A MACHINE LEARNING APPROACH TO IDENTIFY - TELECOM CUSTOMER CHURN AND RECOMMENTING RETENTION STRATEGY

PROBLEM STATEMENT

This project is to predict the customer who will move out from the company in future and recommend appropriate retention tactics using IBM Sample Telecom Customer Churn dataset. This project involves complete analysis of the dataset, visualization generation, Machine Learning Models Training and Evaluation. Finally developing a web app using Fast API to identify churn customer and retention strategy with recommendations.

DATA COLLECTION

The dataset was collected from IBM Blog site. The name of this dataset is “IBM Sample Telco Churn” which has 7047 rows with 33 Columns. The dataset has various information about customer Personal Details, Charges and Service activations.

Import Required Packages

In [3]:

```
# importing libraries

#A data manipulation library for working with tabular data
import pandas as pd

#A numerical computing library for working with arrays and matrices
import numpy as np

#visualisation libraries

#A data visualization library for creating statistical graphics
import seaborn as sns

#A data visualization library for creating static, interactive, and animated visualizatio
import matplotlib.pyplot as plt

#A high-level data visualization library for creating interactive plots and charts
import plotly.express as px

#preprocessing libraries

#A library for performing data preprocessing tasks such as scaling, encoding, and imputin
from sklearn.preprocessing import StandardScaler

#A library to perform model selection tasks such as cross-validation and hyperparameter t
from sklearn.model_selection import cross_val_score, train_test_split
```

Read Dataset

In [4]:

```
#importing datset
dataset = pd.read_excel(r'Telco_customer_churn.xlsx', engine='openpyxl')
```

In [5]:

```
#display first 5 rows of dataset
dataset.head()
```

Out[5]:

	CustomerID	Count	Country	State	City	Zip Code	Lat Long	Latitude	Longitud
0	3668-QPYBK	1	United States	California	Los Angeles	90003	33.964131, -118.272783	33.964131	-118.27278
1	9237-HQITU	1	United States	California	Los Angeles	90005	34.059281, -118.30742	34.059281	-118.30742
2	9305-CDSKC	1	United States	California	Los Angeles	90006	34.048013, -118.293953	34.048013	-118.29395
3	7892-POOKP	1	United States	California	Los Angeles	90010	34.062125, -118.315709	34.062125	-118.31570
4	0280-XJGEX	1	United States	California	Los Angeles	90015	34.039224, -118.266293	34.039224	-118.26629

5 rows × 33 columns

EXPLORE THE DATASET

In [6]:

```
#analysing Target feature Count
dataset["Churn Label"].value_counts()
```

Out[6]:

```
No     5174
Yes    1869
Name: Churn Label, dtype: int64
```

This above result "No - 5174" says that 5174 Customer didn't Churn from the company and " yes - 1869" means 1869 Customer are Churned.

In [7]:

```
#display the total no of rows & columns in dataset
dataset.shape
```

Out[7]:

```
(7043, 33)
```

In [8]:

```
#Check for Duplicates
dataset.duplicated().sum()
```

Out[8]:

```
0
```

No dulpicates found in the dataset as result is 0.

In [9]:

```
#Display summary statistics for a dataset
dataset.describe()
```

Out[9]:

	Count	Zip Code	Latitude	Longitude	Tenure Months	Monthly Charges	Churn Valu
count	7043.0	7043.000000	7043.000000	7043.000000	7043.000000	7043.000000	7043.00000
mean	1.0	93521.964646	36.282441	-119.798880	32.371149	64.761692	0.26537
std	0.0	1865.794555	2.455723	2.157889	24.559481	30.090047	0.44156
min	1.0	90001.000000	32.555828	-124.301372	0.000000	18.250000	0.00000
25%	1.0	92102.000000	34.030915	-121.815412	9.000000	35.500000	0.00000
50%	1.0	93552.000000	36.391777	-119.730885	29.000000	70.350000	0.00000
75%	1.0	95351.000000	38.224869	-118.043237	55.000000	89.850000	1.00000
max	1.0	96161.000000	41.962127	-114.192901	72.000000	118.750000	1.00000

Statistical info from the dataset

Customer tenure is on average (mean) 32.37 months with a standard deviation of 24.55 months.

The average monthly cost is 64.76.

The minimum and maximum churn scores are respectively 100 and 5. With an average churn score of 58.69, almost 26% of the client base have given up.

Check Datatypes

In [10]:

```
#information about datasets with null values & Datatype
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 33 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   CustomerID         7043 non-null   object 
 1   Count              7043 non-null   int64  
 2   Country            7043 non-null   object 
 3   State              7043 non-null   object 
 4   City               7043 non-null   object 
 5   Zip Code           7043 non-null   int64  
 6   Lat Long           7043 non-null   object 
 7   Latitude           7043 non-null   float64
 8   Longitude          7043 non-null   float64
 9   Gender             7043 non-null   object 
 10  Senior Citizen     7043 non-null   object 
 11  Partner            7043 non-null   object 
 12  Dependents         7043 non-null   object 
 13  Tenure Months      7043 non-null   int64  
 14  Phone Service      7043 non-null   object 
 15  Multiple Lines     7043 non-null   object 
 16  Internet Service   7043 non-null   object 
 17  Online Security    7043 non-null   object 
 18  Online Backup      7043 non-null   object 
 19  Device Protection  7043 non-null   object 
 20  Tech Support       7043 non-null   object 
 21  Streaming TV       7043 non-null   object 
 22  Streaming Movies   7043 non-null   object 
 23  Contract           7043 non-null   object 
 24  Paperless Billing  7043 non-null   object 
 25  Payment Method     7043 non-null   object 
 26  Monthly Charges    7043 non-null   float64
 27  Total Charges      7043 non-null   object 
 28  Churn Label        7043 non-null   object 
 29  Churn Value        7043 non-null   int64  
 30  Churn Score        7043 non-null   int64  
 31  CLTV               7043 non-null   int64  
 32  Churn Reason       1869 non-null   object 
dtypes: float64(3), int64(6), object(24)
memory usage: 1.8+ MB
```

The result of above code shows, no missing values found in any other columns except “Churn Reason” Column which is having only 1869 rows. The “DType” represents the datatype of each Columns.

Checking for Missing values

In [11]:

```
(dataset.isnull().mean().sort_values(ascending=False)[0:33])*100
```

Out[11]:

```
Churn Reason         73.463013
Online Security       0.000000
CLTV                  0.000000
Churn Score           0.000000
Churn Value           0.000000
Churn Label           0.000000
Total Charges         0.000000
Monthly Charges       0.000000
Payment Method        0.000000
Paperless Billing     0.000000
Contract              0.000000
Streaming Movies      0.000000
Streaming TV          0.000000
Tech Support          0.000000
Device Protection     0.000000
Online Backup         0.000000
CustomerID            0.000000
Count                 0.000000
Multiple Lines        0.000000
Phone Service         0.000000
Tenure Months         0.000000
Dependents            0.000000
Partner               0.000000
Senior Citizen        0.000000
Gender                0.000000
Longitude             0.000000
Latitude              0.000000
Lat Long              0.000000
Zip Code              0.000000
City                  0.000000
State                 0.000000
Country               0.000000
Internet Service      0.000000
dtype: float64
```

The results show that “Churn Reason” Column has many missing values almost 73.46 % and No missing values found in all other Columns. So, this “Churn Reason” Column can be deleted during the EDA.

Checking Unique values

In [12]:

```
dataset.CustomerID.nunique()
```

Out[12]:

```
7043
```

This CustomerID Columns has 7043 unique Records. It is not useful for any analysis so we can remove it.

Exploring Categorical Features

In [13]:

```
dataset.describe(include=object).T
```

Out[13]:

	count	unique	top	freq
CustomerID	7043	7043	3668-QPYBK	1
Country	7043	1	United States	7043
State	7043	1	California	7043
City	7043	1129	Los Angeles	305
Lat Long	7043	1652	33.964131, -118.272783	5
Gender	7043	2	Male	3555
Senior Citizen	7043	2	No	5901
Partner	7043	2	No	3641
Dependents	7043	2	No	5416
Phone Service	7043	2	Yes	6361
Multiple Lines	7043	3	No	3390
Internet Service	7043	3	Fiber optic	3096
Online Security	7043	3	No	3498
Online Backup	7043	3	No	3088
Device Protection	7043	3	No	3095
Tech Support	7043	3	No	3473
Streaming TV	7043	3	No	2810
Streaming Movies	7043	3	No	2785
Contract	7043	3	Month-to-month	3875
Paperless Billing	7043	2	Yes	4171
Payment Method	7043	4	Electronic check	2365
Total Charges	7043.0	6531.0	20.2	11.0
Churn Label	7043	2	No	5174
Churn Reason	1869	20	Attitude of support person	192

There are 24 Categorical Features found in the dataset. Each categorical features has 7043 records except "Churn Reason" Column as it has only 1869 rows. The Unique represents the unique data in each features and the top represent the frequently avaialble feature with the freq count.

Example, The Payment Method column has 7043 data which has 4 unique payment method and Electronic check payment method is followed by 2365 customer out of 7043 records.

DATA CLEANING

In [14]:

```
dataset = dataset.drop('Count', axis=1)
```

The "Count" Colum has only the count of the CustomerID , so it can removed.

Converting Total Charges Datatype and Filling Missing Values

In [15]:

```
dataset[~dataset['Total Charges'].str.contains('\d+\.?\d*',na=True)]
```

Out[15]:

	CustomerID	Country	State	City	Zip Code	Lat Long	Latitude	Longitu
2234	4472-LVYGI	United States	California	San Bernardino	92408	34.084909, -117.258107	34.084909	-117.2581
2438	3115-CZMZD	United States	California	Independence	93526	36.869584, -118.189241	36.869584	-118.1892
2568	5709-LVOEQ	United States	California	San Mateo	94401	37.590421, -122.306467	37.590421	-122.3064
2667	4367-NUYAO	United States	California	Cupertino	95014	37.306612, -122.080621	37.306612	-122.0806
2856	1371-DWPAZ	United States	California	Redcrest	95569	40.363446, -123.835041	40.363446	-123.8350
4331	7644-OMVMY	United States	California	Los Angeles	90029	34.089953, -118.294824	34.089953	-118.2948
4687	3213-VVOLG	United States	California	Sun City	92585	33.739412, -117.173334	33.739412	-117.1733
5104	2520-SGTTA	United States	California	Ben Lomond	95005	37.078873, -122.090386	37.078873	-122.0903
5719	2923-ARZLG	United States	California	La Verne	91750	34.144703, -117.770299	34.144703	-117.7702
6772	4075-WKNIU	United States	California	Bell	90201	33.970343, -118.171368	33.970343	-118.1713
6840	2775-SEFEE	United States	California	Wilmington	90744	33.782068, -118.262263	33.782068	-118.2622

11 rows × 32 columns

'\d+.?\d*' helps to filter the rows which doesn't have numerical values. So from the above output we can clearly see that there is no numeric value presents in 11 rows in "Total Charges" Column.

Filling Missing Values

In [16]:

```
# convert TotalCharges to float and fill missing values with mean
dataset['Total Charges'] = pd.to_numeric(dataset['Total Charges'], errors='coerce')
mean_total_charges = dataset['Total Charges'].mean()
dataset['Total Charges'].fillna(mean_total_charges, inplace=True)

# Verify any other missing data
print(dataset.isnull().sum())
```

```
CustomerID             0
Country                0
State                  0
City                   0
Zip Code               0
Lat Long               0
Latitude               0
Longitude              0
Gender                 0
Senior Citizen         0
Partner                0
Dependents             0
Tenure Months          0
Phone Service          0
Multiple Lines         0
Internet Service       0
Online Security        0
Online Backup          0
Device Protection      0
Tech Support           0
Streaming TV           0
Streaming Movies       0
Contract               0
Paperless Billing      0
Payment Method         0
Monthly Charges        0
Total Charges          0
Churn Label            0
Churn Value            0
Churn Score            0
CLTV                   0
Churn Reason        5174
dtype: int64
```

There are 11 missing rows in “Total Charges” Columns are filled by “Mean” Value.

Handling Duplicates

In [17]:

```
# Checking for duplicates
dataset.duplicated().sum()
```

Out[17]:

```
0
```

In [18]:

```
dataset.head()
```

Out[18]:

	CustomerID	Country	State	City	Zip Code	Lat Long	Latitude	Longitude	Gend
0	3668-QPYBK	United States	California	Los Angeles	90003	33.964131, -118.272783	33.964131	-118.272783	Ma
1	9237-HQITU	United States	California	Los Angeles	90005	34.059281, -118.30742	34.059281	-118.307420	Fema
2	9305-CDSKC	United States	California	Los Angeles	90006	34.048013, -118.293953	34.048013	-118.293953	Fema
3	7892-POOKP	United States	California	Los Angeles	90010	34.062125, -118.315709	34.062125	-118.315709	Fema
4	0280-XJGEX	United States	California	Los Angeles	90015	34.039224, -118.266293	34.039224	-118.266293	Ma

5 rows × 32 columns

EXPLORATORY DATA ANALYSIS

Exploring Numerical and Categorical Features in the Dataset

The following code to Check the list of Numerical and Categical Columns present in dataset by checking datatype of each columns.

In [19]:

```
# generate numerical & category columns
Telco_numeric_features = [colu for colu in dataset.columns if dataset[colu].dtype != 'O']
Telco_categorical_feature = [colu for colu in dataset.columns if dataset[colu].dtype == '

# print the columns
print('There are {} numerical features in the dataset : {}'.format(len(Telco_numeric_feat
print('\nThere are {} categorical features in the dataset : {}'.format(len(Telco_categori
```

```
There are 9 numerical features in the dataset : ['Zip Code', 'Latitude',
'Longitude', 'Tenure Months', 'Monthly Charges', 'Total Charges', 'Churn V
alue', 'Churn Score', 'CLTV']

There are 23 categorical features in the dataset : ['CustomerID', 'Countr
y', 'State', 'City', 'Lat Long', 'Gender', 'Senior Citizen', 'Partner', 'D
ependents', 'Phone Service', 'Multiple Lines', 'Internet Service', 'Online
Security', 'Online Backup', 'Device Protection', 'Tech Support', 'Streamin
g TV', 'Streaming Movies', 'Contract', 'Paperless Billing', 'Payment Metho
d', 'Churn Label', 'Churn Reason']
```

Checking Proportions of Categorical Data

In [20]:

```
# proportion of count data on categorical columns
for colu in Telco_categorical_feature:
    print(dataset[colu].value_counts(normalize=True) * 100)
    print('---------------------------')
```

```
3668-QPYBK    0.014198
9169-BSVIN    0.014198
0206-OYVOC    0.014198
6418-HNFED    0.014198
8805-JNRAZ    0.014198
              ...
6797-UCJHZ    0.014198
5016-IBERQ    0.014198
3003-CMDUU    0.014198
5148-HKFIR    0.014198
3186-AJIEK    0.014198
Name: CustomerID, Length: 7043, dtype: float64
---------------------------
United States    100.0
Name: Country, dtype: float64
---------------------------
California    100.0
Name: State, dtype: float64
---------------------------
```

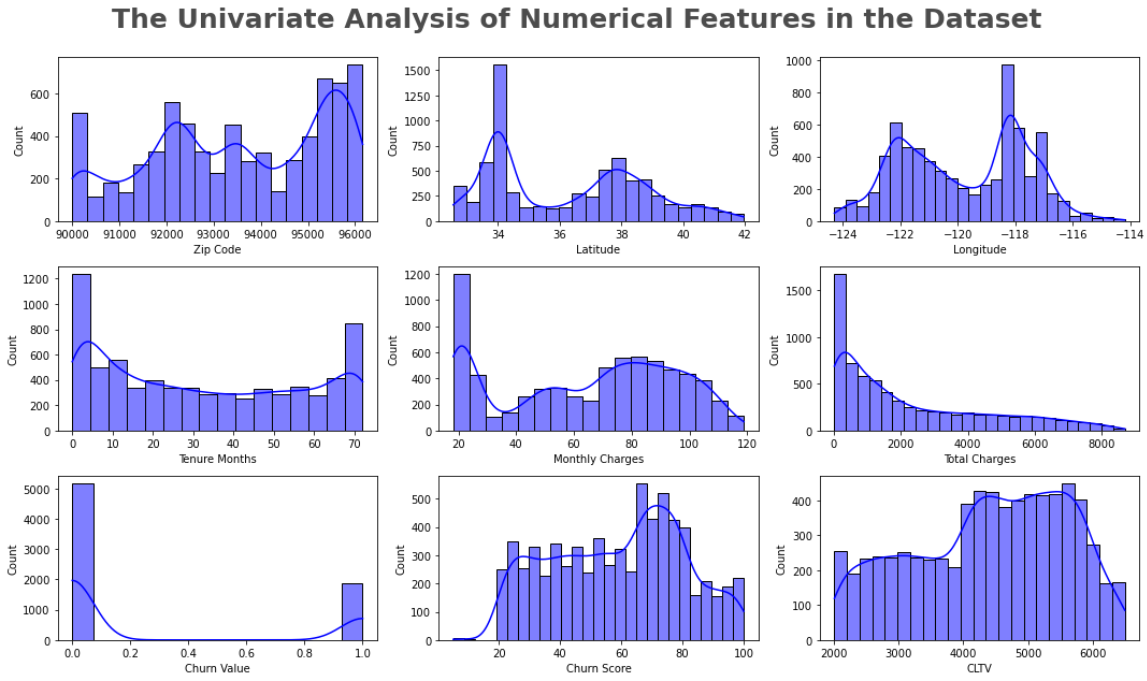
Univariate Analysis

Numerical Features

In Numerical Univariate Analysis, its easy to understand the distribution of each numerical features and identify the potential outliers or skewness.

In [21]:

```
plt.figure(figsize=(15, 30))
plt.suptitle('The Univariate Analysis of Numerical Features in the Dataset', fontsize=25,

for i in range(0, len(Telco_numeric_features)):
    plt.subplot(11, 3, i+1)
    sns.histplot(x=dataset[Telco_numeric_features[i]], color='blue',kde=True)
    plt.xlabel(Telco_numeric_features[i])
    plt.tight_layout()
```



INSIGHTS

This graph distribution of 'Total Charges' is positively skewed, indicating the presence of a small number of customers with much higher charges than the majority of customers.

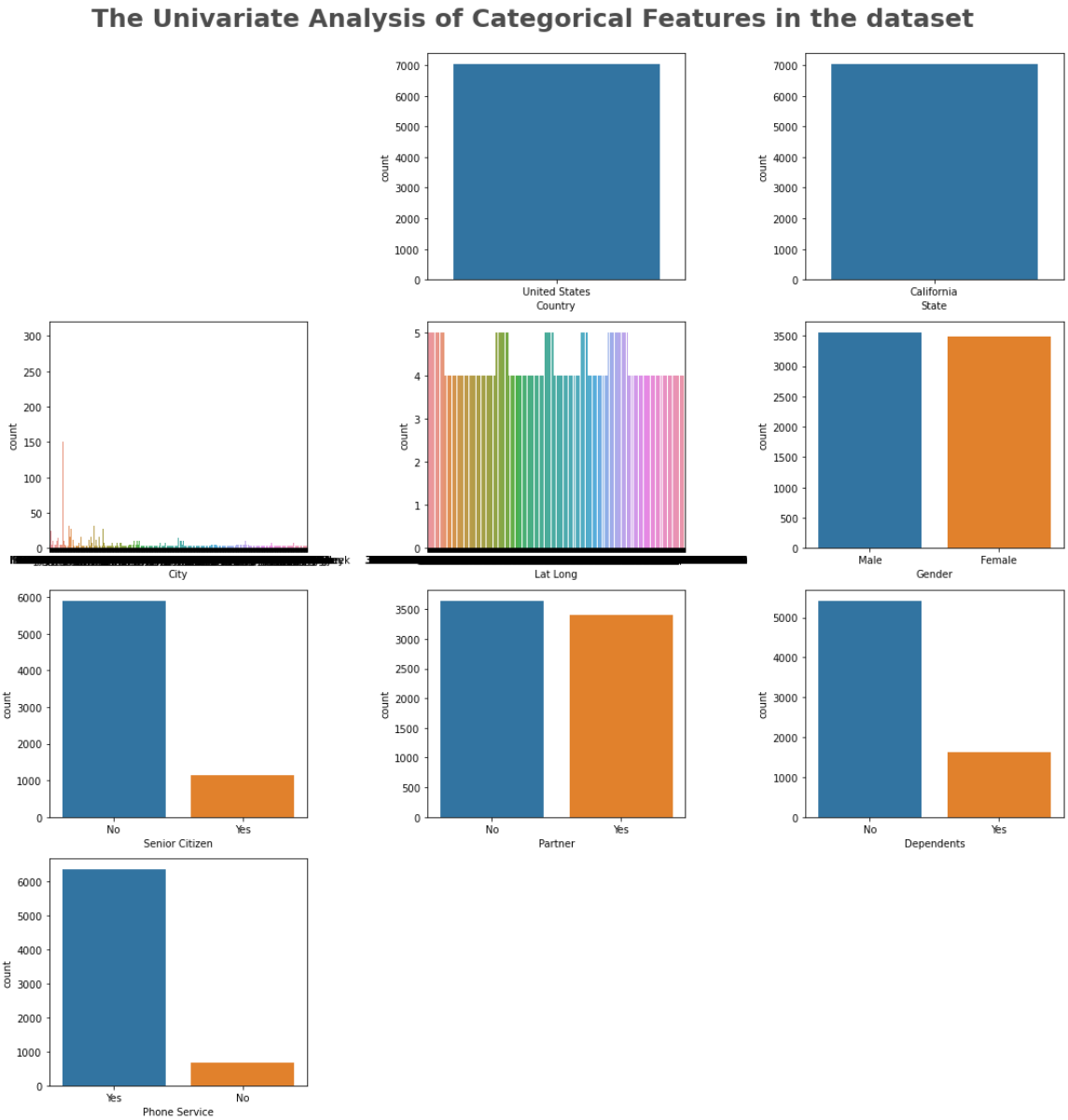
Monthly Charges have a roughly normal distribution, with a peak at around 70-80.

The distributions of Churn Score, Churn Value & CLTV are positively skewed, indicating the presence of some customers with very high values.

Categorical Features

In [22]:

```
#plot for categorical features
plt.figure(figsize=(15, 30))
plt.suptitle('The Univariate Analysis of Categorical Features in the dataset', fontsize=2

for i in range(1, 10):
    plt.subplot(8, 3, i+1)
    sns.countplot(x=dataset[Telco_categorical_feature[i]])
    plt.xlabel(Telco_categorical_feature[i])
    plt.tight_layout()
```



INSIGHTS:

Country: This figure displays the percentage of clients in the dataset from each country. The United States, India, and the United Kingdom account for most of the clientele.

States - This graph displays the percentage of customers in each state represented in the dataset. Customers primarily hail from California, Texas, and New York.

Gender - The proportion of male and female clients is depicted in this graph. Nearly equal numbers of male and female clients make up the dataset.

Senior Citizen - This graph illustrates the dataset's percentage of senior citizen's clients. The proportion of senior citizens customers in the dataset is low.

Partner - This graph displays the percentage of customers who have a partner. The proportion of customers without partners in the dataset is somewhat higher than that of customers with partners.

Dependents: The proportion of customers who have dependents is depicted in this graph. There are not many customers with dependents in the dataset.

Phone Service - This graph displays the percentage of customers who have phone service. Almost all the dataset's clients have access to phone service.

Multivariate Analysis

In [23]:

```
discrete_features=[column for column in Telco_numeric_features if (len(dataset[column].ur

continuous_features=[column for column in Telco_numeric_features if len(dataset[column].u

encoded_categorical = [column for column in Telco_numeric_features if len(dataset[column]

print('There are {} discrete features in this dataset : {}'.format(len(discrete_features)
print('\nThere are {} continuous_features in this dataset : {}'.format(len(continuous_fea
print('\nThere are {} encoded_categorical in this dataset : {}'.format(len(encoded_catego
```

```
There are 0 discrete features in this dataset : []

There are 8 continuous_features in this dataset : ['Zip Code', 'Latitude',
'Longitude', 'Tenure Months', 'Monthly Charges', 'Total Charges', 'Churn S
core', 'CLTV']

There are 1 encoded_categorical in this dataset : ['Churn Value']
```

The above-mentioned code is used to categorize the Numeric features into 3 types:

1. Discreate Features – Discreate Features are Numeric features with the small no of Unique values (range between 6 and 25)
2. Continuous Features – Continuous Features are Numeric Features with the large number of Unique values which is more than 25.
3. Encoded Categorical Features – Encoded Categorical features which was originally categorical but encoded using a numerical representation.

Check Multicollinearity in Numerical features

Checking for multicollinearity in numerical features is important because it helps identify highly correlated independent variables, which can affect the performance of the model. This can lead to overfitting or underfitting, and inaccurate predictions. Removing highly correlated variables can improve the performance of model & provide more reliable predictions.

In [24]:

```
dataset[(list(dataset[continuous_features])[1:])].corr()
```

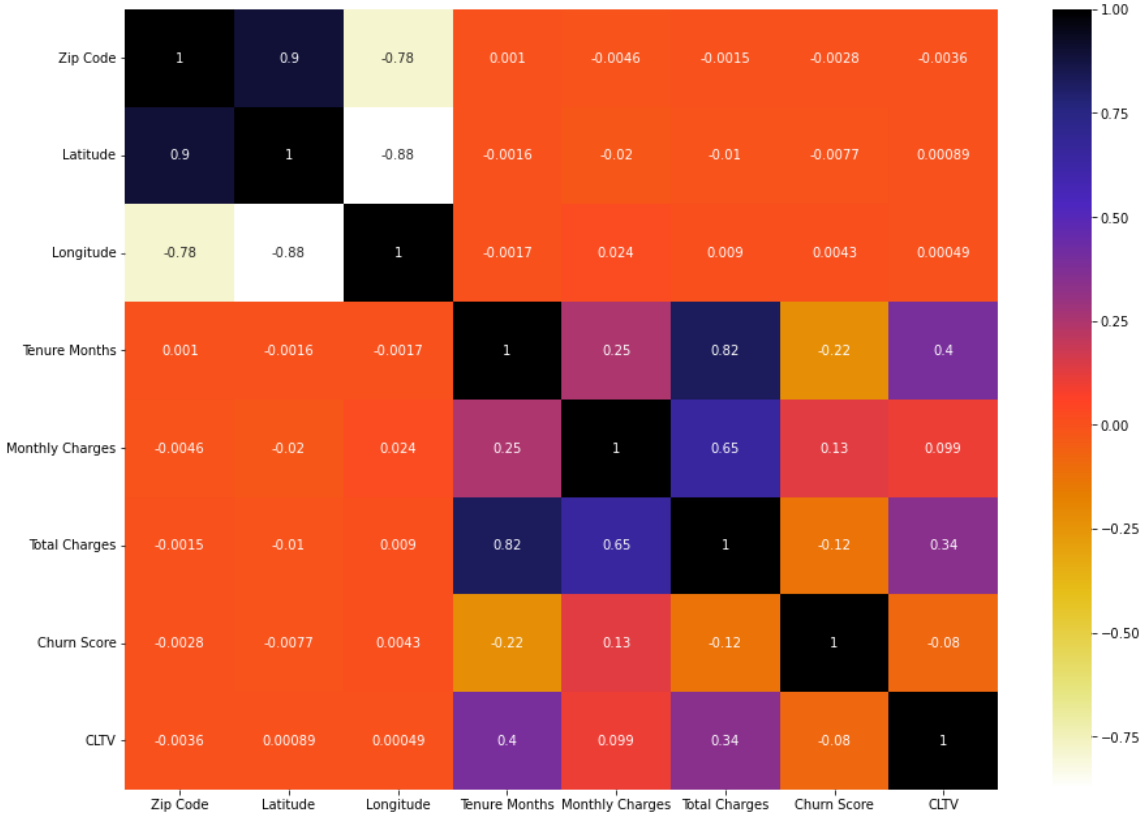
Out[24]:

	Latitude	Longitude	Tenure Months	Monthly Charges	Total Charges	Churn Score	CLTV
Latitude	1.000000	-0.876779	-0.001631	-0.019899	-0.010307	-0.007684	0.000886
Longitude	-0.876779	1.000000	-0.001678	0.024098	0.009039	0.004260	0.000485
Tenure Months	-0.001631	-0.001678	1.000000	0.247900	0.824757	-0.224987	0.396406
Monthly Charges	-0.019899	0.024098	0.247900	1.000000	0.650468	0.133754	0.098693
Total Charges	-0.010307	0.009039	0.824757	0.650468	1.000000	-0.124251	0.341384
Churn Score	-0.007684	0.004260	-0.224987	0.133754	-0.124251	1.000000	-0.079782
CLTV	0.000886	0.000485	0.396406	0.098693	0.341384	-0.079782	1.000000

The above Correlation matrix states that the pairwise correlation between the numerical feature in our dataset. The range of the values from -1 to 1. There 1 means Positive correlation and 0 means No correlation & – 1 means the perfect negative correlation. We can use this to identify if there is any Multicollinearity (High Correlation) between Numerical features in the dataset.

In [25]:

```
plt.figure(figsize = (15,11))
sns.heatmap(dataset[continuous_features].corr(), cmap="CMRmap_r", annot=True)
plt.show()
```



The above graph(Heatmap) shows that few variables are correlated highly with eachothers such as Total charges and Monthly Charges which indicated by the bright red colour in the Heatmap. This suggests that these variables are measuring similar aspects of the dataset and may introduce multicollinearity in the model. So, we should consider removing one of the these variables to improve performance of the model.

Check Multicollinearity in Categorical feature

Check if the Categorical features are corelated with target feature

The chi-square statistic is one method for demonstrating a connection between two categorical data. In this section, we examine the relationship between categorical columns and the target column, i.e., Churn Value/Label.

The Feature is independent of the target column, according to the Null Hypothesis (). (No-Correlation)

Alternate Hypothesis (): The columns Feature and Target are not independent.

In [26]:

```
from scipy.stats import chi2_contingency
chi2_test = []
for feature in Telco_categorical_feature:
    if chi2_contingency(pd.crosstab(dataset['Churn Value'], dataset[feature]))[1] < 0.05:
        chi2_test.append('Reject Null Hypothesis')
    else:
        chi2_test.append('Fail to Reject Null Hypothesis')
result = pd.DataFrame(data=[Telco_categorical_feature, chi2_test]).T
result.columns = ['Column', 'Hypothesis Result']
result
```

Out[26]:

	Column	Hypothesis Result
0	CustomerID	Fail to Reject Null Hypothesis
1	Country	Fail to Reject Null Hypothesis
2	State	Fail to Reject Null Hypothesis
3	City	Reject Null Hypothesis
4	Lat Long	Fail to Reject Null Hypothesis
5	Gender	Fail to Reject Null Hypothesis
6	Senior Citizen	Reject Null Hypothesis
7	Partner	Reject Null Hypothesis
8	Dependents	Reject Null Hypothesis
9	Phone Service	Fail to Reject Null Hypothesis
10	Multiple Lines	Reject Null Hypothesis
11	Internet Service	Reject Null Hypothesis
12	Online Security	Reject Null Hypothesis
13	Online Backup	Reject Null Hypothesis
14	Device Protection	Reject Null Hypothesis
15	Tech Support	Reject Null Hypothesis
16	Streaming TV	Reject Null Hypothesis
17	Streaming Movies	Reject Null Hypothesis
18	Contract	Reject Null Hypothesis
19	Paperless Billing	Reject Null Hypothesis
20	Payment Method	Reject Null Hypothesis
21	Churn Label	Reject Null Hypothesis
22	Churn Reason	Fail to Reject Null Hypothesis

"Fail to reject null hypothesis" represents there is no enough proof to conclude that the categorical features are correlated with the target feature. "Reject null hypothesis" means that there is sufficient evidence to decide that categorical features are correlated with Target Feature.

The results show that City, Senior Citizen, Online Backup, Partner, Multiple Lines, Device Protection, Internet Service, Dependents, Online Security, Streaming Movies, Tech Support, Contract, Payment Method, Paperless Billing, Streaming TV and Churn Label are significantly correlated with Churn Value.

Outlier Analysis

In [27]:

```
dataset=dataset.drop(labels=['CustomerID'],axis=1)
```

The CustomerID column doesn't have any useful information for the analysis as its just has unique ID for each Customer. So, we can drop this Column to avoid unnecessary computations and to reduce the complexity of the dataset.

In [28]:

```
num_data=dataset.select_dtypes(include=['float64','int64']).columns
```

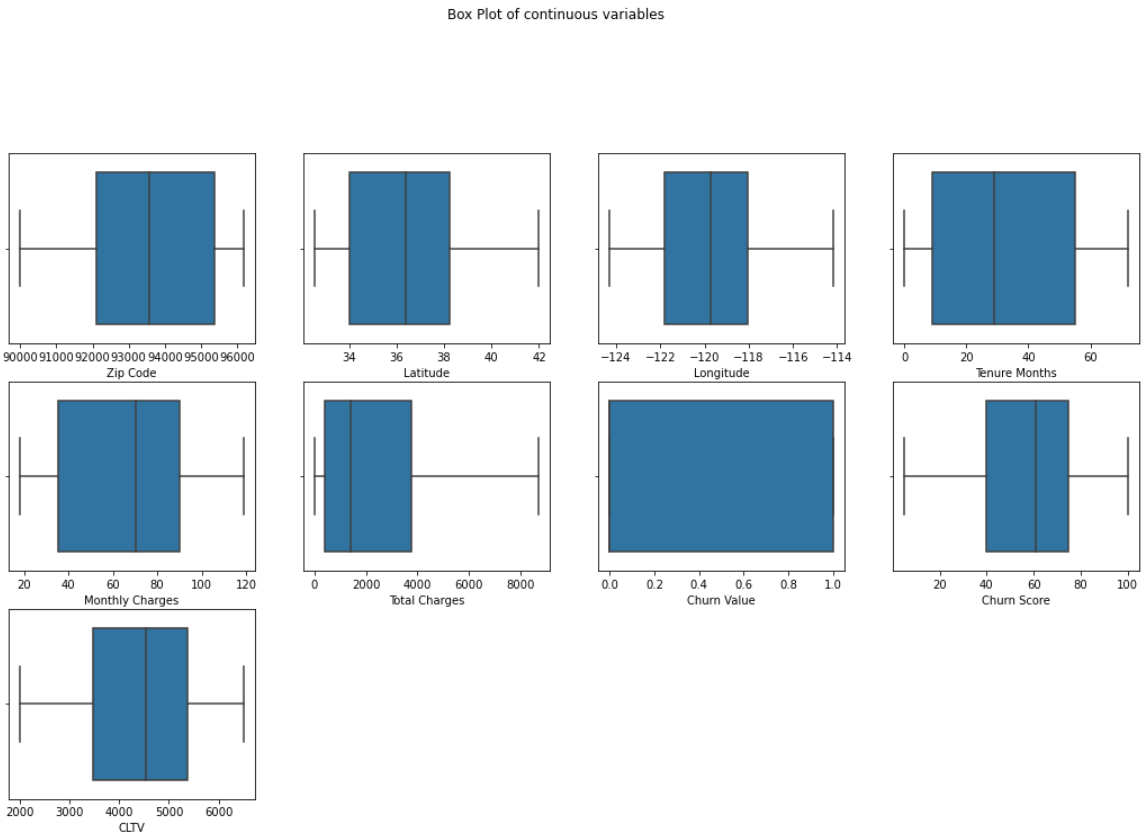
The code above assigns the names of the columns to the variable "Num_data" and selects either the float64 or int64 Datatype Columns.

In [29]:

```
plt.figure(figsize=(18, 18))
for i, colu in enumerate(num_data):
    plt.rcParams['axes.facecolor'] = 'White'
    ax = plt.subplot(5,4, i+1)
    sns.boxplot(data=dataset, x=colu)
plt.suptitle('Box Plot of continuous variables')
```

Out[29]:

```
Text(0.5, 0.98, 'Box Plot of continuous variables')
```



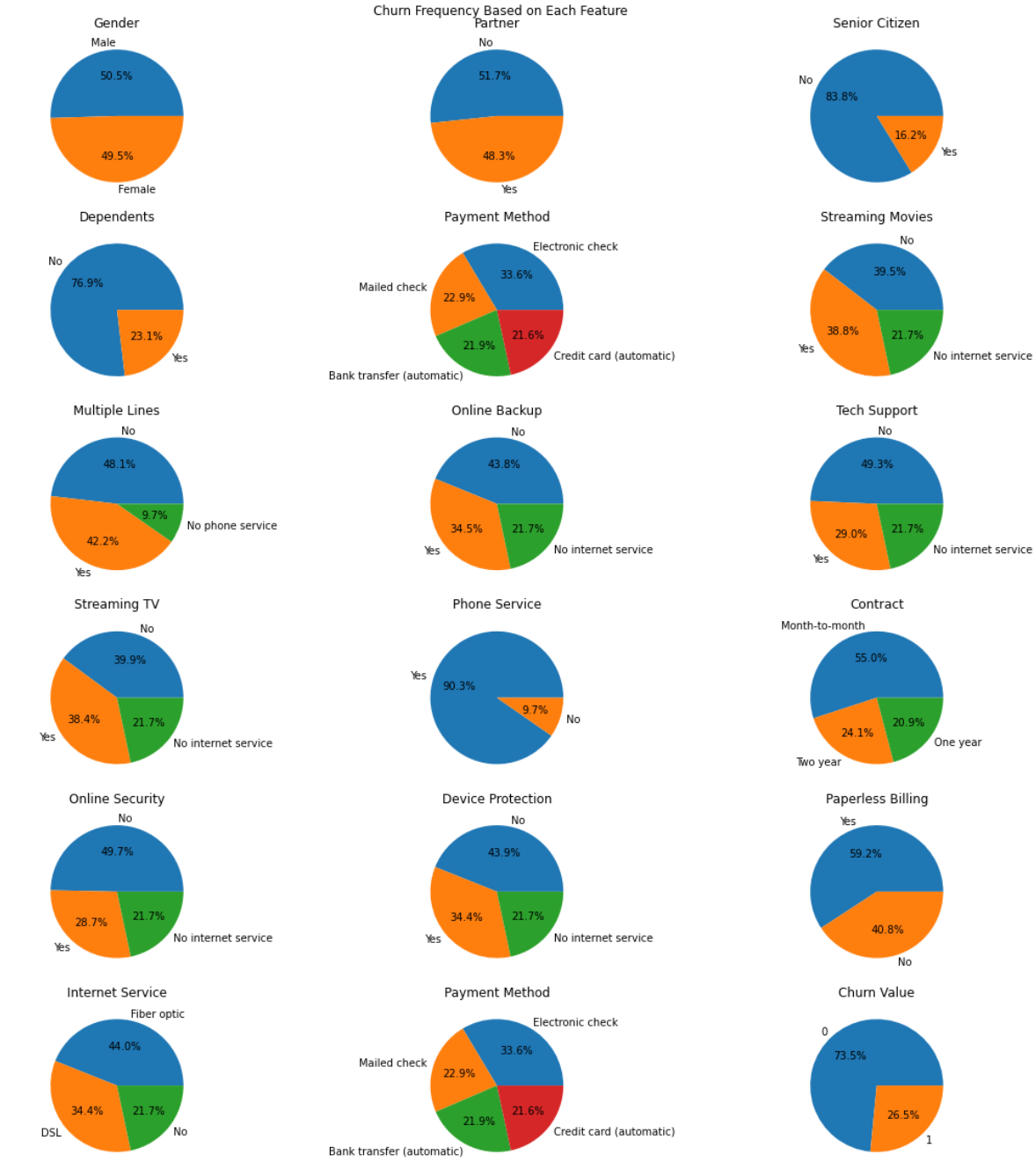
The box plots show the distribution and potential outliers in the continuous variables of the dataset. From the plots, it is evident that variables such as tenure months, monthly charges, and total charges have no outliers. It is important to deal outliers as they can skew the data and potentially affect the results of the analysis.

VISUALISATION

Exploring Churn Frequency in Customer Data

In [30]:

```
# create data for the pie charts
features = ['Gender', 'Partner' , 'Senior Citizen', 'Dependents','Payment Method', 'Strea
            'Online Backup', 'Tech Support', 'Streaming TV', 'Phone Service', 'Contract',
            'Device Protection', 'Paperless Billing', 'Internet Service', 'Payment Method

# create a figure with 5 rows and 4 columns of subplots
fig, axs = plt.subplots(6, 3, figsize=(16, 16))

# loop through each feature and create a pie chart for each subplot
for i, column in enumerate(features):
    rows = i // 3
    colu = i % 3
    labels = dataset[column].value_counts().index.tolist()
    sizes = dataset[column].value_counts().tolist()
    axs[rows, colu].pie(sizes, labels=labels, autopct='%1.1f%%')
    axs[rows, colu].set_title(column)

# set the title of the figure
fig.suptitle('Churn Frequency Based on Each Feature')

# adjust the layout & space
plt.tight_layout()

# print the plot
plt.show()
```



INSIGHTS

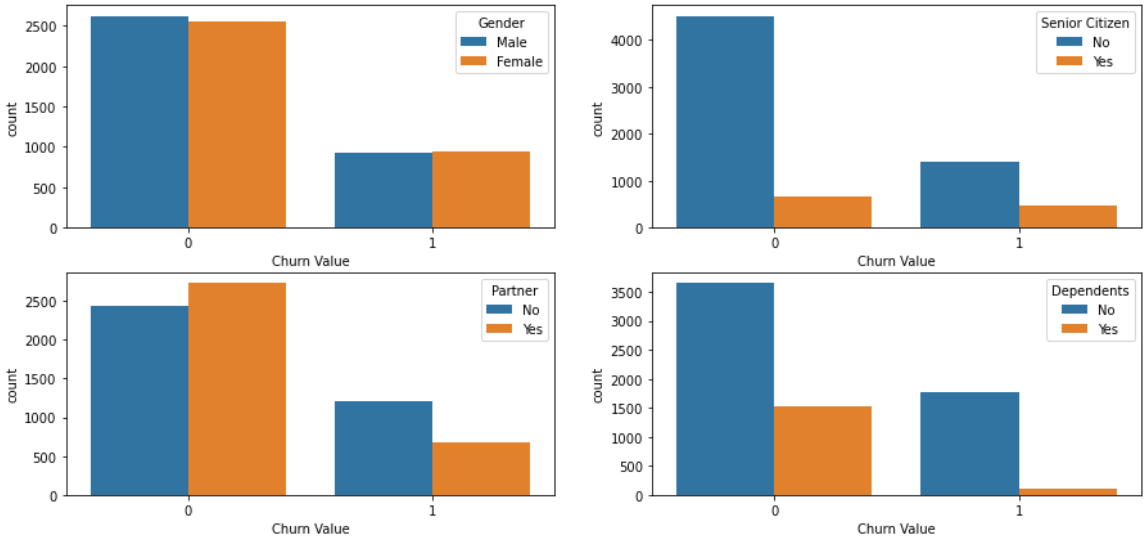
1. From the pie chart, we can see that around 26.5% of customers have churned, while 73.5% have not churned. This implies that the dataset is imbalanced, with a majority of customers not churning. This information is essential when we train our churn prediction model, as we need to make sure the model is not biased towards the majority of class. We may required to use the techniques such as oversampling/undersampling to balance the dataset before training the model.
2. This pie chart displays the % of senior citizens in the data set. It can be seen that the majority of the customers in the dataset are not senior citizens, accounting for approximately 83.8% of the total customers.Meanwhile, only about 16.2% of the customers are senior citizens. This insight is useful for targeting specific marketing campaigns to certain age groups or for offering specific services or discounts to senior citizens.
3. The pie chart shows the percentage of customers who have a partner or not. About 51.7% of the customers do not have a partner, while 48.3% of them have a partner.The plot shows the distribution of customer churn based on the partner status. Customers with partners are less likely to churn as compared to those without partners.

4. visualization shows the percentage of customers with and without dependents. About 70% of customers do not have dependents, while the remaining 30% have dependents.
5. The pie chart shows the distribution of payment methods used by customers. The majority of customers (33.6%) use electronic check as their payment method followed by mailed check is 22.9%, bank transfer is 21.9% and credit card is 21.6%. This information can help the company make decisions regarding their payment options and potentially target promotions or discounts towards customers using less popular payment methods to encourage them to switch.

churn status for each of the customer demographics

In [31]:

```
# Bar chart of customer demographics
demographics = ['Gender', 'Senior Citizen', 'Partner', 'Dependents']
plt.figure(figsize=(15, 7))
for i, colu in enumerate(demographics):
    plt.subplot(2, 2, i+1)
    sns.countplot(data=dataset, x='Churn Value', hue=colu)

# Show the plots
plt.show()
```



INSIGHTS

1. First visualization chart shows the count of churners and non-churners based on gender. It can be seen that the no of male & female churners is almost same, while the no of non-churners are slightly higher for males.
2. It seems like non-senior citizens are more likely to churn compare to senior citizens. This is evident from the count plot as there are more instances of churn in the non-senior citizen group than in the senior citizen group.
3. Among customers without partners, the number of churners is higher than non-churners. Among customers with partners, the number of non-churners is higher than churners. Therefore, having a partner seems to have an impact on customer churn.
4. The countplot with hue set to "Dependents" shows the distribution of churned and non-churned customers based on whether or not they have dependents. From the plot, we can see the customers without dependents have a higher probability of churning compared to those with dependents.

In [32]:

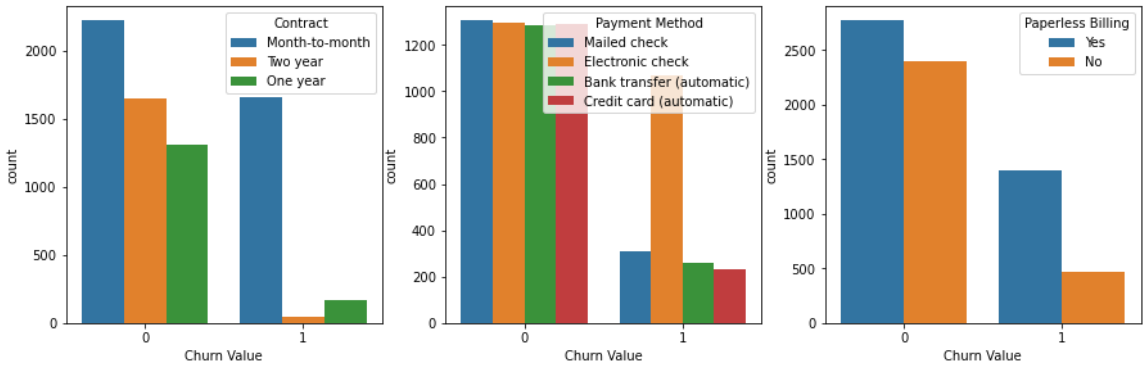
```
print(f'A female customer has a probability of {round(dataset[(dataset["Gender"] == "Fema

print(f'A male customer has a probability of {round(dataset[(dataset["Gender"] == "Male")
```

```
A female customer has a probability of 26.92 % churn
A male customer has a probability of 26.16 % churn
```

churn status for each of the customer Account details

In [33]:

```
# Bar chart of customer Account Details
AccountDetails = ['Contract', 'Payment Method', 'Paperless Billing']
plt.figure(figsize=(15, 10))
for i, colu in enumerate(AccountDetails):
    plt.subplot(2, 3, i+1)
    sns.countplot(data=dataset, x='Churn Value', hue=colu)

# Show the plots
plt.show()
```



INSIGHTS

1. First plot shows the count of churned and non-churned customers based on their contract type. Customers who having a monthly contract are seems more likely to churn, while those with long-term contracts are less seems to churn. This suggests that customers may prefer stability and predictability in their contracts, and that long-term contracts may be more effective in retaining customers.
2. Second plot shows the count of churned and non-churned customers based on their payment method. We can see that customers who pay through electronic check has a higher attrition percentage when compare to other payment method. Meanwhile, customers who are all pay through mailed check has the lowest churn rate. This suggests that payment method could be an important factor in predicting customer churn.
3. The customer who do paperless billing are more interested to Churn than Customer who don't follow paperless Billing

In [34]:

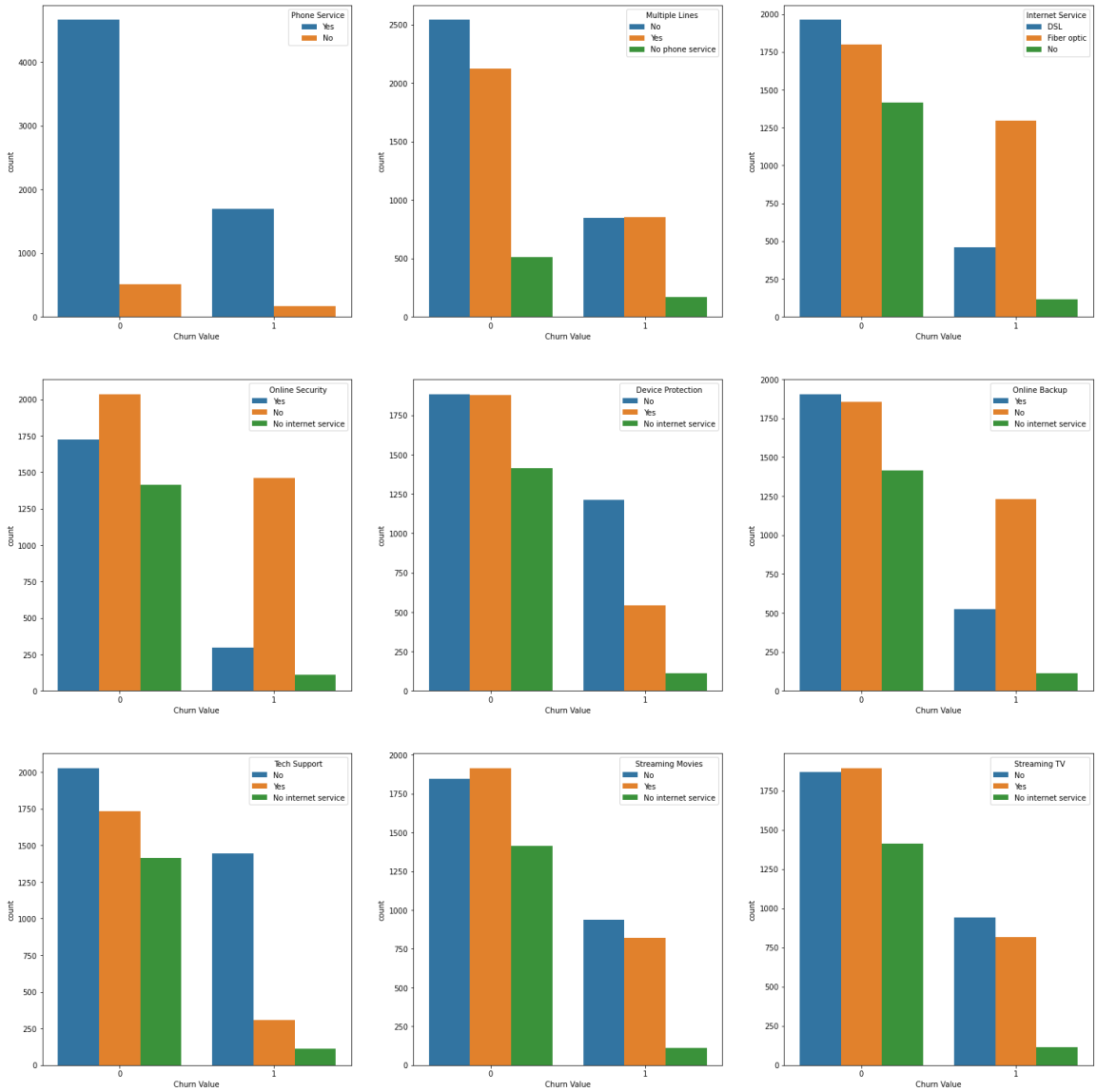
```
print(f'A customer with month-to-month contract has a probability of {round(dataset[(data

print(f'A customer with one year contract has a probability of {round(dataset[(dataset["C

print(f'A customer with two year contract has a probability of {round(dataset[(dataset["C
```

```
A customer with month-to-month contract has a probability of 42.71 % churn
A customer with one year contract has a probability of 11.27 % churn
A customer with two year contract has a probability of 2.83 % churn
```

churn status for each of the customer Service Usage

In [35]:

```
# Bar chart of customer Service Usage
ServiceUsage = ['Phone Service', 'Multiple Lines', 'Internet Service', 'Online Security',
plt.figure(figsize=(25, 35))
for i, colu in enumerate(ServiceUsage):
    plt.subplot(4, 3, i+1)
    sns.countplot(data=dataset, x='Churn Value', hue=colu)

# Show the plots
plt.show()
```



INSIGHTS

1. From the countplot, we can see that among customers who have phone service, those who churned and those who did not churn are almost evenly distributed. However, among customers who do not have phone service, a higher proportion churned compared to those who did not churn.
2. Customer who has no multiple lines are not churning

3. The 3rd Plot seems that customers who has 'fiber optic' services are seems churns more compare to those with 'no internet' or 'DSL' service. This may indicate the quality or pricing of the 'fiber optic' service is might not meeting the expectations of these customers.
4. The 4th, 5th and 6th plot shows the customer who don't have Online Secuity, Device Protection and Online Backup are churning the most.
5. The 7th, 8th and 9th Plot show that the customer who don't subscribed to the services such as Tech

Visualization Based on Total Charges

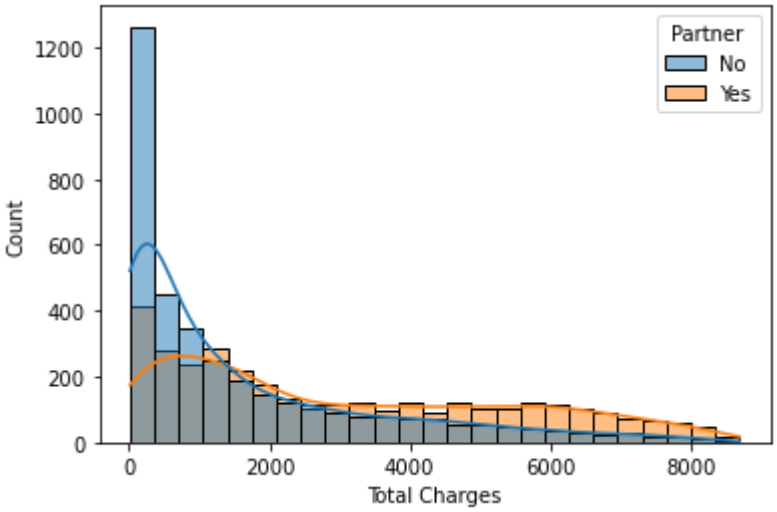
Total Charges by Partner status

In [36]:

```
sns.histplot(data = dataset , x='Total Charges' , hue = 'Partner' , kde = True)
```

Out[36]:

```
<AxesSubplot:xlabel='Total Charges', ylabel='Count'>
```



The above histogram plot shows the distribution of Total Charges based on whether the customer has a partner or not.

We can observe that the distribution of Total Charges is right-skewed, indicating that most of the customers have lower Total Charges.

Also we can observe that the distribution is wider for customers without a partner than those with a partner.

The plot also shows that customers with a partner tend to have slightly higher Total Charges than those without a partner, especially in the range of 0 to 5000.

In [37]:

```
dataset.groupby('Partner')['Total Charges'].agg(['mean','std','median'])
```

Out[37]:

Partner	mean	std	median
No	1585.344027	1874.345006	812.50
Yes	3030.290242	2404.734975	2341.85

The customers who have “Partner” have the average of higher Total Charges (3030.29) compared to the customer who don’t have the Partners (1585.39 – Average Total Charges). Meanwhile, the customer with No partner has a Standard Deviation of 1874.34 when compared to those who has Partner (2404.73). The median of total charges for Customer with Partner is 2341.85 and for Customer who has No partner is 812.50.

Total Charges in Relation to City

In [38]:

```
dataset.groupby('City')['Total Charges'].sum().sort_values(ascending = False)
```

Out[38]:

```
City
Los Angeles      650034.550441
San Diego        354896.600000
Sacramento       256295.050000
San Jose         243735.550000
San Francisco    221624.650000
                        ...
Homeland            668.250000
Dana Point          556.800000
Loleta              484.650000
Truckee             479.350000
Eagleville          203.400000
Name: Total Charges, Length: 1129, dtype: float64
```

This code groups the data by the city and calculates the total charges for each city. It then sorts the cities in descending order of total charges, with Los Angeles having the highest total charges and Eagleville having the lowest.

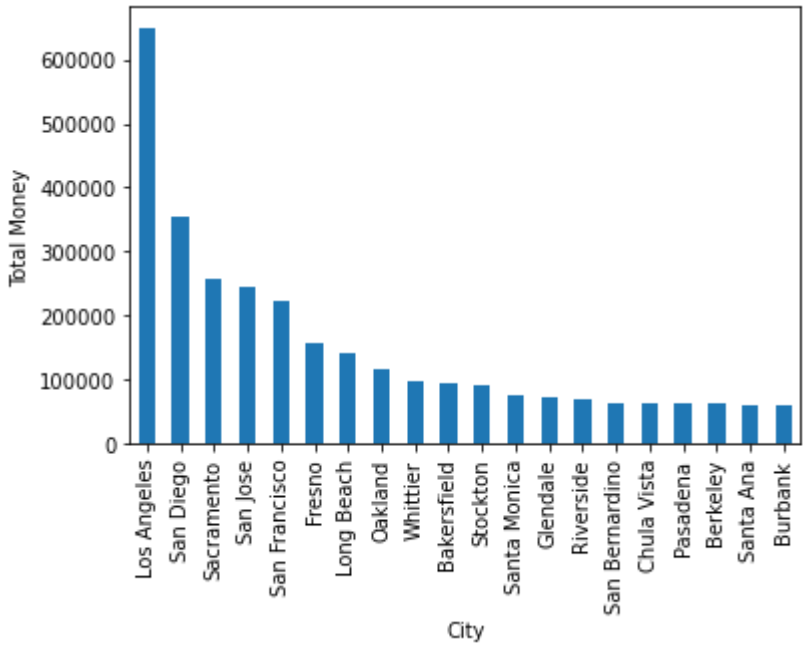
Top 20 cities by Total Charges in the dataset

In [39]:

```
dataset.groupby('City')['Total Charges'].sum().sort_values(ascending = False)[:20].plot(k
```

Out[39]:

```
<AxesSubplot:xlabel='City', ylabel='Total Money'>
```



The above code groups the data by cities and calculates the total charges for each city. It then sorts the cities in descending order of total charges and displays the top 20 cities with the highest total charges in a bar chart. This gives an idea of the cities that contribute the most to the company's revenue.

Recommendation - The telecommunications business should provide incentives or discounts to the most effective cities in order to retain them, as they generate significant revenue for the corporation.

Analysis of Total Charges by Gender

In [40]:

```
dataset.groupby('Gender')['Total Charges'].agg(['mean','median','std'])
```

Out[40]:

Gender	mean	median	std
Female	2283.191142	1389.05	2269.201601
Male	2283.407680	1406.65	2261.189740

The above result shows that the mean and Median of the Total Charges are quite similar for both genders. But slightly higher for “males” Gender. The standard deviation also quite high which indicating the large variation in the “Total Charges” for both Genders. There is no need to focus on churns based on genders.