01_Exploration

January 29, 2026

Connected to Python 3.10.11

1 Exploratory Data Analysis (EDA)

The goal of this analysis is to understand customer churn behavior, assess data quality, explore feature distributions, and identify potential predictors of churn. This EDA is intended to inform preprocessing decisions, feature engineering, and model selection in later stages.

[ ]:
```
# Load Libraries
import kagglehub
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
```

[ ]:
```
# Set Working Directory
folder_path = os.path.dirname(os.path.realpath(__file__))[:-10] # on .py, set␣
 ↪the working directory to the script location
# folder_path = os.getcwd()[:-10] # on jupyter, set the working directory to␣
 ↪the notebook location

plt.style.use('ggplot')

# Colour Palette

# import matplotlib as mpl
# mpl.rcParams['axes.prop_cycle'].by_key()['color']
# print(mpl.rcParams['axes.prop_cycle'].by_key()['color'])
palette = [
'#E24A33',  #(orange)
'#348ABD',  #(blue)
'#8EBA42',   #(green)
'#988ED5',  #(cyan)
'#777777',  #(grey)
'#FBC15E',  #(yellow)
'#FFB5B8'    #(pink)
]
```

[ ]:
```
# Download and Save Dataset (Uncomment to run)
# data_path = kagglehub.dataset_download("ankitverma2010/
 ↪ecommerce-customer-churn-analysis-and-prediction")
# print("Path to dataset files:", data_path)

# info_df = pd.read_excel(data_path + '\\E Commerce Dataset.xlsx',␣
 ↪sheet_name='Data Dict', skiprows=1)[['Data','Variable','Discerption']].
 ↪rename(columns={'Discerption':'Description'})
# df = pd.read_excel(data_path + '\\E Commerce Dataset.xlsx', sheet_name='E␣
 ↪Comm', index_col='CustomerID')

# df.to_csv(folder_path + '\\data\\ecommerce_churn_data.csv')
# info_df.to_csv(folder_path + '\\data\\ecommerce_churn_data_info.csv')

# print("Data and info files saved to data directory.")
```

[ ]:
```
# Load Dataset
df = pd.read_csv(folder_path + '/data/ecommerce_churn_data.csv',␣
 ↪index_col='CustomerID')
print(df.head())

print("\nDataset Info:")
print(df.info())

print("\nMissing Values:")
print(df.isnull().sum())
```

```
            Churn  Tenure PreferredLoginDevice  CityTier  WarehouseToHome  \
CustomerID
50001           1     4.0         Mobile Phone         3              6.0
50002           1     NaN                Phone         1              8.0
50003           1     NaN                Phone         1             30.0
50004           1     0.0                Phone         3             15.0
50005           1     0.0                Phone         1             12.0

           PreferredPaymentMode  Gender  HourSpendOnApp  \
CustomerID
50001                Debit Card  Female             3.0
50002                       UPI    Male             3.0
50003                Debit Card    Male             2.0
50004                Debit Card    Male             2.0
50005                        CC    Male             NaN

            NumberOfDeviceRegistered    PreferedOrderCat  SatisfactionScore  \
CustomerID
50001                              3  Laptop & Accessory                  2
50002                              4              Mobile                  3
```

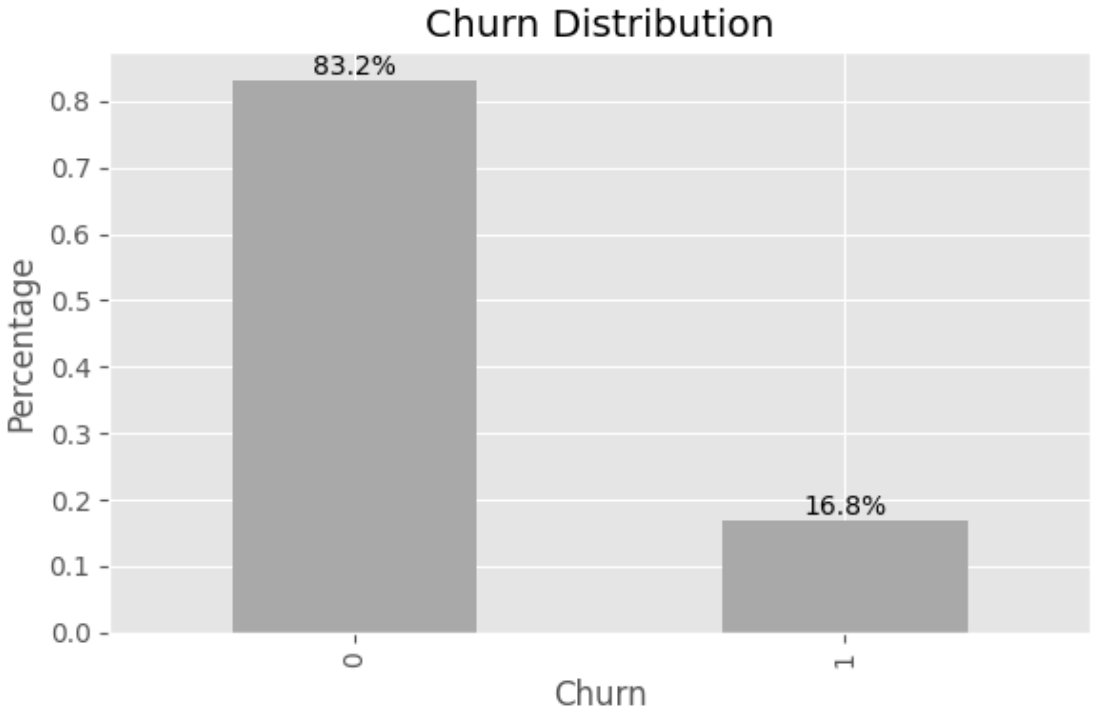
```
50003                            4              Mobile                 3
50004                            4  Laptop & Accessory                 5
50005                            3              Mobile                 5

           MaritalStatus  NumberOfAddress  Complain  \
CustomerID
50001             Single                9         1
50002             Single                7         1
50003             Single                6         1
50004             Single                8         0
50005             Single                3         0

            OrderAmountHikeFromlastYear  CouponUsed  OrderCount  \
CustomerID
50001                              11.0         1.0         1.0
50002                              15.0         0.0         1.0
50003                              14.0         0.0         1.0
50004                              23.0         0.0         1.0
50005                              11.0         1.0         1.0

            DaySinceLastOrder  CashbackAmount
CustomerID
50001                     5.0          159.93
50002                     0.0          120.90
50003                     3.0          120.28
50004                     3.0          134.07
50005                     3.0          129.60

Dataset Info:
<class 'pandas.core.frame.DataFrame'>
Index: 5630 entries, 50001 to 55630
Data columns (total 19 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   Churn                        5630 non-null   int64
 1   Tenure                       5366 non-null   float64
 2   PreferredLoginDevice         5630 non-null   object
 3   CityTier                     5630 non-null   int64
 4   WarehouseToHome              5379 non-null   float64
 5   PreferredPaymentMode         5630 non-null   object
 6   Gender                       5630 non-null   object
 7   HourSpendOnApp               5375 non-null   float64
 8   NumberOfDeviceRegistered     5630 non-null   int64
 9   PreferedOrderCat             5630 non-null   object
 10  SatisfactionScore            5630 non-null   int64
 11  MaritalStatus                5630 non-null   object
 12  NumberOfAddress              5630 non-null   int64
 13  Complain                     5630 non-null   int64
```

```
 14  OrderAmountHikeFromlastYear  5365 non-null   float64
 15  CouponUsed                   5374 non-null   float64
 16  OrderCount                   5372 non-null   float64
 17  DaySinceLastOrder            5323 non-null   float64
 18  CashbackAmount               5630 non-null   float64
dtypes: float64(8), int64(6), object(5)
memory usage: 879.7+ KB
None

Missing Values:
Churn                            0
Tenure                         264
PreferredLoginDevice             0
CityTier                         0
WarehouseToHome                251
PreferredPaymentMode             0
Gender                           0
HourSpendOnApp                 255
NumberOfDeviceRegistered         0
PreferedOrderCat                 0
SatisfactionScore                0
MaritalStatus                    0
NumberOfAddress                  0
Complain                         0
OrderAmountHikeFromlastYear    265
CouponUsed                     256
OrderCount                     258
DaySinceLastOrder              307
CashbackAmount                   0
dtype: int64
```

[ ]:
```
# Churn balance
print("\nChurn Value Counts:")
print(pd.merge(df['Churn'].value_counts(),
               df['Churn'].value_counts(normalize=True),
               left_index=True, right_index=True, suffixes=('_count',␣
 ↪'_percentage'))
               .rename(columns={'count':'Count', 'proportion':'Percentage'}))

plt.figure(figsize=(6,4))
df['Churn'].value_counts(normalize=True).plot(kind='bar', color='darkgrey')
plt.title('Churn Distribution')
plt.xlabel('Churn')
plt.ylabel('Percentage')
plt.bar_label(
    plt.gca().containers[0],
    labels=[f"{v*100:.1f}%" for v in plt.gca().containers[0].datavalues]
```

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

```
Churn Value Counts:
       Count  Percentage
Churn
0       4682    0.831616
1        948    0.168384
```



1.0.1 EDA Snapshot 1

Dataset Overview: - 5,630 rows - 20 columns

Each row represents a unique customer. CustomerID is an identifier and will be excluded from modeling.

Churn → binary target - 1 = churned - 0 = retained

Notes: - Several numerical columns have missing values (eg. Tenure, OrderCount, DaySinceLastOrder etc.) - Categorical features will need encoding (eg. PreferredPaymentMode, Gender etc.)

1.0.2 Findings

Churn Value Counts:

Churn	Count	Percentage
0	4682	83.2 %
1	948	16.8 %

The dataset shows a churn rate of 16.8%, meaning that approximately one in six customers has churned. This level of churn is reasonable for an e-commerce context and does not raise immediate concerns about data quality.

The class distribution is moderately imbalanced, with churned customers representing only 16.8% of the dataset. This imbalance suggests that accuracy alone would be a misleading evaluation metric, and alternative metrics such as recall, precision, and ROC-AUC should be considered.

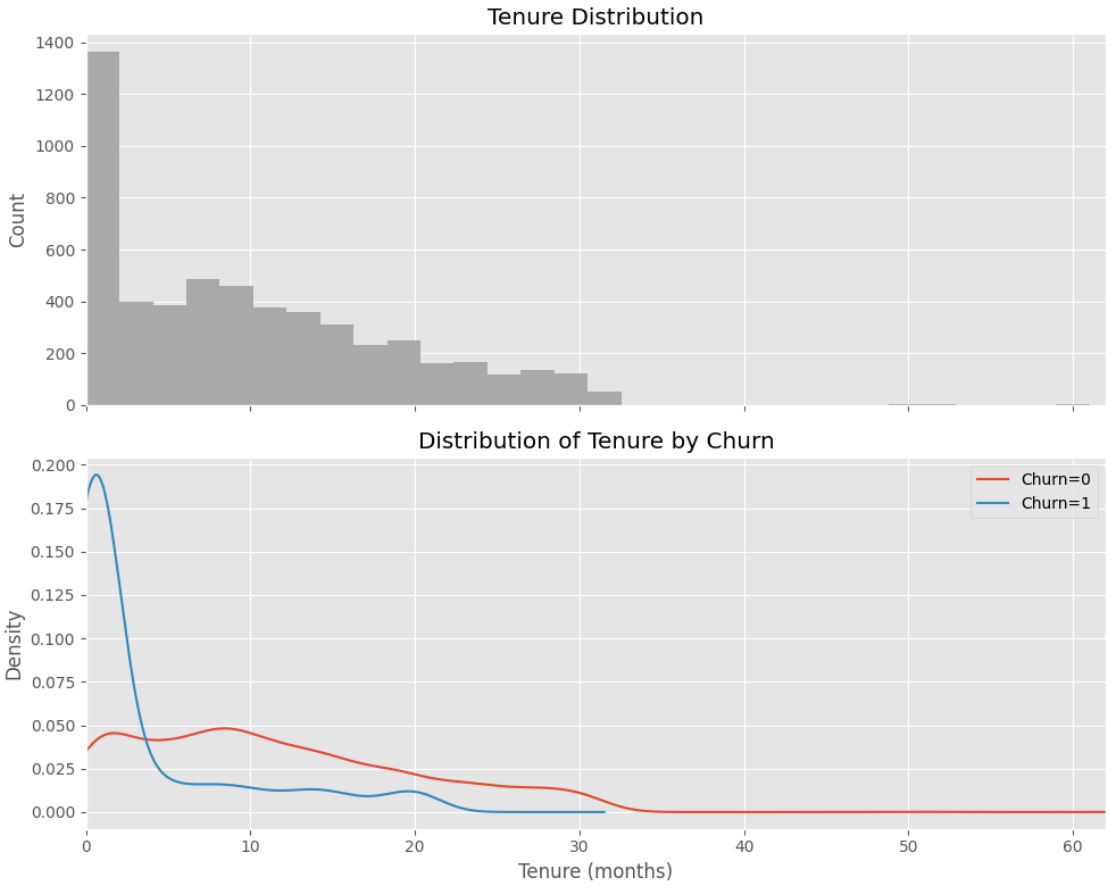
1.1 Numeric Features vs Churn

Numeric features are evaluated by comparing distributions between churned and retained customers. Features showing clear separation are considered stronger candidates for predictive modeling.

Selected numeric features to analyze for potential predictive power: - Tenure - OrderCount - DaySinceLastOrder - HourSpendOnApp - SatisfactionScore - WarehouseToHome - CouponUsed - CashbackAmount

These features describe customer behavior and engagement, which are often correlated with churn.

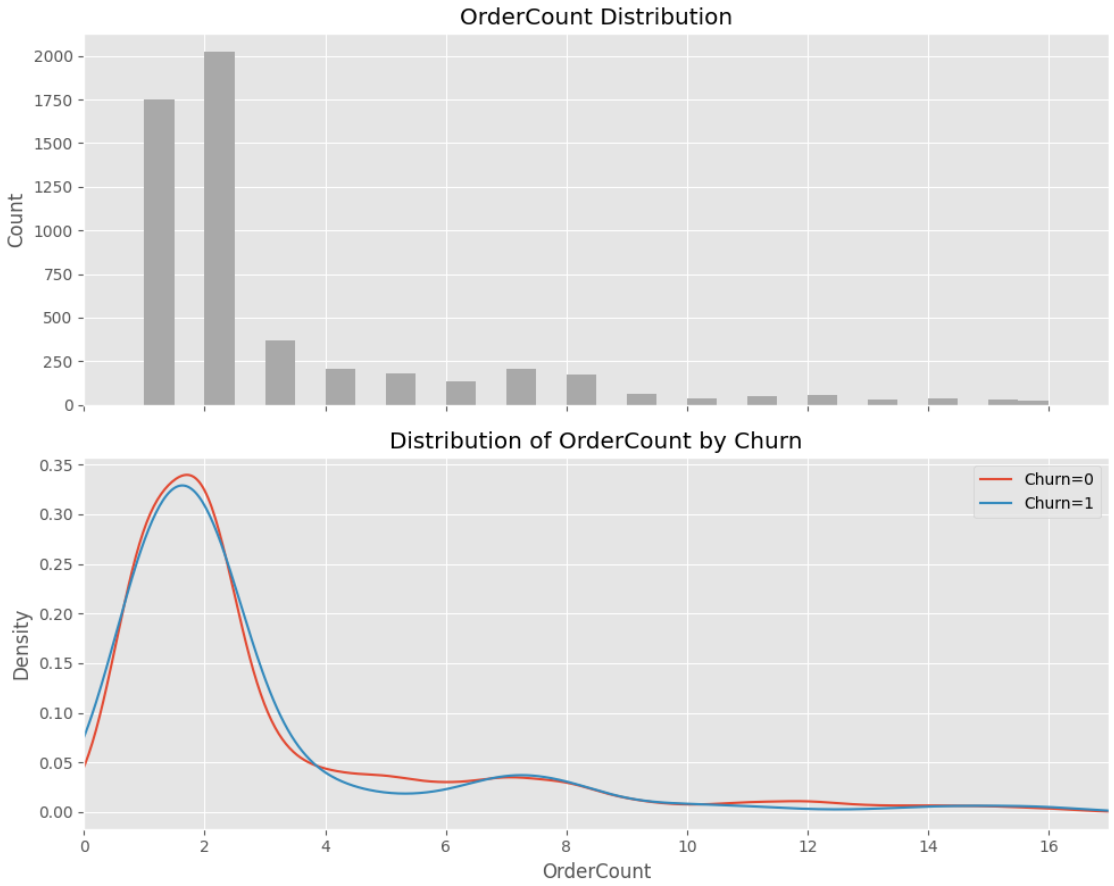
[ ]:
```
# 1. Tenure
fig, ax = plt.subplots(2, 1, figsize=(10, 8), sharex=True)

ax[0].hist(df['Tenure'].dropna(), bins=30, color='darkgrey')
ax[0].set_title('Tenure Distribution')
ax[0].set_xlabel('Tenure (months)')
ax[0].set_ylabel('Count')

for churn_value, group in df.groupby('Churn'):
    group['Tenure'].plot(kind='kde', label=f'Churn={churn_value}', ax=ax[1],␣
 ↪color=palette[churn_value])
ax[1].set_title('Distribution of Tenure by Churn')
ax[1].set_xlabel('Tenure (months)')
ax[1].set_ylabel('Density')
ax[1].legend()
ax[1].set_xlim(0, df['Tenure'].max()+1)

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



Tenure Analysis: - The tenure distribution is right-skewed, with most customers having a tenure of less than 20 months. - Churned customers tend to have shorter tenures, with a peak around 5 months, while retained customers have a broader distribution extending to higher tenures. - This suggests that customers with shorter tenures are more likely to churn, indicating that tenure could be a significant predictor of churn.

[ ]:
```
# 2. OrderCount
fig, ax = plt.subplots(2, 1, figsize=(10, 8), sharex=True)

ax[0].hist(df['OrderCount'].dropna(), bins=30, color='darkgrey')
ax[0].set_title('OrderCount Distribution')
ax[0].set_xlabel('OrderCount')
ax[0].set_ylabel('Count')

for churn_value, group in df.groupby('Churn'):
    group['OrderCount'].plot(kind='kde', label=f'Churn={churn_value}',␣
 ↪ax=ax[1], color=palette[churn_value])
ax[1].set_title('Distribution of OrderCount by Churn')
ax[1].set_xlabel('OrderCount')
```

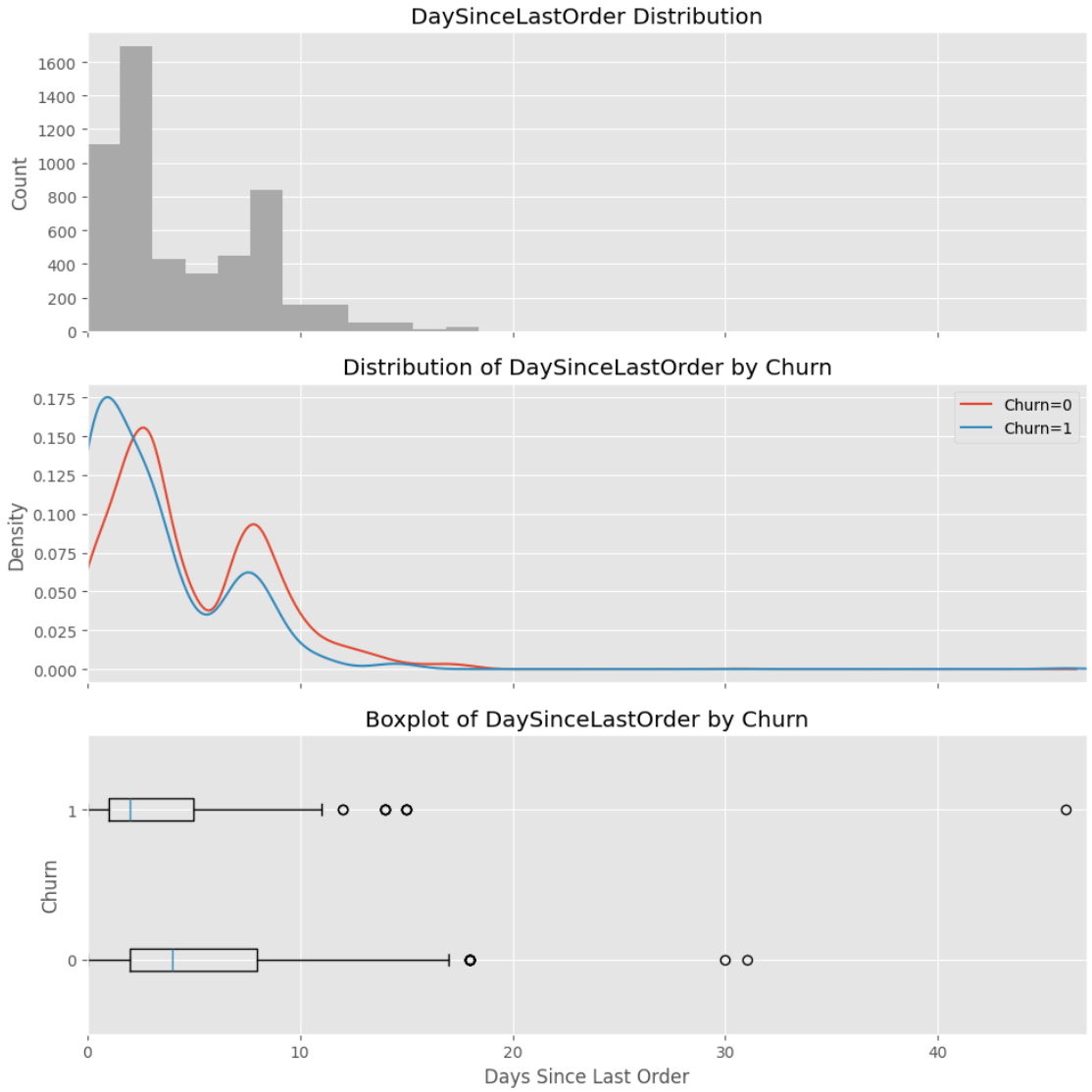
```
ax[1].set_ylabel('Density')
ax[1].legend()
ax[1].set_xlim(0, df['OrderCount'].max()+1)

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



OrderCount Analysis: - The OrderCount distribution is right-skewed, with most customers having placed fewer than 3 orders. - No significant difference is observed in the OrderCount distributions between churned and retained customers. - This suggests that OrderCount may not be a strong predictor of churn in this dataset but could still be considered in combination with other features.

[ ]:
```
# 3. DaySinceLastOrder
fig, ax = plt.subplots(3, 1, figsize=(10, 10), sharex=True)

ax[0].hist(df['DaySinceLastOrder'].dropna(), bins=30, color='darkgrey')
ax[0].set_title('DaySinceLastOrder Distribution')
ax[0].set_xlabel('Days Since Last Order')
```

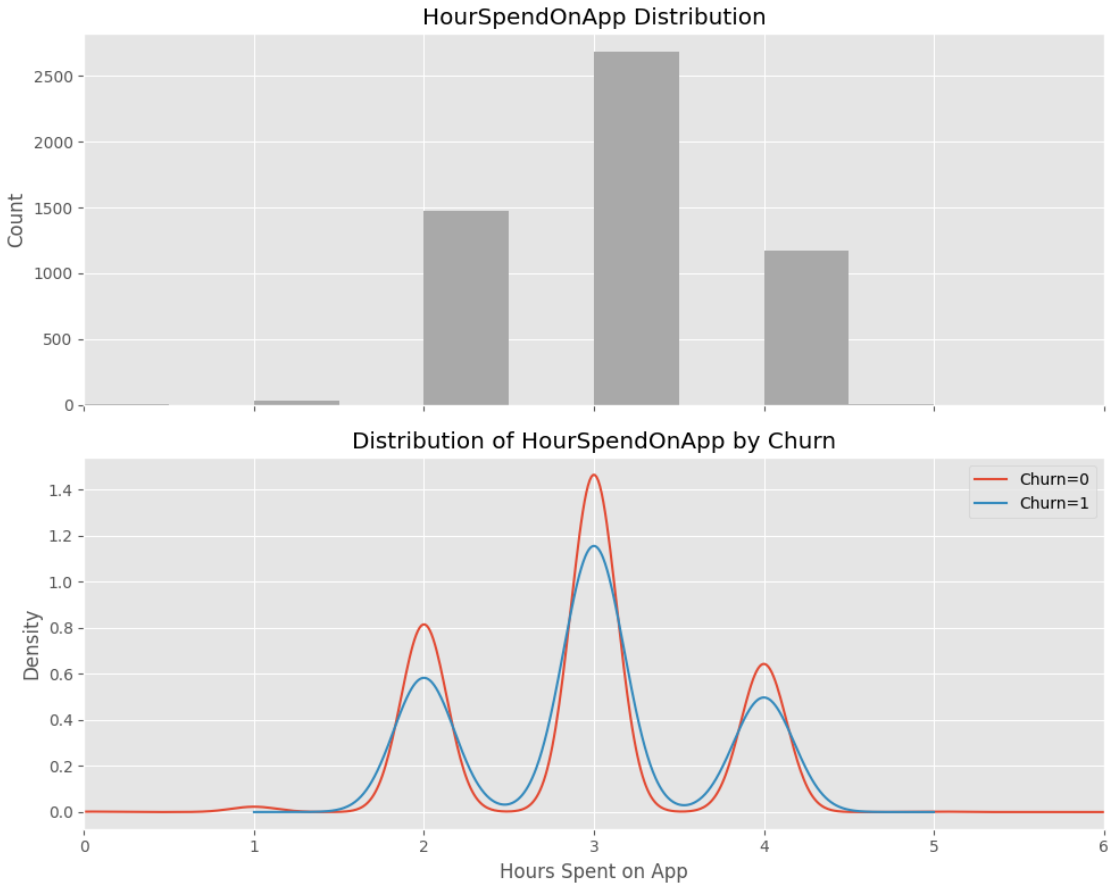
```
ax[0].set_ylabel('Count')

for churn_value, group in df.groupby('Churn'):
    group['DaySinceLastOrder'].plot(kind='kde', label=f'Churn={churn_value}',␣
 ↪ax=ax[1])
ax[1].set_title('Distribution of DaySinceLastOrder by Churn')
ax[1].set_xlabel('Days Since Last Order')
ax[1].set_ylabel('Density')
ax[1].legend()
ax[1].set_xlim(0, df['DaySinceLastOrder'].max()+1)

ax[2].boxplot([group['DaySinceLastOrder'].dropna() for _, group in df.
 ↪groupby('Churn')], labels=[churn_value for churn_value, _ in df.
 ↪groupby('Churn')], vert=False)
ax[2].set_title('Boxplot of DaySinceLastOrder by Churn')
ax[2].set_xlabel('Days Since Last Order')
ax[2].set_ylabel('Churn')

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



DaySinceLastOrder Analysis: - The DaySinceLastOrder distribution is right-skewed, with most customers having placed an order within the last 10 days. - Churned customers tend to have higher values for DaySinceLastOrder, indicating they have not placed an order in a longer time compared to retained customers. - This suggests that DaySinceLastOrder may have some predictive power for churn.

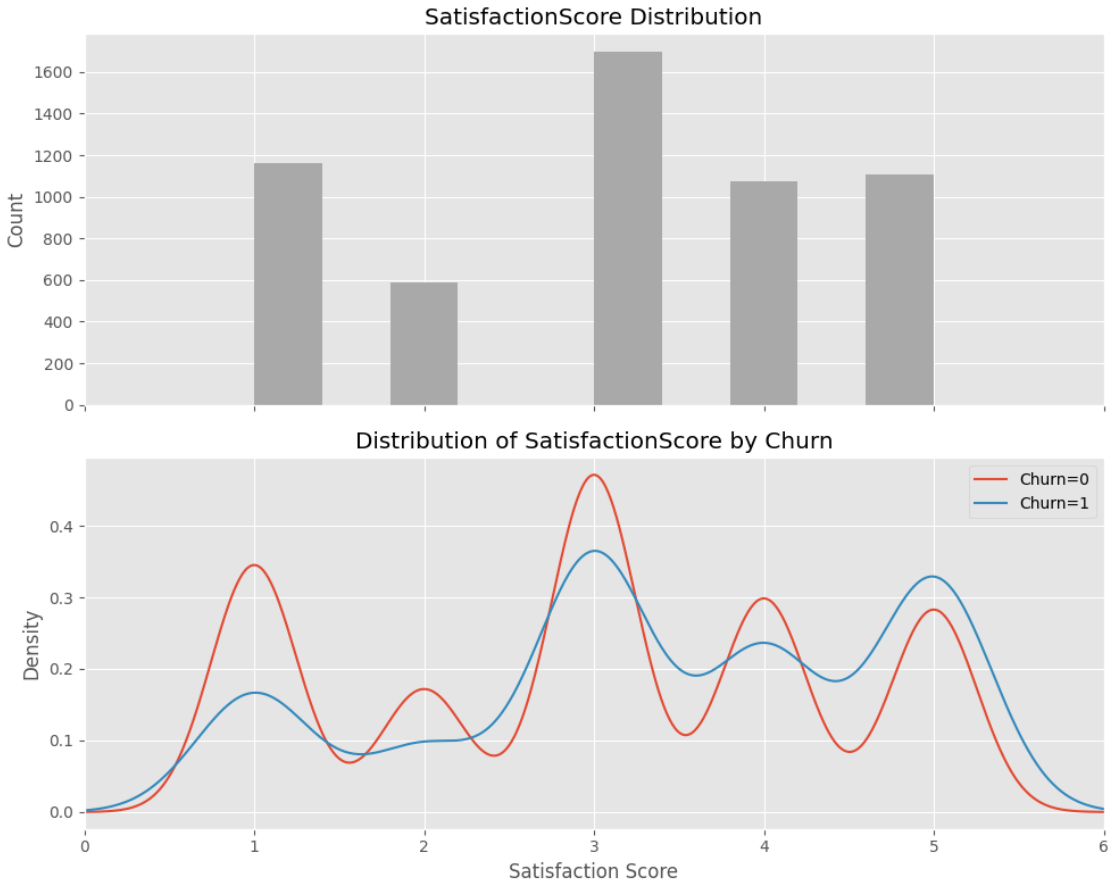
[ ]:
```
# 4. HourSpendOnApp
fig, ax = plt.subplots(2, 1, figsize=(10, 8), sharex=True)

ax[0].hist(df['HourSpendOnApp'].dropna(), bins=10, color='darkgrey')
ax[0].set_title('HourSpendOnApp Distribution')
ax[0].set_xlabel('Hours Spent on App')
ax[0].set_ylabel('Count')
```

```
for churn_value, group in df.groupby('Churn'):
    group['HourSpendOnApp'].plot(kind='kde', label=f'Churn={churn_value}',␣
 ↪ax=ax[1], color=palette[churn_value])
ax[1].set_title('Distribution of HourSpendOnApp by Churn')
ax[1].set_xlabel('Hours Spent on App')
ax[1].set_ylabel('Density')
ax[1].legend()
ax[1].set_xlim(0, df['HourSpendOnApp'].max()+1)

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



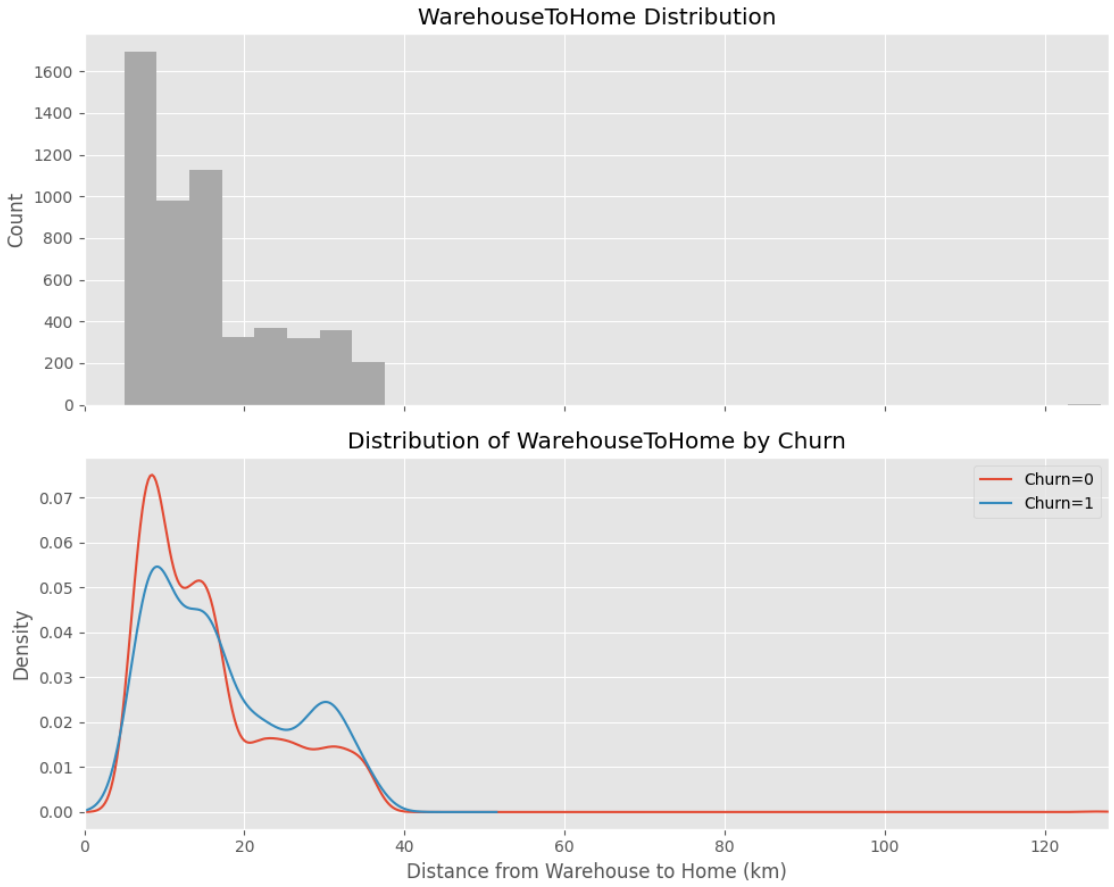
HourSpendOnApp Analysis: - The HourSpendOnApp distribution is slightly right-skewed, with most customers spending between 1 to 4 hours on the app. - No significant difference is observed in the HourSpendOnApp distributions between churned and retained customers. - This suggests that HourSpendOnApp shows limited predictive power for churn in isolation.

[ ]:
```
# 5. SatisfactionScore
fig, ax = plt.subplots(2, 1, figsize=(10, 8), sharex=True)

ax[0].hist(df['SatisfactionScore'].dropna(), bins=10, color='darkgrey')
ax[0].set_title('SatisfactionScore Distribution')
ax[0].set_xlabel('Satisfaction Score')
ax[0].set_ylabel('Count')

for churn_value, group in df.groupby('Churn'):
    group['SatisfactionScore'].plot(kind='kde', label=f'Churn={churn_value}',␣
 ↪ax=ax[1], color=palette[churn_value])
ax[1].set_title('Distribution of SatisfactionScore by Churn')
ax[1].set_xlabel('Satisfaction Score')
ax[1].set_ylabel('Density')
ax[1].legend()
ax[1].set_xlim(0, df['SatisfactionScore'].max()+1)

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



SatisfactionScore Analysis: - The SatisfactionScore distribution is approximately normal, centered around a score of 3. - Shows an interesting pattern where churned customers have higher density for higher satisfaction scores (3-5), while retained customers peak around lower scores (1-3). - This counterintuitive finding suggests that SatisfactionScore may not be a straightforward predictor of churn and warrants further investigation. - This pattern may indicate reporting bias, delayed dissatisfaction, or that satisfaction scores are collected before churn occurs.

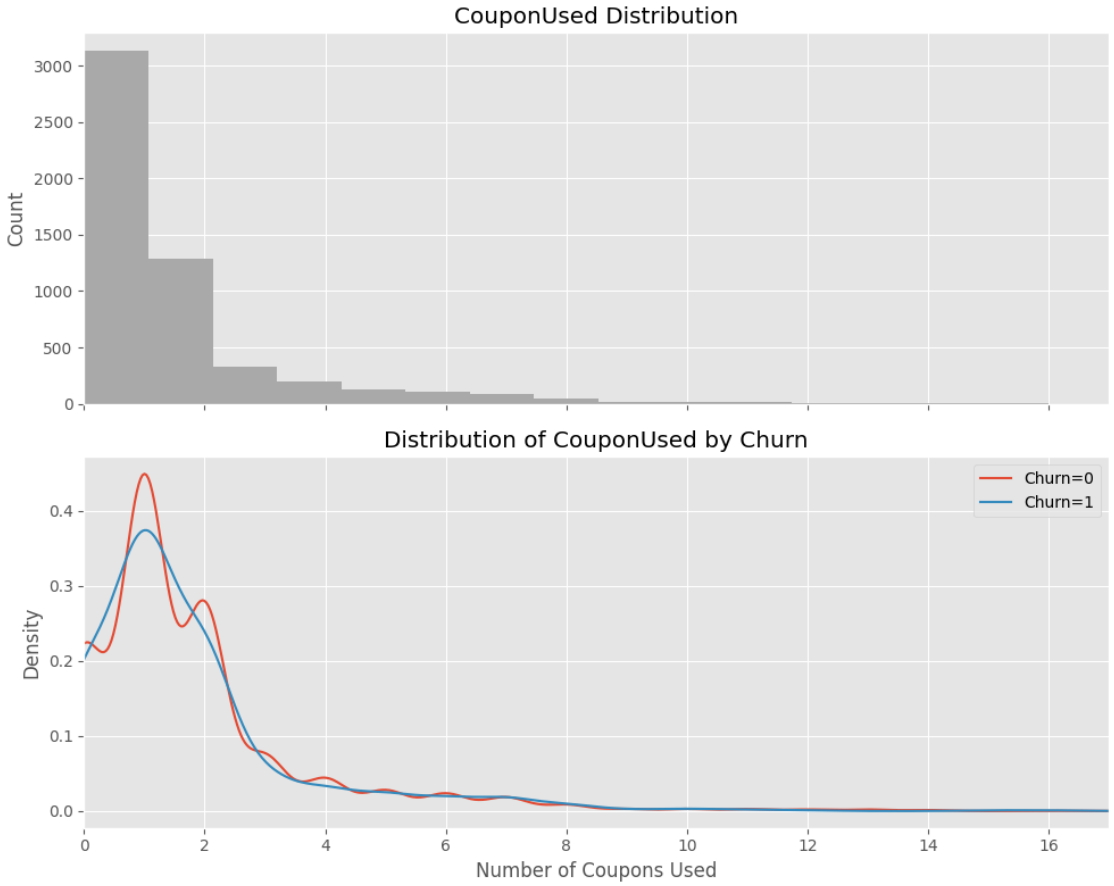
[ ]:
```
# 6. WarehouseToHome
fig, ax = plt.subplots(2, 1, figsize=(10, 8), sharex=True)

ax[0].hist(df['WarehouseToHome'].dropna(), bins=30, color='darkgrey')
ax[0].set_title('WarehouseToHome Distribution')
ax[0].set_xlabel('Distance from Warehouse to Home (km)')
ax[0].set_ylabel('Count')

for churn_value, group in df.groupby('Churn'):
    group['WarehouseToHome'].plot(kind='kde', label=f'Churn={churn_value}',␣
 ↪ax=ax[1], color=palette[churn_value])
ax[1].set_title('Distribution of WarehouseToHome by Churn')
ax[1].set_xlabel('Distance from Warehouse to Home (km)')
ax[1].set_ylabel('Density')
ax[1].legend()
ax[1].set_xlim(0, df['WarehouseToHome'].max()+1)

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



WarehouseToHome Analysis: - The WarehouseToHome distribution is right-skewed, with most customers living within 50 km of the warehouse. - No significant difference is observed in the WarehouseToHome distributions between churned and retained customers. - This suggests that WarehouseToHome shows limited predictive power for churn in isolation.

[ ]:
```
# 7. CouponUsed
fig, ax = plt.subplots(2, 1, figsize=(10, 8), sharex=True)

ax[0].hist(df['CouponUsed'].dropna(), bins=15, color='darkgrey')
ax[0].set_title('CouponUsed Distribution')
ax[0].set_xlabel('Number of Coupons Used')
ax[0].set_ylabel('Count')

for churn_value, group in df.groupby('Churn'):
    group['CouponUsed'].plot(kind='kde', label=f'Churn={churn_value}',␣
 ↪ax=ax[1], color=palette[churn_value])
ax[1].set_title('Distribution of CouponUsed by Churn')
ax[1].set_xlabel('Number of Coupons Used')
ax[1].set_ylabel('Density')
```

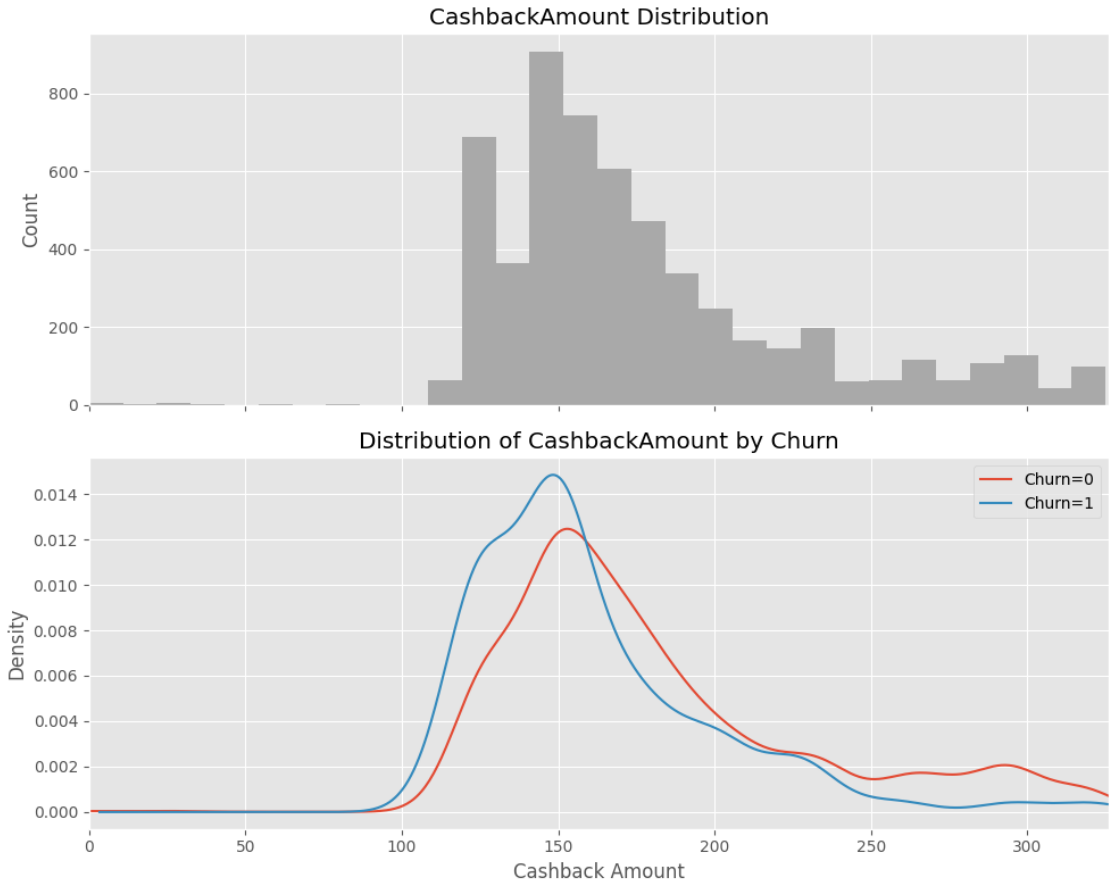
```
ax[1].legend()
ax[1].set_xlim(0, df['CouponUsed'].max()+1)

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



CouponUsed Analysis: - The CouponUsed distribution is right-skewed, with most customers using fewer than 5 coupons. - No significant difference is observed in the CouponUsed distributions between churned and retained customers. - This suggests that CouponUsed shows limited predictive power for churn in isolation. - However, coupon usage may reflect retention interventions rather than organic behavior.

[ ]:
```
# 8. CashbackAmount
fig, ax = plt.subplots(2, 1, figsize=(10, 8), sharex=True)

ax[0].hist(df['CashbackAmount'].dropna(), bins=30, color='darkgrey')
ax[0].set_title('CashbackAmount Distribution')
ax[0].set_xlabel('Cashback Amount')
ax[0].set_ylabel('Count')
```

```
for churn_value, group in df.groupby('Churn'):
    group['CashbackAmount'].plot(kind='kde', label=f'Churn={churn_value}',␣
 ↪ax=ax[1], color=palette[churn_value])
ax[1].set_title('Distribution of CashbackAmount by Churn')
ax[1].set_xlabel('Cashback Amount')
ax[1].set_ylabel('Density')
ax[1].legend()
ax[1].set_xlim(0, df['CashbackAmount'].max()+1)

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



CashbackAmount Analysis: - The CashbackAmount distribution is right-skewed with a long tail toward higher cashback values, with most customers receiving a cashback amount between 100 and 200. - No significant difference is observed in the CashbackAmount distributions between churned and retained customers. - This suggests that CashbackAmount shows limited predictive power for churn in isolation.

1.1.1 Summary of Numeric Feature Analysis:

1. Tenure shows clear separation between churned and retained customers, suggesting strong predictive potential.
2. DaySinceLastOrder shows a strong association with churn but may introduce data leakage, as the feature may be directly influenced by the churn definition itself. This feature will be handled carefully during model development.
3. OrderCount, HourSpendOnApp, WarehouseToHome, CouponUsed, and CashbackAmount show limited standalone separation but may still contribute predictive value in combination with other features.
4. SatisfactionScore displays a counterintuitive pattern that warrants further investigation.

1.2 Categorical Features vs Churn

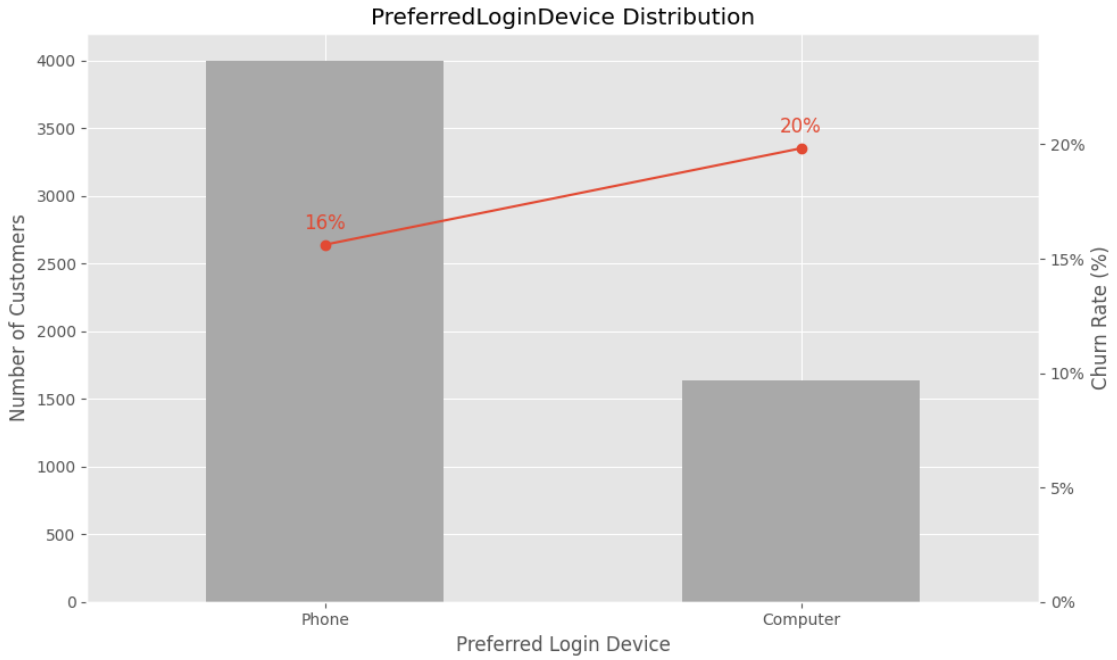
The following analysis examines the relationship between selected categorical features and customer churn. For each feature, we compare category distributions and churn rates to assess potential predictive value. These insights will guide encoding strategies and feature selection in the modeling phase.

Selected categorical features to analyze for potential predictive power: - PreferedLoginDevice - PreferredPaymentMode - Gender - PreferedOrderCat - MaritalStatus - CityTier (ordinal but categorical in behavior)

These features describe customer preferences and demographics, which may influence engagement and churn behavior.

[ ]:
```
# 1. PreferredLoginDevice
comb_df = df.copy()
comb_df['PreferredLoginDevice'] = comb_df['PreferredLoginDevice'].
 ↪replace({'Mobile Phone':'Phone'})

fig, ax = plt.subplots(figsize=(10, 6))
ax2 = ax.twinx()

counts = comb_df['PreferredLoginDevice'].value_counts()
counts.plot(kind='bar', color='darkgrey', ax=ax)
ax.set_xlabel('Preferred Login Device')
ax.xaxis.set_ticklabels(counts.index, rotation=0)
ax.set_ylabel('Number of Customers')

churn_rates = comb_df.groupby('PreferredLoginDevice')['Churn'].mean().
 ↪reindex(counts.index)
churn_rates.plot(kind='line', marker='o', ax=ax2, grid=False, color=palette[0])
ax2.set_ylabel('Churn Rate (%)')
ax.set_title('PreferredLoginDevice Distribution')

ax2.yaxis.set_major_formatter(lambda x, pos: f'{x*100:.0f}%')
ax2.set_ylim(0, churn_rates.max() + 0.05)
```

```
for i, value in enumerate(churn_rates):
    ax2.text(i, value + 0.005, f'{value*100:.0f}%', ha='center', va='bottom',␣
 ↪color=palette[0], fontsize=12)

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

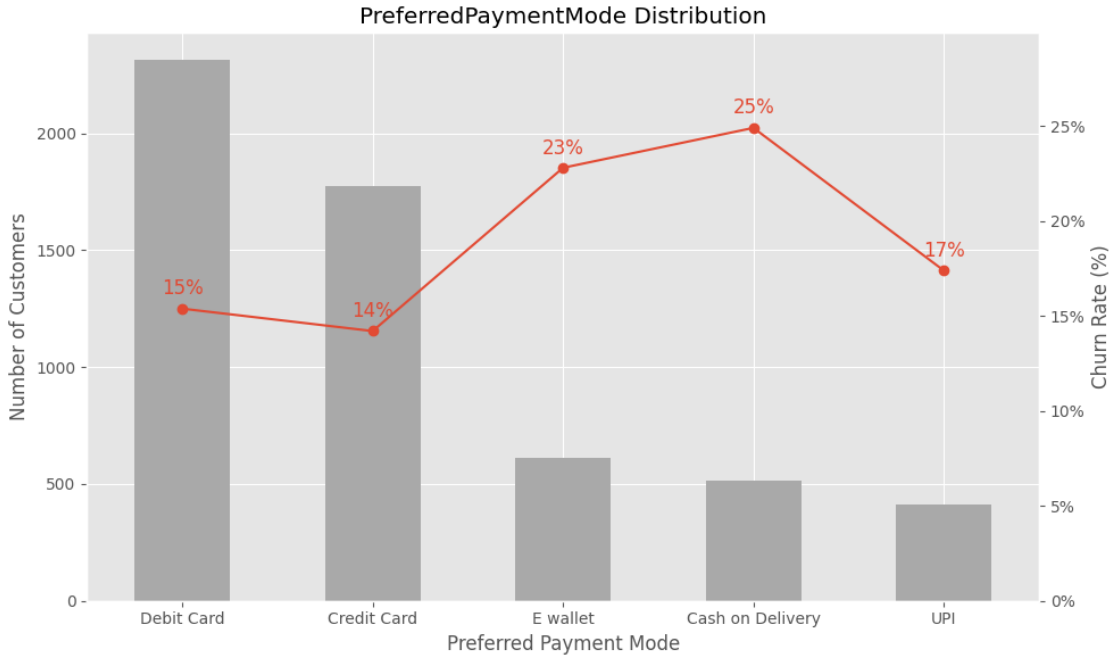


PreferredLoginDevice Analysis:

Assumption: Mobile Phone and Phone categories are combined under Phone. Need to verify if they in fact do not indicate different platforms. - The PreferredLoginDevice distribution shows that the majority of customers prefer using a Phone. - Churn rates vary between devices. - This suggests that PreferredLoginDevice may contribute predictive signal, potentially capturing differences in engagement patterns across devices rather than acting as a standalone churn driver.

[ ]:
```
# 2. PreferredPaymentMode
comb_df['PreferredPaymentMode'] = comb_df['PreferredPaymentMode'].replace({'CC':
 ↪'Credit Card', 'COD':'Cash on Delivery'})

fig, ax = plt.subplots(figsize=(10, 6))
ax2 = ax.twinx()

counts = comb_df['PreferredPaymentMode'].value_counts()
counts.plot(kind='bar', color='darkgrey', ax=ax)
ax.set_xlabel('Preferred Payment Mode')
ax.xaxis.set_ticklabels(counts.index, rotation=0)
```

```
ax.set_ylabel('Number of Customers')

churn_rates = comb_df.groupby('PreferredPaymentMode')['Churn'].mean().
 ↪reindex(counts.index)
churn_rates.plot(kind='line', marker='o', ax=ax2, grid=False,
                 color=palette[0])
ax2.set_ylabel('Churn Rate (%)')
ax.set_title('PreferredPaymentMode Distribution')
ax2.yaxis.set_major_formatter(lambda x, pos: f'{x*100:.0f}%')
ax2.set_ylim(0, churn_rates.max() + 0.05)

for i, value in enumerate(churn_rates):
    ax2.text(i, value + 0.005, f'{value*100:.0f}%', ha='center', va='bottom',␣
 ↪color=palette[0], fontsize=12)

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

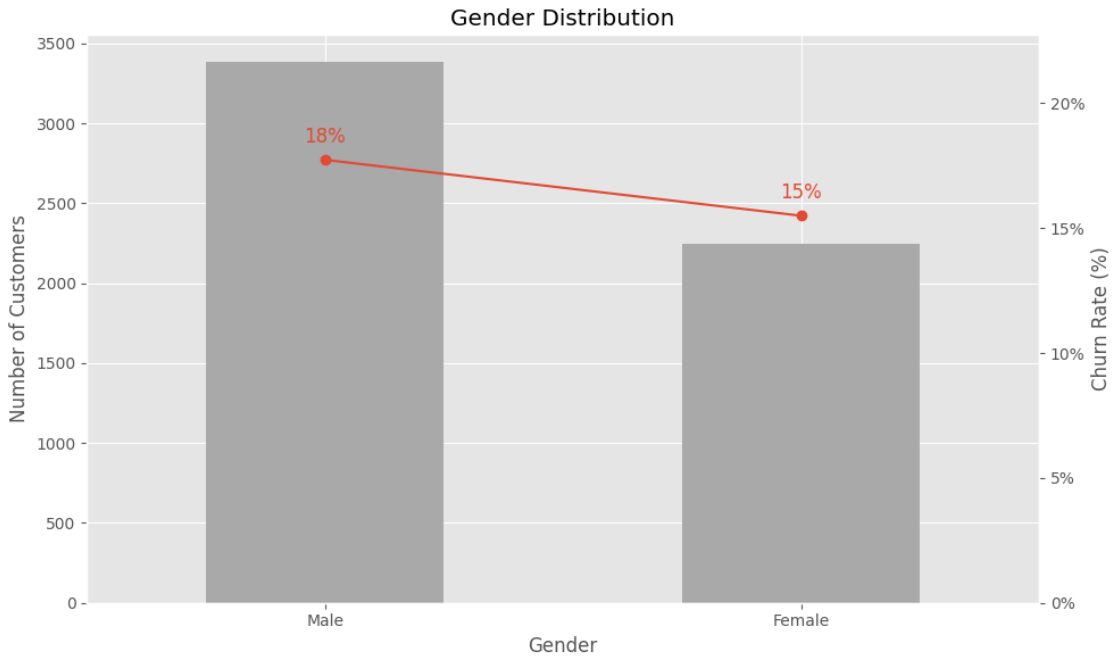


PreferredPaymentMode Analysis:

Assumption: CC and Credit Card categories are combined under Credit Card, COD and Cash on Delivery categories are combined under Cash on Delivery. Need to verify if they in fact do not indicate different payment methods. - The PreferredPaymentMode distribution shows that the majority of customers prefer using Debit and Credit Card. - Churn rates vary between payment modes, with a significantly higher churn rate for Cash on Delivery. - Given the clear separation across categories, PreferredPaymentMode is a strong candidate for one-hot encoding and inclusion

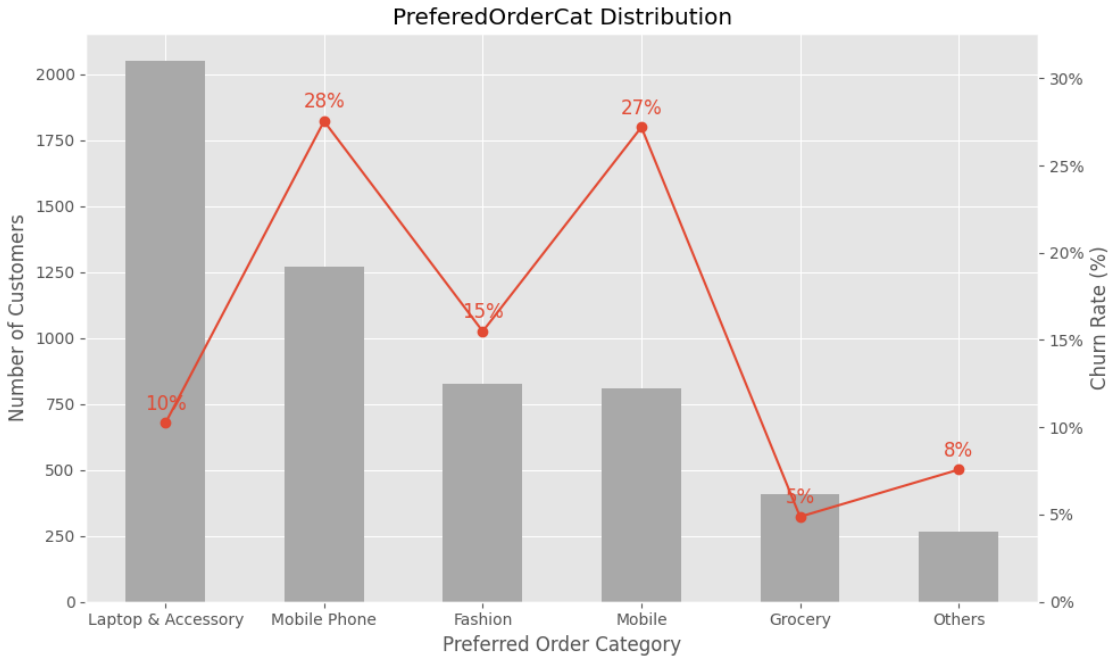
in baseline models.

[ ]:
```
# 3. Gender
fig, ax = plt.subplots(figsize=(10, 6))
ax2 = ax.twinx()

counts = df['Gender'].value_counts()
counts.plot(kind='bar', color='darkgrey', ax=ax)
ax.set_xlabel('Gender')
ax.xaxis.set_ticklabels(counts.index, rotation=0)
ax.set_ylabel('Number of Customers')

churn_rates = df.groupby('Gender')['Churn'].mean().reindex(counts.index)
churn_rates.plot(kind='line', marker='o', ax=ax2, grid=False, color=palette[0])
ax2.set_ylabel('Churn Rate (%)')
ax.set_title('Gender Distribution')
ax2.yaxis.set_major_formatter(lambda x, pos: f'{x*100:.0f}%')
ax2.set_ylim(0, churn_rates.max() + 0.05)

for i, value in enumerate(churn_rates):
    ax2.text(i, value + 0.005, f'{value*100:.0f}%', ha='center', va='bottom',␣
 ↪color=palette[0], fontsize=12)

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



Gender Analysis: - The Gender distribution shows a higher number of male customers compared to female customers. - Churn rates are slightly higher for male customers. - Gender shows a minor difference in churn rates and is unlikely to be a strong predictor on its own, but may provide marginal value when combined with other behavioral features.

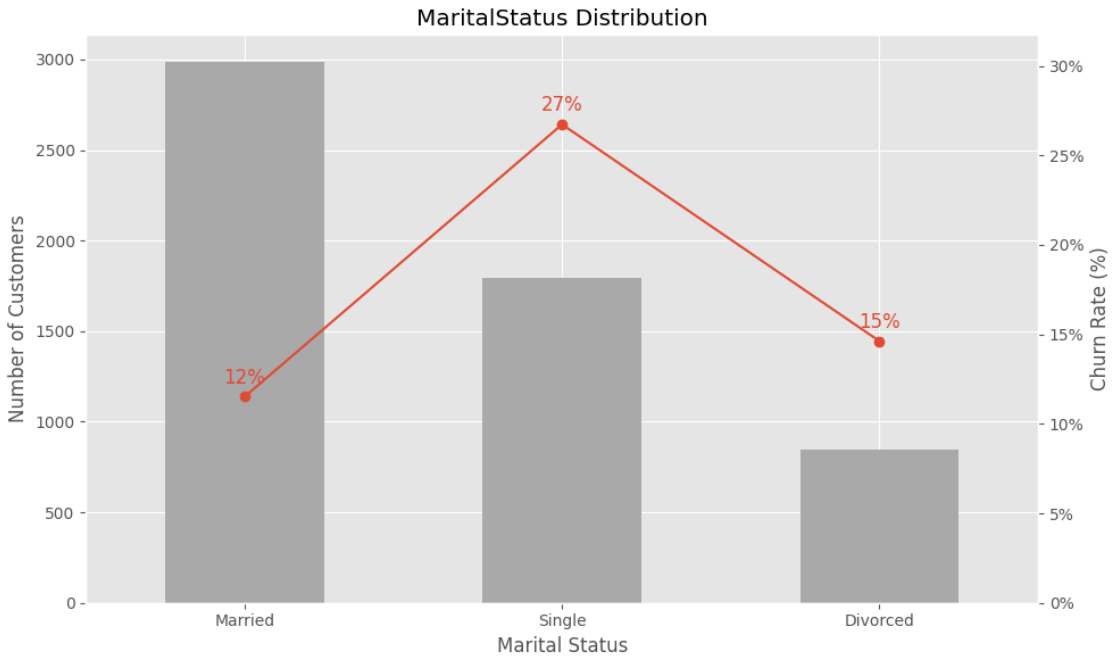
[ ]:
```
# 4. PreferedOrderCat
fig, ax = plt.subplots(figsize=(10, 6))
ax2 = ax.twinx()

counts = df['PreferedOrderCat'].value_counts()
counts.plot(kind='bar', color='darkgrey', ax=ax)
ax.set_xlabel('Preferred Order Category')
ax.xaxis.set_ticklabels(counts.index, rotation=0)
ax.set_ylabel('Number of Customers')

churn_rates = df.groupby('PreferedOrderCat')['Churn'].mean().reindex(counts.
 ↪index)
churn_rates.plot(kind='line', marker='o', ax=ax2, grid=False, color=palette[0])
ax2.set_ylabel('Churn Rate (%)')
ax.set_title('PreferedOrderCat Distribution')
ax2.yaxis.set_major_formatter(lambda x, pos: f'{x*100:.0f}%')
ax2.set_ylim(0, churn_rates.max() + 0.05)

for i, value in enumerate(churn_rates):
    ax2.text(i, value + 0.005, f'{value*100:.0f}%', ha='center', va='bottom',␣
 ↪color=palette[0], fontsize=12)

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



PreferedOrderCat Analysis: - The PreferedOrderCat distribution shows that Electronics is the most preferred category among customers. - Churn rates vary across categories, with Groceries showing a notably lower churn rate compared to others. - This suggests that PreferedOrderCat may have predictive power for churn.

[ ]:
```
# 5. MaritalStatus
fig, ax = plt.subplots(figsize=(10, 6))
ax2 = ax.twinx()

counts = df['MaritalStatus'].value_counts()
counts.plot(kind='bar', color='darkgrey', ax=ax)
ax.set_xlabel('Marital Status')
ax.xaxis.set_ticklabels(counts.index, rotation=0)
ax.set_ylabel('Number of Customers')

churn_rates = df.groupby('MaritalStatus')['Churn'].mean().reindex(counts.index)
churn_rates.plot(kind='line', marker='o', ax=ax2, grid=False, color=palette[0])
ax2.set_ylabel('Churn Rate (%)')
ax.set_title('MaritalStatus Distribution')
ax2.yaxis.set_major_formatter(lambda x, pos: f'{x*100:.0f}%')
ax2.set_ylim(0, churn_rates.max() + 0.05)

for i, value in enumerate(churn_rates):
    ax2.text(i, value + 0.005, f'{value*100:.0f}%', ha='center', va='bottom',␣
 ↪color=palette[0], fontsize=12)
```

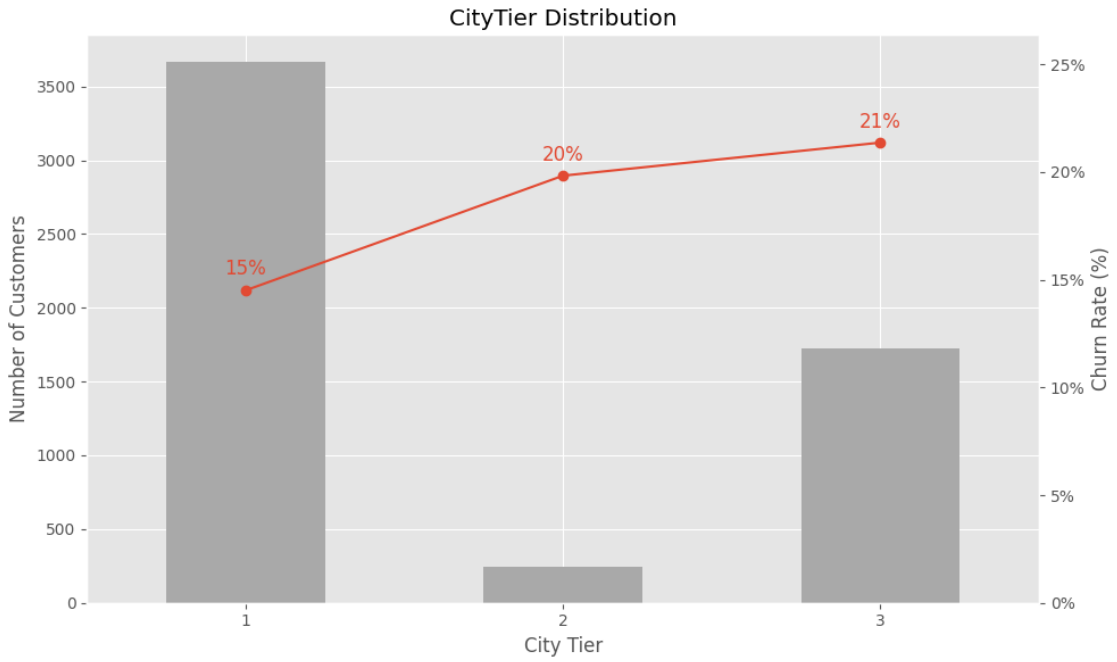
```
plt.tight_layout()
plt.show()
```



MaritalStatus Analysis: - The MaritalStatus distribution shows that the majority of customers are Married. - MaritalStatus shows moderate variation in churn rates across categories. - While the feature is not directly actionable, it may act as a proxy for household structure and purchasing behavior.

[ ]:
```
# 6. CityTier
fig, ax = plt.subplots(figsize=(10, 6))
ax2 = ax.twinx()

counts = df['CityTier'].value_counts().sort_index()
counts.plot(kind='bar', color='darkgrey', ax=ax)
ax.set_xlabel('City Tier')
ax.xaxis.set_ticklabels(counts.index, rotation=0)
ax.set_ylabel('Number of Customers')

churn_rates = df.groupby('CityTier')['Churn'].mean().reindex(counts.index)
x = range(len(counts))  # 0,1,2 aligned with bar centers
ax2.plot(x, churn_rates.values, marker='o', color=palette[0])
ax2.grid(False)
ax2.set_ylabel('Churn Rate (%)')
ax.set_title('CityTier Distribution')
ax2.yaxis.set_major_formatter(lambda x, pos: f'{x*100:.0f}%')
ax2.set_ylim(0, churn_rates.max() + 0.05)
```

```
for i, value in enumerate(churn_rates):
    ax2.text(i, value + 0.005, f'{value*100:.0f}%', ha='center', va='bottom',␣
 ↪color=palette[0], fontsize=12)

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



CityTier Analysis: - The CityTier distribution shows that most customers are from Tier 1 cities. - Churn rates increase with higher city tiers, with Tier 3 cities showing the highest churn rate. - This suggests that CityTier may have predictive power for churn.

1.2.1 Summary of Categorical Feature Analysis:

Overall, several categorical features demonstrate meaningful variation in churn rates. **PreferredPaymentMode**, **PreferedOrderCat**, and **MaritalStatus** show the strongest potential signal, while **PreferredLoginDevice**, **Gender**, and **CityTier** may provide complementary information.

These features will be retained for modeling, with appropriate encoding strategies applied and their contribution evaluated using model-based feature importance.

1.3 Key EDA Takeaways

- Churn is moderately imbalanced (16.8%), requiring metric-aware evaluation.
- Tenure and recency-related features show the strongest separation.
- Several categorical features exhibit meaningful churn variation.
- Some features may pose leakage risk and will be treated cautiously.

[ ]:
```
comb_df.to_csv(folder_path + '/data/ecommerce_churn_data_eda.csv')
```