In [262…

```
import pandas as pd
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv(r"C:\Users\Cali\Downloads\datamarch24\churn_clean_208.csv")
df.shape
#Cleaning the data:
df.duplicated().any() #checking for duplicated records
print(df.isna().sum()) #checking for NaN values.
print(df['InternetService'].unique())
df['InternetService'] = df['InternetService'].fillna('None')
print(df.isna().sum())
df = df.drop(['CaseOrder','Customer_id','Interaction','UID','City','State','County'
z_children = np.abs(stats.zscore(df['Children']))
outliers_children = df[(z_children > 3)]
print(outliers_children[['Children']])
filter_children = df[df['Children'] > 12]
select_children = filter_children[['Age', 'Children']]
print(select_children) #shows how many customers have more than 12 children. None f
z_age = np.abs(stats.zscore(df['Age']))
outliers_age = df[(z_age > 3)]
print(outliers_age[['Age']])
filter_age = df[df['Age'] < 18]
select_age = filter_age[['Age']]
print("Under 18 Years Old:")
print(filter_age)
z_income = np.abs(stats.zscore(df['Income']))
outliers_income = df[(z_income > 3)]
print(outliers_income[['Income']])
filter_income = df[df['Income'] > 200000.00] #only 3 records return and seem reason
select_income = filter_income[['Income']]
print(select_income)
z_tenure = np.abs(stats.zscore(df['Tenure']))
outliers_tenure = df[(z_tenure > 3)]
print(outliers_tenure) #no outliers found
z_bandwidth = np.abs(stats.zscore(df['Bandwidth_GB_Year']))
outliers_bandwidth = df[z_bandwidth > 3]
print(outliers_bandwidth)
z_outage = np.abs(stats.zscore(df['Outage_sec_perweek']))
outlier_outage = df[z_outage > 5]
print(outlier_outage[['Outage_sec_perweek']])
z_email = np.abs(stats.zscore(df['Email']))
outlier_email = df[z_email > 3]
print(outlier_email[['Email']])
z_contacts = np.abs(stats.zscore(df['Contacts']))
outlier_contacts = df[z_contacts > 5]
print(outlier_contacts[['Contacts']])
z_eqfail = np.abs(stats.zscore(df['Yearly_equip_failure']))
outlier_eqfail = df[z_eqfail > 5]
print(outlier_eqfail[['Yearly_equip_failure']])
z_monthlycharge= np.abs(stats.zscore(df['MonthlyCharge']))
outlier_monthlycharge =df[z_monthlycharge > 3]
print(outlier_monthlycharge[['MonthlyCharge']])
```

```
print(df['Marital'].unique()) #Checking for spelling errors/unusual data
print(df['Gender'].unique())
print(df['Churn'].unique())
print(df['Techie'].unique())
print(df['Contract'].unique())
print(df['Port_modem'].unique())
print(df['Tablet'].unique())
print(df['InternetService'].unique())
print(df['Phone'].unique())
print(df['Multiple'].unique())
print(df['OnlineSecurity'].unique())
print(df['OnlineBackup'].unique())
print(df['DeviceProtection'].unique())
print(df['TechSupport'].unique())
print(df['StreamingTV'].unique())
print(df['StreamingMovies'].unique())
print(df['PaperlessBilling'].unique())
```

```
CaseOrder                 0
Customer_id               0
Interaction               0
UID                       0
City                      0
State                     0
County                    0
Zip                       0
Lat                       0
Lng                       0
Population                0
Area                      0
TimeZone                  0
Job                       0
Children                  0
Age                       0
Income                    0
Marital                   0
Gender                    0
Churn                     0
Outage_sec_perweek        0
Email                     0
Contacts                  0
Yearly_equip_failure      0
Techie                    0
Contract                  0
Port_modem                0
Tablet                    0
InternetService        2129
Phone                     0
Multiple                  0
OnlineSecurity            0
OnlineBackup              0
DeviceProtection          0
TechSupport               0
StreamingTV               0
StreamingMovies           0
PaperlessBilling          0
PaymentMethod             0
Tenure                    0
MonthlyCharge             0
Bandwidth_GB_Year         0
Item1                     0
Item2                     0
Item3                     0
Item4                     0
Item5                     0
Item6                     0
Item7                     0
Item8                     0
dtype: int64
['Fiber Optic' 'DSL' nan]
CaseOrder             0
Customer_id           0
Interaction           0
UID                   0
```

```
City                    0
State                   0
County                  0
Zip                     0
Lat                     0
Lng                     0
Population              0
Area                    0
TimeZone                0
Job                     0
Children                0
Age                     0
Income                  0
Marital                 0
Gender                  0
Churn                   0
Outage_sec_perweek      0
Email                   0
Contacts                0
Yearly_equip_failure    0
Techie                  0
Contract                0
Port_modem              0
Tablet                  0
InternetService         0
Phone                   0
Multiple                0
OnlineSecurity          0
OnlineBackup            0
DeviceProtection        0
TechSupport             0
StreamingTV             0
StreamingMovies         0
PaperlessBilling        0
PaymentMethod           0
Tenure                  0
MonthlyCharge           0
Bandwidth_GB_Year       0
Item1                   0
Item2                   0
Item3                   0
Item4                   0
Item5                   0
Item6                   0
Item7                   0
Item8                   0
dtype: int64
      Children
30           9
97          10
144         10
329          9
334          9
...        ...
9623        10
9676         9
```

```
9790        10
9871        10
9901         9

[191 rows x 1 columns]
Empty DataFrame
Columns: [Age, Children]
Index: []
Empty DataFrame
Columns: [Age]
Index: []
Under 18 Years Old:
Empty DataFrame
Columns: [Children, Age, Income, Marital, Gender, Churn, Outage_sec_perweek, Email,
Contacts, Yearly_equip_failure, Techie, Contract, Port_modem, Tablet, InternetServic
e, Phone, Multiple, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, Str
eamingTV, StreamingMovies, PaperlessBilling, Tenure, MonthlyCharge, Bandwidth_GB_Yea
r]
Index: []

[0 rows x 27 columns]
         Income
46    132116.33
130   125814.88
186   135727.71
470   156740.67
511   146494.70
...         ...
9615  130319.30
9639  149952.70
9656  136818.50
9849  134443.30
9876  128468.00

[145 rows x 1 columns]
        Income
4249  258900.7
5599  220383.0
5801  212255.3
6649  231252.0
9180  256998.4
Empty DataFrame
Columns: [Children, Age, Income, Marital, Gender, Churn, Outage_sec_perweek, Email,
Contacts, Yearly_equip_failure, Techie, Contract, Port_modem, Tablet, InternetServic
e, Phone, Multiple, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, Str
eamingTV, StreamingMovies, PaperlessBilling, Tenure, MonthlyCharge, Bandwidth_GB_Yea
r]
Index: []

[0 rows x 27 columns]
Empty DataFrame
Columns: [Children, Age, Income, Marital, Gender, Churn, Outage_sec_perweek, Email,
Contacts, Yearly_equip_failure, Techie, Contract, Port_modem, Tablet, InternetServic
e, Phone, Multiple, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, Str
eamingTV, StreamingMovies, PaperlessBilling, Tenure, MonthlyCharge, Bandwidth_GB_Yea
r]
```

```
Index: []

[0 rows x 27 columns]
Empty DataFrame
Columns: [Outage_sec_perweek]
Index: []
      Email
795       2
1152      2
1381      1
1399      2
1473     23
1746     22
6320      1
7408      2
8365      1
8948      2
9248      2
9475     22
      Contacts
426          6
4673         6
4811         7
5840         6
7746         7
9380         7
9713         6
9750         6
      Yearly_equip_failure
1116                     4
1228                     4
5166                     4
5471                     6
6345                     4
9386                     4
9623                     4
9763                     4
Empty DataFrame
Columns: [MonthlyCharge]
Index: []
['Widowed' 'Married' 'Separated' 'Never Married' 'Divorced']
['Male' 'Female' 'Nonbinary']
['No' 'Yes']
['No' 'Yes']
['One year' 'Month-to-month' 'Two Year']
['Yes' 'No']
['Yes' 'No']
['Fiber Optic' 'DSL' 'None']
['Yes' 'No']
['No' 'Yes']
['Yes' 'No']
['Yes' 'No']
['No' 'Yes']
['No' 'Yes']
['No' 'Yes']
```

```
['Yes' 'No']
['Yes' 'No']
```

In [263…

```
print(df['Children'].describe()) #description statistics
print(df['Age'].describe())
print(df['Marital'].describe())
print(df['Gender'].describe())
print(df['Churn'].describe())
print(df['Outage_sec_perweek'].describe())
print(df['Email'].describe())
print(df['Contacts'].describe())
print(df['Yearly_equip_failure'].describe())
print(df['Contract'].describe())
print(df['Port_modem'].describe())
print(df['Tablet'].describe())
print(df['InternetService'].describe())
print(df['Phone'].describe())
print(df['Multiple'].describe())
print(df['OnlineSecurity'].describe())
print(df['OnlineBackup'].describe())
print(df['DeviceProtection'].describe())
print(df['TechSupport'].describe())
print(df['StreamingTV'].describe())
print(df['StreamingMovies'].describe())
print(df['Tenure'].describe())
print(df['Bandwidth_GB_Year'].describe())
print(df['MonthlyCharge'].describe())
```

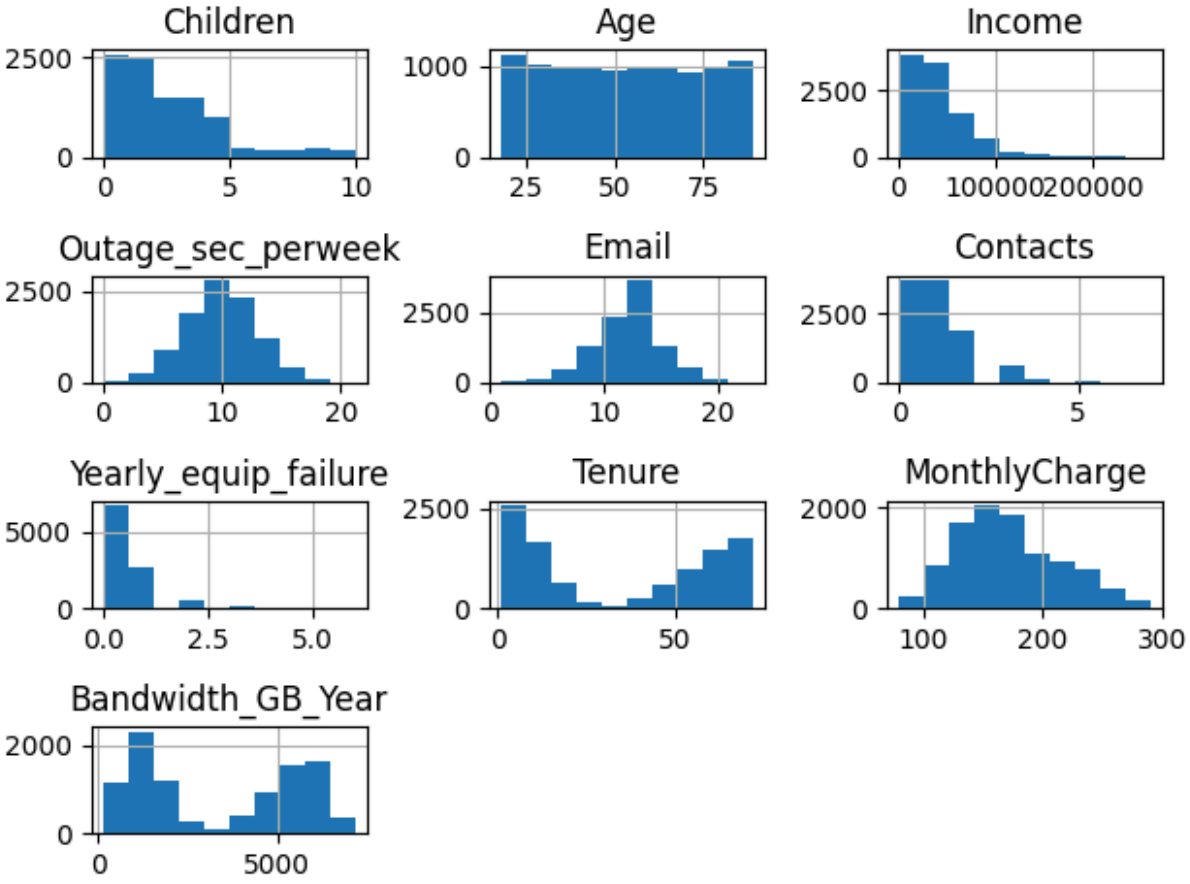
```
count    10000.0000
mean         2.0877
std          2.1472
min          0.0000
25%          0.0000
50%          1.0000
75%          3.0000
max         10.0000
Name: Children, dtype: float64
count    10000.000000
mean        53.078400
std         20.698882
min         18.000000
25%         35.000000
50%         53.000000
75%         71.000000
max         89.000000
Name: Age, dtype: float64
count        10000
unique           5
top       Divorced
freq          2092
Name: Marital, dtype: object
count      10000
unique         3
top       Female
freq        5025
Name: Gender, dtype: object
count     10000
unique        2
top          No
freq       7350
Name: Churn, dtype: object
count    10000.000000
mean        10.001848
std          2.976019
min          0.099747
25%          8.018214
50%         10.018560
75%         11.969485
max         21.207230
Name: Outage_sec_perweek, dtype: float64
count    10000.000000
mean        12.016000
std          3.025898
min          1.000000
25%         10.000000
50%         12.000000
75%         14.000000
max         23.000000
Name: Email, dtype: float64
count    10000.000000
mean         0.994200
std          0.988466
min          0.000000
25%          0.000000
```

```
50%          1.000000
75%          2.000000
max          7.000000
Name: Contacts, dtype: float64
count    10000.000000
mean         0.398000
std          0.635953
min          0.000000
25%          0.000000
50%          0.000000
75%          1.000000
max          6.000000
Name: Yearly_equip_failure, dtype: float64
count              10000
unique                 3
top       Month-to-month
freq                5456
Name: Contract, dtype: object
count     10000
unique        2
top          No
freq       5166
Name: Port_modem, dtype: object
count     10000
unique        2
top          No
freq       7009
Name: Tablet, dtype: object
count           10000
unique              3
top       Fiber Optic
freq             4408
Name: InternetService, dtype: object
count     10000
unique        2
top         Yes
freq       9067
Name: Phone, dtype: object
count     10000
unique        2
top          No
freq       5392
Name: Multiple, dtype: object
count     10000
unique        2
top          No
freq       6424
Name: OnlineSecurity, dtype: object
count     10000
unique        2
top          No
freq       5494
Name: OnlineBackup, dtype: object
count     10000
unique        2
top          No
```

```
freq       5614
Name: DeviceProtection, dtype: object
count     10000
unique        2
top          No
freq       6250
Name: TechSupport, dtype: object
count     10000
unique        2
top          No
freq       5071
Name: StreamingTV, dtype: object
count     10000
unique        2
top          No
freq       5110
Name: StreamingMovies, dtype: object
count    10000.000000
mean        34.526188
std         26.443063
min          1.000259
25%          7.917694
50%         35.430507
75%         61.479795
max         71.999280
Name: Tenure, dtype: float64
count    10000.000000
mean      3392.341550
std       2185.294852
min        155.506715
25%       1236.470827
50%       3279.536903
75%       5586.141370
max       7158.981530
Name: Bandwidth_GB_Year, dtype: float64
count    10000.000000
mean       172.624816
std         42.943094
min         79.978860
25%        139.979239
50%        167.484700
75%        200.734725
max        290.160419
Name: MonthlyCharge, dtype: float64
```

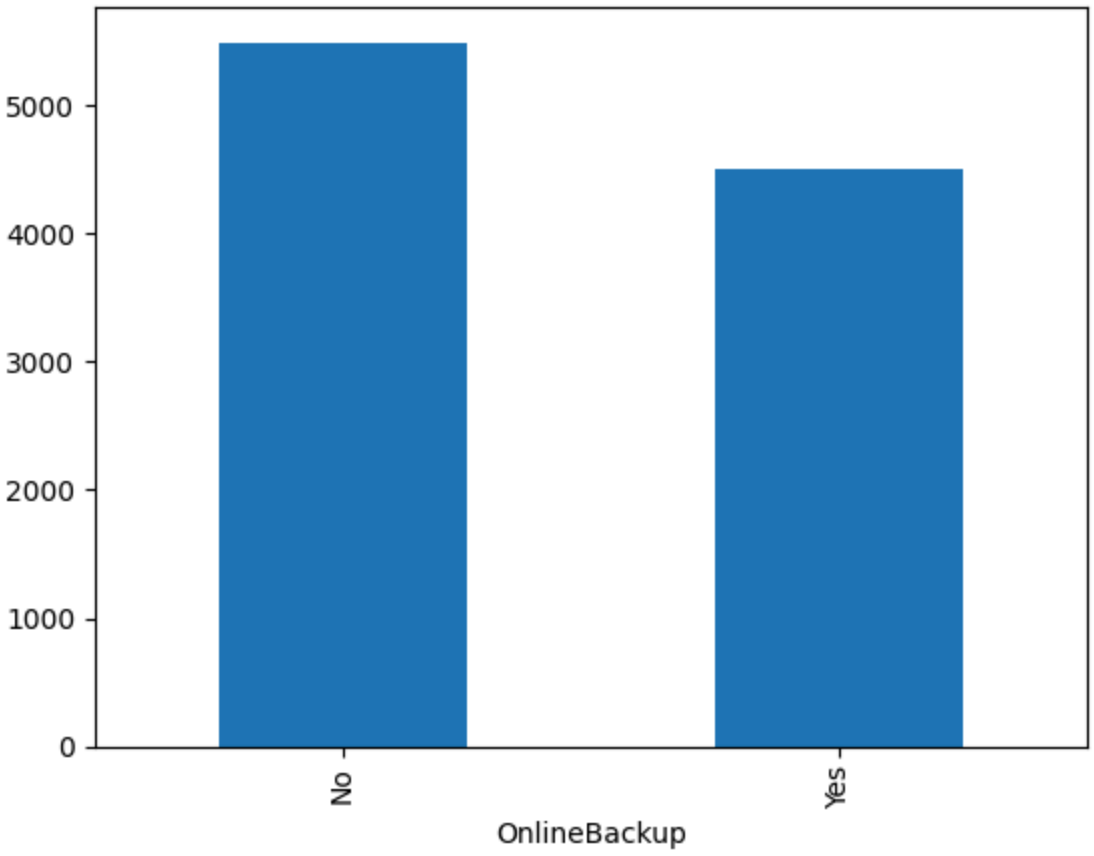
In [264…

```
df[['Children','Age', 'Income', 'Outage_sec_perweek','Email','Contacts','Yearly_equ
plt.savefig('churn_pyplot.jpg')
plt.tight_layout()
```



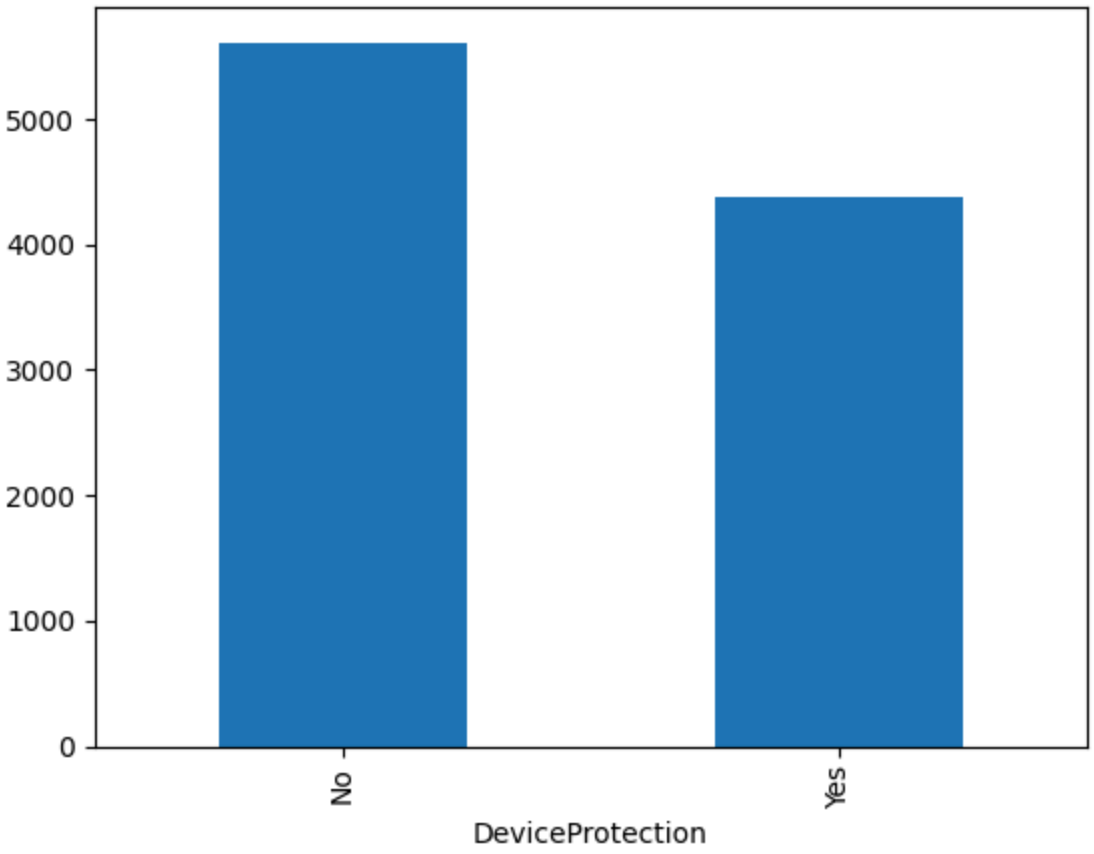
In [265…
```
df['OnlineBackup'].value_counts().plot(kind='bar') #Categorical Univariate visuals
```

Out[265…
```
<Axes: xlabel='OnlineBackup'>
```



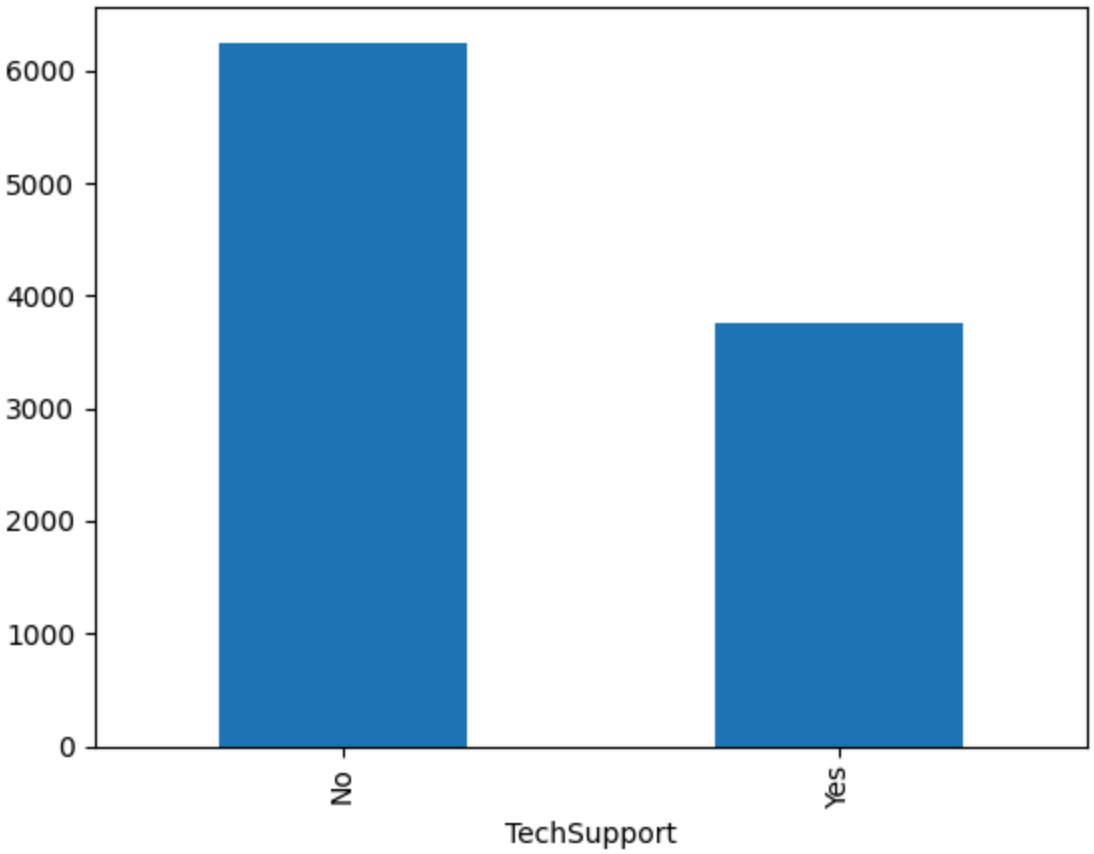
In [266…
```
df['DeviceProtection'].value_counts().plot(kind='bar')
```

Out[266…
```
<Axes: xlabel='DeviceProtection'>
```



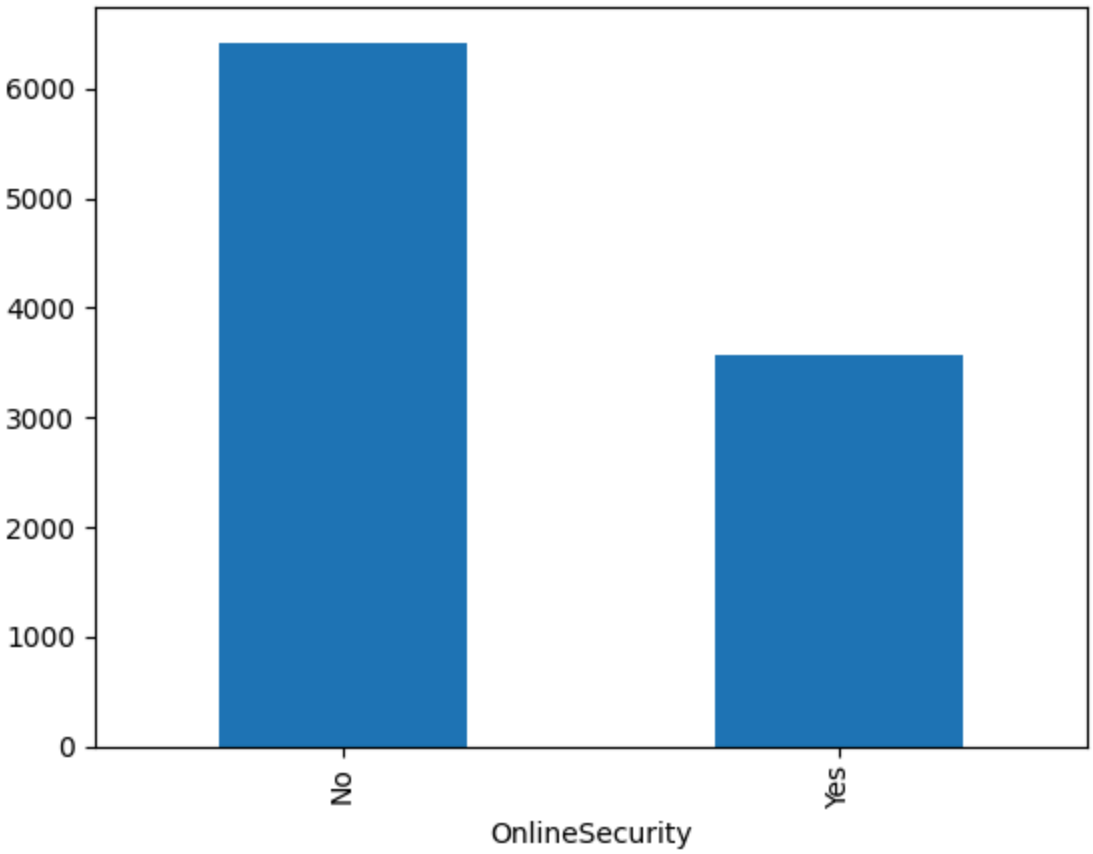
In [267…
```
df['TechSupport'].value_counts().plot(kind='bar')
```

Out[267…
```
<Axes: xlabel='TechSupport'>
```



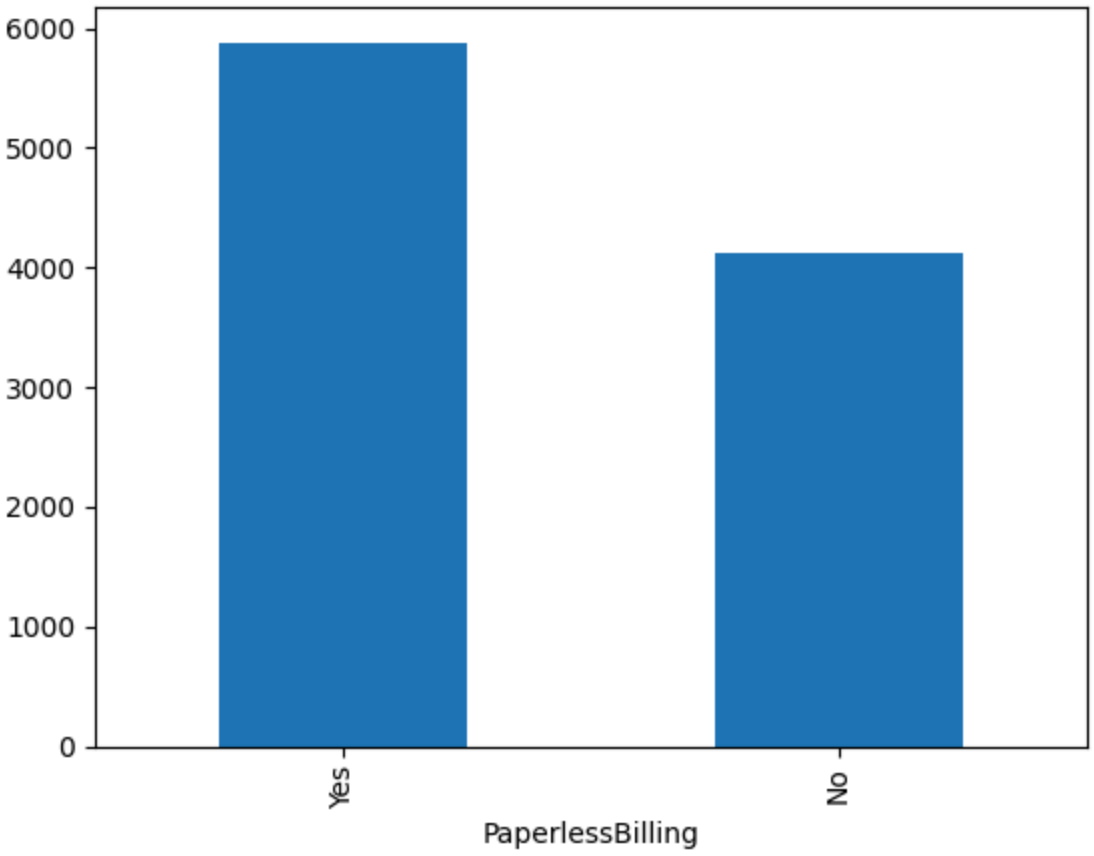
In [269…
```
df['OnlineSecurity'].value_counts().plot(kind='bar')
```

Out[269…
```
<Axes: xlabel='OnlineSecurity'>
```



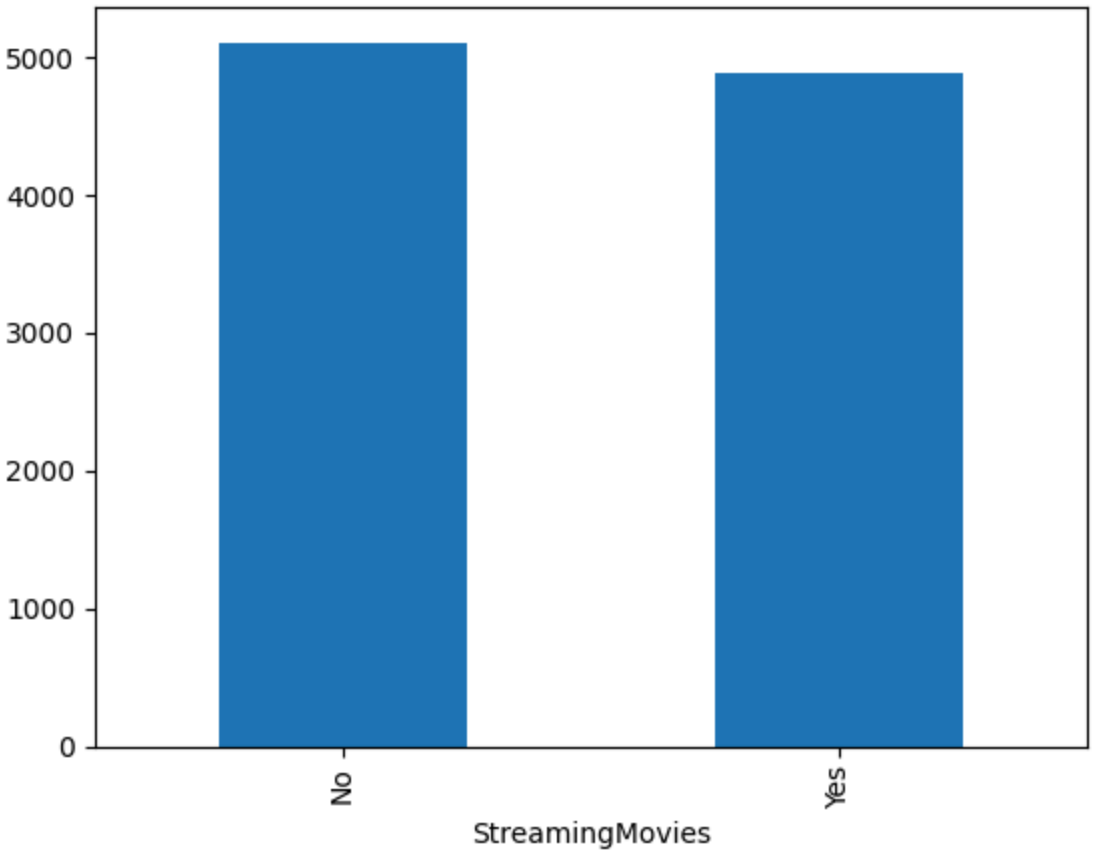
In [270…
```
df['PaperlessBilling'].value_counts().plot(kind='bar')
```

Out[270…
```
<Axes: xlabel='PaperlessBilling'>
```



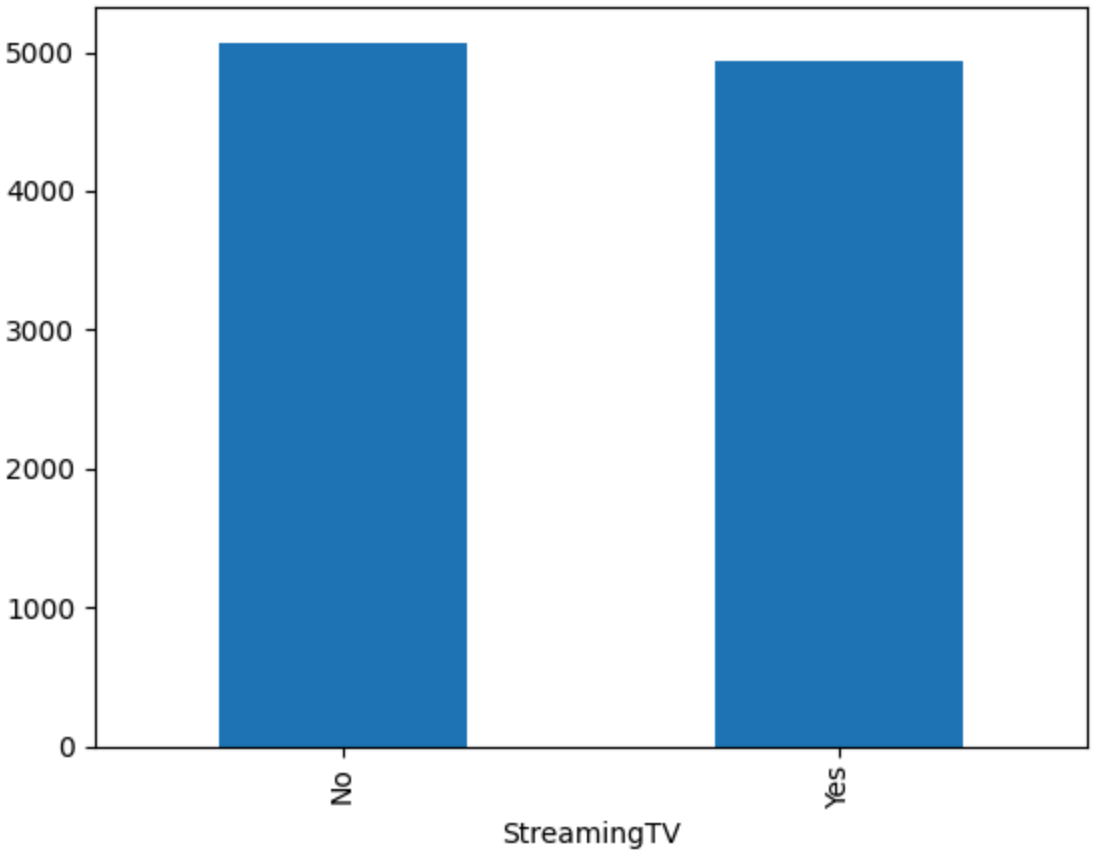
In [271…
```
df['StreamingMovies'].value_counts().plot(kind='bar')
```

Out[271…
```
<Axes: xlabel='StreamingMovies'>
```



In [272…
```
df['StreamingTV'].value_counts().plot(kind='bar')
```

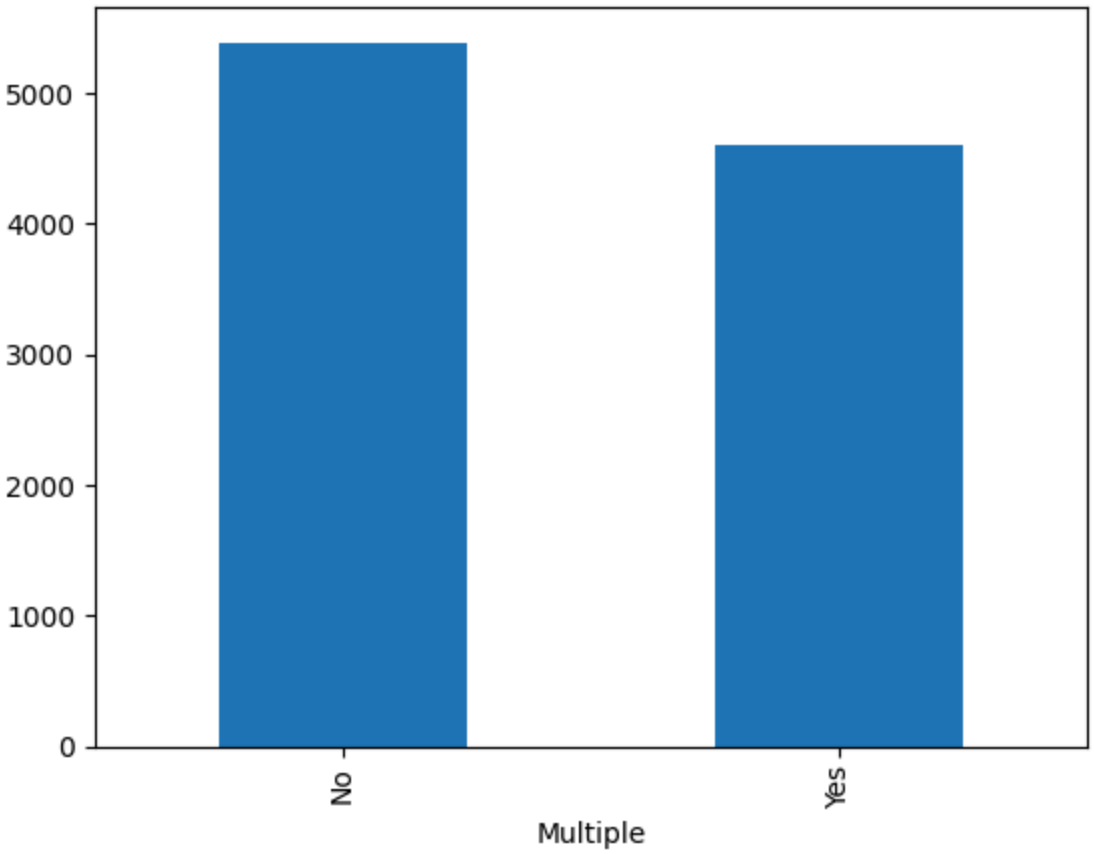
Out[272…
```
<Axes: xlabel='StreamingTV'>
```



In [ ]:

In [273…
```
df['Multiple'].value_counts().plot(kind='bar')
```

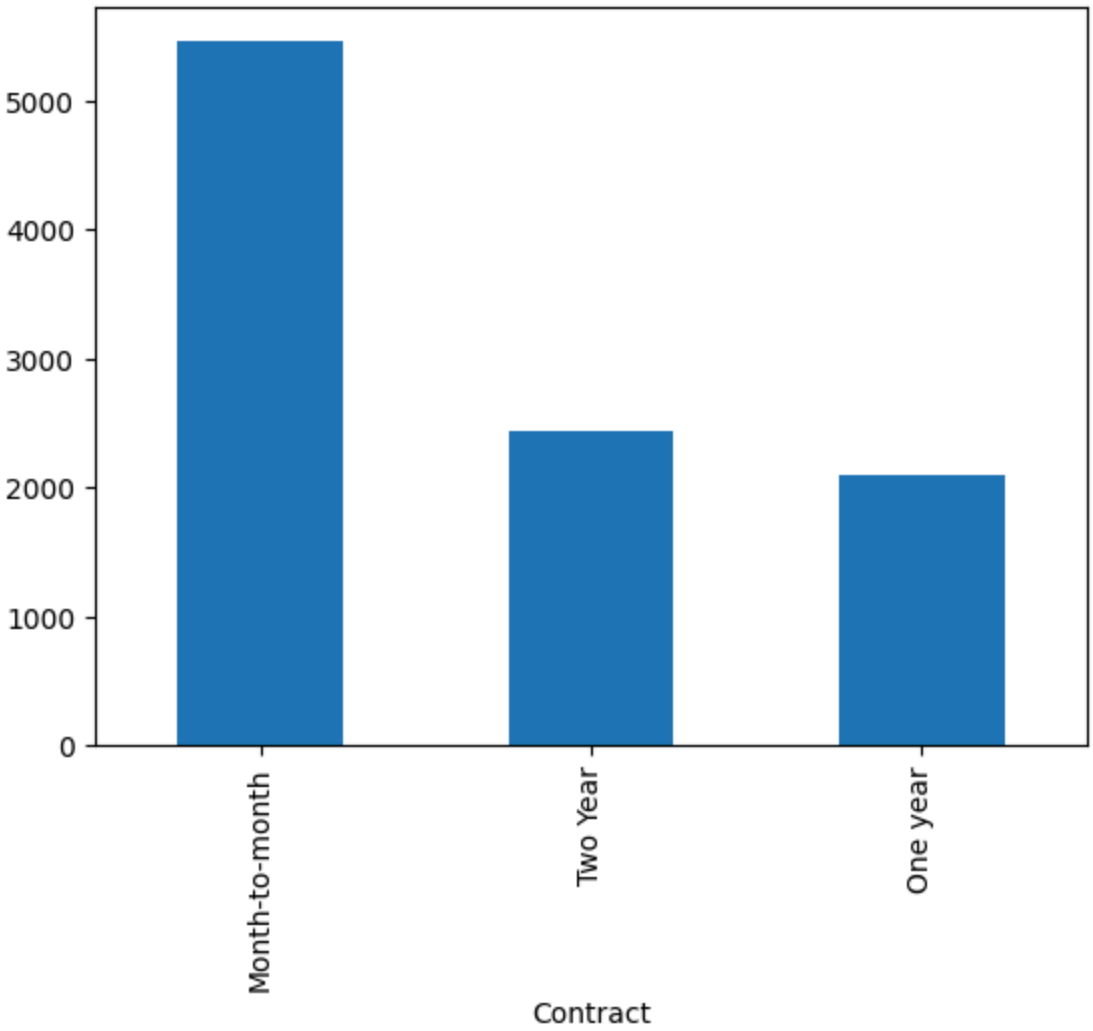
Out[273…
```
<Axes: xlabel='Multiple'>
```



In [ ]:

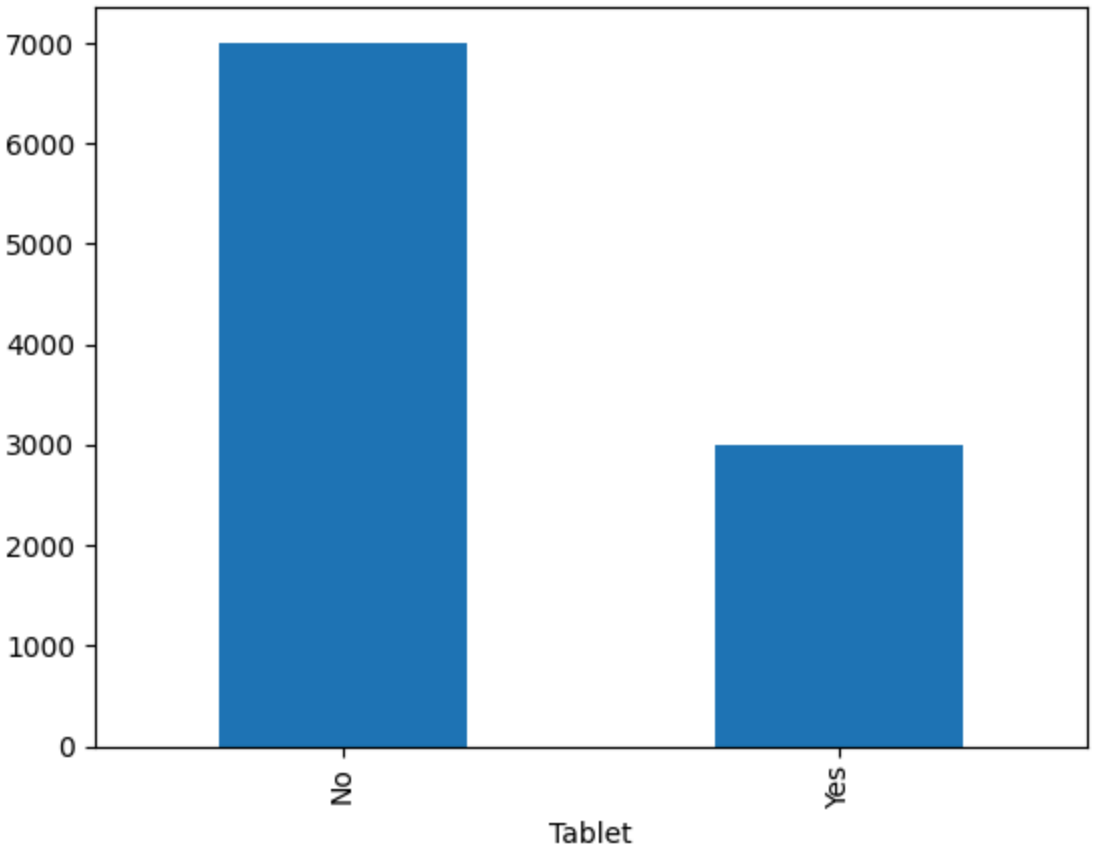
In [274…
```
df['Contract'].value_counts().plot(kind='bar')
```

Out[274…
```
<Axes: xlabel='Contract'>
```



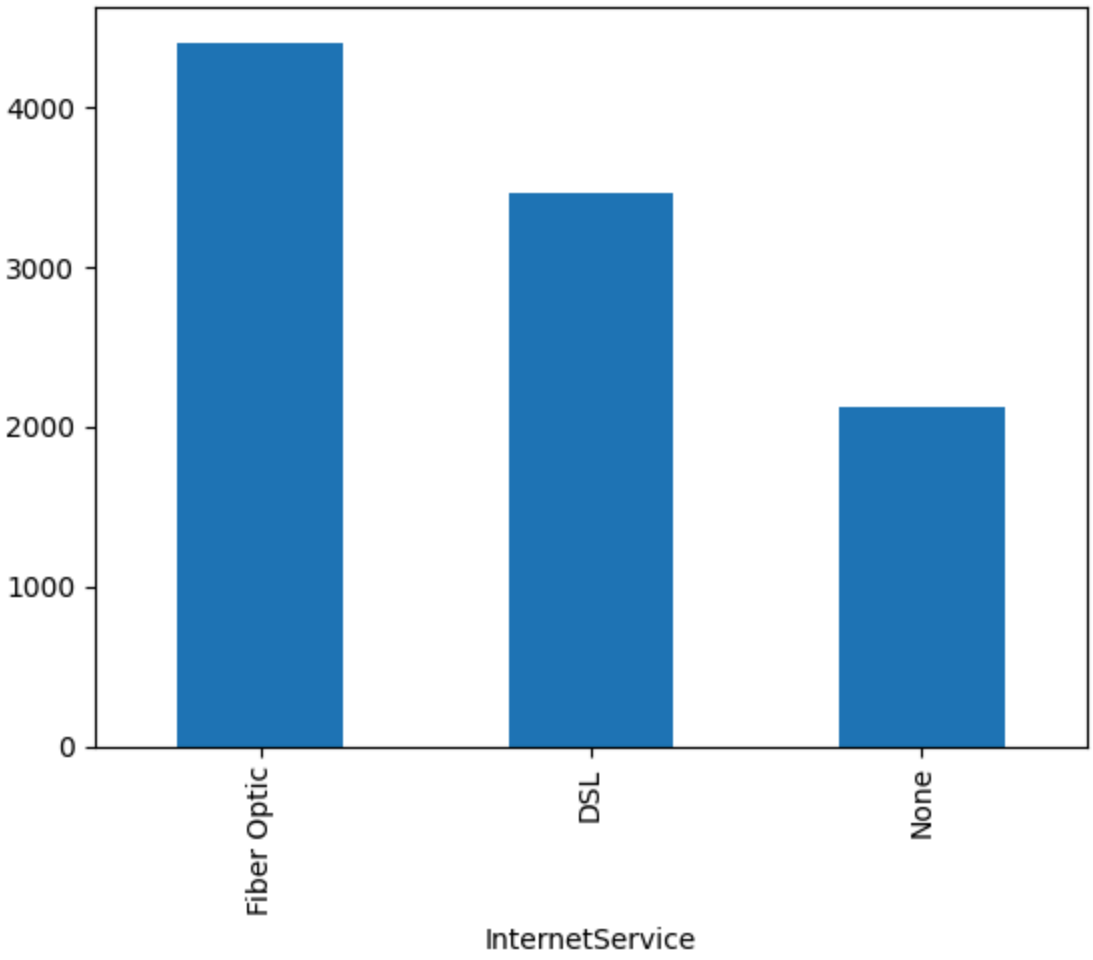
In [275…
```
df['Tablet'].value_counts().plot(kind='bar')
```

Out[275…
```
<Axes: xlabel='Tablet'>
```



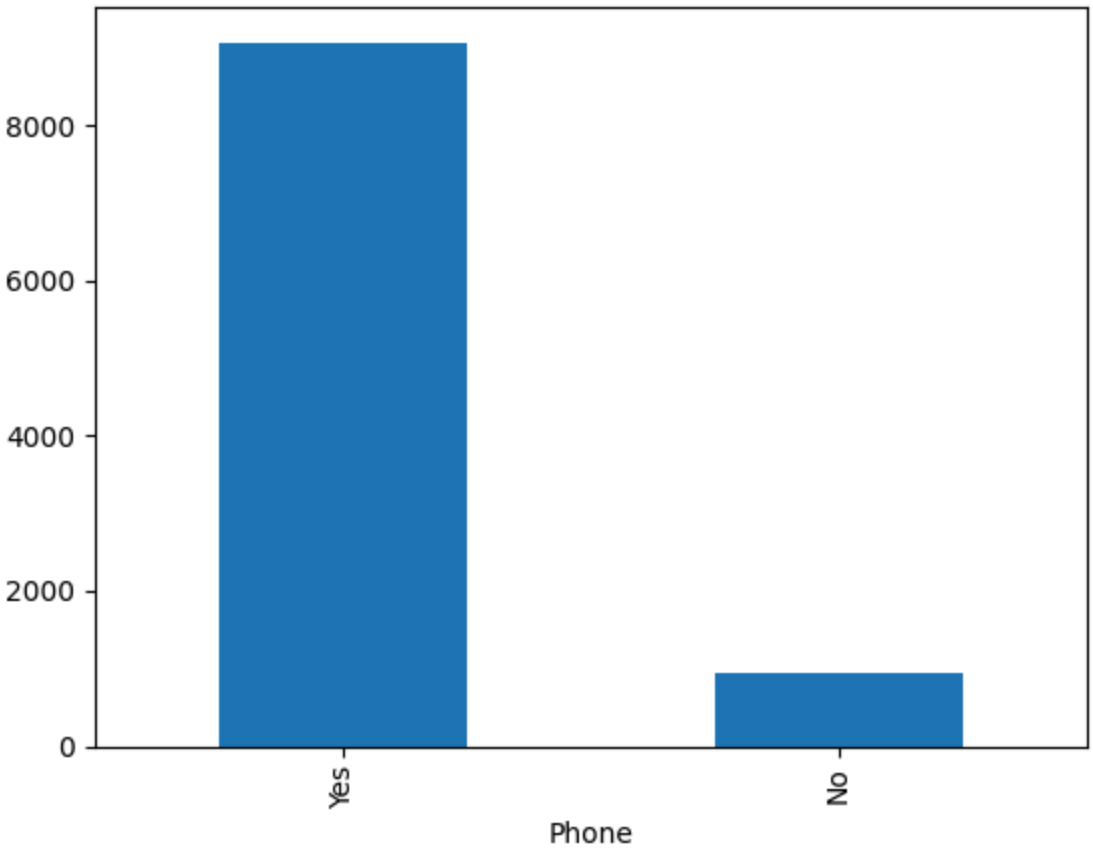
In [276…
```
df['InternetService'].value_counts().plot(kind='bar')
```

Out[276…
```
<Axes: xlabel='InternetService'>
```



In [277…
```
df['Phone'].value_counts().plot(kind='bar')
```

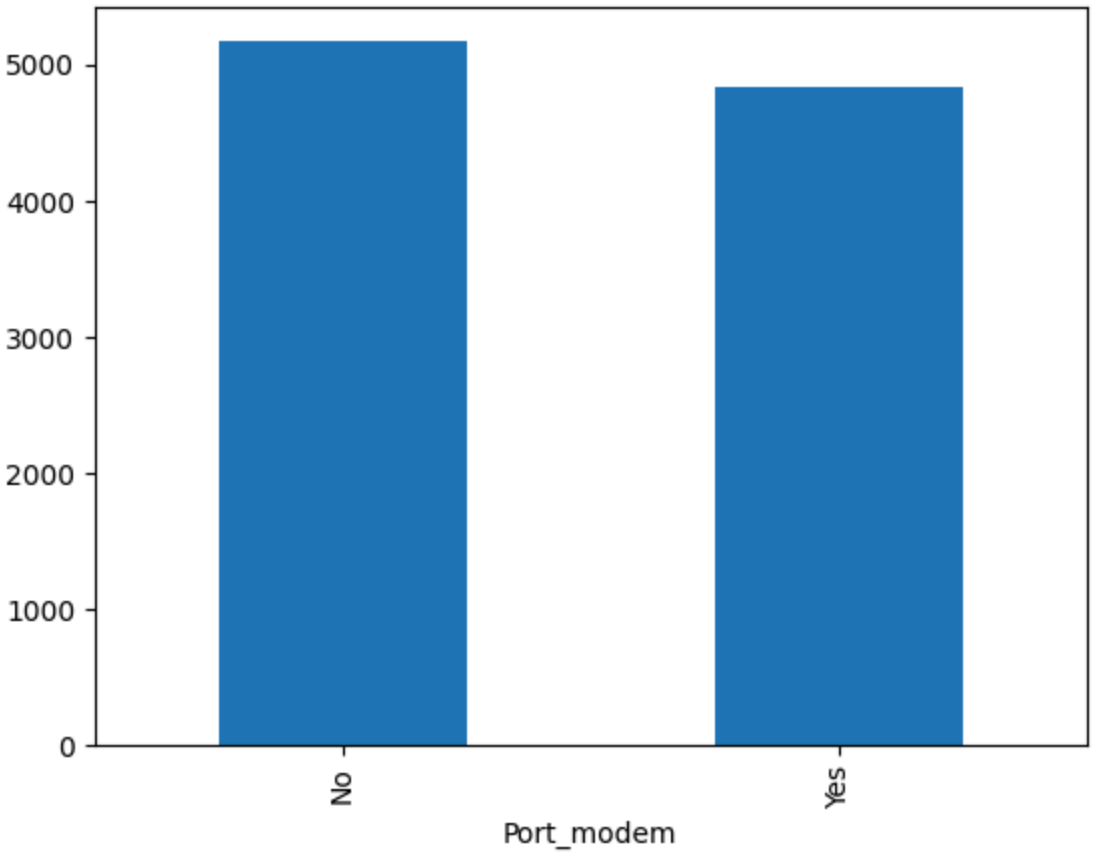
Out[277…
```
<Axes: xlabel='Phone'>
```



In [ ]:

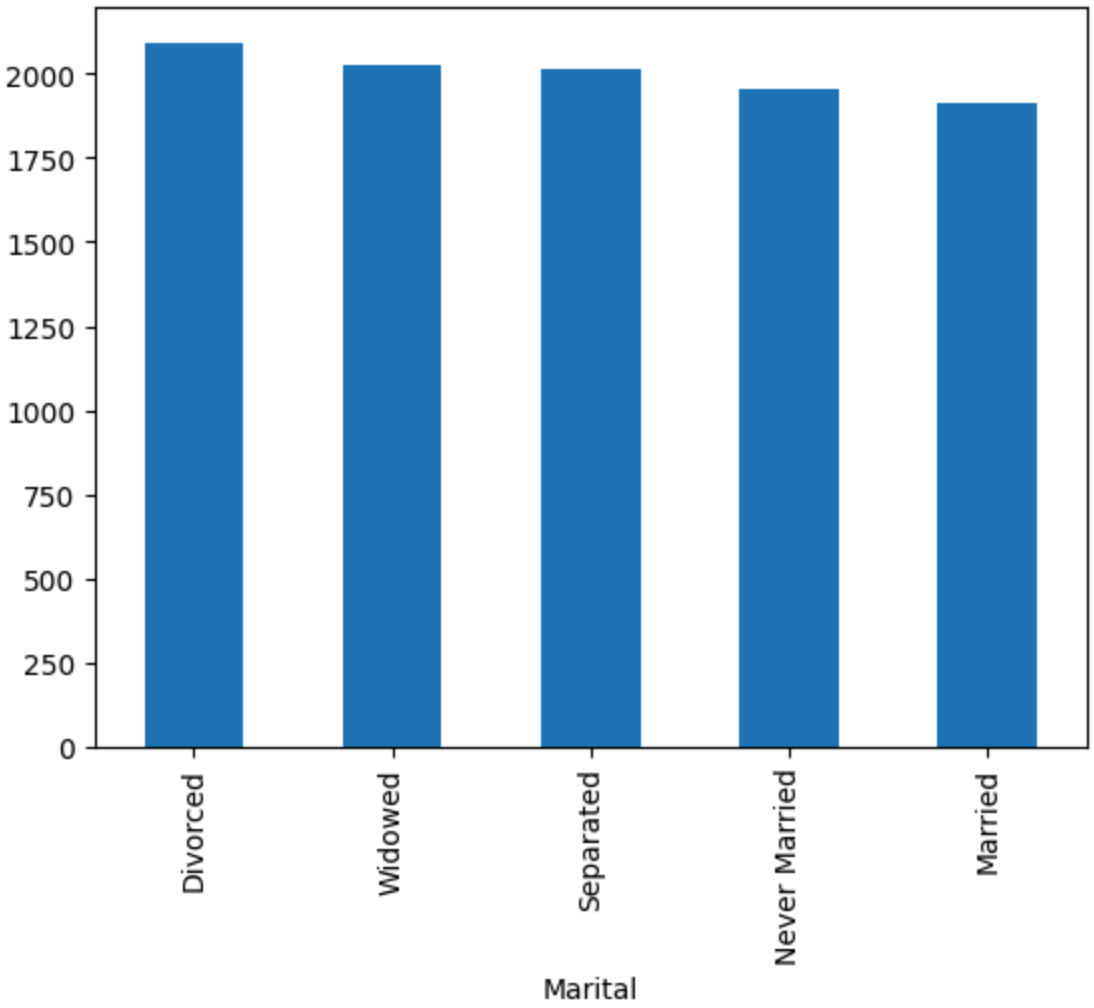
In [278…
```
df['Port_modem'].value_counts().plot(kind='bar')
```

Out[278…
```
<Axes: xlabel='Port_modem'>
```



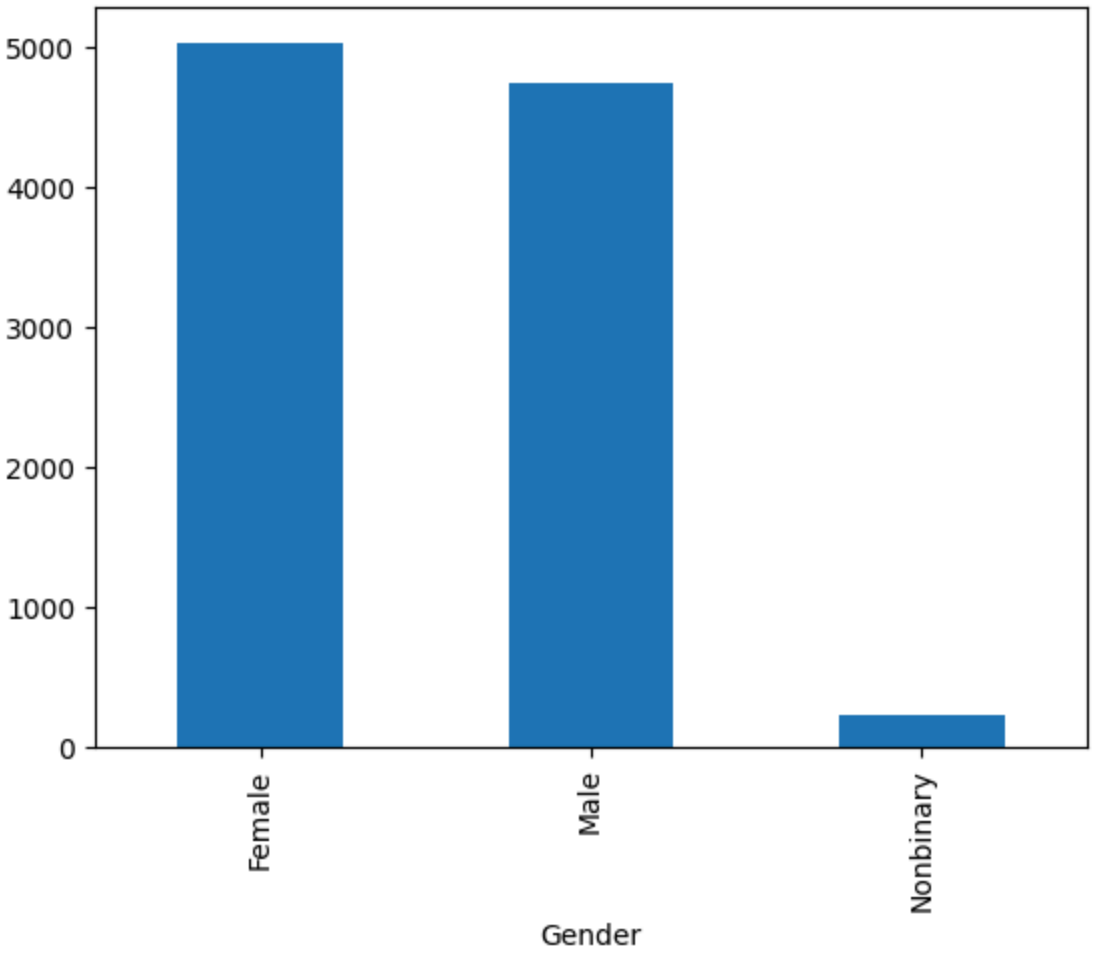
In [279…
```
df['Marital'].value_counts().plot(kind='bar')
```

Out[279…
```
<Axes: xlabel='Marital'>
```



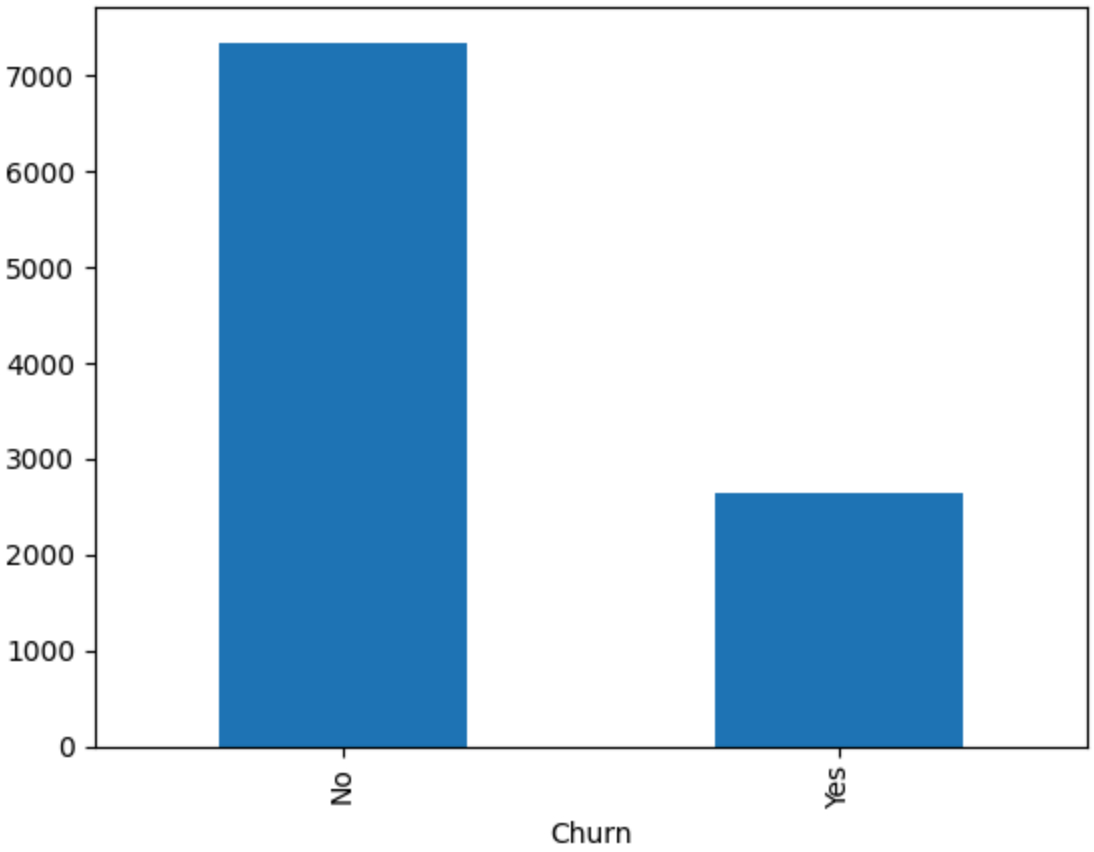
In [280…
```
df['Gender'].value_counts().plot(kind='bar')
```

Out[280…
```
<Axes: xlabel='Gender'>
```



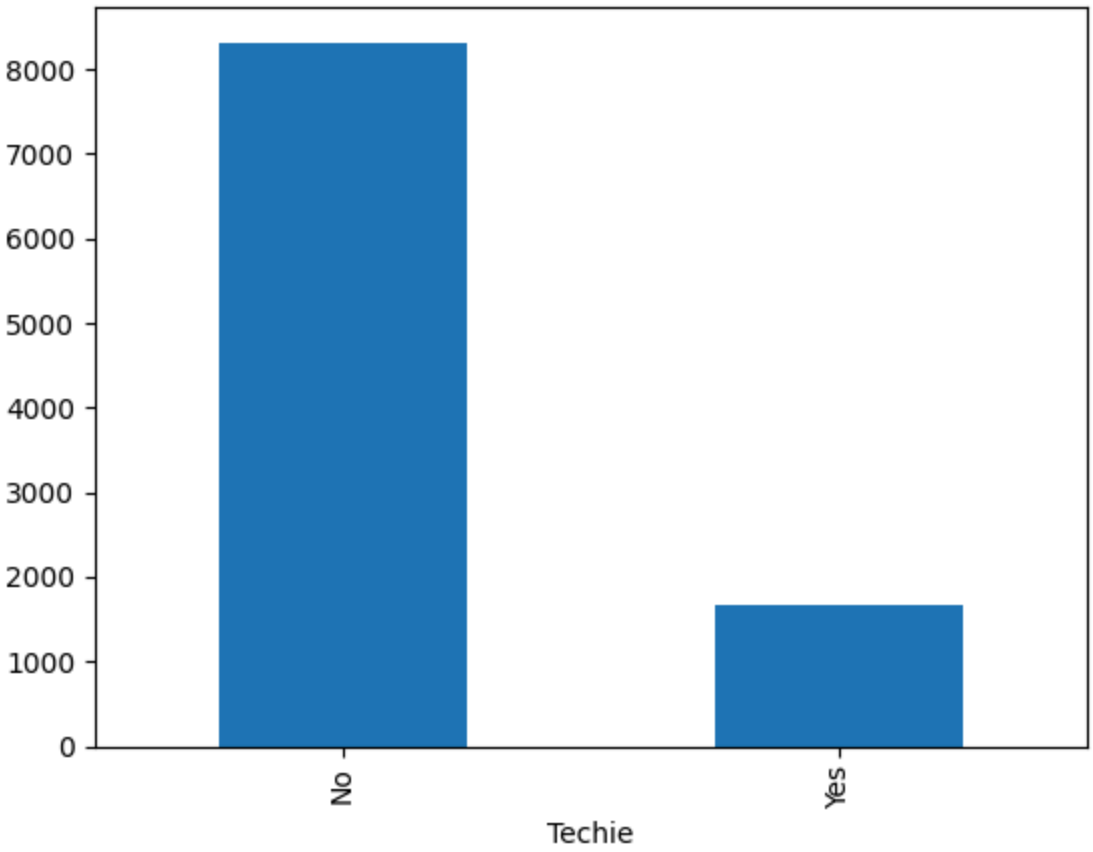
In [281…
```
df['Churn'].value_counts().plot(kind='bar')
```

Out[281…
```
<Axes: xlabel='Churn'>
```



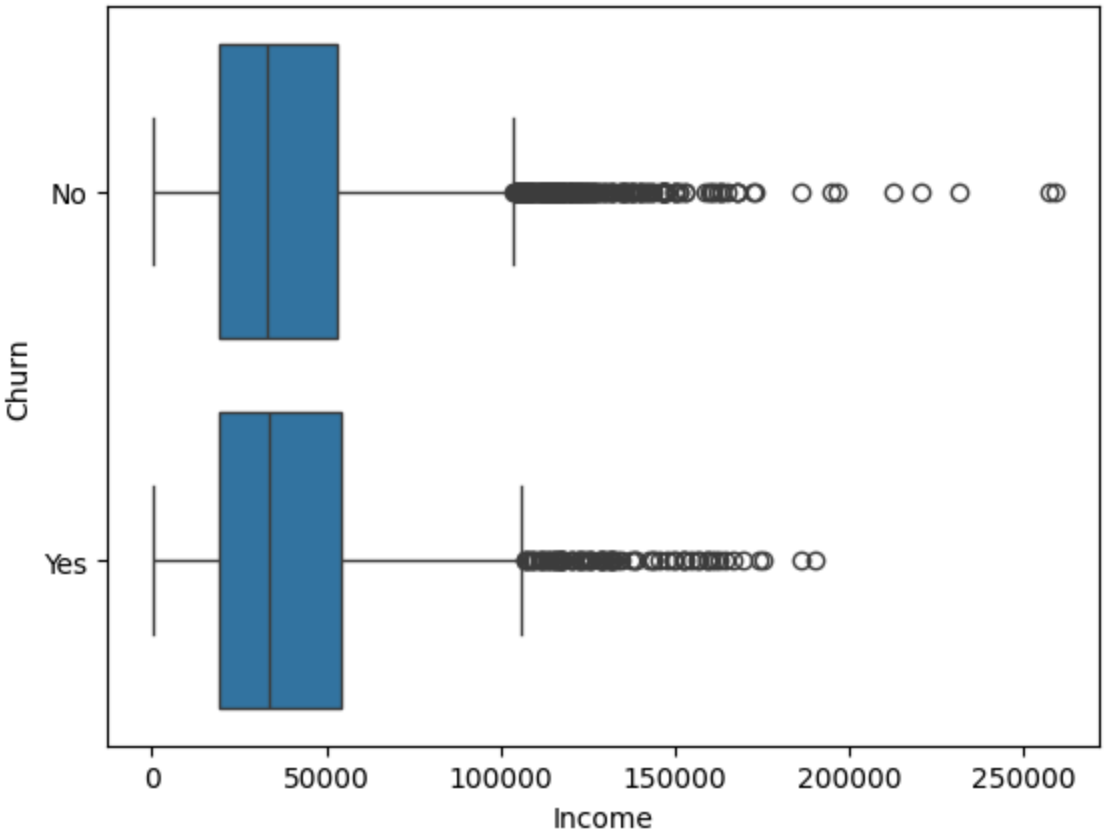
In [282…
```
df['Techie'].value_counts().plot(kind='bar')
```

Out[282…
```
<Axes: xlabel='Techie'>
```



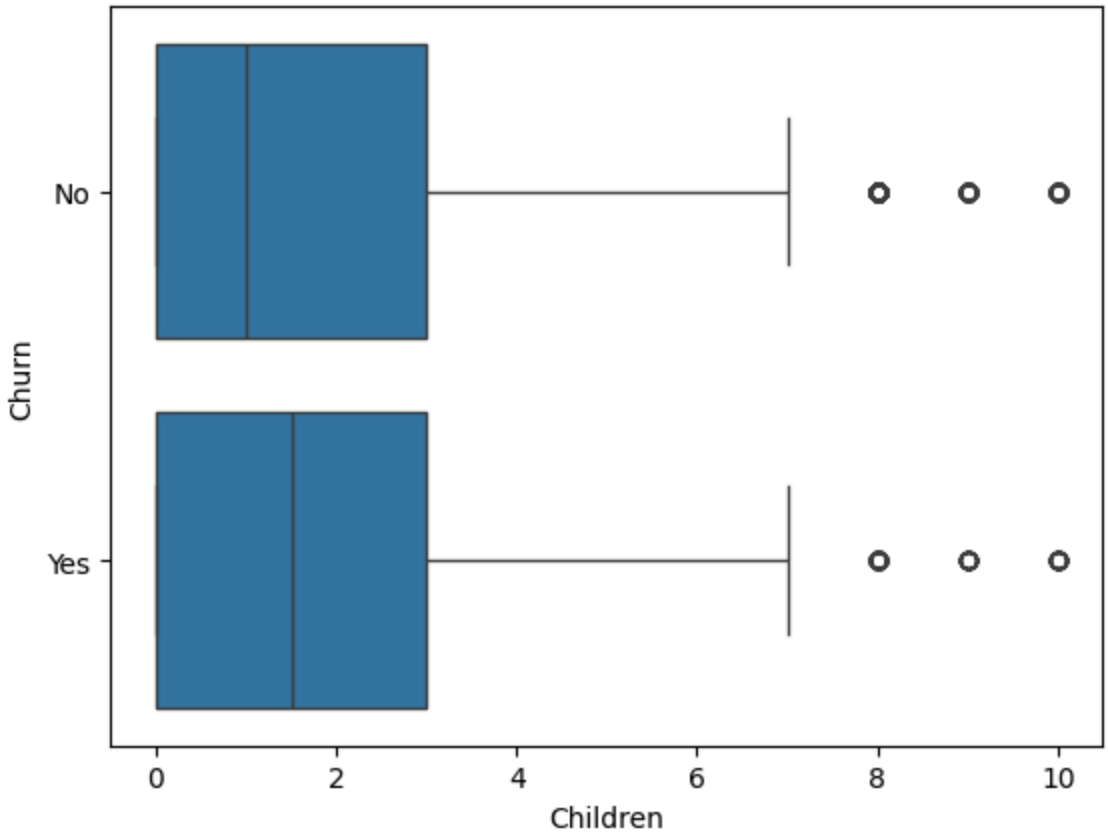
In [283…
```
sns.boxplot(x='Income', y='Churn', data=df)
```

Out[283…
```
<Axes: xlabel='Income', ylabel='Churn'>
```



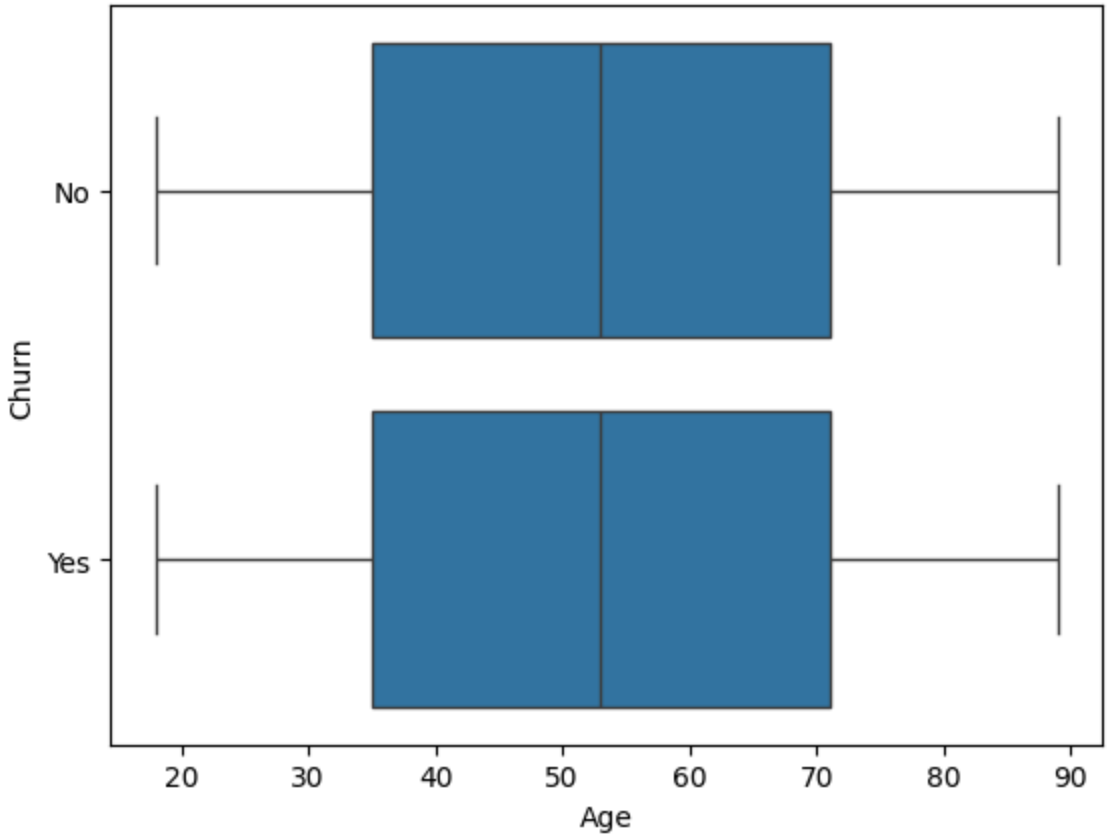
In [284…
```
sns.boxplot(x='Children', y='Churn', data=df)
```

Out[284…
```
<Axes: xlabel='Children', ylabel='Churn'>
```



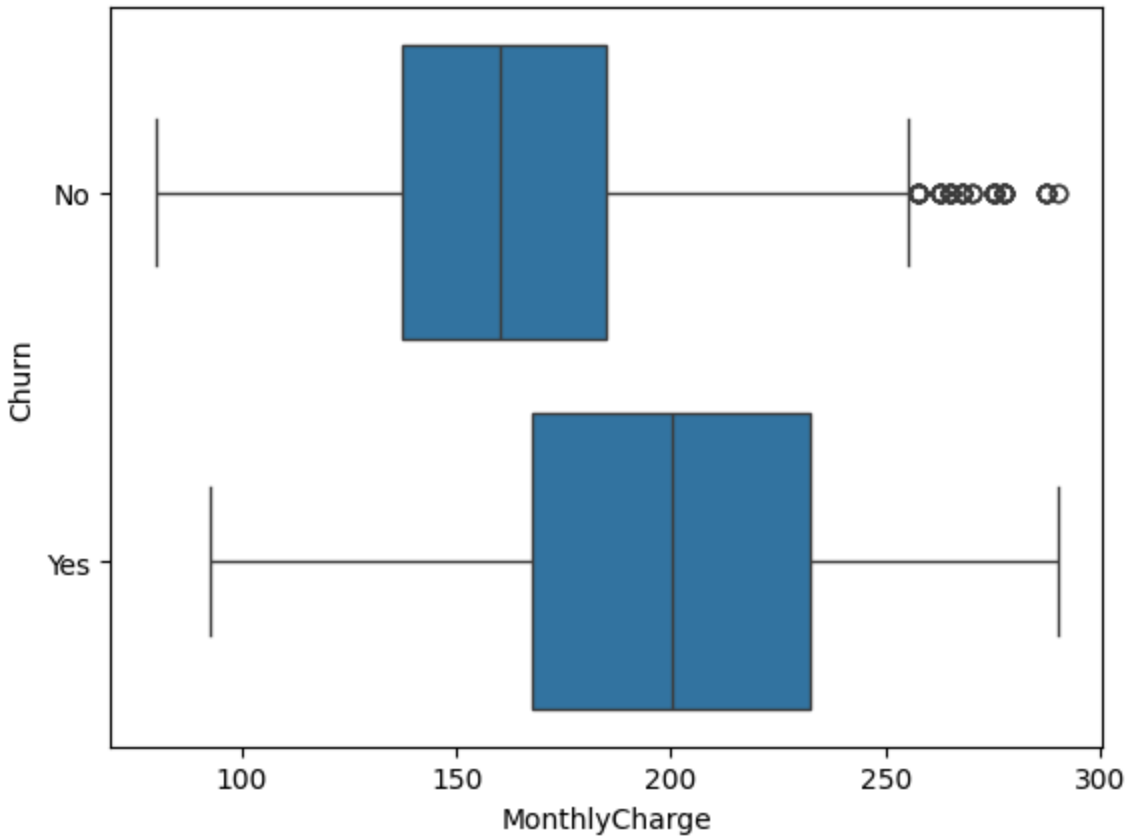
In [285…
```
sns.boxplot(x='Age', y='Churn', data=df)
```

Out[285…
```
<Axes: xlabel='Age', ylabel='Churn'>
```



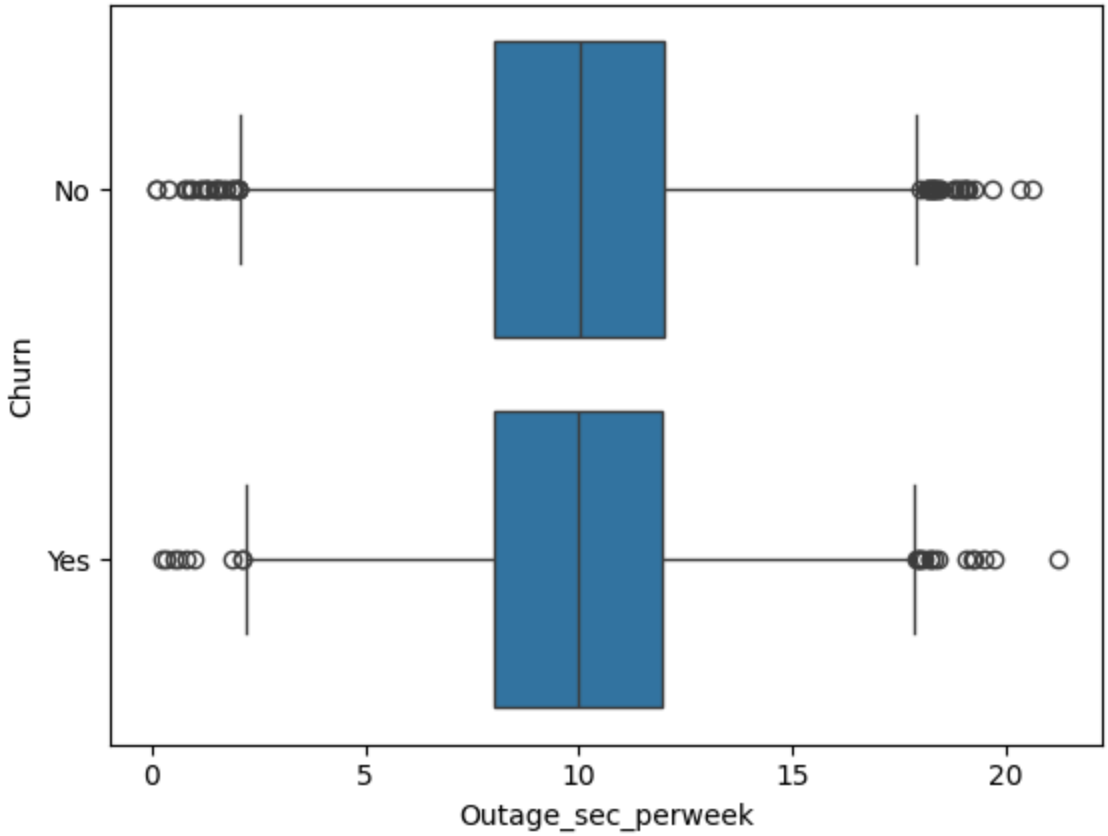
In [286…
```
sns.boxplot(x='MonthlyCharge', y='Churn', data=df)
```

Out[286…
```
<Axes: xlabel='MonthlyCharge', ylabel='Churn'>
```



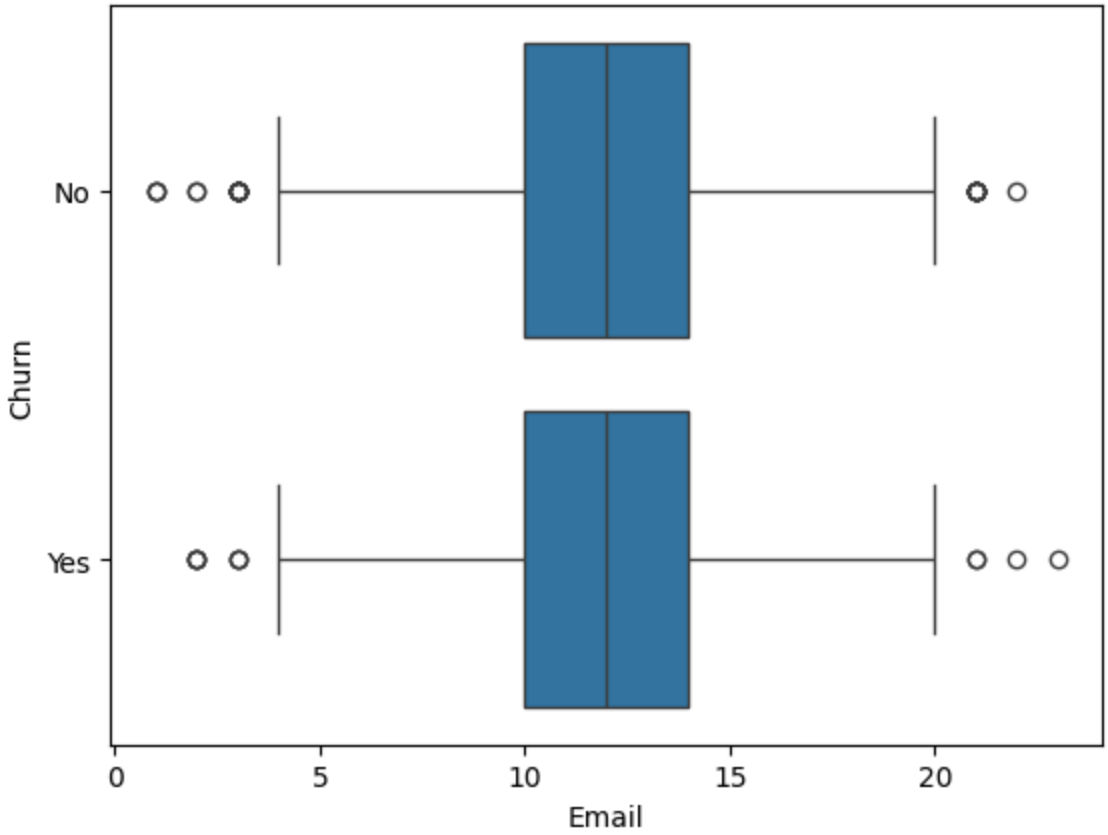
In [287…
```
sns.boxplot(x='Outage_sec_perweek', y='Churn', data=df)
```

Out[287…
```
<Axes: xlabel='Outage_sec_perweek', ylabel='Churn'>
```



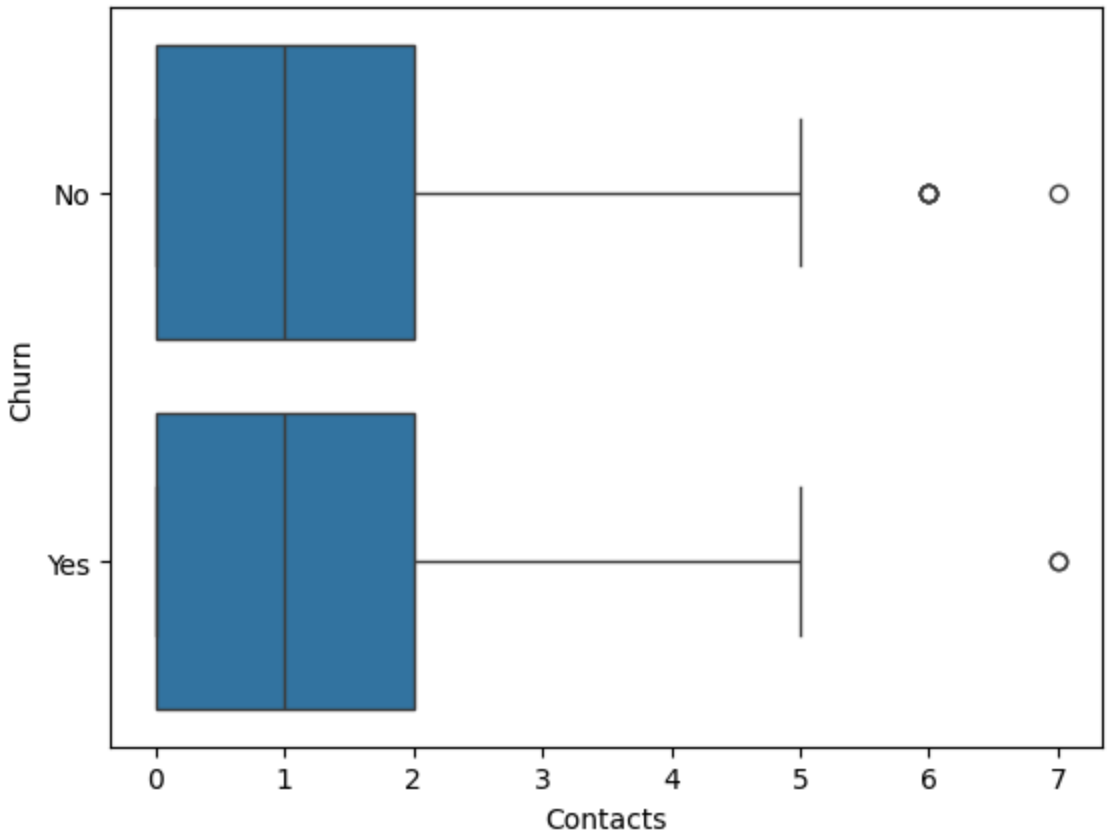
In [288…
```
sns.boxplot(x='Email', y='Churn', data=df)
```

Out[288…
```
<Axes: xlabel='Email', ylabel='Churn'>
```



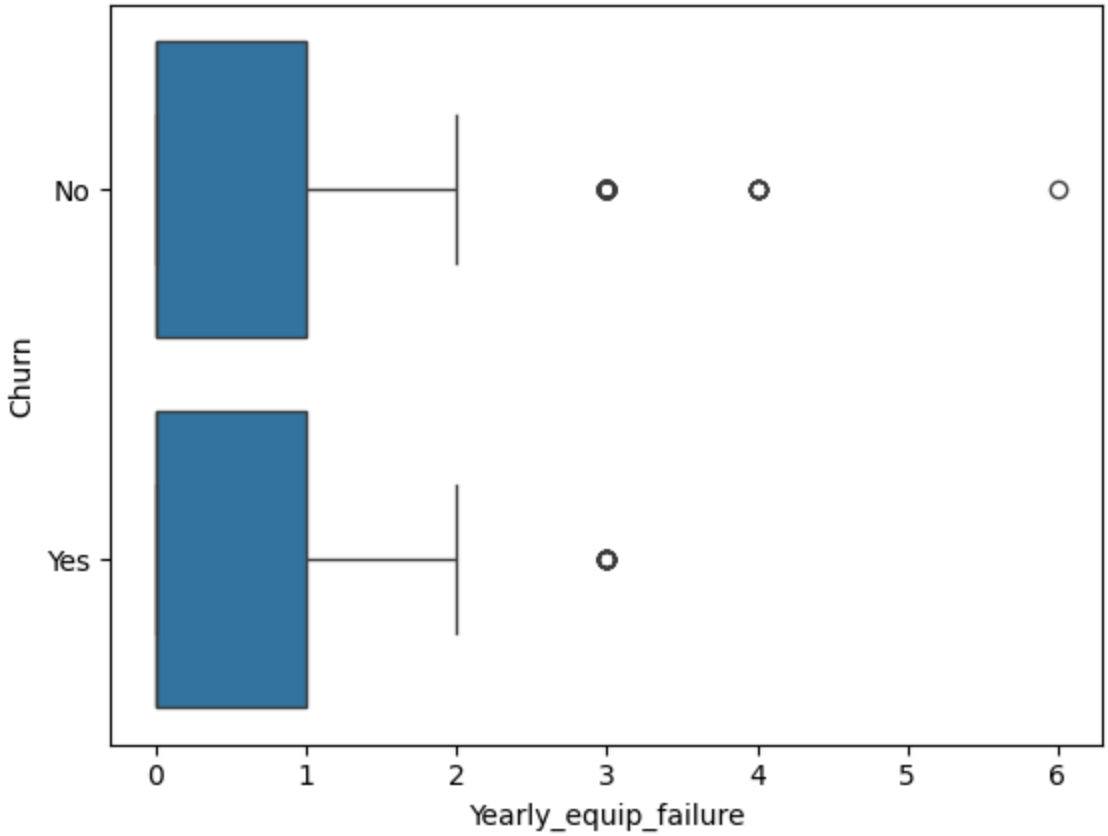
In [289…
```
sns.boxplot(x='Contacts', y='Churn', data=df)
```

Out[289…
```
<Axes: xlabel='Contacts', ylabel='Churn'>
```



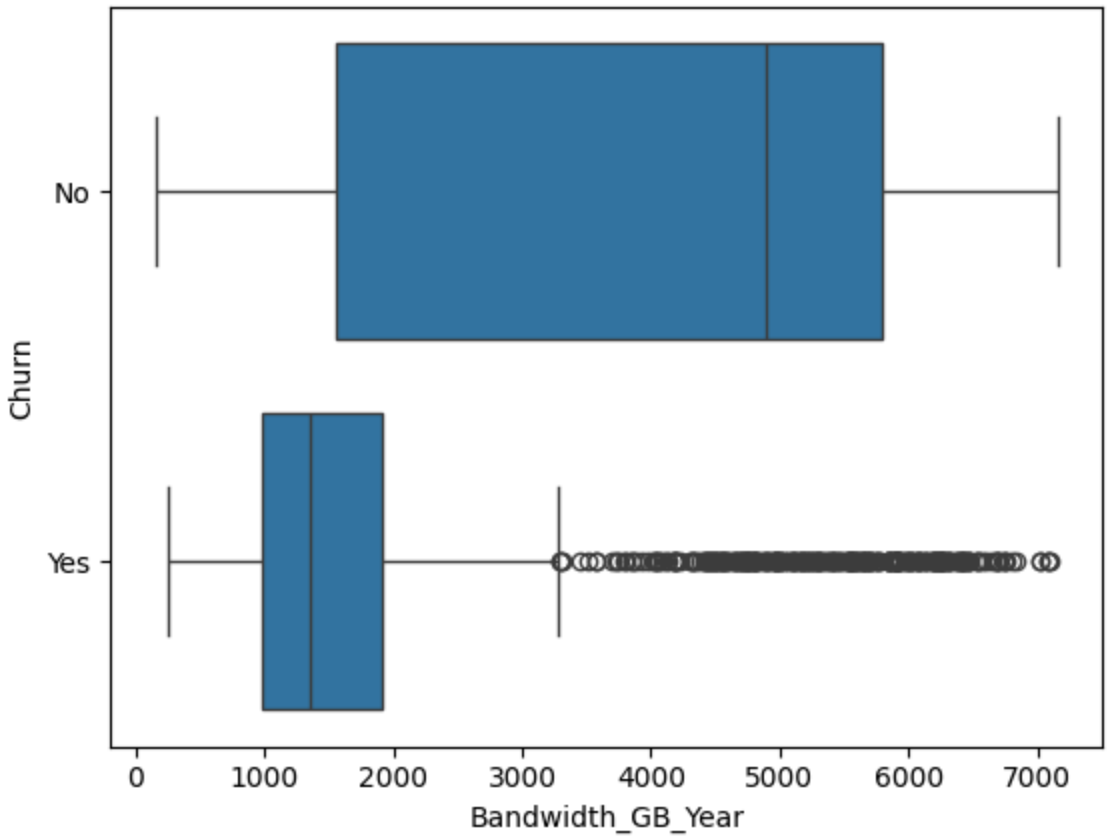
In [290…
```
sns.boxplot(x='Yearly_equip_failure', y='Churn', data=df)
```

Out[290…
```
<Axes: xlabel='Yearly_equip_failure', ylabel='Churn'>
```



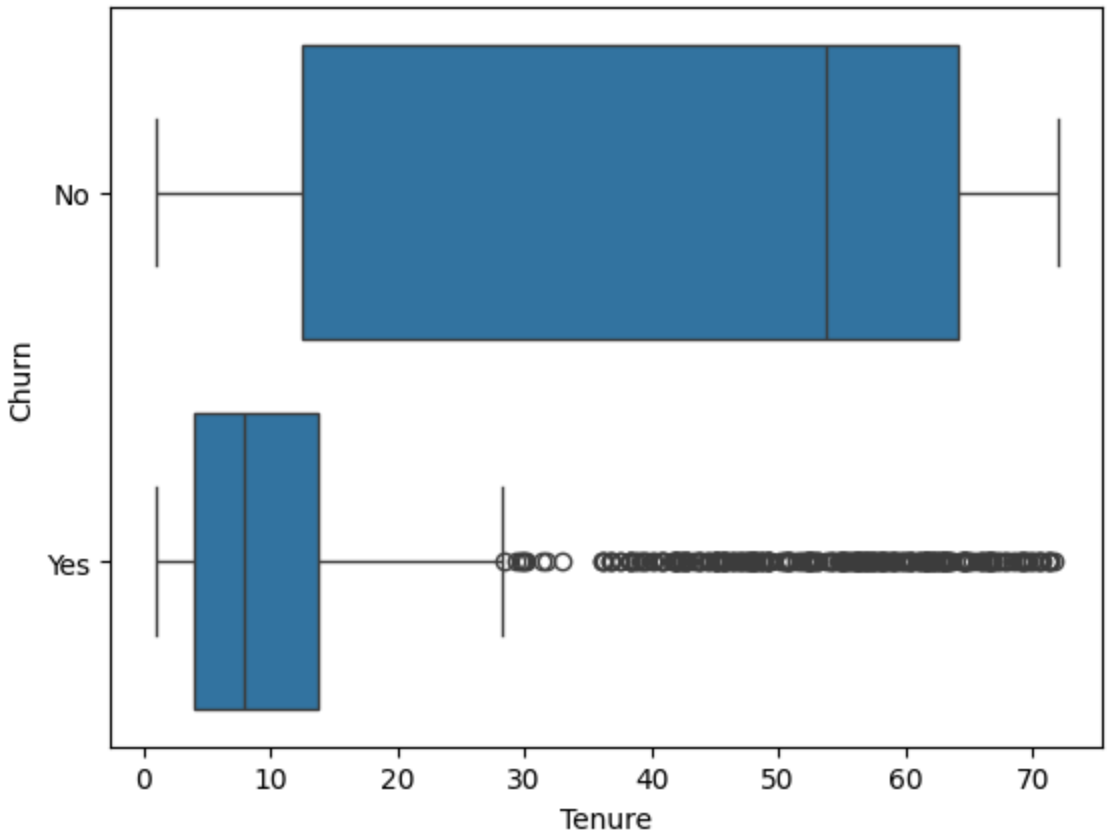
In [291…
```
sns.boxplot(x='Bandwidth_GB_Year', y='Churn', data=df)
```

Out[291…
```
<Axes: xlabel='Bandwidth_GB_Year', ylabel='Churn'>
```

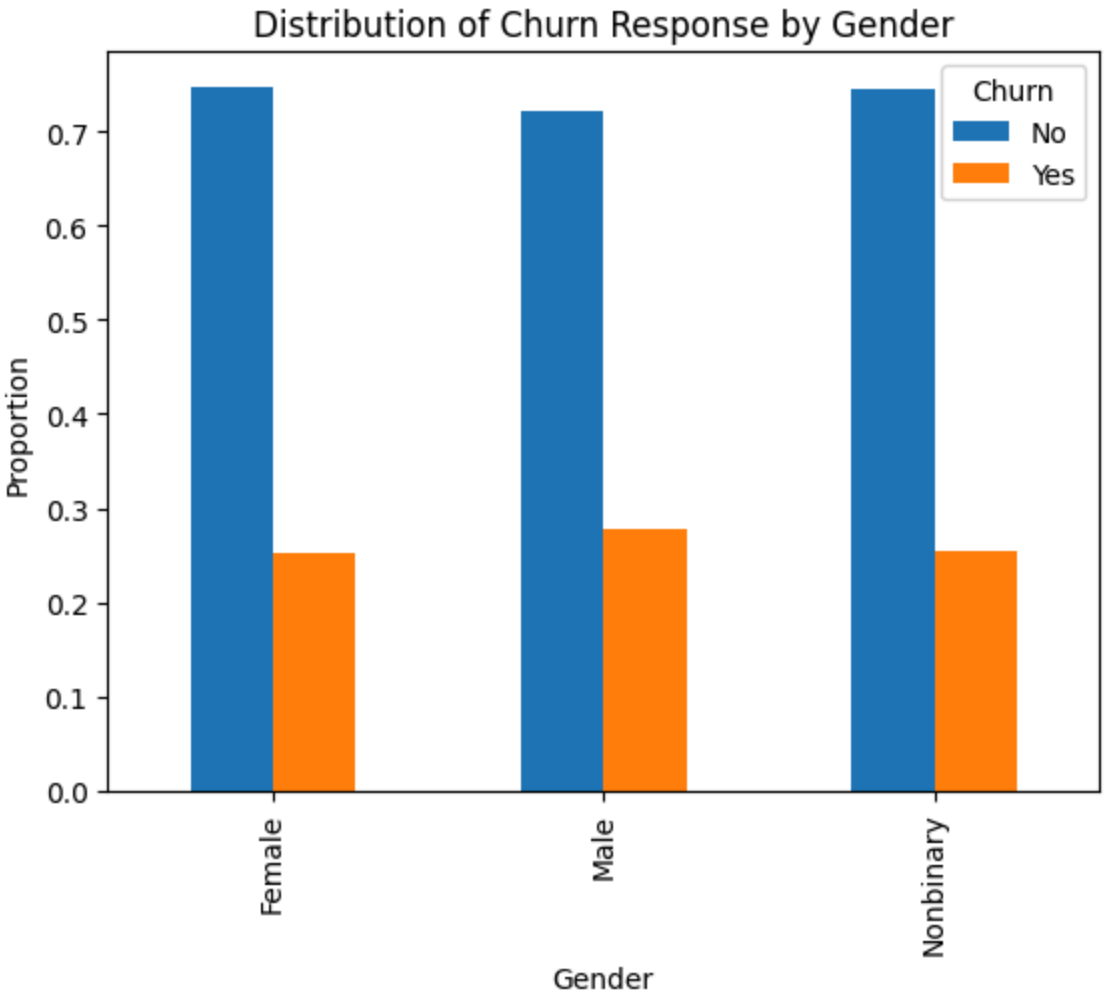


In [292…
```
sns.boxplot(x='Tenure', y='Churn', data=df)
```

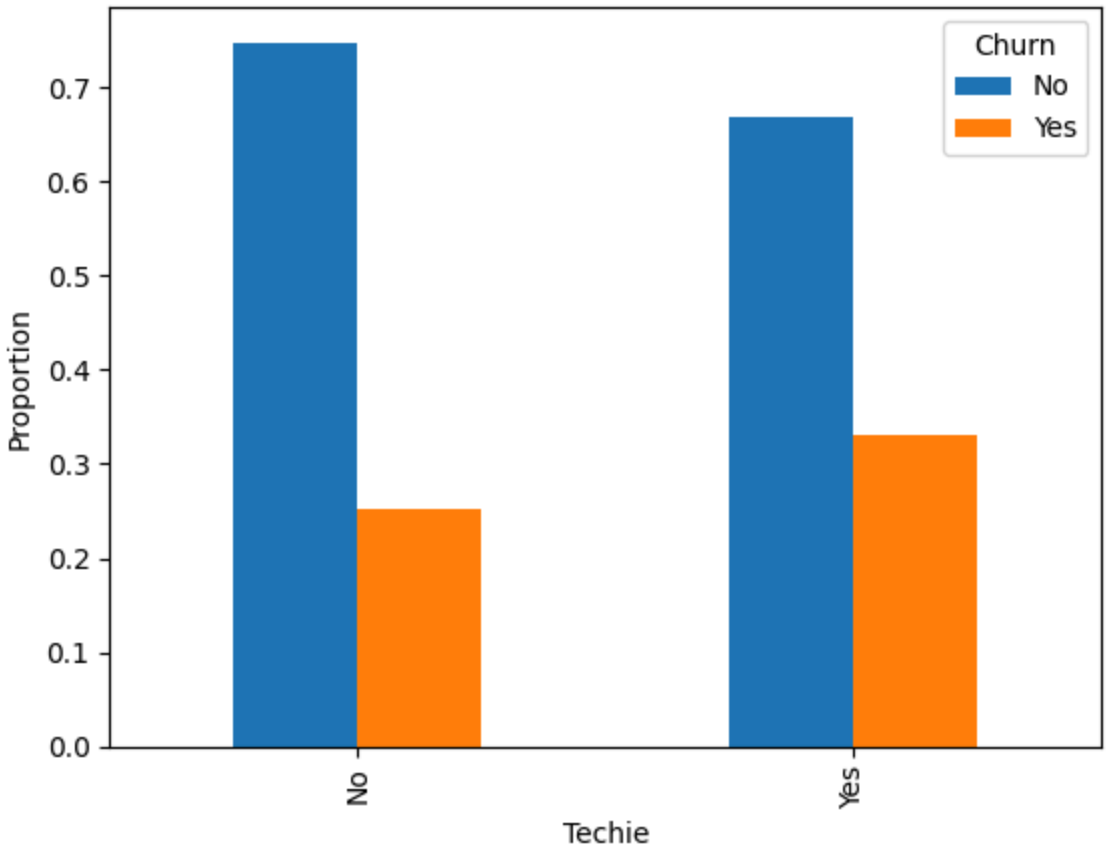
Out[292…
```
<Axes: xlabel='Tenure', ylabel='Churn'>
```



In [293…
```
df.groupby('Gender')['Churn'].value_counts(normalize=True).unstack().plot.bar(stack
plt.ylabel('Proportion')
plt.title('Distribution of Churn Response by Gender')
plt.show()
```

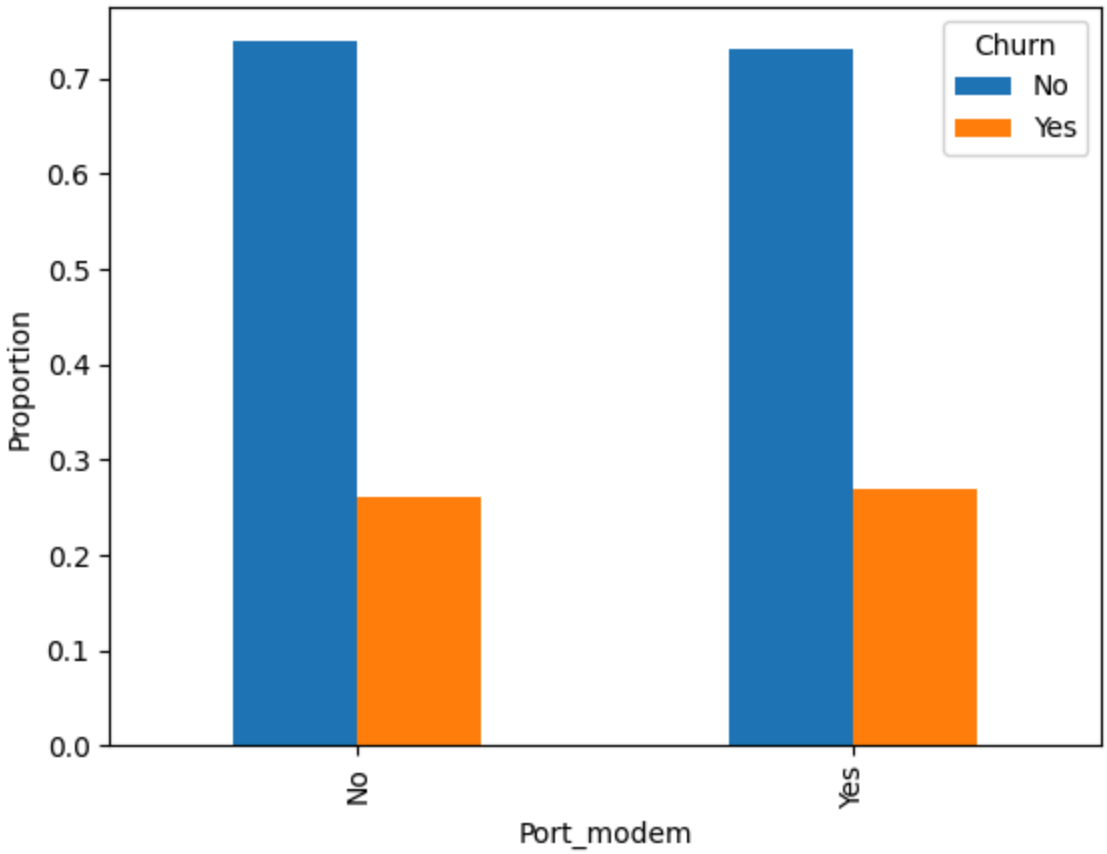


In [294…
```
df.groupby('Techie')['Churn'].value_counts(normalize=True).unstack().plot.bar(stack
plt.ylabel('Proportion')
plt.show()
```



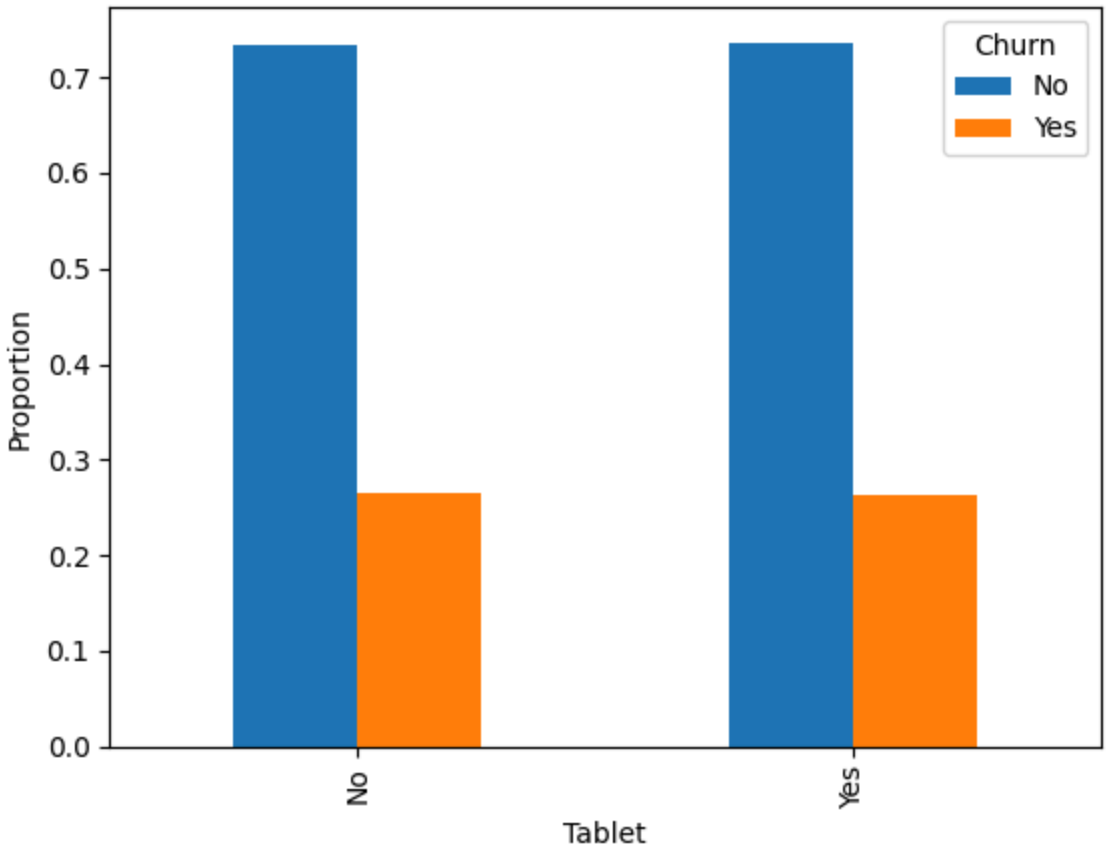
In [295…

```
df.groupby('Port_modem')['Churn'].value_counts(normalize=True).unstack().plot.bar(s
plt.ylabel('Proportion')
plt.show()
```



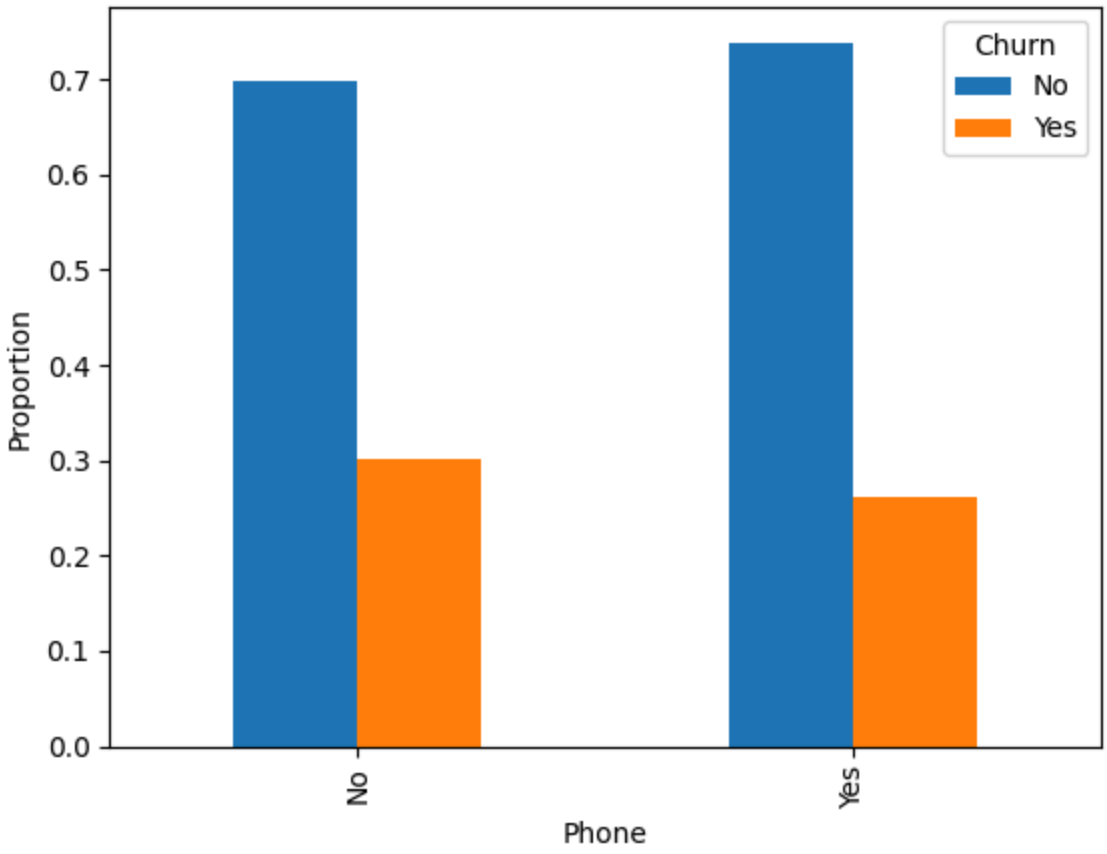
In [296…

```
df.groupby('Tablet')['Churn'].value_counts(normalize=True).unstack().plot.bar(stack
plt.ylabel('Proportion')
plt.show()
```



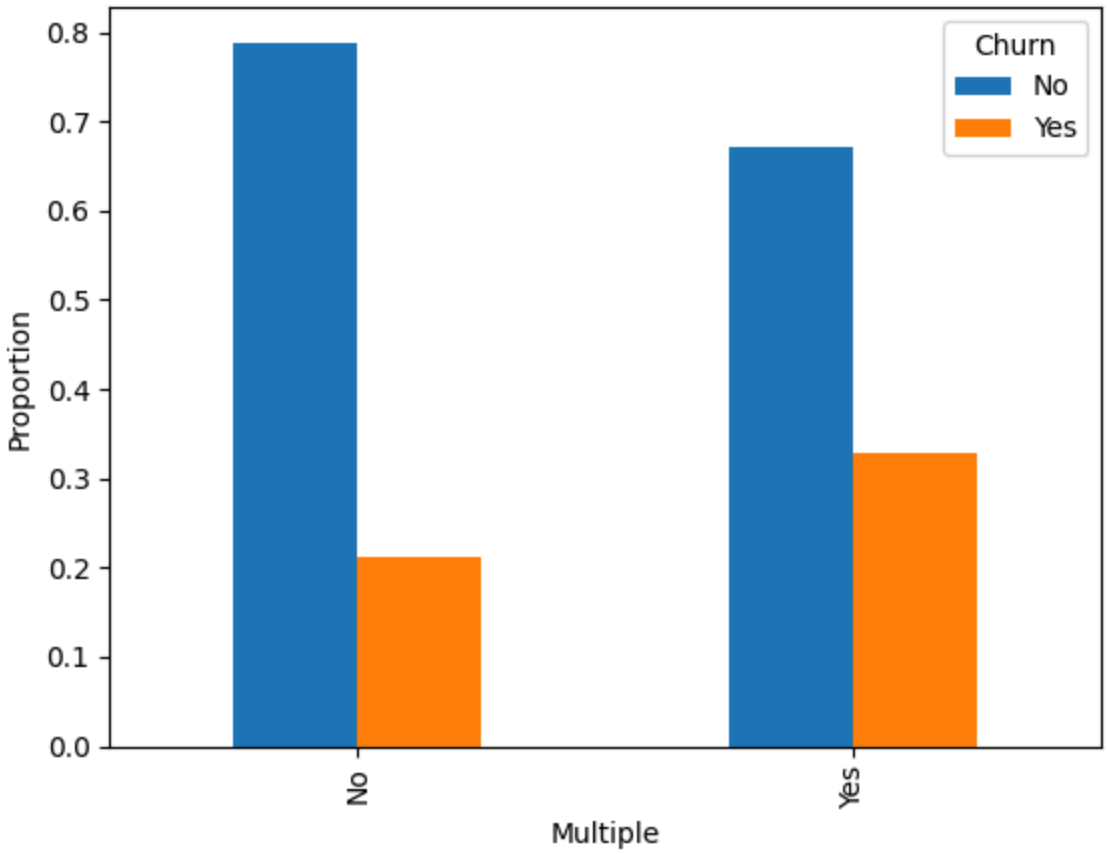
In [297…

```
df.groupby('Phone')['Churn'].value_counts(normalize=True).unstack().plot.bar(stacke
plt.ylabel('Proportion')
plt.show()
```



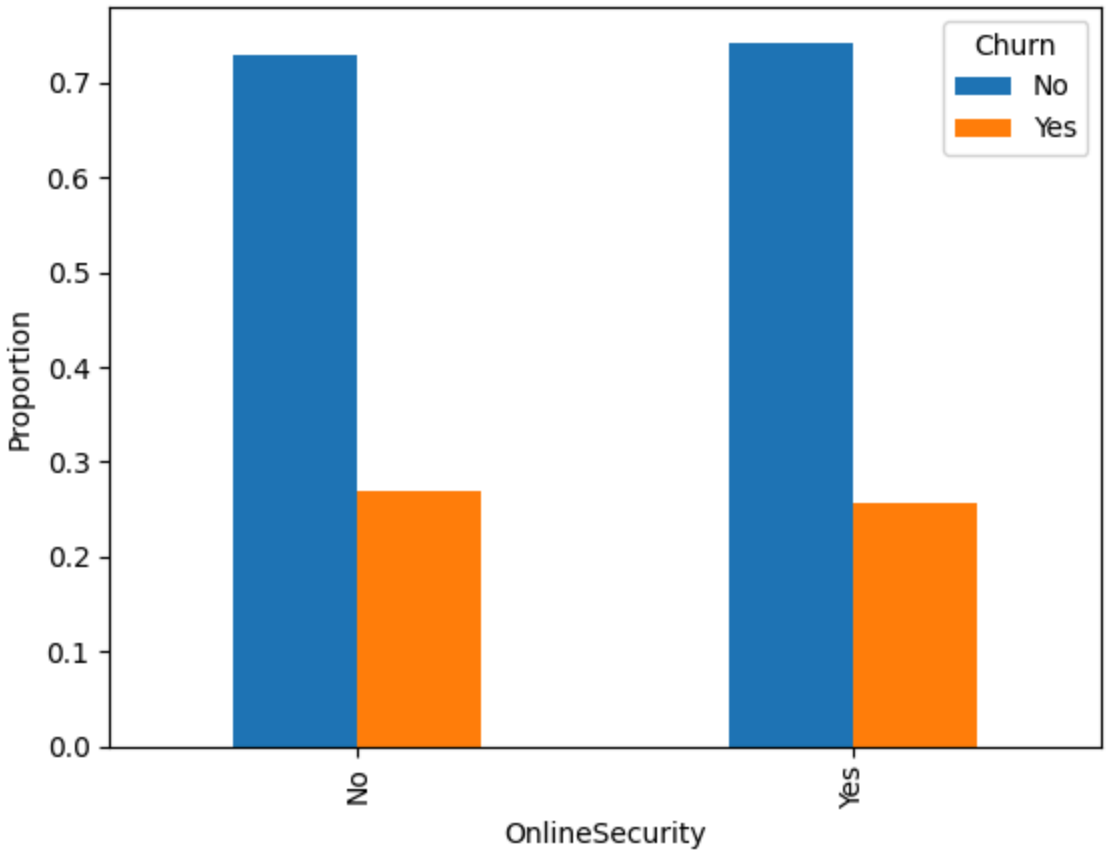
In [298…

```
df.groupby('Multiple')['Churn'].value_counts(normalize=True).unstack().plot.bar(sta
plt.ylabel('Proportion')
plt.show()
```



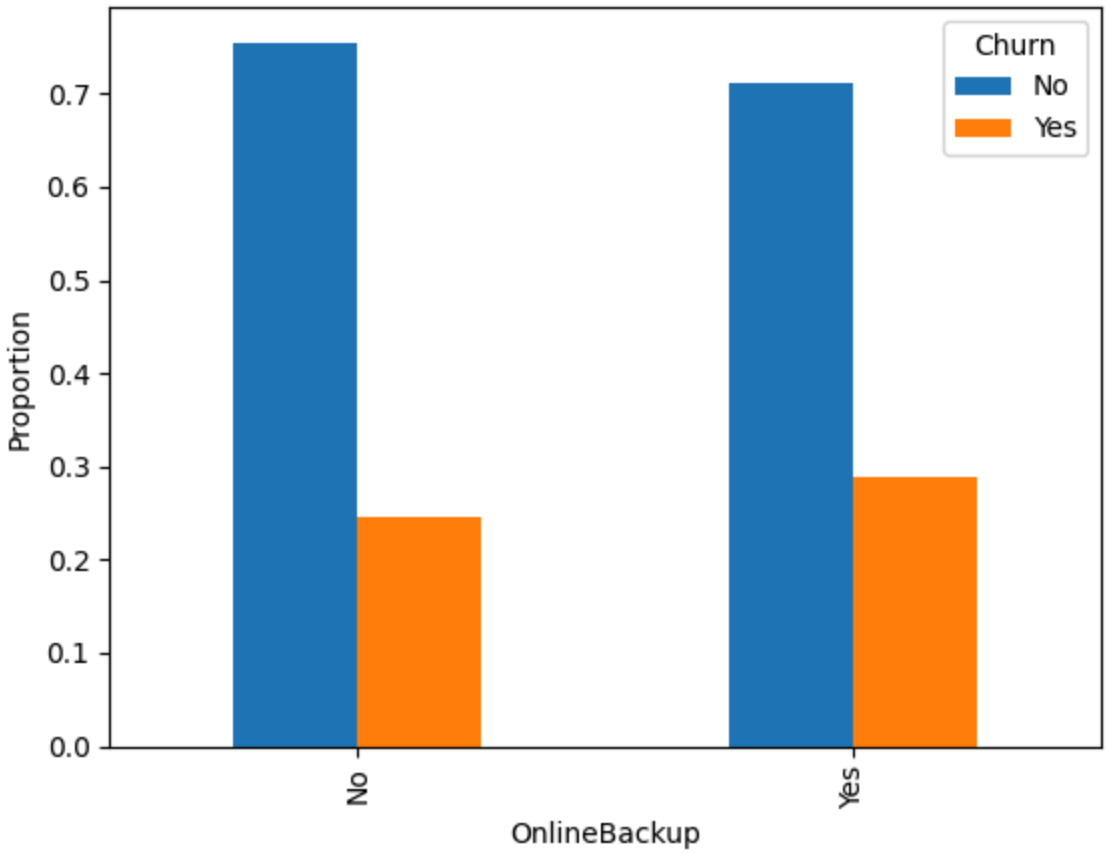
In [299…

```
df.groupby('OnlineSecurity')['Churn'].value_counts(normalize=True).unstack().plot.b
plt.ylabel('Proportion')
plt.show()
```



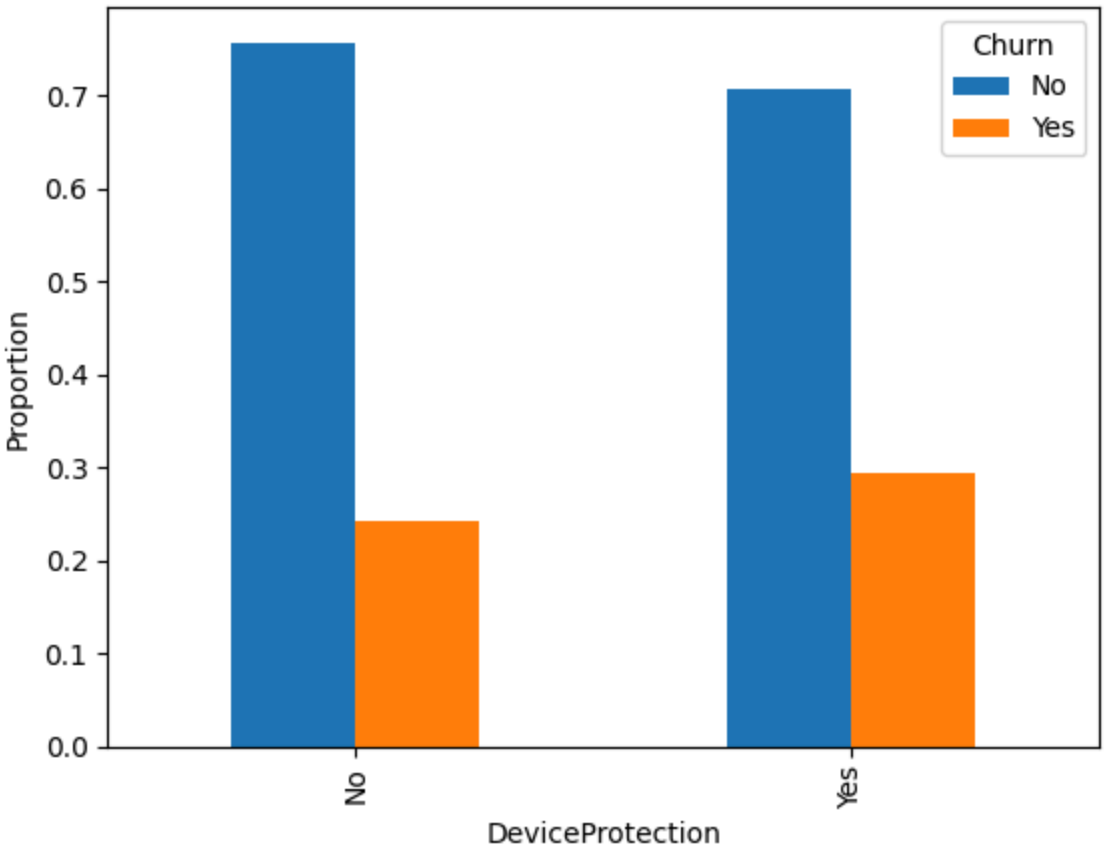
In [300…

```
df.groupby('OnlineBackup')['Churn'].value_counts(normalize=True).unstack().plot.bar
plt.ylabel('Proportion')
plt.show()
```



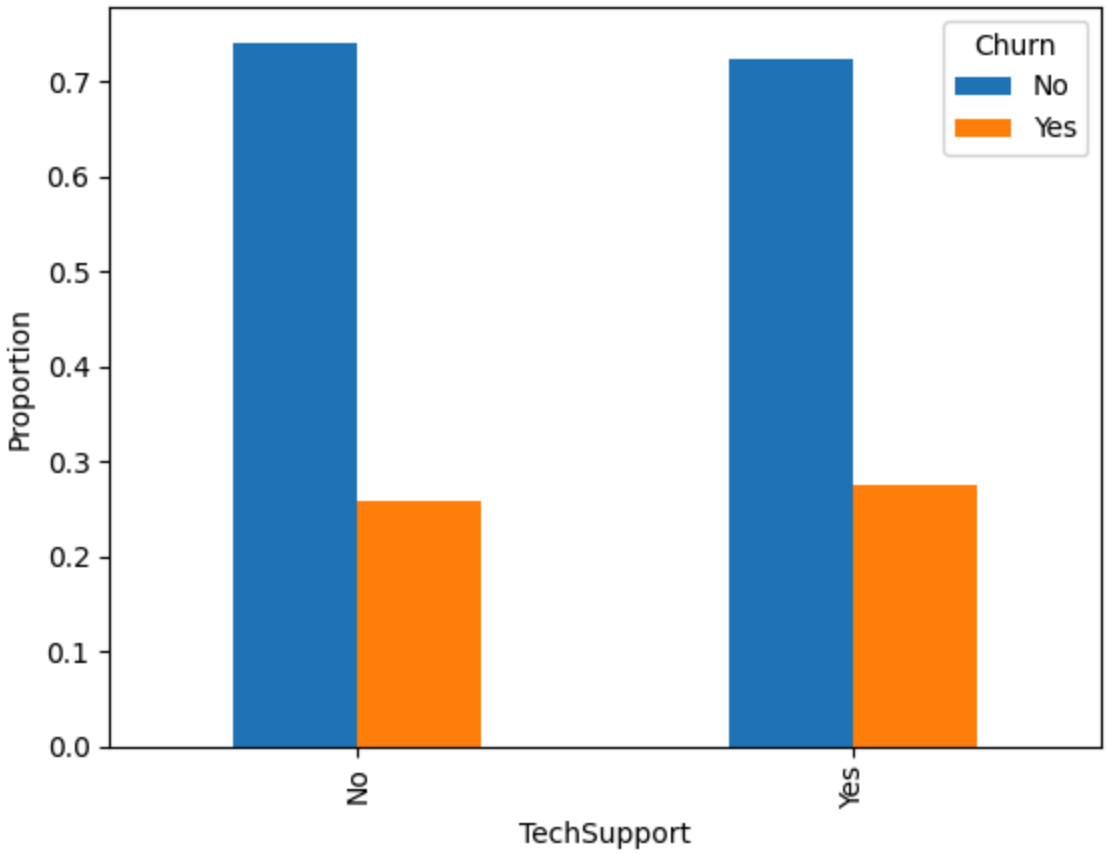
In [301…

```
df.groupby('DeviceProtection')['Churn'].value_counts(normalize=True).unstack().plot
plt.ylabel('Proportion')
plt.show()
```



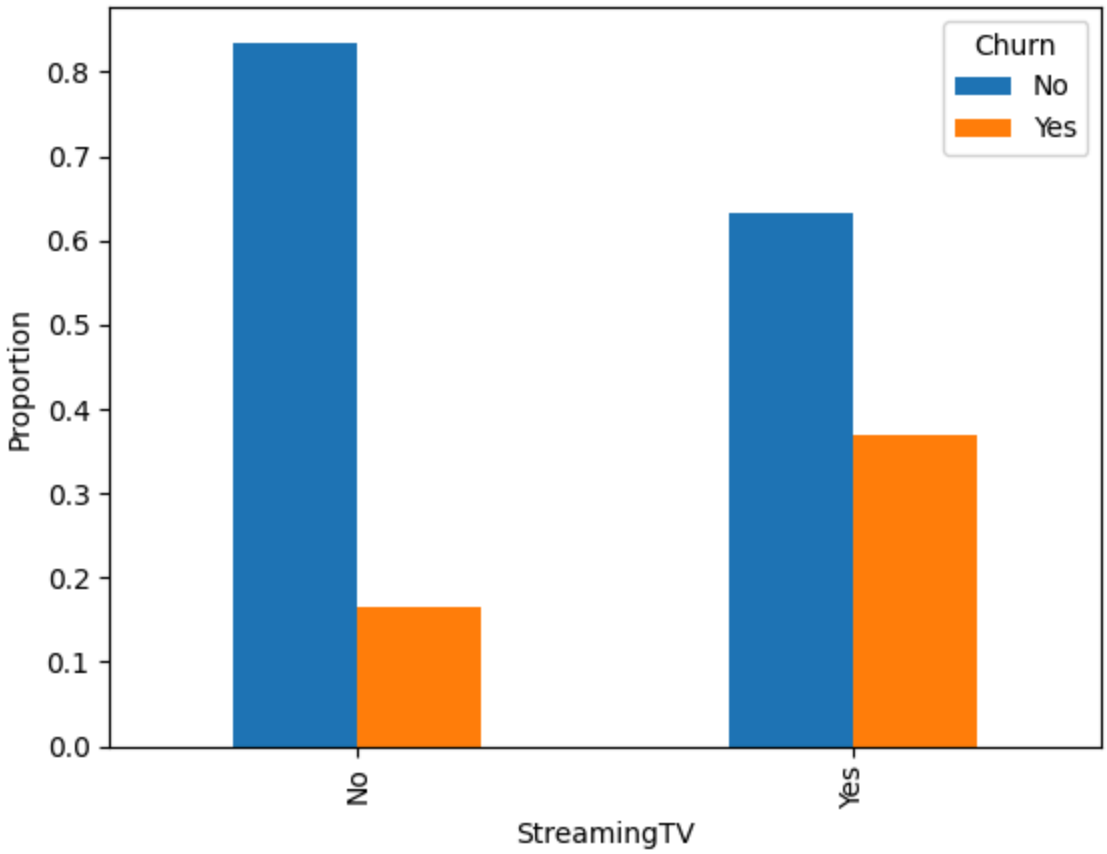
In [302…

```
df.groupby('TechSupport')['Churn'].value_counts(normalize=True).unstack().plot.bar(
plt.ylabel('Proportion')
plt.show()
```



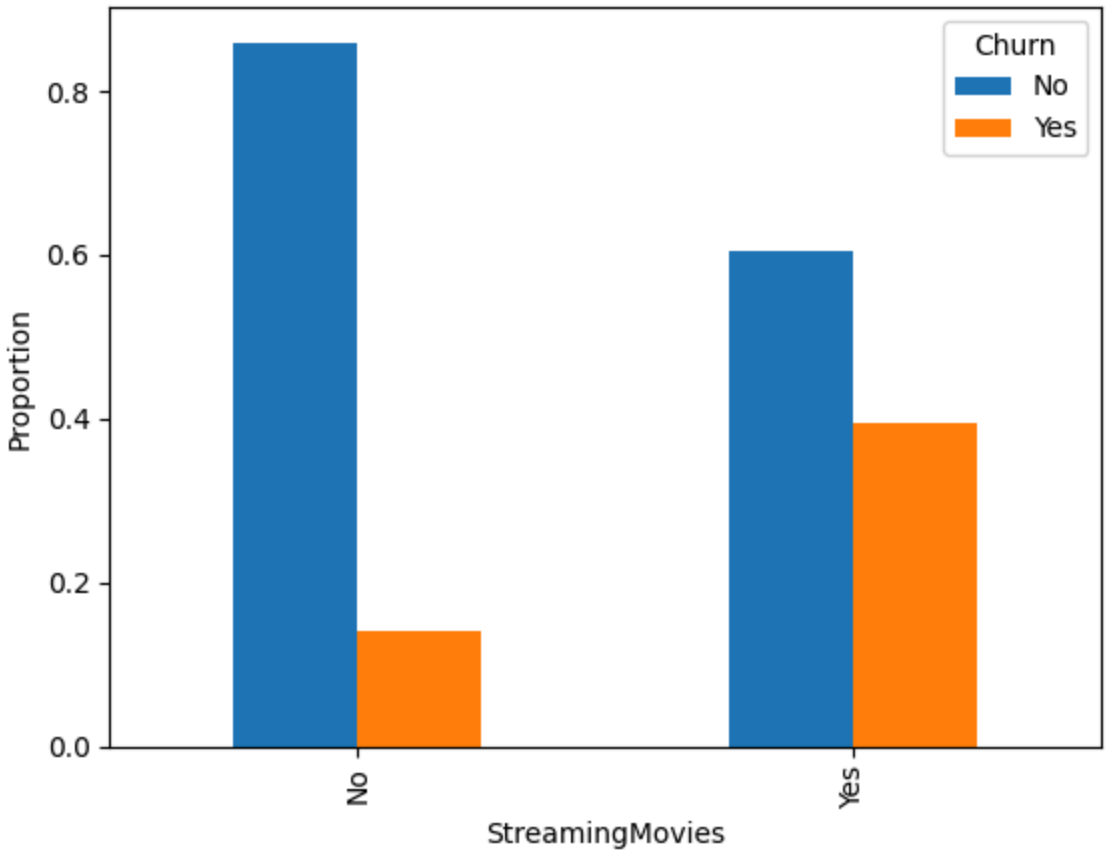
In [303…

```
df.groupby('StreamingTV')['Churn'].value_counts(normalize=True).unstack().plot.bar(
plt.ylabel('Proportion')
plt.show()
```



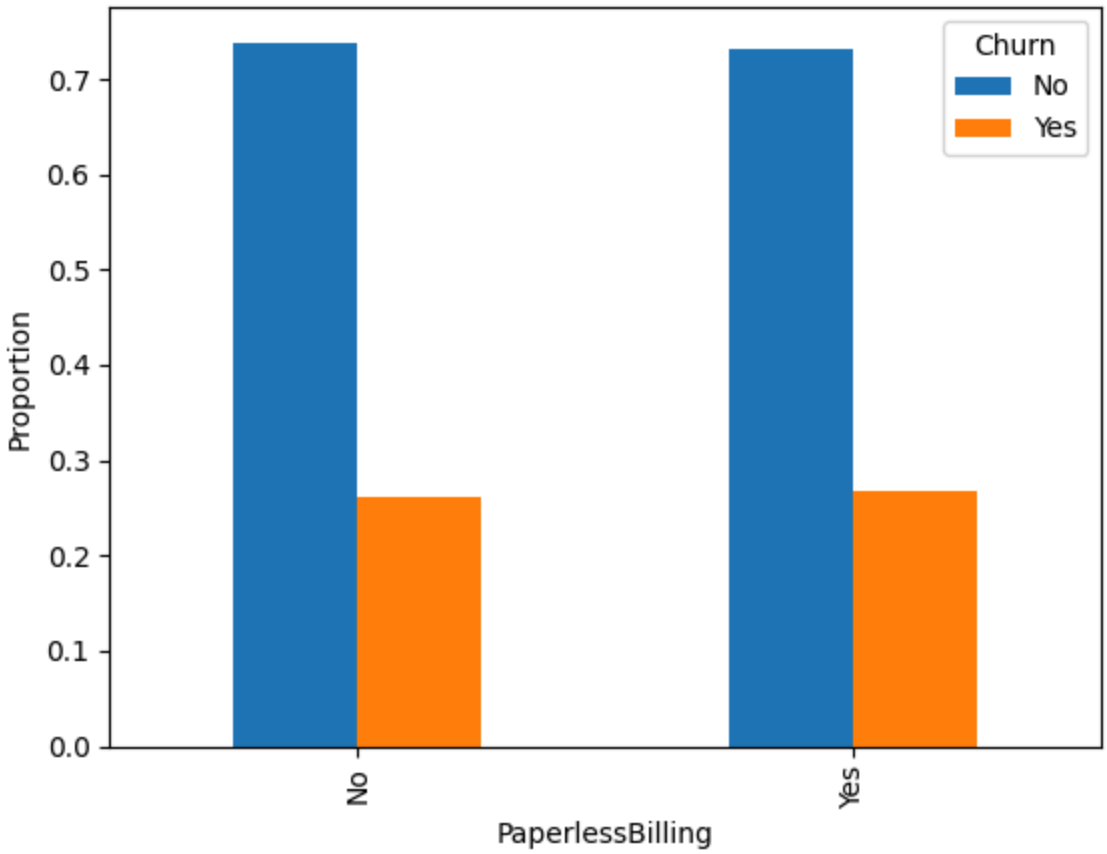
In [304…

```
df.groupby('StreamingMovies')['Churn'].value_counts(normalize=True).unstack().plot.
plt.ylabel('Proportion')
plt.show()
```



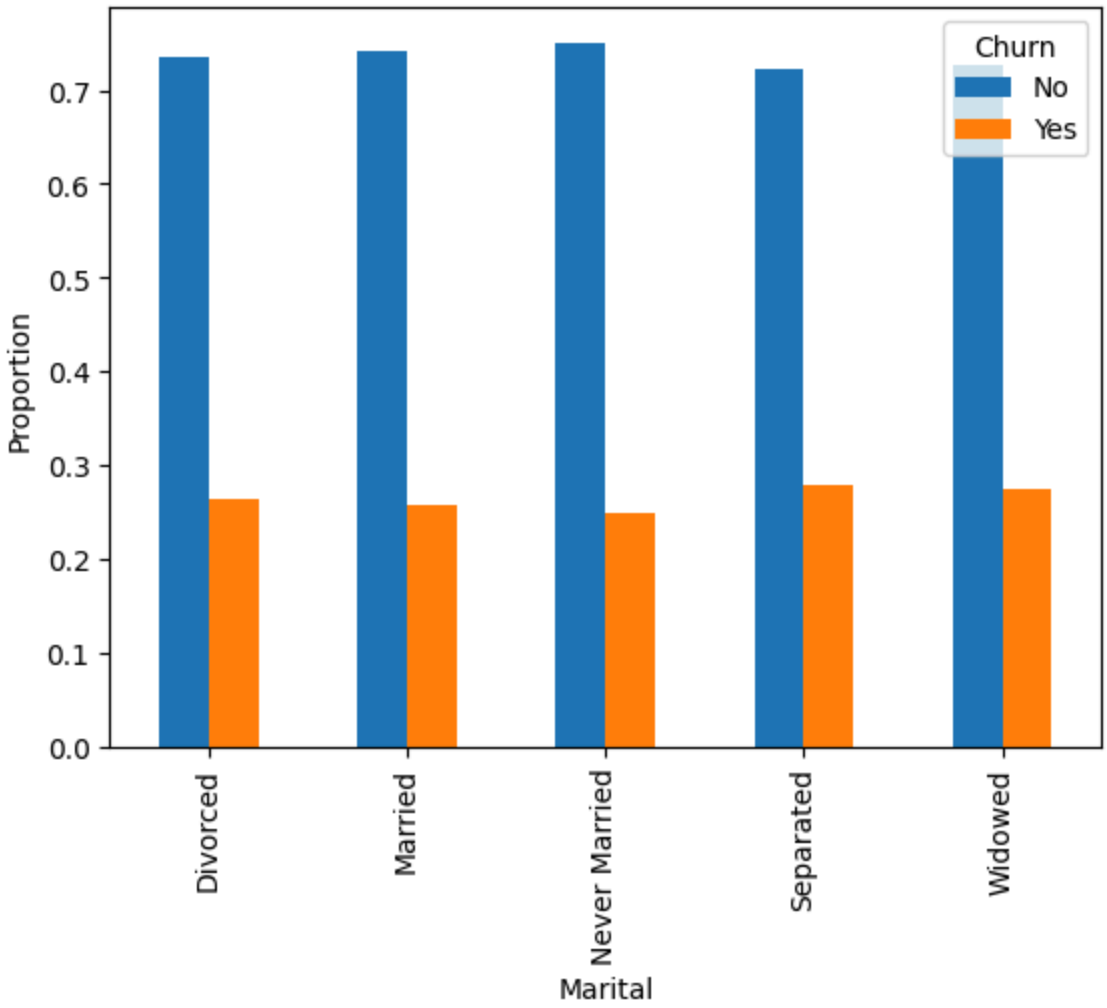
In [305…

```
df.groupby('PaperlessBilling')['Churn'].value_counts(normalize=True).unstack().plot
plt.ylabel('Proportion')
plt.show()
```



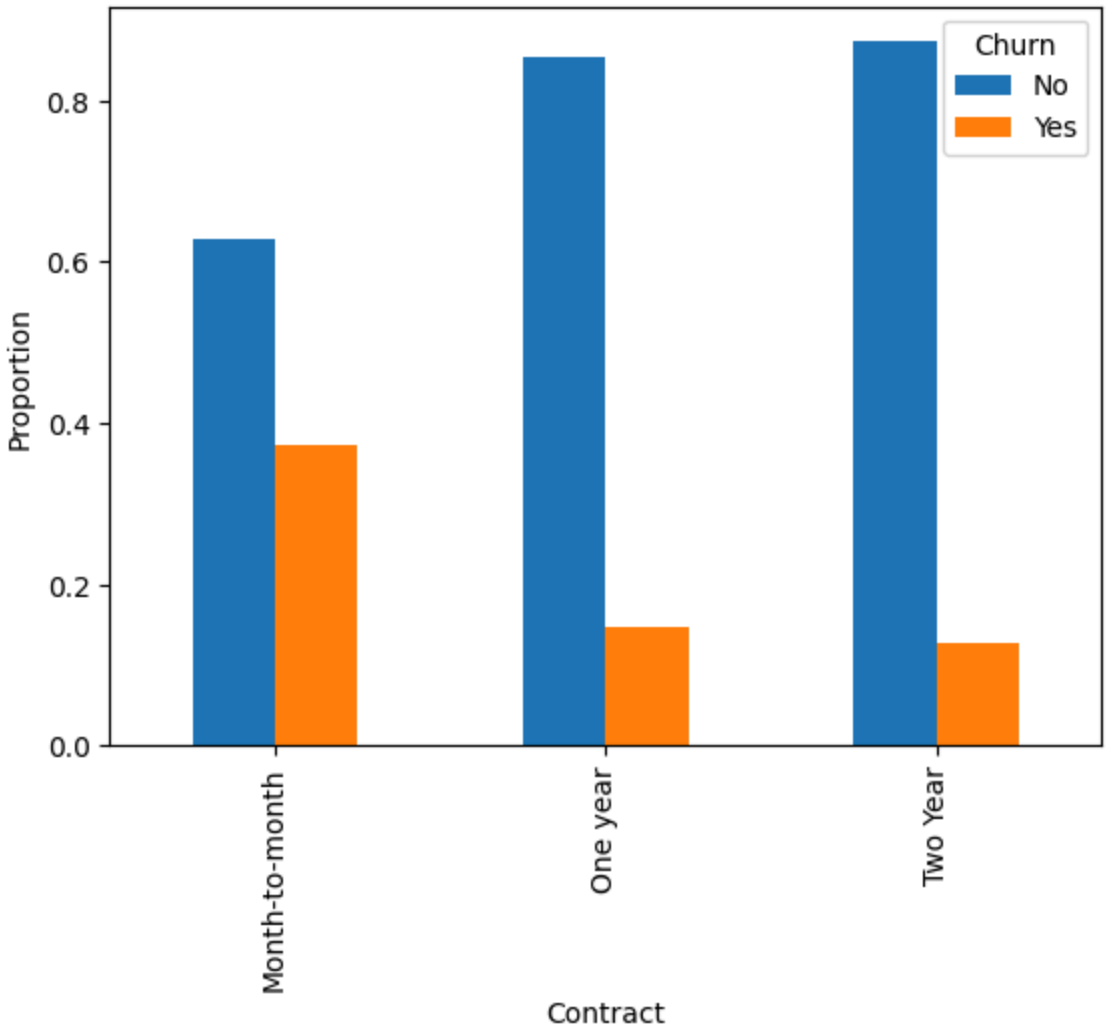
In [306…

```
df.groupby('Marital')['Churn'].value_counts(normalize=True).unstack().plot.bar(stac
plt.ylabel('Proportion')
plt.show()
```



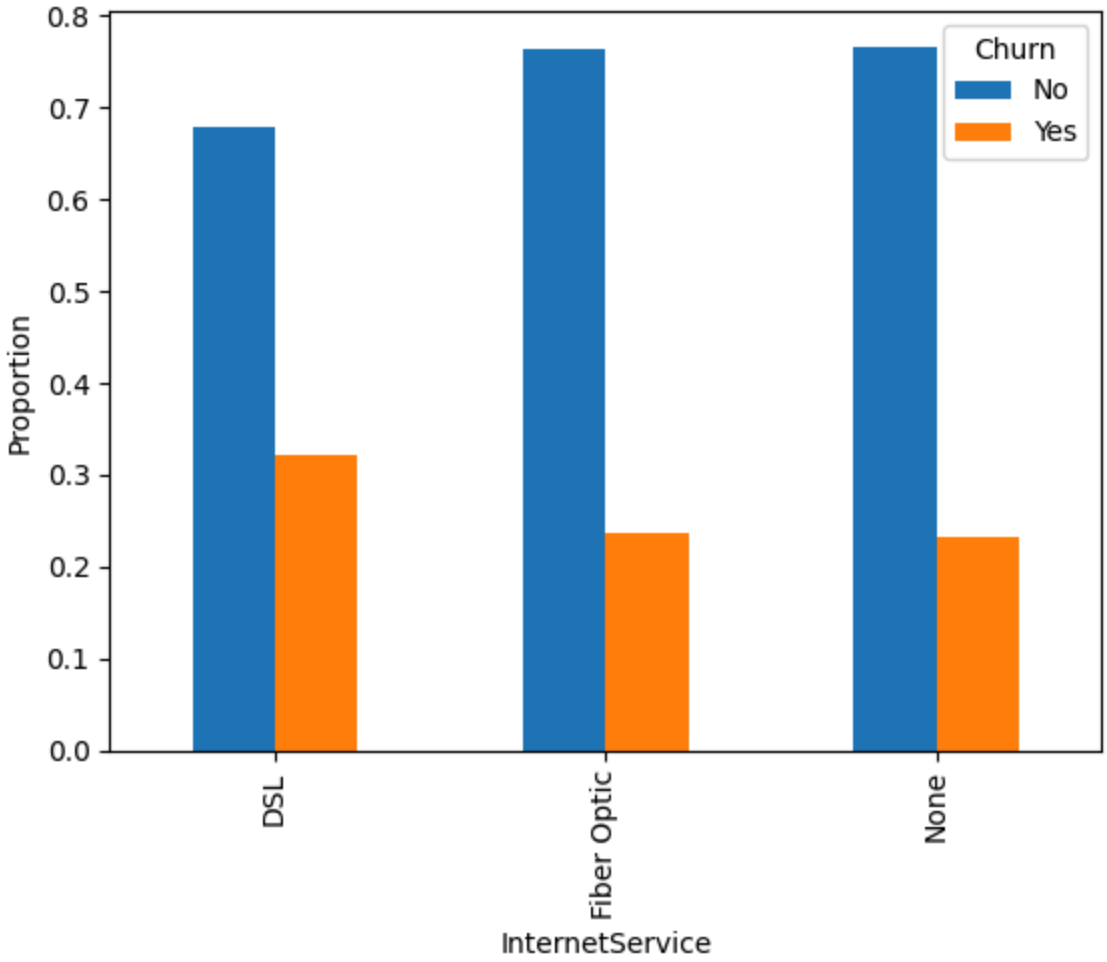
In [307…

```
df.groupby('Contract')['Churn'].value_counts(normalize=True).unstack().plot.bar(sta
plt.ylabel('Proportion')
plt.show()
```



In [308…

```
df.groupby('InternetService')['Churn'].value_counts(normalize=True).unstack().plot.
plt.ylabel('Proportion')
plt.show()
```



In [309…

```
df['Churn'] = df['Churn'].replace({'Yes': 1, 'No': 0}) #Replacing yes with 1 and no
df['Techie'] = df['Techie'].replace({'Yes': 1, 'No': 0})
df['Port_modem'] = df['Port_modem'].replace({'Yes': 1, 'No': 0})
df['Tablet'] = df['Tablet'].replace({'Yes': 1, 'No': 0})
df['Phone'] = df['Phone'].replace({'Yes': 1, 'No': 0})
df['Multiple'] = df['Multiple'].replace({'Yes': 1, 'No': 0})
df['OnlineSecurity'] = df['OnlineSecurity'].replace({'Yes': 1, 'No': 0})
df['OnlineBackup'] = df['OnlineBackup'].replace({'Yes': 1, 'No': 0})
df['DeviceProtection'] = df['DeviceProtection'].replace({'Yes': 1, 'No': 0})
df['TechSupport'] = df['TechSupport'].replace({'Yes': 1, 'No': 0})
df['StreamingMovies'] = df['StreamingMovies'].replace({'Yes': 1, 'No': 0})
df['StreamingTV'] = df['StreamingTV'].replace({'Yes': 1, 'No': 0})
df['PaperlessBilling'] = df['PaperlessBilling'].replace({'Yes': 1, 'No': 0})
```

```
C:\Users\Cali\AppData\Local\Temp\ipykernel_9136\977928063.py:1: FutureWarning: Downc
asting behavior in `replace` is deprecated and will be removed in a future version.
To retain the old behavior, explicitly call `result.infer_objects(copy=False)`. To o
pt-in to the future behavior, set `pd.set_option('future.no_silent_downcasting', Tru
e)`
  df['Churn'] = df['Churn'].replace({'Yes': 1, 'No': 0}) #Replacing yes with 1 and n
o with 0 for yes/no categorical variables
C:\Users\Cali\AppData\Local\Temp\ipykernel_9136\977928063.py:2: FutureWarning: Downc
asting behavior in `replace` is deprecated and will be removed in a future version.
To retain the old behavior, explicitly call `result.infer_objects(copy=False)`. To o
pt-in to the future behavior, set `pd.set_option('future.no_silent_downcasting', Tru
e)`
  df['Techie'] = df['Techie'].replace({'Yes': 1, 'No': 0})
C:\Users\Cali\AppData\Local\Temp\ipykernel_9136\977928063.py:3: FutureWarning: Downc
asting behavior in `replace` is deprecated and will be removed in a future version.
To retain the old behavior, explicitly call `result.infer_objects(copy=False)`. To o
pt-in to the future behavior, set `pd.set_option('future.no_silent_downcasting', Tru
e)`
  df['Port_modem'] = df['Port_modem'].replace({'Yes': 1, 'No': 0})
C:\Users\Cali\AppData\Local\Temp\ipykernel_9136\977928063.py:4: FutureWarning: Downc
asting behavior in `replace` is deprecated and will be removed in a future version.
To retain the old behavior, explicitly call `result.infer_objects(copy=False)`. To o
pt-in to the future behavior, set `pd.set_option('future.no_silent_downcasting', Tru
e)`
  df['Tablet'] = df['Tablet'].replace({'Yes': 1, 'No': 0})
C:\Users\Cali\AppData\Local\Temp\ipykernel_9136\977928063.py:5: FutureWarning: Downc
asting behavior in `replace` is deprecated and will be removed in a future version.
To retain the old behavior, explicitly call `result.infer_objects(copy=False)`. To o
pt-in to the future behavior, set `pd.set_option('future.no_silent_downcasting', Tru
e)`
  df['Phone'] = df['Phone'].replace({'Yes': 1, 'No': 0})
C:\Users\Cali\AppData\Local\Temp\ipykernel_9136\977928063.py:6: FutureWarning: Downc
asting behavior in `replace` is deprecated and will be removed in a future version.
To retain the old behavior, explicitly call `result.infer_objects(copy=False)`. To o
pt-in to the future behavior, set `pd.set_option('future.no_silent_downcasting', Tru
e)`
  df['Multiple'] = df['Multiple'].replace({'Yes': 1, 'No': 0})
C:\Users\Cali\AppData\Local\Temp\ipykernel_9136\977928063.py:7: FutureWarning: Downc
asting behavior in `replace` is deprecated and will be removed in a future version.
To retain the old behavior, explicitly call `result.infer_objects(copy=False)`. To o
pt-in to the future behavior, set `pd.set_option('future.no_silent_downcasting', Tru
e)`
  df['OnlineSecurity'] = df['OnlineSecurity'].replace({'Yes': 1, 'No': 0})
C:\Users\Cali\AppData\Local\Temp\ipykernel_9136\977928063.py:8: FutureWarning: Downc
asting behavior in `replace` is deprecated and will be removed in a future version.
To retain the old behavior, explicitly call `result.infer_objects(copy=False)`. To o
pt-in to the future behavior, set `pd.set_option('future.no_silent_downcasting', Tru
e)`
  df['OnlineBackup'] = df['OnlineBackup'].replace({'Yes': 1, 'No': 0})
C:\Users\Cali\AppData\Local\Temp\ipykernel_9136\977928063.py:9: FutureWarning: Downc
asting behavior in `replace` is deprecated and will be removed in a future version.
To retain the old behavior, explicitly call `result.infer_objects(copy=False)`. To o
pt-in to the future behavior, set `pd.set_option('future.no_silent_downcasting', Tru
e)`
  df['DeviceProtection'] = df['DeviceProtection'].replace({'Yes': 1, 'No': 0})
C:\Users\Cali\AppData\Local\Temp\ipykernel_9136\977928063.py:10: FutureWarning: Down
```

```
casting behavior in `replace` is deprecated and will be removed in a future version.
To retain the old behavior, explicitly call `result.infer_objects(copy=False)`. To o
pt-in to the future behavior, set `pd.set_option('future.no_silent_downcasting', Tru
e)`
  df['TechSupport'] = df['TechSupport'].replace({'Yes': 1, 'No': 0})
C:\Users\Cali\AppData\Local\Temp\ipykernel_9136\977928063.py:11: FutureWarning: Down
casting behavior in `replace` is deprecated and will be removed in a future version.
To retain the old behavior, explicitly call `result.infer_objects(copy=False)`. To o
pt-in to the future behavior, set `pd.set_option('future.no_silent_downcasting', Tru
e)`
  df['StreamingMovies'] = df['StreamingMovies'].replace({'Yes': 1, 'No': 0})
C:\Users\Cali\AppData\Local\Temp\ipykernel_9136\977928063.py:12: FutureWarning: Down
casting behavior in `replace` is deprecated and will be removed in a future version.
To retain the old behavior, explicitly call `result.infer_objects(copy=False)`. To o
pt-in to the future behavior, set `pd.set_option('future.no_silent_downcasting', Tru
e)`
  df['StreamingTV'] = df['StreamingTV'].replace({'Yes': 1, 'No': 0})
C:\Users\Cali\AppData\Local\Temp\ipykernel_9136\977928063.py:13: FutureWarning: Down
casting behavior in `replace` is deprecated and will be removed in a future version.
To retain the old behavior, explicitly call `result.infer_objects(copy=False)`. To o
pt-in to the future behavior, set `pd.set_option('future.no_silent_downcasting', Tru
e)`
  df['PaperlessBilling'] = df['PaperlessBilling'].replace({'Yes': 1, 'No': 0})
```

In [310…

```
#encoding for non-ordinal categorical variables
df_clean = pd.get_dummies(df, columns = ['Gender', 'Marital', 'Contract', 'Internet
df_clean = df_clean.rename(columns=lambda x: x.replace(' ', '_'))
print(df_clean.head(3))
```

```
   Children  Age    Income  Churn  Outage_sec_perweek  Email  Contacts  \
0         0   68  28561.99      0            7.978323     10         0
1         1   27  21704.77      1           11.699080     12         0
2         4   50   9609.57      0           10.752800      9         0

   Yearly_equip_failure  Techie  Port_modem  ...  dmy_Male  dmy_Nonbinary  \
0                     1       0           1  ...         1              0
1                     1       1           0  ...         0              0
2                     1       1           1  ...         0              0

   dmy_Married  dmy_Never_Married  dmy_Separated  dmy_Widowed  dmy_One_year  \
0            0                  0              0            1             1
1            1                  0              0            0             0
2            0                  0              0            1             0

   dmy_Two_Year  dmy_Fiber_Optic  dmy_None
0             0                1         0
1             0                1         0
2             1                0         0

[3 rows x 33 columns]
```

In [311…

```
#Initial Logistic Regression Model
import statsmodels.api as sm
y = df_clean['Churn']
X = df_clean[['MonthlyCharge','Children', 'Age', 'Income', 'Outage_sec_perweek', 'E
    'Yearly_equip_failure', 'Techie', 'Port_modem', 'Tablet', 'Phone', 'Multiple',
```

```
    'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'Streaming
    'StreamingMovies', 'PaperlessBilling', 'Tenure', 'Bandwidth_GB_Year', 'dmy_Male
    'dmy_Nonbinary', 'dmy_Married', 'dmy_Never_Married', 'dmy_Separated', 'dmy_Wido
    'dmy_One_year', 'dmy_Two_Year', 'dmy_Fiber_Optic', 'dmy_None'
   ]]
X = sm.add_constant(X)
churn_logit_model = sm.Logit(y, X).fit()
print(churn_logit_model.params)
print(churn_logit_model.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.219251
         Iterations 9
const                  -4.895584e+00
MonthlyCharge           4.009128e-02
Children                1.204473e-02
Age                     7.861768e-04
Income                  6.596053e-07
Outage_sec_perweek     -1.783531e-03
Email                  -8.201071e-03
Contacts                5.610634e-02
Yearly_equip_failure   -3.535594e-02
Techie                  1.089206e+00
Port_modem              1.364907e-01
Tablet                 -4.897958e-02
Phone                  -2.870376e-01
Multiple                3.460300e-01
OnlineSecurity         -2.464721e-01
OnlineBackup           -9.915757e-02
DeviceProtection       -7.363981e-02
TechSupport            -2.291598e-01
StreamingTV             1.173877e+00
StreamingMovies         1.315509e+00
PaperlessBilling        1.599347e-01
Tenure                 -1.220300e-01
Bandwidth_GB_Year       8.474127e-05
dmy_Male                2.658920e-01
dmy_Nonbinary          -8.803457e-02
dmy_Married             1.131029e-01
dmy_Never_Married       1.503989e-02
dmy_Separated           1.331966e-01
dmy_Widowed             2.606359e-01
dmy_One_year           -3.387243e+00
dmy_Two_Year           -3.477657e+00
dmy_Fiber_Optic        -2.140716e+00
dmy_None               -9.027119e-01
dtype: float64
                           Logit Regression Results
==============================================================================
Dep. Variable:                  Churn   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9967
Method:                           MLE   Df Model:                           32
Date:                Mon, 15 Jul 2024   Pseudo R-squ.:                  0.6208
Time:                        12:34:11   Log-Likelihood:                -2192.5
converged:                       True   LL-Null:                       -5782.2
Covariance Type:            nonrobust   LLR p-value:                     0.000
==============================================================================
====
                           coef    std err          z      P>|z|      [0.025      0.
975]
------------------------------------------------------------------------------
----
const                   -4.8956      1.459     -3.356      0.001      -7.754      -
2.037
MonthlyCharge            0.0401      0.014      2.918      0.004       0.013
0.067
```

```
Children                0.0120      0.136      0.088      0.930      -0.255
0.279
Age                     0.0008      0.015      0.054      0.957      -0.028
0.029
Income                6.596e-07   1.37e-06      0.482      0.629   -2.02e-06    3.34
e-06
Outage_sec_perweek      -0.0018      0.013     -0.138      0.891      -0.027
0.024
Email                   -0.0082      0.013     -0.649      0.516      -0.033
0.017
Contacts                 0.0561      0.039      1.452      0.146      -0.020
0.132
Yearly_equip_failure    -0.0354      0.061     -0.582      0.560      -0.154
0.084
Techie                   1.0892      0.102     10.668      0.000       0.889
1.289
Port_modem               0.1365      0.077      1.773      0.076      -0.014
0.287
Tablet                  -0.0490      0.084     -0.583      0.560      -0.214
0.116
Phone                   -0.2870      0.132     -2.182      0.029      -0.545      -
0.029
Multiple                 0.3460      0.200      1.727      0.084      -0.047
0.739
OnlineSecurity          -0.2465      0.310     -0.796      0.426      -0.853
0.360
OnlineBackup            -0.0992      0.180     -0.552      0.581      -0.451
0.253
DeviceProtection        -0.0736      0.232     -0.317      0.751      -0.529
0.381
TechSupport             -0.2292      0.172     -1.331      0.183      -0.567
0.108
StreamingTV              1.1739      0.506      2.318      0.020       0.181
2.166
StreamingMovies          1.3155      0.361      3.647      0.000       0.609
2.022
PaperlessBilling         0.1599      0.078      2.044      0.041       0.007
0.313
Tenure                  -0.1220      0.360     -0.339      0.735      -0.828
0.584
Bandwidth_GB_Year     8.474e-05      0.004      0.019      0.985      -0.009
0.009
dmy_Male                 0.2659      0.286      0.928      0.353      -0.296
0.827
dmy_Nonbinary           -0.0880      0.282     -0.312      0.755      -0.641
0.465
dmy_Married              0.1131      0.121      0.932      0.351      -0.125
0.351
dmy_Never_Married        0.0150      0.121      0.124      0.901      -0.223
0.253
dmy_Separated            0.1332      0.120      1.113      0.266      -0.101
0.368
dmy_Widowed              0.2606      0.120      2.175      0.030       0.026
0.495
dmy_One_year            -3.3872      0.128    -26.500      0.000      -3.638      -
3.137
```

```
dmy_Two_Year            -3.4777      0.126    -27.697      0.000      -3.724      -
3.232
dmy_Fiber_Optic         -2.1407      2.082     -1.028      0.304      -6.221
1.940
dmy_None                -0.9027      1.660     -0.544      0.587      -4.156
2.351
================================================================================
====
```

In [312…

```
# Checking the variance inflation factors to see if any variables should be elimina
from statsmodels.stats.outliers_influence import variance_inflation_factor

X = df_clean[['MonthlyCharge','Children', 'Age', 'Income', 'Outage_sec_perweek', 'E
    'Yearly_equip_failure', 'Techie', 'Port_modem', 'Tablet', 'Phone', 'Multiple',
    'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'Streaming
    'StreamingMovies', 'PaperlessBilling', 'Tenure', 'Bandwidth_GB_Year', 'dmy_Male
    'dmy_Nonbinary', 'dmy_Married', 'dmy_Never_Married', 'dmy_Separated', 'dmy_Wido
    'dmy_One_year', 'dmy_Two_Year', 'dmy_Fiber_Optic', 'dmy_None']]

vif_df = pd.DataFrame()
vif_df["feature"] = X.columns

vif_df["VIF"] = [variance_inflation_factor(X.values, i)
for i in range(len(X.columns))]

print(vif_df)
```

	feature	VIF
0	MonthlyCharge	1323.991893
1	Children	14.769253
2	Age	50.068970
3	Income	2.995079
4	Outage_sec_perweek	12.223377
5	Email	16.641678
6	Contacts	2.016130
7	Yearly_equip_failure	1.395189
8	Techie	1.205472
9	Port_modem	1.936395
10	Tablet	1.431834
11	Phone	10.664622
12	Multiple	10.094820
13	OnlineSecurity	3.798473
14	OnlineBackup	3.482868
15	DeviceProtection	2.660855
16	TechSupport	3.679731
17	StreamingTV	7.532098
18	StreamingMovies	11.641175
19	PaperlessBilling	2.437531
20	Tenure	18155.096929
21	Bandwidth_GB_Year	23223.766510
22	dmy_Male	5.179510
23	dmy_Nonbinary	1.055119
24	dmy_Married	1.914623
25	dmy_Never_Married	1.936711
26	dmy_Separated	1.961871
27	dmy_Widowed	1.971200
28	dmy_One_year	1.388923
29	dmy_Two_Year	1.451303
30	dmy_Fiber_Optic	166.315045
31	dmy_None	37.504894

In [313…

```
#Looks like there is a lot of multicollinearity issues. I will have to remove one b
#Removing variable with highest VIF first: Bandwidth_GB_Year
X = df_clean[['MonthlyCharge','Children', 'Age', 'Income', 'Outage_sec_perweek', 'E
    'Yearly_equip_failure', 'Techie', 'Port_modem', 'Tablet', 'Phone', 'Multiple',
    'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'Streaming
    'StreamingMovies', 'PaperlessBilling', 'Tenure', 'dmy_Male',
    'dmy_Nonbinary', 'dmy_Married', 'dmy_Never_Married', 'dmy_Separated', 'dmy_Wido
    'dmy_One_year', 'dmy_Two_Year', 'dmy_Fiber_Optic', 'dmy_None']]

vif_df = pd.DataFrame()
vif_df["feature"] = X.columns

vif_df["VIF"] = [variance_inflation_factor(X.values, i)
for i in range(len(X.columns))]

print(vif_df)
```

	feature	VIF
0	MonthlyCharge	171.742454
1	Children	1.939833
2	Age	7.310630
3	Income	2.961142
4	Outage_sec_perweek	11.445883
5	Email	15.118184
6	Contacts	2.005128
7	Yearly_equip_failure	1.390891
8	Techie	1.204451
9	Port_modem	1.927069
10	Tablet	1.428597
11	Phone	9.994437
12	Multiple	4.909622
13	OnlineSecurity	1.592534
14	OnlineBackup	3.313111
15	DeviceProtection	2.272893
16	TechSupport	2.020461
17	StreamingTV	7.284791
18	StreamingMovies	9.846766
19	PaperlessBilling	2.417543
20	Tenure	2.683132
21	dmy_Male	1.936554
22	dmy_Nonbinary	1.047586
23	dmy_Married	1.893268
24	dmy_Never_Married	1.906809
25	dmy_Separated	1.938157
26	dmy_Widowed	1.943057
27	dmy_One_year	1.386418
28	dmy_Two_Year	1.449272
29	dmy_Fiber_Optic	3.540515
30	dmy_None	1.692819

In [314…

```
#That changed things a lot! Looks like I will need to remove Monthly Charge next. I
X = df_clean[['Children', 'Age', 'Income', 'Outage_sec_perweek', 'Email', 'Contacts
    'Yearly_equip_failure', 'Techie', 'Port_modem', 'Tablet', 'Phone', 'Multiple',
    'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'Streaming
    'StreamingMovies', 'PaperlessBilling', 'Tenure', 'dmy_Male',
    'dmy_Nonbinary', 'dmy_Married', 'dmy_Never_Married', 'dmy_Separated', 'dmy_Wido
    'dmy_One_year', 'dmy_Two_Year', 'dmy_Fiber_Optic', 'dmy_None']]

vif_df = pd.DataFrame()
vif_df["feature"] = X.columns

vif_df["VIF"] = [variance_inflation_factor(X.values, i)
for i in range(len(X.columns))]

print(vif_df)
```

	feature	VIF
0	Children	1.922207
1	Age	6.860939
2	Income	2.891930
3	Outage_sec_perweek	10.261262
4	Email	12.754369
5	Contacts	1.991141
6	Yearly_equip_failure	1.386540
7	Techie	1.199381
8	Port_modem	1.916731
9	Tablet	1.425603
10	Phone	9.112589
11	Multiple	1.833909
12	OnlineSecurity	1.545766
13	OnlineBackup	1.802731
14	DeviceProtection	1.774471
15	TechSupport	1.591372
16	StreamingTV	1.948203
17	StreamingMovies	1.938250
18	PaperlessBilling	2.381761
19	Tenure	2.643940
20	dmy_Male	1.924462
21	dmy_Nonbinary	1.047027
22	dmy_Married	1.860490
23	dmy_Never_Married	1.868673
24	dmy_Separated	1.900113
25	dmy_Widowed	1.904233
26	dmy_One_year	1.379691
27	dmy_Two_Year	1.443847
28	dmy_Fiber_Optic	2.224824
29	dmy_None	1.593077

In [315…

```
#Looking much better. We want the VIF values to be under 5, so I will remove "Email
X = df_clean[['Children', 'Age', 'Income', 'Outage_sec_perweek', 'Contacts',
    'Yearly_equip_failure', 'Techie', 'Port_modem', 'Tablet', 'Phone', 'Multiple',
    'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'Streaming
    'StreamingMovies', 'PaperlessBilling', 'Tenure', 'dmy_Male',
    'dmy_Nonbinary', 'dmy_Married', 'dmy_Never_Married', 'dmy_Separated', 'dmy_Wido
    'dmy_One_year', 'dmy_Two_Year', 'dmy_Fiber_Optic', 'dmy_None']]

vif_df = pd.DataFrame()
vif_df["feature"] = X.columns

vif_df["VIF"] = [variance_inflation_factor(X.values, i)
for i in range(len(X.columns))]

print(vif_df)
```

	feature	VIF
0	Children	1.911341
1	Age	6.621272
2	Income	2.863929
3	Outage_sec_perweek	9.576113
4	Contacts	1.981858
5	Yearly_equip_failure	1.385534
6	Techie	1.198761
7	Port_modem	1.905632
8	Tablet	1.424304
9	Phone	8.627060
10	Multiple	1.826307
11	OnlineSecurity	1.544496
12	OnlineBackup	1.798298
13	DeviceProtection	1.769054
14	TechSupport	1.581615
15	StreamingTV	1.937505
16	StreamingMovies	1.930520
17	PaperlessBilling	2.369677
18	Tenure	2.629033
19	dmy_Male	1.912313
20	dmy_Nonbinary	1.046749
21	dmy_Married	1.836931
22	dmy_Never_Married	1.842611
23	dmy_Separated	1.875180
24	dmy_Widowed	1.879306
25	dmy_One_year	1.378206
26	dmy_Two_Year	1.441226
27	dmy_Fiber_Optic	2.206313
28	dmy_None	1.584983

In [316…

```
#Next I am removing "Outage_sec_perweek" which has a VIF value of 9.58. This means
X = df_clean[['Children', 'Age', 'Income', 'Contacts',
    'Yearly_equip_failure', 'Techie', 'Port_modem', 'Tablet', 'Phone', 'Multiple',
    'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'Streaming
    'StreamingMovies', 'PaperlessBilling', 'Tenure', 'dmy_Male',
    'dmy_Nonbinary', 'dmy_Married', 'dmy_Never_Married', 'dmy_Separated', 'dmy_Wido
    'dmy_One_year', 'dmy_Two_Year', 'dmy_Fiber_Optic', 'dmy_None']]

vif_df = pd.DataFrame()
vif_df["feature"] = X.columns

vif_df["VIF"] = [variance_inflation_factor(X.values, i)
for i in range(len(X.columns))]

print(vif_df)
```

	feature	VIF
0	Children	1.899183
1	Age	6.366338
2	Income	2.832118
3	Contacts	1.966153
4	Yearly_equip_failure	1.382166
5	Techie	1.197593
6	Port_modem	1.894359
7	Tablet	1.421189
8	Phone	8.074461
9	Multiple	1.812307
10	OnlineSecurity	1.540264
11	OnlineBackup	1.790605
12	DeviceProtection	1.756649
13	TechSupport	1.577599
14	StreamingTV	1.923250
15	StreamingMovies	1.915732
16	PaperlessBilling	2.355677
17	Tenure	2.602616
18	dmy_Male	1.898271
19	dmy_Nonbinary	1.046164
20	dmy_Married	1.811284
21	dmy_Never_Married	1.823875
22	dmy_Separated	1.850146
23	dmy_Widowed	1.859091
24	dmy_One_year	1.375284
25	dmy_Two_Year	1.435594
26	dmy_Fiber_Optic	2.180671
27	dmy_None	1.571573

In [317…

```
#Removing "Phone" next. VIF value is at 8.07.
X = df_clean[['Children', 'Age', 'Income', 'Contacts',
    'Yearly_equip_failure', 'Techie', 'Port_modem', 'Tablet', 'Multiple',
    'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'Streaming
    'StreamingMovies', 'PaperlessBilling', 'Tenure', 'dmy_Male',
    'dmy_Nonbinary', 'dmy_Married', 'dmy_Never_Married', 'dmy_Separated', 'dmy_Wido
    'dmy_One_year', 'dmy_Two_Year', 'dmy_Fiber_Optic', 'dmy_None']]

vif_df = pd.DataFrame()
vif_df["feature"] = X.columns

vif_df["VIF"] = [variance_inflation_factor(X.values, i)
for i in range(len(X.columns))]

print(vif_df)
```

	feature	VIF
0	Children	1.882469
1	Age	5.913609
2	Income	2.778089
3	Contacts	1.947028
4	Yearly_equip_failure	1.379178
5	Techie	1.195510
6	Port_modem	1.879079
7	Tablet	1.412204
8	Multiple	1.794746
9	OnlineSecurity	1.532844
10	OnlineBackup	1.780388
11	DeviceProtection	1.753295
12	TechSupport	1.570282
13	StreamingTV	1.910975
14	StreamingMovies	1.901877
15	PaperlessBilling	2.333943
16	Tenure	2.564190
17	dmy_Male	1.879943
18	dmy_Nonbinary	1.044982
19	dmy_Married	1.768807
20	dmy_Never_Married	1.794152
21	dmy_Separated	1.812929
22	dmy_Widowed	1.822590
23	dmy_One_year	1.367808
24	dmy_Two_Year	1.429090
25	dmy_Fiber_Optic	2.150404
26	dmy_None	1.553371

In [318…

```
#Next, I will remove Age. The VIF is just over 5 at 5.91.
X = df_clean[['Children', 'Income', 'Contacts',
    'Yearly_equip_failure', 'Techie', 'Port_modem', 'Tablet', 'Multiple',
    'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'Streaming
    'StreamingMovies', 'PaperlessBilling', 'Tenure', 'dmy_Male',
    'dmy_Nonbinary', 'dmy_Married', 'dmy_Never_Married', 'dmy_Separated', 'dmy_Wido
    'dmy_One_year', 'dmy_Two_Year', 'dmy_Fiber_Optic', 'dmy_None']]

vif_df = pd.DataFrame()
vif_df["feature"] = X.columns

vif_df["VIF"] = [variance_inflation_factor(X.values, i)
for i in range(len(X.columns))]

print(vif_df)
```

	feature	VIF
0	Children	1.873675
1	Income	2.718840
2	Contacts	1.920105
3	Yearly_equip_failure	1.372859
4	Techie	1.193754
5	Port_modem	1.859767
6	Tablet	1.407526
7	Multiple	1.778151
8	OnlineSecurity	1.527497
9	OnlineBackup	1.766566
10	DeviceProtection	1.738305
11	TechSupport	1.554999
12	StreamingTV	1.892433
13	StreamingMovies	1.880658
14	PaperlessBilling	2.303495
15	Tenure	2.507585
16	dmy_Male	1.858825
17	dmy_Nonbinary	1.044847
18	dmy_Married	1.733438
19	dmy_Never_Married	1.754964
20	dmy_Separated	1.776311
21	dmy_Widowed	1.786031
22	dmy_One_year	1.363157
23	dmy_Two_Year	1.424461
24	dmy_Fiber_Optic	2.111249
25	dmy_None	1.532583

In [319…

```
#Multicollinearity appears to be gone now. Lets re-run the model. From here we will
import statsmodels.api as sm
y = df_clean['Churn']

X = df_clean[['Children', 'Income', 'Contacts',
    'Yearly_equip_failure', 'Techie', 'Port_modem', 'Tablet', 'Multiple',
    'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'Streaming
    'StreamingMovies', 'PaperlessBilling', 'Tenure', 'dmy_Male',
    'dmy_Nonbinary', 'dmy_Married', 'dmy_Never_Married', 'dmy_Separated', 'dmy_Wido
    'dmy_One_year', 'dmy_Two_Year', 'dmy_Fiber_Optic', 'dmy_None'
   ]]

X = sm.add_constant(X)

ReducedModel = sm.Logit(y, X).fit()

print(ReducedModel.params)

print(ReducedModel.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.223279
         Iterations 9
const                  -2.040853e+00
Children                1.632716e-02
Income                  8.146701e-07
Contacts                5.476795e-02
Yearly_equip_failure   -4.340989e-02
Techie                  1.091454e+00
Port_modem              1.154628e-01
Tablet                 -4.845187e-02
Multiple                1.647623e+00
OnlineSecurity         -1.418315e-01
OnlineBackup            7.965764e-01
DeviceProtection        4.460474e-01
TechSupport             2.797929e-01
StreamingTV             2.945500e+00
StreamingMovies         3.456637e+00
PaperlessBilling        1.562116e-01
Tenure                 -1.092602e-01
dmy_Male                2.711008e-01
dmy_Nonbinary          -1.346092e-01
dmy_Married             1.172775e-01
dmy_Never_Married       3.857449e-02
dmy_Separated           1.566388e-01
dmy_Widowed             2.561387e-01
dmy_One_year           -3.320117e+00
dmy_Two_Year           -3.401284e+00
dmy_Fiber_Optic        -1.374400e+00
dmy_None               -1.450038e+00
dtype: float64
                           Logit Regression Results
==============================================================================
Dep. Variable:                  Churn   No. Observations:                10000
Model:                          Logit   Df Residuals:                    9973
Method:                           MLE   Df Model:                          26
Date:                Mon, 15 Jul 2024   Pseudo R-squ.:                 0.6139
Time:                        12:34:25   Log-Likelihood:                -2232.8
converged:                       True   LL-Null:                       -5782.2
Covariance Type:            nonrobust   LLR p-value:                     0.000
==============================================================================
                           coef    std err          z      P>|z|      [0.025      0.
975]
------------------------------------------------------------------------------
----
const                   -2.0409      0.185    -11.042      0.000      -2.403      -
1.679
Children                 0.0163      0.018      0.910      0.363      -0.019
0.052
Income                8.147e-07   1.36e-06      0.599      0.549   -1.85e-06    3.48
e-06
Contacts                 0.0548      0.038      1.430      0.153      -0.020
0.130
Yearly_equip_failure    -0.0434      0.060     -0.721      0.471      -0.161
0.075
```

```
Techie                  1.0915      0.102     10.752      0.000       0.892
1.290
Port_modem              0.1155      0.076      1.513      0.130      -0.034
0.265
Tablet                 -0.0485      0.084     -0.580      0.562      -0.212
0.115
Multiple                1.6476      0.083     19.797      0.000       1.485
1.811
OnlineSecurity         -0.1418      0.080     -1.781      0.075      -0.298
0.014
OnlineBackup            0.7966      0.078     10.190      0.000       0.643
0.950
DeviceProtection        0.4460      0.077      5.790      0.000       0.295
0.597
TechSupport             0.2798      0.079      3.558      0.000       0.126
0.434
StreamingTV             2.9455      0.096     30.534      0.000       2.756
3.135
StreamingMovies         3.4566      0.102     33.864      0.000       3.257
3.657
PaperlessBilling        0.1562      0.078      2.012      0.044       0.004
0.308
Tenure                 -0.1093      0.003    -40.321      0.000      -0.115      -
0.104
dmy_Male                0.2711      0.077      3.498      0.000       0.119
0.423
dmy_Nonbinary          -0.1346      0.260     -0.517      0.605      -0.644
0.375
dmy_Married             0.1173      0.120      0.974      0.330      -0.119
0.353
dmy_Never_Married       0.0386      0.120      0.321      0.748      -0.197
0.274
dmy_Separated           0.1566      0.119      1.319      0.187      -0.076
0.389
dmy_Widowed             0.2561      0.119      2.153      0.031       0.023
0.489
dmy_One_year           -3.3201      0.124    -26.823      0.000      -3.563      -
3.078
dmy_Two_Year           -3.4013      0.121    -28.038      0.000      -3.639      -
3.164
dmy_Fiber_Optic        -1.3744      0.090    -15.190      0.000      -1.552      -
1.197
dmy_None               -1.4500      0.110    -13.141      0.000      -1.666      -
1.234
================================================================================
====
```

In [320…

```
#Beginning eliminating statistically insignificant variables for the model. One by
#Multicollinearity appears to be gone now. Lets re-run the model. From here we will
import statsmodels.api as sm
y = df_clean['Churn']

X = df_clean[['Children', 'Income', 'Contacts',
    'Yearly_equip_failure', 'Techie', 'Port_modem', 'Tablet', 'Multiple',
    'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'Streaming
    'StreamingMovies', 'PaperlessBilling', 'Tenure', 'dmy_Male',
```

```
    'dmy_Nonbinary', 'dmy_Married', 'dmy_Separated', 'dmy_Widowed',
    'dmy_One_year', 'dmy_Two_Year', 'dmy_Fiber_Optic', 'dmy_None'
   ]]

X = sm.add_constant(X)

ReducedModel = sm.Logit(y, X).fit()

print(ReducedModel.params)

print(ReducedModel.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.223284
         Iterations 9
const                  -2.022502e+00
Children                1.636791e-02
Income                  8.106214e-07
Contacts                5.476065e-02
Yearly_equip_failure   -4.336616e-02
Techie                  1.091295e+00
Port_modem              1.155098e-01
Tablet                 -4.866447e-02
Multiple                1.648419e+00
OnlineSecurity         -1.416506e-01
OnlineBackup            7.967989e-01
DeviceProtection        4.461764e-01
TechSupport             2.800090e-01
StreamingTV             2.944937e+00
StreamingMovies         3.456157e+00
PaperlessBilling        1.564954e-01
Tenure                 -1.092544e-01
dmy_Male                2.704978e-01
dmy_Nonbinary          -1.336284e-01
dmy_Married             9.886889e-02
dmy_Separated           1.382134e-01
dmy_Widowed             2.376913e-01
dmy_One_year           -3.319259e+00
dmy_Two_Year           -3.401134e+00
dmy_Fiber_Optic        -1.374193e+00
dmy_None               -1.450692e+00
dtype: float64
                           Logit Regression Results
==============================================================================
Dep. Variable:                  Churn   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9974
Method:                           MLE   Df Model:                           25
Date:                Mon, 15 Jul 2024   Pseudo R-squ.:                  0.6138
Time:                        12:34:25   Log-Likelihood:                -2232.8
converged:                       True   LL-Null:                       -5782.2
Covariance Type:            nonrobust   LLR p-value:                     0.000
==============================================================================
                           coef    std err          z      P>|z|      [0.025      0.
975]
------------------------------------------------------------------------------
const                   -2.0225      0.176    -11.512      0.000      -2.367      -
1.678
Children                 0.0164      0.018      0.912      0.362      -0.019
0.052
Income                8.106e-07   1.36e-06      0.596      0.551   -1.86e-06    3.48
e-06
Contacts                 0.0548      0.038      1.430      0.153      -0.020
0.130
Yearly_equip_failure    -0.0434      0.060     -0.720      0.472      -0.161
0.075
Techie                   1.0913      0.102     10.750      0.000       0.892
```

```
1.290
Port_modem               0.1155      0.076      1.513      0.130      -0.034
0.265
Tablet                  -0.0487      0.084     -0.583      0.560      -0.212
0.115
Multiple                 1.6484      0.083     19.815      0.000       1.485
1.811
OnlineSecurity          -0.1417      0.080     -1.779      0.075      -0.298
0.014
OnlineBackup             0.7968      0.078     10.193      0.000       0.644
0.950
DeviceProtection         0.4462      0.077      5.791      0.000       0.295
0.597
TechSupport              0.2800      0.079      3.561      0.000       0.126
0.434
StreamingTV              2.9449      0.096     30.535      0.000       2.756
3.134
StreamingMovies          3.4562      0.102     33.865      0.000       3.256
3.656
PaperlessBilling         0.1565      0.078      2.016      0.044       0.004
0.309
Tenure                  -0.1093      0.003    -40.321      0.000      -0.115      -
0.104
dmy_Male                 0.2705      0.077      3.491      0.000       0.119
0.422
dmy_Nonbinary           -0.1336      0.260     -0.514      0.607      -0.643
0.376
dmy_Married              0.0989      0.106      0.935      0.350      -0.108
0.306
dmy_Separated            0.1382      0.104      1.330      0.184      -0.065
0.342
dmy_Widowed              0.2377      0.104      2.284      0.022       0.034
0.442
dmy_One_year            -3.3193      0.124    -26.823      0.000      -3.562      -
3.077
dmy_Two_Year            -3.4011      0.121    -28.038      0.000      -3.639      -
3.163
dmy_Fiber_Optic         -1.3742      0.090    -15.188      0.000      -1.552      -
1.197
dmy_None                -1.4507      0.110    -13.150      0.000      -1.667      -
1.234
==============================================================================
====
```

In [321…

```
#Removing Income next.
#Multicollinearity appears to be gone now. Lets re-run the model. From here we will
import statsmodels.api as sm
y = df_clean['Churn']

X = df_clean[['Children', 'Contacts',
    'Yearly_equip_failure', 'Techie', 'Port_modem', 'Tablet', 'Multiple',
    'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'Streaming
    'StreamingMovies', 'PaperlessBilling', 'Tenure', 'dmy_Male',
    'dmy_Nonbinary', 'dmy_Married', 'dmy_Separated', 'dmy_Widowed',
    'dmy_One_year', 'dmy_Two_Year', 'dmy_Fiber_Optic', 'dmy_None'
   ]]
```

```
X = sm.add_constant(X)

ReducedModel = sm.Logit(y, X).fit()

print(ReducedModel.params)

print(ReducedModel.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.223302
         Iterations 9
const                  -1.988882
Children                0.016584
Contacts                0.054784
Yearly_equip_failure   -0.042906
Techie                  1.091539
Port_modem              0.114761
Tablet                 -0.049248
Multiple                1.648146
OnlineSecurity         -0.143196
OnlineBackup            0.796385
DeviceProtection        0.447401
TechSupport             0.280082
StreamingTV             2.944591
StreamingMovies         3.455915
PaperlessBilling        0.155950
Tenure                 -0.109252
dmy_Male                0.269583
dmy_Nonbinary          -0.131628
dmy_Married             0.099819
dmy_Separated           0.137759
dmy_Widowed             0.236424
dmy_One_year           -3.318334
dmy_Two_Year           -3.401511
dmy_Fiber_Optic        -1.374619
dmy_None               -1.451399
dtype: float64
                           Logit Regression Results
==============================================================================
Dep. Variable:                  Churn   No. Observations:                10000
Model:                          Logit   Df Residuals:                    9975
Method:                           MLE   Df Model:                          24
Date:                Mon, 15 Jul 2024   Pseudo R-squ.:                  0.6138
Time:                        12:34:25   Log-Likelihood:                -2233.0
converged:                       True   LL-Null:                       -5782.2
Covariance Type:            nonrobust   LLR p-value:                     0.000
==============================================================================
====
                           coef    std err          z      P>|z|      [0.025      0.
975]
------------------------------------------------------------------------------
----
const                   -1.9889      0.166    -11.961      0.000      -2.315      -
1.663
Children                 0.0166      0.018      0.924      0.355      -0.019
0.052
Contacts                 0.0548      0.038      1.430      0.153      -0.020
0.130
Yearly_equip_failure    -0.0429      0.060     -0.712      0.476      -0.161
0.075
Techie                   1.0915      0.102     10.754      0.000       0.893
1.290
Port_modem               0.1148      0.076      1.504      0.133      -0.035
0.264
```

```
Tablet                -0.0492      0.084     -0.590      0.555      -0.213
0.114
Multiple               1.6481      0.083     19.812      0.000       1.485
1.811
OnlineSecurity        -0.1432      0.080     -1.800      0.072      -0.299
0.013
OnlineBackup           0.7964      0.078     10.189      0.000       0.643
0.950
DeviceProtection       0.4474      0.077      5.809      0.000       0.296
0.598
TechSupport            0.2801      0.079      3.562      0.000       0.126
0.434
StreamingTV            2.9446      0.096     30.535      0.000       2.756
3.134
StreamingMovies        3.4559      0.102     33.867      0.000       3.256
3.656
PaperlessBilling       0.1559      0.078      2.009      0.044       0.004
0.308
Tenure                -0.1093      0.003    -40.320      0.000      -0.115      -
0.104
dmy_Male               0.2696      0.077      3.481      0.001       0.118
0.421
dmy_Nonbinary         -0.1316      0.260     -0.506      0.613      -0.641
0.378
dmy_Married            0.0998      0.106      0.944      0.345      -0.108
0.307
dmy_Separated          0.1378      0.104      1.326      0.185      -0.066
0.341
dmy_Widowed            0.2364      0.104      2.273      0.023       0.033
0.440
dmy_One_year          -3.3183      0.124    -26.822      0.000      -3.561      -
3.076
dmy_Two_Year          -3.4015      0.121    -28.044      0.000      -3.639      -
3.164
dmy_Fiber_Optic       -1.3746      0.090    -15.195      0.000      -1.552      -
1.197
dmy_None              -1.4514      0.110    -13.155      0.000      -1.668      -
1.235
==============================================================================
====
```

In [322…

```
#Next highest P value is for dmy_Nonbinary. I will remove this variable.
import statsmodels.api as sm
y = df_clean['Churn']

X = df_clean[['Children', 'Contacts',
    'Yearly_equip_failure', 'Techie', 'Port_modem', 'Tablet', 'Multiple',
    'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'Streaming
    'StreamingMovies', 'PaperlessBilling', 'Tenure', 'dmy_Male', 'dmy_Married', 'dm
    'dmy_One_year', 'dmy_Two_Year', 'dmy_Fiber_Optic', 'dmy_None'
   ]]

X = sm.add_constant(X)

ReducedModel = sm.Logit(y, X).fit()
```

```
print(ReducedModel.params)

print(ReducedModel.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.223315
         Iterations 9
const                  -1.994583
Children                0.016589
Contacts                0.054852
Yearly_equip_failure   -0.043955
Techie                  1.092445
Port_modem              0.115048
Tablet                 -0.049498
Multiple                1.648261
OnlineSecurity         -0.143075
OnlineBackup            0.795459
DeviceProtection        0.447685
TechSupport             0.280267
StreamingTV             2.944528
StreamingMovies         3.456383
PaperlessBilling        0.156025
Tenure                 -0.109267
dmy_Male                0.275422
dmy_Married             0.100704
dmy_Separated           0.138876
dmy_Widowed             0.236405
dmy_One_year           -3.318401
dmy_Two_Year           -3.401753
dmy_Fiber_Optic        -1.375078
dmy_None               -1.451565
dtype: float64
                           Logit Regression Results
==============================================================================
Dep. Variable:                  Churn   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9976
Method:                           MLE   Df Model:                           23
Date:                Mon, 15 Jul 2024   Pseudo R-squ.:                  0.6138
Time:                        12:34:25   Log-Likelihood:                -2233.1
converged:                       True   LL-Null:                       -5782.2
Covariance Type:            nonrobust   LLR p-value:                     0.000
==============================================================================
====
                           coef    std err          z      P>|z|      [0.025      0.
975]
------------------------------------------------------------------------------
----
const                   -1.9946      0.166    -12.022      0.000      -2.320      -
1.669
Children                 0.0166      0.018      0.924      0.355      -0.019
0.052
Contacts                 0.0549      0.038      1.433      0.152      -0.020
0.130
Yearly_equip_failure    -0.0440      0.060     -0.730      0.465      -0.162
0.074
Techie                   1.0924      0.101     10.766      0.000       0.894
1.291
Port_modem               0.1150      0.076      1.508      0.132      -0.034
0.265
Tablet                  -0.0495      0.084     -0.593      0.553      -0.213
```

```
0.114
Multiple                 1.6483      0.083     19.815      0.000       1.485
1.811
OnlineSecurity          -0.1431      0.080     -1.798      0.072      -0.299
0.013
OnlineBackup             0.7955      0.078     10.180      0.000       0.642
0.949
DeviceProtection         0.4477      0.077      5.814      0.000       0.297
0.599
TechSupport              0.2803      0.079      3.565      0.000       0.126
0.434
StreamingTV              2.9445      0.096     30.538      0.000       2.756
3.134
StreamingMovies          3.4564      0.102     33.871      0.000       3.256
3.656
PaperlessBilling         0.1560      0.078      2.011      0.044       0.004
0.308
Tenure                  -0.1093      0.003    -40.317      0.000      -0.115      -
0.104
dmy_Male                 0.2754      0.077      3.595      0.000       0.125
0.426
dmy_Married              0.1007      0.106      0.952      0.341      -0.107
0.308
dmy_Separated            0.1389      0.104      1.337      0.181      -0.065
0.343
dmy_Widowed              0.2364      0.104      2.273      0.023       0.033
0.440
dmy_One_year            -3.3184      0.124    -26.827      0.000      -3.561      -
3.076
dmy_Two_Year            -3.4018      0.121    -28.048      0.000      -3.639      -
3.164
dmy_Fiber_Optic         -1.3751      0.090    -15.203      0.000      -1.552      -
1.198
dmy_None                -1.4516      0.110    -13.156      0.000      -1.668      -
1.235
==========================================================================
====
```

In [323…

```
#Next highest P value is for Tablet. I will remove this variable.
import statsmodels.api as sm
y = df_clean['Churn']

X = df_clean[['Children', 'Contacts',
    'Yearly_equip_failure', 'Techie', 'Port_modem', 'Multiple',
    'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'Streaming
    'StreamingMovies', 'PaperlessBilling', 'Tenure', 'dmy_Male', 'dmy_Married', 'dm
    'dmy_One_year', 'dmy_Two_Year', 'dmy_Fiber_Optic', 'dmy_None'
   ]]

X = sm.add_constant(X)

ReducedModel = sm.Logit(y, X).fit()

print(ReducedModel.params)

print(ReducedModel.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.223332
         Iterations 9
const                  -2.011281
Children                0.016677
Contacts                0.055549
Yearly_equip_failure   -0.043449
Techie                  1.090712
Port_modem              0.115583
Multiple                1.650042
OnlineSecurity         -0.143510
OnlineBackup            0.795870
DeviceProtection        0.447761
TechSupport             0.281171
StreamingTV             2.945675
StreamingMovies         3.455478
PaperlessBilling        0.154925
Tenure                 -0.109271
dmy_Male                0.276714
dmy_Married             0.100715
dmy_Separated           0.137756
dmy_Widowed             0.236726
dmy_One_year           -3.318294
dmy_Two_Year           -3.401908
dmy_Fiber_Optic        -1.375350
dmy_None               -1.452482
dtype: float64
                           Logit Regression Results
==============================================================================
Dep. Variable:                  Churn   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9977
Method:                           MLE   Df Model:                           22
Date:                Mon, 15 Jul 2024   Pseudo R-squ.:                  0.6138
Time:                        12:34:25   Log-Likelihood:                -2233.3
converged:                       True   LL-Null:                       -5782.2
Covariance Type:            nonrobust   LLR p-value:                     0.000
==============================================================================
====
                           coef    std err          z      P>|z|      [0.025      0.
975]
------------------------------------------------------------------------------
----
const                   -2.0113      0.164    -12.299      0.000      -2.332      -
1.691
Children                 0.0167      0.018      0.930      0.353      -0.018
0.052
Contacts                 0.0555      0.038      1.451      0.147      -0.019
0.131
Yearly_equip_failure    -0.0434      0.060     -0.722      0.470      -0.161
0.075
Techie                   1.0907      0.101     10.757      0.000       0.892
1.289
Port_modem               0.1156      0.076      1.515      0.130      -0.034
0.265
Multiple                 1.6500      0.083     19.846      0.000       1.487
1.813
```

```
OnlineSecurity          -0.1435      0.080     -1.804      0.071      -0.299
0.012
OnlineBackup             0.7959      0.078     10.186      0.000       0.643
0.949
DeviceProtection         0.4478      0.077      5.815      0.000       0.297
0.599
TechSupport              0.2812      0.079      3.577      0.000       0.127
0.435
StreamingTV              2.9457      0.096     30.548      0.000       2.757
3.135
StreamingMovies          3.4555      0.102     33.870      0.000       3.256
3.655
PaperlessBilling         0.1549      0.078      1.997      0.046       0.003
0.307
Tenure                  -0.1093      0.003    -40.319      0.000      -0.115      -
0.104
dmy_Male                 0.2767      0.077      3.614      0.000       0.127
0.427
dmy_Married              0.1007      0.106      0.953      0.341      -0.107
0.308
dmy_Separated            0.1378      0.104      1.326      0.185      -0.066
0.341
dmy_Widowed              0.2367      0.104      2.276      0.023       0.033
0.441
dmy_One_year            -3.3183      0.124    -26.833      0.000      -3.561      -
3.076
dmy_Two_Year            -3.4019      0.121    -28.051      0.000      -3.640      -
3.164
dmy_Fiber_Optic         -1.3754      0.090    -15.205      0.000      -1.553      -
1.198
dmy_None                -1.4525      0.110    -13.167      0.000      -1.669      -
1.236
==============================================================================
====
```

In [324…

```python
#Next highest P value is for Yearly_Equip_Failure. I will remove this variable.
import statsmodels.api as sm
y = df_clean['Churn']

X = df_clean[['Children', 'Contacts',
    'Techie', 'Port_modem', 'Multiple',
    'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'Streaming
    'StreamingMovies', 'PaperlessBilling', 'Tenure', 'dmy_Male', 'dmy_Married', 'dm
    'dmy_One_year', 'dmy_Two_Year', 'dmy_Fiber_Optic', 'dmy_None'
   ]]

X = sm.add_constant(X)

ReducedModel = sm.Logit(y, X).fit()

print(ReducedModel.params)

print(ReducedModel.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.223359
         Iterations 9
const              -2.027478
Children            0.016591
Contacts            0.055607
Techie              1.090831
Port_modem          0.115378
Multiple            1.649627
OnlineSecurity     -0.143005
OnlineBackup        0.796514
DeviceProtection    0.447719
TechSupport         0.279729
StreamingTV         2.944537
StreamingMovies     3.455223
PaperlessBilling    0.154337
Tenure             -0.109279
dmy_Male            0.276627
dmy_Married         0.102269
dmy_Separated       0.140092
dmy_Widowed         0.236153
dmy_One_year       -3.318037
dmy_Two_Year       -3.401526
dmy_Fiber_Optic    -1.375881
dmy_None           -1.452771
dtype: float64
                           Logit Regression Results
==============================================================================
Dep. Variable:                  Churn   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9978
Method:                           MLE   Df Model:                           21
Date:                Mon, 15 Jul 2024   Pseudo R-squ.:                  0.6137
Time:                        12:34:26   Log-Likelihood:                -2233.6
converged:                       True   LL-Null:                       -5782.2
Covariance Type:            nonrobust   LLR p-value:                     0.000
==============================================================================
                     coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const              -2.0275      0.162    -12.515      0.000      -2.345      -1.710
Children            0.0166      0.018      0.925      0.355      -0.019       0.052
Contacts            0.0556      0.038      1.453      0.146      -0.019       0.131
Techie              1.0908      0.101     10.758      0.000       0.892       1.290
Port_modem          0.1154      0.076      1.512      0.130      -0.034       0.265
Multiple            1.6496      0.083     19.844      0.000       1.487       1.813
OnlineSecurity     -0.1430      0.080     -1.798      0.072      -0.299       0.013
OnlineBackup        0.7965      0.078     10.196      0.000       0.643       0.950
DeviceProtection    0.4477      0.077      5.815      0.000       0.297       0.599
TechSupport         0.2797      0.079      3.560      0.000       0.126       0.434
StreamingTV         2.9445      0.096     30.546      0.000       2.756       3.133
StreamingMovies     3.4552      0.102     33.874      0.000       3.255       3.655
PaperlessBilling    0.1543      0.078      1.990      0.047       0.002       0.306
Tenure             -0.1093      0.003    -40.325      0.000      -0.115      -0.104
dmy_Male            0.2766      0.077      3.613      0.000       0.127       0.427
dmy_Married         0.1023      0.106      0.968      0.333      -0.105       0.309
dmy_Separated       0.1401      0.104      1.349      0.177      -0.063       0.344
dmy_Widowed         0.2362      0.104      2.270      0.023       0.032       0.440
```

```
dmy_One_year        -3.3180      0.124    -26.835      0.000      -3.560      -3.076
dmy_Two_Year        -3.4015      0.121    -28.053      0.000      -3.639      -3.164
dmy_Fiber_Optic     -1.3759      0.090    -15.212      0.000      -1.553      -1.199
dmy_None            -1.4528      0.110    -13.171      0.000      -1.669      -1.237
==============================================================================
```

In [325…

```python
#Next highest P value is for dmy_Married. I will remove this variable.
import statsmodels.api as sm
y = df_clean['Churn']

X = df_clean[['Children', 'Contacts',
    'Techie', 'Port_modem', 'Multiple',
    'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'Streaming
    'StreamingMovies', 'PaperlessBilling', 'Tenure', 'dmy_Male', 'dmy_Separated', '
    'dmy_One_year', 'dmy_Two_Year', 'dmy_Fiber_Optic', 'dmy_None'
   ]]

X = sm.add_constant(X)

ReducedModel = sm.Logit(y, X).fit()

print(ReducedModel.params)

print(ReducedModel.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.223405
         Iterations 9
const              -1.995163
Children            0.016621
Contacts            0.055114
Techie              1.090880
Port_modem          0.116112
Multiple            1.647549
OnlineSecurity     -0.145410
OnlineBackup        0.797701
DeviceProtection    0.448100
TechSupport         0.281145
StreamingTV         2.943465
StreamingMovies     3.453877
PaperlessBilling    0.153656
Tenure             -0.109220
dmy_Male            0.278718
dmy_Separated       0.107290
dmy_Widowed         0.203315
dmy_One_year       -3.317317
dmy_Two_Year       -3.399971
dmy_Fiber_Optic    -1.375394
dmy_None           -1.451364
dtype: float64
```

Logit Regression Results

Dep. Variable:	Churn	No. Observations:	10000
Model:	Logit	Df Residuals:	9979
Method:	MLE	Df Model:	20
Date:	Mon, 15 Jul 2024	Pseudo R-squ.:	0.6136
Time:	12:34:26	Log-Likelihood:	-2234.1
converged:	True	LL-Null:	-5782.2
Covariance Type:	nonrobust	LLR p-value:	0.000

	coef	std err	z	P>|z|	[0.025	0.975]
const	-1.9952	0.158	-12.593	0.000	-2.306	-1.685
Children	0.0166	0.018	0.926	0.354	-0.019	0.052
Contacts	0.0551	0.038	1.440	0.150	-0.020	0.130
Techie	1.0909	0.101	10.761	0.000	0.892	1.290
Port_modem	0.1161	0.076	1.522	0.128	-0.033	0.266
Multiple	1.6475	0.083	19.832	0.000	1.485	1.810
OnlineSecurity	-0.1454	0.080	-1.829	0.067	-0.301	0.010
OnlineBackup	0.7977	0.078	10.212	0.000	0.645	0.951
DeviceProtection	0.4481	0.077	5.820	0.000	0.297	0.599
TechSupport	0.2811	0.079	3.579	0.000	0.127	0.435
StreamingTV	2.9435	0.096	30.544	0.000	2.755	3.132
StreamingMovies	3.4539	0.102	33.873	0.000	3.254	3.654
PaperlessBilling	0.1537	0.078	1.981	0.048	0.002	0.306
Tenure	-0.1092	0.003	-40.336	0.000	-0.115	-0.104
dmy_Male	0.2787	0.077	3.641	0.000	0.129	0.429
dmy_Separated	0.1073	0.098	1.094	0.274	-0.085	0.300
dmy_Widowed	0.2033	0.098	2.069	0.039	0.011	0.396
dmy_One_year	-3.3173	0.124	-26.837	0.000	-3.560	-3.075
dmy_Two_Year	-3.4000	0.121	-28.053	0.000	-3.638	-3.162

```
dmy_Fiber_Optic     -1.3754      0.090    -15.207      0.000      -1.553      -1.198
dmy_None            -1.4514      0.110    -13.161      0.000      -1.668      -1.235
==============================================================================
```

In [326…

```python
#Next highest P value is for Children. I will remove this variable.
import statsmodels.api as sm
y = df_clean['Churn']

X = df_clean[['Contacts',
    'Techie', 'Port_modem', 'Multiple',
    'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'Streaming
    'StreamingMovies', 'PaperlessBilling', 'Tenure', 'dmy_Male', 'dmy_Separated', '
    'dmy_One_year', 'dmy_Two_Year', 'dmy_Fiber_Optic', 'dmy_None'
   ]]

X = sm.add_constant(X)

ReducedModel = sm.Logit(y, X).fit()

print(ReducedModel.params)

print(ReducedModel.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.223448
         Iterations 9
const              -1.957427
Contacts            0.053413
Techie              1.087899
Port_modem          0.116791
Multiple            1.643990
OnlineSecurity     -0.144713
OnlineBackup        0.796912
DeviceProtection    0.447902
TechSupport         0.279383
StreamingTV         2.944010
StreamingMovies     3.453294
PaperlessBilling    0.153679
Tenure             -0.109208
dmy_Male            0.278408
dmy_Separated       0.106788
dmy_Widowed         0.202799
dmy_One_year       -3.316080
dmy_Two_Year       -3.395317
dmy_Fiber_Optic    -1.374861
dmy_None           -1.449739
dtype: float64
```

Logit Regression Results

Dep. Variable:	Churn	No. Observations:	10000
Model:	Logit	Df Residuals:	9980
Method:	MLE	Df Model:	19
Date:	Mon, 15 Jul 2024	Pseudo R-squ.:	0.6136
Time:	12:34:26	Log-Likelihood:	-2234.5
converged:	True	LL-Null:	-5782.2
Covariance Type:	nonrobust	LLR p-value:	0.000

	coef	std err	z	P>|z|	[0.025	0.975]
const	-1.9574	0.153	-12.796	0.000	-2.257	-1.658
Contacts	0.0534	0.038	1.397	0.162	-0.022	0.128
Techie	1.0879	0.101	10.735	0.000	0.889	1.287
Port_modem	0.1168	0.076	1.531	0.126	-0.033	0.266
Multiple	1.6440	0.083	19.820	0.000	1.481	1.807
OnlineSecurity	-0.1447	0.080	-1.820	0.069	-0.301	0.011
OnlineBackup	0.7969	0.078	10.204	0.000	0.644	0.950
DeviceProtection	0.4479	0.077	5.818	0.000	0.297	0.599
TechSupport	0.2794	0.079	3.558	0.000	0.125	0.433
StreamingTV	2.9440	0.096	30.550	0.000	2.755	3.133
StreamingMovies	3.4533	0.102	33.871	0.000	3.253	3.653
PaperlessBilling	0.1537	0.078	1.982	0.048	0.002	0.306
Tenure	-0.1092	0.003	-40.336	0.000	-0.115	-0.104
dmy_Male	0.2784	0.077	3.638	0.000	0.128	0.428
dmy_Separated	0.1068	0.098	1.089	0.276	-0.085	0.299
dmy_Widowed	0.2028	0.098	2.064	0.039	0.010	0.395
dmy_One_year	-3.3161	0.124	-26.828	0.000	-3.558	-3.074
dmy_Two_Year	-3.3953	0.121	-28.061	0.000	-3.632	-3.158
dmy_Fiber_Optic	-1.3749	0.090	-15.203	0.000	-1.552	-1.198

```
dmy_None            -1.4497      0.110    -13.148      0.000      -1.666      -1.234
==============================================================================================
```

In [327…

```
#Next highest P value is for dmy_Separated. I will remove this variable.
import statsmodels.api as sm
y = df_clean['Churn']

X = df_clean[['Contacts',
    'Techie', 'Port_modem', 'Multiple',
    'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'Streaming
    'StreamingMovies', 'PaperlessBilling', 'Tenure', 'dmy_Male', 'dmy_Widowed',
    'dmy_One_year', 'dmy_Two_Year', 'dmy_Fiber_Optic', 'dmy_None'
   ]]

X = sm.add_constant(X)

ReducedModel = sm.Logit(y, X).fit()

print(ReducedModel.params)

print(ReducedModel.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.223507
         Iterations 9
const              -1.931857
Contacts            0.053243
Techie              1.088275
Port_modem          0.116258
Multiple            1.644984
OnlineSecurity     -0.144284
OnlineBackup        0.795288
DeviceProtection    0.448175
TechSupport         0.278450
StreamingTV         2.945876
StreamingMovies     3.455323
PaperlessBilling    0.153507
Tenure             -0.109244
dmy_Male            0.277773
dmy_Widowed         0.175727
dmy_One_year       -3.316442
dmy_Two_Year       -3.395594
dmy_Fiber_Optic    -1.373355
dmy_None           -1.447508
dtype: float64
                           Logit Regression Results
==============================================================================
Dep. Variable:                  Churn   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9981
Method:                           MLE   Df Model:                           18
Date:                Mon, 15 Jul 2024   Pseudo R-squ.:                  0.6135
Time:                        12:34:26   Log-Likelihood:                -2235.1
converged:                       True   LL-Null:                       -5782.2
Covariance Type:            nonrobust   LLR p-value:                     0.000
==============================================================================
                     coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const              -1.9319      0.151    -12.788      0.000      -2.228      -1.636
Contacts            0.0532      0.038      1.394      0.163      -0.022       0.128
Techie              1.0883      0.101     10.744      0.000       0.890       1.287
Port_modem          0.1163      0.076      1.525      0.127      -0.033       0.266
Multiple            1.6450      0.083     19.836      0.000       1.482       1.808
OnlineSecurity     -0.1443      0.079     -1.815      0.069      -0.300       0.011
OnlineBackup        0.7953      0.078     10.188      0.000       0.642       0.948
DeviceProtection    0.4482      0.077      5.823      0.000       0.297       0.599
TechSupport         0.2784      0.078      3.547      0.000       0.125       0.432
StreamingTV         2.9459      0.096     30.568      0.000       2.757       3.135
StreamingMovies     3.4553      0.102     33.890      0.000       3.255       3.655
PaperlessBilling    0.1535      0.078      1.980      0.048       0.002       0.305
Tenure             -0.1092      0.003    -40.342      0.000      -0.115      -0.104
dmy_Male            0.2778      0.077      3.630      0.000       0.128       0.428
dmy_Widowed         0.1757      0.095      1.849      0.064      -0.011       0.362
dmy_One_year       -3.3164      0.124    -26.826      0.000      -3.559      -3.074
dmy_Two_Year       -3.3956      0.121    -28.076      0.000      -3.633      -3.159
dmy_Fiber_Optic    -1.3734      0.090    -15.193      0.000      -1.551      -1.196
dmy_None           -1.4475      0.110    -13.137      0.000      -1.663      -1.232
==============================================================================
```

In [328…

```
#Next highest P value is for Contacts. I will remove this variable.
import statsmodels.api as sm
y = df_clean['Churn']

X = df_clean[[
    'Techie', 'Port_modem', 'Multiple',
    'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'Streaming
    'StreamingMovies', 'PaperlessBilling', 'Tenure', 'dmy_Male', 'dmy_Widowed',
    'dmy_One_year', 'dmy_Two_Year', 'dmy_Fiber_Optic', 'dmy_None'
   ]]

X = sm.add_constant(X)

ReducedModel = sm.Logit(y, X).fit()

print(ReducedModel.params)

print(ReducedModel.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.223604
         Iterations 9
const              -1.876819
Techie              1.089493
Port_modem          0.114287
Multiple            1.641024
OnlineSecurity     -0.143750
OnlineBackup        0.794754
DeviceProtection    0.449318
TechSupport         0.276876
StreamingTV         2.943944
StreamingMovies     3.455059
PaperlessBilling    0.153289
Tenure             -0.109181
dmy_Male            0.279595
dmy_Widowed         0.174485
dmy_One_year       -3.312423
dmy_Two_Year       -3.393907
dmy_Fiber_Optic    -1.372410
dmy_None           -1.448400
dtype: float64
                           Logit Regression Results
==============================================================================
Dep. Variable:                  Churn   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9982
Method:                           MLE   Df Model:                           17
Date:                Mon, 15 Jul 2024   Pseudo R-squ.:                  0.6133
Time:                        12:34:26   Log-Likelihood:                -2236.0
converged:                       True   LL-Null:                       -5782.2
Covariance Type:            nonrobust   LLR p-value:                     0.000
==============================================================================
                     coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const              -1.8768      0.146    -12.889      0.000      -2.162      -1.591
Techie              1.0895      0.101     10.754      0.000       0.891       1.288
Port_modem          0.1143      0.076      1.499      0.134      -0.035       0.264
Multiple            1.6410      0.083     19.814      0.000       1.479       1.803
OnlineSecurity     -0.1437      0.079     -1.809      0.070      -0.299       0.012
OnlineBackup        0.7948      0.078     10.184      0.000       0.642       0.948
DeviceProtection    0.4493      0.077      5.839      0.000       0.299       0.600
TechSupport         0.2769      0.078      3.528      0.000       0.123       0.431
StreamingTV         2.9439      0.096     30.558      0.000       2.755       3.133
StreamingMovies     3.4551      0.102     33.901      0.000       3.255       3.655
PaperlessBilling    0.1533      0.078      1.977      0.048       0.001       0.305
Tenure             -0.1092      0.003    -40.351      0.000      -0.114      -0.104
dmy_Male            0.2796      0.076      3.655      0.000       0.130       0.430
dmy_Widowed         0.1745      0.095      1.836      0.066      -0.012       0.361
dmy_One_year       -3.3124      0.124    -26.821      0.000      -3.554      -3.070
dmy_Two_Year       -3.3939      0.121    -28.075      0.000      -3.631      -3.157
dmy_Fiber_Optic    -1.3724      0.090    -15.187      0.000      -1.550      -1.195
dmy_None           -1.4484      0.110    -13.152      0.000      -1.664      -1.233
==============================================================================
```

In [329…

```
#Next highest P value is for Port_modem. I will remove this variable.
import statsmodels.api as sm
```

```
y = df_clean['Churn']

X = df_clean[[
    'Techie', 'Multiple',
    'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'Streaming
    'StreamingMovies', 'PaperlessBilling', 'Tenure', 'dmy_Male', 'dmy_Widowed',
    'dmy_One_year', 'dmy_Two_Year', 'dmy_Fiber_Optic', 'dmy_None'
   ]]

X = sm.add_constant(X)

ReducedModel = sm.Logit(y, X).fit()

print(ReducedModel.params)

print(ReducedModel.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.223717
         Iterations 9
const              -1.821014
Techie              1.086836
Multiple            1.638284
OnlineSecurity     -0.143586
OnlineBackup        0.794134
DeviceProtection    0.446425
TechSupport         0.277140
StreamingTV         2.943781
StreamingMovies     3.456480
PaperlessBilling    0.152822
Tenure             -0.109109
dmy_Male            0.280589
dmy_Widowed         0.171899
dmy_One_year       -3.312679
dmy_Two_Year       -3.390414
dmy_Fiber_Optic    -1.370699
dmy_None           -1.448468
dtype: float64
```

Logit Regression Results

Dep. Variable:	Churn	No. Observations:	10000
Model:	Logit	Df Residuals:	9983
Method:	MLE	Df Model:	16
Date:	Mon, 15 Jul 2024	Pseudo R-squ.:	0.6131
Time:	12:34:26	Log-Likelihood:	-2237.2
converged:	True	LL-Null:	-5782.2
Covariance Type:	nonrobust	LLR p-value:	0.000

	coef	std err	z	P>|z|	[0.025	0.975]
const	-1.8210	0.141	-12.961	0.000	-2.096	-1.546
Techie	1.0868	0.101	10.728	0.000	0.888	1.285
Multiple	1.6383	0.083	19.796	0.000	1.476	1.800
OnlineSecurity	-0.1436	0.079	-1.808	0.071	-0.299	0.012
OnlineBackup	0.7941	0.078	10.178	0.000	0.641	0.947
DeviceProtection	0.4464	0.077	5.806	0.000	0.296	0.597
TechSupport	0.2771	0.078	3.533	0.000	0.123	0.431
StreamingTV	2.9438	0.096	30.563	0.000	2.755	3.133
StreamingMovies	3.4565	0.102	33.916	0.000	3.257	3.656
PaperlessBilling	0.1528	0.078	1.972	0.049	0.001	0.305
Tenure	-0.1091	0.003	-40.360	0.000	-0.114	-0.104
dmy_Male	0.2806	0.076	3.669	0.000	0.131	0.430
dmy_Widowed	0.1719	0.095	1.811	0.070	-0.014	0.358
dmy_One_year	-3.3127	0.123	-26.840	0.000	-3.555	-3.071
dmy_Two_Year	-3.3904	0.121	-28.074	0.000	-3.627	-3.154
dmy_Fiber_Optic	-1.3707	0.090	-15.176	0.000	-1.548	-1.194
dmy_None	-1.4485	0.110	-13.156	0.000	-1.664	-1.233

In [330…

```
#Next highest P value is for OnlineSecurity. I will remove this variable.
import statsmodels.api as sm
y = df_clean['Churn']
```

```
X = df_clean[[
    'Techie', 'Multiple','DeviceProtection', 'TechSupport', 'StreamingTV',
    'StreamingMovies', 'PaperlessBilling', 'Tenure', 'dmy_Male', 'dmy_Widowed',
    'dmy_One_year', 'dmy_Two_Year', 'dmy_Fiber_Optic', 'dmy_None'
   ]]

X = sm.add_constant(X)

ReducedModel = sm.Logit(y, X).fit()

print(ReducedModel.params)

print(ReducedModel.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.229226
         Iterations 8
const              -1.480068
Techie              1.069971
Multiple            1.593078
DeviceProtection    0.445032
TechSupport         0.264034
StreamingTV         2.868940
StreamingMovies     3.371040
PaperlessBilling    0.148798
Tenure             -0.106107
dmy_Male            0.249801
dmy_Widowed         0.176453
dmy_One_year       -3.243951
dmy_Two_Year       -3.305637
dmy_Fiber_Optic    -1.321062
dmy_None           -1.401881
dtype: float64
                           Logit Regression Results
==============================================================================
Dep. Variable:                  Churn   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9985
Method:                           MLE   Df Model:                           14
Date:                Mon, 15 Jul 2024   Pseudo R-squ.:                  0.6036
Time:                        12:34:27   Log-Likelihood:                -2292.3
converged:                       True   LL-Null:                       -5782.2
Covariance Type:            nonrobust   LLR p-value:                     0.000
==============================================================================
                     coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const              -1.4801      0.130    -11.377      0.000      -1.735      -1.225
Techie              1.0700      0.100     10.703      0.000       0.874       1.266
Multiple            1.5931      0.081     19.564      0.000       1.433       1.753
DeviceProtection    0.4450      0.076      5.856      0.000       0.296       0.594
TechSupport         0.2640      0.078      3.403      0.001       0.112       0.416
StreamingTV         2.8689      0.094     30.479      0.000       2.684       3.053
StreamingMovies     3.3710      0.099     33.919      0.000       3.176       3.566
PaperlessBilling    0.1488      0.077      1.945      0.052      -0.001       0.299
Tenure             -0.1061      0.003    -40.588      0.000      -0.111      -0.101
dmy_Male            0.2498      0.075      3.313      0.001       0.102       0.398
dmy_Widowed         0.1765      0.094      1.883      0.060      -0.007       0.360
dmy_One_year       -3.2440      0.121    -26.706      0.000      -3.482      -3.006
dmy_Two_Year       -3.3056      0.118    -27.998      0.000      -3.537      -3.074
dmy_Fiber_Optic    -1.3211      0.089    -14.861      0.000      -1.495      -1.147
dmy_None           -1.4019      0.108    -12.958      0.000      -1.614      -1.190
==============================================================================
```

In [331…

```
#Next highest P value is for dmy_Widowed. I will remove this variable.
import statsmodels.api as sm
y = df_clean['Churn']

X = df_clean[[
    'Techie', 'Multiple', 'DeviceProtection', 'TechSupport', 'StreamingTV',
    'StreamingMovies', 'PaperlessBilling', 'Tenure', 'dmy_Male',
    'dmy_One_year', 'dmy_Two_Year', 'dmy_Fiber_Optic', 'dmy_None'
```

```
   ]]

X = sm.add_constant(X)

FinalReducedModel = sm.Logit(y, X).fit()

print(FinalReducedModel.params)

print(FinalReducedModel.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.229403
         Iterations 8
const              -1.445044
Techie              1.072172
Multiple            1.592332
DeviceProtection    0.443596
TechSupport         0.261103
StreamingTV         2.867359
StreamingMovies     3.368873
PaperlessBilling    0.153330
Tenure             -0.106123
dmy_Male            0.246450
dmy_One_year       -3.240318
dmy_Two_Year       -3.296896
dmy_Fiber_Optic    -1.321183
dmy_None           -1.393460
dtype: float64
                           Logit Regression Results
==============================================================================
Dep. Variable:                  Churn   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9986
Method:                           MLE   Df Model:                           13
Date:                Mon, 15 Jul 2024   Pseudo R-squ.:                  0.6033
Time:                        12:34:27   Log-Likelihood:                -2294.0
converged:                       True   LL-Null:                       -5782.2
Covariance Type:            nonrobust   LLR p-value:                     0.000
=====================================================================================
                        coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------------
const                -1.4450      0.129    -11.241      0.000      -1.697      -1.193
Techie                1.0722      0.100     10.732      0.000       0.876       1.268
Multiple              1.5923      0.081     19.564      0.000       1.433       1.752
DeviceProtection      0.4436      0.076      5.841      0.000       0.295       0.592
TechSupport           0.2611      0.078      3.368      0.001       0.109       0.413
StreamingTV           2.8674      0.094     30.484      0.000       2.683       3.052
StreamingMovies       3.3689      0.099     33.914      0.000       3.174       3.564
PaperlessBilling      0.1533      0.076      2.006      0.045       0.004       0.303
Tenure               -0.1061      0.003    -40.593      0.000      -0.111      -0.101
dmy_Male              0.2465      0.075      3.272      0.001       0.099       0.394
dmy_One_year         -3.2403      0.121    -26.697      0.000      -3.478      -3.002
dmy_Two_Year         -3.2969      0.118    -27.990      0.000      -3.528      -3.066
dmy_Fiber_Optic      -1.3212      0.089    -14.866      0.000      -1.495      -1.147
dmy_None             -1.3935      0.108    -12.906      0.000      -1.605      -1.182
=====================================================================================
```

In [ ]:

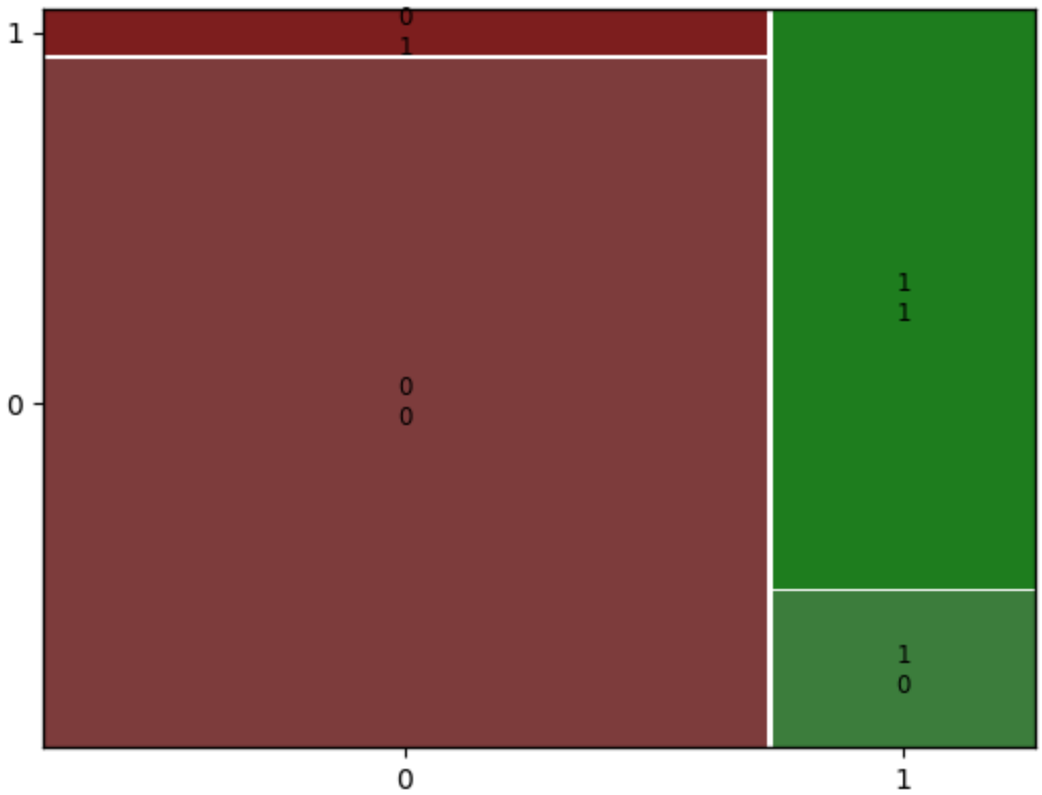
In [332…
```
#Getting Confusion Matrix
actual_response = df_clean["Churn"]

predicted_response = np.round(FinalReducedModel.predict())

outcomes = pd.DataFrame({"actual_response": actual_response,
                         "predicted_response": predicted_response})
print(outcomes.value_counts(sort = False))

#creating mosaic plot of confusion matrix
from statsmodels.graphics.mosaicplot import mosaic
conf_matrix = FinalReducedModel.pred_table()
print(conf_matrix)
mosaic(conf_matrix)
plt.show()
```

```
actual_response  predicted_response
0                0.0                 6878
                 1.0                  472
1                0.0                  563
                 1.0                 2087
Name: count, dtype: int64
[[6878.  472.]
 [ 563. 2087.]]
```



In [ ]:

In [333…
```
# Extracting TN, TP, FN and FP from conf_matrix
TN = conf_matrix[0,0]
```

```
TP = conf_matrix[1,1]
FN = conf_matrix[1,0]
FP = conf_matrix[0,1]

accuracy = (TN + TP) / (TN + FN + FP + TP)
print("accuracy: ", accuracy)

sensitivity = TP / (TP + FN)
print("sensitivity: ", sensitivity)

specificity = TN / (TN + FP)
print("specificity: ", specificity)
```

```
accuracy:  0.8965
sensitivity:  0.7875471698113208
specificity:  0.9357823129251701
```

In [335…

```
File_Path = r'C:\Users\Cali\Documents\Cleaned_Data_CSV_D208_Task2_Revised.csv'
df_clean.to_csv(File_Path, index = False)
```

In [ ]:

In [ ]: