```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

data =
pd.read_csv("C:/Users/ANUNG/OneDrive/Documents/KULIAH/ADS/telecom_user
_churn.csv", sep=";")
data.head()
```

```
  Customer ID  Gender  Age Married  Number of Dependents          City
\
0  0002-ORFBO  Female   37     Yes                     0  Frazier Park

1  0003-MKNFE    Male   46      No                     0      Glendale

2  0004-TLHLJ    Male   50      No                     0    Costa Mesa

3  0011-IGKFF    Male   78     Yes                     0      Martinez

4  0013-EXCHZ  Female   75     Yes                     0     Camarillo


   Zip Code    Latitude     Longitude  Number of Referrals  ...  \
0     93225  34.827.662  -118.999.073                    2  ...
1     91206  34.162.515  -118.203.869                    0  ...
2     92627  33.645.672  -117.922.613                    0  ...
3     94553  38.014.457  -122.115.432                    1  ...
4     93010  34.227.846  -119.079.903                    3  ...

    Payment Method Monthly Charge Total Charges  Total Refunds  \
0      Credit Card           65.6        593.30           0.00
1      Credit Card           -4.0        542.40          38.33
2  Bank Withdrawal           73.9        280.85           0.00
3  Bank Withdrawal           98.0       1237.85           0.00
4      Credit Card           83.9        267.40           0.00

  Total Extra Data Charges Total Long Distance Charges Total
Revenue  \
0                        0                      381.51        974.81

1                       10                       96.21        610.28

2                        0                      134.60        415.45

3                        0                      361.66       1599.51

4                        0                       22.14        289.54


   Customer Status   Churn Category                    Churn Reason
```

```
0           Stayed              NaN                                NaN
1           Stayed              NaN                                NaN
2          Churned       Competitor  Competitor had better devices
3          Churned  Dissatisfaction       Product dissatisfaction
4          Churned  Dissatisfaction           Network reliability

[5 rows x 38 columns]
```

Cleaning

```
#CLEANING DATA
data.dtypes
```

```
Customer ID                         object
Gender                              object
Age                                  int64
Married                             object
Number of Dependents                 int64
City                                object
Zip Code                             int64
Latitude                            object
Longitude                           object
Number of Referrals                  int64
Tenure in Months                     int64
Offer                               object
Phone Service                       object
Avg Monthly Long Distance Charges  float64
Multiple Lines                      object
Internet Service                    object
Internet Type                       object
Avg Monthly GB Download            float64
Online Security                     object
Online Backup                       object
Device Protection Plan              object
Premium Tech Support                object
Streaming TV                        object
Streaming Movies                    object
Streaming Music                     object
Unlimited Data                      object
Contract                            object
Paperless Billing                   object
Payment Method                      object
Monthly Charge                     float64
Total Charges                      float64
Total Refunds                      float64
Total Extra Data Charges             int64
Total Long Distance Charges        float64
Total Revenue                      float64
Customer Status                     object
```

```
Churn Category                       object
Churn Reason                         object
dtype: object
```

```
#missing value
data.isna().sum()
```

```
Customer ID                            0
Gender                                 0
Age                                    0
Married                                0
Number of Dependents                   0
City                                   0
Zip Code                               0
Latitude                               0
Longitude                              0
Number of Referrals                    0
Tenure in Months                       0
Offer                               3877
Phone Service                          0
Avg Monthly Long Distance Charges    682
Multiple Lines                       682
Internet Service                       0
Internet Type                       1526
Avg Monthly GB Download             1526
Online Security                     1526
Online Backup                       1526
Device Protection Plan              1526
Premium Tech Support                1526
Streaming TV                        1526
Streaming Movies                    1526
Streaming Music                     1526
Unlimited Data                      1526
Contract                               0
Paperless Billing                      0
Payment Method                         0
Monthly Charge                         0
Total Charges                          0
Total Refunds                          0
Total Extra Data Charges               0
Total Long Distance Charges            0
Total Revenue                          0
Customer Status                        0
Churn Category                      5174
Churn Reason                        5174
dtype: int64
```

```
#NA pada columns offer
data['Offer'].fillna('None',inplace = True)
data['Offer'].isna().sum()
```

```
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\3628354784.py:2:
FutureWarning: A value is trying to be set on a copy of a DataFrame or
Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never
work because the intermediate object on which we are setting values
always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try
using 'df.method({col: value}, inplace=True)' or df[col] =
df[col].method(value) instead, to perform the operation inplace on the
original object.


  data['Offer'].fillna('None',inplace = True)

0

#Missing value pada Avg dan Multiple line
data['Avg Monthly Long Distance Charges'].fillna(0,inplace = True)
data['Multiple Lines'].fillna('No', inplace = True)
data[['Avg Monthly Long Distance Charges', 'Multiple
Lines']].isna().sum()

C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\2998822864.py:2:
FutureWarning: A value is trying to be set on a copy of a DataFrame or
Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never
work because the intermediate object on which we are setting values
always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try
using 'df.method({col: value}, inplace=True)' or df[col] =
df[col].method(value) instead, to perform the operation inplace on the
original object.


  data['Avg Monthly Long Distance Charges'].fillna(0,inplace = True)
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\2998822864.py:3:
FutureWarning: A value is trying to be set on a copy of a DataFrame or
Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never
work because the intermediate object on which we are setting values
always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try
using 'df.method({col: value}, inplace=True)' or df[col] =
df[col].method(value) instead, to perform the operation inplace on the
original object.


  data['Multiple Lines'].fillna('No', inplace = True)
```

```
Avg Monthly Long Distance Charges    0
Multiple Lines                       0
dtype: int64
```

```
#Missing Value pada 'Internet Type', 'Avg Monthly GB Download','Online
Security', 'Online Backup', 'Device Protection Plan','Premium Tech
Support', 'Streaming TV', 'Streaming Movies','Streaming Music',
'Unlimited Data'
missing = ['Online Security', 'Online Backup', 'Device Protection
Plan','Premium Tech Support',
           'Streaming TV', 'Streaming Movies','Streaming Music',
'Unlimited Data']
for i in missing:
    data[i].fillna('No',inplace = True)
```

```
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\2210962076.py:5:
FutureWarning: A value is trying to be set on a copy of a DataFrame or
Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never
work because the intermediate object on which we are setting values
always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try
using 'df.method({col: value}, inplace=True)' or df[col] =
df[col].method(value) instead, to perform the operation inplace on the
original object.
```

```
  data[i].fillna('No',inplace = True)
```

```
data['Internet Type'].fillna('None',inplace = True)
data['Avg Monthly GB Download'].fillna(0,inplace = True)
```

```
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\1064503569.py:1:
FutureWarning: A value is trying to be set on a copy of a DataFrame or
Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never
work because the intermediate object on which we are setting values
always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try
using 'df.method({col: value}, inplace=True)' or df[col] =
df[col].method(value) instead, to perform the operation inplace on the
original object.
```

```
  data['Internet Type'].fillna('None',inplace = True)
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\1064503569.py:2:
FutureWarning: A value is trying to be set on a copy of a DataFrame or
Series through chained assignment using an inplace method.
```

```
The behavior will change in pandas 3.0. This inplace method will never
work because the intermediate object on which we are setting values
always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try
using 'df.method({col: value}, inplace=True)' or df[col] =
df[col].method(value) instead, to perform the operation inplace on the
original object.


  data['Avg Monthly GB Download'].fillna(0,inplace = True)
```

```
#Missing pada Churn category dan Churn Reason
data['Churn Category'].fillna('-',inplace = True)
data['Churn Reason'].fillna('-',inplace = True)
```

```
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\18591545.py:2:
FutureWarning: A value is trying to be set on a copy of a DataFrame or
Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never
work because the intermediate object on which we are setting values
always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try
using 'df.method({col: value}, inplace=True)' or df[col] =
df[col].method(value) instead, to perform the operation inplace on the
original object.


  data['Churn Category'].fillna('-',inplace = True)
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\18591545.py:3:
FutureWarning: A value is trying to be set on a copy of a DataFrame or
Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never
work because the intermediate object on which we are setting values
always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try
using 'df.method({col: value}, inplace=True)' or df[col] =
df[col].method(value) instead, to perform the operation inplace on the
original object.


  data['Churn Reason'].fillna('-',inplace = True)
```

```
#Cek duplicate
data['Customer ID'].duplicated().sum()
```

```
0
```

```
data.columns
```

```
Index(['Customer ID', 'Gender', 'Age', 'Married', 'Number of
Dependents',
       'City', 'Zip Code', 'Latitude', 'Longitude', 'Number of
Referrals',
       'Tenure in Months', 'Offer', 'Phone Service',
       'Avg Monthly Long Distance Charges', 'Multiple Lines',
       'Internet Service', 'Internet Type', 'Avg Monthly GB Download',
       'Online Security', 'Online Backup', 'Device Protection Plan',
       'Premium Tech Support', 'Streaming TV', 'Streaming Movies',
       'Streaming Music', 'Unlimited Data', 'Contract', 'Paperless
Billing',
       'Payment Method', 'Monthly Charge', 'Total Charges', 'Total
Refunds',
       'Total Extra Data Charges', 'Total Long Distance Charges',
       'Total Revenue', 'Customer Status', 'Churn Category', 'Churn
Reason'],
      dtype='object')
```

```
#Unique dari data kategorik
unique = ['Gender','Married','City','Offer', 'Phone Service','Internet
Service', 'Internet Type','Online Security', 'Online Backup', 'Device
Protection Plan',
       'Premium Tech Support', 'Streaming TV', 'Streaming Movies',
       'Streaming Music', 'Unlimited Data','Contract', 'Paperless
Billing',
       'Payment Method','Customer Status']

for i in unique:
    print(i, ':', data[i].unique(),'\n')
```

```
Gender : ['Female' 'Male']

Married : ['Yes' 'No']

City : ['Frazier Park' 'Glendale' 'Costa Mesa' ... 'Jacumba'
'Carpinteria'
 'Meadow Valley']

Offer : ['None' 'Offer E' 'Offer D' 'Offer A' 'Offer B' 'Offer C']

Phone Service : ['Yes' 'No']

Internet Service : ['Yes' 'No']

Internet Type : ['Cable' 'Fiber Optic' 'DSL' 'None']

Online Security : ['No' 'Yes']

Online Backup : ['Yes' 'No']

Device Protection Plan : ['No' 'Yes']
```

```
Premium Tech Support : ['Yes' 'No']

Streaming TV : ['Yes' 'No']

Streaming Movies : ['No' 'Yes']

Streaming Music : ['No' 'Yes']

Unlimited Data : ['Yes' 'No']

Contract : ['One Year' 'Month-to-Month' 'Two Year']

Paperless Billing : ['Yes' 'No']

Payment Method : ['Credit Card' 'Bank Withdrawal' 'Mailed Check']

Customer Status : ['Stayed' 'Churned' 'Joined']
```

```
data = data[data['Monthly Charge'] >= 0]

data
```

```
     Customer ID  Gender  Age Married  Number of Dependents
City  \
0     0002-ORFBO  Female   37     Yes                     0  Frazier
Park
2     0004-TLHLJ    Male   50      No                     0    Costa
Mesa
3     0011-IGKFF    Male   78     Yes                     0
Martinez
4     0013-EXCHZ  Female   75     Yes                     0
Camarillo
5     0013-MHZWF  Female   23      No                     3
Midpines
...          ...     ...  ...     ...                   ...
...
7038  9987-LUTYD  Female   20      No                     0       La
Mesa
7039  9992-RRAMN    Male   40     Yes                     0
Riverbank
7040  9992-UJOEL    Male   22      No                     0
Elk
7041  9993-LHIEB    Male   21     Yes                     0  Solana
Beach
7042  9995-HOTOH    Male   36     Yes                     0   Sierra
City

      Zip Code    Latitude     Longitude  Number of Referrals  ...  \
0        93225  34.827.662  -118.999.073                    2  ...
```

```
2        92627  33.645.672  -117.922.613                    0  ...
3        94553  38.014.457  -122.115.432                    1  ...
4        93010  34.227.846  -119.079.903                    3  ...
5        95345  37.581.496  -119.972.762                    0  ...
...        ...         ...           ...                  ...  ...
7038     91941  32.759.327   -11.699.726                    0  ...
7039     95367  37.734.971  -120.954.271                    1  ...
7040     95432  39.108.252  -123.645.121                    0  ...
7041     92075  33.001.813  -117.263.628                    5  ...
7042     96125  39.600.599  -120.636.358                    1  ...

       Payment Method Monthly Charge Total Charges  Total Refunds  \
0         Credit Card          65.60        593.30            0.0
2     Bank Withdrawal          73.90        280.85            0.0
3     Bank Withdrawal          98.00       1237.85            0.0
4         Credit Card          83.90        267.40            0.0
5         Credit Card          69.40        571.45            0.0
...               ...            ...           ...            ...
7038      Credit Card          55.15        742.90            0.0
7039  Bank Withdrawal          85.10       1873.70            0.0
7040      Credit Card          50.30         92.75            0.0
7041      Credit Card          67.85       4627.65            0.0
7042  Bank Withdrawal          59.00       3707.60            0.0

     Total Extra Data Charges Total Long Distance Charges Total
Revenue  \
0                           0                      381.51
974.81
2                           0                      134.60
415.45
3                           0                      361.66
1599.51
4                           0                       22.14
289.54
5                           0                      150.93
722.38
...                       ...                         ...            ..
.
7038                        0                      606.84
1349.74
7039                        0                      356.40
2230.10
7040                        0                       37.24
129.99
7041                        0                      142.04
4769.69
7042                        0                        0.00
3707.60

      Customer Status   Churn Category                    Churn Reason
```

```
0               Stayed                -                                  -
2              Churned       Competitor  Competitor had better devices
3              Churned  Dissatisfaction        Product dissatisfaction
4              Churned  Dissatisfaction            Network reliability
5               Stayed                -                                  -
...                ...              ...                                ...
7038            Stayed                -                                  -
7039           Churned  Dissatisfaction        Product dissatisfaction
7040            Joined                -                                  -
7041            Stayed                -                                  -
7042            Stayed                -                                  -

[6923 rows x 38 columns]
```

```
# Misalkan 'data_cleaned' adalah DataFrame yang ingin disimpan
data.to_csv("Data_Telecom.csv", index=False)
```

```
data.describe()
```

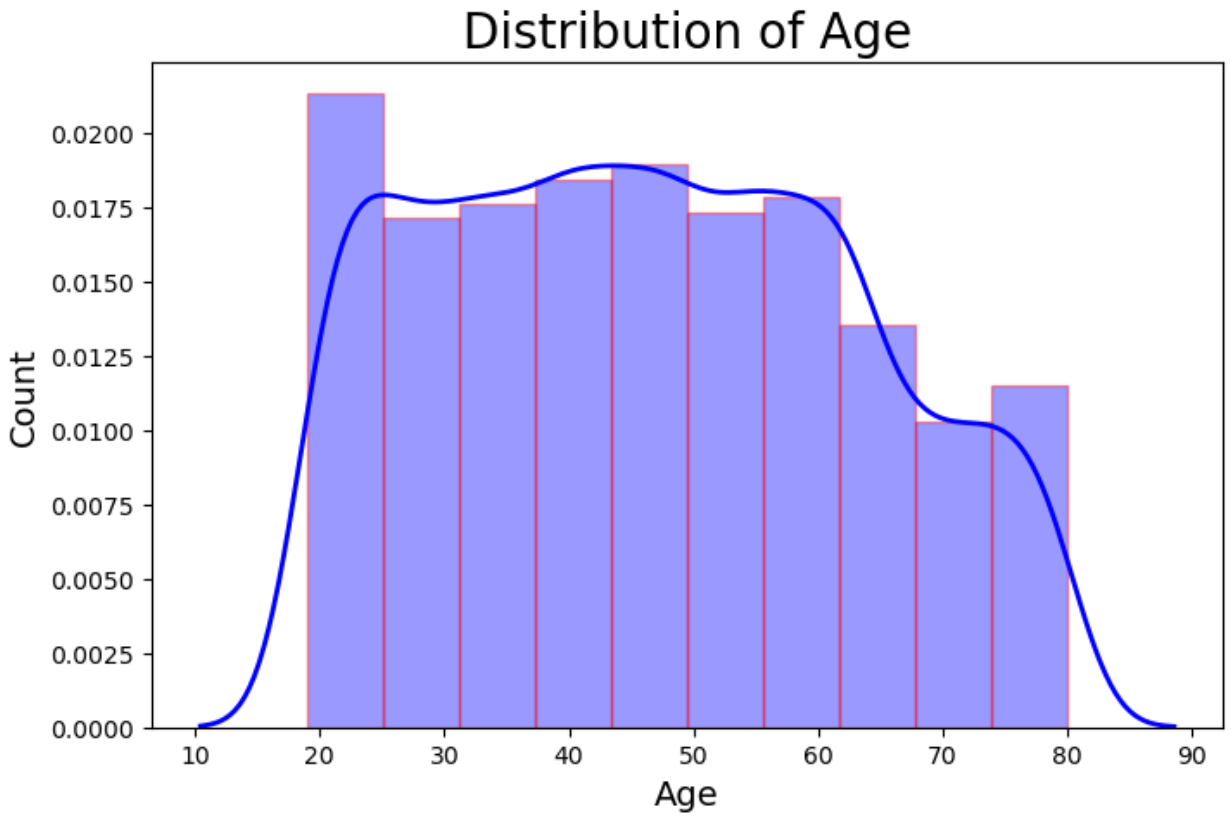
```
                Age  Number of Dependents      Zip Code  Number of
Referrals  \
count  6923.000000           6923.000000   6923.000000
6923.000000
mean     46.488661              0.468583  93492.072512
1.950744
std      16.744592              0.962000   1855.816719
2.997548
min      19.000000              0.000000  90001.000000
0.000000
25%      32.000000              0.000000  92104.000000
0.000000
50%      46.000000              0.000000  93522.000000
0.000000
75%      60.000000              0.000000  95333.000000
3.000000
max      80.000000              9.000000  96150.000000
11.000000

       Tenure in Months  Avg Monthly Long Distance Charges  \
```

```
count       6923.000000                          6923.000000
mean          32.432183                            22.952809
std           24.539612                            15.453558
min            1.000000                             0.000000
25%            9.000000                             9.190000
50%           29.000000                            22.890000
75%           55.000000                            36.390000
max           72.000000                            49.990000

       Avg Monthly GB Download  Monthly Charge  Total Charges  Total
Refunds  \
count              6923.000000     6923.000000    6923.000000
6923.000000
mean                 20.517695       64.792366    2284.741079
1.941131
std                  20.424725       30.107716    2268.562423
7.863733
min                   0.000000       18.250000      18.800000
0.000000
25%                   3.000000       35.500000     401.200000
0.000000
50%                  17.000000       70.350000    1396.900000
0.000000
75%                  27.000000       89.900000    3794.975000
0.000000
max                  85.000000      118.750000    8684.800000
49.790000

       Total Extra Data Charges  Total Long Distance Charges  Total
Revenue
count               6923.000000                  6923.000000
6923.000000
mean                   6.871299                   750.030946
3039.702193
std                   25.123884                   847.440416
2867.566650
min                    0.000000                     0.000000
21.360000
25%                    0.000000                    70.810000
608.850000
50%                    0.000000                   401.100000
2109.870000
75%                    0.000000                  1195.785000
4813.445000
max                  150.000000                  3564.720000
11979.340000
```

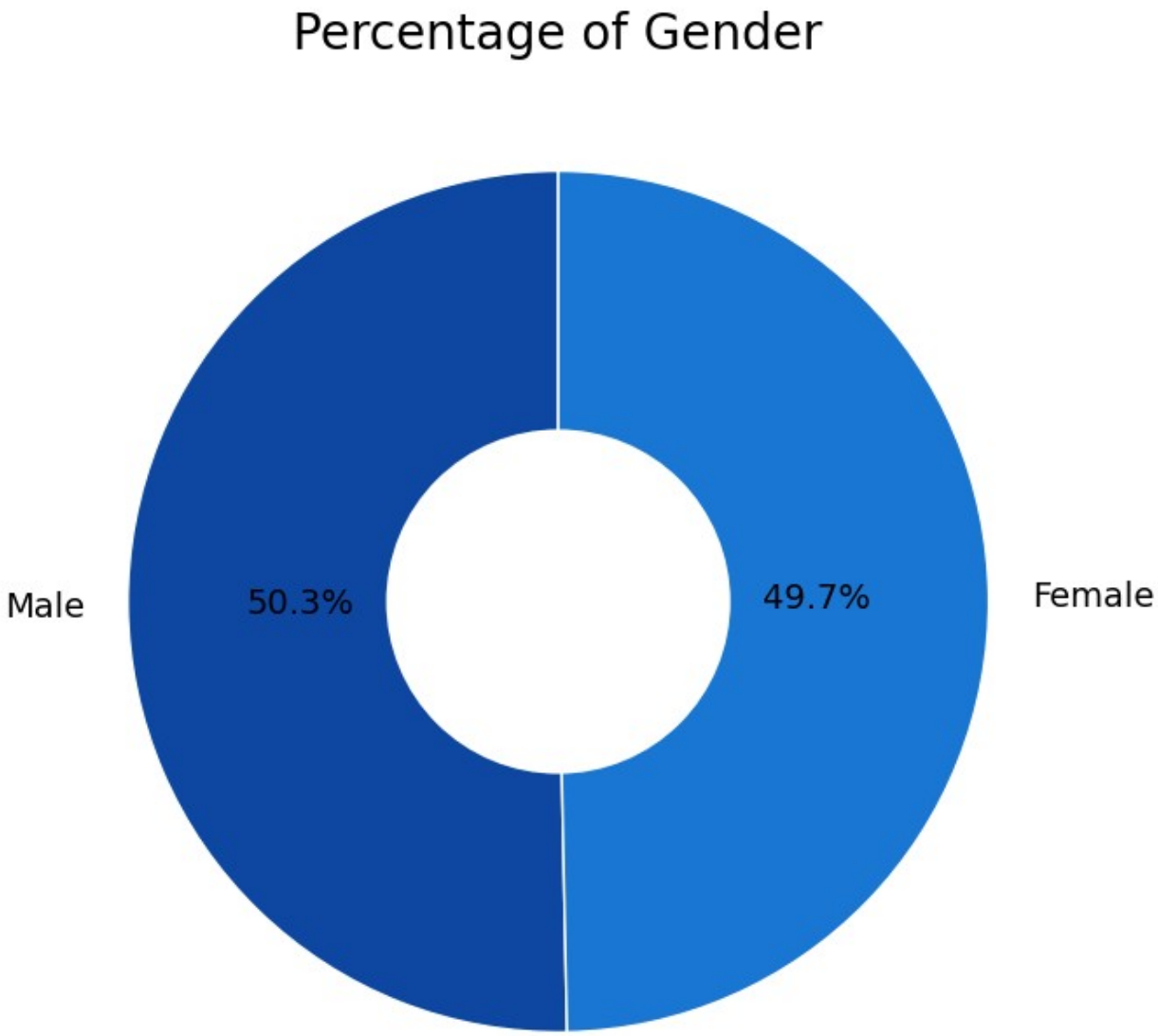
EDA

```
#DISTRIBUSI DARI UMUR

plt.figure(figsize=(8, 5))
sns.distplot(data['Age'],bins = 10, hist=True,kde=True,
color='blue', hist_kws={'edgecolor':'red' },
kde_kws={'linewidth':2})
plt.title('Distribution of Age', fontsize=20)
plt.xlabel('Age', fontsize=14)
plt.ylabel('Count', fontsize=14)
plt.show()
```

```
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\2140319610.py:4:
UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `histplot` (an axes-level function for
histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(data['Age'],bins = 10, hist=True,kde=True,
```

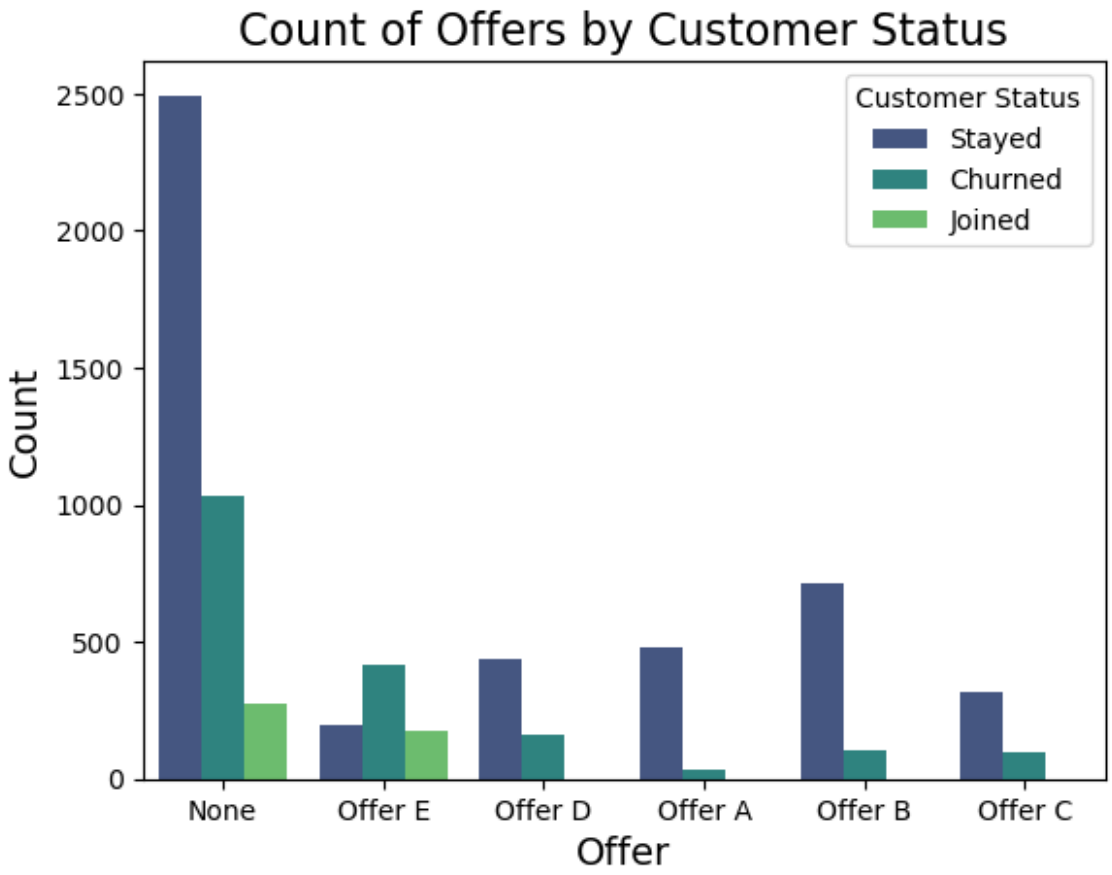



```
#proporsi dari gender
pie_gender = data['Gender'].value_counts()
plt.figure(figsize=(8, 8))
plt.pie(pie_gender.values, labels= pie_gender.index,
colors=['#0D47A1', '#1976D2'], wedgeprops={'edgecolor': 'white'},
                startangle=90,autopct='%1.1f%%',radius = 1)
centre_circle = plt.Circle((0, 0), 0.40, fc='white')
plt.setp(plt.gca().texts, size=14)  # Mengatur ukuran label
plt.setp(plt.gca().patches, edgecolor='w')  # Mengatur warna tepi pada
pie

plt.gca().add_artist(centre_circle)

# Memberikan judul
plt.title('Percentage of Gender', fontsize=20)
plt.show()
```




```
#apakah offer mempengaruhi churn
sns.countplot(data=data, x='Offer', hue='Customer Status',
palette='viridis')
plt.title('Count of Offers by Customer Status', fontsize=16)
plt.xlabel('Offer', fontsize=14)
plt.ylabel('Count', fontsize=14)
plt.legend(title='Customer Status')
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```

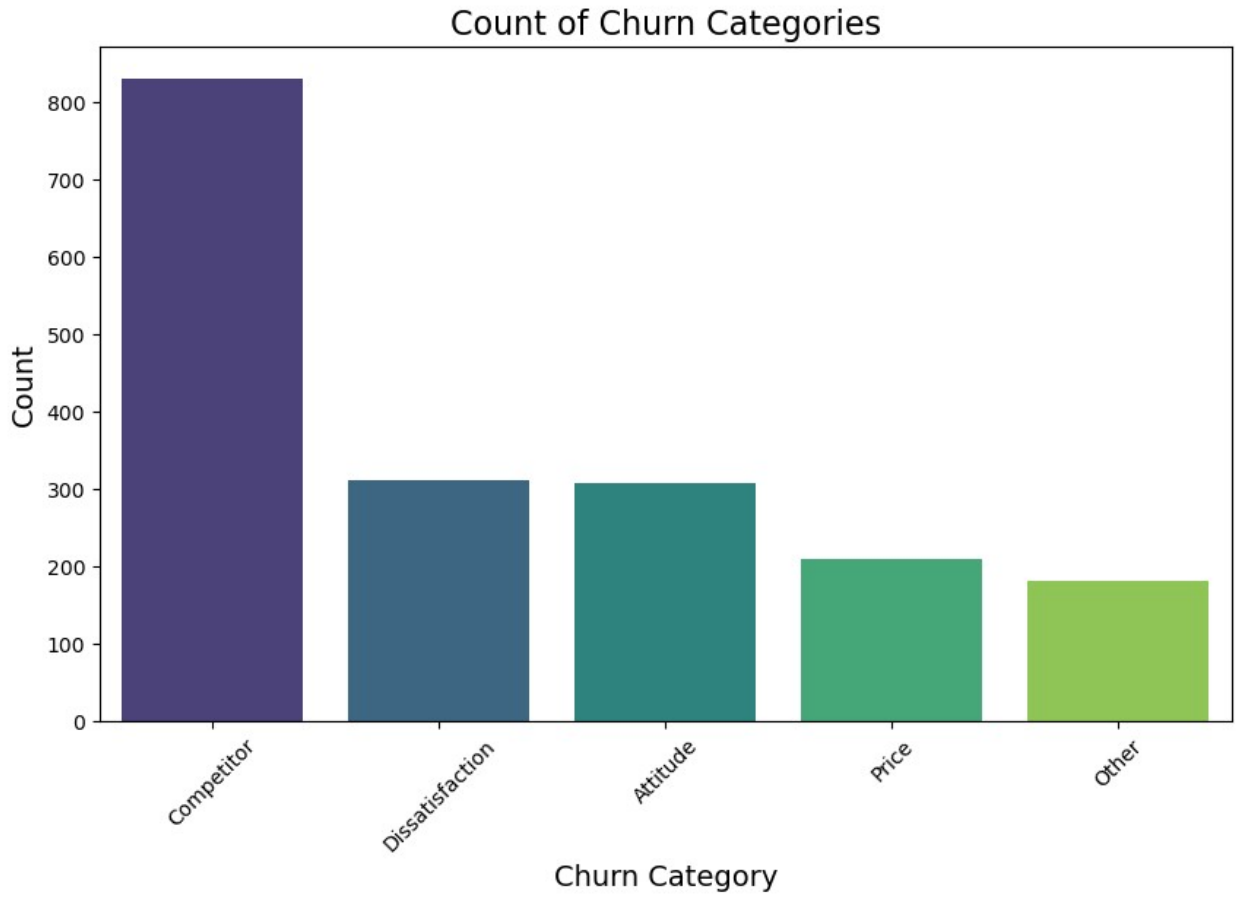



```
#alasan mengapa orang pada keluar
category = data['Churn Category'].value_counts()
category_df = category.reset_index(name = 'Count')
category_df.columns = ['Churn Category', 'Count']
df_cleaned = category_df[category_df['Churn Category'] != '-']
df_cleaned
```

	Churn Category	Count
1	Competitor	831
2	Dissatisfaction	312
3	Attitude	307
4	Price	209
5	Other	180

```
plt.figure(figsize=(10, 6))
sns.barplot(data=df_cleaned, x='Churn Category', y='Count',
palette='viridis')
plt.title('Count of Churn Categories', fontsize=16)
plt.xlabel('Churn Category', fontsize=14)
plt.ylabel('Count', fontsize=14)
plt.xticks(rotation=45)
```

```
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\1598762817.py:2:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(data=df_cleaned, x='Churn Category', y='Count',
palette='viridis')

([0, 1, 2, 3, 4],
 [Text(0, 0, 'Competitor'),
  Text(1, 0, 'Dissatisfaction'),
  Text(2, 0, 'Attitude'),
  Text(3, 0, 'Price'),
  Text(4, 0, 'Other')])
```




```
#Decision Tree
from sklearn.tree import DecisionTreeClassifier,export_graphviz
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report,
```

```
confusion_matrix
from sklearn.tree import plot_tree

df_tree = data[['Married', 'Number of Dependents',
         'Latitude', 'Number of Referrals',
       'Tenure in Months', 'Offer', 'Phone Service',
       'Avg Monthly Long Distance Charges', 'Multiple Lines',
       'Internet Service', 'Internet Type', 'Avg Monthly GB Download',
       'Online Security', 'Online Backup', 'Device Protection Plan',
       'Premium Tech Support', 'Streaming TV', 'Streaming Movies',
       'Streaming Music', 'Unlimited Data', 'Contract', 'Paperless
Billing',
       'Payment Method', 'Monthly Charge', 'Total Charges', 'Total
Refunds',
       'Total Extra Data Charges', 'Total Long Distance Charges',
       'Total Revenue', 'Customer Status']]

from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
for column in df_tree.select_dtypes(include=[ 'object',
'category']).columns:
    df_tree[column] = label_encoder.fit_transform(df_tree[column])

C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\804447711.py:4:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_tree[column] = label_encoder.fit_transform(df_tree[column])
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\804447711.py:4:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_tree[column] = label_encoder.fit_transform(df_tree[column])
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\804447711.py:4:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_tree[column] = label_encoder.fit_transform(df_tree[column])
```

```
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\804447711.py:4:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_tree[column] = label_encoder.fit_transform(df_tree[column])
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\804447711.py:4:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_tree[column] = label_encoder.fit_transform(df_tree[column])
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\804447711.py:4:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_tree[column] = label_encoder.fit_transform(df_tree[column])
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\804447711.py:4:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_tree[column] = label_encoder.fit_transform(df_tree[column])
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\804447711.py:4:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_tree[column] = label_encoder.fit_transform(df_tree[column])
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\804447711.py:4:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_tree[column] = label_encoder.fit_transform(df_tree[column])
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\804447711.py:4:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_tree[column] = label_encoder.fit_transform(df_tree[column])
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\804447711.py:4:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_tree[column] = label_encoder.fit_transform(df_tree[column])
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\804447711.py:4:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_tree[column] = label_encoder.fit_transform(df_tree[column])
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\804447711.py:4:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_tree[column] = label_encoder.fit_transform(df_tree[column])
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\804447711.py:4:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_tree[column] = label_encoder.fit_transform(df_tree[column])
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\804447711.py:4:
```

```
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_tree[column] = label_encoder.fit_transform(df_tree[column])
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\804447711.py:4:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_tree[column] = label_encoder.fit_transform(df_tree[column])
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\804447711.py:4:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_tree[column] = label_encoder.fit_transform(df_tree[column])
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\804447711.py:4:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_tree[column] = label_encoder.fit_transform(df_tree[column])
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\804447711.py:4:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_tree[column] = label_encoder.fit_transform(df_tree[column])
```

```
df_tree.columns
```

```
Index(['Married', 'Number of Dependents', 'Latitude', 'Number of
Referrals',
       'Tenure in Months', 'Offer', 'Phone Service',
```

```
       'Avg Monthly Long Distance Charges', 'Multiple Lines',
       'Internet Service', 'Internet Type', 'Avg Monthly GB Download',
       'Online Security', 'Online Backup', 'Device Protection Plan',
       'Premium Tech Support', 'Streaming TV', 'Streaming Movies',
       'Streaming Music', 'Unlimited Data', 'Contract', 'Paperless
Billing',
       'Payment Method', 'Monthly Charge', 'Total Charges', 'Total
Refunds',
       'Total Extra Data Charges', 'Total Long Distance Charges',
       'Total Revenue', 'Customer Status'],
      dtype='object')
```

```
X = df_tree.drop('Customer Status', axis=1)
y = df_tree['Customer Status']
```

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.20, random_state = 42)
```

```
X_train.shape, X_test.shape
```

```
((5538, 29), (1385, 29))
```

```
clf_gini = DecisionTreeClassifier(criterion='gini', random_state=0,
max_depth=3)
```

```
# fit the model
clf_gini.fit(X_train, y_train)
```

```
DecisionTreeClassifier(max_depth=3, random_state=0)
```

```
y_pred_gini = clf_gini.predict(X_test)
```

```
from sklearn.metrics import accuracy_score
```

```
print('Model accuracy score with criterion gini index: {0:0.4f}'.
format(accuracy_score(y_test, y_pred_gini)))
```

```
Model accuracy score with criterion gini index: 0.7971
```

```
print('Training set score: {:.4f}'.format(clf_gini.score(X_train,
y_train)))
```

```
print('Test set score: {:.4f}'.format(clf_gini.score(X_test, y_test)))
```

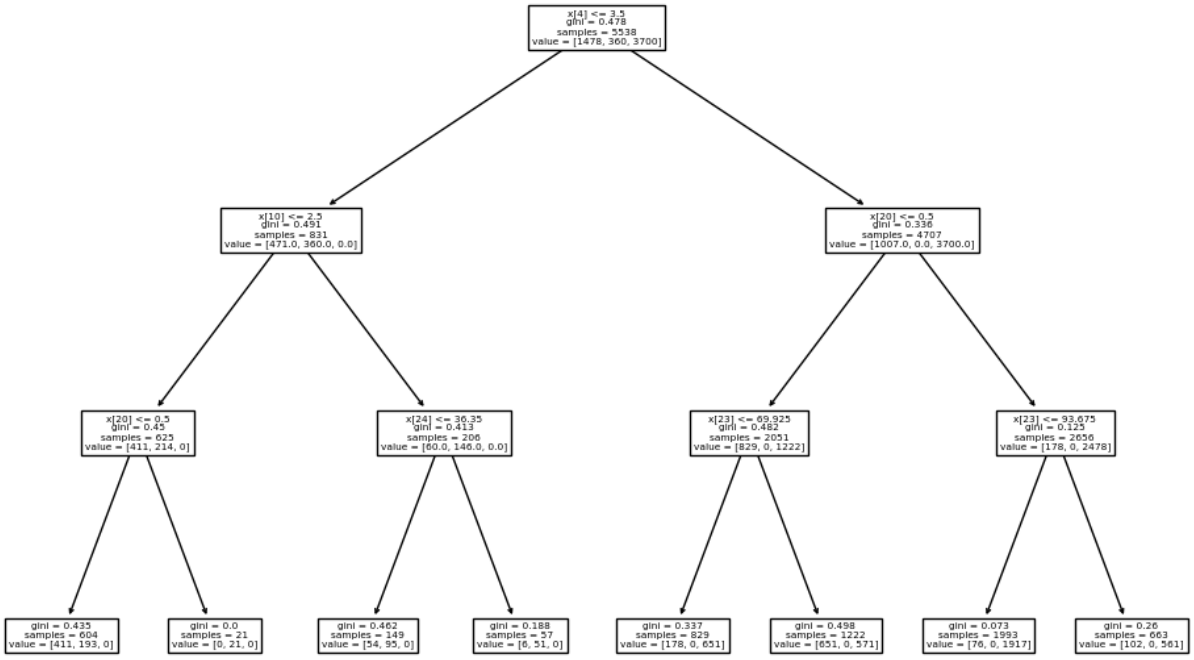
```
Training set score: 0.7869
Test set score: 0.7971
```

```
class_name = data['Customer Status'].unique
```

```
plt.figure(figsize=(12,8))
```

```
from sklearn import tree

tree.plot_tree(clf_gini.fit(X_train, y_train))
```

```
[Text(0.5, 0.875, 'x[4] <= 3.5\ngini = 0.478\nsamples = 5538\nvalue =
[1478, 360, 3700]'),
 Text(0.25, 0.625, 'x[10] <= 2.5\ngini = 0.491\nsamples = 831\nvalue =
[471.0, 360.0, 0.0]'),
 Text(0.125, 0.375, 'x[20] <= 0.5\ngini = 0.45\nsamples = 625\nvalue =
[411, 214, 0]'),
 Text(0.0625, 0.125, 'gini = 0.435\nsamples = 604\nvalue = [411, 193,
0]'),
 Text(0.1875, 0.125, 'gini = 0.0\nsamples = 21\nvalue = [0, 21, 0]'),
 Text(0.375, 0.375, 'x[24] <= 36.35\ngini = 0.413\nsamples = 206\
nvalue = [60.0, 146.0, 0.0]'),
 Text(0.3125, 0.125, 'gini = 0.462\nsamples = 149\nvalue = [54, 95,
0]'),
 Text(0.4375, 0.125, 'gini = 0.188\nsamples = 57\nvalue = [6, 51,
0]'),
 Text(0.75, 0.625, 'x[20] <= 0.5\ngini = 0.336\nsamples = 4707\nvalue
= [1007.0, 0.0, 3700.0]'),
 Text(0.625, 0.375, 'x[23] <= 69.925\ngini = 0.482\nsamples = 2051\
nvalue = [829, 0, 1222]'),
 Text(0.5625, 0.125, 'gini = 0.337\nsamples = 829\nvalue = [178, 0,
651]'),
 Text(0.6875, 0.125, 'gini = 0.498\nsamples = 1222\nvalue = [651, 0,
571]'),
 Text(0.875, 0.375, 'x[23] <= 93.675\ngini = 0.125\nsamples = 2656\
nvalue = [178, 0, 2478]'),
 Text(0.8125, 0.125, 'gini = 0.073\nsamples = 1993\nvalue = [76, 0,
1917]'),
 Text(0.9375, 0.125, 'gini = 0.26\nsamples = 663\nvalue = [102, 0,
561]')]
```




```
#Top ten city use Internet Service
city_df = data.groupby('City')['Internet Service'].value_counts()
citycount = city_df.reset_index()
citycount
```

```
              City Internet Service  count
0           Acampo              Yes      4
1            Acton              Yes      4
2         Adelanto              Yes      3
3         Adelanto               No      2
4             Adin              Yes      3
...            ...              ...    ...
1846     Yuba City               No      1
1847       Yucaipa              Yes      4
1848  Yucca Valley              Yes      5
1849         Zenia               No      2
1850         Zenia              Yes      2

[1851 rows x 3 columns]
```

```
df_city = citycount[citycount['Internet Service'] != 'No']
df_city
```

```
              City Internet Service  count
0           Acampo              Yes      4
```

```
1            Acton              Yes      4
2         Adelanto              Yes      3
4             Adin              Yes      3
6     Agoura Hills              Yes      4
...            ...              ...    ...
1844         Yreka              Yes      4
1845     Yuba City              Yes      7
1847       Yucaipa              Yes      4
1848  Yucca Valley              Yes      5
1850         Zenia              Yes      2

[1104 rows x 3 columns]
```

```
df_city.sort_values(by = 'count',ascending= False, inplace = True)
ten_city = df_city[:10]
ten_city
```

```
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\3279581647.py:1:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_city.sort_values(by = 'count',ascending= False, inplace = True)
```

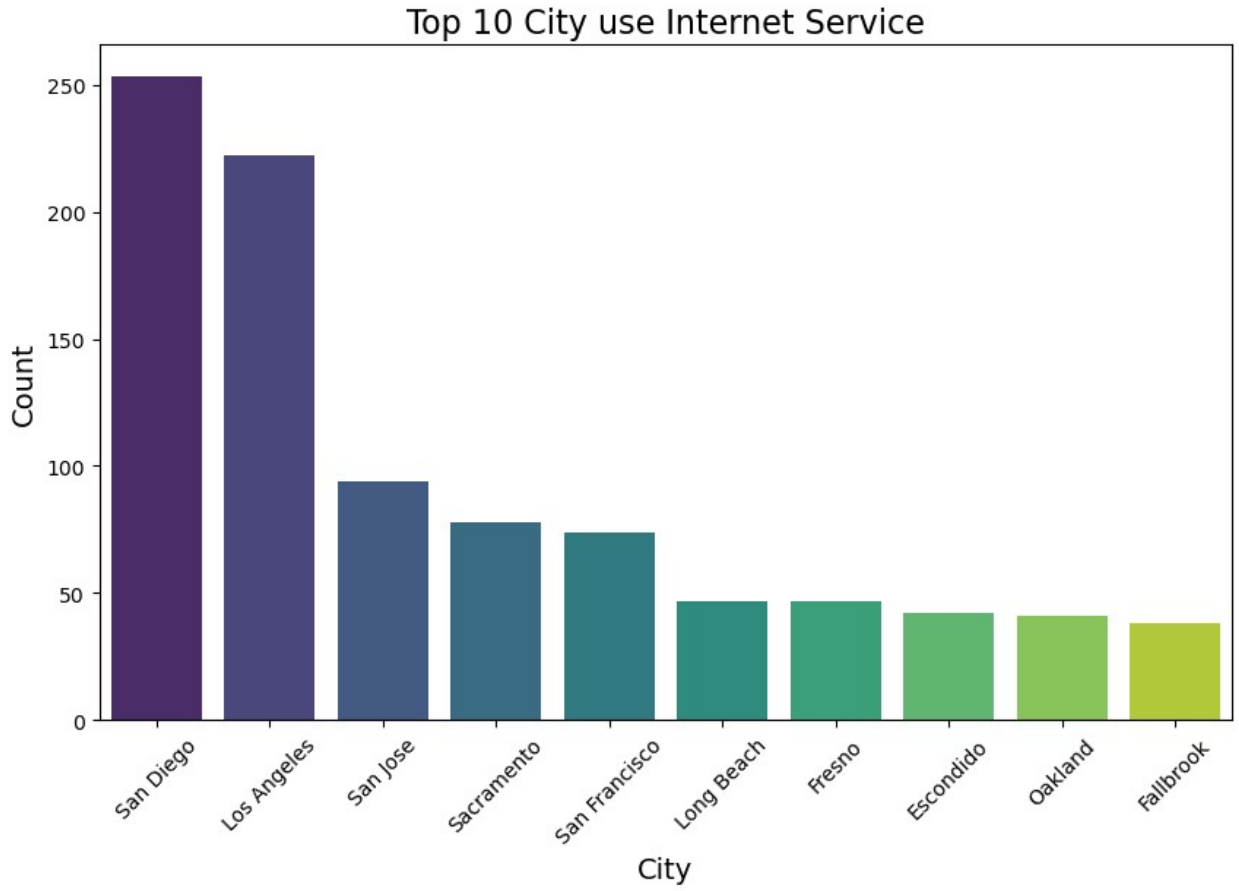
```
               City Internet Service  count
1423      San Diego              Yes    253
912     Los Angeles              Yes    222
1440       San Jose              Yes     94
1396     Sacramento              Yes     78
1429  San Francisco              Yes     74
902      Long Beach              Yes     47
574          Fresno              Yes     47
504       Escondido              Yes     42
1142        Oakland              Yes     41
519       Fallbrook              Yes     38
```

```
plt.figure(figsize=(10, 6))
sns.barplot(data=ten_city, x='City', y='count', palette='viridis')
plt.title('Top 10 City use Internet Service', fontsize=16)
plt.xlabel('City', fontsize=14)
plt.ylabel('Count', fontsize=14)
plt.xticks(rotation=45)
```

```
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\3411506481.py:2:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.
```

```
  sns.barplot(data=ten_city, x='City', y='count', palette='viridis')
```

```
([0, 1, 2, 3, 4, 5, 6, 7, 8, 9],
 [Text(0, 0, 'San Diego'),
  Text(1, 0, 'Los Angeles'),
  Text(2, 0, 'San Jose'),
  Text(3, 0, 'Sacramento'),
  Text(4, 0, 'San Francisco'),
  Text(5, 0, 'Long Beach'),
  Text(6, 0, 'Fresno'),
  Text(7, 0, 'Escondido'),
  Text(8, 0, 'Oakland'),
  Text(9, 0, 'Fallbrook')])
```




```
#Correlation plot
data.columns
df_corr = data[[ 'Gender', 'Age', 'Married', 'Number of
Dependents','Tenure in Months', 'Offer', 'Phone Service',
       'Avg Monthly Long Distance Charges', 'Multiple Lines',
       'Internet Service', 'Internet Type', 'Avg Monthly GB Download',
       'Online Security', 'Online Backup', 'Device Protection Plan',
```

```
       'Premium Tech Support', 'Streaming TV', 'Streaming Movies',
       'Streaming Music', 'Unlimited Data', 'Contract', 'Paperless
Billing',
       'Payment Method', 'Total Charges', 'Total Refunds',
       'Total Extra Data Charges', 'Total Long Distance
Charges','Total Revenue', 'Customer Status']]
label_encoder = LabelEncoder()
for column in df_corr.select_dtypes(include=[ 'object',
'category']).columns:
    df_corr[column] = label_encoder.fit_transform(df_corr[column])
```

```
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\355198599.py:13:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_corr[column] = label_encoder.fit_transform(df_corr[column])
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\355198599.py:13:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_corr[column] = label_encoder.fit_transform(df_corr[column])
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\355198599.py:13:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_corr[column] = label_encoder.fit_transform(df_corr[column])
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\355198599.py:13:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_corr[column] = label_encoder.fit_transform(df_corr[column])
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\355198599.py:13:
SettingWithCopyWarning:
```

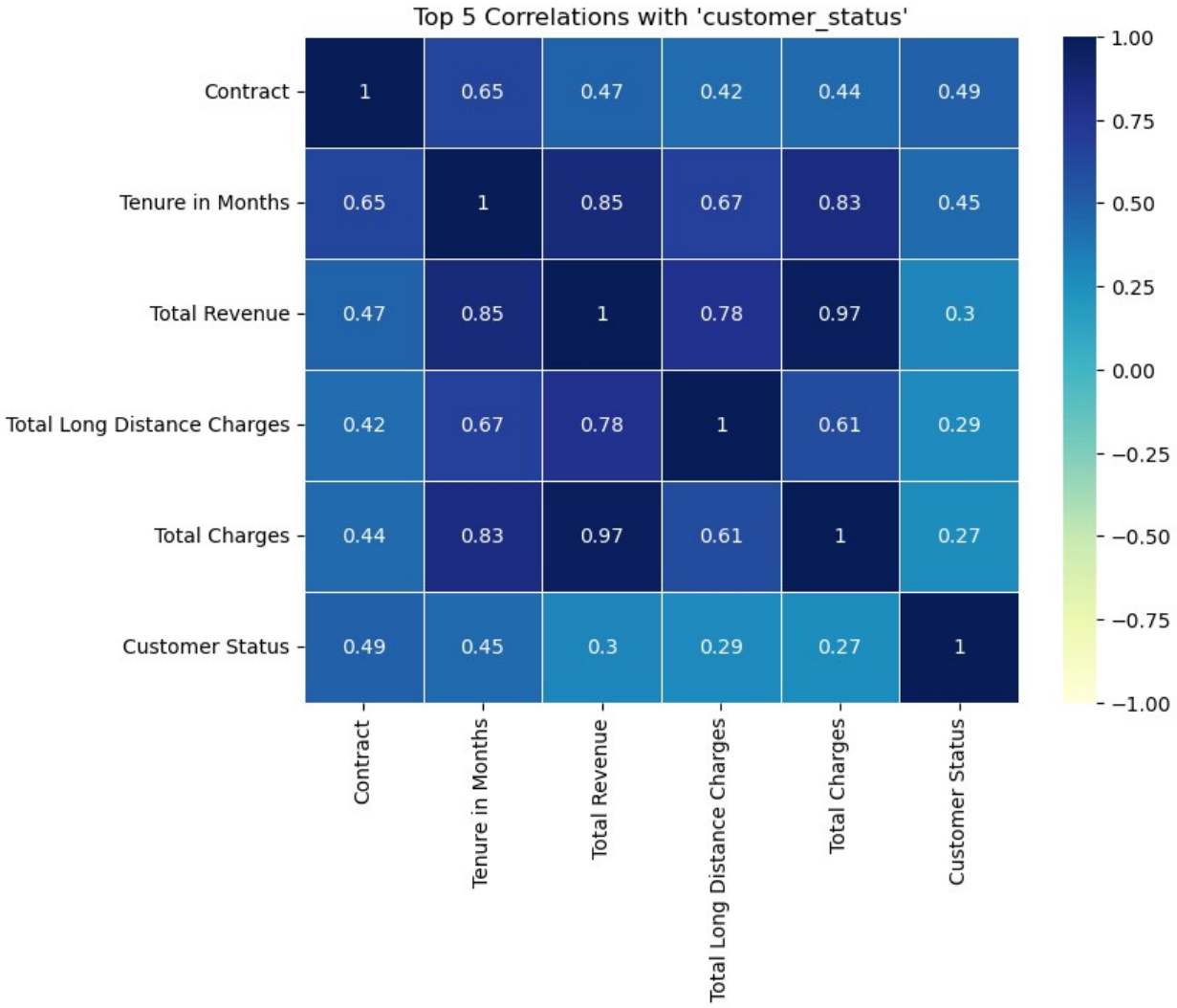
```
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_corr[column] = label_encoder.fit_transform(df_corr[column])
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\355198599.py:13:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_corr[column] = label_encoder.fit_transform(df_corr[column])
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\355198599.py:13:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_corr[column] = label_encoder.fit_transform(df_corr[column])
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\355198599.py:13:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_corr[column] = label_encoder.fit_transform(df_corr[column])
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\355198599.py:13:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_corr[column] = label_encoder.fit_transform(df_corr[column])
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\355198599.py:13:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
```

```
returning-a-view-versus-a-copy
  df_corr[column] = label_encoder.fit_transform(df_corr[column])
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\355198599.py:13:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_corr[column] = label_encoder.fit_transform(df_corr[column])
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\355198599.py:13:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_corr[column] = label_encoder.fit_transform(df_corr[column])
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\355198599.py:13:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_corr[column] = label_encoder.fit_transform(df_corr[column])
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\355198599.py:13:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_corr[column] = label_encoder.fit_transform(df_corr[column])
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\355198599.py:13:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_corr[column] = label_encoder.fit_transform(df_corr[column])
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\355198599.py:13:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
```

```
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_corr[column] = label_encoder.fit_transform(df_corr[column])
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\355198599.py:13:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_corr[column] = label_encoder.fit_transform(df_corr[column])
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\355198599.py:13:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_corr[column] = label_encoder.fit_transform(df_corr[column])
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\355198599.py:13:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_corr[column] = label_encoder.fit_transform(df_corr[column])
```

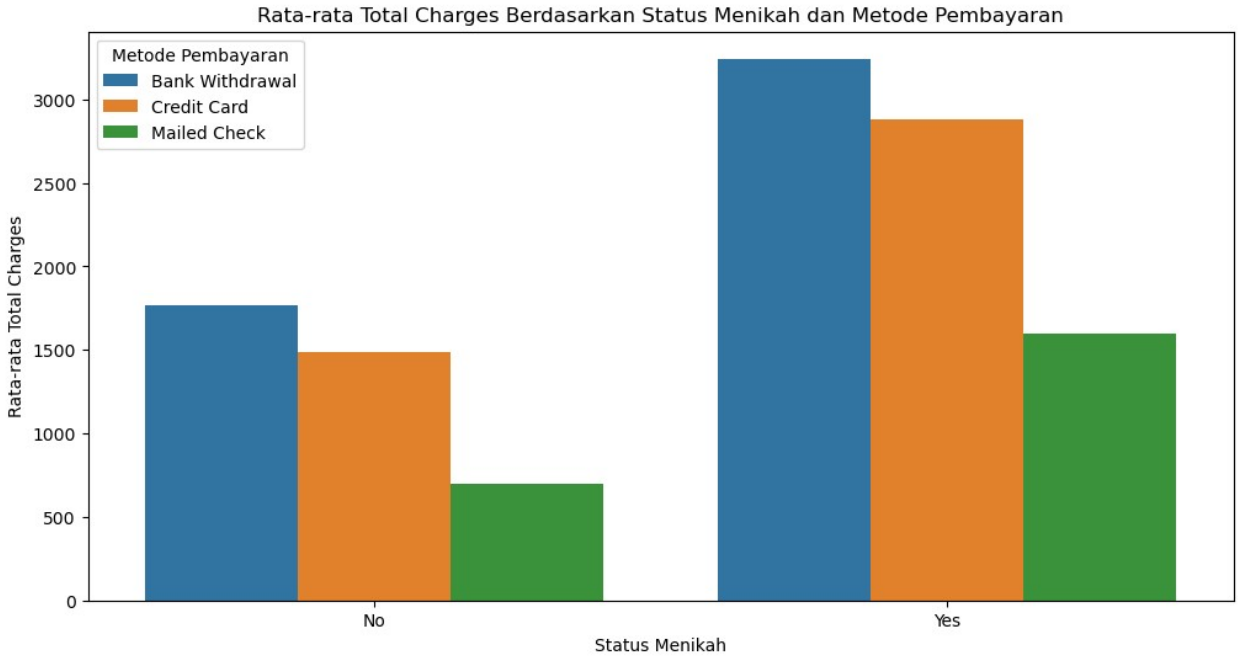
```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
df_corr_scaled = scaler.fit_transform(df_corr)

# Konversi hasil scaling kembali menjadi DataFrame
df_corr_scaled = pd.DataFrame(df_corr_scaled, columns=df_corr.columns)

# Hitung korelasi pada DataFrame yang telah dinormalisasi
correlation_matrix = df_corr_scaled.corr()

customer_status_corr = correlation_matrix['Customer
Status'].drop('Customer Status')  # Drop diri sendiri
top_5_correlations =
customer_status_corr.abs().sort_values(ascending=False).head(5)

# Filter matriks korelasi untuk menampilkan 'customer_status' dan
```

```
fitur teratas
top_features = top_5_correlations.index.tolist() + ['Customer Status']
heatmap_data = correlation_matrix.loc[top_features, top_features]

# Plot heatmap dengan palet warna yang menarik
plt.figure(figsize=(8, 6))
sns.heatmap(heatmap_data, annot=True, cmap='YlGnBu', linewidths=0.5,
vmin=-1, vmax=1)
plt.title("Top 5 Correlations with 'customer_status'")
plt.show()
```




```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Menghitung rata-rata Total Charges berdasarkan Married dan Payment
Method
```

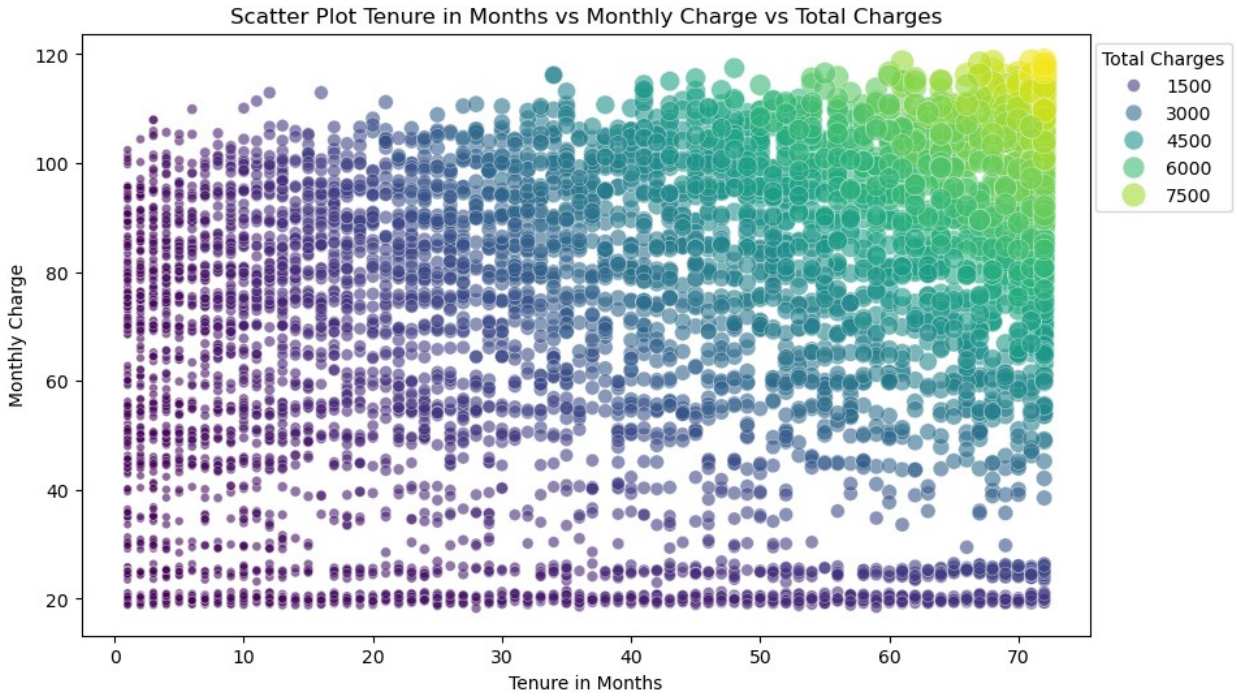
```
avg_total_charges = data.groupby(['Payment Method', 'Married'])['Total
Charges'].mean().reset_index()

# Membuat bar chart
plt.figure(figsize=(12, 6))
sns.barplot(data=avg_total_charges, x='Married', y='Total Charges',
hue='Payment Method')
plt.title('Rata-rata Total Charges Berdasarkan Status Menikah dan
Metode Pembayaran')
plt.xlabel('Status Menikah')
plt.ylabel('Rata-rata Total Charges')
plt.legend(title='Metode Pembayaran')
plt.show()
```




```
data.columns

Index(['Customer ID', 'Gender', 'Age', 'Married', 'Number of
Dependents',
       'City', 'Zip Code', 'Latitude', 'Longitude', 'Number of
Referrals',
       'Tenure in Months', 'Offer', 'Phone Service',
       'Avg Monthly Long Distance Charges', 'Multiple Lines',
       'Internet Service', 'Internet Type', 'Avg Monthly GB Download',
       'Online Security', 'Online Backup', 'Device Protection Plan',
       'Premium Tech Support', 'Streaming TV', 'Streaming Movies',
       'Streaming Music', 'Unlimited Data', 'Contract', 'Paperless
Billing',
       'Payment Method', 'Monthly Charge', 'Total Charges', 'Total
Refunds',
```

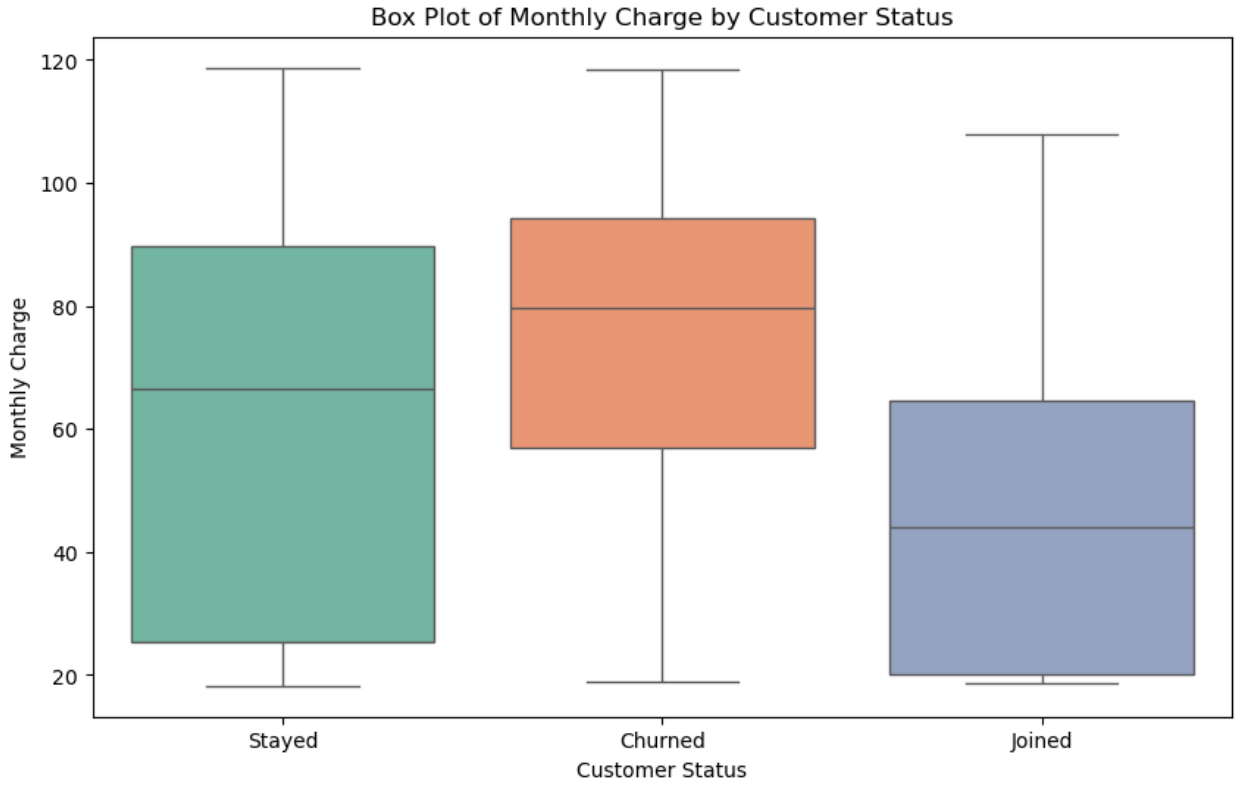
```
       'Total Extra Data Charges', 'Total Long Distance Charges',
       'Total Revenue', 'Customer Status', 'Churn Category', 'Churn
Reason'],
      dtype='object')
```

```
import matplotlib.pyplot as plt
import seaborn as sns

# Membuat scatter plot dengan 3 variabel
plt.figure(figsize=(10, 6))
sns.scatterplot(
    data=data,
    x='Tenure in Months',
    y='Monthly Charge',
    size='Total Charges',  # Ukuran titik berdasarkan Total Charges
    hue='Total Charges',   # Warna titik berdasarkan Total Charges
    palette='viridis',     # Skema warna
    sizes=(20, 200),       # Ukuran minimum dan maksimum titik
    alpha=0.6              # Transparansi titik
)

# Menambah judul dan label sumbu
plt.title('Scatter Plot Tenure in Months vs Monthly Charge vs Total
Charges')
plt.xlabel('Tenure in Months')
plt.ylabel('Monthly Charge')
plt.legend(title='Total Charges', loc='upper right',
bbox_to_anchor=(1.15, 1))
plt.show()
```




```
# Create a box plot for Monthly Charge based on Customer Status
(Churned vs. Stayed)
plt.figure(figsize=(10, 6))
box_plot = sns.boxplot(x='Customer Status', y='Monthly Charge',
data=data, palette='Set2')

# Set plot title and labels
box_plot.set_title('Box Plot of Monthly Charge by Customer Status')
box_plot.set_xlabel('Customer Status')
box_plot.set_ylabel('Monthly Charge')

plt.show()
```

```
C:\Users\ANUNG\AppData\Local\Temp\ipykernel_4080\2813320808.py:3:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  box_plot = sns.boxplot(x='Customer Status', y='Monthly Charge',
data=data, palette='Set2')
```


Box Plot of Monthly Charge by Customer Status
Monthly Charge
120
100
80
60
40
20
Stayed
Churned
Joined
Customer Status