SS_Task2

November 22, 2025

1 TASK 2 — Exploratory Data Analysis (EDA)

1. Loading the Dataset

1.1 Import Libraries & Load Dataset from GitHub

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

sns.set(style="whitegrid")
plt.rcParams["figure.figsize"] = (10, 6)

# GitHub RAW link
url = "https://raw.githubusercontent.com/ShivaHariny07/SaiKet_System_Internship/
 ↪main/Telco_Customer_Churn_Dataset%20%20(3).csv"

# Load dataset
data = pd.read_csv(url)
print("  Dataset Loaded Successfully!")
```

```
 Dataset Loaded Successfully!
```

1.2 Display First 5 Rows

[2]:
```
print("  First Five Rows:\n")
data.head()
```

```
 First Five Rows:
```

[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
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]
```

2. Exploratory Data Analysis

2.1 Calculate Overall Churn Rate

[3]:
```python
churn_rate = data["Churn"].value_counts(normalize=True) * 100
print("  Churn Rate (%):\n")
print(churn_rate)
```

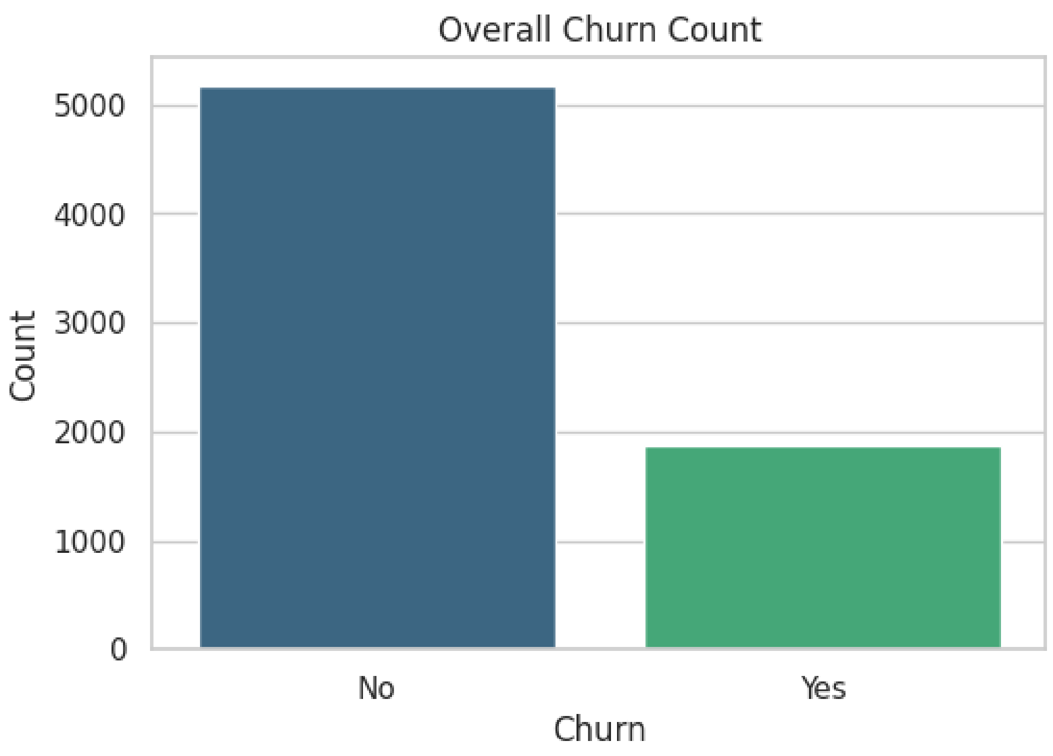
```
  Churn Rate (%):

Churn
No     73.463013
Yes    26.536987
Name: proportion, dtype: float64
```

Visualize Churn Rate

[4]:
```python
plt.figure(figsize=(6, 4))
sns.countplot(x="Churn", data=data, palette="viridis")
plt.title("Overall Churn Count")
plt.xlabel("Churn")
plt.ylabel("Count")
plt.show()
```

```
/tmp/ipython-input-2395782218.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
```

```
effect.

  sns.countplot(x="Churn", data=data, palette="viridis")
```

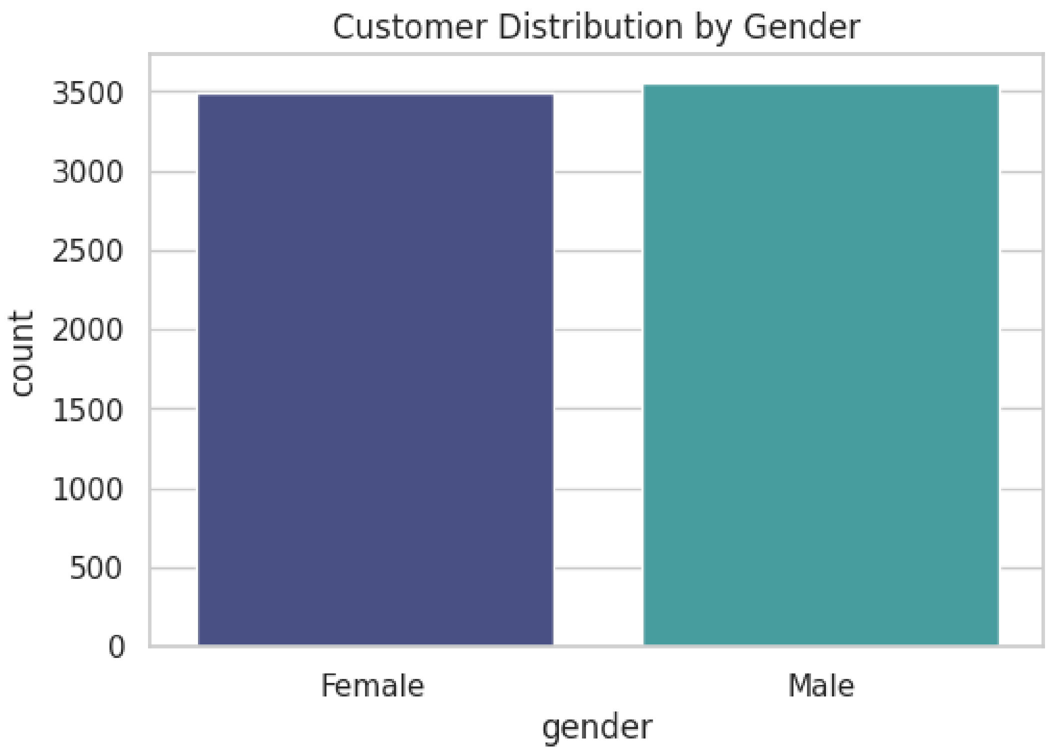


2.2 Customer Distribution by Demographics

Gender Distribution

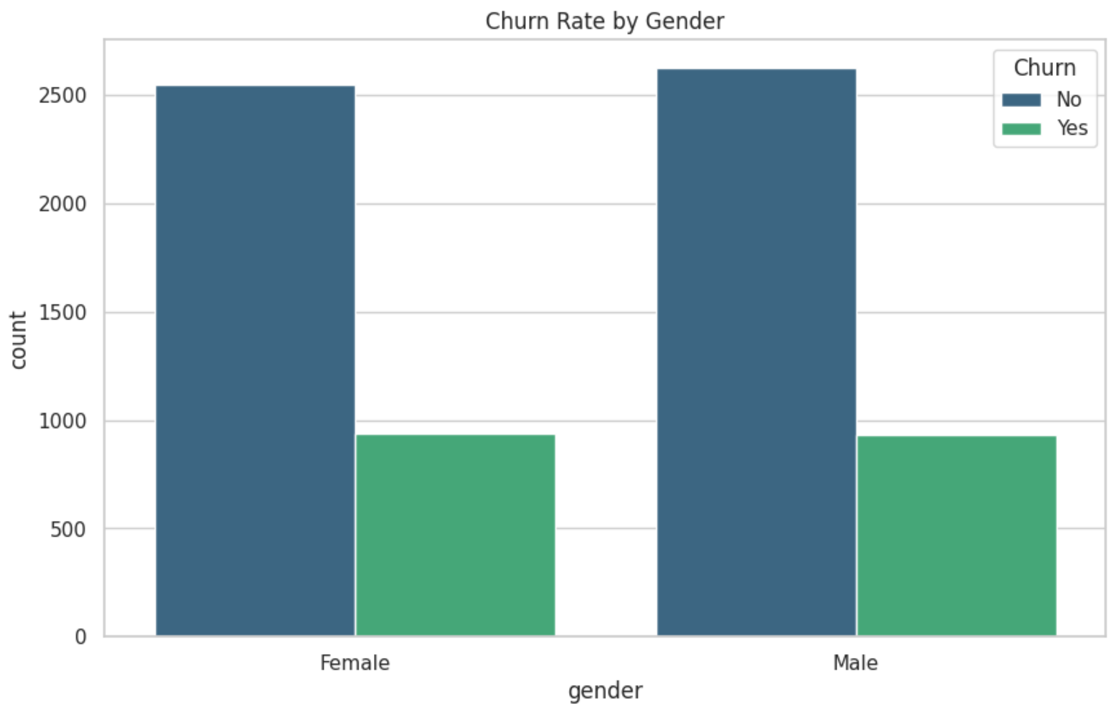
```
[5]: plt.figure(figsize=(6, 4))
     sns.countplot(x="gender", data=data, palette="mako")
     plt.title("Customer Distribution by Gender")
     plt.show()
```

```
/tmp/ipython-input-1604685517.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.countplot(x="gender", data=data, palette="mako")
```



Gender vs Churn

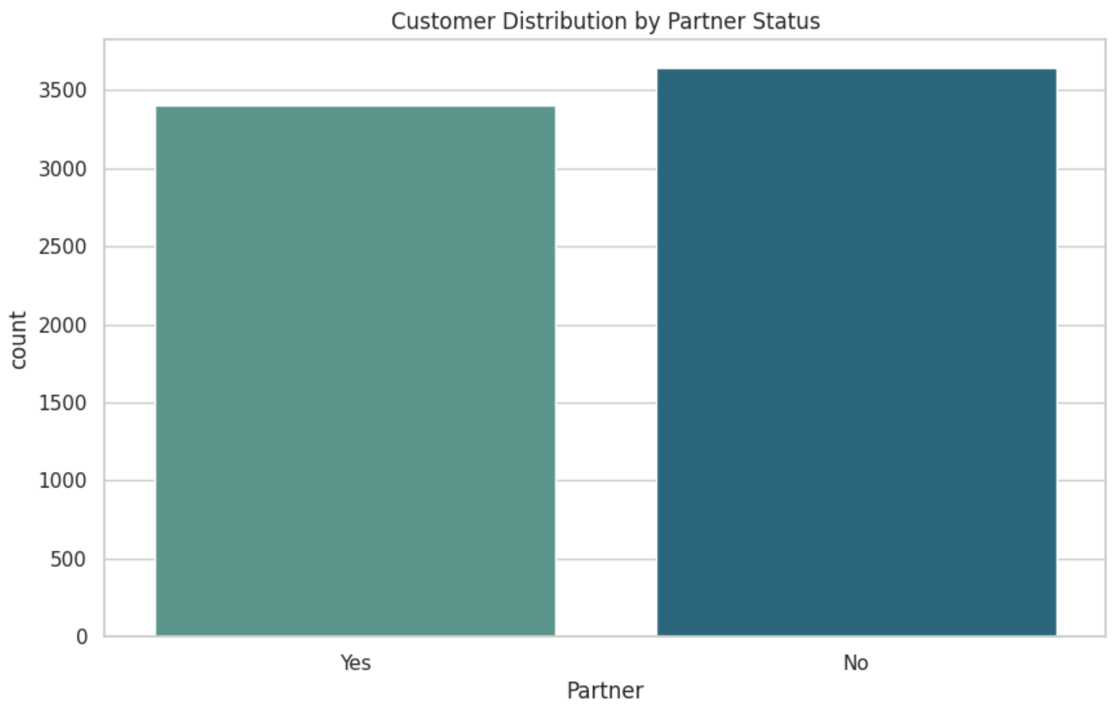
[6]:
```
sns.countplot(x="gender", hue="Churn", data=data, palette="viridis")
plt.title("Churn Rate by Gender")
plt.show()
```



Partner Status Distribution

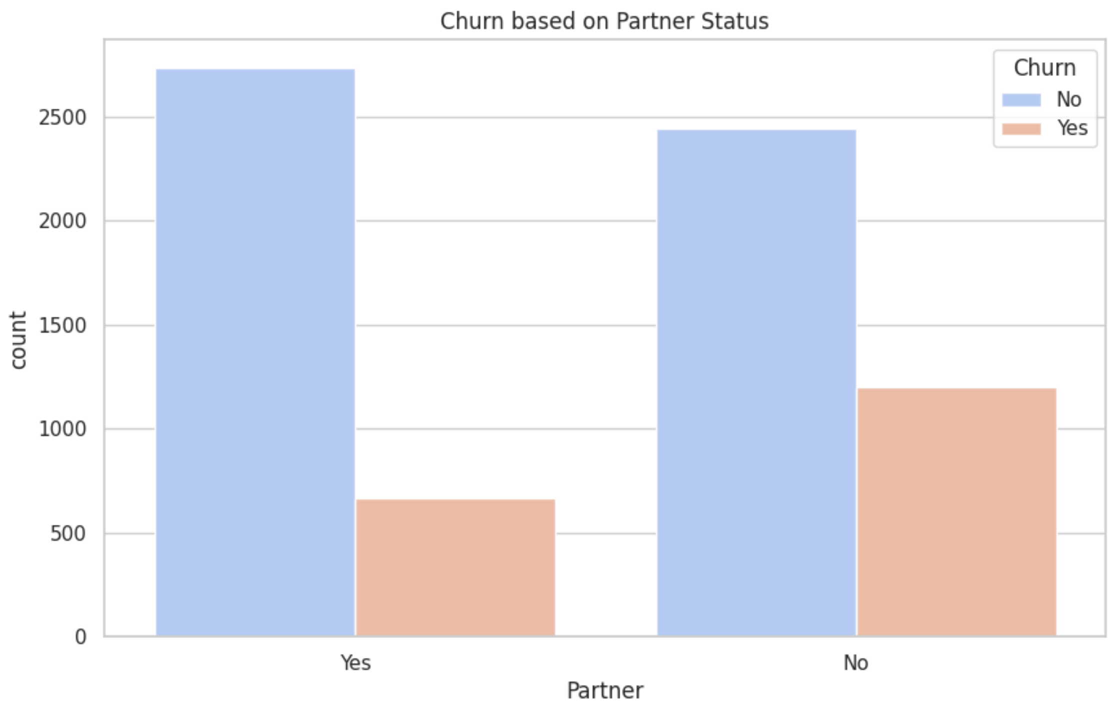
```
[7]: sns.countplot(x="Partner", data=data, palette="crest")
     plt.title("Customer Distribution by Partner Status")
     plt.show()
```

```
/tmp/ipython-input-1283863235.py:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.countplot(x="Partner", data=data, palette="crest")
```



Partner Status vs Churn

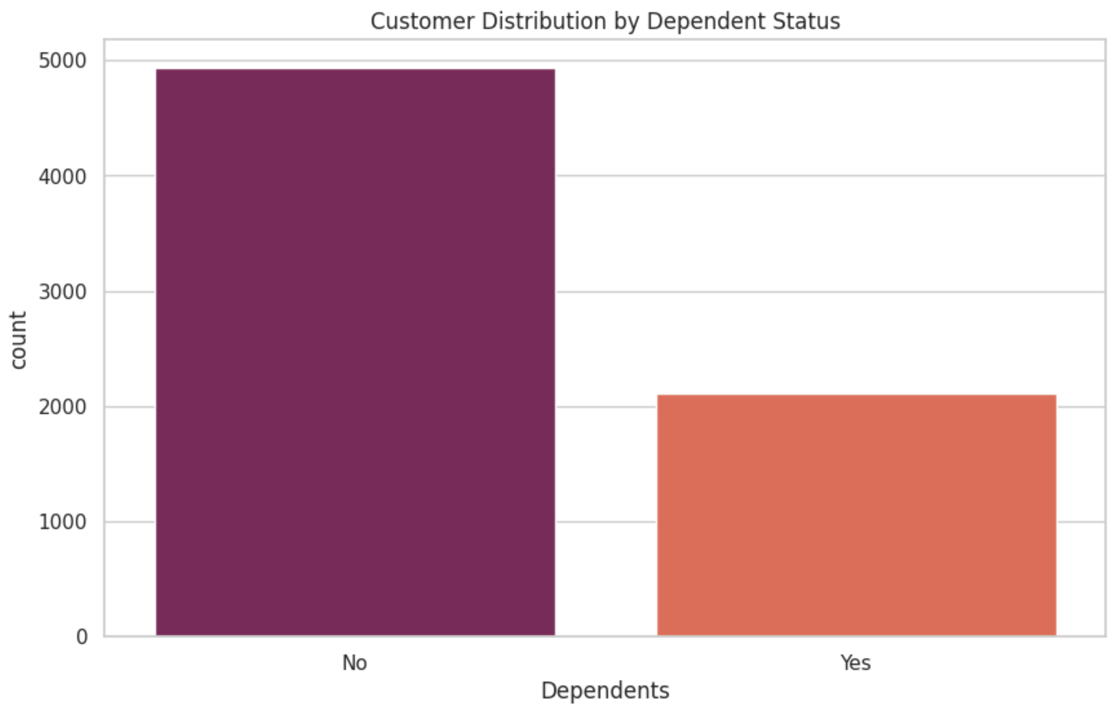
[8]:
```python
sns.countplot(x="Partner", hue="Churn", data=data, palette="coolwarm")
plt.title("Churn based on Partner Status")
plt.show()
```



Dependents Distribution

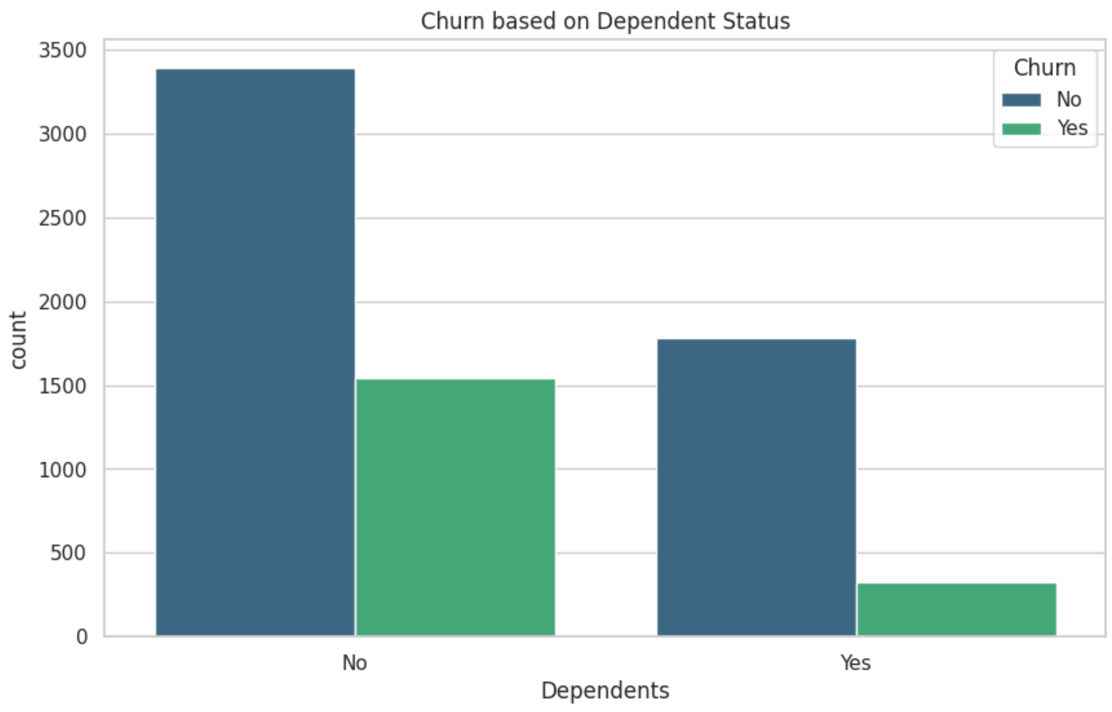
```
[9]: sns.countplot(x="Dependents", data=data, palette="rocket")
     plt.title("Customer Distribution by Dependent Status")
     plt.show()
```

```
/tmp/ipython-input-1584975736.py:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.countplot(x="Dependents", data=data, palette="rocket")
```



Dependents vs Churn

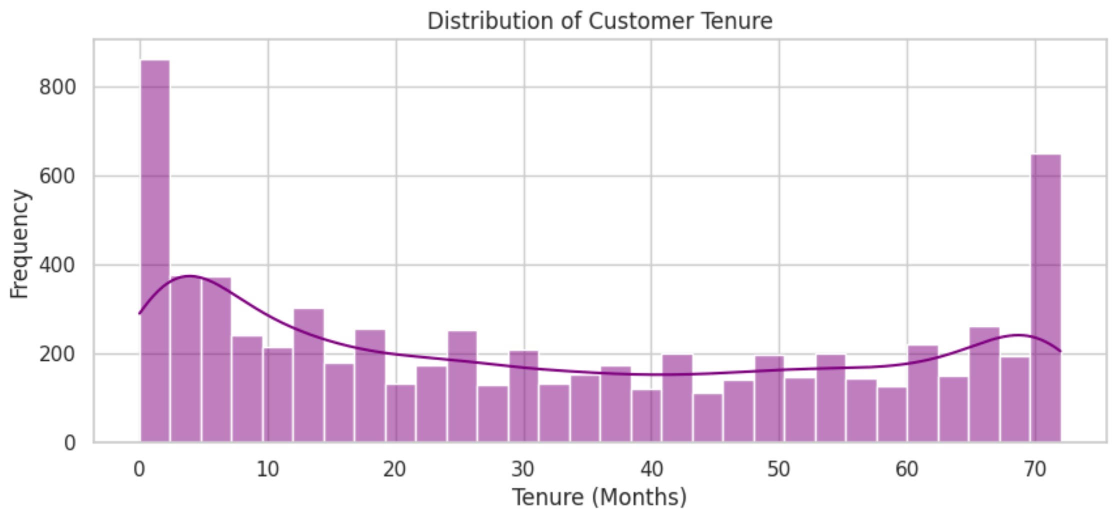
```
[10]: sns.countplot(x="Dependents", hue="Churn", data=data, palette="viridis")
      plt.title("Churn based on Dependent Status")
      plt.show()
```



2.3 Tenure Distribution and Relation With Churn

Tenure Histogram

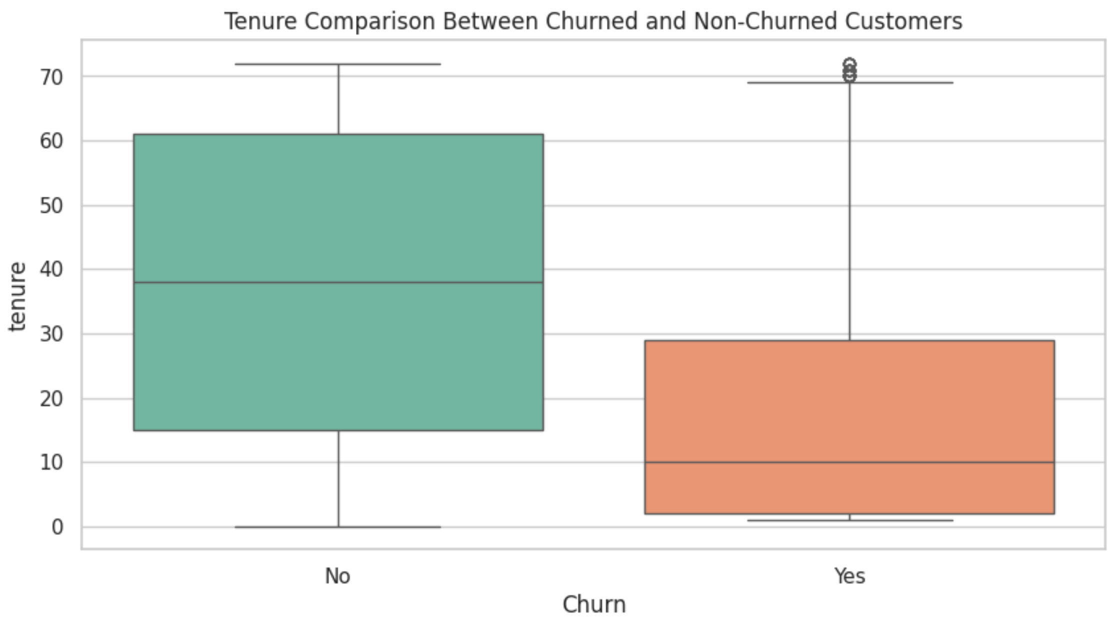
```
[11]: plt.figure(figsize=(10, 4))
sns.histplot(data["tenure"], kde=True, bins=30, color="purple")
plt.title("Distribution of Customer Tenure")
plt.xlabel("Tenure (Months)")
plt.ylabel("Frequency")
plt.show()
```



Tenure vs Churn

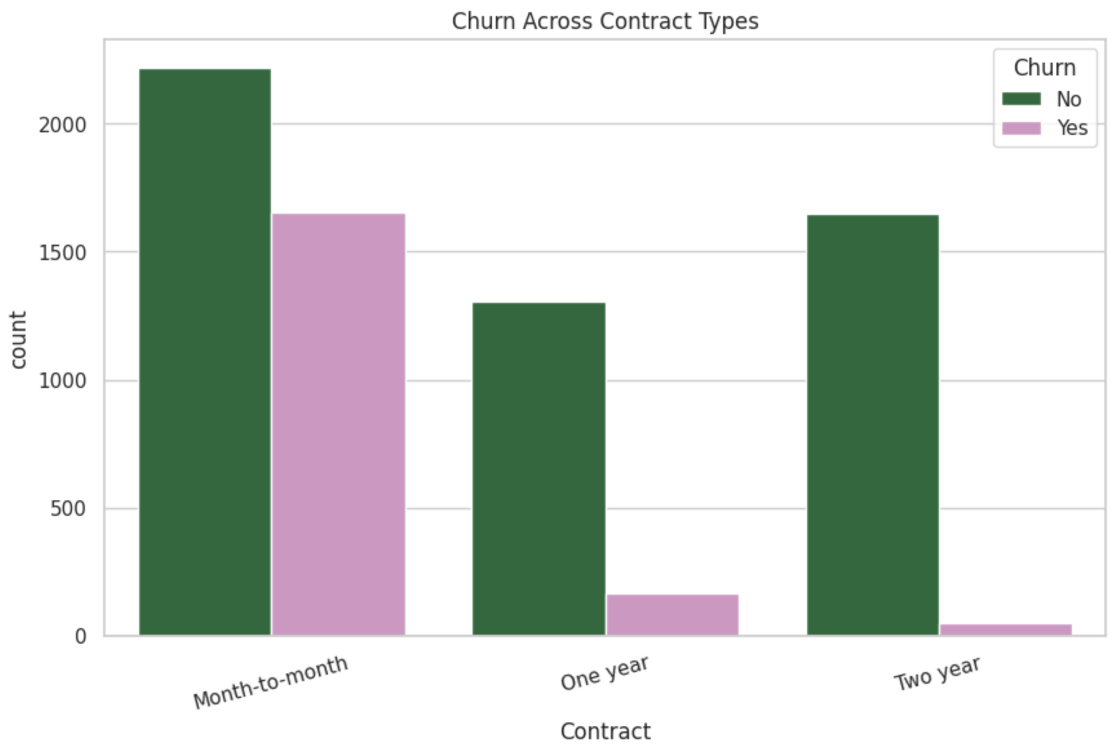
```
[12]: plt.figure(figsize=(10, 5))
      sns.boxplot(x="Churn", y="tenure", data=data, palette="Set2")
      plt.title("Tenure Comparison Between Churned and Non-Churned Customers")
      plt.show()
```

```
/tmp/ipython-input-1753422553.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.boxplot(x="Churn", y="tenure", data=data, palette="Set2")
```



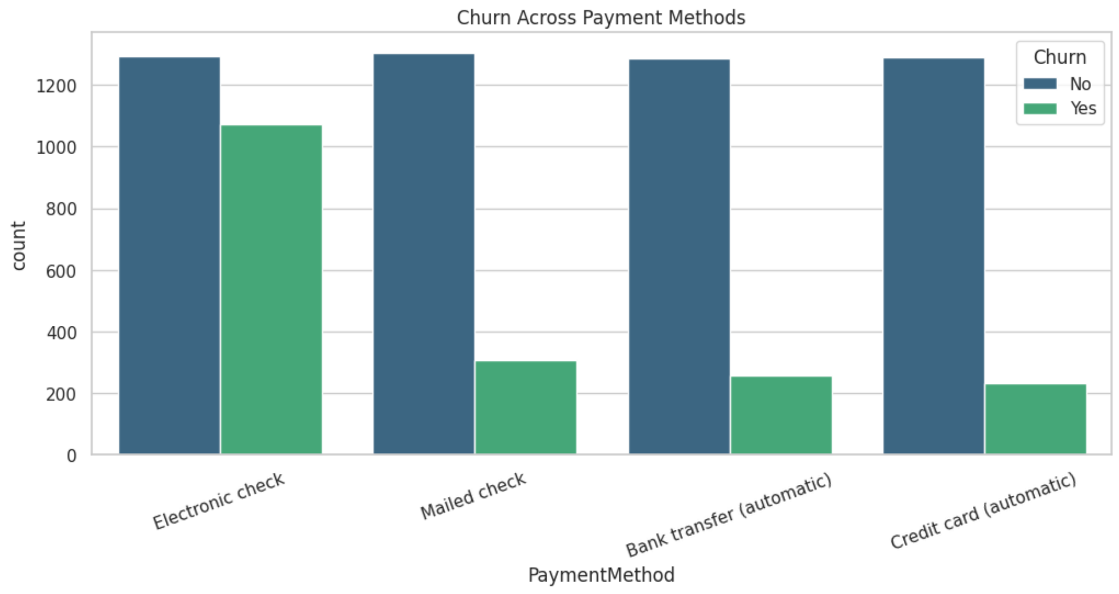
2.4 Churn Across Contract Types

```
[13]: sns.countplot(x="Contract", hue="Churn", data=data, palette="cubehelix")
      plt.title("Churn Across Contract Types")
      plt.xticks(rotation=15)
      plt.show()
```



2.5 Churn Across Payment Methods

```
[14]: plt.figure(figsize=(12, 5))
sns.countplot(x="PaymentMethod", hue="Churn", data=data, palette="viridis")
plt.title("Churn Across Payment Methods")
plt.xticks(rotation=20)
plt.show()
```


Churn Across Payment Methods
Churn
No
Yes
count
0
200
400
600
800
1000
1200
Electronic check
Mailed check
Bank transfer (automatic)
Credit card (automatic)
PaymentMethod