Churn Prediction ETL

[1]:

**import**

**numpy**

**as**

**np**

**import**

**pandas**

**as**

**pd**

**import**

**seaborn**

**as**

**sns**

**import**

**matplotlib**

**.**

**ticker**

**as**

**mtick**

**import**

**matplotlib**

**.**

**pyplot**

**as**

**plt**

%

**matplotlib**

inline

[2]:

telco\_base\_data

=

pd

.

read\_csv(

'

Telco-Customer-Churn.csv

'

)

[3]:

telco\_base\_data

.

head()

[3]: 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  PaymentMethod MonthlyCharges TotalCharges Churn | | Yes |
| 0 | Electronic check 29.85 29.85 No | |  |
| 1 | Mailed check 56.95 1889.5 No | |  |
| 2 | Mailed check 53.85 108.15 Yes | |  |
| 1. Bank transfer (automatic) 42.30 1840.75 No 2. Electronic check 70.70 151.65 Yes | | |

[5 rows x 21 columns]

[4]:

telco\_base\_data

.

shape

[4]: (7043, 21)

[5]:

telco\_base\_data

.

columns

[5]: Index(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',

'tenure', 'PhoneService', 'MultipleLines', 'InternetService',

'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport',

'StreamingTV', 'StreamingMovies', 'Contract', 'PaperlessBilling',

'PaymentMethod', 'MonthlyCharges', 'TotalCharges', 'Churn'], dtype='object')

[6]:

telco\_base\_data

.

dtypes

|  |  |
| --- | --- |
| [6]: customerID | object |
| gender | object |
| SeniorCitizen | int64 |
| Partner | object |
| Dependents | object |
| tenure | int64 |
| PhoneService | object |
| MultipleLines | object |
| InternetService | object |
| OnlineSecurity | object |
| OnlineBackup | object |
| DeviceProtection | object |
| TechSupport | object |
| StreamingTV | object |
| StreamingMovies | object |
| Contract | object |
| PaperlessBilling | object |
| PaymentMethod | object |
| MonthlyCharges | float64 |
| TotalCharges | object |
| Churn dtype: object | object |

[7]:

telco\_base\_data

.

describe()

[7]: SeniorCitizen tenure MonthlyCharges

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| count | 7043.000000 7043.000000 | | | 7043.000000 | |
| mean | 0.162147 32.371149 | | | 64.761692 | |
| std | 0.368612 24.559481 | | | 30.090047 | |
| min | 0.000000 0.000000 | | | 18.250000 | |
| 25% | | 0.000000 | 9.000000 | | 35.500000 |
| 50% | | 0.000000 | 29.000000 | | 70.350000 |
| 75% | | 0.000000 | 55.000000 | | 89.850000 |
| max | | 1.000000 | 72.000000 | | 118.750000 |

Churn visulation

[8]:

telco\_base\_data[

'

Churn

'

]

.

value\_counts()

.

plot(kind

=

'

barh

'

, figsize

=

(

8

,

6

))

plt

.

xlabel(

"

count

"

, labelpad

=

14

)

plt

.

ylabel(

"

target variable

"

, labelpad

=

14

)

plt

.

title(

"

count of target variable per category

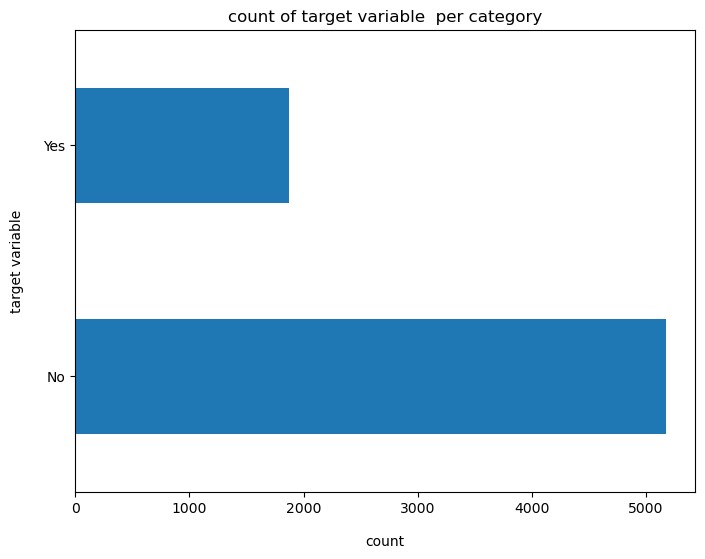
"

)

plt

.

show()



Churn percentage

[9]: 100\*telco\_base\_data['Churn'].value\_counts()/len(telco\_base\_data['Churn'])

[9]: Churn No 73.463013

Yes 26.536987

Name: count, dtype: float64

The data is highly imbalanced because it have more no will results as biased decision. Balanced data means 50:50 or 45:55 ratio.The imbalanced data ratio is 73:27

[10]:

telco\_base\_data[

'

Churn

'

]

.

value\_counts()

[10]: Churn

No 5174

Yes 1869

Name: count, dtype: int64

[11]:

telco\_base\_data

.

info(verbose

=

**True**

)

<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

[12]: *## calculating missing values this template can be used for all data set* missing = pd.DataFrame({

'column\_names': telco\_base\_data.columns,

'MissingPercentage': (telco\_base\_data.isnull().sum() \* 100) /␣ ↪telco\_base\_data.shape[0]

})

plt

.

figure(figsize

=

(

16

,

5

))

ax

=

sns

.

pointplot(x

=

"

column\_names

"

,y

=

"

MissingPercentage

"

, data

=

missing)

plt

.

xticks(rotation

=

90

, fontsize

=

7

)

plt

.

title(

"

Percentage of Missing Values

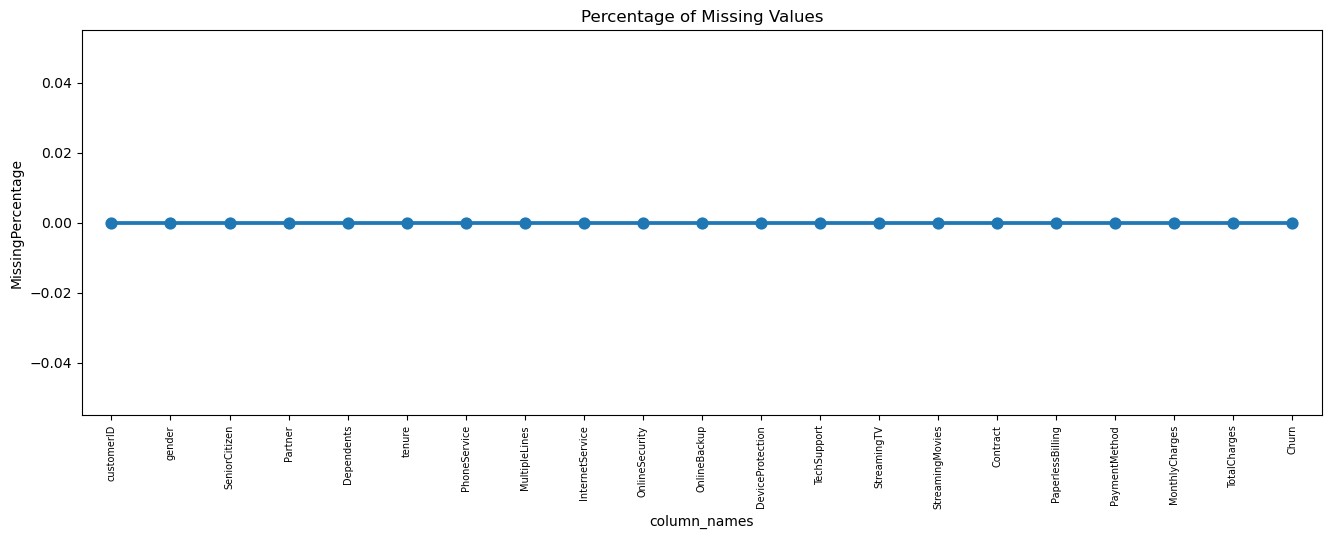
"

)

plt

.

show()



# 0.1 Data Cleaning

1.Create a copy of data for manupulation & processing

[13]:

telco\_data

=

telco\_base\_data

.

copy()

2.Total charges dhould be numerical values. Let’s convert it into numeric data type

[14]:

telco\_data

.

TotalCharges

=

pd

.

to\_numeric(telco\_data

.

TotalCharges,

␣

↪

errors

=

'

coerce

'

)

telco\_data

.

isnull()

.

sum()

|  |  |  |  |
| --- | --- | --- | --- |
| [14]: 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 | | 11 |
| Churn dtype: int64 | | 0 |

3.As we can see after converting the object into numerical we got 11 missing values in total charges

[15]: telco\_data.loc[telco\_data['TotalCharges'].isnull()]

[15]: customerID gender SeniorCitizen Partner Dependents tenure \

488 4472-LVYGI Female 0 Yes Yes 0

753 3115-CZMZD Male 0 No Yes 0

936 5709-LVOEQ Female 0 Yes Yes 0

1082 4367-NUYAO Male 0 Yes Yes 0

1340 1371-DWPAZ Female 0 Yes Yes 0

3331 7644-OMVMY Male 0 Yes Yes 0

3826 3213-VVOLG Male 0 Yes Yes 0

4380 2520-SGTTA Female 0 Yes Yes 0

5218 2923-ARZLG Male 0 Yes Yes 0

6670 4075-WKNIU Female 0 Yes Yes 0

6754 2775-SEFEE Male 0 No Yes 0

PhoneService MultipleLines InternetService OnlineSecurity … \

488 No No phone service DSL Yes … 753 Yes No No No internet service …

936 Yes No DSL Yes …

1082 Yes Yes No No internet service … 1340 No No phone service DSL Yes … 3331 Yes No No No internet service …

3826 Yes Yes No No internet service …

4380 Yes No No No internet service … 5218 Yes No No No internet service …

6670 Yes Yes DSL No …

6754 Yes Yes DSL Yes …

DeviceProtection TechSupport StreamingTV \

488 Yes Yes Yes

753 No internet service No internet service No internet service

936 Yes No Yes

1082 No internet service No internet service No internet service

1340 Yes Yes Yes

3331 No internet service No internet service No internet service

3826 No internet service No internet service No internet service

4380 No internet service No internet service No internet service

5218 No internet service No internet service No internet service

6670 Yes Yes Yes

6754 No Yes No

StreamingMovies Contract PaperlessBilling \

|  |  |
| --- | --- |
| 488 No Two year | Yes |
| 753 No internet service Two year | No |
| 936 Yes Two year | No |
| 1082 No internet service Two year | No |
| 1340 No Two year | No |
| 3331 No internet service Two year | No |
| 3826 No internet service Two year | No |
| 4380 No internet service Two year | No |
| 5218 No internet service One year | Yes |
| 6670 No Two year | No |
| 6754 No Two year | Yes |

PaymentMethod MonthlyCharges TotalCharges Churn

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 488 | Bank transfer (automatic) | 52.55 | NaN | No |
| 753 | Mailed check | 20.25 | NaN | No |
| 936 | Mailed check | 80.85 | NaN | No |
| 1082 | Mailed check | 25.75 | NaN | No |
| 1340 | Credit card (automatic) | 56.05 | NaN | No |
| 3331 | Mailed check | 19.85 | NaN | No |
| 3826 | Mailed check | 25.35 | NaN | No |
| 4380 | Mailed check | 20.00 | NaN | No |
| 5218 | Mailed check | 19.70 | NaN | No |
| 6670 | Mailed check | 73.35 | NaN | No |
| 6754 Bank transfer (automatic) | | 61.90 | NaN | No |

[11 rows x 21 columns]

4.Since the % of missing records compared to the dataset is very 0.15%. It safe to ignore them from further processing

[16]:

*# Droping missing values rows*

telco\_data

.

dropna(how

=

'

any

'

,inplace

=

**True**

)

5.Dividing customers into bins based on tenure. For example tenure < 12 months Assign tenure group of 1-12,for tenure 1-2years assign 1-12,13-24 and so on.

[17]:

*# Get the max value of tenure*

telco\_data[

'

tenure

'

]

.

max()

[17]: 72

[18]: *# Group tenure into bins of 12 months* labels = [f"**{**i**}** - **{**i + 11**}**" **for** i **in** range (1, 72, 12)]

telco\_data['tenure\_group'] = pd.cut(telco\_data.tenure, range(1, 80, 12),right =␣ ↪**False**, labels = labels)

[19]: telco\_data['tenure\_group'].value\_counts()

[19]: tenure\_group

1 - 12 2175

61 - 72 1407

13 - 24 1024

25 - 36 832

49 - 60 832

37 - 48 762

Name: count, dtype: int64

6.Remove columns not required for processing

[20]: *# drop column customerID and tenure* telco\_data.drop(columns= ['customerID','tenure'], axis=1, inplace=**True**) telco\_data.head()

[20]: gender SeniorCitizen Partner Dependents PhoneService MultipleLines \

1. Female 0 Yes No No No phone service
2. Male 0 No No Yes No
3. Male 0 No No Yes No
4. Male 0 No No No No phone service
5. Female 0 No No Yes No

InternetService OnlineSecurity OnlineBackup DeviceProtection TechSupport \

0 DSL No Yes No No 1 DSL Yes No Yes No

1. DSL Yes Yes No No
2. DSL Yes No Yes Yes
3. Fiber optic No No No No

StreamingTV StreamingMovies Contract PaperlessBilling \

1. No No Month-to-month Yes
2. No No One year No
3. No No Month-to-month Yes
4. No No One year No
5. No No Month-to-month Yes

PaymentMethod MonthlyCharges TotalCharges Churn tenure\_group

0 Electronic check 29.85 29.85 No 1 - 12 1 Mailed check 56.95 1889.50 No 25 - 36

1. Mailed check 53.85 108.15 Yes 1 - 12
2. Bank transfer (automatic) 42.30 1840.75 No 37 - 48
3. Electronic check 70.70 151.65 Yes 1 - 12

# 0.2 Data Exploration 0.3 1. Univariate Analysis

[21]:

categorical\_cols

=

telco\_data

.

↪

drop(columns

=

[

'

Churn

'

,

'

TotalCharges

'

,

'

MonthlyCharges

'

])

.

↪

select\_dtypes(include

=

[

'

object

'

,

'

category

'

])

.

columns

*# Plot count plots for each categorical predictor*

**for**

i, predictor

**in**

enumerate

):

(

categorical\_cols

plt

.

figure(figsize

=

(

6

,

4

))

*# Set figure size*

sns

.

countplot(data

=

telco\_data, x

=

predictor, hue

=

'

Churn

'

, palette

=

'

viridis

'

)

plt

.

title(

f

'

Churn Distribution for

**{**

predictor

**}**

'

)

plt

.

xticks(rotation

=

45

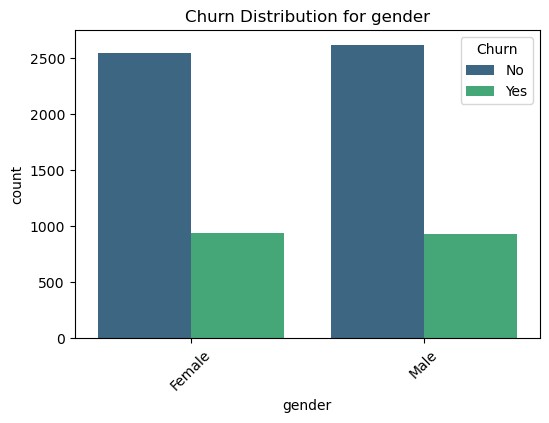
)

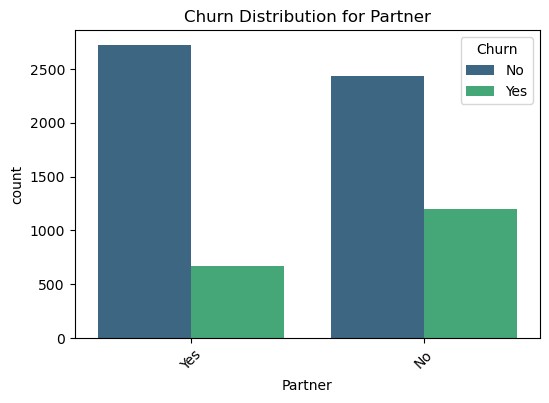
*# Rotate x-axis labels for better readability*

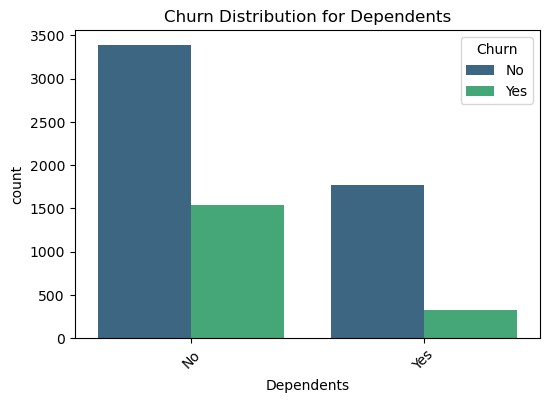
plt

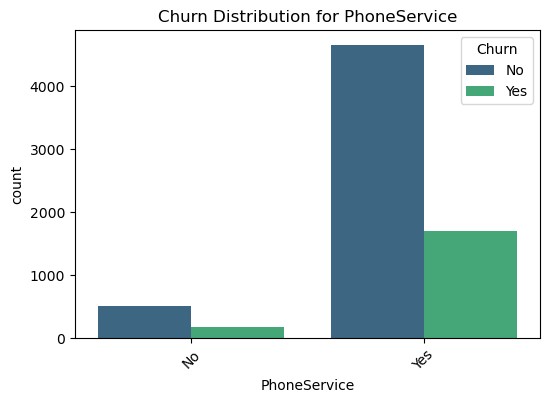
.

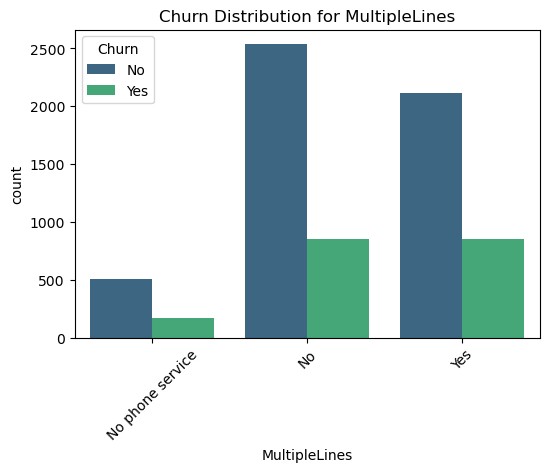
show()

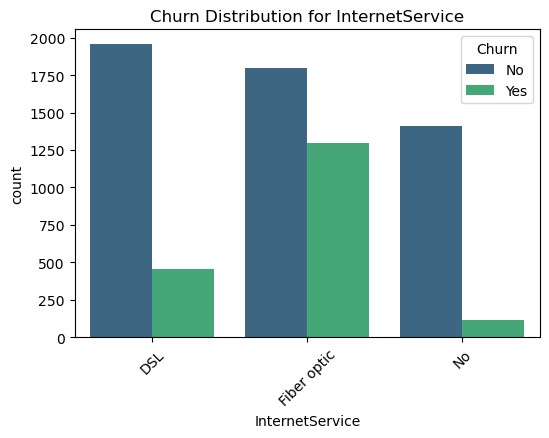


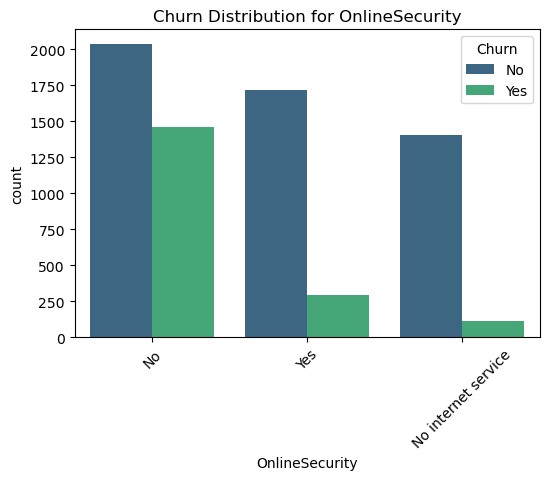


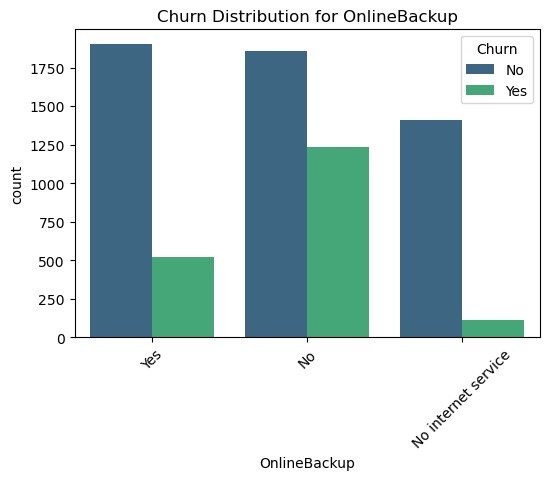


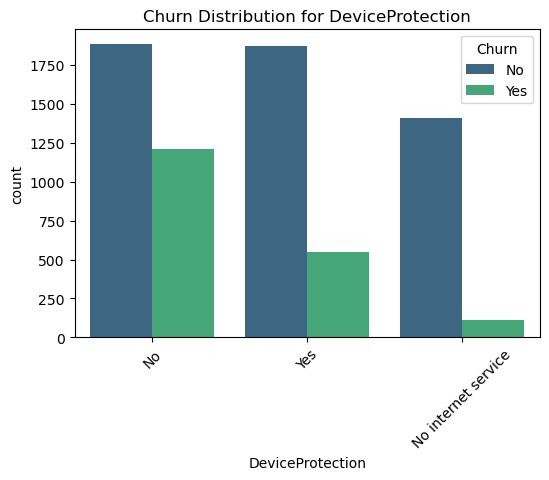


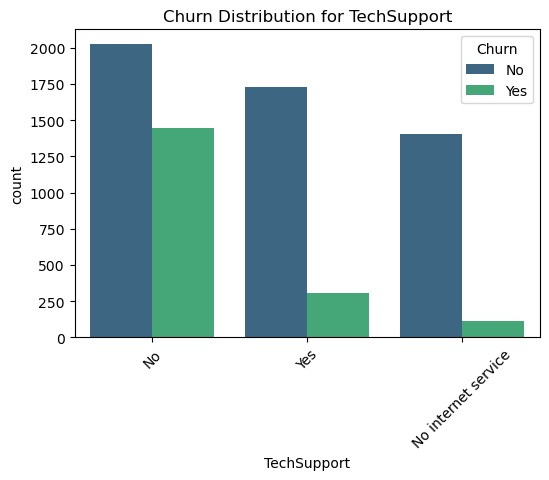


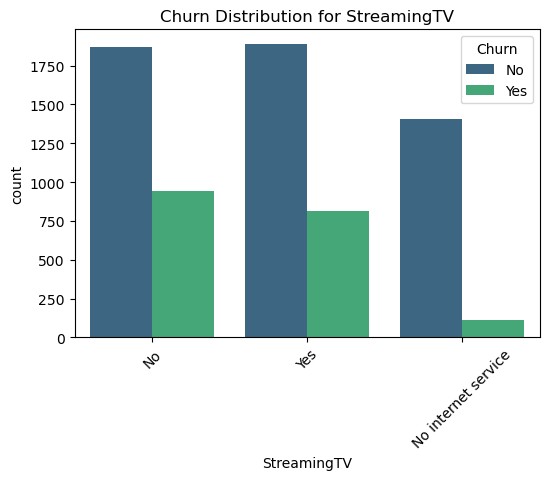


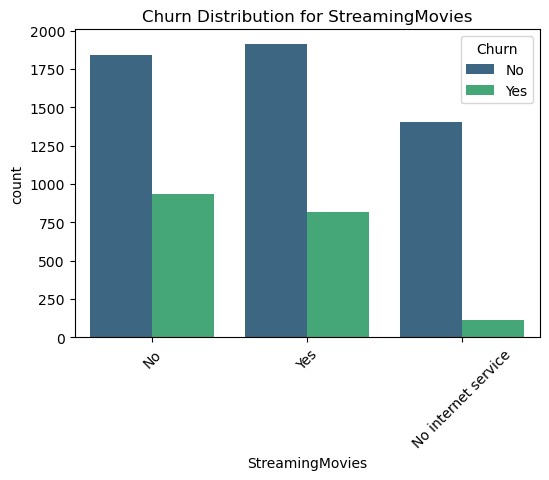


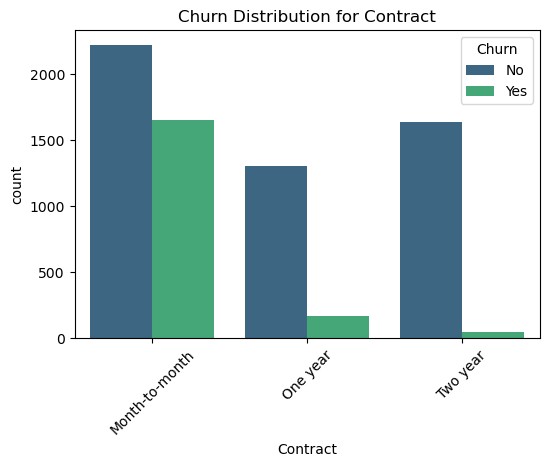


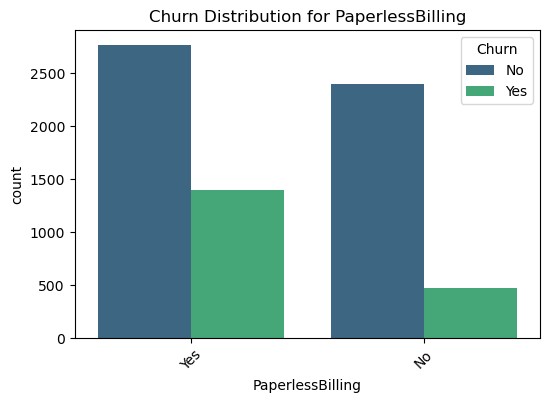


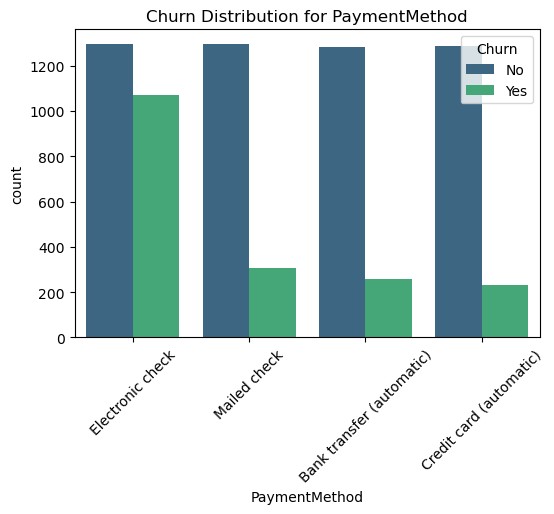


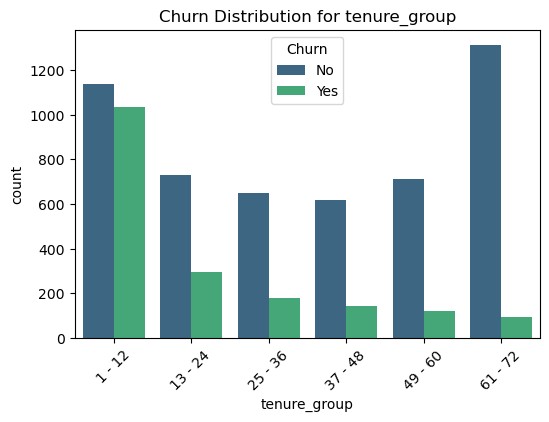












1. Convert the target variable ‘churn’ into a binary numeric variable i.e Yes = 1 and No = 0

[22]:

telco\_data[

'

Churn

'

]

=

np

.

where(telco\_data

.

Churn

==

'

Yes

'

,

1

,

0

)

[23]:

telco\_data

.

head()

[23]: gender SeniorCitizen Partner Dependents PhoneService MultipleLines \

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 Female | 0 | Yes | No | No No phone service |
| 1 Male | 0 | No | No | Yes No |
| 2 Male | 0 | No | No | Yes No |
| 3 Male | 0 | No | No | No No phone service |
| 4 Female | 0 | No | No | Yes No |

InternetService OnlineSecurity OnlineBackup DeviceProtection TechSupport \

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0 DSL No | | | Yes No | | | No |
| 1 DSL Yes | | | No Yes | | | No |
| 2 DSL Yes | | | Yes No | | | No |
| 3 DSL Yes | | | No Yes | | | Yes |
| 4 Fiber optic No | | | No No | | | No |
| StreamingTV StreamingMovies | | | Contract PaperlessBilling \ | | |  |
| 0 No No Month-to-month Yes | | | | | |
| 1 | No | No One year | | No |
| 2 | No | No Month-to-month | | Yes |
| 3 | No | No One year | | No |
| 4 | No | No Month-to-month | | Yes |

PaymentMethod MonthlyCharges TotalCharges Churn tenure\_group

|  |  |  |  |
| --- | --- | --- | --- |
| 0 Electronic check | 29.85 | 29.85 | 0 1 - 12 |
| 1 Mailed check | 56.95 | 1889.50 | 0 25 - 36 |
| 2 Mailed check | 53.85 | 108.15 | 1 1 - 12 |
| 3 Bank transfer (automatic) | 42.30 | 1840.75 | 0 37 - 48 |
| 4 Electronic check | 70.70 | 151.65 | 1 1 - 12 |

1. Convert all the categorical variables into dummy variables

[24]:

telco\_data\_dummies

=

pd

.

get\_dummies(telco\_data)

telco\_data\_dummies

.

head()

[24]: SeniorCitizen MonthlyCharges TotalCharges Churn gender\_Female \

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 0 | 29.85 | 29.85 | 0 | True |
| 1 | 0 | 56.95 | 1889.50 | 0 | False |
| 2 | 0 | 53.85 | 108.15 | 1 | False |
| 3 | 0 | 42.30 | 1840.75 | 0 | False |
| 4 | 0 | 70.70 | 151.65 | 1 | True |

gender\_Male Partner\_No Partner\_Yes Dependents\_No Dependents\_Yes … \

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | False | False | True | True | False … |
| 1 | True | True | False | True | False … |
| 2 | True | True | False | True | False … |
| 3 | True | True | False | True | False … |
| 4 | False | True | False | True | False … |

PaymentMethod\_Bank transfer (automatic) \

1. False
2. False
3. False
4. True
5. False

PaymentMethod\_Credit card (automatic) PaymentMethod\_Electronic check \

|  |  |  |
| --- | --- | --- |
| 0 | False | True |
| 1 | False | False |
| 2 | False | False |
| 3 | False | False |
| 4 | False | True |

PaymentMethod\_Mailed check tenure\_group\_1 - 12 tenure\_group\_13 - 24 \

1. False True False
2. True False False

|  |  |  |  |
| --- | --- | --- | --- |
| 2 | True | True | False |
| 3 | False | False | False |
| 4 | False | True | False |

tenure\_group\_25 - 36 tenure\_group\_37 - 48 tenure\_group\_49 - 60 \

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | False | False | False |
| 1 | True | False | False |
| 2 | False | False | False |
| 3 | False | True | False |
| 4 | False | False | False |

tenure\_group\_61 - 72

1. False
2. False
3. False
4. False
5. False

[5 rows x 51 columns]

4. Relationship between monthlycharges and total charges

[25]:

sns

.

lmplot(data

=

telco\_data\_dummies, x

=

'

MonthlyCharges

'

,y

=

'

TotalCharges

'

,

␣

↪

fit\_reg

=

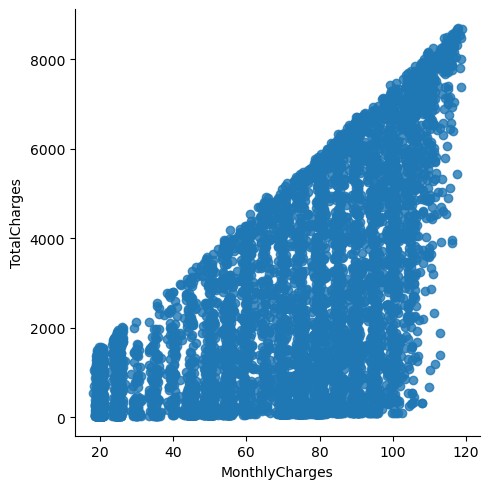
**False**

)

plt

.

show()



Total charges increases as monthly charges increase - as excepted

[26]: Mth = sns.kdeplot(telco\_data\_dummies.

↪MonthlyCharges[(telco\_data\_dummies["Churn"] == 0) ], color='Red', fill =␣

↪**True**)

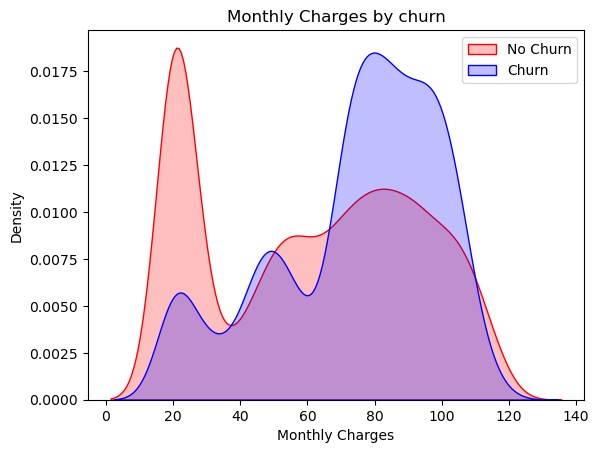
Mth = sns.kdeplot(telco\_data\_dummies.

↪MonthlyCharges[(telco\_data\_dummies["Churn"] == 1) ], ax = Mth, color='Blue',␣ ↪fill = **True**)

Mth.legend(["No Churn","Churn"], loc = "upper right")

Mth.set\_ylabel('Density')

Mth.set\_xlabel('Monthly Charges') Mth.set\_title('Monthly Charges by churn') plt.show()



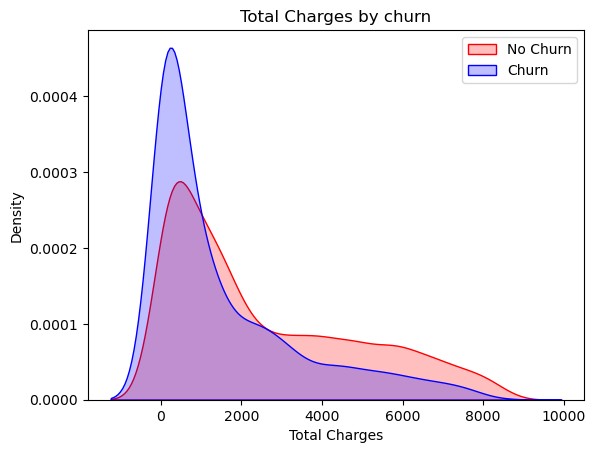
# 0.4 Insight: Churn is high when monthly charges are high

[27]: tot = sns.kdeplot(telco\_data\_dummies.TotalCharges[(telco\_data\_dummies["Churn"]␣

↪== 0) ], color='Red', fill = **True**) tot = sns.kdeplot(telco\_data\_dummies.TotalCharges[(telco\_data\_dummies["Churn"]␣

↪== 1) ], ax = tot, color='Blue', fill = **True**) tot.legend(["No Churn","Churn"], loc = "upper right")

tot.set\_ylabel('Density') tot.set\_xlabel('Total Charges') tot.set\_title('Total Charges by churn') plt.show()



# 0.5 Insigt Higher churn at lower total charges

However if we combine the 3 main features tenure, monthly charges, & total charges we get better insights. :- Higher monthly charge at lower tenure results into lower total charge. Hence all the these 3 attributes are linked to High Churn

5.

Correlation of all predictors with ‘Churn’

[28]:

plt

.

figure(figsize

=

(

20

,

8

))

telco\_data\_dummies

.

corr()[

'

Churn

'

]

.

sort\_values(ascending

=

**False**

)

.

↪

plot(kind

=

'

bar

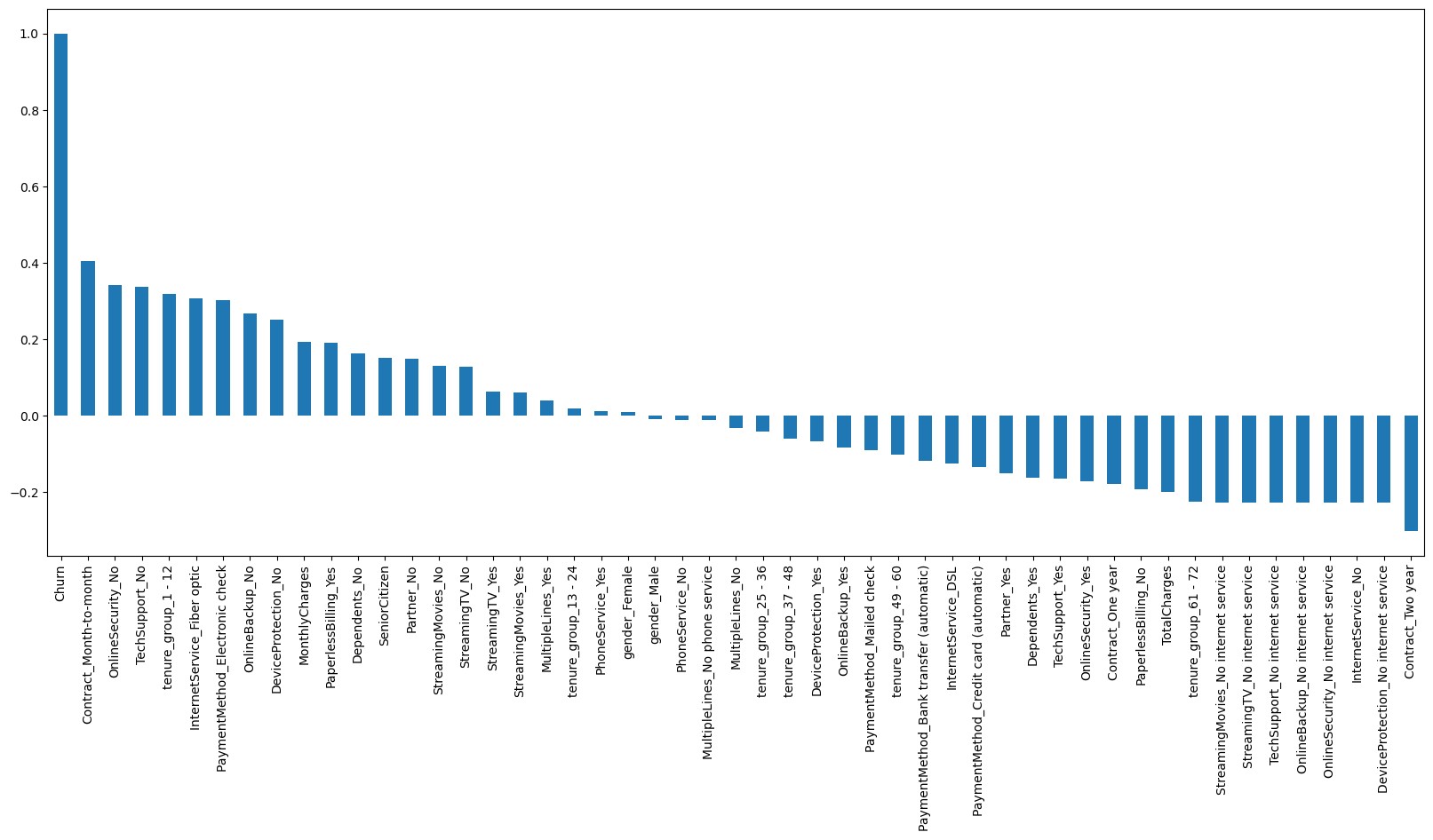
'

)

plt

.

show()



# 0.6 Derived Insight

high churn seen in case of month to month contrats. No online security. No tech support. First year of subscription fiber optics internet and LOW churn is seen in case of Long term contracts, subscription without internet service and the customer engaged for over 5+ years. factors like gender, availability of phone service and multiple lines have no impact on Churn.

# 0.7 2.Bivariant Analysis

[29]:

new\_df1\_target0

=

telco\_data

.

loc[telco\_data[

'

Churn

'

]

==

0

]

new\_df1\_target1

=

telco\_data

.

loc[telco\_data[

'

Churn

'

]

==

1

]

[30]:

**def**

bivariate\_plot

(

df, col, hue\_col, title

):

*"""*

*Plots a countplot for a given categorical column against another*

␣

↪

*categorical column.*

*Parameters:*

*-*

*df: DataFrame containing the data*

*-*

*col: Column name for the x-axis*

*-*

*hue\_col: Column name for hue (grouping*

*)*

*-*

*title: Title for the plot*

*"""*

sns

.

set\_style(

'

whitegrid

'

)

sns

.

set\_context(

'

talk

'

)

*# Dynamic figure size based on number of categories*

unique\_vals = df[col].nunique()

width = min(max(8, unique\_vals \* 1.5), 15) *# Ensures it doesn't get too*␣

↪*wide* fig, ax = plt.subplots(figsize=(width, 6))

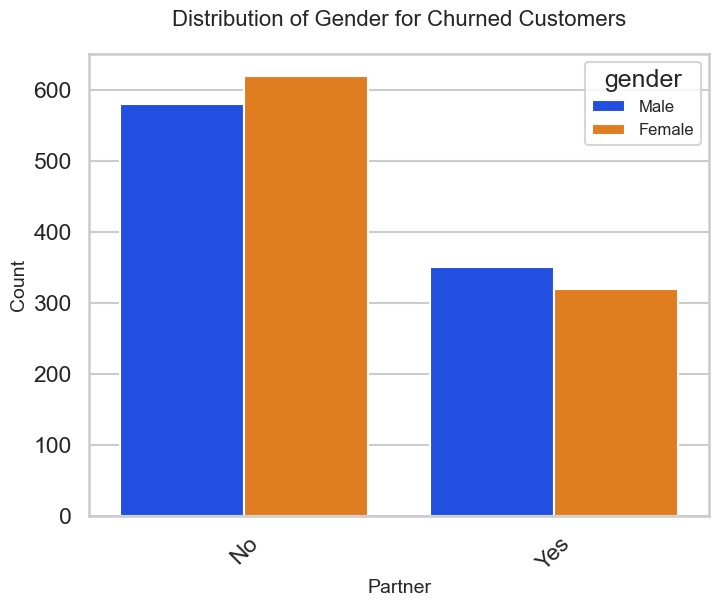
*# Plot countplot* sns.countplot(data=df, x=col, hue=hue\_col, order=df[col].value\_counts(). ↪index, palette="bright")

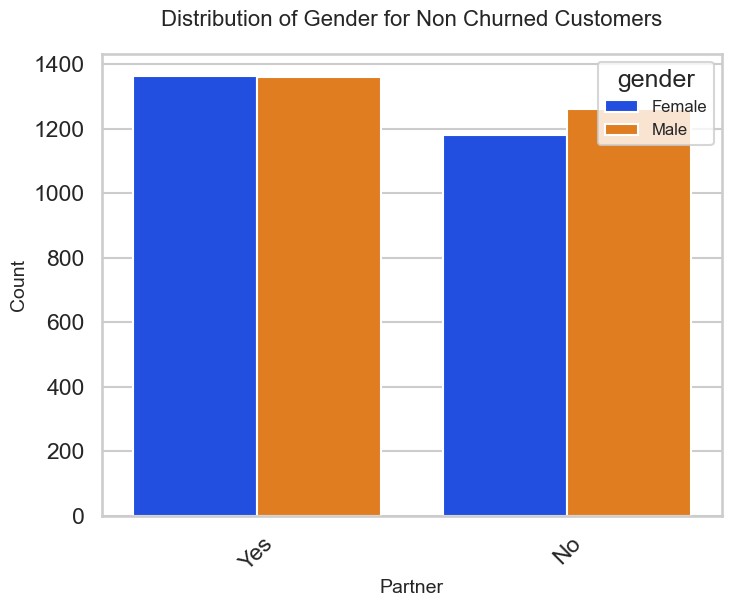
*# Formatting* plt.xticks(rotation=45) plt.title(title, fontsize=16, pad=20) plt.xlabel(col, fontsize=14) plt.ylabel("Count", fontsize=14) plt.legend(title=hue\_col, fontsize=12) plt.show()

[31]: bivariate\_plot(new\_df1\_target1, col="Partner", hue\_col="gender",␣

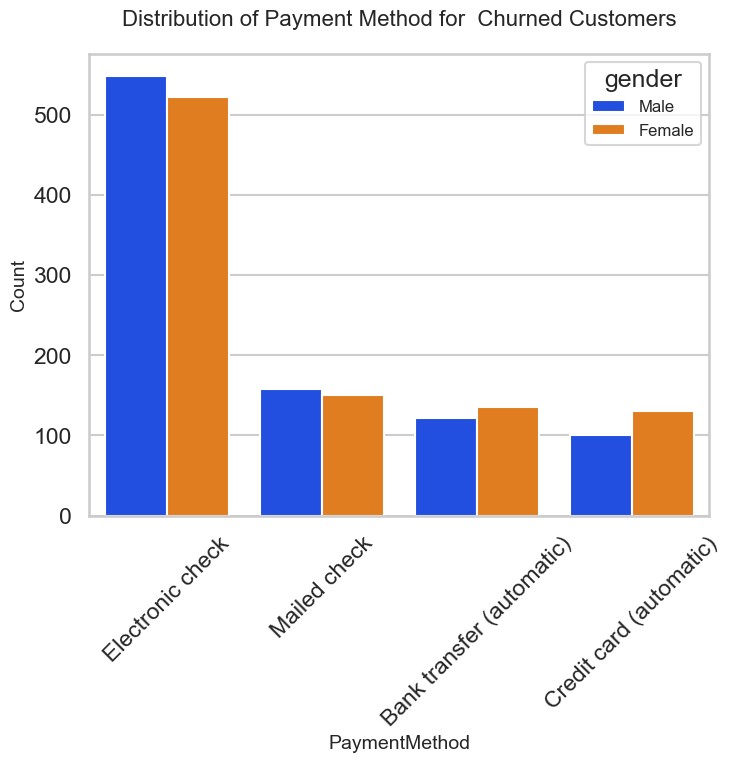
↪title="Distribution of Gender for Churned Customers") bivariate\_plot(new\_df1\_target0, col="Partner", hue\_col="gender",␣

↪title="Distribution of Gender for Non Churned Customers")

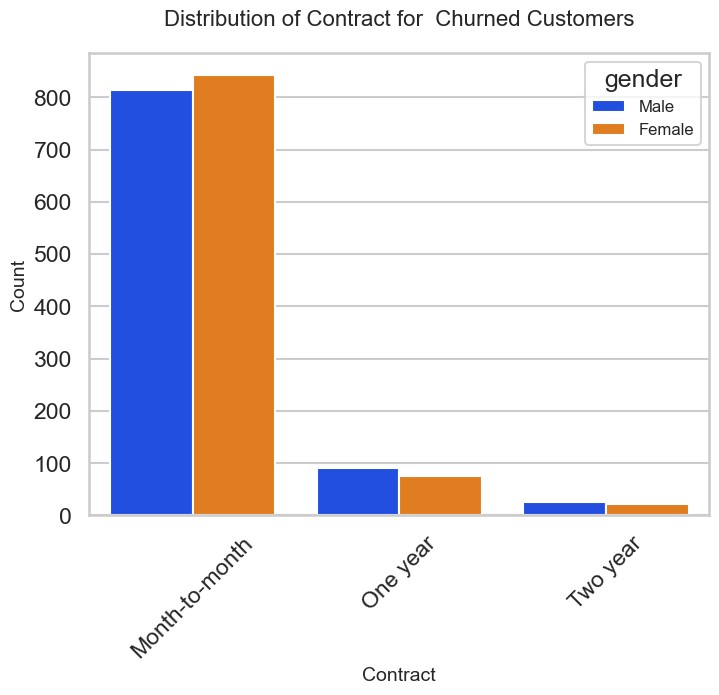




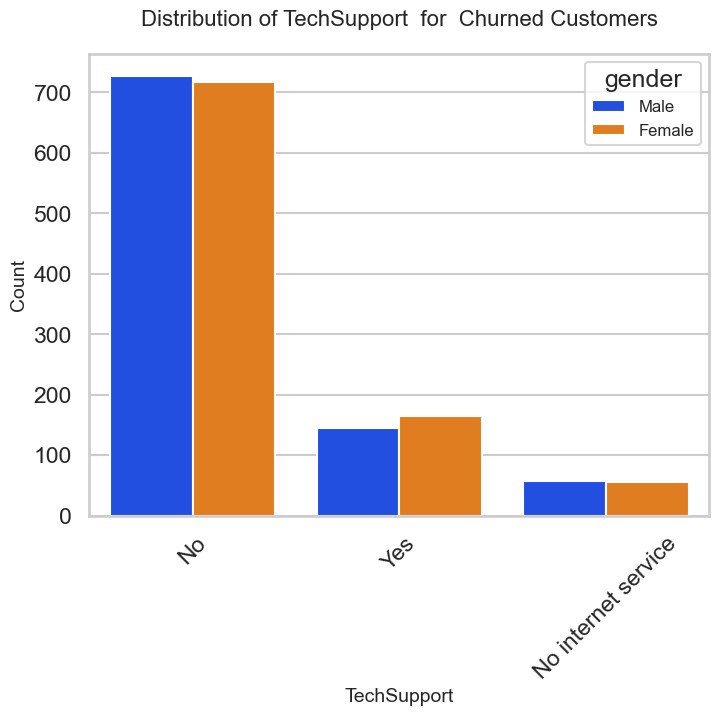
[32]: bivariate\_plot(new\_df1\_target1, col="PaymentMethod", hue\_col="gender",␣ ↪title="Distribution of Payment Method for Churned Customers")



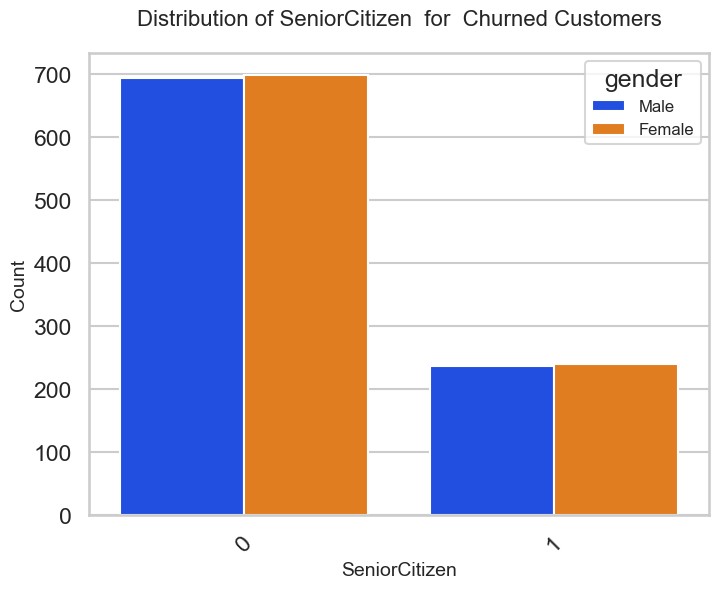
[33]: bivariate\_plot(new\_df1\_target1, col="Contract", hue\_col="gender",␣ ↪title="Distribution of Contract for Churned Customers")



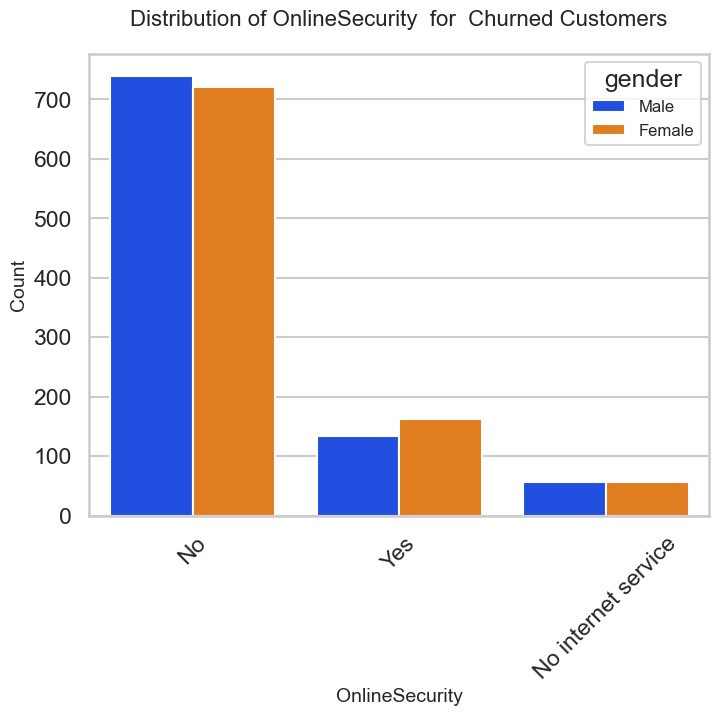
[34]: bivariate\_plot(new\_df1\_target1, col="TechSupport", hue\_col="gender",␣ ↪title="Distribution of TechSupport for Churned Customers")



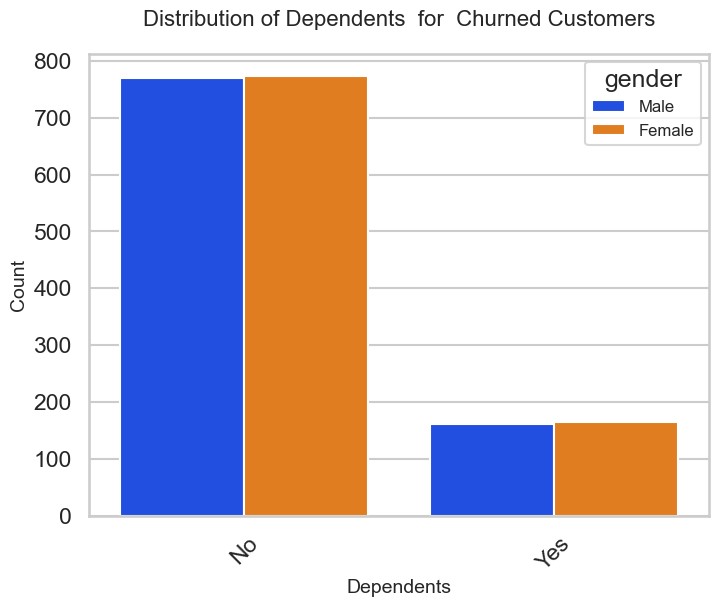
[35]: bivariate\_plot(new\_df1\_target1, col="SeniorCitizen", hue\_col="gender",␣ ↪title="Distribution of SeniorCitizen for Churned Customers")



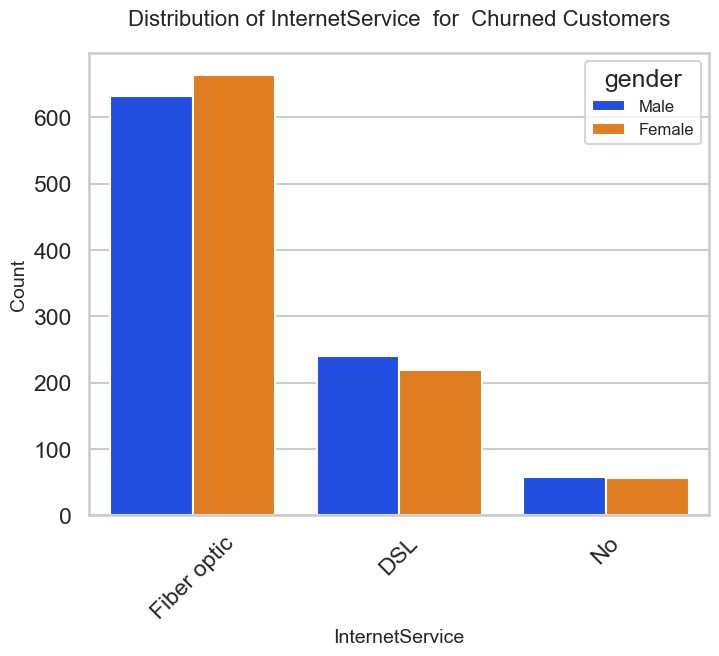
[36]: bivariate\_plot(new\_df1\_target1, col="OnlineSecurity", hue\_col="gender",␣ ↪title="Distribution of OnlineSecurity for Churned Customers")



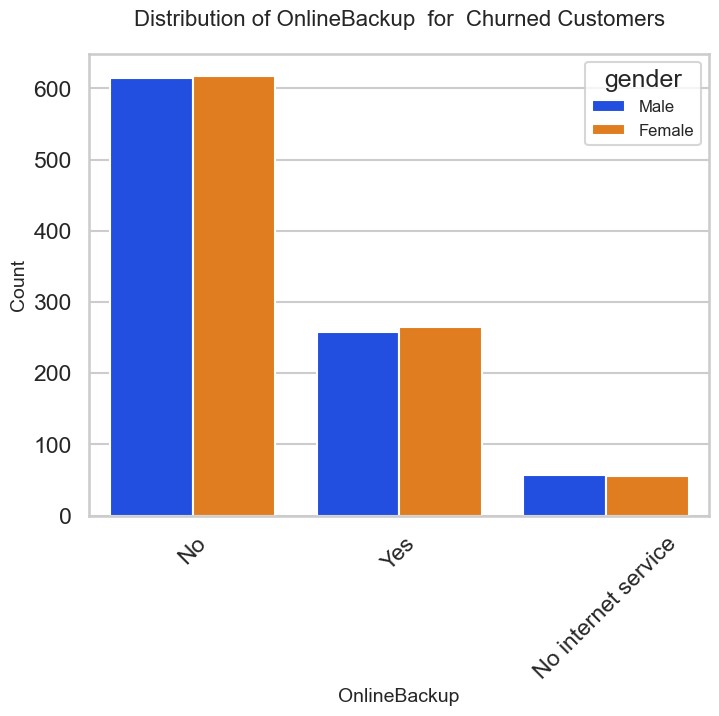
[37]: bivariate\_plot(new\_df1\_target1, col="Dependents", hue\_col="gender",␣ ↪title="Distribution of Dependents for Churned Customers")



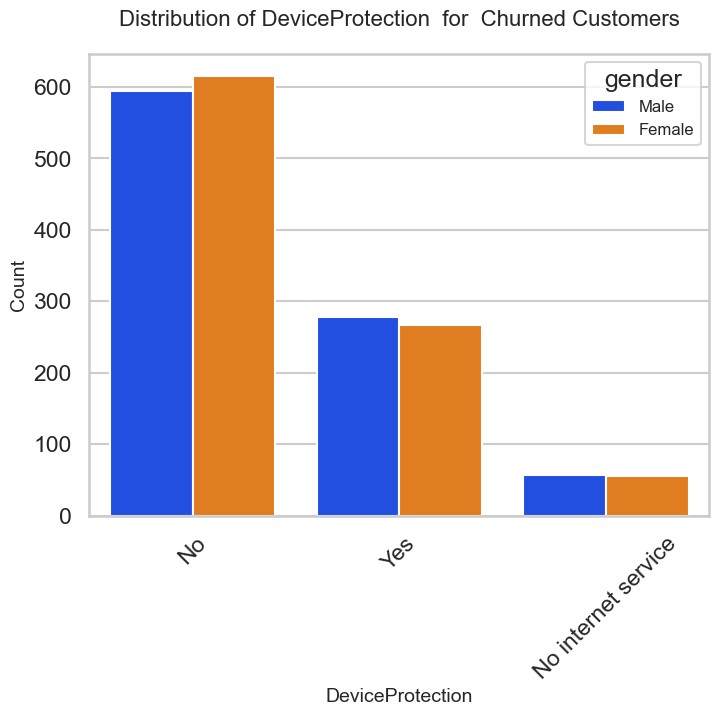
[38]: bivariate\_plot(new\_df1\_target1, col="InternetService", hue\_col="gender",␣ ↪title="Distribution of InternetService for Churned Customers")



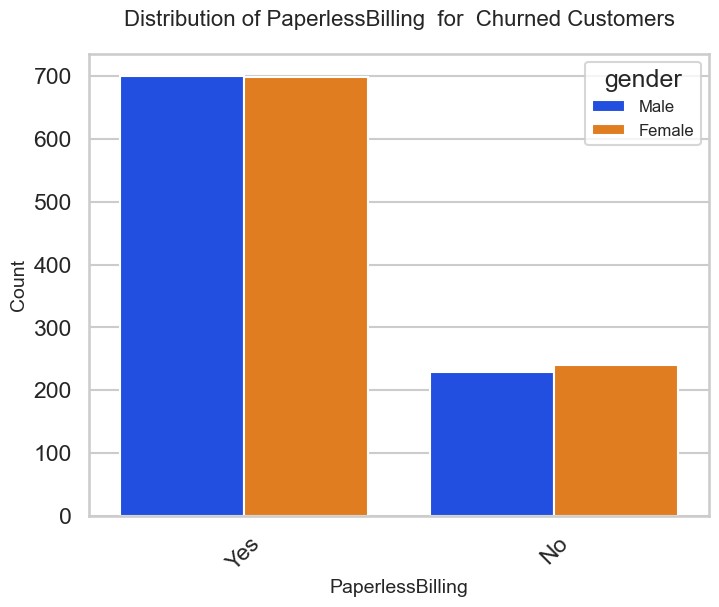
[39]: bivariate\_plot(new\_df1\_target1, col="OnlineBackup", hue\_col="gender",␣ ↪title="Distribution of OnlineBackup for Churned Customers")



[40]: bivariate\_plot(new\_df1\_target1, col="DeviceProtection", hue\_col="gender",␣ ↪title="Distribution of DeviceProtection for Churned Customers")



[41]: bivariate\_plot(new\_df1\_target1, col="PaperlessBilling", hue\_col="gender",␣ ↪title="Distribution of PaperlessBilling for Churned Customers")



# 0.8 CONCLUSION

These are some of the insights from this Dataset:

1. Electronic check medium are the highest chuners
2. Contract Type - Monthly customers are more likely to churn because of no contract terms, as they are free to go customers.
3. No Online security, No Tech Support category are high chumers
4. Non senior Citizens are high churners

[42]:

telco\_data\_dummies

.

to\_csv(

'

tel\_churn.csv

'

)