Customer Churn Prediction Using Machine Learning

Phase 3 Submission Document

Name	P. Sabari Raj
Reg. No	410121104039
NM ID	Au410121104029
Department	CSE-III
Domain	Data Analytics with Cognos
Project Title	Customer Churn Prediction
Phase 3	Development Part I
College	4101-Adhi College of Engineering and Technology, Kanchipuram

Customer Churn Prediction

Introduction to Telco Customer Churn:

In the dynamic and fiercely competitive telecommunications (telco) industry, customer churn is a persistent challenge that can significantly impact a company's bottom line and market position. Customer churn, also known as customer attrition or turnover, occurs when subscribers decide to switch their telecom service providers. This phenomenon is driven by a myriad of factors, including pricing, service quality, customer service, and evolving technology. To combat this issue, telco companies employ various strategies and initiatives aimed at retaining their customers, collectively known as customer retention programs. This introduction provides an overview of the critical concept of customer churn in the telco industry and the need for effective customer retention efforts to mitigate its negative consequences.

Given Data Set:

WA_Fn-UseC_-Telco-Customer-Churn.csv (977.5 kB)

Detail Compact Column 21 of 21 columns

customerID	gender	SeniorCiti...	Partner	Dependents	tenure	PhoneSer...	MultipleLi...	InternetSe...	OnlineSec...	OnlineBac...
7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	Yes
5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	No
3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	Yes
7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	No
9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	No
9305-CDSKC	Female	0	No	No	8	Yes	Yes	Fiber optic	No	No
1452-KIOVK	Male	0	No	Yes	22	Yes	Yes	Fiber optic	No	Yes
6713-OKOMC	Female	0	No	No	10	No	No phone service	DSL	Yes	No
7892-POOKP	Female	0	Yes	No	28	Yes	Yes	Fiber optic	No	No
6388-TABGU	Male	0	No	Yes	62	Yes	No	DSL	Yes	Yes
9763-GRSKD	Male	0	Yes	Yes	13	Yes	No	DSL	Yes	No
7469-LKBCI	Male	0	No	No	16	Yes	No	No	No internet service	No internet service
8091-TTVAX	Male	0	Yes	No	58	Yes	Yes	Fiber optic	No	No

Necessary step to follow:

1.Import Libraries:

Program:

```
import pandas as pd

from sklearn import metrics

from sklearn.model_selection import train_test_split

from sklearn.metrics import recall_score

from sklearn.metrics import classification_report

from sklearn.metrics import confusion_matrix

from sklearn.tree import DecisionTreeClassifier

from imblearn.combine import SMOTEENN
```

2. Load the Dataset:

```
df=pd.read_csv("C:\Users\PAZHANI SABARI RAJ\Downloads\MLProject-ChurnPrediction-main\MLProject-ChurnPrediction-main\tel_churn.csv")

df.head()
```

3. Exploratory Data Analysis(EDA):

- Exploratory Data Analysis is an approach to analyse the datasets to summarize their main characteristics in form of visual methods.
- EDA is nothing but an data exploration technique to understand various aspects of the data.
- The main aim of EDA is to obtain confidence in a data to an extent where we are ready to engage a machine learning model.
- EDA is important to analyse the data it’s a first steps in data analysis process.
- EDA give a basic idea to understand the data and make sense of the data to figure out the question you need to ask and find out the best way to manipulate the dataset to get the answer of your question.
- Exploratory data analysis help us to finding the errors, discovering data, mapping out data structure, finding out anomalies.

- ❖ Exploratory data analysis is important for business process because we are preparing dataset for deep through analysis that will detect you business problem.
- ❖ EDA help to build a quick and dirty model, or a baseline model, which can serve as a comparison against later models that you will build.

Programs:

```
#Check the various attributes of data like shape (rows and cols), Columns, datatypes

telco_base_data.shape

telco_base_data.shape

telco_base_data.columns.values
```

Output:

```
array(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',

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

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

       'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract',

       'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges',

       'TotalCharges', 'Churn'], dtype=object)
```

```
# Checking the data types of all the columns

telco_base_data.dtypes
```

Output:

```
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               object
dtype: object
```

```
# Check the descriptive statistics of numeric variables
telco_base_data.describe()
```

Output:

	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

\

SeniorCitizen is actually a categorical hence the 25%-50%-75% distribution is not proper.

75% customers have tenure less than 55 months.

Average Monthly charges are USD 64.76 whereas 25% customers pay more than USD 89.85 per month.

4.Feature Engineering:

Depending on your dataset, you may need to create new features or transform existing ones. This can involve one-hot encoding categorical variables, handling data/time data, or scaling numerical features.

5. Split the Data:

Split your dataset into training and testing sets. This helps you evaluate your model's performance later.

```
#Train Test Split

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
```

Steps Involved in EDA:=

- Data Sourcing
- Data Cleaning
- Univariate Analysis with Visualisation
- Bivariate Analysis with Visualisation
- Derived Metrics

Data Sourcing:

Data Sourcing is the process of gathering data from multiple sources as external or internal data collection.

There are two major kind of data which can be classified according to the source:

- Public data
- Private data

Public Data:- The data which is easy to access without taking any permission from the agencies is called public data. The agencies made the data public for the purpose of the research. Like government and other public sector or ecommerce sites made there data public.

Private Data:- The data which is not available on public platform and to access the data we have to take the permission of organisation is called private data. Like Banking ,telecom ,retail sector are there which not made their data publicly available.

The following are some steps involve in Data Cleaning:

- Handle Missing Values
- Standardisation of the data
- Outlier Treatment
- Handle Invalid values

Missing Data - Initial Intuition

Here, we don't have any missing data.

General Thumb Rules:

- For features with less missing values- can use regression to predict the missing values or fill with the mean of the values present, depending on the feature.
- For features with very high number of missing values- it is better to drop those columns as they give very less insight on analysis.
- As there's no thumb rule on what criteria do we delete the columns with high number of missing values, but generally you can delete the columns, if you have more than 30-40% of missing values. But again there's a catch here, for example, Is_Car & Car_Type, People having no cars, will obviously have Car_Type as NaN (null), but that doesn't make this column useless, so decisions has to be taken wisely.

Data Cleaning:

Data cleaning is the process of identifying and correcting errors, inconsistencies, and inaccuracies in a dataset to ensure its quality and reliability for analysis. This involves tasks such as handling missing values, removing duplicates, and addressing outliers to improve data integrity.

Program:

1. Create a copy of base data for manupulation & processing

```
telco_data = telco_base_data.copy()
```

2. Total Charges should be numeric amount. Let's convert it to numerical data type

```
telco_data.TotalCharges=pd.to_numeric(telco_data.TotalCharges,errors='coerce’)
```

```
telco_data.isnull().sum()
```

Output:

```
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             0
dtype: int64
```

3. As we can see there are 11 missing values in TotalCharges column. Let's check these records

telco_data.loc[telco_data ['TotalCharges'].isnull() == True]

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechSupport	Stre
488	4472-LVYGI	Female	0	Yes	Yes	0	No	No phone service	DSL	Yes	...	Yes	Yes	
753	3115-CZMZD	Male	0	No	Yes	0	Yes	No	No	No internet service	...	No internet service	No internet service	N
936	5709-LVOEQ	Female	0	Yes	Yes	0	Yes	No	DSL	Yes	...	Yes	No	
1082	4367-NUYAO	Male	0	Yes	Yes	0	Yes	Yes	No	No internet service	...	No internet service	No internet service	N
1340	1371-DWPAZ	Female	0	Yes	Yes	0	No	No phone service	DSL	Yes	...	Yes	Yes	
3331	7644-OMVMY	Male	0	Yes	Yes	0	Yes	No	No	No internet service	...	No internet service	No internet service	N
3826	3213-VVOLG	Male	0	Yes	Yes	0	Yes	Yes	No	No internet service	...	No internet service	No internet service	N
4380	2520-SGTTA	Female	0	Yes	Yes	0	Yes	No	No	No internet service	...	No internet service	No internet service	N
5218	2923-ARZLG	Male	0	Yes	Yes	0	Yes	No	No	No internet service	...	No internet service	No internet service	N
6670	4075-WKNIU	Female	0	Yes	Yes	0	Yes	Yes	DSL	No	...	Yes	Yes	
6754	2775-SEFEE	Male	0	No	Yes	0	Yes	Yes	DSL	Yes	...	No	Yes	

11 rows × 21 columns

4. Missing Value Treatement

Since the % of these records compared to total dataset is very low ie 0.15%, it is safe to ignore them from further processing.

```
#Removing missing values

telco_data.dropna(how = 'any', inplace = True)

#telco_data.fillna(0)
```

5. Divide customers into bins based on tenure e.g. for tenure < 12 months: assign a tenure group if 1-12, for tenure between 1 to 2 Yrs, tenure group of 13-24; so on...

```
# Get the max tenure

print(telco_data['tenure'].max()) #72
```

Output:

72

```
# Group the tenure in bins of 12 months

labels = ["{0}- {1}".format(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)

telco_data['tenure_group'].value_counts()
```

Output:

1- 12   2175

61- 72   1407

13- 24   1024

49- 60   832

25- 36   832

37- 48   762

Name: tenure_group, dtype: int64

6. Remove columns not required for processing

```
#drop column customerID and tenure

telco_data.drop(columns= ['customerID','tenure'], axis=1, inplace=True)

telco_data.head()
```

Output:

	gender	SeniorCitizen	Partner	Dependents	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport	StreamingTV	Str
0	Female	0	Yes	No	No	No phone service	DSL	No	Yes	No	No	No	
1	Male	0	No	No	Yes	No	DSL	Yes	No	Yes	No	No	
2	Male	0	No	No	Yes	No	DSL	Yes	Yes	No	No	No	
3	Male	0	No	No	No	No phone service	DSL	Yes	No	Yes	Yes	No	
4	Female	0	No	No	Yes	No	Fiber optic	No	No	No	No	No	

Data Exploration:

Data exploration refers to the process of examining and analyzing a dataset to understand its key characteristics, patterns, and relationships. It involves summarizing and visualizing data to gain insights and inform further data analysis and decision-making. Data exploration helps identify outliers, trends, and potential issues in the data, making it a crucial step in the data analysis process.

1. Plot distibution of individual predictors by churn

Univariate Analysis:

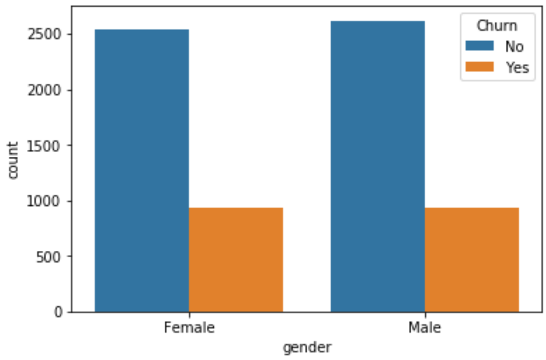
Segmented Univariate Analysis allow you to compare subset of data it help us to understand how the relevant metric varies across the different segment.

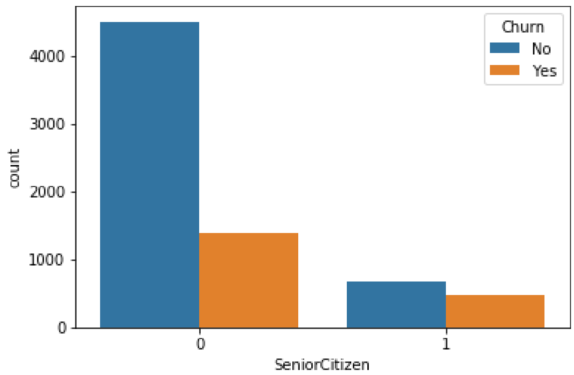
The Standard process of segmented univariate analysis is as follow:

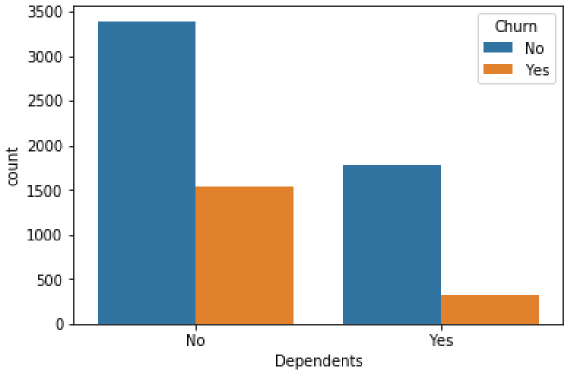
- Take a raw data
- Group by dimensions
- Summarise using a relevant metric like mean ,median.
- Compare the aggregate metric across the categories

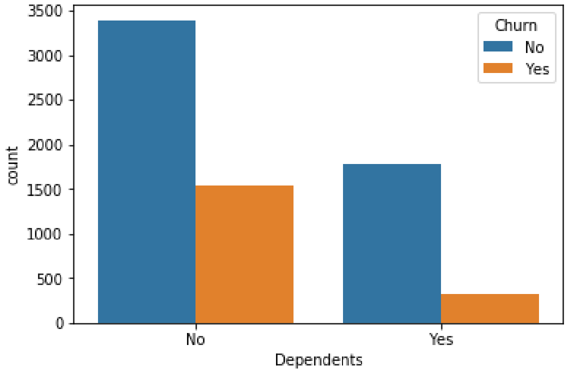
Program:

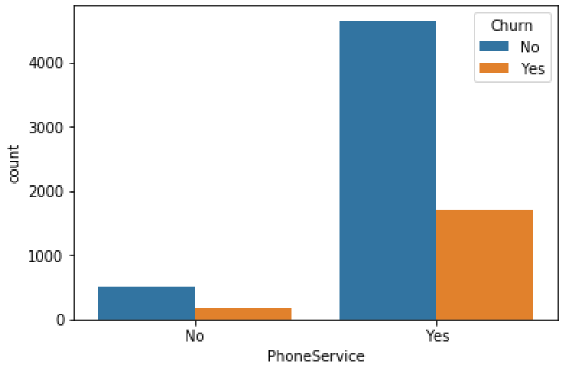
```
for i, predictor in enumerate(telco_data.drop(columns=['Churn', 'TotalCharges',
'MonthlyCharges'])):
    plt.figure(i)
    sns.countplot(data=telco_data, x=predictor, hue='Churn')
```

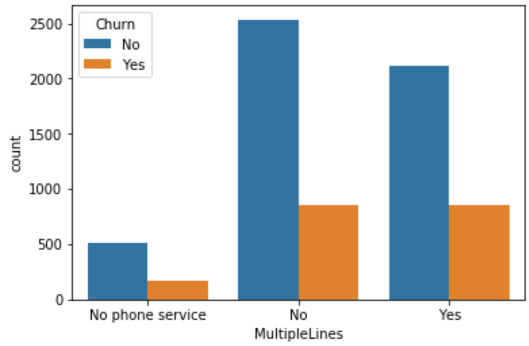


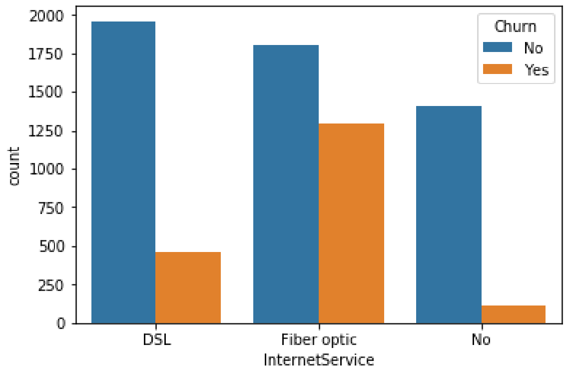


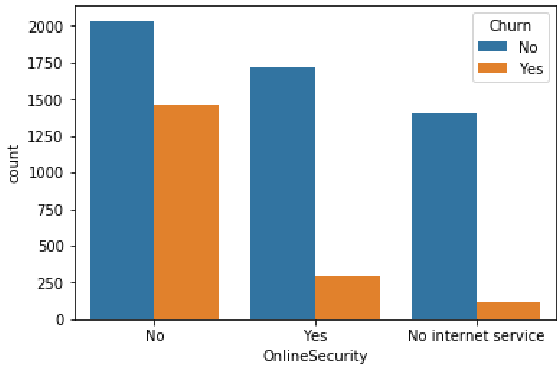


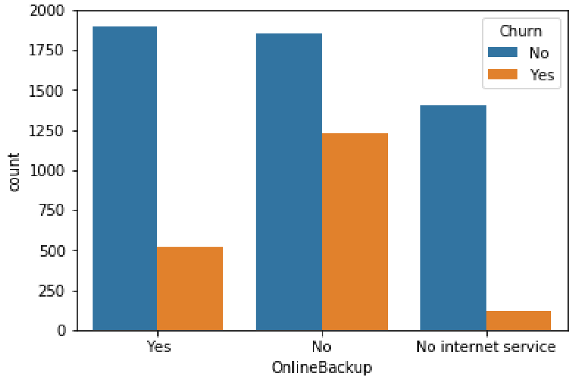


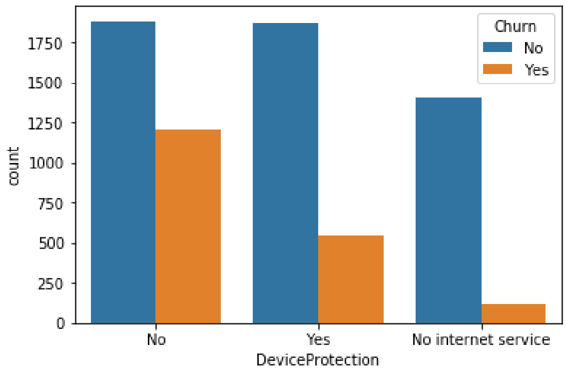

Churn
No
Yes
count
0
1000
2000
3000
4000
No
Yes
PhoneService

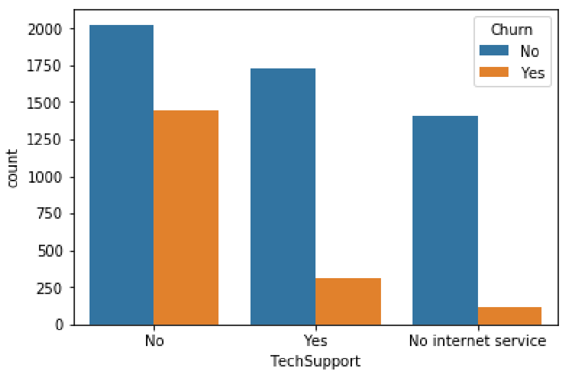

Churn
No
Yes
count
0
500
1000
1500
2000
2500
No phone service
No
Yes
MultipleLines

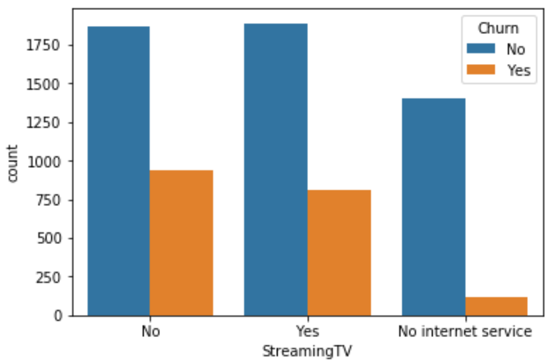

Churn
No
Yes
count
0
250
500
750
1000
1250
1500
1750
2000
DSL
Fiber optic
No
InternetService

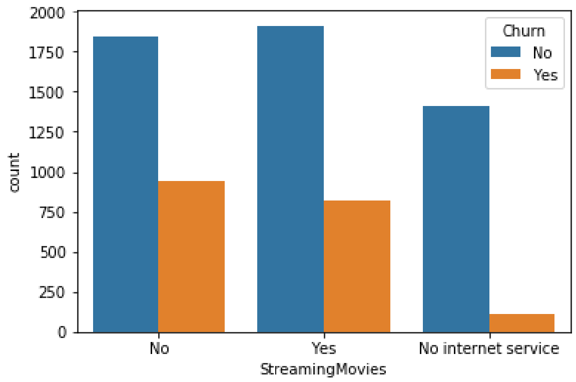

Churn
No
Yes
count
0
250
500
750
1000
1250
1500
1750
2000
No
Yes
No internet service
OnlineSecurity

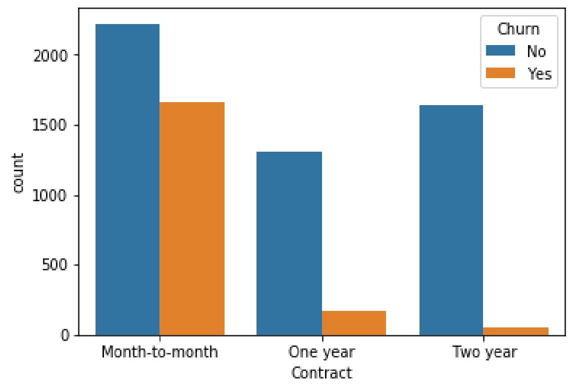

Churn
No
Yes
count
0
250
500
750
1000
1250
1500
1750
2000
Yes
No
No internet service
OnlineBackup

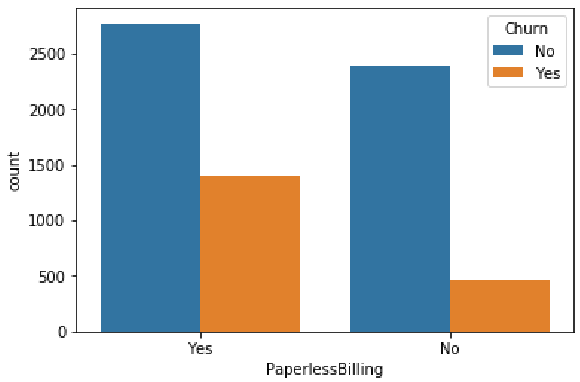

Churn
No
Yes
count
0
250
500
750
1000
1250
1500
1750
No
Yes
No internet service
DeviceProtection

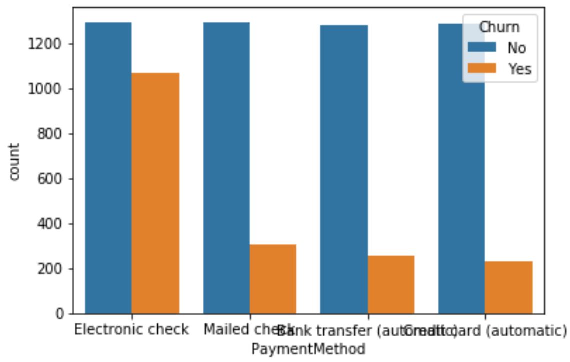

Churn
No
Yes
count
0
250
500
750
1000
1250
1500
1750
2000
No
Yes
No internet service
TechSupport

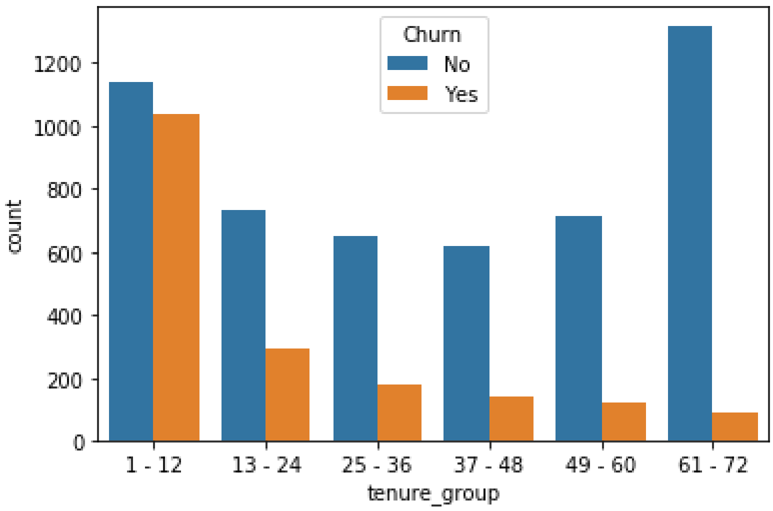

Churn
No
Yes
count
0
250
500
750
1000
1250
1500
1750
No
Yes
No internet service
StreamingTV











2. Convert the target variable 'Churn' in a binary numeric variable i.e. Yes=1 ; No = 0

```
telco_data['Churn'] = np.where(telco_data.Churn == 'Yes',1,0)
```

```
telco_data.head()
```

Output:

	gender	SeniorCitizen	Partner	Dependents	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport	StreamingTV	Str
0	Female	0	Yes	No	No	No phone service	DSL	No	Yes	No	No	No	
1	Male	0	No	No	Yes	No	DSL	Yes	No	Yes	No	No	
2	Male	0	No	No	Yes	No	DSL	Yes	Yes	No	No	No	
3	Male	0	No	No	No	No phone service	DSL	Yes	No	Yes	Yes	No	
4	Female	0	No	No	Yes	No	Fiber optic	No	No	No	No	No	

3. Convert all the categorical variables into dummy variables

telco_data_dummies = pd.get_dummies(telco_data)

telco_data_dummies.head()

Output:

	SeniorCitizen	MonthlyCharges	TotalCharges	Churn	gender_Female	gender_Male	Partner_No	Partner_Yes	Dependents_No	Dependents_Yes	...	PaymentMethod_Bank transfer (automatic)
0	0	29.85	29.85	0	1	0	0	1	1	0	...	0
1	0	56.95	1889.50	0	0	1	1	0	1	0	...	0
2	0	53.85	108.15	1	0	1	1	0	1	0	...	0
3	0	42.30	1840.75	0	0	1	1	0	1	0	...	1
4	0	70.70	151.65	1	1	0	1	0	1	0	...	0

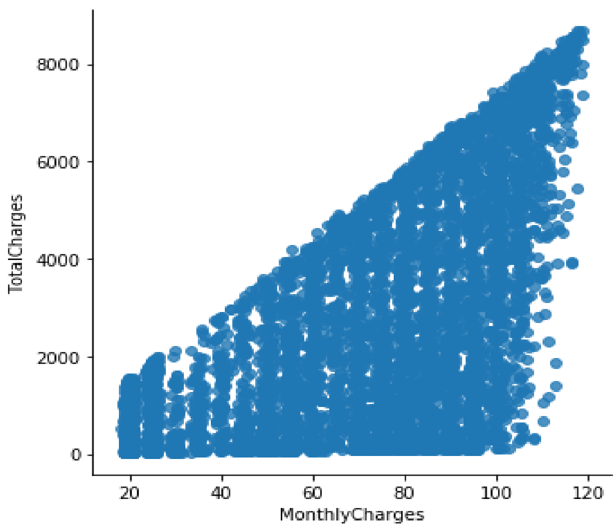
5 rows × 51 columns

4. Relationship between Monthly Charges and Total Charges

sns.lmplot(data=telco_data_dummies, x='MonthlyCharges', y='TotalCharges', fit_reg=False)

Output:

<seaborn.axisgrid.FacetGrid at 0x20d8a9289e8>



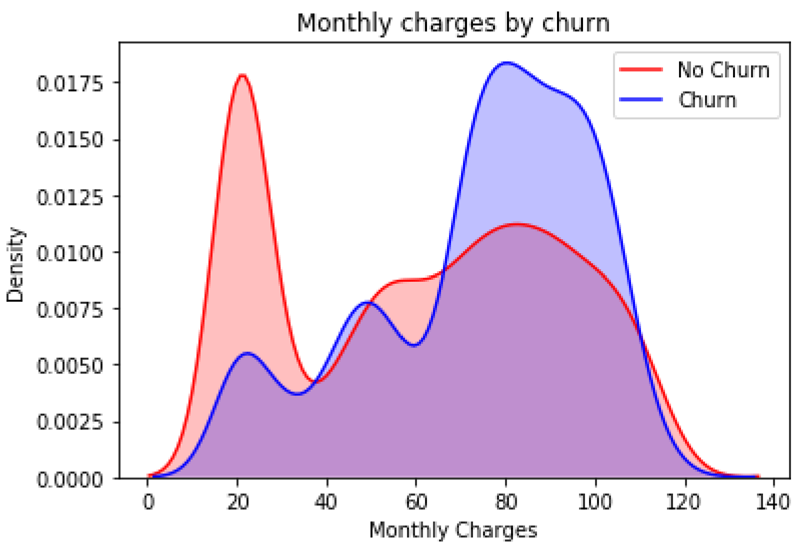
Total Charges increase as Monthly Charges increase - as expected.

5. Churn by Monthly Charges and Total Charges

```
Mth = sns.kdeplot(telco_data_dummies.MonthlyCharges[(telco_data_dummies["Churn"] == 0) ],
                  color="Red", shade = True)
Mth = sns.kdeplot(telco_data_dummies.MonthlyCharges[(telco_data_dummies["Churn"] == 1) ],
                  ax =Mth, color="Blue", shade= 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')
```

<u>Output:</u>

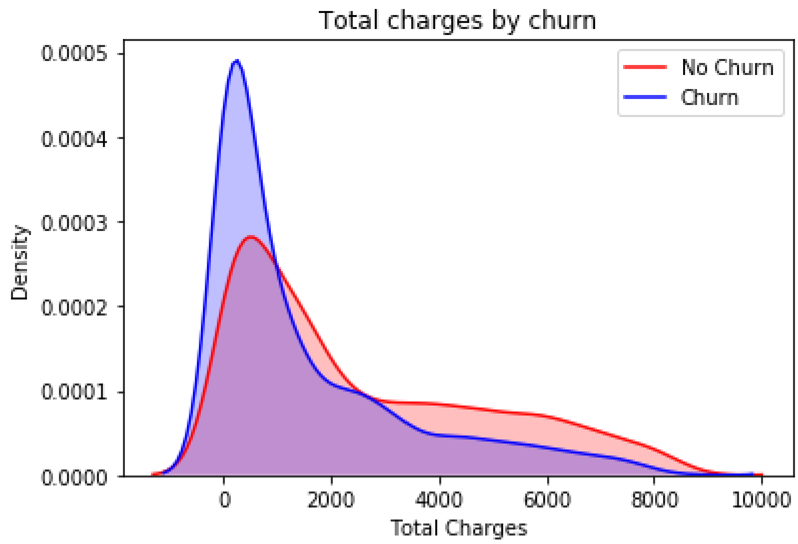
Text(0.5, 1.0, 'Monthly charges by churn')



Insight: Churn is high when Monthly Charges ar high

```
Tot = sns.kdeplot(telco_data_dummies.TotalCharges[(telco_data_dummies["Churn"] == 0) ],
                  color="Red", shade = True)
Tot = sns.kdeplot(telco_data_dummies.TotalCharges[(telco_data_dummies["Churn"] == 1) ],
                  ax =Tot, color="Blue", shade= 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')
```

Output:



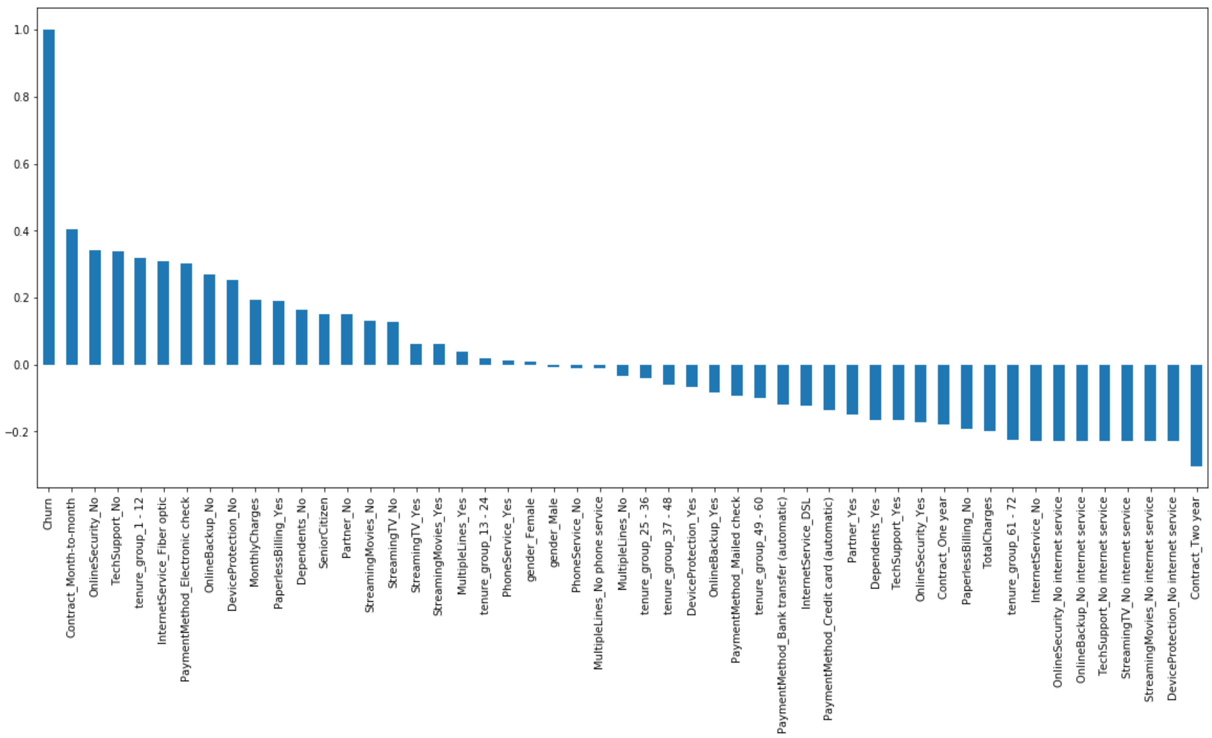
**Surprising insight ** as higher Churn at lower Total Charges

6. Build a corelation of all predictors with 'Churn'

```
plt.figure(figsize=(20,8))

telco_data_dummies.corr()['Churn'].sort_values(ascending = False).plot(kind='bar')
```

Output:

<matplotlib.axes._subplots.AxesSubplot at 0x20d8a979f98>



plt.figure(figsize=(12,12))

sns.heatmap(telco_data_dummies.corr(), cmap="Paired")

<u>Output:</u>

<matplotlib.axes._subplots.AxesSubplot at 0x1809ebfef60>

BivariateAnalysis Correlation:

Data which has two variables ,you often want to measure the relationship that exists between these two variables.

Bi-variate Types:

Correlation: Correlation measure the strength as well as the direction of the linear relationship between the two variables. Its range is from -1 to +1.

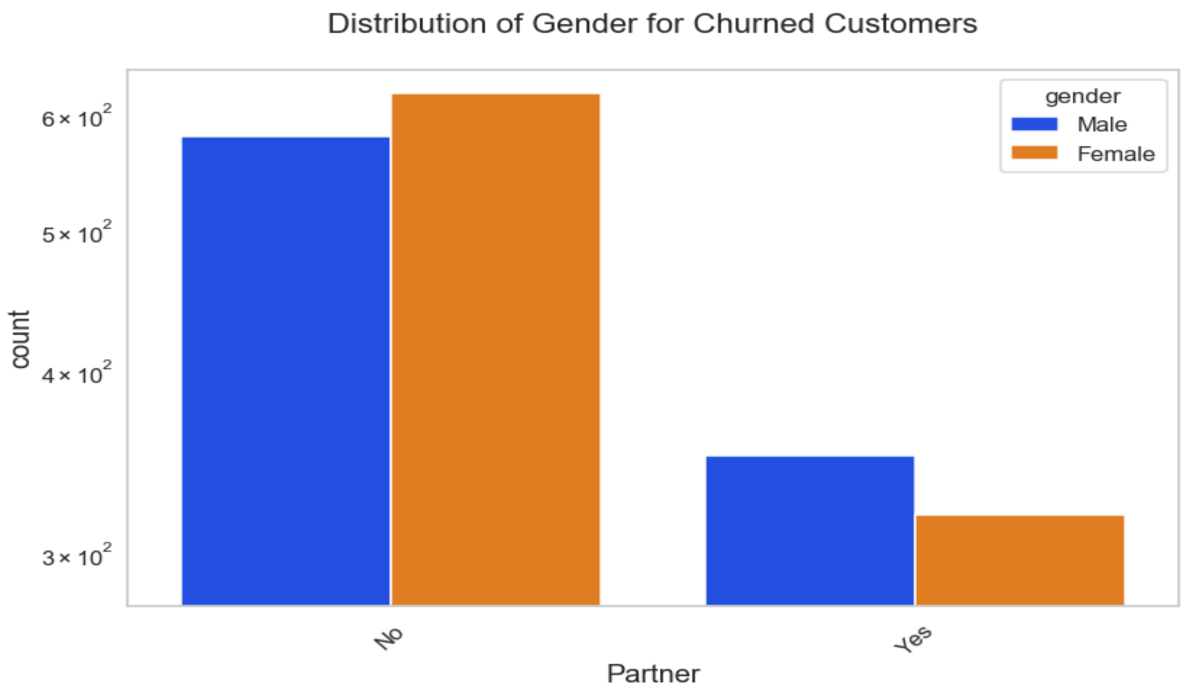
- ● If one increases as the other increases, the correlation is positive
- ● If one decreases as the other increases, the correlation is negative
- ● If one stays constant as the other varies, the correlation is zero

Covariance: Covariance measure how much two random variable vary together. Its range is from -∞ to +∞.

Program:

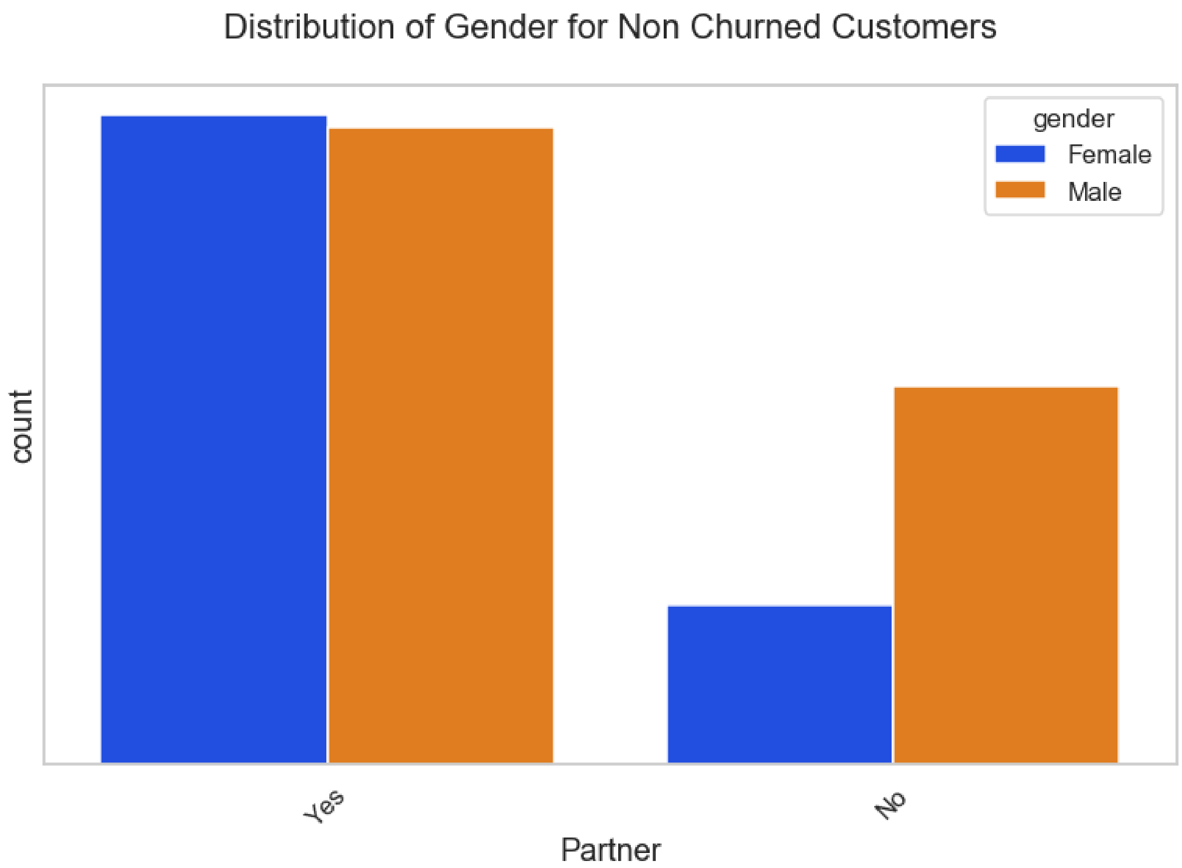
```
new_df1_target0=telco_data.loc[telco_data["Churn"]==0]

new_df1_target1=telco_data.loc[telco_data["Churn"]==1]

def uniplot(df,col,title,hue =None):

   sns.set_style('whitegrid')

   sns.set_context('talk')

   plt.rcParams["axes.labelsize"] = 20

   plt.rcParams['axes.titlesize'] = 22

   plt.rcParams['axes.titlepad'] = 30

   temp = pd.Series(data = hue)

   fig, ax = plt.subplots()

   width = len(df[col].unique()) + 7 + 4*len(temp.unique())

   fig.set_size_inches(width , 8)

   plt.xticks(rotation=45)

   plt.yscale('log')

   plt.title(title)

   ax = sns.countplot(data = df, x= col, order=df[col].value_counts().index,hue = hue,palette='bright')

   plt.show()

niplot(new_df1_target1,col='Partner',title='Distribution of Gender for Churned Customers',hue='gender')
```

Output:



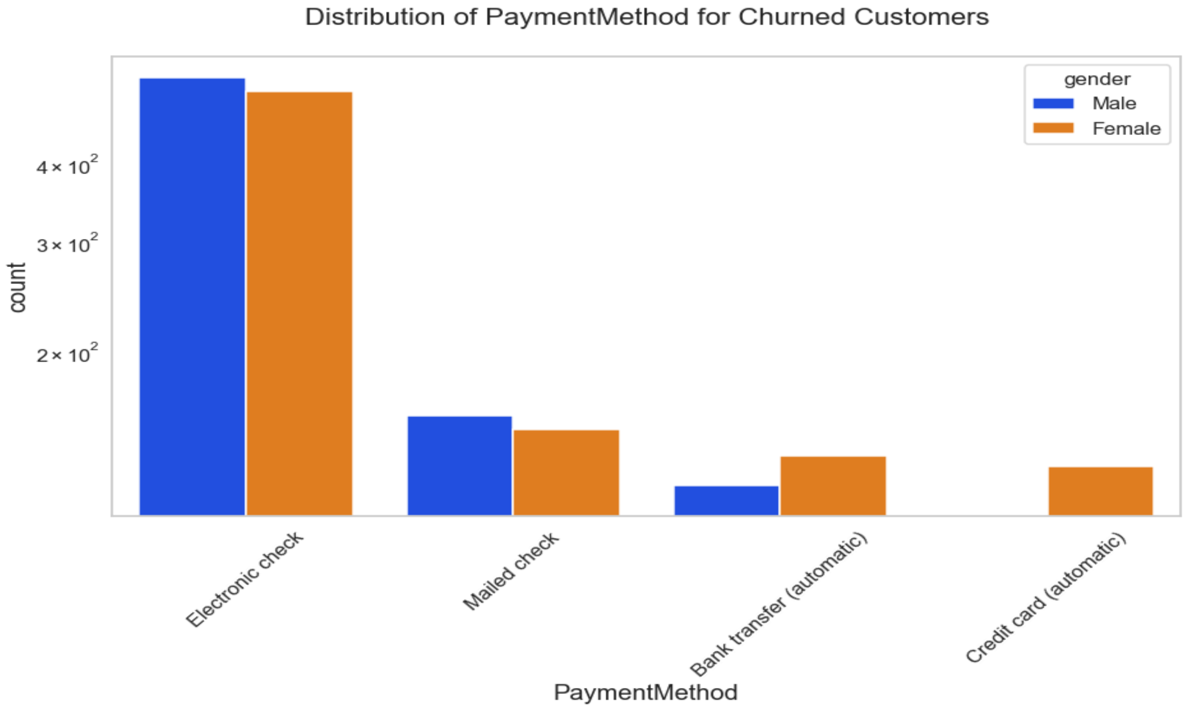
uniplot(new_df1_target0,col='Partner',title='Distribution of Gender for Non Churned Customers',hue='gender')

Output:



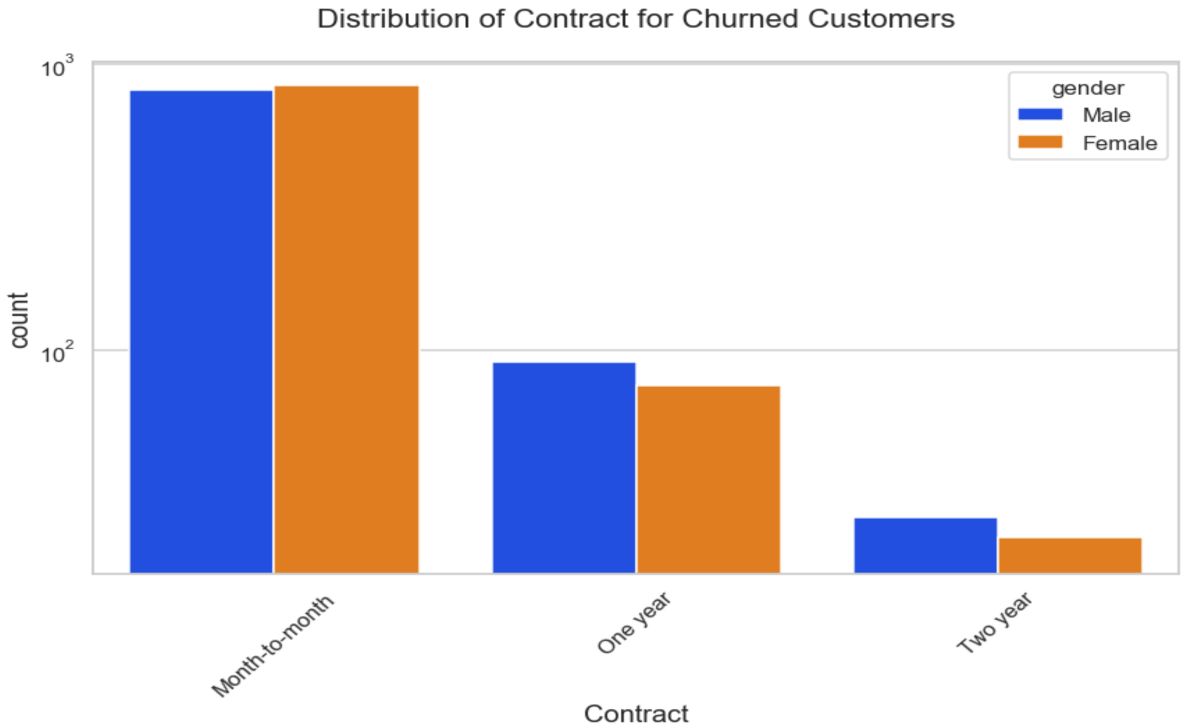
```
uniplot(new_df1_target1,col='PaymentMethod',title='Distribution of PaymentMethod for Churned Customers',hue='gender')
```

Output:



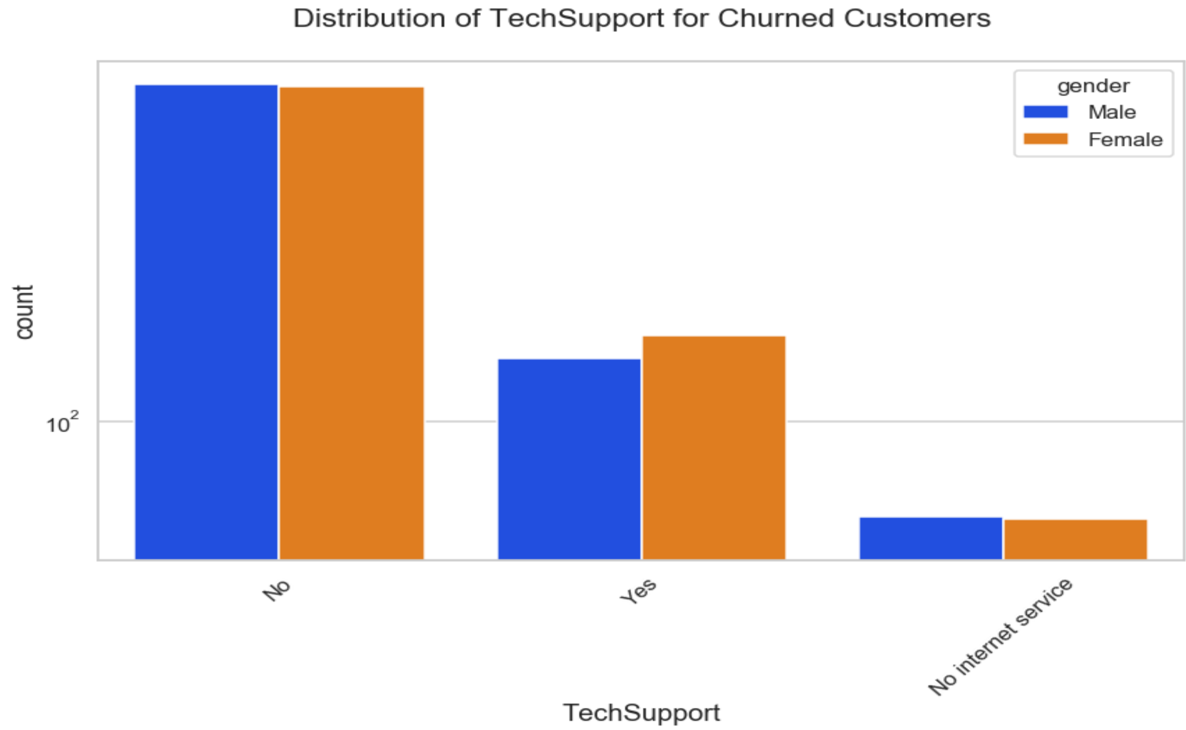

```
uniplot(new_df1_target1,col='Contract',title='Distribution of Contract for Churned Customers',hue='gender')
```

Output:



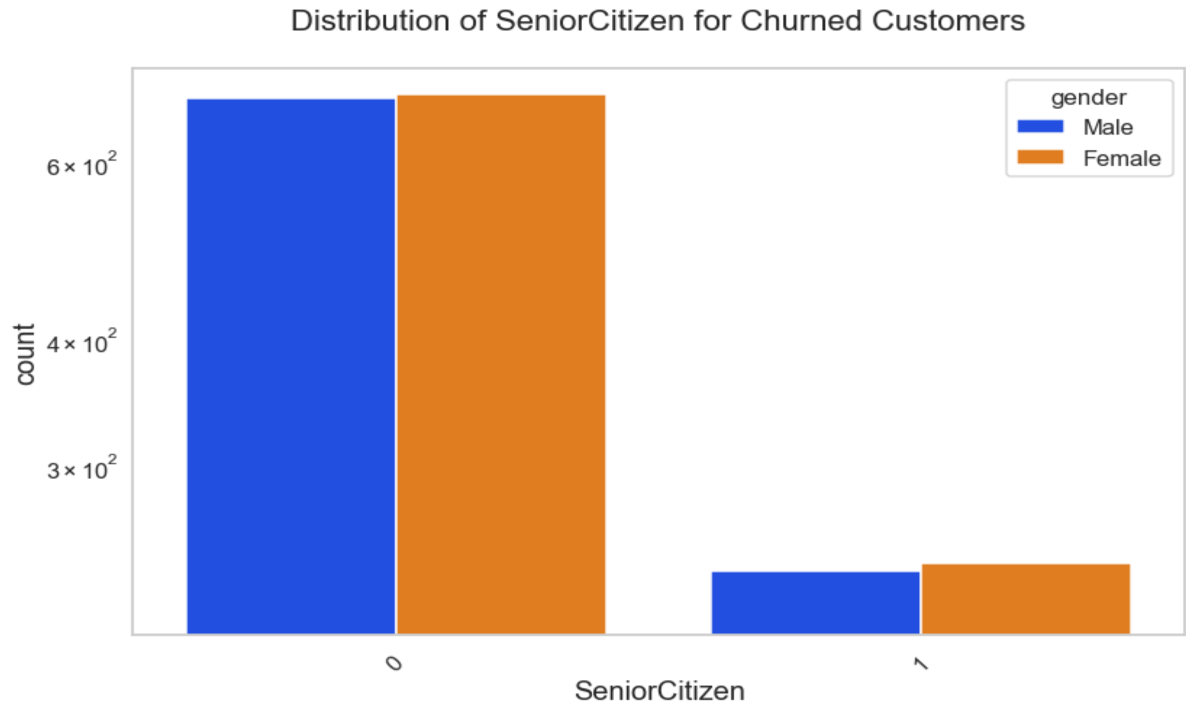
```
uniplot(new_df1_target1,col='TechSupport',title='Distribution of TechSupport for Churned Customers',hue='gender')
```

Output:




```
uniplot(new_df1_target1,col='SeniorCitizen',title='Distribution of SeniorCitizen for Churned Customers',hue='gender')
```

Output:



Derived Metrics Feature Binning :

Derived metrics create a new variable from the existing variable to get a insightful information from the data by analysing the data.

- Feature Binning
- Feature Encoding
- From Domain Knowledge
- Calculated from Data

Conclusion:

These are some of the quick insights from this exercise:

1. Electronic check medium are the highest churners.

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 churners.

4. Non senior Citizens are high churners.

*****