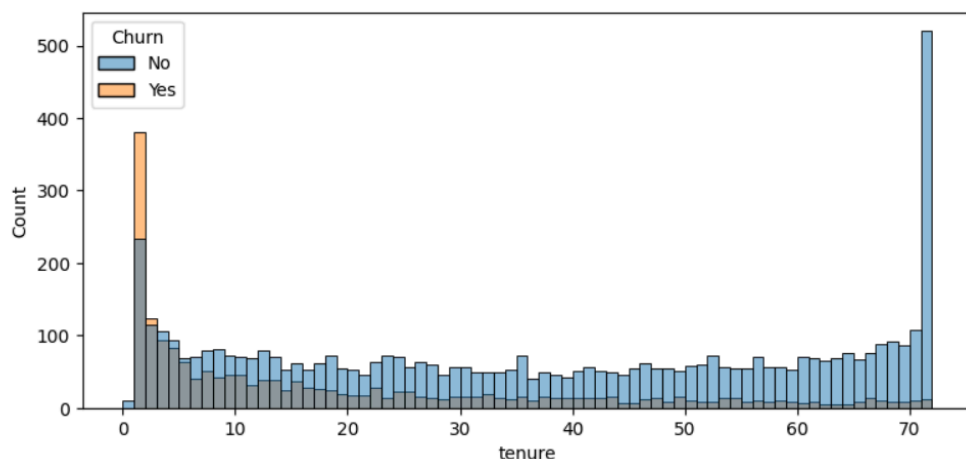
plt.title('Churn by Senior Citizen (Stacked Bar Chart)')
plt.xlabel('SeniorCitizen')
plt.ylabel('Percentage (%)')
plt.xticks(rotation=0)
plt.legend(title='Churn', bbox_to_anchor = (0.9,0.9)) # Customize Legend Location

plt.show()
```



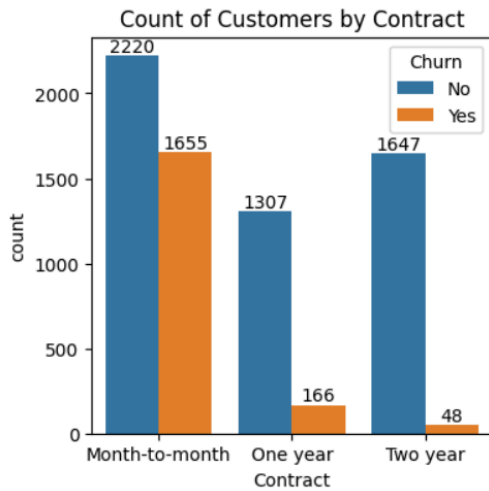
Senior citizens have a higher churn rate compared to non-senior citizens, even though they are fewer in number overall. Senior status may be a significant factor in churn prediction.

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



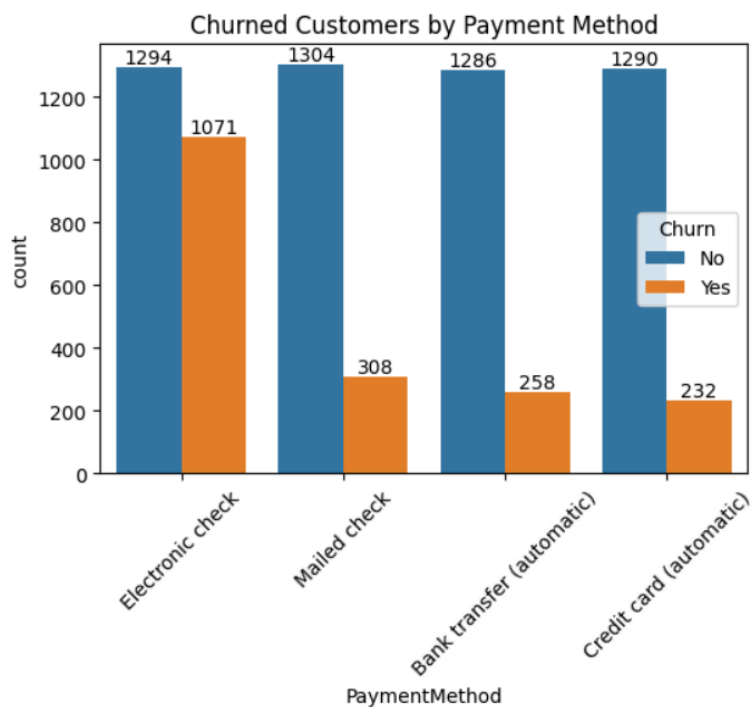
Customers are more likely to churn early in their relationship. Once they stay longer (especially beyond 1-2 years), the likelihood of churn decreases significantly.

```
[55]: plt.figure(figsize = (4,4))
ax = sns.countplot(x = "Contract", data = df, hue = "Churn")
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
plt.title("Count of Customers by Contract")
plt.show()
```



Customers on longer contracts (1 or 2 years) are less likely to churn, while month-to-month customers are at the highest risk of churn.

```
[56]: plt.figure(figsize = (6,4))
ax = sns.countplot(x = "PaymentMethod", data = df, hue = "Churn")
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
plt.title("Churned Customers by Payment Method")
plt.xticks(rotation = 45)
plt.show()
```



Customer is likely to churn when he is using electronic check as a payment method.

```
[57]: columns = ['PhoneService', 'MultipleLines', 'InternetService', 'OnlineSecurity',
               'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies']

# Number of columns for the subplot grid (you can change this)
n_cols = 3
n_rows = (len(columns) + n_cols - 1) // n_cols # Calculate number of rows needed

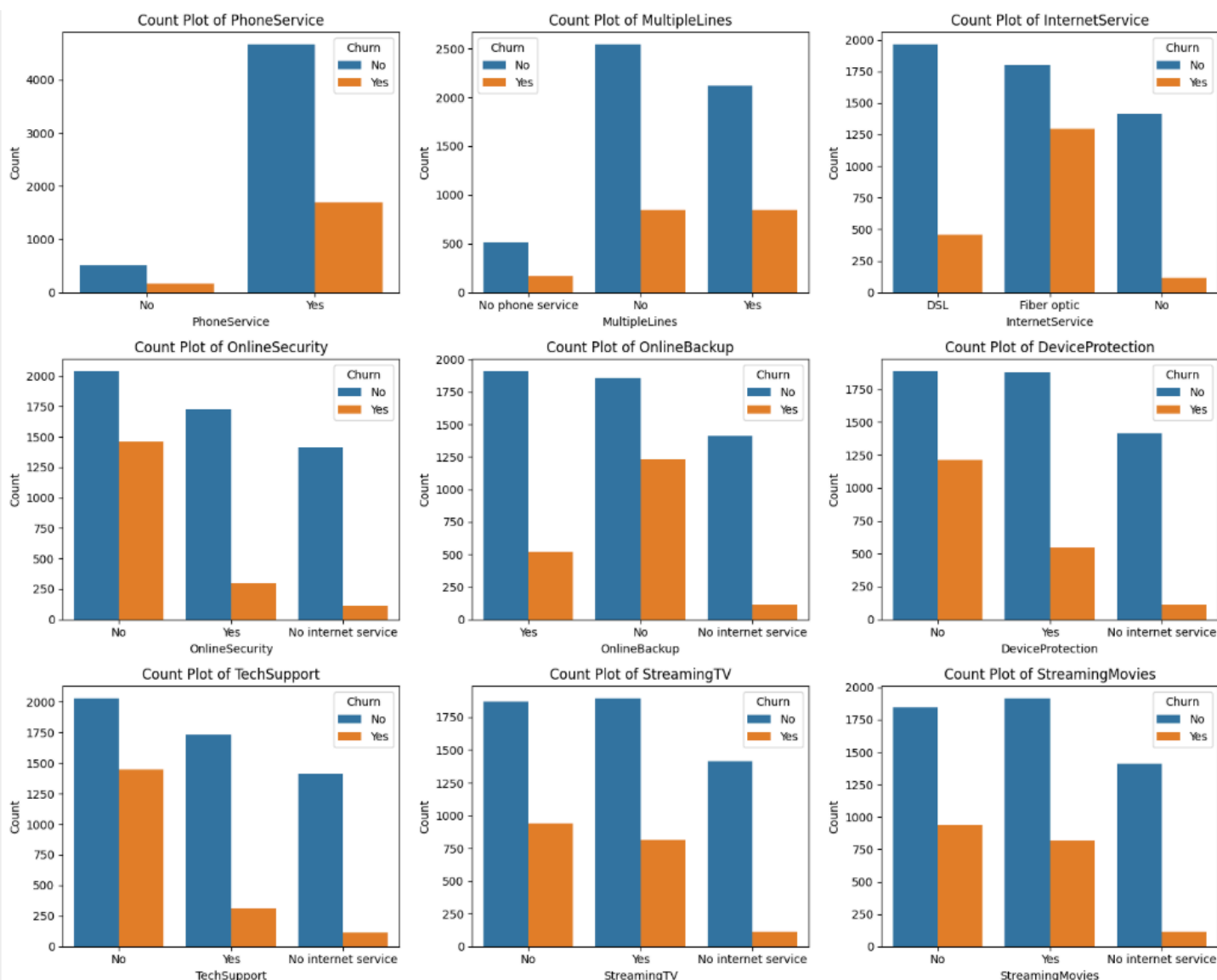
# Create subplots
fig, axes = plt.subplots(n_rows, n_cols, figsize=(15, n_rows * 4)) # Adjust figsize as needed

# Flatten the axes array for easy iteration (handles both 1D and 2D arrays)
axes = axes.flatten()

# Iterate over columns and plot count plots
for i, col in enumerate(columns):
    sns.countplot(x=col, data=df, ax=axes[i], hue = df["Churn"])
    axes[i].set_title(f'Count Plot of {col}')
    axes[i].set_xlabel(col)
    axes[i].set_ylabel('Count')

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

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



The majority of customers who do not churn tend to have services like PhoneService, InternetService (particularly DSL), and OnlineSecurity enabled. For services like OnlineBackup, TechSupport, and StreamingTV, churn rates are noticeably higher when these services are not used or are unavailable.

▼ 🔍 Key Observations:

1. Overall Churn Rate:
About 26.54% of customers have churned.
2. Senior Citizens:
Senior citizens churn more than non-senior citizens.
This segment may need better support, pricing, or engagement.
3. Tenure:
Churn is very high in the first few months.
Longer-tenure customers are more loyal.
Indicates early-stage customer onboarding and experience is critical.
4. Gender:
Gender is not a strong predictor of churn.

▶ 📌 Recommendations to Decrease Churn: 📄

1. 🕒 Improve Early Customer Experience:
Since most churn happens early, offer:
Welcome offers
Personalized onboarding
Proactive support in first 3–6 months
2. 📞 Target Senior Citizens with Tailored Support:
Design senior-friendly plans or bundles.
Provide simplified services or dedicated help channels.
3. 📄 Incentivize Long-Term Contracts:
Offer discounts for 1- or 2-year contracts (as seen in previous charts).
Reduce churn by locking in customers longer.
4. 🛡️ Promote Security & Support Services:
Customers with services like:
OnlineSecurity, TechSupport, DeviceProtection are less likely to churn.
Bundle these with base plans to increase retention.