

# Business Report

Using Data Analysis

Finance Division

# Importing various libraries

## NumPy

A popular open-source library for multi-dimensional data, mathematical functions, and large matrices

## Pandas

A library for data analysis and manipulation that helps clean and preprocess raw data

## Matplotlib

A widget library for plotting 2D graphs with a variety of chart types

- **SciPy:** A library for scientific and technical computing built on top of NumPy

## Importing required libraries

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

## Reading CSV File

```
df = pd.read_csv('Customer.csv')
df.head()
```

	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-CF0CW	Male	0	No	No	45
4	9237-HQITU	Female	0	No	No	2

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

	TechSupport	StreamingTV	StreamingMovies	Contract
0	No	No	No	Month-to-month
1	No	No	No	One year
2	No	No	No	Month-to-month
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]

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
```

## Replacing blank space with 0

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

## Changing data of TotalCharge coloumn to float

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

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
```

## Checking if data set have any null values

```
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        0
Churn               0
dtype: int64

df.describe()
```

	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

Checking any duplicated value in the data set and we used sum get a count of any values if found

```
df["customerID"].duplicated().sum()
0
```

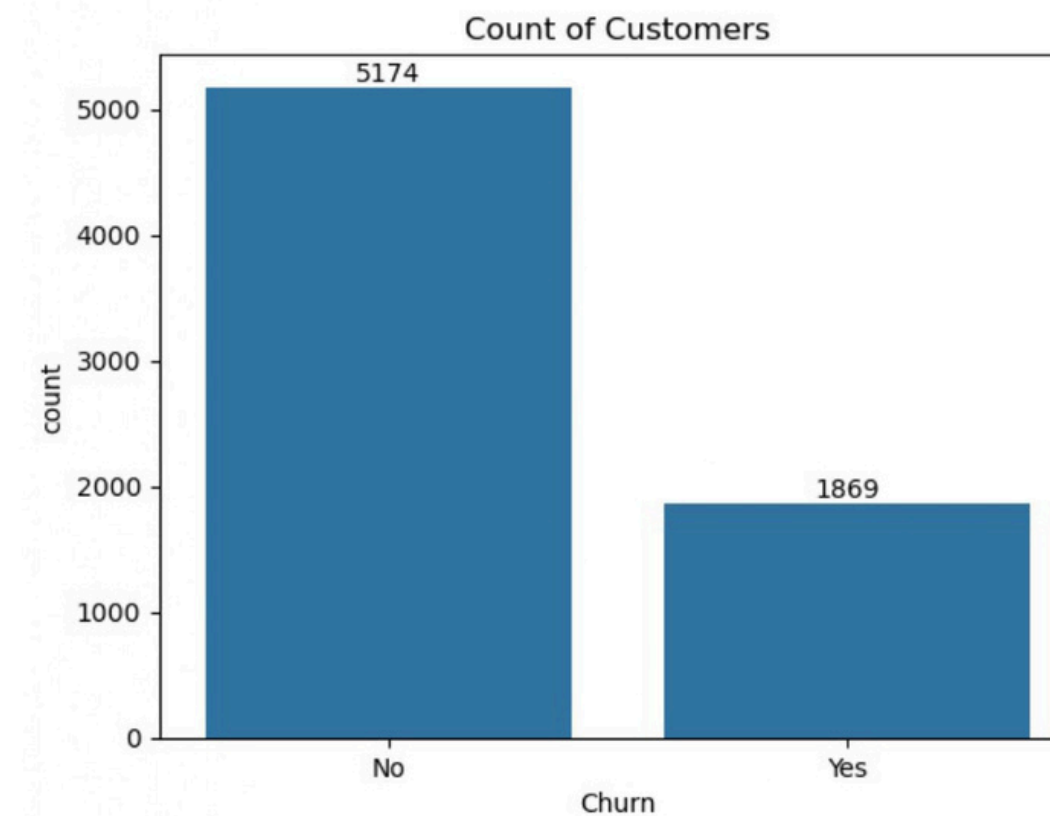
We wrote a function to replace the SeniorCitizen values 0 and 1 to yes and no

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

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

ax = sns.countplot(x = 'Churn', data = df)

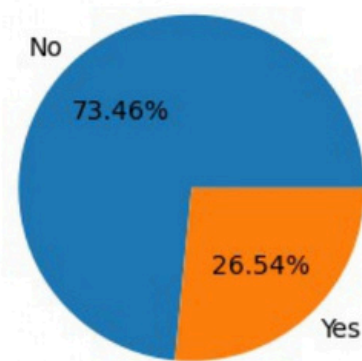
ax.bar_label(ax.containers[0])
plt.title("Count of Customers")
plt.show()
```



In the given pie chart we can say that 26.54% of the customers have churned out. Now let's explore the reason behind it

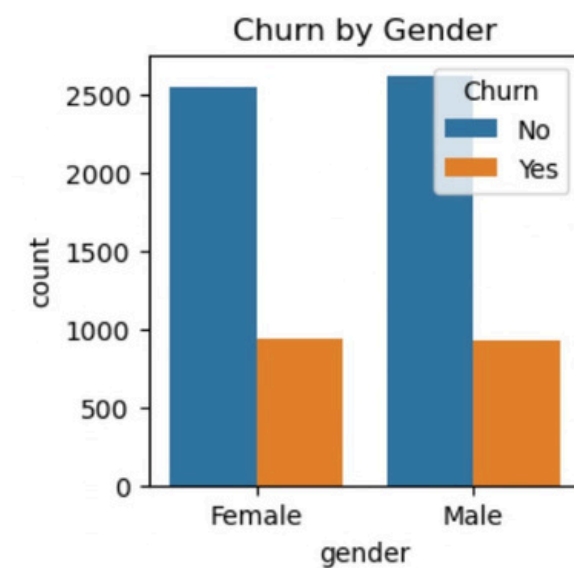
```
plt.figure(figsize = (3,4))
gb = df.groupby("Churn").agg({'Churn':"count"})
plt.pie(gb['Churn'], labels = gb.index, autopct = "%1.2f%%")
plt.title("Percentage of Churned Customeres", fontsize = 10)
plt.show()
```

Percentage of Churned Customeres



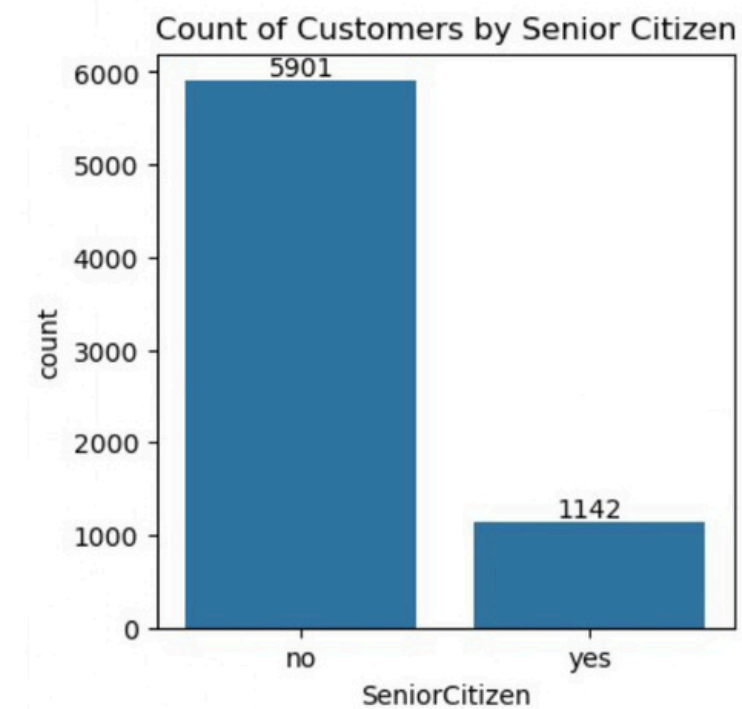
Here we can see by Gender almost the same amount of our customers male and female memeber have charmed out

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



Now lets see what number of Senior Citizen have cherned out

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



We can here that large no. of Senior Citizen have cherned out

```
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:
```

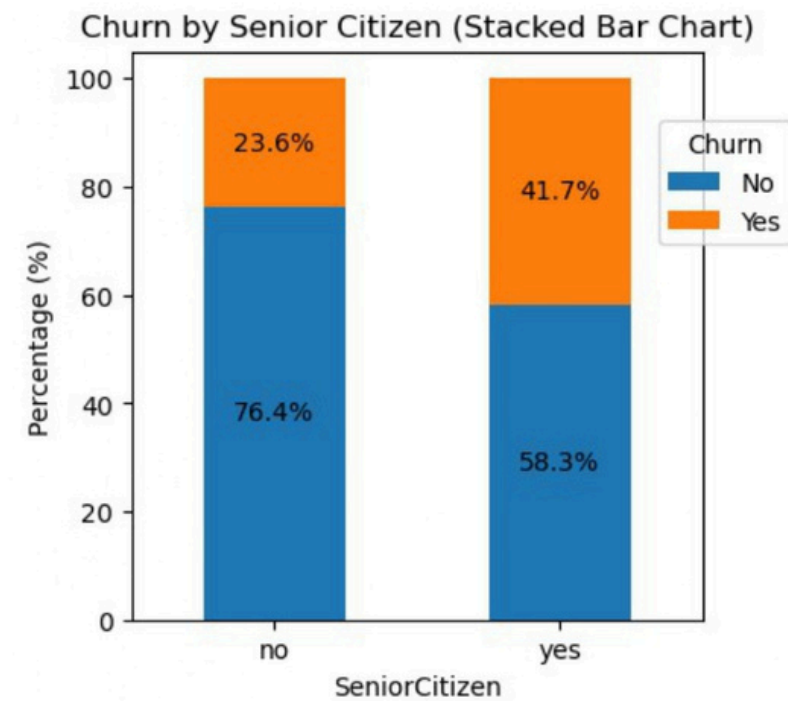
```

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()

```

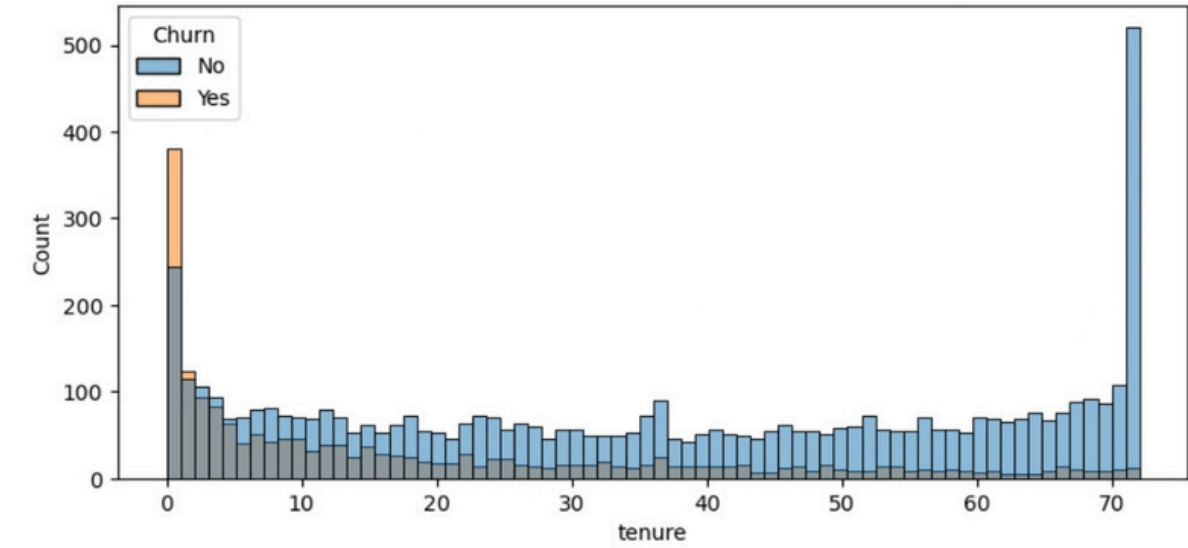


Here we can see most of the members have cherned out in the initial month and long term members are happy

```

plt.figure(figsize = (9,4))
sns.histplot(x = "tenure", data = df, bins = 70, hue = "Churn")
plt.show()

```

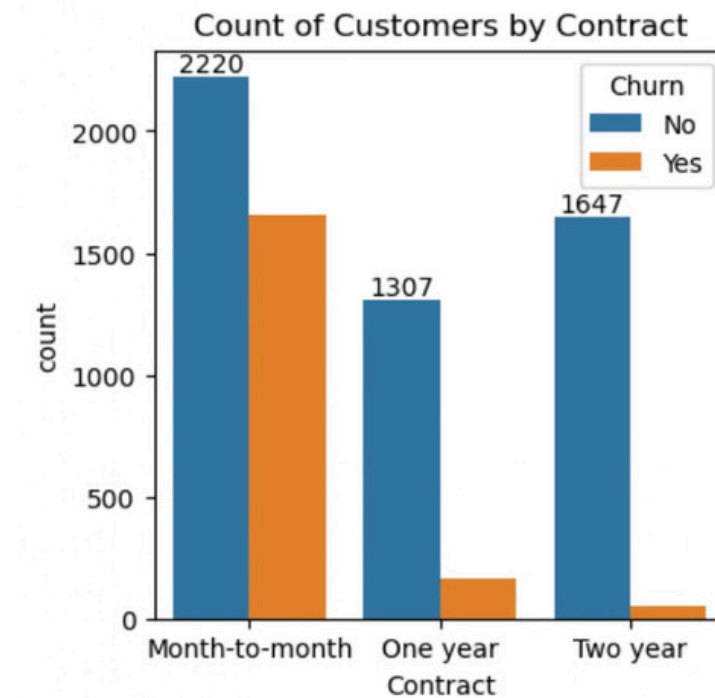


People who have month to month contract are likely to churn then from those who have 1 or 2 years or contract.

```

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()

```





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.

```
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

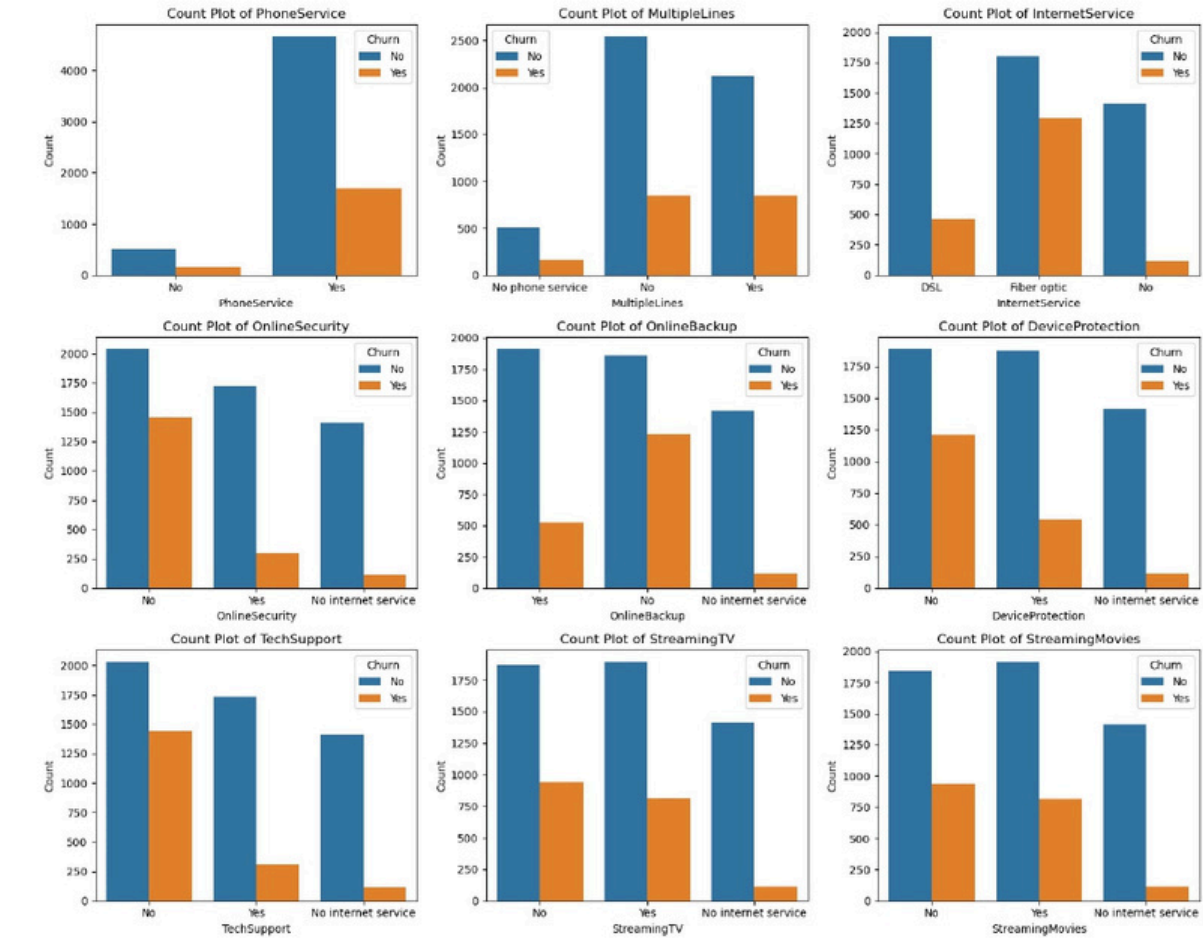
# Create subplots
fig, axes = plt.subplots(n_rows, n_cols, figsize=(15, n_rows * 4)) #
Adjust figsize as needed

# 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])

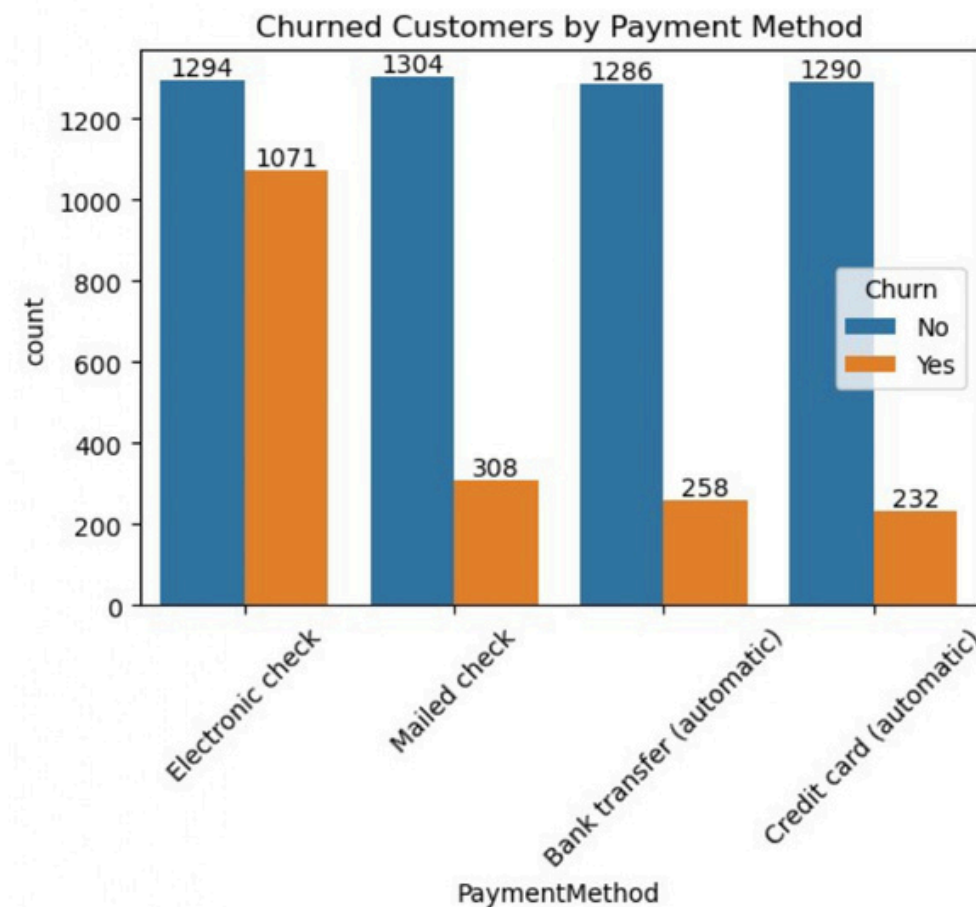
plt.tight_layout()
plt.show()
```



We can see by the payment method Eletronic checked payments has the most member have cherned out

```
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()
```

# End Analysis



- 1. Focus on Early Retention:** Since all churned customers left within the first two months, improving onboarding experiences, personalized engagement, and incentives could enhance early retention.
- 2. Target Senior Customers Above 80 Years Old:** Understanding their needs, providing simplified service plans, and offering better support may reduce their churn.
- 3. Enhance Key Services to Reduce Churn:** Encouraging the use of OnlineBackup, TechSupport, and StreamingTV or improving their quality may help retain more customers.
- 4. Leverage Retention-Associated Services:** Since customers with PhoneService, DSL Internet, and OnlineSecurity are more likely to stay, bundling these services or highlighting their benefits could improve retention.



# Conclusion

The company has experienced a churn of 1,869 customers, while 5,174 customers have remained. The churned customers were evenly split between male and female demographics, indicating no significant gender-based influence on churn.

A deeper look into senior citizens reveals that 1,142 of them left, whereas 5,901 stayed. Notably, all senior citizens who churned were above 80 years old, suggesting that age played a major role in customer attrition.

A critical finding is that all customers who left had only been with the company for 1 or 2 months, implying that early customer experience and retention strategies may need improvement. In contrast, customers who stayed beyond the initial period showed a much higher likelihood of remaining loyal.

Regarding service preferences, customers who did not churn tend to use PhoneService, InternetService (especially DSL), and OnlineSecurity. These services likely contribute to higher customer satisfaction and engagement. On the other hand, churn rates were noticeably higher among those who did not use or did not have access to OnlineBackup, TechSupport, and StreamingTV, suggesting that customers value these additional services and may leave if they are unavailable or not beneficial.