Over view Dataset

"Predict behavior to retain customers. You can analyze all relevant customer data and develop focused customer retention programs." [IBM Sample Data Sets] https://www.kaggle.com/blastchar/telco-customer-churn Each row represents a customer, each column contains customer's attributes described on the column Metadata. The raw data contains 7043 rows (customers) and 21 columns (features). The "Churn" column is our target.

Research Questions:

1. Bagi yang berhenti berlangganan layanan, biasanya berapa lama mereka bertahan dalam layanan? dan berapa rata-rata LTV (Life Time Value) mereka?
2. Terkait dengan Pertanyaan 1, jenis layanan apa yang mereka berlangganan ketika mereka masih berlangganan?
3. Bagi mereka yang masih membayar layanan ini, berapa rata-rata LTV (Life Time Value) mereka? dan berapa lama biasanya mereka berada dalam dinas?
4. Berkaitan dengan Pertanyaan 3, berapa proporsi masing-masing jenis layanan yang mereka bayarkan?
5. Bagi yang masih dalam layanan dan memiliki LTV lebih besar dari LTV pelanggan yang bocor, layanan mana yang paling mahal?
6. Bagi yang berhenti berlangganan dan masih berlangganan layanan, berapa proporsi masing-masing jenis kontrak menurut kelompok masing-masing?
7. Di antara 'gender', 'Partner', 'Dependents', 'PhoneService', 'InternetService', 'contract', dan 'PaymentMethod', variabel apa yang paling memengaruhi LTV?

Let's start by explaining my whole data analysis steps in this project:

`Step 1` : Gather the data

`Step 2` : Assess and clean the data

`Step 3` : Conduct exploratory data analysis to answer the questions & create visualizations

`Step 4` : Understand the limilations

`Step 5` : Summaries

`Step 6` : Actionable insights

Prepare Data Wrangling

In [7]:
```
#Importing the basic librarires fot analysis

import numpy as np
import pandas as pd
import seaborn as sns
import statsmodels.api as sm
import os
import opendatasets as od
import matplotlib.pyplot as plt
plt.style.use("ggplot")  #using style ggplot

%matplotlib inline

# suppress warnings from final output
import warnings
warnings.simplefilter("ignore")
```

Step 1: Gather the data

In [2]:
```
# Download the data set files

# Assign the Kaggle data set URL into variable
dataset = 'https://www.kaggle.com/datasets/blastchar/telco-customer-churn'
# Using opendatasets let's download the data sets
od.download(dataset)
```

```
Please provide your Kaggle credentials to download this dataset. Learn more: http://bit.ly/kaggle-creds
Your Kaggle username:
Your Kaggle Key:
Downloading telco-customer-churn.zip to ./telco-customer-churn
100%|██████████| 172k/172k [00:00<00:00, 465kB/s]
```

In [3]:
```
data_dir ='./telco-customer-churn'
```

In [4]:
```
os.listdir(data_dir)
```

Out[4]:
```
['WA_Fn-UseC_-Telco-Customer-Churn.csv']
```

In [5]:
```
#Importing the dataset

df = pd.read_csv('./telco-customer-churn/WA_Fn-UseC_-Telco-Customer-Churn.cs
```

In [6]:
```
# show field dataset
df.info()
```

```
<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
```

Data Assessing

In [8]:
```
# set up to view all the info of the columns
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
```

In [9]:
```
df.sample(5)
```

Out[9]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService
6187	0224-NIJLP	Male	0	Yes	Yes	8	Yes
5608	9705-ZJBCG	Female	0	Yes	Yes	13	Yes
2778	6599-GZWCM	Female	0	No	No	13	Yes
5702	5287-QWLKY	Male	1	Yes	Yes	71	Yes
4307	6899-PPEEA	Female	1	No	No	37	Yes

In [10]:
```
# View number of Contract type
df.Contract.value_counts()
```

Out[10]:
```
Contract
Month-to-month    3875
Two year          1695
One year          1473
Name: count, dtype: int64
```

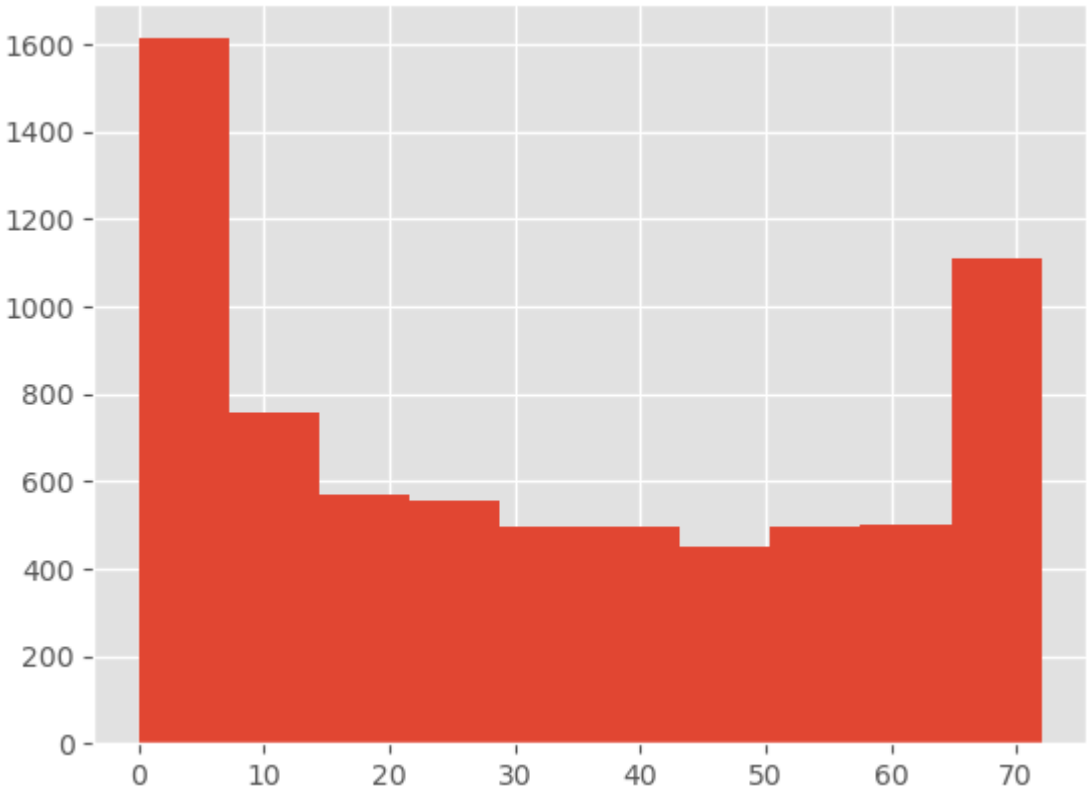
In [11]:
```
# View number of Churn
df.Churn.value_counts()
```

Out[11]:
```
Churn
No     5174
Yes    1869
Name: count, dtype: int64
```

In [12]:
```
# View number of tenure
df.tenure.describe()
```

Out[12]:
```
count    7043.000000
mean       32.371149
std        24.559481
min         0.000000
25%         9.000000
50%        29.000000
75%        55.000000
max        72.000000
Name: tenure, dtype: float64
```

In [13]:
```
# show the distribution of tenure.
plt.hist(data = df, x = 'tenure');
```



- Hal ini rupanya bukan distribusi normal. Dan dengan dua puncak, ada dua jenis orang ekstrem di antara semua pelanggan, dan saya akan menyelidiki layanan apa yang paling bertahan lama bagi mereka yang bertahan lebih dari 70 bulan.
- There is no tidiness issue and only 2 issues here to consider a very clean dataset. Here are the 2 Quality issues:
    1. The data type of “TotalCharges” should be the float64 type instead of the object type.
    2. Many rows of “TotalCharges” do not equal each tenue times monthly charges.

In [14]:
```
# First thing first, we should copy our original dataset:

df_copy = df.copy()
```

> # Step 2:

Cleaning 1: The data type of “TotalCharges” should be the float64 type instead of the object type

`Give None value to all rows, then convert it to the data type of float64` . (I will recalculate it later)

In [15]:
```
df_copy.TotalCharges = None
df_copy.TotalCharges=df_copy.TotalCharges.astype(float)
```

In [16]:
```
# Test the results:
df_copy.TotalCharges.dtype
```

Out[16]:
```
dtype('float64')
```

Cleaning 2: Many rows of "TotalCharges" do not equal each tenue times monthly charges.

`Recalculate it, let each tenures times monthly charges`

In [17]:
```
df_copy.TotalCharges = df_copy.tenure * df_copy.MonthlyCharges
```

In [18]:
```
# test result
df_copy[df_copy.tenure * df_copy.MonthlyCharges != df_copy.TotalCharges].sha
```

Out[18]:
```
(0, 21)
```

The final step of the cleaning process: save the data.

In [20]:
```
# store the clean data
df_copy.reset_index(drop=True)
df_copy.to_csv('./telco-customer-churn/Telco-Customer-Churn_clean.csv')
```

In [24]:
```
#load data
clean_df = pd.read_csv('./telco-customer-churn/Telco-Customer-Churn_clean.cs
```

> Step 3:

Exploratory Data Analysis

Sebelum menulis visualisasi apa pun, saya ingin membuat fungsi yang dapat digunakan kembali, sehingga saya dapat menghemat banyak waktu tanpa menulis kode yang sama:

In [21]:
```
def desc(title=None, xscale=None, yscale=None, xlabel=None, ylabel=None, xli
    if title:
        plt.title(title);
    if xscale:
        plt.xscale(xscale);
    if yscale:
        plt.yscale(yscale);
    if xlabel:
        plt.xlabel(xlabel);
    if ylabel:
        plt.ylabel(ylabel);
```

```
    if xlim:
        plt.xlim(xlim);
    if ylim:
        plt.ylim(ylim);
    if xticks1:
        plt.xticks(xticks1, xticks2);
    if yticks1:
        plt.yticks(yticks1, yticks2);
    if legend_title:
        plt.legend(title=legend_title);
        if legend_labels:
            plt.legend(title=legend_title, labels=legend_labels);
```

`Pertanyaan 1` : Berapa lama orang yang unsubscribe dan membayar layanan biasanya bertahan dalam layanan? Dan berapa LTV (Live Time Value) rata-rata mereka?

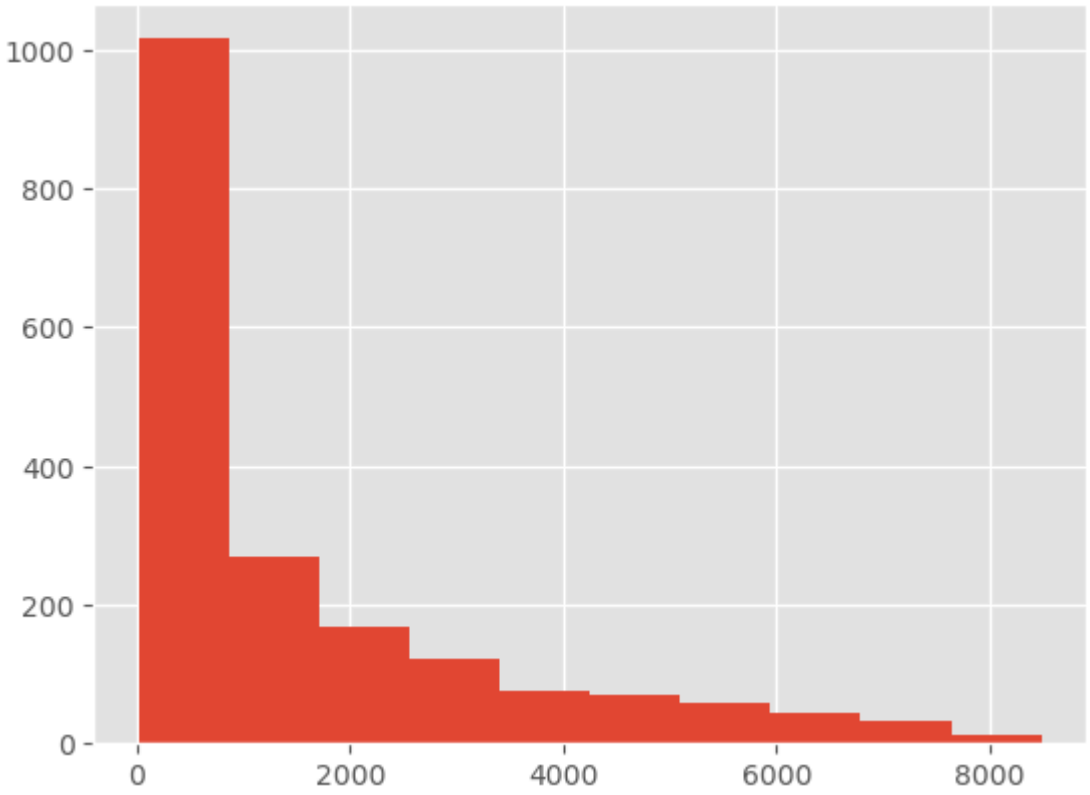
In [25]:
```
Churn_df = clean_df.query('Churn=="Yes"')
Churn_df.TotalCharges.describe()
```

Out[25]:
```
count    1869.000000
mean     1531.608828
std      1886.774930
min        18.850000
25%       137.900000
50%       700.000000
75%      2334.800000
max      8481.600000
Name: TotalCharges, dtype: float64
```

In [26]:
```
# Examine the distribution of TotalCharges
clean_df.TotalCharges.describe()
```

Out[26]:
```
count    7043.000000
mean     2279.581350
std      2264.729447
min         0.000000
25%       394.000000
50%      1393.600000
75%      3786.100000
max      8550.000000
Name: TotalCharges, dtype: float64
```

In [28]:
```
# Visualize
plt.hist(data = Churn_df, x = 'TotalCharges');
```



Note :

- Saya menemukan bahwa sekitar 20% datanya sangat tinggi, jadi saya memutuskan untuk membaginya untuk melihat setiap distribusi data.

In [29]:
```
#find the 80th percentile of the data in total charges
Churn_df.TotalCharges.quantile(0.8)
```

Out[29]:
```
2827.5900000000006
```

In [30]:
```
# Divide the data by the 80th percentile of the data, and show the distribut

TotalCharges_under80 = Churn_df.query('TotalCharges<=2827.59')
TotalCharges_above80 = Churn_df.query('TotalCharges>2827.59')
TotalCharges_under80.TotalCharges.describe()
```

Out[30]:
```
count    1495.000000
mean      713.561672
std       769.669864
min        18.850000
25%        85.900000
50%       377.600000
75%      1132.575000
max      2825.650000
Name: TotalCharges, dtype: float64
```

In [31]:
```
#show the distribution of its TotalCharges above 80th percentile

TotalCharges_above80.TotalCharges.describe()
```

Out[31]:
```
count     374.000000
mean     4801.610160
std      1432.384076
min      2830.500000
25%      3523.275000
50%      4607.300000
75%      5863.450000
max      8481.600000
Name: TotalCharges, dtype: float64
```

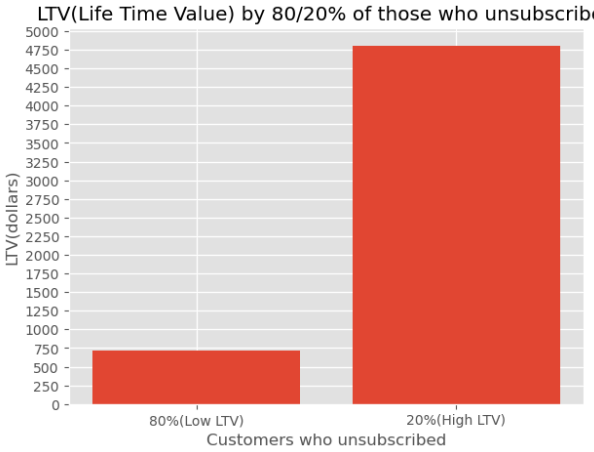
In [32]:
```
#show the distribution of its tenure under 80th percentile
TotalCharges_under80.tenure.describe()
```

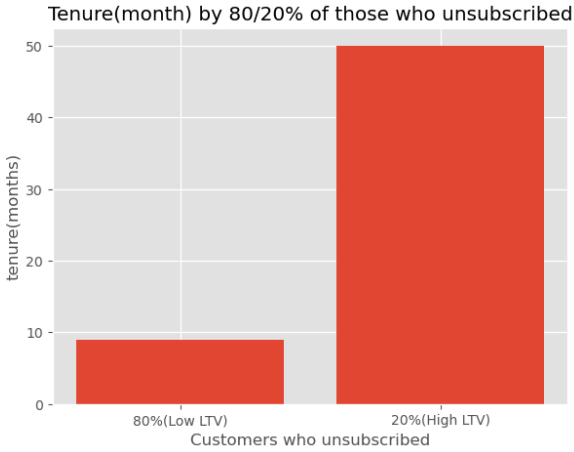
Out[32]:
```
count    1495.000000
mean        9.933110
std        10.738504
min         1.000000
25%         1.000000
50%         6.000000
75%        15.000000
max        61.000000
Name: tenure, dtype: float64
```

In [33]:
```
#show the distribution of its tenure above 80th percentile
TotalCharges_above80.tenure.describe()
```

Out[33]:
```
count    374.000000
mean      50.141711
std       12.322030
min       28.000000
25%       40.000000
50%       49.500000
75%       60.000000
max       72.000000
Name: tenure, dtype: float64
```

In [34]:
```
# Visualize both

plt.figure(figsize = [15, 5])
# left plot: LTV by above and under 80th percentile of data who unsubscribed
plt.subplot(1, 2, 1)
plt.bar([1, 2], [713, 4801])
desc(yticks1=range(0,5250,250), yticks2=range(0,5250,250), xticks1=[1,2],xti

# # right plot: Tenure by above and under 80th percentile of data who unsubs

plt.subplot(1, 2, 2)
plt.bar([1, 2], [9, 50])
desc(xticks1=[1,2],xticks2=['80%(Low LTV)', '20%(High LTV)'],ylabel='tenure(
```





`Note:`

- LTV rata-rata 80% dari mereka yang berhenti berlangganan adalah 750 dolar, dan jangka waktunya hampir 10 bulan.
- Di sisi lain, rata-rata LTV dari 20% teratas dari mereka yang berhenti berlangganan adalah 4.750 dolar, dan jangka waktunya mendekati 50 bulan.
- Dan rasio jumlah total LTV tiap grup adalah 750*4 : 4750 = 1 : 1,6, yang berarti kita harus fokus melayani 20% pelanggan dengan LTV tinggi, yang menghasilkan 60% (1,6/2,6) pendapatan kita .

`Pertanyaan 2` : Sehubungan dengan Pertanyaan 1, jenis layanan apa yang mereka berlangganan ketika mereka masih berlangganan?

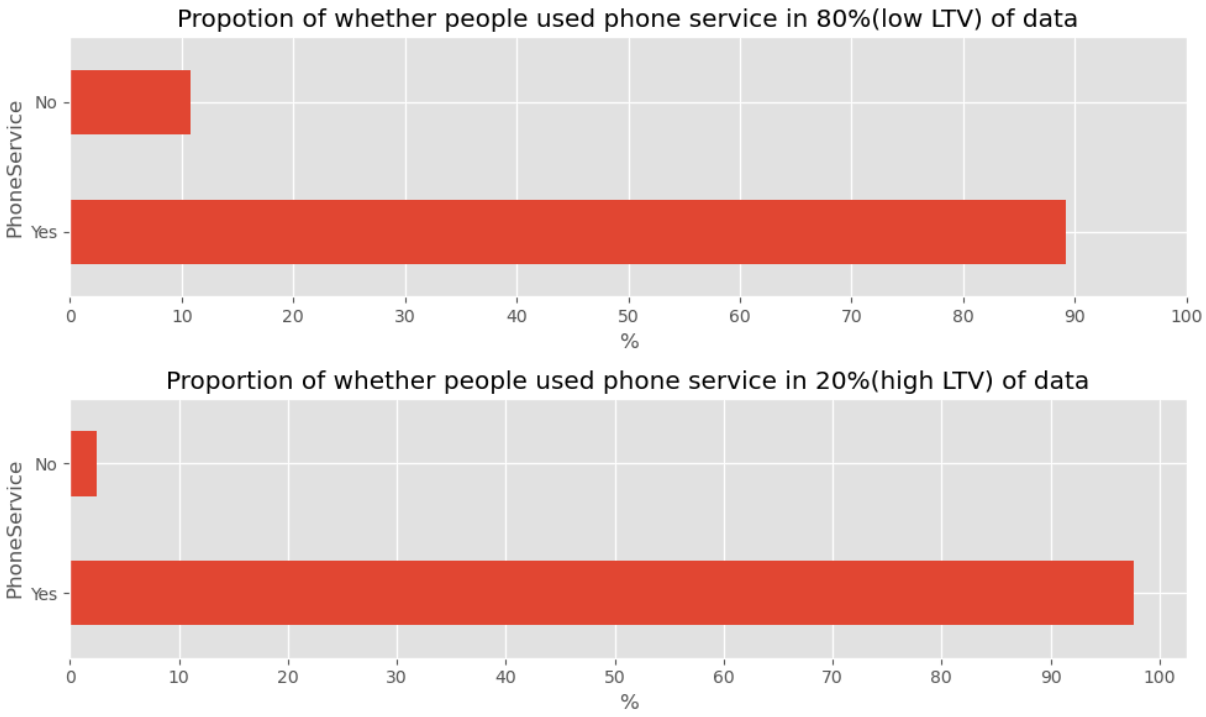
`Catatan:` Karena saya baru mengetahui bahwa ada perbedaan besar dalam LTV dan masa kerja antara 80/20% dari mereka yang berhenti berlangganan, maka saya memutuskan untuk menyelidiki pertanyaan ini pada kedua kelompok 80/20%.

In [35]:
```
# Extract 80% with low LTV who used the internet service, and save each prop
#in the variable "proportion_internet_sub_service"

TotalCharges_under80_use_internet = TotalCharges_under80.query('InternetServ
proportion_internet_sub_service = np.array([TotalCharges_under80_use_interne
TotalCharges_under80_use_internet.query('TechSupport=="Yes"').shape[0]/Total
TotalCharges_under80_use_internet.query('OnlineBackup=="Yes"').shape[0]/Tota
TotalCharges_under80_use_internet.query('DeviceProtection=="Yes"').shape[0]/
TotalCharges_under80_use_internet.query('StreamingTV=="Yes"').shape[0]/Total
TotalCharges_under80_use_internet.query('StreamingMovies=="Yes"').shape[0]/T
```

In [36]:
```
# Extract 20% with high LTV who used the internet service, and save each pro
#in the variable "proportion_internet_sub_service_above80"

TotalCharges_above80_use_internet = TotalCharges_above80.query('InternetServ
proportion_internet_sub_service_above80 = np.array([TotalCharges_above80_use
TotalCharges_above80_use_internet.query('TechSupport=="Yes"').shape[0]/Total
TotalCharges_above80_use_internet.query('OnlineBackup=="Yes"').shape[0]/Tota
```

```
TotalCharges_above80_use_internet.query('DeviceProtection=="Yes"').shape[0]/
TotalCharges_above80_use_internet.query('StreamingTV=="Yes"').shape[0]/Total
TotalCharges_above80_use_internet.query('StreamingMovies=="Yes"').shape[0]/T
```

In [37]:
```
# Investigate the proportion of people used phone service by each groups
plt.figure(figsize = [10, 6])

plt.subplot(2, 1, 1)
# Group 1: 80%(low LTV) of data
((TotalCharges_under80.PhoneService.value_counts()/TotalCharges_under80.shap
desc(title="Propotion of whether people used phone service in 80%(low LTV) o

plt.subplot(2, 1, 2)
# Group 2: 20%(high LTV) of data
((TotalCharges_above80.PhoneService.value_counts()/TotalCharges_above80.shap
desc(title="Proportion of whether people used phone service in 20%(high LTV)
plt.tight_layout()
```

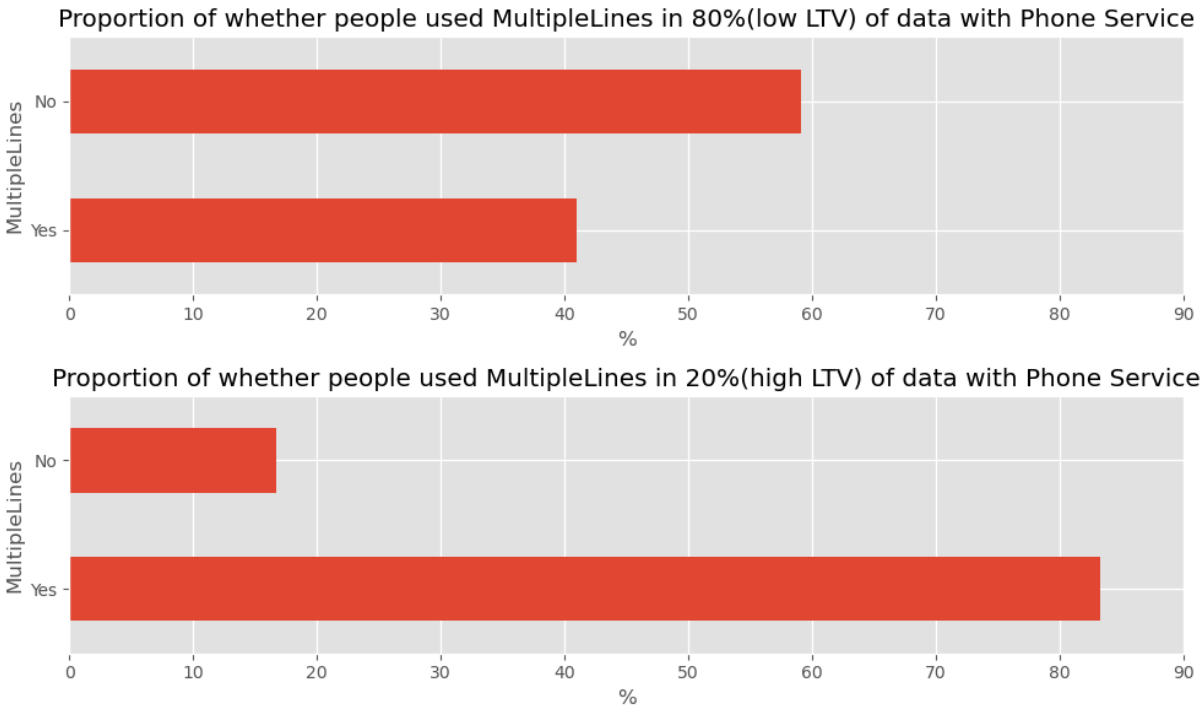


Note:

- 20% (LTV tinggi) data hanya dimiliki 2% yang tidak menggunakan layanan telepon.
- Di sisi lain, 80% (LTV rendah) data terdapat 11% yang tidak menggunakan layanan telepon, yang berarti 5 kali lebih besar dari proporsi layanan telepon sebesar 20% (LTV tinggi) data.

In [38]:
```
# Investigate the proportion of people who used phone service with muiltple
plt.figure(figsize = [10, 6])

plt.subplot(2, 1, 1)
# Group 1: 80%(low LTV) of data
TotalCharges_under80_use_phone = TotalCharges_under80.query('PhoneService=="
(((TotalCharges_under80_use_phone.MultipleLines.value_counts()/TotalCharges_
```

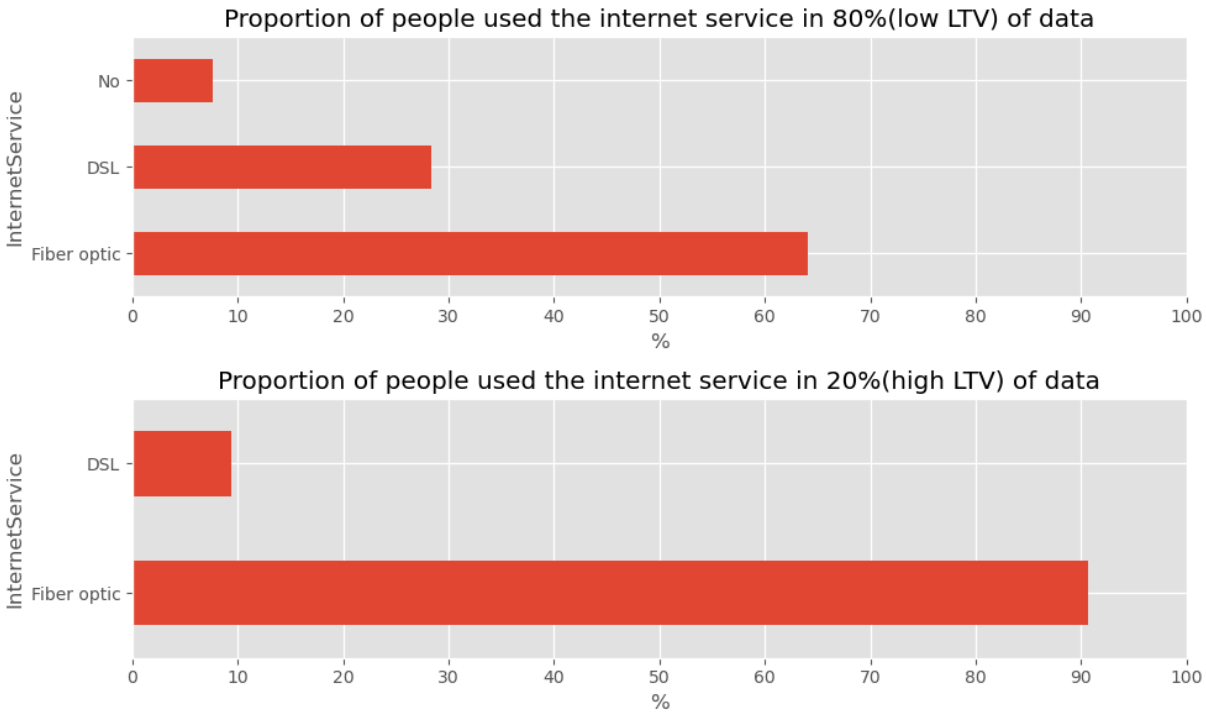
```
desc(title="Proportion of whether people used MultipleLines in 80%(low LTV)

plt.subplot(2, 1, 2)
# Group 2: 20%(high LTV) of data
TotalCharges_above80_use_phone = TotalCharges_above80.query('PhoneService=="
((TotalCharges_above80_use_phone.MultipleLines.value_counts()/TotalCharges_a
desc(title="Proportion of whether people used MultipleLines in 20%(high LTV)

plt.tight_layout()
```



Note:

- 20% (LTV tinggi) data dari mereka yang menggunakan layanan telepon memiliki 84% menggunakan banyak saluran, yang merupakan 2 kali lebih banyak dari proporsi beberapa saluran dalam 80% (LTV rendah) data dari mereka yang menggunakan layanan telepon .

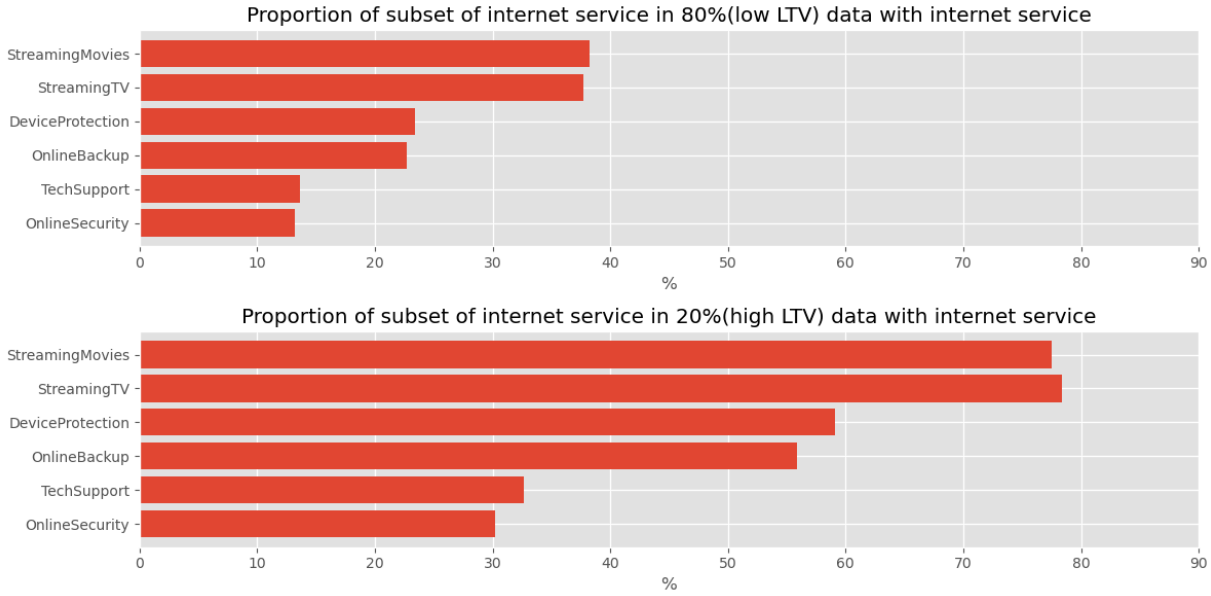
In [39]:
```
# Investigate proportion of people who used the internet service by each gou
plt.figure(figsize = [10, 6])

plt.subplot(2, 1, 1)
# Group 1: 80%(low LTV) of data
((TotalCharges_under80.InternetService.value_counts()/TotalCharges_under80.s
desc(title="Proportion of people used the internet service in 80%(low LTV) o

plt.subplot(2, 1, 2)
# Group 2: 20%(high LTV) of data
((TotalCharges_above80.InternetService.value_counts()/TotalCharges_above80.s
desc(title="Proportion of people used the internet service in 20%(high LTV)

plt.tight_layout()
```



Note:

- Saya menemukan wawasan bahwa semua yang memiliki LTV tinggi semuanya menggunakan layanan internet.
- Sebaliknya, dari 80% responden yang memiliki LTV rendah, 8% diantaranya tidak menggunakan layanan internet.
- Dan, pada 20% data (LTV tinggi), terdapat 90% masyarakat menggunakan serat optik sebagai layanan internetnya.

In [40]:
```
# Investigate the Proportion by subset of internet service by each groups
plt.figure(figsize = [12, 6])

plt.subplot(2, 1, 1)
# Group 1: 80%(low LTV) of data
plt.barh(range(0,6), proportion_internet_sub_service)
desc(title='Proportion of subset of internet service in 80%(low LTV) data wi
plt.subplot(2, 1, 2)
# Group 2: 20%(high LTV) of data
plt.barh(range(0,6), proportion_internet_sub_service_above80)
desc(title='Proportion of subset of internet service in 20%(high LTV) data w
plt.tight_layout()
```



`Note :`

1. Proporsi film streaming dan layanan TV streaming di 20% (LTV tinggi) data keduanya mendekati 80%, yaitu 2 kali lebih besar dari proporsi serupa di 80% (LTV rendah) data.
2. Proporsi perlindungan perangkat dan layanan pencadangan online keduanya mendekati 58%, yaitu 2,5-3 kali lebih besar dari proporsi yang sama pada data sebesar 80% (LTV rendah).
3. Proporsi dukungan teknis dan layanan keamanan online keduanya mendekati 31%, yaitu sekitar 2,5-3 kali lipat dari proporsi yang sama pada data sebesar 80% (LTV rendah).

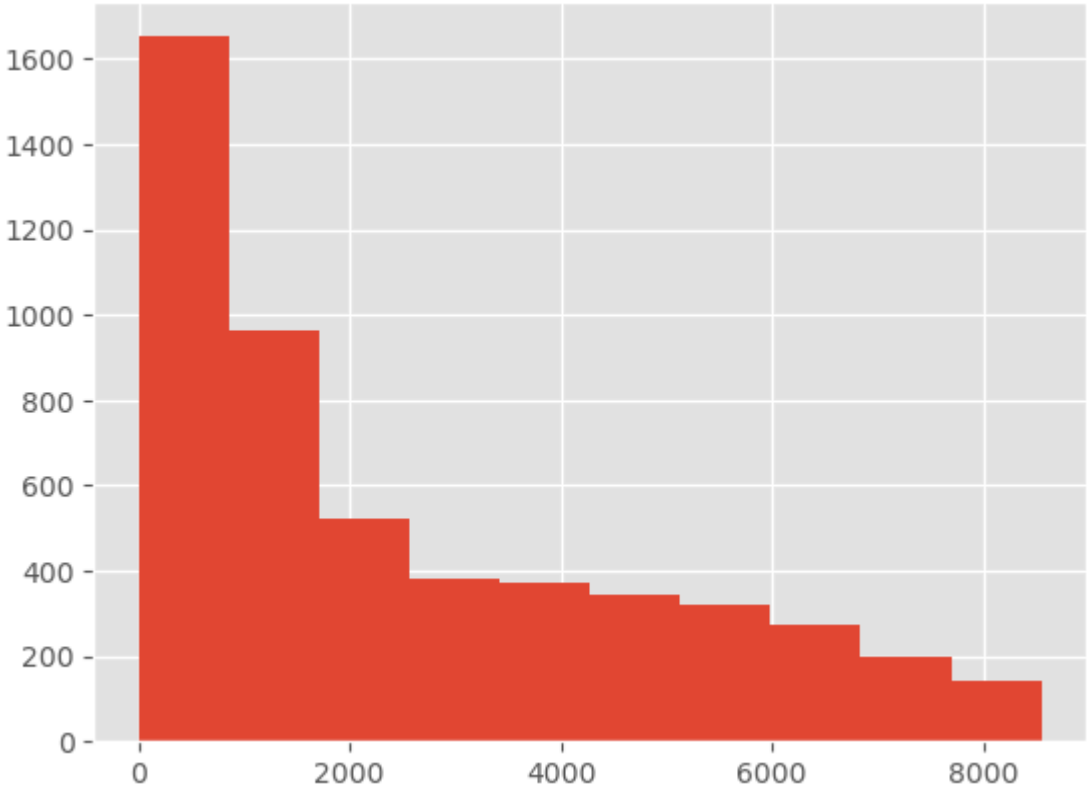
`Pertanyaan 3` : Bagi mereka yang masih membayar layanan, berapa rata-rata LTV (Life Time Value) mereka? dan berapa lama biasanya mereka berada dalam dinas?

In [41]:
```
# Extract those who are paying the service
paying_df = clean_df.query('Churn=="No"')
```

In [42]:
```
paying_df.TotalCharges.describe()
```

Out[42]:
```
count    5174.000000
mean     2549.770883
std      2328.399619
min         0.000000
25%       574.562500
50%      1687.125000
75%      4244.812500
max      8550.000000
Name: TotalCharges, dtype: float64
```

In [43]:
```
# Visualize
plt.hist(data = paying_df, x = 'TotalCharges');
```



`Note:` Untuk menyelidiki data secara merata, saya memutuskan untuk membagi data menjadi 80/20% seperti dua pertanyaan terakhir

In [44]:
```
#find the 80th percentile of the data in total charges
paying_df.TotalCharges.quantile(0.8)
```

Out[44]:
```
4890.720000000001
```

In [45]:
```
# Divide the data by the 80th percentile of the data, and show the distribut
paying_TotalCharges_under80 = paying_df.query('TotalCharges<=4890')
paying_TotalCharges_above80 = paying_df.query('TotalCharges>4890')
paying_TotalCharges_under80.TotalCharges.mean(),paying_TotalCharges_above80.
```

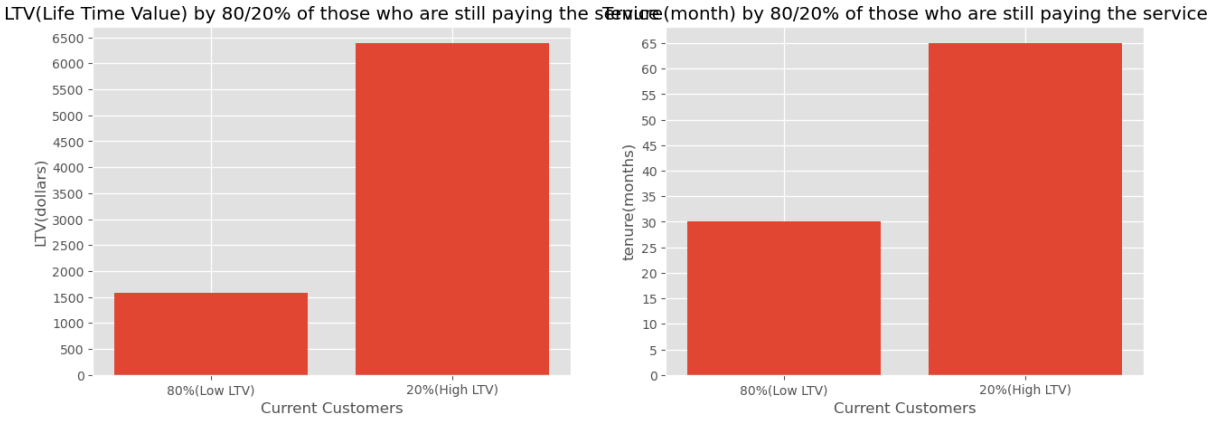
Out[45]:
```
(1589.5728195216236, 6389.63541062802)
```

In [46]:
```
paying_TotalCharges_under80.tenure.mean(),paying_TotalCharges_above80.tenure
```

Out[46]:
```
(30.60207779656922, 65.43478260869566)
```

In [47]:
```
# Visualize both

plt.figure(figsize = [15, 5])

# left plot: LTV by above and under 80th percentile of data who are still pa

plt.subplot(1, 2, 1)
```

```
plt.bar([1, 2], [1589, 6389])
desc(yticks1=range(0,7000,500), yticks2=range(0,7000,500), xticks1=[1,2],xti

# # right plot: Tenure by above and under 80th percentile of data who are st

plt.subplot(1, 2, 2)
plt.bar([1, 2], [30, 65])
desc(yticks1=range(0,70,5), yticks2=range(0,70,5), xticks1=[1,2],xticks2=['8
```



`Note:`

- Rata-rata LTV di 80% (LTV rendah) data adalah 1589, dan satu di 20% (LTV tinggi) data adalah 6389.
- Rata-rata tenor di 80% (LTV rendah) data adalah 30, dan satu dalam 20% data (LTV tinggi) adalah 65.
- Dan rasio jumlah total LTV tiap kelompok adalah 1590*4 : 6389 = 1 : 1, yang berarti kita harus fokus melayani 20% pelanggan dengan LTV tinggi, yang menghasilkan 50% (1/2) pendapatan kami.

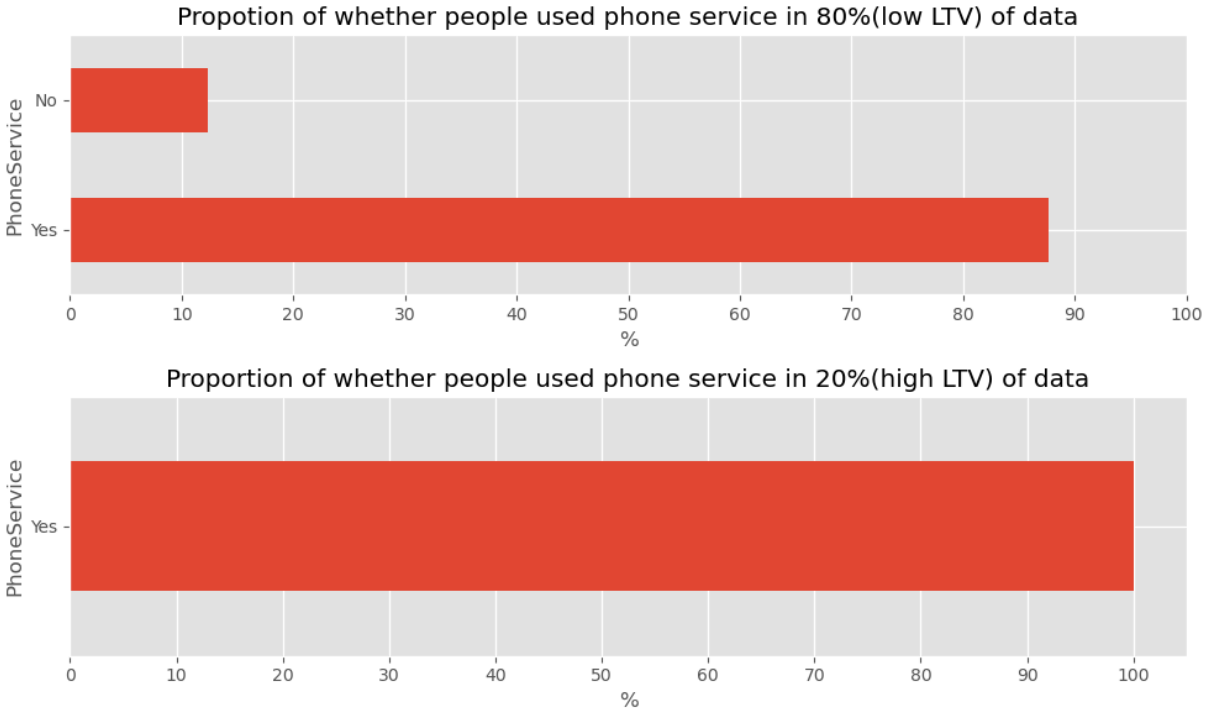
`Pertanyaan 4` : Terkait dengan Pertanyaan 3, berapa proporsi masing-masing jenis layanan yang mereka bayarkan?

In [48]:
```
# Extract 80% with low LTV who used the internet service, and save each prop
#in the variable "proportion_internet_sub_service"

paying_TotalCharges_under80_use_internet = paying_TotalCharges_under80.query
paying_proportion_internet_sub_service_under80 = np.array([paying_TotalCharg
paying_TotalCharges_under80_use_internet.query('TechSupport=="Yes"').shape[0
paying_TotalCharges_under80_use_internet.query('OnlineBackup=="Yes"').shape[
paying_TotalCharges_under80_use_internet.query('DeviceProtection=="Yes"').sh
paying_TotalCharges_under80_use_internet.query('StreamingTV=="Yes"').shape[0
paying_TotalCharges_under80_use_internet.query('StreamingMovies=="Yes"').sha
```

In [49]:
```
# Extract 20% with high LTV who used the internet service, and save each pro
#in the variable "proportion_internet_sub_service_above80"

paying_TotalCharges_above80_use_internet = TotalCharges_above80.query('Inter
paying_proportion_internet_sub_service_above80 = np.array([paying_TotalCharg
paying_TotalCharges_above80.query('TechSupport=="Yes"').shape[0]/paying_Tota
```

```
paying_TotalCharges_above80.query('OnlineBackup=="Yes"').shape[0]/paying_Tot
paying_TotalCharges_above80.query('DeviceProtection=="Yes"').shape[0]/paying
paying_TotalCharges_above80.query('StreamingTV=="Yes"').shape[0]/paying_Tota
paying_TotalCharges_above80.query('StreamingMovies=="Yes"').shape[0]/paying_
```

In [50]:
```
# Investigate the proportion of people used phone service by each groups
plt.figure(figsize = [10, 6])

plt.subplot(2, 1, 1)
# Group 1: 80%(low LTV) of data
((paying_TotalCharges_under80.PhoneService.value_counts()/paying_TotalCharge
desc(title="Propotion of whether people used phone service in 80%(low LTV) o

plt.subplot(2, 1, 2)
# Group 2: 20%(high LTV) of data
((paying_TotalCharges_above80.PhoneService.value_counts()/paying_TotalCharge
desc(title="Proportion of whether people used phone service in 20%(high LTV)
plt.tight_layout()
```

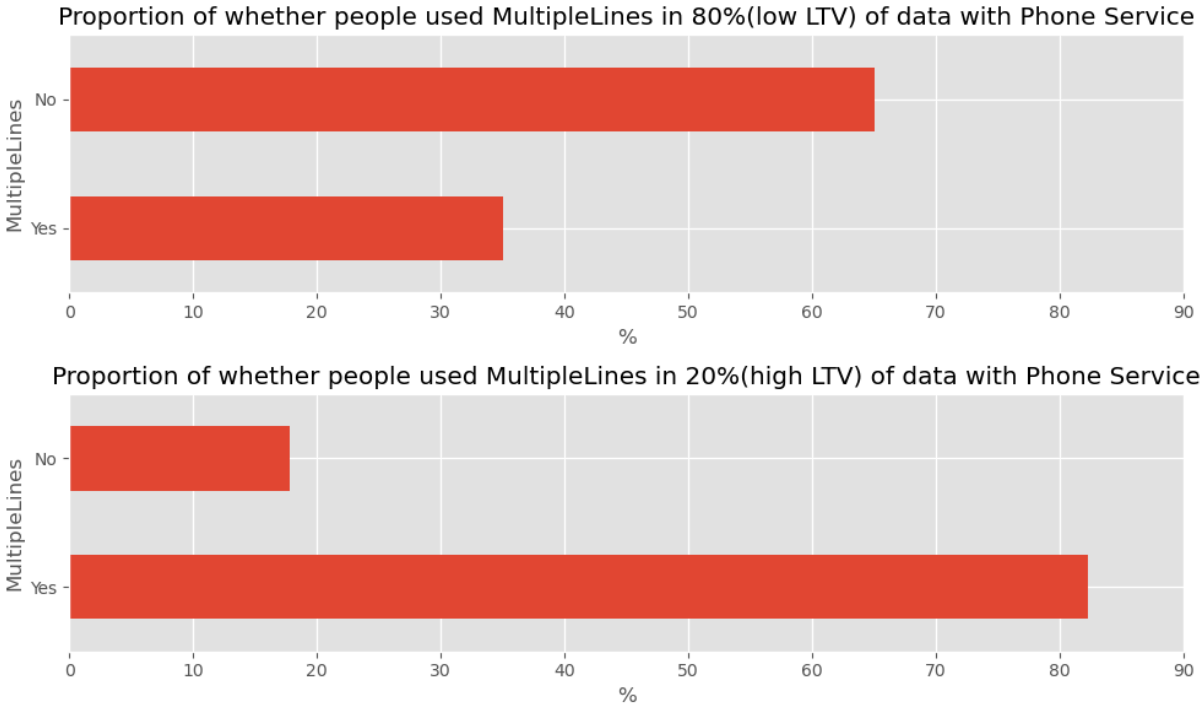


`Note:`

- 20% (LTV tinggi) data hanya dimiliki 0.2% (<< 10 %) yang tidak menggunakan layanan telepon.
- Di sisi lain, 80% (LTV rendah) data terdapat 11% yang tidak menggunakan layanan telepon, yang berarti 5 kali lebih besar dari proporsi layanan telepon sebesar 20% (LTV tinggi) data.

In [51]:
```
# Investigate the proportion of people who used phone service withn muiltple
plt.figure(figsize = [10, 6])

plt.subplot(2, 1, 1)
# Group 1: 80%(low LTV) of data
```

```
paying_TotalCharges_under80_use_phone = paying_TotalCharges_under80.query('P
(((paying_TotalCharges_under80_use_phone.MultipleLines.value_counts()/paying
desc(title="Proportion of whether people used MultipleLines in 80%(low LTV)

plt.subplot(2, 1, 2)
# Group 2: 20%(high LTV) of data
paying_TotalCharges_above80_use_phone = paying_TotalCharges_above80.query('P
((paying_TotalCharges_above80_use_phone.MultipleLines.value_counts()/paying_
desc(title="Proportion of whether people used MultipleLines in 20%(high LTV)

plt.tight_layout()
```

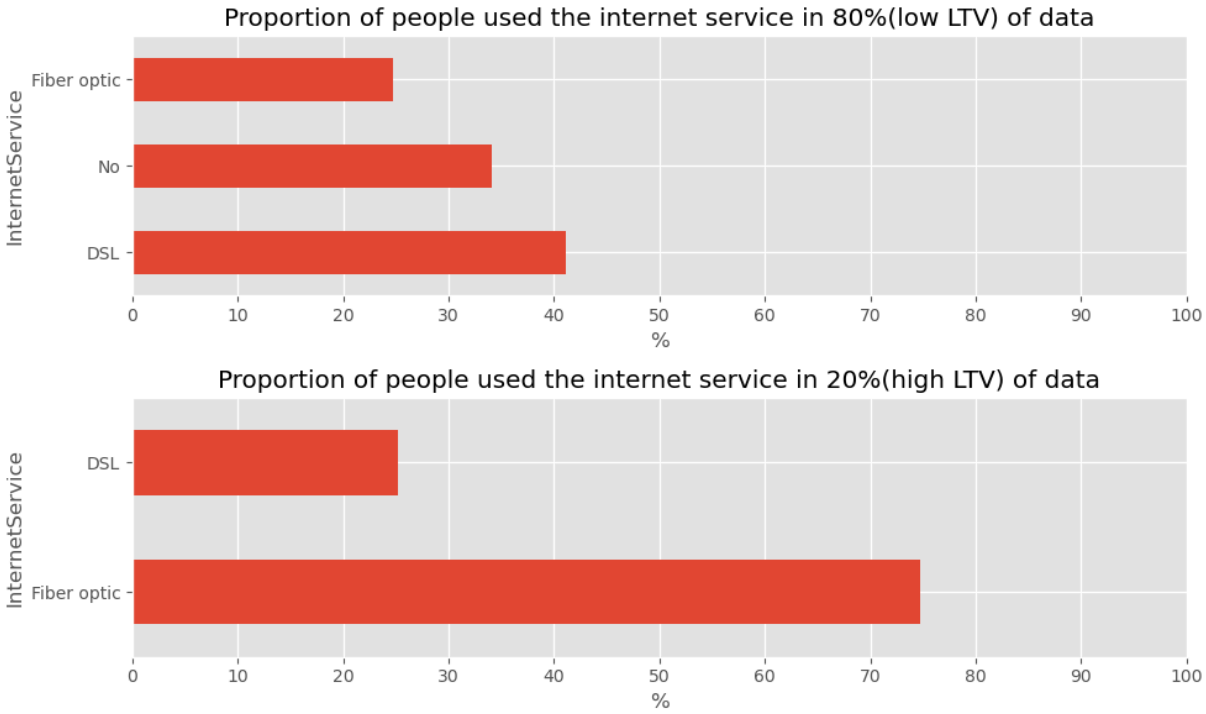


`Note:`

- 20% (LTV tinggi) data dari mereka yang menggunakan layanan telepon memiliki 83% yang menggunakan banyak saluran, yang merupakan 2,4 kali lebih besar dari proporsi beberapa saluran dalam 80% (LTV rendah) data dari mereka yang menggunakan layanan telepon .

In [52]:
```
# VInvestigate proportion of people who used the internet service by each gc
plt.figure(figsize = [10, 6])

plt.subplot(2, 1, 1)
# Group 1: 80%(low LTV) of data
((paying_TotalCharges_under80.InternetService.value_counts()/paying_TotalCha
desc(title="Proportion of people used the internet service in 80%(low LTV) c

plt.subplot(2, 1, 2)
# Group 2: 20%(high LTV) of data
((paying_TotalCharges_above80.InternetService.value_counts()/paying_TotalCha
desc(title="Proportion of people used the internet service in 20%(high LTV)
```

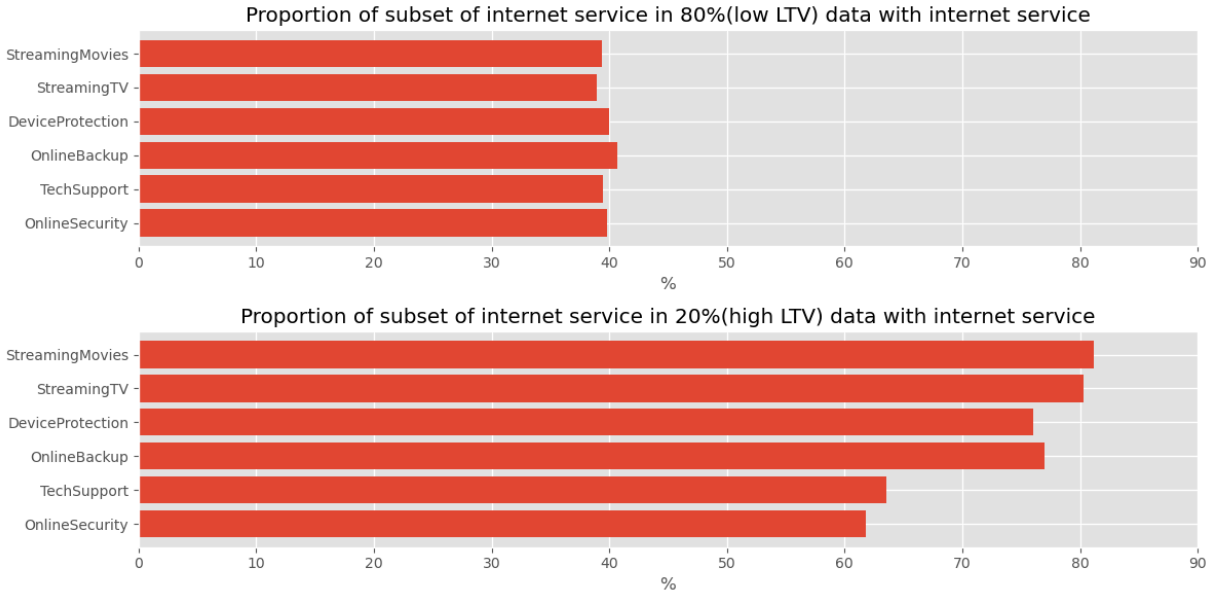
```
plt.tight_layout()
```



`Note:`

- Saya menemukan wawasan bahwa semua yang memiliki LTV tinggi semuanya menggunakan layanan internet.
- Sebaliknya, dari 80% responden yang memiliki LTV rendah, 33% diantaranya tidak menggunakan layanan internet.
- Dan, pada 20% data (LTV tinggi), terdapat 90% masyarakat menggunakan serat optik sebagai layanan internetnya.

In [53]:
```
# Investigate the Proportion by subset of internet service by each groups
plt.figure(figsize = [12, 6])

plt.subplot(2, 1, 1)
# Group 1: 80%(low LTV) of data
plt.barh(range(0,6),paying_proportion_internet_sub_service_under80)
desc(title='Proportion of subset of internet service in 80%(low LTV) data wi
plt.subplot(2, 1, 2)
# Group 2: 20%(high LTV) of data
plt.barh(range(0,6), paying_proportion_internet_sub_service_above80)
desc(title='Proportion of subset of internet service in 20%(high LTV) data w
plt.tight_layout()
```



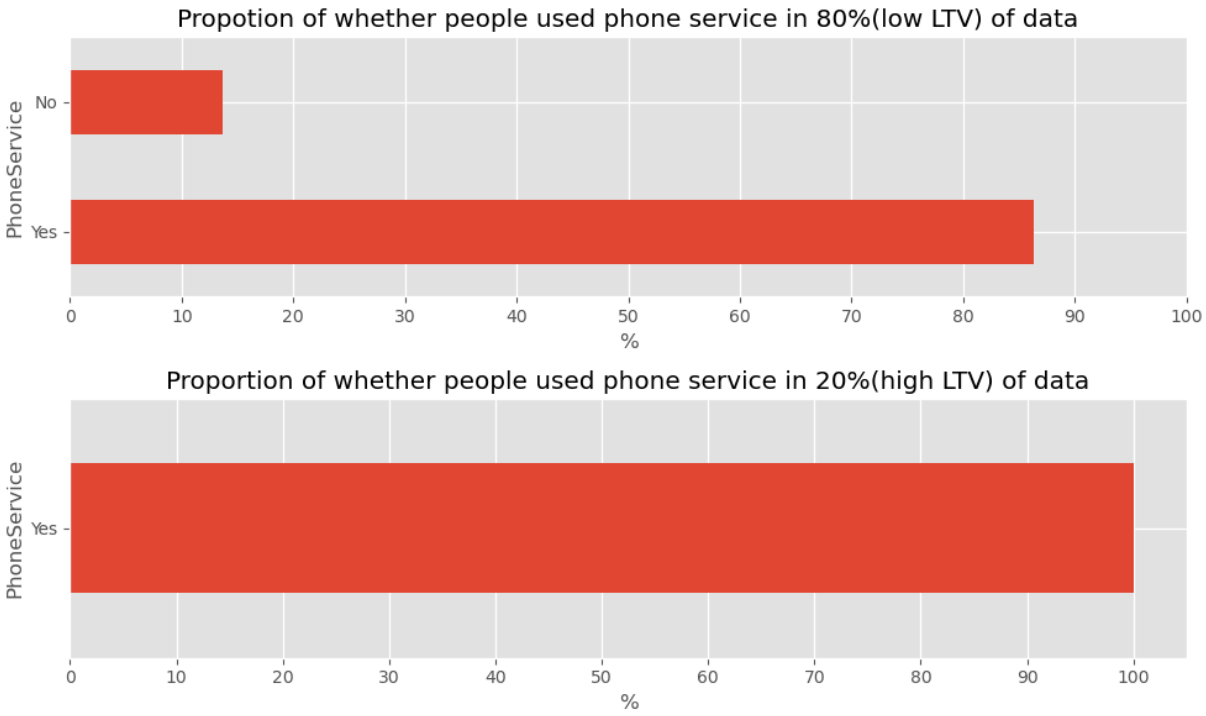
`Note:`

- Seluruh proporsi masing-masing layanan internet pada 80% data (LTV rendah) mendekati 40%.
- Pada 20% data (LTV tinggi), streaming film dan streaming TV mendekati 80%, proporsi perlindungan perangkat dan pencadangan online mendekati 75%, dan proporsi dukungan teknis dan keamanan online mendekati 62%.

`Pertanyaan 5` : Bagi yang masih dalam layanan dengan LTV lebih besar dari LTV pelanggan yang bocor, layanan manakah yang paling mahal dibayarnya?

In [54]:
```
# The average LTV in 80%(low LTV) of data of leaked customers is 750
# The average LTV in 20%(high LTV) of data of leaked customers is 4750

# Extract the 80%(low LTV) of data that the LTV is higher than 750
paying_TotalCharges_under80_higherthanleak = paying_TotalCharges_under80.que
# Extract the 20%(high LTV) of data that the LTV is higher than 4750
paying_TotalCharges_above80_higherthanleak = paying_TotalCharges_above80.que
```

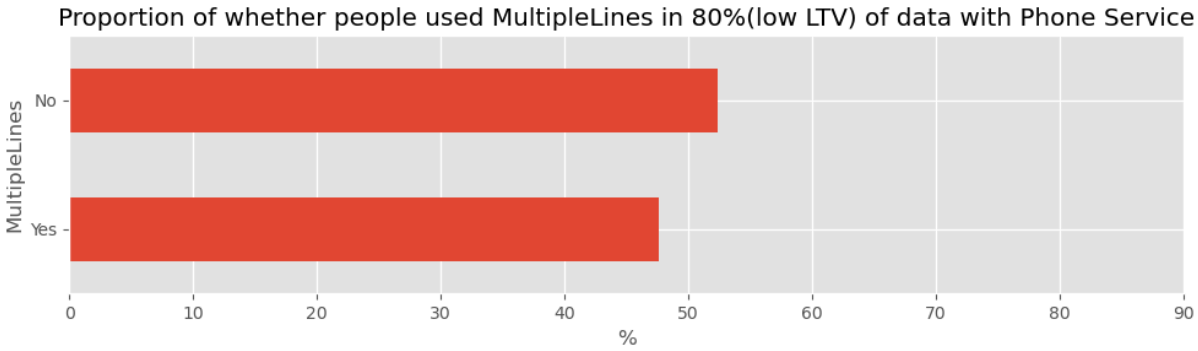
In [55]:
```
# Investigate the proportion of people used phone service by each groups
plt.figure(figsize = [10, 6])

plt.subplot(2, 1, 1)
# Group 1: 80%(low LTV) of data
((paying_TotalCharges_under80_higherthanleak.PhoneService.value_counts()/pay
desc(title="Propotion of whether people used phone service in 80%(low LTV) c

plt.subplot(2, 1, 2)
# Group 2: 20%(high LTV) of data
((paying_TotalCharges_above80_higherthanleak.PhoneService.value_counts()/pay
desc(title="Proportion of whether people used phone service in 20%(high LTV)
plt.tight_layout()
```

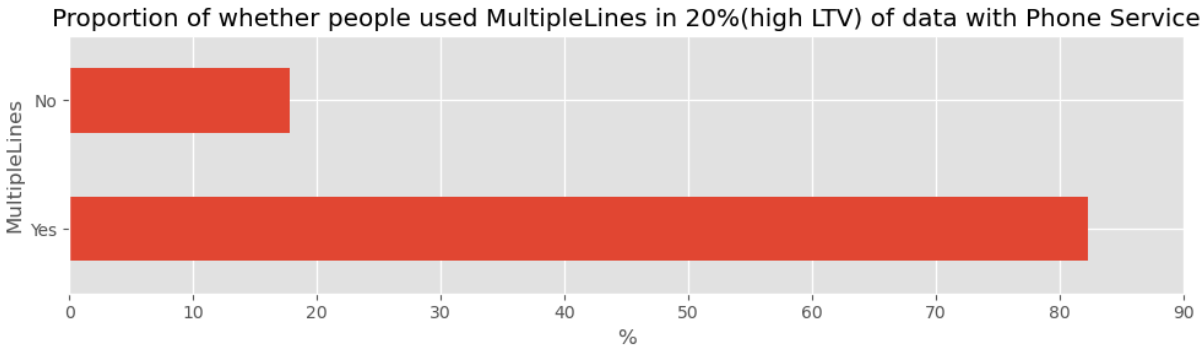


`Note:`

- Pada 80% (LTV rendah) data pelanggan saat ini, yang LTV-nya lebih tinggi dari LTV Pada 80% (LTV rendah) data pelanggan yang bocor, terdapat 86% yang menggunakan layanan telepon.
- Pada 20% (LTV tinggi) data pelanggan saat ini, yang LTV-nya lebih tinggi dari LTV Pada 20% (LTV tinggi) data pelanggan yang bocor, terdapat 100% yang menggunakan layanan telepon.

In [56]:
```
# Investigate the proportion of people who used phone service withn muiltple
plt.figure(figsize = [10, 6])

plt.subplot(2, 1, 1)
# Group 1: 80%(low LTV) of data
paying_TotalCharges_under80_use_phone_higherthanleak = paying_TotalCharges_u
(((paying_TotalCharges_under80_use_phone_higherthanleak.MultipleLines.value_
desc(title="Proportion of whether people used MultipleLines in 80%(low LTV)

plt.subplot(2, 1, 2)
# Group 2: 20%(high LTV) of data
paying_TotalCharges_above80_use_phone_higherthanleak = paying_TotalCharges_a
((paying_TotalCharges_above80_use_phone_higherthanleak.MultipleLines.value_c
desc(title="Proportion of whether people used MultipleLines in 20%(high LTV)

plt.tight_layout()
```

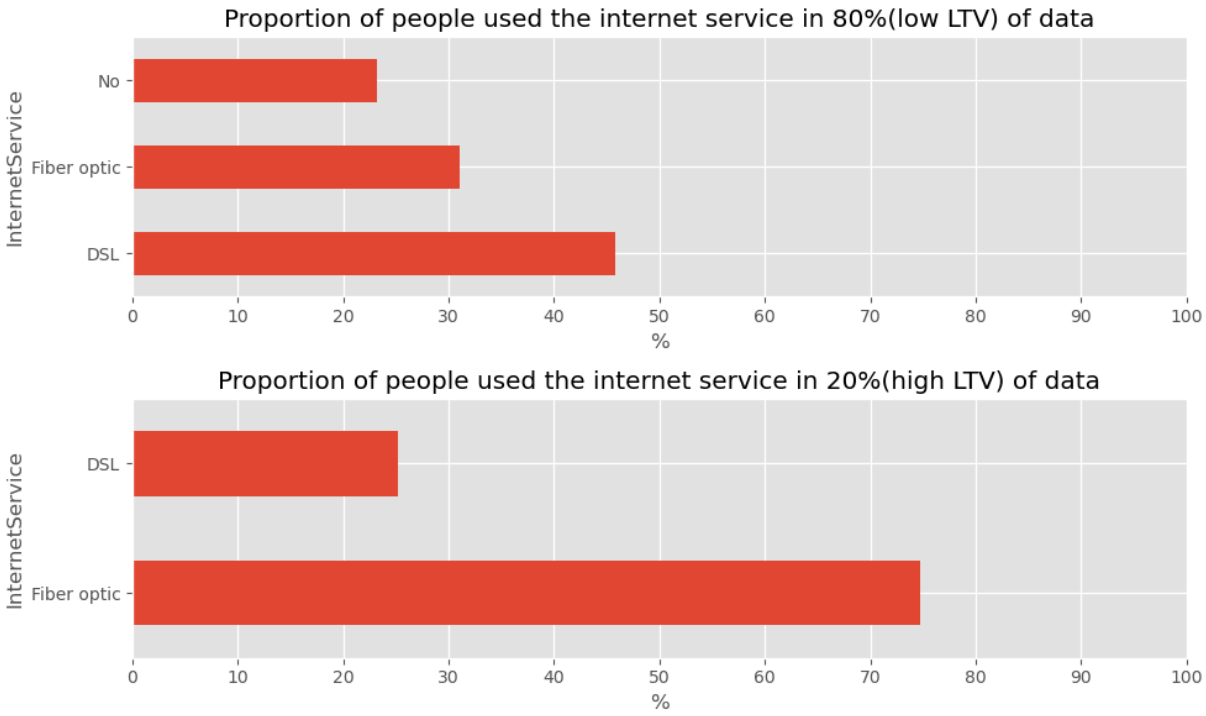




`Note:`

- Pada 80% (LTV rendah) data pelanggan saat ini, yang LTV-nya lebih tinggi dari LTV Pada 80% (LTV rendah) data pelanggan yang bocor, terdapat 48% yang menggunakan layanan telepon dengan banyak saluran.
- Pada 20% (LTV tinggi) data pelanggan saat ini, yang LTV-nya lebih tinggi dari LTV Pada 20% (LTV tinggi) data pelanggan yang bocor, terdapat 89% yang menggunakan layanan telepon.

In [57]:
```
# VInvestigate proportion of people who used the internet service by each gc
plt.figure(figsize = [10, 6])

plt.subplot(2, 1, 1)
# Group 1: 80%(low LTV) of data
((paying_TotalCharges_under80_higherthanleak.InternetService.value_counts()/
desc(title="Proportion of people used the internet service in 80%(low LTV) c

plt.subplot(2, 1, 2)
# Group 2: 20%(high LTV) of data
((paying_TotalCharges_above80_higherthanleak.InternetService.value_counts()/
desc(title="Proportion of people used the internet service in 20%(high LTV)

plt.tight_layout()
```



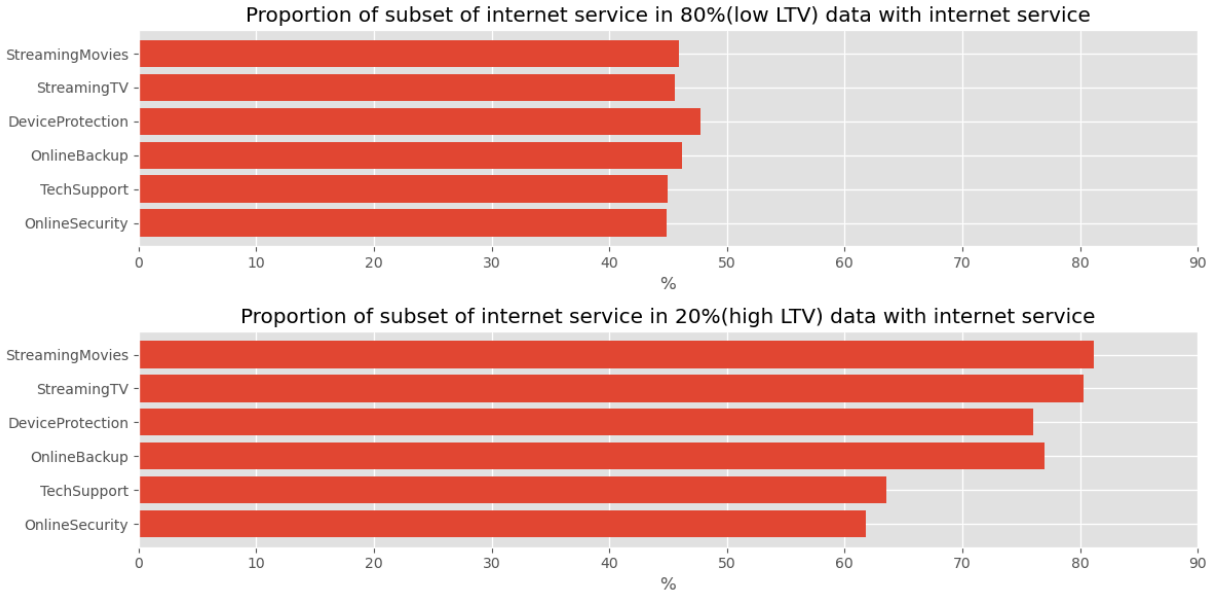
Note :

- Pada 80% (LTV rendah) data pelanggan saat ini, yang LTV-nya lebih tinggi dari LTV Pada 80% (LTV rendah) data pelanggan yang bocor, terdapat 22% yang tidak menggunakan layanan internet
- Pada 20% (LTV tinggi) data pelanggan saat ini, yang LTV-nya lebih tinggi dari LTV Pada 20% (LTV tinggi) data pelanggan yang bocor, terdapat 100% yang menggunakan layanan internet.

In [58]:
```
# Extract 80% with low LTV who used the internet service, and save each prop
#in the variable "proportion_internet_sub_service"

paying_TotalCharges_under80_use_internet_higherthanleak = paying_TotalCharge
paying_proportion_internet_sub_service_under80_higherthanleak = np.array([pa
paying_TotalCharges_under80_use_internet_higherthanleak.query('TechSupport==
paying_TotalCharges_under80_use_internet_higherthanleak.query('OnlineBackup=
paying_TotalCharges_under80_use_internet_higherthanleak.query('DeviceProtect
paying_TotalCharges_under80_use_internet_higherthanleak.query('StreamingTV==
paying_TotalCharges_under80_use_internet_higherthanleak.query('StreamingMovi
```

In [59]:
```
# Extract 20% with high LTV who used the internet service, and save each pro
#in the variable "proportion_internet_sub_service_above80"

paying_TotalCharges_above80_higherthanleak = paying_TotalCharges_above80_hig
paying_proportion_internet_sub_service_above80_higherthanleak = np.array([pa
paying_TotalCharges_above80_higherthanleak.query('TechSupport=="Yes"').shape
paying_TotalCharges_above80_higherthanleak.query('OnlineBackup=="Yes"').shap
paying_TotalCharges_above80_higherthanleak.query('DeviceProtection=="Yes"').
paying_TotalCharges_above80_higherthanleak.query('StreamingTV=="Yes"').shape
paying_TotalCharges_above80_higherthanleak.query('StreamingMovies=="Yes"').s
```

In [60]:
```
# Investigate the Proportion by subset of internet service by each groups
plt.figure(figsize = [12, 6])

plt.subplot(2, 1, 1)
# Group 1: 80%(low LTV) of data
plt.barh(range(0,6),paying_proportion_internet_sub_service_under80_highertha
desc(title='Proportion of subset of internet service in 80%(low LTV) data wi
plt.subplot(2, 1, 2)
# Group 2: 20%(high LTV) of data
plt.barh(range(0,6), paying_proportion_internet_sub_service_above80_higherth
desc(title='Proportion of subset of internet service in 20%(high LTV) data w
plt.tight_layout()
```



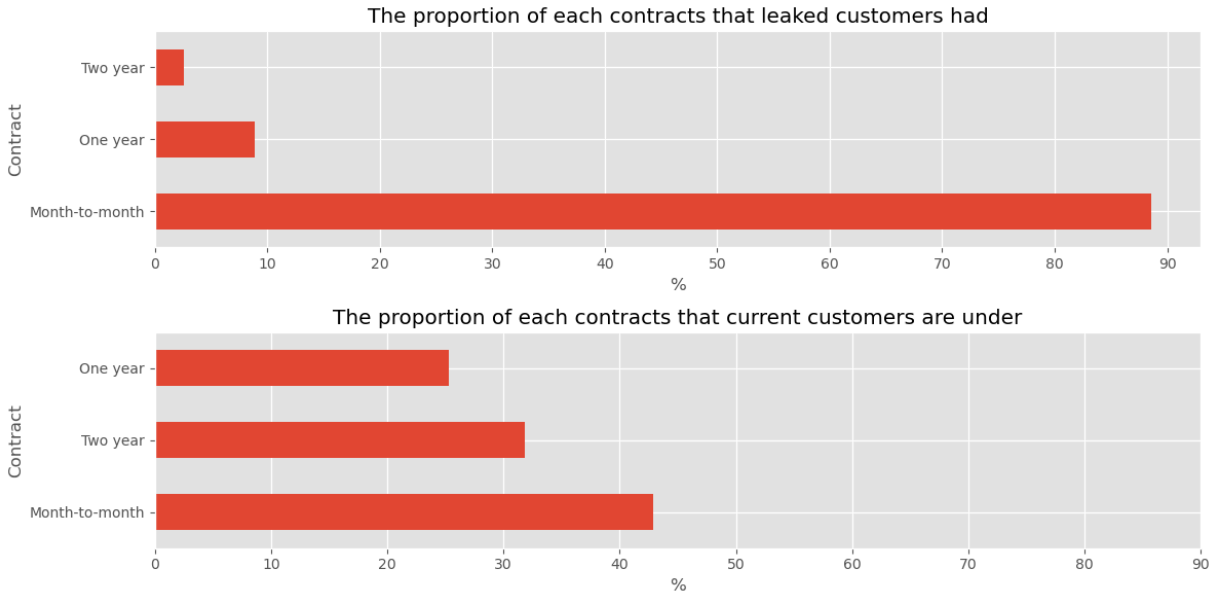
`Note:`

- Pada 80% (LTV rendah) data pelanggan saat ini, yang LTV-nya lebih tinggi dari LTV Pada 80% (LTV rendah) data pelanggan yang bocor, terdapat seluruh proporsi masing-masing layanan internet dalam 80% (LTV rendah) dari datanya mendekati 45%.
- Dalam 20% (LTV tinggi) data pelanggan saat ini, yang LTV-nya lebih tinggi dari LTV Dalam 20% (LTV tinggi) data pelanggan yang bocor, baik streaming film maupun streaming tv mendekati 90%, proporsi perlindungan kedua perangkat dan pencadangan online mendekati 70%, dan proporsi dukungan teknis dan keamanan online mendekati 37%.

`Pertanyaan 6` : Untuk dua kelompok yang bocor dan masih membayar jasa, berapa proporsi masing-masing jenis kontrak oleh masing-masing kelompok?

In [61]:
```
plt.figure(figsize = [12, 6])
# Visualize the plot of leaked customers
plt.subplot(2, 1, 1)
```

```
((Churn_df.Contract.value_counts()/Churn_df.shape[0])*100).plot.barh();
desc(xticks1=range(0,100,10),xticks2=range(0,100,10),ylabel='Contract',xlabe

# Visualize the plot of current customers
plt.subplot(2, 1, 2)
((paying_df.Contract.value_counts()/paying_df.shape[0])*100).plot.barh();
desc(xticks1=range(0,100,10),xticks2=range(0,100,10),ylabel='Contract',xlabe
plt.tight_layout()
```



Note :

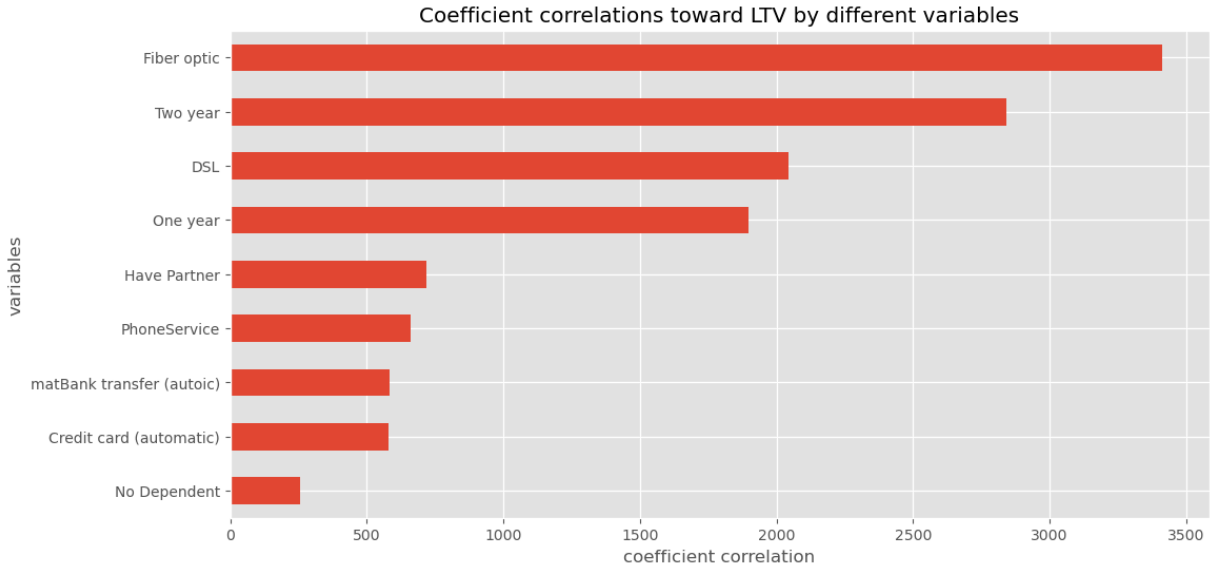
- Dalam data pelanggan yang bocor, terdapat 88% kontrak bekas sebulan, 9% kontrak bekas satu tahun, dan 2% kontrak dua tahun.
- Pada data pelanggan saat ini, terdapat 43% yang menggunakan kontrak sebulan, 25% menggunakan kontrak satu tahun, dan 32% menggunakan kontrak dua tahun.

`Pertanyaan 7` : Di antara 'gender', 'Partner', 'Dependents', 'PhoneService', 'InternetService', 'contract', dan 'PaymentMethod', variabel apa yang paling mempengaruhi LTV?

In [129…

```
clean_df[['Female', 'Male']] = pd.get_dummies(clean_df['gender'])
clean_df[['No', 'Have Partner']] = pd.get_dummies(clean_df['Partner'])
clean_df[['No Dependent', 'Dependents_Yes']] = pd.get_dummies(clean_df['Depe
clean_df[['No', 'PhoneService']] = pd.get_dummies(clean_df['PhoneService'])
clean_df[['DSL', 'Fiber optic', 'No']] = pd.get_dummies(clean_df['InternetSe
clean_df[['Month-to-month', 'One year', 'Two year']] = pd.get_dummies(clean_
clean_df[['matBank transfer (autoic)', 'Credit card (automatic)', 'Electroni

# use mutiple linear regression
clean_df['intercept'] = 1
lm = sm.OLS(clean_df['TotalCharges'], clean_df[['intercept', 'Male', 'Have P
                                                'PhoneService', 'DSL', 'Fiber
                                                'matBank transfer (autoic)',
```

```
                                                    'Electronic check']].astype(
results = lm.fit()

# Visualize the order of the variables that affects LTV from high to low.
plt.figure(figsize = [12, 6])
results.params.sort_values()[3:].plot.barh()
desc(xlabel="coefficient correlation", ylabel='variables',title='Coefficient
```



In [130…

```
clean_df[['Female', 'Male']] = pd.get_dummies(clean_df['gender'])
clean_df[['No', 'Partner_Yes']] = pd.get_dummies(clean_df['Partner'])
clean_df[['Dependents_No', 'Dependents_Yes']] = pd.get_dummies(clean_df['Dep
clean_df[['No', 'PhoneService_Yes']] = pd.get_dummies(clean_df['PhoneService
clean_df[['DSL', 'Fiber optic', 'No']] = pd.get_dummies(clean_df['InternetSe
clean_df[['Month-to-month', 'One year', 'Two year']] = pd.get_dummies(clean_
clean_df[['matBank transfer (autoic)', 'Credit card (automatic)', 'Electroni
```

In [131…

```
# use mutiple linear regression
clean_df['intercept'] = 1
lm = sm.OLS(clean_df['TotalCharges'], clean_df[['intercept', 'Male', 'Partne
                                                'PhoneService_Yes', 'DSL', 'F
                                                'matBank transfer (autoic)',
                                                 'Electronic check']].astype(

results = lm.fit()
results.summary()
```

Out [131…

OLS Regression Results

Dep. Variable:	TotalCharges	**R-squared:**	0.592
Model:	OLS	**Adj. R-squared:**	0.592
Method:	Least Squares	**F-statistic:**	928.1
Date:	Sat, 27 Apr 2024	**Prob (F-statistic):**	0.00
Time:	08:18:52	**Log-Likelihood:**	-61243.
No. Observations:	7043	**AIC:**	1.225e+05
Df Residuals:	7031	**BIC:**	1.226e+05
Df Model:	11		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
intercept	-2408.0897	90.795	-26.522	0.000	-2586.075	-2230.105
Male	35.3705	34.506	1.025	0.305	-32.272	103.013
Partner_Yes	717.1189	40.290	17.799	0.000	638.138	796.100
Dependents_No	254.2784	43.314	5.871	0.000	169.371	339.186
PhoneService_Yes	662.5648	65.517	10.113	0.000	534.132	790.998
DSL	2043.2198	52.409	38.986	0.000	1940.483	2145.957
Fiber optic	3412.4872	52.277	65.276	0.000	3310.008	3514.967
One year	1897.2454	46.573	40.737	0.000	1805.947	1988.543
Two year	2840.6051	48.427	58.658	0.000	2745.674	2935.536
matBank transfer (autoic)	583.9313	54.209	10.772	0.000	477.665	690.198
Credit card (automatic)	579.3687	54.315	10.667	0.000	472.895	685.842
Electronic check	32.3058	52.282	0.618	0.537	-70.183	134.795

Omnibus:	170.388	**Durbin-Watson:**	2.019
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	170.448
Skew:	0.354	**Prob(JB):**	9.72e-38
Kurtosis:	2.719	**Cond. No.**	12.1

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

In [132…

```
results.params.sort_values()[3:]
```

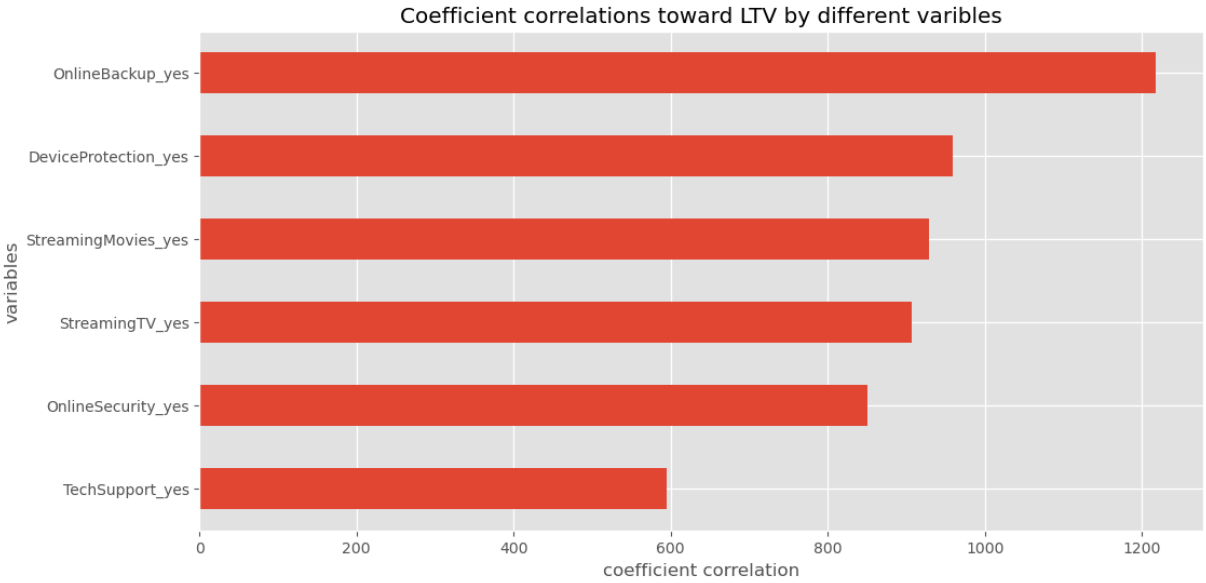
Out[132…
```
Dependents_No                  254.278405
Credit card (automatic)        579.368673
matBank transfer (autoic)      583.931304
PhoneService_Yes               662.564814
Partner_Yes                    717.118948
One year                      1897.245382
DSL                           2043.219839
Two year                      2840.605106
Fiber optic                   3412.487222
dtype: float64
```

> `Note`:

1. `gender` tidak memiliki signifikansi statistik dalam mempengaruhi LTV.
2. `Mitra` memiliki signifikansi statistik dalam mempengaruhi LTV, yang koefisien korelasinya adalah 717.
3. `Dependen` memiliki signifikansi statistik dalam mempengaruhi LTV, yang koefisien korelasinya adalah 254.
4. `Layanan Telepon` memiliki signifikansi statistik dalam mempengaruhi LTV, yang koefisien korelasinya adalah 662.
5. `Layanan Internet` memiliki signifikansi statistik dalam mempengaruhi LTV, untuk `DSL`, yang koefisien korelasinya adalah 2043. Dan, untuk `Fiber optic`, yang koefisien korelasinya adalah 3412.
6. `Kontrak` memiliki signifikansi statistik dalam mempengaruhi LTV, untuk `Satu tahun` yang koefisien korelasinya sebesar 1897. Dan untuk `Dua tahun` yang koefisien korelasinya sebesar 2840.
7. `Metode Pembayaran` memiliki signifikansi statistik dalam mempengaruhi LTV, untuk `transfer matBank` yang koefisien korelasinya adalah 583. Dan, untuk `Kartu kredit (otomatis)`, yang koefisien korelasinya adalah 579.

In [134…
```
# convert the categorical variables to 0,1
clean_df[['No','No internet service', 'OnlineSecurity_yes']] = pd.get_dummie
clean_df[['No','No internet service', 'OnlineBackup_yes']] = pd.get_dummies(
clean_df[['No','No internet service', 'DeviceProtection_yes']] = pd.get_dumm
clean_df[['No','No internet service', 'TechSupport_yes']] = pd.get_dummies(c
clean_df[['No','No internet service', 'StreamingTV_yes']] = pd.get_dummies(c
clean_df[['No','No internet service', 'StreamingMovies_yes']] = pd.get_dummi
```

In [136…
```
# use mutiple linear regression
# and visualize the order of the variables that affects LTV from high to low
clean_df['intercept'] = 1
lm = sm.OLS(clean_df['TotalCharges'], clean_df[['intercept', 'OnlineSecurity
                                                'TechSupport_yes', 'Streaming
                                                 'StreamingMovies_yes']].asty

results = lm.fit()
plt.figure(figsize = [12, 6])
results.params.sort_values()[1:].plot.barh();
desc(xlabel="coefficient correlation", ylabel='variables',title='Coefficient
```



Note: Pencadangan Online paling memengaruhi LTV, yang koefisien korelasinya 2 kali lebih besar dibandingkan dukungan teknis.

> # Step 4: Understand the limitations

Conclusion

Before drawing any conclusion, it is always better to inform the limitations. During the process of assessing data and conducting the exploratory data analysis, I have found some limitations:

`Batasan 1` : Dalam kumpulan data ini, kita hanya dapat melihat satu jenis dari setiap variabel, bukan situasi dunia nyata yang mengubah pilihan berbeda seiring berjalannya waktu, misalnya, di dunia nyata, orang mungkin ingin mencoba layanan streaming, namun mereka mungkin berubah pikiran untuk tinggalkan layanan bulan depan.

`Keterbatasan 2` : Kita tidak bisa hanya melihat variabel-variabel ini sebagai faktor keseluruhan untuk memahami alasan pasti mengapa pelanggan keluar karena mereka mungkin akan pergi karena harga yang lebih baik yang ditawarkan oleh pesaing atau kondisi perekonomian yang buruk dalam waktu tertentu, dll. Kita juga tidak bisa melihat kapan mereka bocor, jadi sulit untuk menyimpulkan situasi eksternal tersebut.

> # Step 5: Summaries

1. 80% (LTV rendah) pelanggan yang bocor hanya bertahan di bawah 10 bulan. Dan, rata-ratanya adalah 750 dolar. Di sisi lain, rata-rata LTV dari 20% teratas yang membocorkan adalah 4750 dolar. Dan rasio jumlah total LTV tiap grup adalah 750*4: 4750 = 1: 1,6, yang berarti kami harus fokus melayani 20% pelanggan dengan LTV tinggi, yang menghasilkan 60% (1,6/2,6) pendapatan kami dari pelanggan yang bocor.
2. 81% dari mereka yang memiliki LTV tinggi cenderung menggunakan saluran bekas. Tidak ada perbedaan besar antara mereka yang masih membayar layanan atau mereka yang berhenti berlangganan.
3. Mereka yang memiliki LTV tinggi senang menggunakan Fiber optic (75%-90%) dan DSL (10–20%), dan tidak ada satupun yang menggunakan layanan internet. Sedangkan yang memiliki LTV rendah, dari segi pelanggan saat ini, sekitar 30% di antaranya tidak menggunakan layanan internet.
4. Di antara 80% (LTV rendah) pelanggan saat ini, seluruh layanan internet digunakan secara merata oleh 40% orang. Dan di antara 80% (LTV rendah) pelanggan saat ini, streaming film dan streaming TV adalah dua bagian teratas dari layanan internet yang digunakan orang-orang, dan perlindungan perangkat serta pencadangan online berada di peringkat kedua, dan dukungan teknis serta keamanan online berada di peringkat ketiga, kesenjangan di antara keduanya mendekati 10% dari keseluruhan data.
5. Di antara 80% pelanggan saat ini, mereka yang LTV-nya lebih besar dari rata-rata LTV 80% pelanggan yang bocor memiliki kemungkinan 10% lebih besar untuk menggunakan banyak saluran dan layanan internet dibandingkan rata-rata 80% pelanggan saat ini.
6. 89% pelanggan yang membocorkan menggunakan kontrak bulanan, sedangkan pelanggan saat ini hanya 42% yang menggunakan kontrak bulanan.
7. Apakah masyarakat menggunakan layanan internet merupakan faktor terpenting dalam menciptakan LTV yang tinggi, dan kontrak tahunan adalah faktor kedua.
8. Di antara seluruh subset layanan internet, pencadangan online merupakan faktor terpenting dalam menciptakan LTV yang tinggi.

Step 6: Actionable insights

`Untuk mempertahankan pelanggan`:

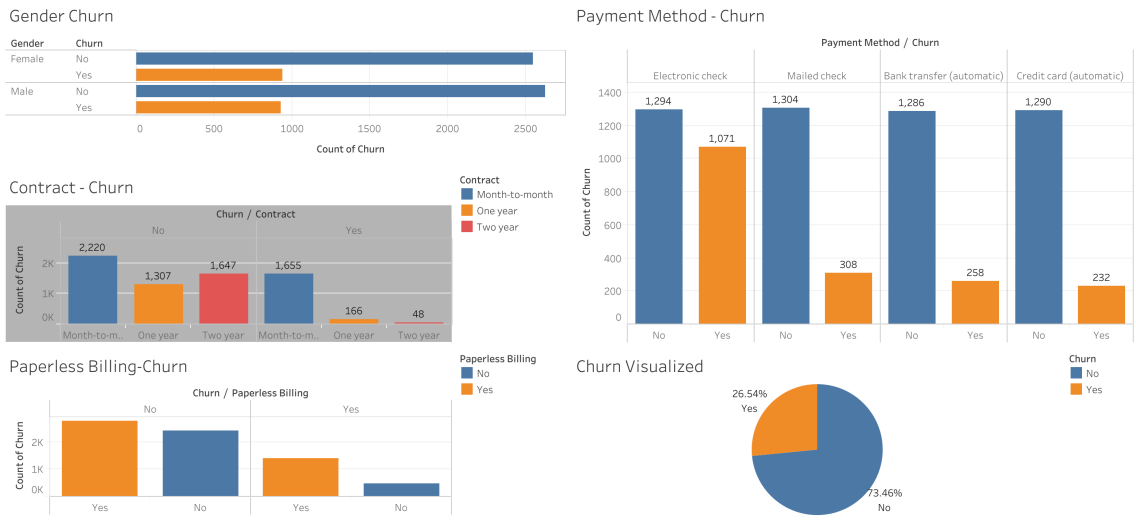
- Buat kampanye pemasaran untuk meningkatkan penjualan mereka yang saat ini berlangganan film streaming dan layanan TV di layanan internet kami yang lain. (Berdasarkan grafik dari Q5, semakin banyak layanan internet yang mereka bayar, semakin lama mereka cenderung menginap.)

`Untuk meningkatkan nilai seumur hidup pelanggan`:

- Tingkatkan anggaran pemasaran kami pada layanan streaming film dan TV karena kedua sektor ini memiliki kesenjangan terbesar antara yang menghasilkan 80% (LTV rendah) dan 20% (LTV tinggi). (Berdasarkan grafik dari Q5)
- Tingkatkan anggaran pemasaran kami untuk pencadangan online, yang merupakan faktor terpenting yang berkontribusi terhadap LTV tinggi. (Berdasarkan grafik dari Q8)
- Meningkatkan anggaran pemasaran kami bagi mereka yang ingin menggunakan banyak jalur. (Berdasarkan grafik dari Q6)

Dashboard

(https://public.tableau.com/app/profile/burhanudin.badiuzaman/viz/DataAnalysisProjectTelcc
publish=yes)



In [138…
```
# Convert notebook to pdf

!jupyter nbconvert --to webpdf --allow-chromium-download TelcoCustomerChurn.
```

```
[NbConvertApp] Converting notebook TelcoCustomerChurn.ipynb to webpdf
[NbConvertApp] WARNING | Alternative text is missing on 20 image(s).
[NbConvertApp] Building PDF
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 1175473 bytes to TelcoCustomerChurn.pdf
```

In [ ]: