task-1

August 3, 2023

1 Task 1

1. Churn Prediction in Telecom Industry using Logistic Regression

[1]:
```
import warnings
warnings.filterwarnings('ignore')
```

[2]:
```
import numpy as np
```

[3]:
```
import pandas as pd
```

[4]:
```
import matplotlib.pyplot as plt
```

[5]:
```
import seaborn as sns
```

[6]:
```
churn_data = pd.read_csv('churn_data.csv')
```

[7]:
```
churn_data.head()
```

[7]:
```
   customerID  tenure PhoneService        Contract PaperlessBilling  \
0  7590-VHVEG       1           No  Month-to-month              Yes
1  5575-GNVDE      34          Yes        One year               No
2  3668-QPYBK       2          Yes  Month-to-month              Yes
3  7795-CFOCW      45           No        One year               No
4  9237-HQITU       2          Yes  Month-to-month              Yes

            PaymentMethod  MonthlyCharges TotalCharges Churn
0          Electronic check           29.85        29.85    No
1              Mailed check           56.95       1889.5    No
2              Mailed check           53.85       108.15   Yes
3 Bank transfer (automatic)           42.30      1840.75    No
4          Electronic check           70.70       151.65   Yes
```

[8]:
```
customer_data = pd.read_csv('customer_data.csv')
customer_data
```

[8]:
```
      customerID  gender  SeniorCitizen Partner Dependents
0     7590-VHVEG  Female              0     Yes         No
```

```
1     5575-GNVDE    Male              0      No         No
2     3668-QPYBK    Male              0      No         No
3     7795-CFOCW    Male              0      No         No
4     9237-HQITU  Female              0      No         No
…          …     …            …     …        …
7038  6840-RESVB    Male              0     Yes        Yes
7039  2234-XADUH  Female              0     Yes        Yes
7040  4801-JZAZL  Female              0     Yes        Yes
7041  8361-LTMKD    Male              1     Yes         No
7042  3186-AJIEK    Male              0      No         No

[7043 rows x 5 columns]
```

[9]:

```
internet_data = pd.read_csv('internet_data.csv')
internet_data
```

[9]:

```
      customerID     MultipleLines InternetService OnlineSecurity  \
0     7590-VHVEG  No phone service             DSL             No
1     5575-GNVDE                No             DSL            Yes
2     3668-QPYBK                No             DSL            Yes
3     7795-CFOCW  No phone service             DSL            Yes
4     9237-HQITU                No     Fiber optic             No
…          …               …             …            …
7038  6840-RESVB               Yes             DSL            Yes
7039  2234-XADUH               Yes     Fiber optic             No
7040  4801-JZAZL  No phone service             DSL            Yes
7041  8361-LTMKD               Yes     Fiber optic             No
7042  3186-AJIEK                No     Fiber optic            Yes

     OnlineBackup DeviceProtection TechSupport StreamingTV StreamingMovies
0             Yes               No          No          No              No
1              No              Yes          No          No              No
2             Yes               No          No          No              No
3              No              Yes         Yes          No              No
4              No               No          No          No              No
…           …              …         …         …             …
7038           No              Yes         Yes         Yes             Yes
7039          Yes              Yes          No         Yes             Yes
7040           No               No          No          No              No
7041           No               No          No          No              No
7042           No              Yes         Yes         Yes             Yes

[7043 rows x 9 columns]
```

2 Merging tables using customer id

[10]:
```
df = pd.merge(churn_data, customer_data, how='inner', on = 'customerID')
```

[11]:
```
telecom_df = pd.merge(df, internet_data, how='inner', on='customerID')
```

[12]:
```
telecom_df
```

[12]:
```
      customerID  tenure PhoneService        Contract PaperlessBilling  \
0     7590-VHVEG       1           No  Month-to-month              Yes
1     5575-GNVDE      34          Yes        One year               No
2     3668-QPYBK       2          Yes  Month-to-month              Yes
3     7795-CFOCW      45           No        One year               No
4     9237-HQITU       2          Yes  Month-to-month              Yes
…          …     …          …             …               …
7038  6840-RESVB      24          Yes        One year              Yes
7039  2234-XADUH      72          Yes        One year              Yes
7040  4801-JZAZL      11           No  Month-to-month              Yes
7041  8361-LTMKD       4          Yes  Month-to-month              Yes
7042  3186-AJIEK      66          Yes        Two year              Yes

                PaymentMethod  MonthlyCharges TotalCharges Churn  gender  \
0            Electronic check           29.85        29.85    No  Female
1                Mailed check           56.95       1889.5    No    Male
2                Mailed check           53.85       108.15   Yes    Male
3     Bank transfer (automatic)           42.30      1840.75    No    Male
4            Electronic check           70.70       151.65   Yes  Female
…                         …             …          …   …     …
7038             Mailed check           84.80       1990.5    No    Male
7039  Credit card (automatic)          103.20       7362.9    No  Female
7040         Electronic check           29.60       346.45    No  Female
7041             Mailed check           74.40        306.6   Yes    Male
7042  Bank transfer (automatic)          105.65       6844.5    No    Male

      …  Partner Dependents     MultipleLines InternetService  \
0     …      Yes         No  No phone service             DSL
1     …       No         No                No             DSL
2     …       No         No                No             DSL
3     …       No         No  No phone service             DSL
4     …       No         No                No     Fiber optic
…   …      …        …               …             …
7038  …      Yes        Yes               Yes             DSL
7039  …      Yes        Yes               Yes     Fiber optic
7040  …      Yes        Yes  No phone service             DSL
7041  …      Yes         No               Yes     Fiber optic
7042  …       No         No                No     Fiber optic
```

```
     OnlineSecurity OnlineBackup DeviceProtection TechSupport StreamingTV  \
0                No          Yes               No          No          No
1               Yes           No              Yes          No          No
2               Yes          Yes               No          No          No
3               Yes           No              Yes         Yes          No
4                No           No               No          No          No
…             …          …              …         …         …
7038            Yes           No              Yes         Yes         Yes
7039             No          Yes              Yes          No         Yes
7040            Yes           No               No          No          No
7041             No           No               No          No          No
7042            Yes           No              Yes         Yes         Yes

     StreamingMovies
0                 No
1                 No
2                 No
3                 No
4                 No
…               …
7038             Yes
7039             Yes
7040              No
7041              No
7042             Yes

[7043 rows x 21 columns]
```

[13]:

```
for col in telecom_df.columns:
    print(col)
```

```
customerID
tenure
PhoneService
Contract
PaperlessBilling
PaymentMethod
MonthlyCharges
TotalCharges
Churn
gender
SeniorCitizen
Partner
Dependents
MultipleLines
InternetService
OnlineSecurity
```

```
OnlineBackup
DeviceProtection
TechSupport
StreamingTV
StreamingMovies
```

[14]:
```
telecom_df.shape
```

[14]:
```
(7043, 21)
```

[15]:
```
telecom_df.describe()
```

[15]:
```
            tenure  MonthlyCharges  SeniorCitizen
count  7043.000000     7043.000000    7043.000000
mean     32.371149       64.761692       0.162147
std      24.559481       30.090047       0.368612
min       0.000000       18.250000       0.000000
25%       9.000000       35.500000       0.000000
50%      29.000000       70.350000       0.000000
75%      55.000000       89.850000       0.000000
max      72.000000      118.750000       1.000000
```

[16]:
```
telecom_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   customerID        7043 non-null   object
 1   tenure            7043 non-null   int64
 2   PhoneService      7043 non-null   object
 3   Contract          7043 non-null   object
 4   PaperlessBilling  7043 non-null   object
 5   PaymentMethod     7043 non-null   object
 6   MonthlyCharges    7043 non-null   float64
 7   TotalCharges      7043 non-null   object
 8   Churn             7043 non-null   object
 9   gender            7043 non-null   object
 10  SeniorCitizen     7043 non-null   int64
 11  Partner           7043 non-null   object
 12  Dependents        7043 non-null   object
 13  MultipleLines     7043 non-null   object
 14  InternetService   7043 non-null   object
 15  OnlineSecurity    7043 non-null   object
 16  OnlineBackup      7043 non-null   object
 17  DeviceProtection  7043 non-null   object
 18  TechSupport       7043 non-null   object
 19  StreamingTV       7043 non-null   object
 20  StreamingMovies   7043 non-null   object
dtypes: float64(1), int64(2), object(18)
memory usage: 1.2+ MB
```

3 Data Cleaning

[17]:
```
telecom_df.isnull().sum()*100/telecom_df.shape[0]
```

[17]:
```
customerID          0.0
tenure              0.0
PhoneService        0.0
Contract            0.0
PaperlessBilling    0.0
PaymentMethod       0.0
MonthlyCharges      0.0
TotalCharges        0.0
Churn               0.0
gender              0.0
SeniorCitizen       0.0
Partner             0.0
Dependents          0.0
MultipleLines       0.0
InternetService     0.0
OnlineSecurity      0.0
OnlineBackup        0.0
DeviceProtection    0.0
TechSupport         0.0
StreamingTV         0.0
StreamingMovies     0.0
dtype: float64
```

[18]:
```
telecom_df['TotalCharges'].describe()
```

[18]:
```
count     7043
unique    6531
top
freq        11
Name: TotalCharges, dtype: object
```

[19]:
```
print(telecom_df['MonthlyCharges'])
```

```
0        29.85
1        56.95
2        53.85
3        42.30
4        70.70
```

```
          …
7038     84.80
7039    103.20
7040     29.60
7041     74.40
7042    105.65
Name: MonthlyCharges, Length: 7043, dtype: float64
```

[20]:
```
telecom_df['TotalCharges'] = telecom_df['TotalCharges'].replace(' ', np.nan)
telecom_df['TotalCharges'] = pd.to_numeric(telecom_df['TotalCharges'])
```

[21]:
```
value = (telecom_df['TotalCharges']/telecom_df['MonthlyCharges']).
 ↪median()*telecom_df['MonthlyCharges']
```

[22]:
```
telecom_df['TotalCharges'].describe()
```

[22]:
```
count    7032.000000
mean     2283.300441
std      2266.771362
min        18.800000
25%       401.450000
50%      1397.475000
75%      3794.737500
max      8684.800000
Name: TotalCharges, dtype: float64
```

[23]:
```
telecom_df['TotalCharges'] = value.where(telecom_df['TotalCharges'] == np.nan,␣
 ↪other =telecom_df['TotalCharges'])
```

[24]:
```
telecom_df['TotalCharges'].describe()
```

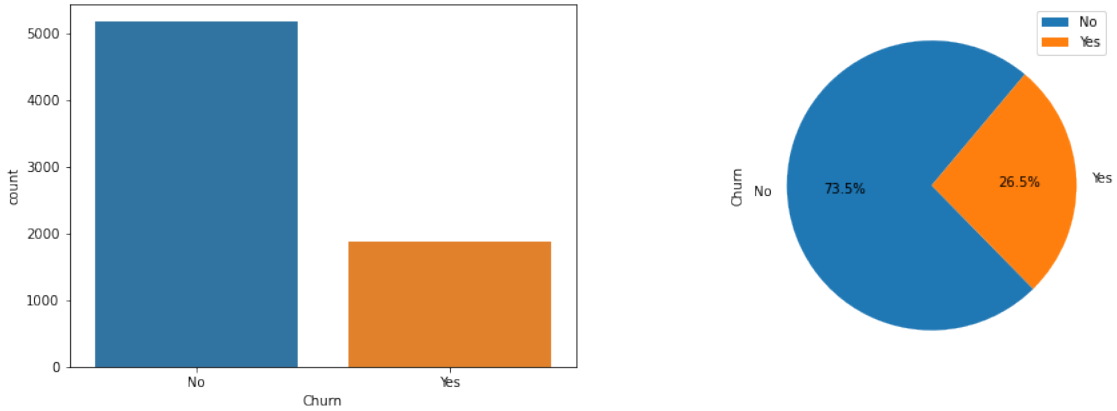
[24]:
```
count    7032.000000
mean     2283.300441
std      2266.771362
min        18.800000
25%       401.450000
50%      1397.475000
75%      3794.737500
max      8684.800000
Name: TotalCharges, dtype: float64
```

4 Data Analysis

[25]:
```
telecom_df.Churn.describe()
```

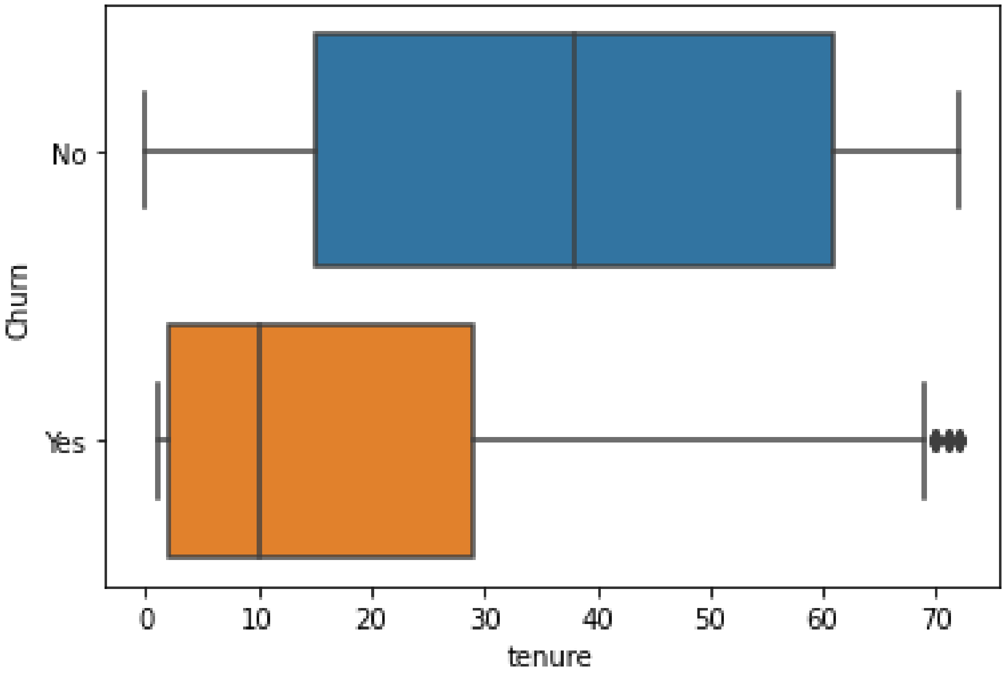
```
[25]: count      7043
      unique        2
      top          No
      freq       5174
      Name: Churn, dtype: object
```

```
[26]: fig, axs = plt.subplots(1,2, figsize = (15,5))
      plt1 = sns.countplot(telecom_df['Churn'], ax = axs[0])

      pie_churn = pd.DataFrame(telecom_df['Churn'].value_counts())
      pie_churn.plot.pie( subplots=True,labels = pie_churn.index.values, autopct='%1.
       ↪1f%%', figsize = (15,5), startangle= 50, ax = axs[1])
      # Unsquish the pie.

      plt.gca().set_aspect('equal')
      plt.show()
```



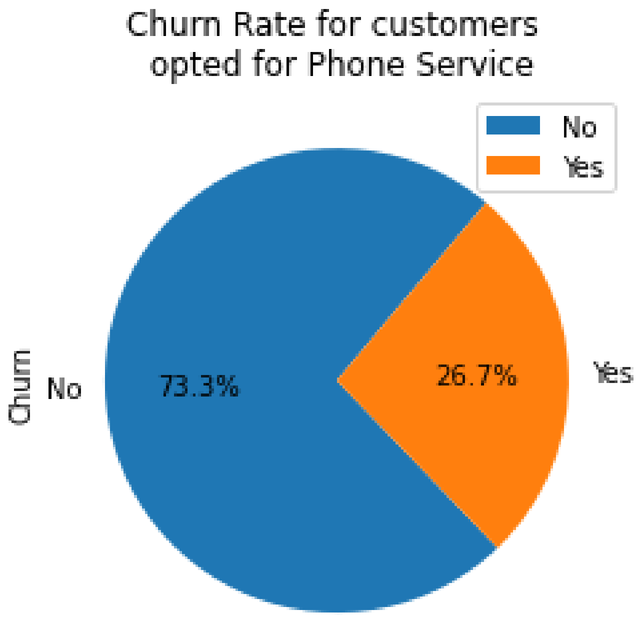
5 Tenure

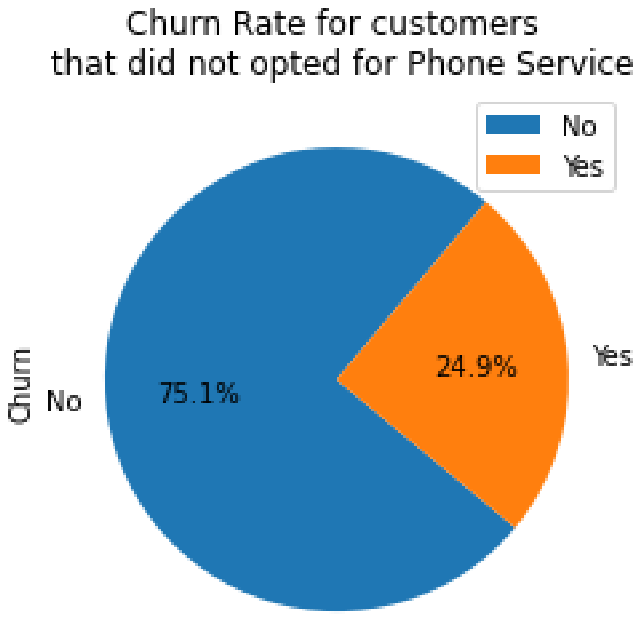
```
[27]: sns.boxplot(x = 'tenure', y = 'Churn', data = telecom_df)
      plt.show()
```



6 Phone Service

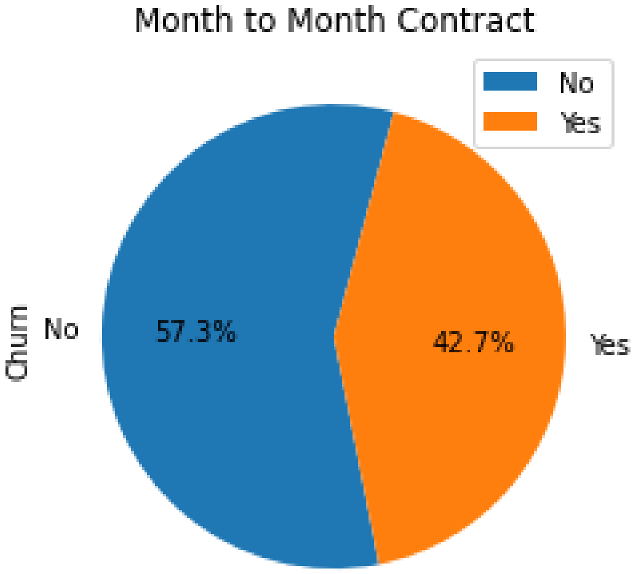
[28]:
```
pie_PhoneService_Yes = pd.DataFrame(telecom_df[telecom_df['PhoneService'] ==␣
↪"Yes"]['Churn'].value_counts())
pie_PhoneService_Yes.plot.pie(subplots=True, labels = pie_PhoneService_Yes.
↪index.values, autopct='%1.1f%%', startangle= 50 )
plt.title('Churn Rate for customers \n opted for Phone Service')
plt.gca().set_aspect('equal')

pie_PhoneService_No = pd.DataFrame(telecom_df[telecom_df['PhoneService'] ==␣
↪"No"]['Churn'].value_counts())
pie_PhoneService_No.plot.pie(subplots=True, labels = pie_PhoneService_Yes.index.
↪values, autopct='%1.1f%%', startangle= 50)
plt.title('Churn Rate for customers \n that did not opted for Phone Service')
plt.gca().set_aspect('equal')

plt.show()
```

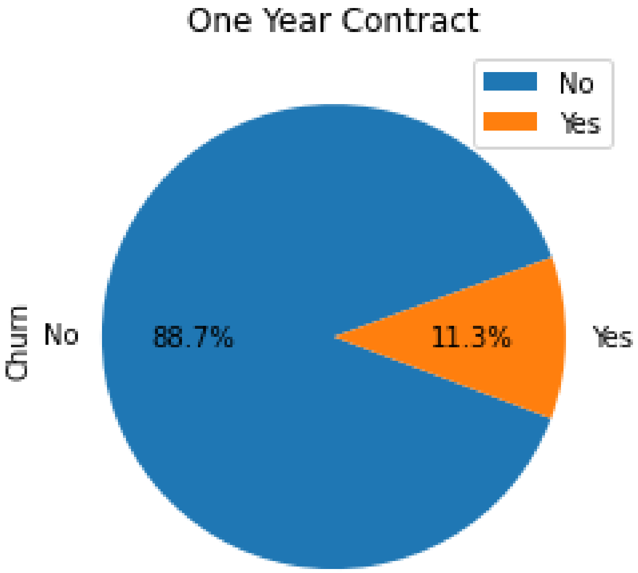

Churn Rate for customers
opted for Phone Service
No
Yes
Churn
No
73.3%
26.7%
Yes


Churn Rate for customers
that did not opted for Phone Service
No
Yes
Churn
No
75.1%
24.9%
Yes

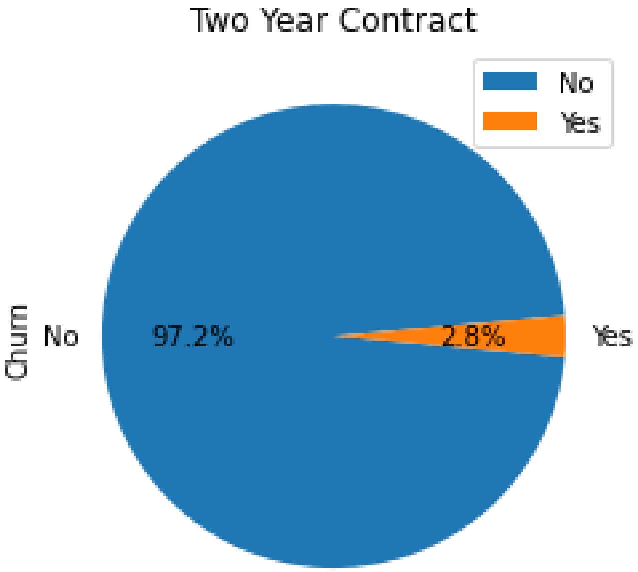
7 Contract

```
[29]: pie_Contract_m2m = pd.DataFrame(telecom_df[telecom_df['Contract'] ==␣
       ↪"Month-to-month"]['Churn'].value_counts())
      pie_Contract_m2m.plot.pie(subplots=True, labels = pie_Contract_m2m.index.
       ↪values, autopct='%1.1f%%', startangle= 75)
      plt.title('Month to Month Contract')
      plt.gca().set_aspect('equal')

      pie_Contract_1y = pd.DataFrame(telecom_df[telecom_df['Contract'] == "One␣
       ↪year"]['Churn'].value_counts())
      pie_Contract_1y.plot.pie(subplots=True, labels = pie_Contract_1y.index.values,␣
       ↪autopct='%1.1f%%', startangle= 20)
      plt.title('One Year Contract')
      plt.gca().set_aspect('equal')

      pie_Contract_2y = pd.DataFrame(telecom_df[telecom_df['Contract'] == "Two␣
       ↪year"]['Churn'].value_counts())
      pie_Contract_2y.plot.pie(subplots=True, labels = pie_Contract_2y.index.values,␣
       ↪autopct='%1.1f%%', startangle= 5)
      plt.title('Two Year Contract')
      plt.gca().set_aspect('equal')

      plt.show()
```





One Year Contract



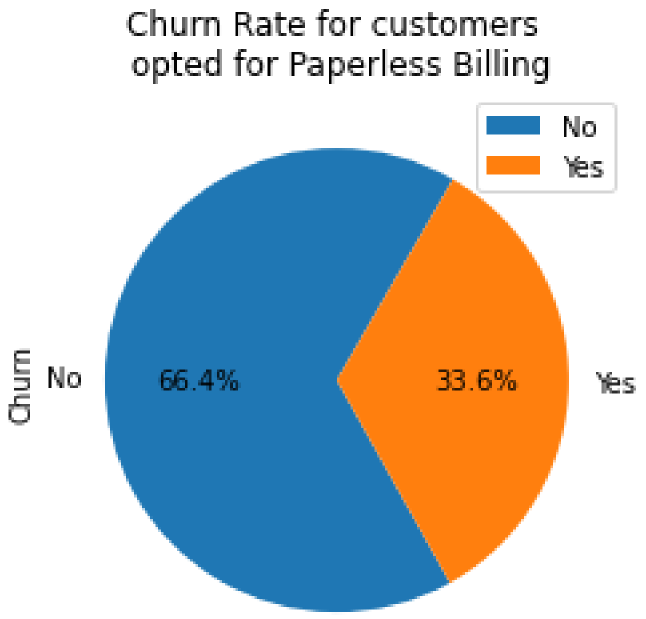
Two Year Contract

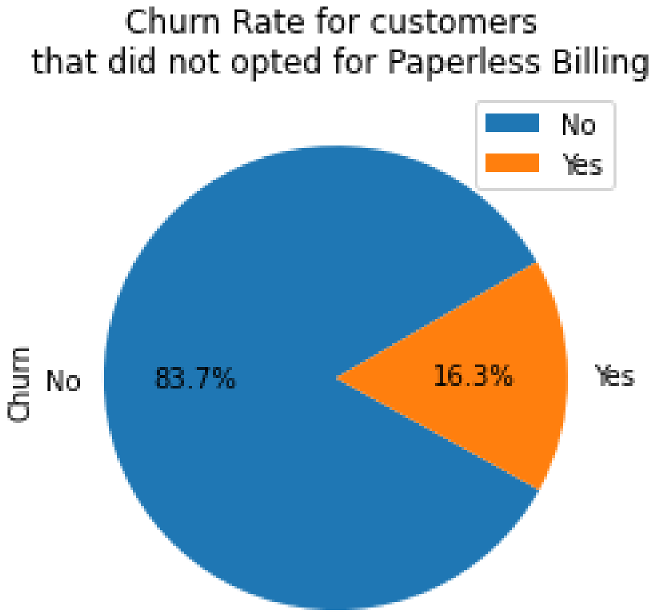
8 Paperless Bills

[30]:

```
plt.figure(figsize=(15,5))

pie_PaperlessBilling_Yes = pd.
 ↪DataFrame(telecom_df[telecom_df['PaperlessBilling'] == "Yes"]['Churn'].
 ↪value_counts())
pie_PaperlessBilling_Yes.plot.pie(subplots=True, labels =␣
 ↪pie_PaperlessBilling_Yes.index.values, autopct='%1.1f%%', startangle= 60)
plt.title('Churn Rate for customers \n opted for Paperless Billing')
plt.gca().set_aspect('equal')


pie_PaperlessBilling_No = pd.
 ↪DataFrame(telecom_df[telecom_df['PaperlessBilling'] == "No"]['Churn'].
 ↪value_counts())
pie_PaperlessBilling_No.plot.pie(subplots=True, labels =␣
 ↪pie_PaperlessBilling_No.index.values, autopct='%1.1f%%', startangle= 30)
plt.title('Churn Rate for customers \n that did not opted for Paperless␣
 ↪Billing')
plt.gca().set_aspect('equal')

plt.show()
```

```
<Figure size 1080x360 with 0 Axes>
```





9 Payment Method

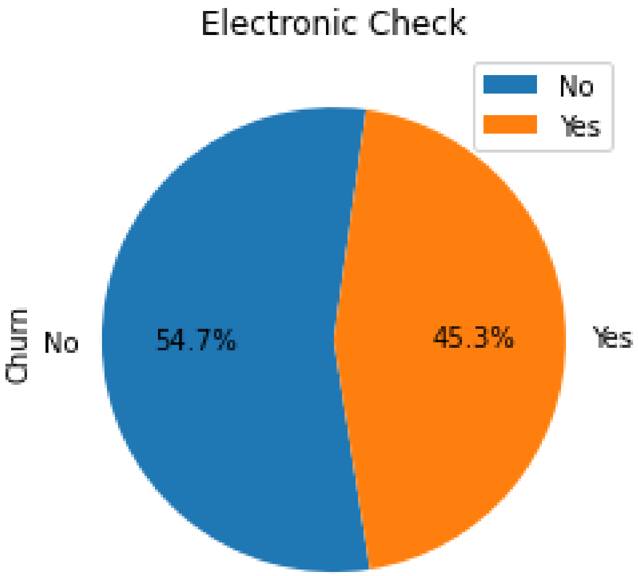
```
[31]: telecom_df.PaymentMethod.describe()
```

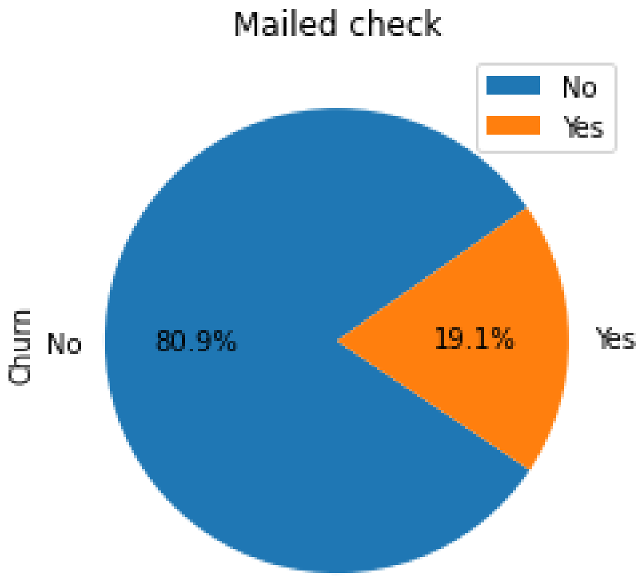
```
[31]: count                   7043
      unique                     4
      top       Electronic check
      freq                    2365
      Name: PaymentMethod, dtype: object
```

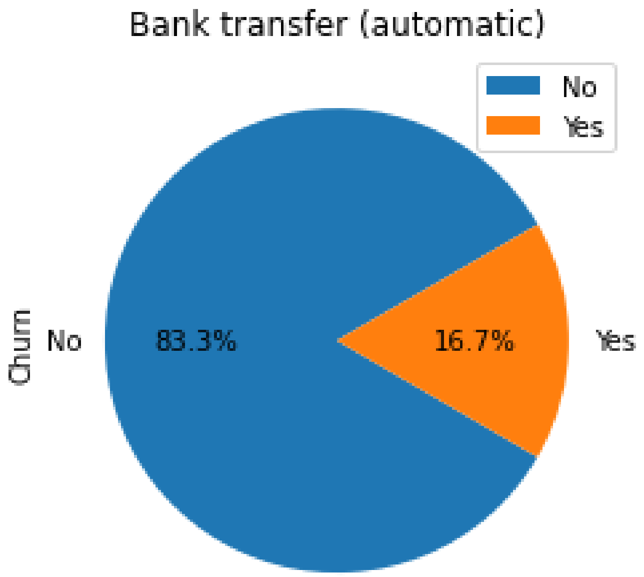
```
[32]: plt.figure(figsize=(15,10))
pie_PaymentMethod_ec = pd.DataFrame(telecom_df[telecom_df['PaymentMethod'] ==␣
 ↪"Electronic check"]['Churn'].value_counts())
pie_PaymentMethod_ec.plot.pie(subplots=True, labels = pie_PaymentMethod_ec.
 ↪index.values, autopct='%1.1f%%', startangle= 82)
plt.title('Electronic Check')
plt.gca().set_aspect('equal')

pie_PaymentMethod_mc = pd.DataFrame(telecom_df[telecom_df['PaymentMethod'] ==␣
 ↪"Mailed check"]['Churn'].value_counts())
pie_PaymentMethod_mc.plot.pie(subplots=True, labels = pie_PaymentMethod_mc.
 ↪index.values, autopct='%1.1f%%', startangle= 35)
```

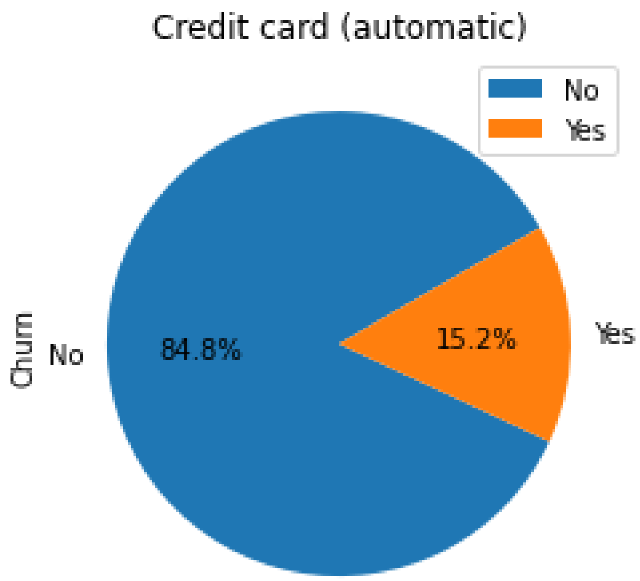
```
plt.title('Mailed check')
plt.gca().set_aspect('equal')

pie_PaymentMethod_bta = pd.DataFrame(telecom_df[telecom_df['PaymentMethod'] ==␣
 ↪"Bank transfer (automatic)"]['Churn'].value_counts())
pie_PaymentMethod_bta.plot.pie(subplots=True, labels = pie_PaymentMethod_bta.
 ↪index.values, autopct='%1.1f%%', startangle= 30)
plt.title('Bank transfer (automatic)')
plt.gca().set_aspect('equal')

pie_PaymentMethod_cca = pd.DataFrame(telecom_df[telecom_df['PaymentMethod'] ==␣
 ↪"Credit card (automatic)"]['Churn'].value_counts())
pie_PaymentMethod_cca.plot.pie(subplots=True, labels = pie_PaymentMethod_cca.
 ↪index.values, autopct='%1.1f%%', startangle= 30)
plt.title('Credit card (automatic)')
plt.gca().set_aspect('equal')

plt.show()
```

```
<Figure size 1080x720 with 0 Axes>
```



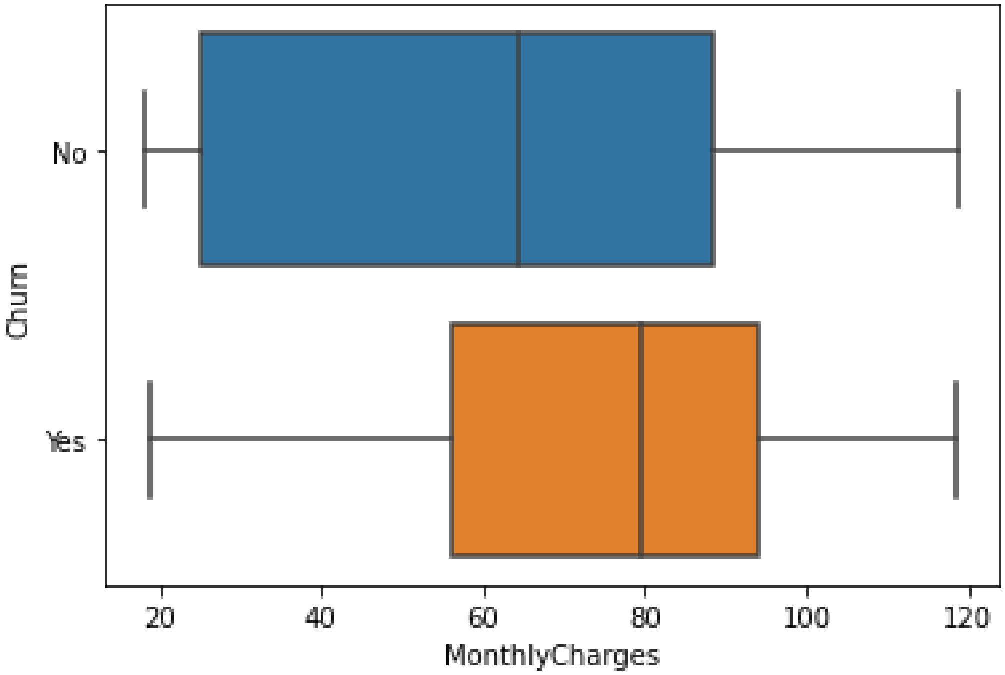

Mailed check


Bank transfer (automatic)



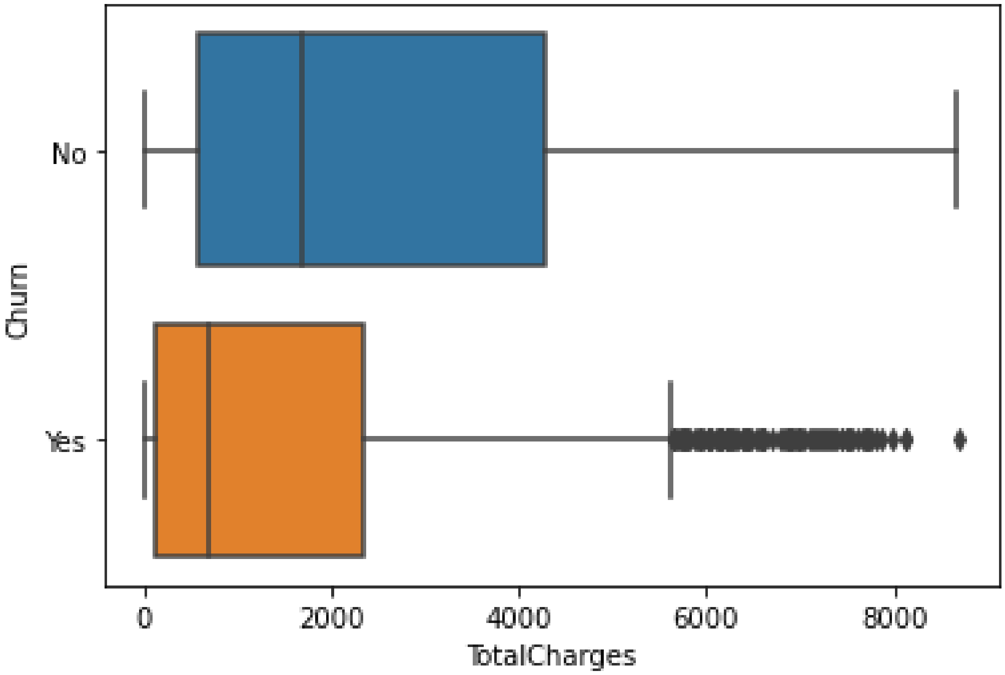
10 Monthly Charges

[33]:
```
sns.boxplot(x = 'MonthlyCharges', y = 'Churn', data = telecom_df)
plt.show()
```



11 Total Charges

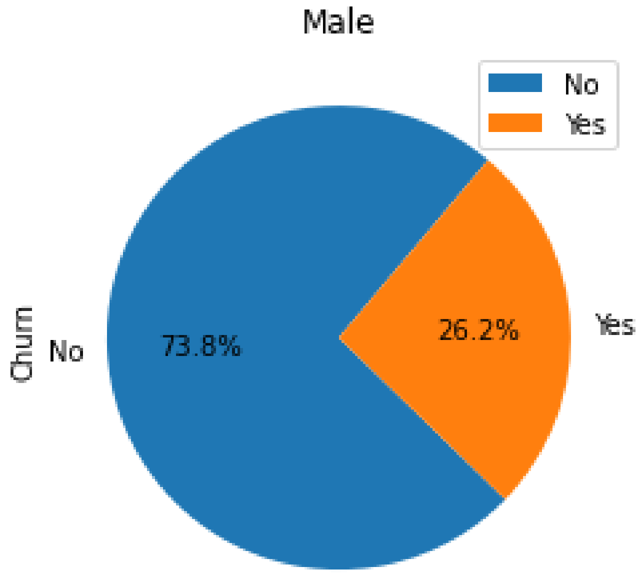
[34]:
```
sns.boxplot(x = 'TotalCharges', y= 'Churn', data = telecom_df)
plt.show()
```

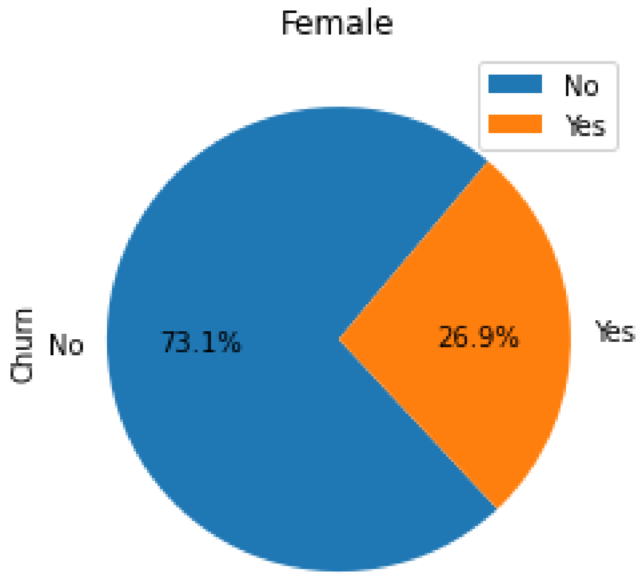


12 Gender

[35]:
```
plt.figure(figsize=(15,5))
pie_Gender_M = pd.DataFrame(telecom_df[telecom_df['gender'] == "Male"]['Churn'].
 ↪value_counts())
pie_Gender_M.plot.pie(subplots = True, labels = pie_Gender_M.index.values,␣
 ↪autopct='%1.1f%%', startangle= 50)
plt.title('Male')
plt.gca().set_aspect('equal')

pie_Gender_F = pd.DataFrame(telecom_df[telecom_df['gender'] ==␣
 ↪"Female"]['Churn'].value_counts())
pie_Gender_F.plot.pie(subplots = True,  labels = pie_Gender_F.index.values,␣
 ↪autopct='%1.1f%%', startangle= 50)
plt.title('Female')
plt.gca().set_aspect('equal')

plt.show()
```

```
<Figure size 1080x360 with 0 Axes>
```

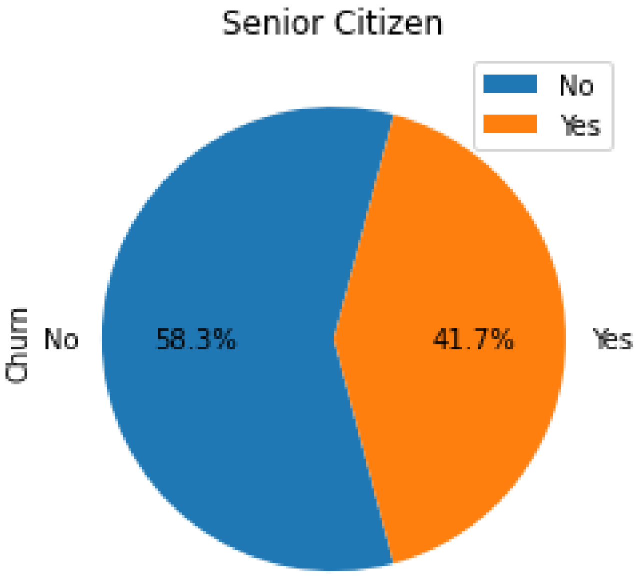


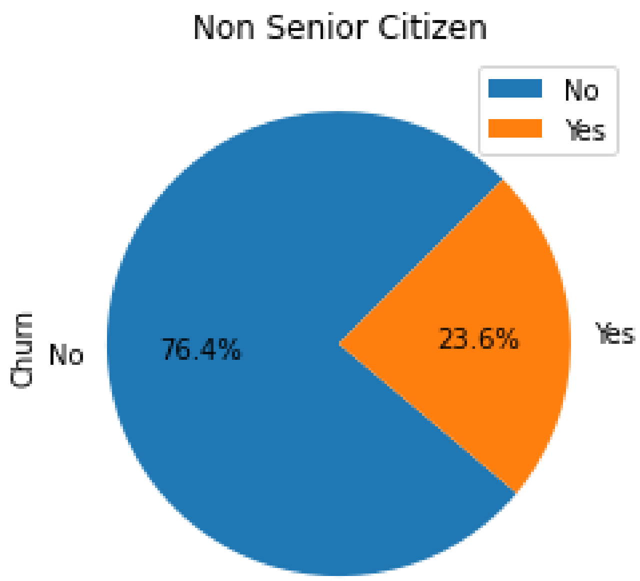


13 Senior Citizen

[36]:
```
plt.figure(figsize=(15,5))
pie_SeniorCitizen_Y = pd.DataFrame(telecom_df[telecom_df['SeniorCitizen'] ==␣
 ↪1]['Churn'].value_counts())
pie_SeniorCitizen_Y.plot.pie(subplots = True, labels = pie_SeniorCitizen_Y.
 ↪index.values, autopct='%1.1f%%', startangle= 75)
plt.title('Senior Citizen')
plt.gca().set_aspect('equal')

pie_SeniorCitizen_N = pd.DataFrame(telecom_df[telecom_df['SeniorCitizen'] ==␣
 ↪0]['Churn'].value_counts())
pie_SeniorCitizen_N.plot.pie(subplots = True, labels = pie_SeniorCitizen_N.
 ↪index.values, autopct='%1.1f%%', startangle= 45)
plt.title('Non Senior Citizen')

plt.gca().set_aspect('equal')
plt.show()
```

```
<Figure size 1080x360 with 0 Axes>
```



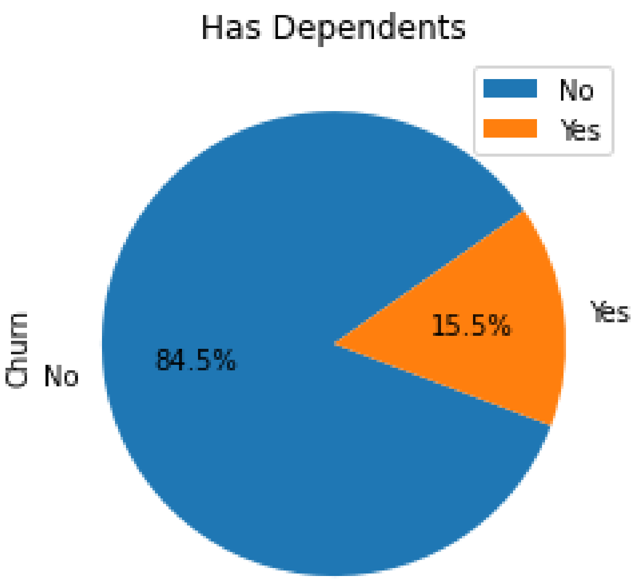


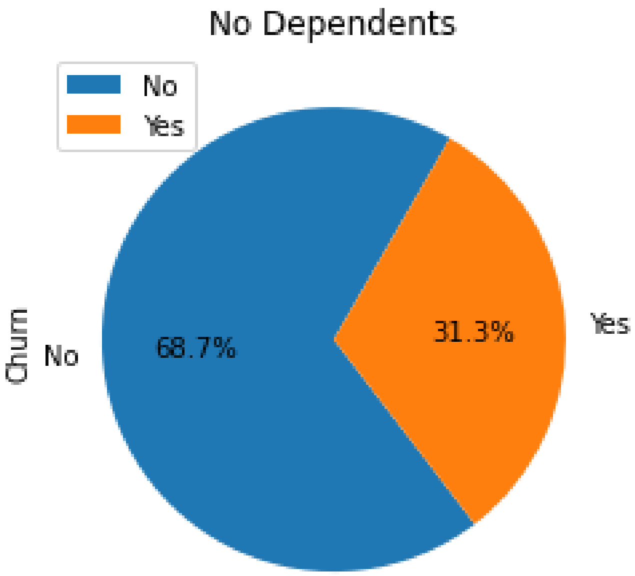
14 Dependents

[37]:

```
plt.figure(figsize=(15,5))
pie_Dependents_Y = pd.DataFrame(telecom_df[telecom_df['Dependents'] ==␣
 ↪'Yes']['Churn'].value_counts())
pie_Dependents_Y.plot.pie(subplots = True,  labels = pie_Dependents_Y.index.
 ↪values, autopct='%1.1f%%', startangle= 35)
plt.title('Has Dependents')
plt.gca().set_aspect('equal')

pie_Dependents_N = pd.DataFrame(telecom_df[telecom_df['Dependents'] ==␣
 ↪'No']['Churn'].value_counts())
pie_Dependents_N.plot.pie(subplots = True,  labels = pie_Dependents_N.index.
 ↪values, autopct='%1.1f%%', startangle= 60)
plt.title('No Dependents')

plt.gca().set_aspect('equal')
plt.show()
```

```
<Figure size 1080x360 with 0 Axes>
```

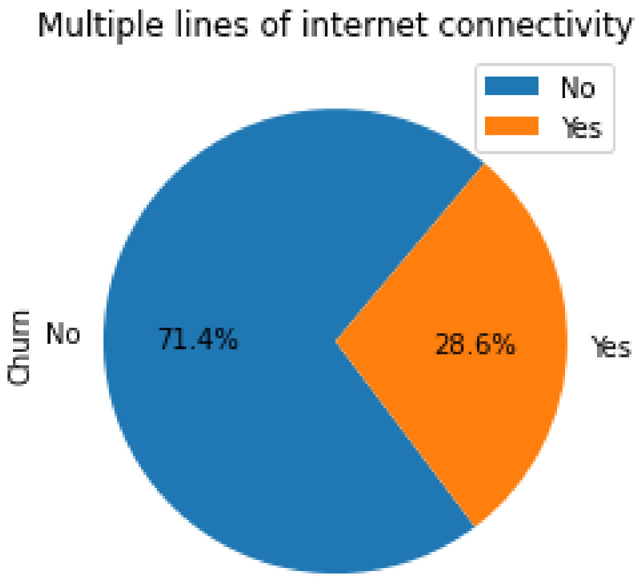

Has Dependents
No
Yes
Churn
No
84.5%
15.5%
Yes

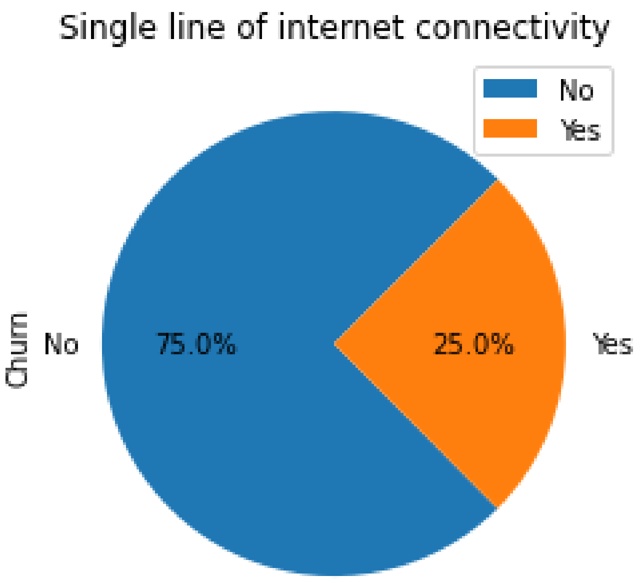

No Dependents
No
Yes
Churn
No
68.7%
31.3%
Yes

15 Multiple Lines

```
[38]: plt.figure(figsize=(15,5))
pie_MultipleLines_Y = pd.DataFrame(telecom_df[telecom_df['MultipleLines'] ==␣
 ↪'Yes']['Churn'].value_counts())
pie_MultipleLines_Y.plot.pie(subplots = True,  labels = pie_MultipleLines_Y.
 ↪index.values, autopct='%1.1f%%', startangle= 50)
plt.title('Multiple lines of internet connectivity')
plt.gca().set_aspect('equal')

pie_MultipleLines_N = pd.DataFrame(telecom_df[telecom_df['MultipleLines'] ==␣
 ↪'No']['Churn'].value_counts())
pie_MultipleLines_N.plot.pie(subplots = True,  labels = pie_MultipleLines_N.
 ↪index.values, autopct='%1.1f%%', startangle= 45)
plt.title('Single line of internet connectivity')

plt.gca().set_aspect('equal')
plt.show()
```

```
<Figure size 1080x360 with 0 Axes>
```





[39]:
```
import jovian
```

[40]:
```
jovian.commit
```

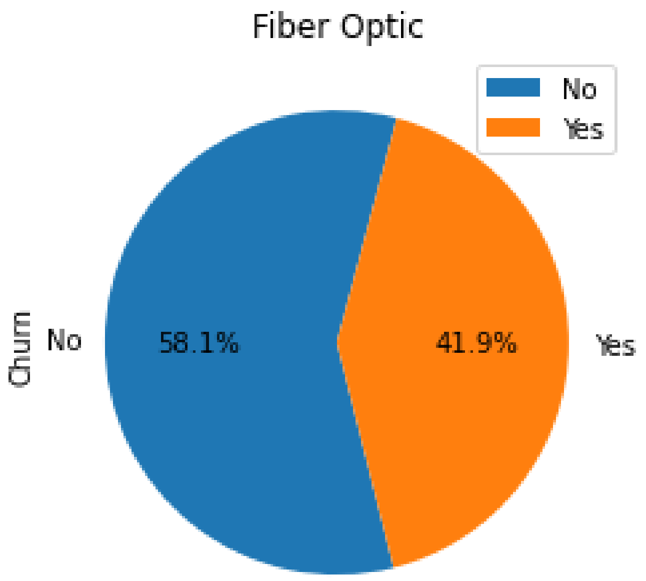
[40]:
```
<function jovian.utils.commit.commit(message=None, files=[], outputs=[],
environment=None, privacy='auto', filename=None, project=None, new_project=None,
git_commit=False, git_message='auto', require_write_access=False, **kwargs)>
```

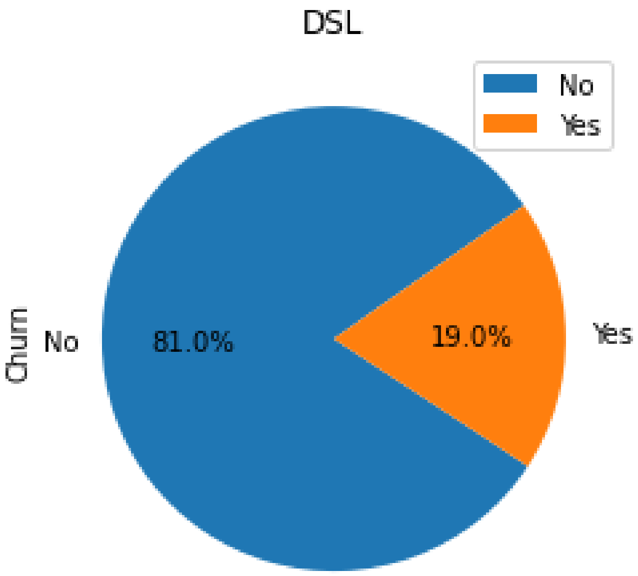
16 Internet service

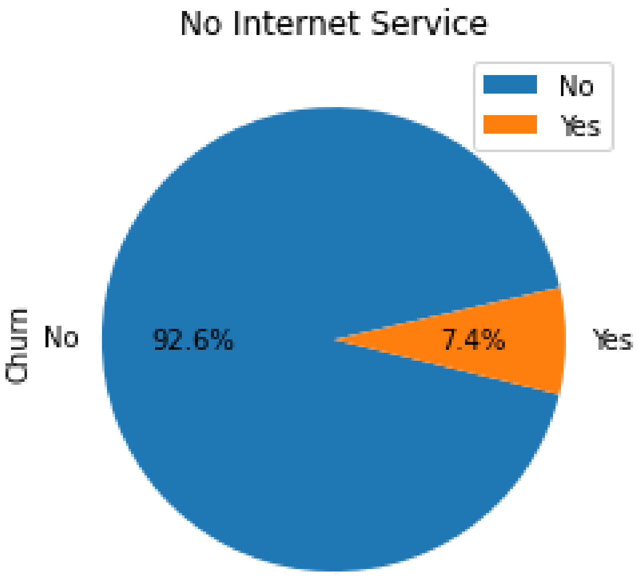
[41]:
```
plt.figure(figsize=(15,5))
pie_InternetService_fo = pd.DataFrame(telecom_df[telecom_df['InternetService']␣
 ↪== "Fiber optic"]['Churn'].value_counts())
pie_InternetService_fo.plot.pie(subplots = True, labels =␣
 ↪pie_InternetService_fo.index.values, autopct='%1.1f%%', startangle= 75)
plt.title('Fiber Optic')
plt.gca().set_aspect('equal')

pie_InternetService_dsl = pd.DataFrame(telecom_df[telecom_df['InternetService']␣
 ↪== "DSL"]['Churn'].value_counts())
pie_InternetService_dsl.plot.pie(subplots = True, labels =␣
 ↪pie_InternetService_dsl.index.values, autopct='%1.1f%%', startangle= 35)
plt.title('DSL')
plt.gca().set_aspect('equal')
```

```
pie_InternetService_no = pd.DataFrame(telecom_df[telecom_df['InternetService']␣
 ↪== "No"]['Churn'].value_counts())
pie_InternetService_no.plot.pie(subplots = True, labels =␣
 ↪pie_InternetService_no.index.values, autopct='%1.1f%%', startangle= 13)
plt.title('No Internet Service')
plt.gca().set_aspect('equal')

plt.show()
```

```
<Figure size 1080x360 with 0 Axes>
```



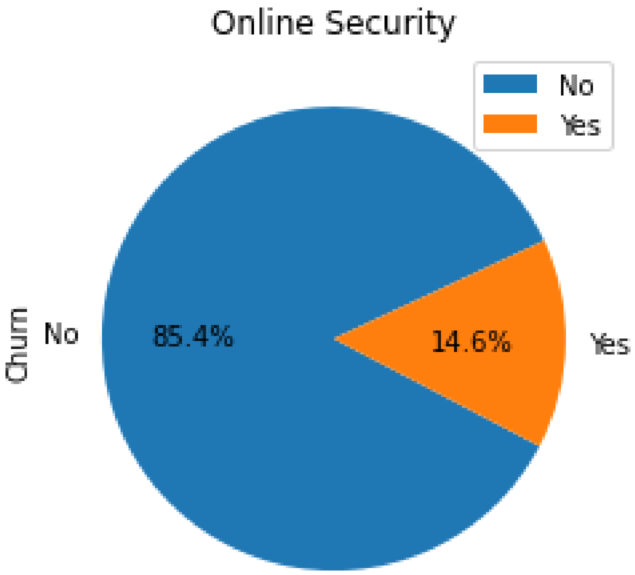

DSL
No
Yes
Churn
No
81.0%
19.0%
Yes

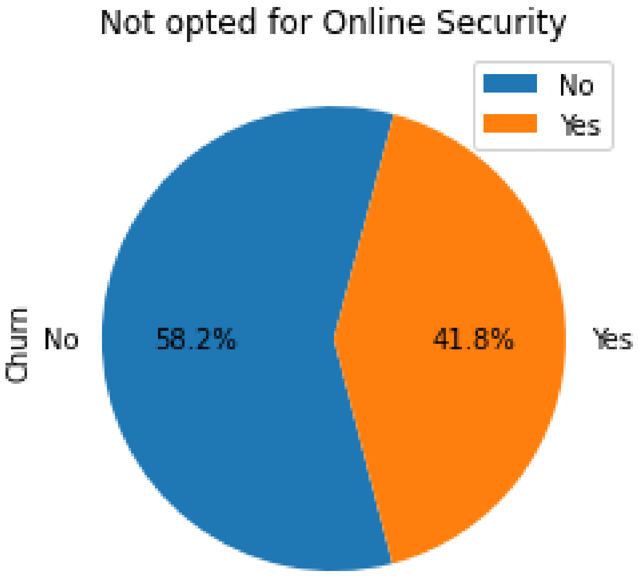

No Internet Service
No
Yes
Churn
No
92.6%
7.4%
Yes

17 Online Security

```
[42]: plt.figure(figsize=(15,5))
      pie_OnlineSecurity_Y = pd.DataFrame(telecom_df[telecom_df['OnlineSecurity'] ==␣
       ↪'Yes']['Churn'].value_counts())
      pie_OnlineSecurity_Y.plot.pie(subplots = True,  labels = pie_OnlineSecurity_Y.
       ↪index.values, autopct='%1.1f%%', startangle= 25)
      plt.title('Online Security')
      plt.gca().set_aspect('equal')

      pie_OnlineSecurity_N = pd.DataFrame(telecom_df[telecom_df['OnlineSecurity'] ==␣
       ↪'No']['Churn'].value_counts())
      pie_OnlineSecurity_N.plot.pie(subplots = True, labels = pie_OnlineSecurity_N.
       ↪index.values, autopct='%1.1f%%', startangle= 75)
      plt.title('Not opted for Online Security')
      plt.gca().set_aspect('equal')
      plt.show()
```

```
<Figure size 1080x360 with 0 Axes>
```

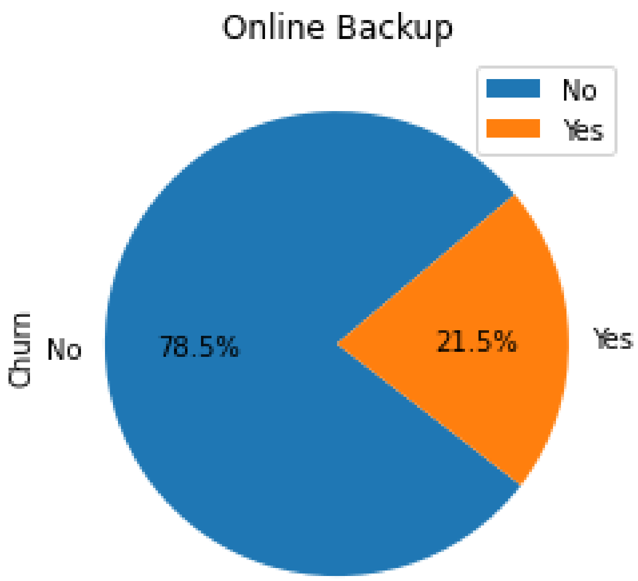


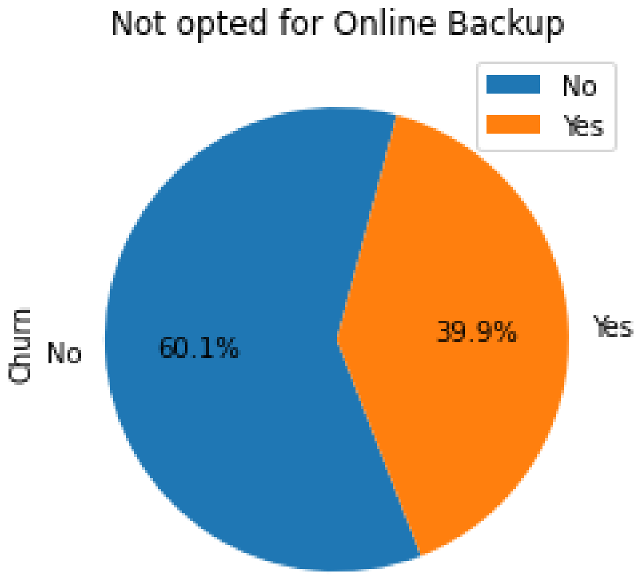


18 online backup

```
[43]: plt.figure(figsize=(15,5))
pie_OnlineBackup_Y = pd.DataFrame(telecom_df[telecom_df['OnlineBackup'] ==␣
 ↪'Yes']['Churn'].value_counts())
pie_OnlineBackup_Y.plot.pie(subplots = True,  labels = pie_OnlineBackup_Y.index.
 ↪values, autopct='%1.1f%%', startangle= 40)
plt.title('Online Backup')
plt.gca().set_aspect('equal')

pie_OnlineBackup_N = pd.DataFrame(telecom_df[telecom_df['OnlineBackup'] ==␣
 ↪'No']['Churn'].value_counts())
pie_OnlineBackup_N.plot.pie(subplots = True, labels = pie_OnlineBackup_N.index.
 ↪values, autopct='%1.1f%%', startangle= 75)
plt.title('Not opted for Online Backup')
plt.gca().set_aspect('equal')

plt.show()
```

```
<Figure size 1080x360 with 0 Axes>
```


Online Backup

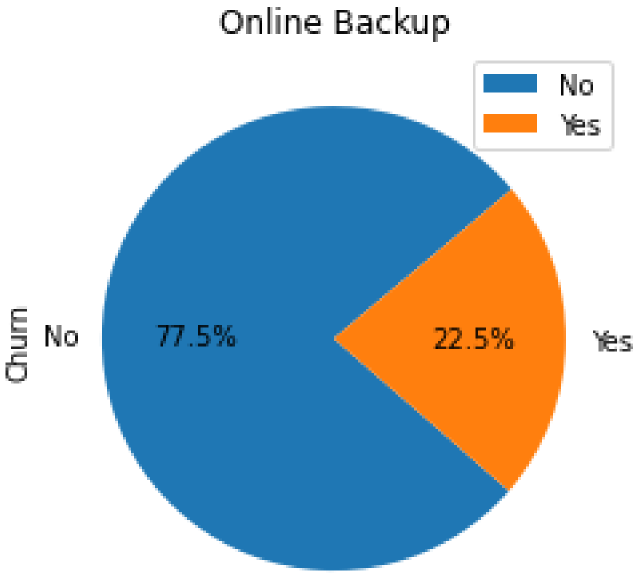

Not opted for Online Backup

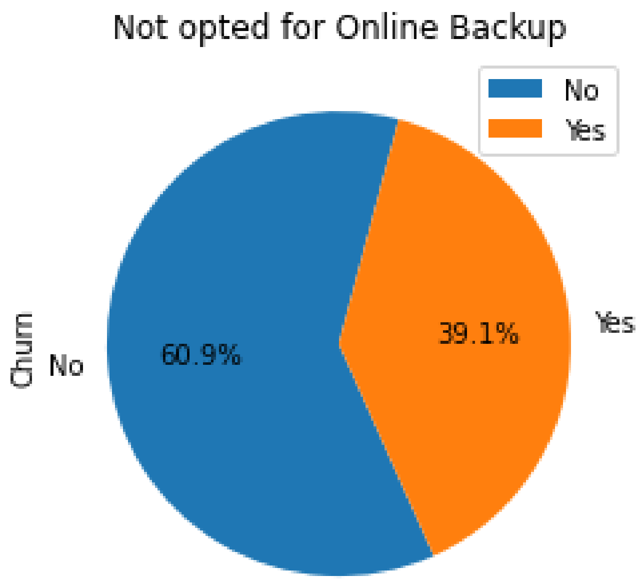
19 Device Protection

[44]:

```python
plt.figure(figsize=(15,5))

pie_DeviceProtection_Y = pd.DataFrame(telecom_df[telecom_df['DeviceProtection'] == 'Yes']['Churn'].value_counts())
pie_DeviceProtection_Y.plot.pie(subplots = True, labels = pie_DeviceProtection_Y.index.values, autopct='%1.1f%%', startangle= 40)
plt.title('Online Backup')
plt.gca().set_aspect('equal')

pie_DeviceProtection_N = pd.DataFrame(telecom_df[telecom_df['DeviceProtection'] == 'No']['Churn'].value_counts())
pie_DeviceProtection_N.plot.pie(subplots = True, labels = pie_DeviceProtection_N.index.values, autopct='%1.1f%%', startangle= 75)
plt.title('Not opted for Online Backup')
plt.gca().set_aspect('equal')
plt.show()
```

```
<Figure size 1080x360 with 0 Axes>
```

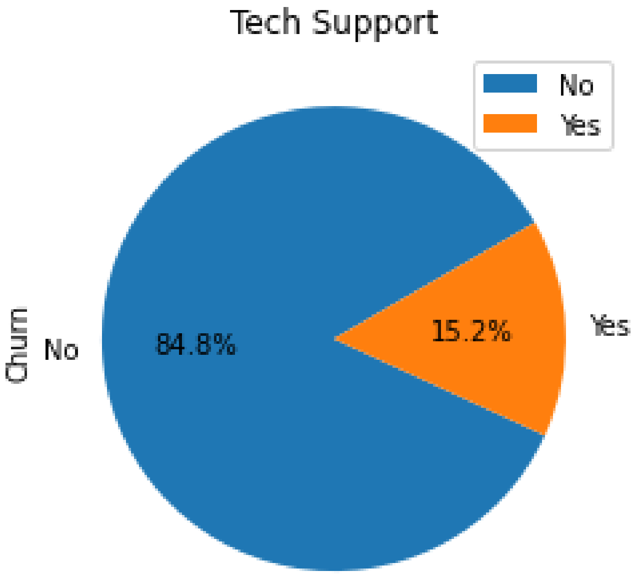


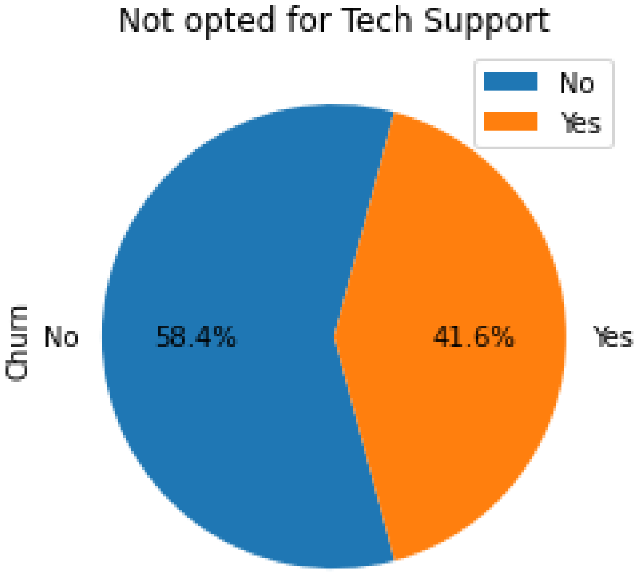


20 Tech Support

```
[45]: plt.figure(figsize=(15,5))
pie_TechSupport_Y = pd.DataFrame(telecom_df[telecom_df['TechSupport'] ==␣
 ↪'Yes']['Churn'].value_counts())
pie_TechSupport_Y.plot.pie(subplots = True,labels = pie_TechSupport_Y.index.
 ↪values, autopct='%1.1f%%', startangle= 30)
plt.title('Tech Support')
plt.gca().set_aspect('equal')

pie_TechSupport_N = pd.DataFrame(telecom_df[telecom_df['TechSupport'] ==␣
 ↪'No']['Churn'].value_counts())
pie_TechSupport_N.plot.pie(subplots = True, labels = pie_TechSupport_N.index.
 ↪values, autopct='%1.1f%%', startangle= 75)
plt.title('Not opted for Tech Support')

plt.gca().set_aspect('equal')
plt.show()
```

```
<Figure size 1080x360 with 0 Axes>
```


Tech Support
No
Yes
Churn
No
84.8%
15.2%
Yes


Not opted for Tech Support
No
Yes
Churn
No
58.4%
41.6%
Yes

21 Streaming Tv and Movies doesn't make such impact on churning.

22 Model Building

DATA PREPRATION

[46]:
```
# List of variables to map

varlist =  ['PhoneService', 'PaperlessBilling', 'Churn', 'Partner',␣
 ↪'Dependents']

# Defining the map function
def binary_map(x):
    return x.map({'Yes': 1, "No": 0})

# Applying the function to the housing list
telecom_df[varlist] = telecom_df[varlist].apply(binary_map)
```

[47]:
```
telecom_df.head()
```

[47]:
```
  customerID  tenure  PhoneService        Contract  PaperlessBilling  \
0  7590-VHVEG       1             0  Month-to-month                 1
1  5575-GNVDE      34             1        One year                 0
2  3668-QPYBK       2             1  Month-to-month                 1
3  7795-CFOCW      45             0        One year                 0
4  9237-HQITU       2             1  Month-to-month                 1

               PaymentMethod  MonthlyCharges  TotalCharges  Churn  gender  \
0           Electronic check           29.85         29.85      0  Female
1               Mailed check           56.95       1889.50      0    Male
2               Mailed check           53.85        108.15      1    Male
3  Bank transfer (automatic)           42.30       1840.75      0    Male
4           Electronic check           70.70        151.65      1  Female

   …  Partner  Dependents     MultipleLines InternetService OnlineSecurity  \
0  …        1           0  No phone service             DSL             No
1  …        0           0                No             DSL            Yes
2  …        0           0                No             DSL            Yes
3  …        0           0  No phone service             DSL            Yes
4  …        0           0                No     Fiber optic             No

  OnlineBackup DeviceProtection TechSupport StreamingTV StreamingMovies
0          Yes               No          No          No              No
1           No              Yes          No          No              No
2          Yes               No          No          No              No
3           No              Yes         Yes          No              No
```

```
4            No                 No           No           No                No

[5 rows x 21 columns]
```

23 For categorical variables with multiple levels, create dummy features (one-hot encoded)

[48]:

```python
# Creating a dummy variable for some of the categorical variables and dropping␣
↪the first one.
dummy1 = pd.get_dummies(telecom_df[['Contract', 'PaymentMethod', 'gender',␣
↪'InternetService']], drop_first=True)

# Adding the results to the master dataframe
telecom_df = pd.concat([telecom_df, dummy1], axis=1)
```

[49]:

```python
telecom_df.head()
```

[49]:

```
   customerID  tenure  PhoneService        Contract  PaperlessBilling  \
0  7590-VHVEG       1             0  Month-to-month                 1
1  5575-GNVDE      34             1        One year                 0
2  3668-QPYBK       2             1  Month-to-month                 1
3  7795-CFOCW      45             0        One year                 0
4  9237-HQITU       2             1  Month-to-month                 1

               PaymentMethod  MonthlyCharges  TotalCharges  Churn  gender  \
0           Electronic check           29.85         29.85      0  Female
1               Mailed check           56.95       1889.50      0    Male
2               Mailed check           53.85        108.15      1    Male
3  Bank transfer (automatic)           42.30       1840.75      0    Male
4           Electronic check           70.70        151.65      1  Female

   … StreamingTV StreamingMovies Contract_One year Contract_Two year  \
0  …          No              No                 0                 0
1  …          No              No                 1                 0
2  …          No              No                 0                 0
3  …          No              No                 1                 0
4  …          No              No                 0                 0

  PaymentMethod_Credit card (automatic) PaymentMethod_Electronic check  \
0                                     0                              1
1                                     0                              0
2                                     0                              0
3                                     0                              0
4                                     0                              1
```

```
  PaymentMethod_Mailed check gender_Male InternetService_Fiber optic  \
0                          0           0                           0
1                          1           1                           0
2                          1           1                           0
3                          0           1                           0
4                          0           0                           1

  InternetService_No
0                  0
1                  0
2                  0
3                  0
4                  0

[5 rows x 29 columns]
```

[50]:

```
# Creating dummy variables for the remaining categorical variables and dropping␣
↪the level with big names.

# Creating dummy variables for the variable 'MultipleLines'
ml = pd.get_dummies(telecom_df['MultipleLines'], prefix='MultipleLines')
# Dropping MultipleLines_No phone service column
ml1 = ml.drop(['MultipleLines_No phone service'], 1)
#Adding the results to the master dataframe
telecom_df = pd.concat([telecom_df,ml1], axis=1)

# Creating dummy variables for the variable 'OnlineSecurity'.
os = pd.get_dummies(telecom_df['OnlineSecurity'], prefix='OnlineSecurity')
os1 = os.drop(['OnlineSecurity_No internet service'], 1)
# Adding the results to the master dataframe
telecom_df = pd.concat([telecom_df,os1], axis=1)

# Creating dummy variables for the variable 'OnlineBackup'.
ob = pd.get_dummies(telecom_df['OnlineBackup'], prefix='OnlineBackup')
ob1 = ob.drop(['OnlineBackup_No internet service'], 1)
# Adding the results to the master dataframe
telecom_df = pd.concat([telecom_df,ob1], axis=1)

# Creating dummy variables for the variable 'DeviceProtection'.
dp = pd.get_dummies(telecom_df['DeviceProtection'], prefix='DeviceProtection')
dp1 = dp.drop(['DeviceProtection_No internet service'], 1)
# Adding the results to the master dataframe
telecom_df = pd.concat([telecom_df,dp1], axis=1)

# Creating dummy variables for the variable 'TechSupport'.
ts = pd.get_dummies(telecom_df['TechSupport'], prefix='TechSupport')
ts1 = ts.drop(['TechSupport_No internet service'], 1)
```

```
# Adding the results to the master dataframe
telecom_df = pd.concat([telecom_df,ts1], axis=1)

# Creating dummy variables for the variable 'StreamingTV'.
st =pd.get_dummies(telecom_df['StreamingTV'], prefix='StreamingTV')
st1 = st.drop(['StreamingTV_No internet service'], 1)
# Adding the results to the master dataframe
telecom_df = pd.concat([telecom_df,st1], axis=1)

# Creating dummy variables for the variable 'StreamingMovies'.
sm = pd.get_dummies(telecom_df['StreamingMovies'], prefix='StreamingMovies')
sm1 = sm.drop(['StreamingMovies_No internet service'], 1)
# Adding the results to the master dataframe
telecom_df = pd.concat([telecom_df,sm1], axis=1)
```

[51]:
```
telecom_df.head()
```

[51]:
```
   customerID  tenure  PhoneService        Contract  PaperlessBilling  \
0  7590-VHVEG       1             0  Month-to-month                 1
1  5575-GNVDE      34             1        One year                 0
2  3668-QPYBK       2             1  Month-to-month                 1
3  7795-CFOCW      45             0        One year                 0
4  9237-HQITU       2             1  Month-to-month                 1

               PaymentMethod  MonthlyCharges  TotalCharges  Churn  gender  \
0           Electronic check           29.85         29.85      0  Female
1               Mailed check           56.95       1889.50      0    Male
2               Mailed check           53.85        108.15      1    Male
3  Bank transfer (automatic)           42.30       1840.75      0    Male
4           Electronic check           70.70        151.65      1  Female

   …  OnlineBackup_No  OnlineBackup_Yes  DeviceProtection_No  \
0  …                0                 1                    1
1  …                1                 0                    0
2  …                0                 1                    1
3  …                1                 0                    0
4  …                1                 0                    1

  DeviceProtection_Yes TechSupport_No TechSupport_Yes StreamingTV_No  \
0                    0              1               0              1
1                    1              1               0              1
2                    0              1               0              1
3                    1              0               1              1
4                    0              1               0              1

  StreamingTV_Yes StreamingMovies_No StreamingMovies_Yes
0               0                  1                   0
```

```
1              0                  1                   0
2              0                  1                   0
3              0                  1                   0
4              0                  1                   0

[5 rows x 43 columns]
```

[52]:

```
# We have created dummies for the below variables, so we can drop them
telecom_df = telecom_df.
 ↪drop(['Contract','PaymentMethod','gender','MultipleLines','InternetService',␣
 ↪'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
       'TechSupport', 'StreamingTV', 'StreamingMovies'], 1)
```

[53]:

```
telecom_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 7043 entries, 0 to 7042
Data columns (total 32 columns):
 #   Column                                 Non-Null Count  Dtype
---  ------                                 --------------  -----
 0   customerID                             7043 non-null   object
 1   tenure                                 7043 non-null   int64
 2   PhoneService                           7043 non-null   int64
 3   PaperlessBilling                       7043 non-null   int64
 4   MonthlyCharges                         7043 non-null   float64
 5   TotalCharges                           7032 non-null   float64
 6   Churn                                  7043 non-null   int64
 7   SeniorCitizen                          7043 non-null   int64
 8   Partner                                7043 non-null   int64
 9   Dependents                             7043 non-null   int64
 10  Contract_One year                      7043 non-null   uint8
 11  Contract_Two year                      7043 non-null   uint8
 12  PaymentMethod_Credit card (automatic)  7043 non-null   uint8
 13  PaymentMethod_Electronic check         7043 non-null   uint8
 14  PaymentMethod_Mailed check             7043 non-null   uint8
 15  gender_Male                            7043 non-null   uint8
 16  InternetService_Fiber optic            7043 non-null   uint8
 17  InternetService_No                     7043 non-null   uint8
 18  MultipleLines_No                       7043 non-null   uint8
 19  MultipleLines_Yes                      7043 non-null   uint8
 20  OnlineSecurity_No                      7043 non-null   uint8
 21  OnlineSecurity_Yes                     7043 non-null   uint8
 22  OnlineBackup_No                        7043 non-null   uint8
 23  OnlineBackup_Yes                       7043 non-null   uint8
 24  DeviceProtection_No                    7043 non-null   uint8
 25  DeviceProtection_Yes                   7043 non-null   uint8
 26  TechSupport_No                         7043 non-null   uint8
 27  TechSupport_Yes                       7043 non-null   uint8
 28  StreamingTV_No                        7043 non-null   uint8
 29  StreamingTV_Yes                       7043 non-null   uint8
 30  StreamingMovies_No                    7043 non-null   uint8
 31  StreamingMovies_Yes                   7043 non-null   uint8
dtypes: float64(2), int64(7), object(1), uint8(22)
memory usage: 1014.6+ KB
```

[54]:

```
# Checking for outliers in the continuous variables
num_telecom =␣
 ↪telecom_df[['tenure','MonthlyCharges','SeniorCitizen','TotalCharges']]
```

[55]:

```
# Checking outliers at 25%, 50%, 75%, 90%, 95% and 99%
num_telecom.describe(percentiles=[.25, .5, .75, .90, .95, .99])
```

[55]:

```
            tenure  MonthlyCharges  SeniorCitizen  TotalCharges
count  7043.000000     7043.000000    7043.000000   7032.000000
mean     32.371149       64.761692       0.162147   2283.300441
std      24.559481       30.090047       0.368612   2266.771362
min       0.000000       18.250000       0.000000     18.800000
25%       9.000000       35.500000       0.000000    401.450000
50%      29.000000       70.350000       0.000000   1397.475000
75%      55.000000       89.850000       0.000000   3794.737500
90%      69.000000      102.600000       1.000000   5976.640000
95%      72.000000      107.400000       1.000000   6923.590000
99%      72.000000      114.729000       1.000000   8039.883000
max      72.000000      118.750000       1.000000   8684.800000
```

[56]:

```
# Adding up the missing values (column-wise)
telecom_df.isnull().sum()
```

[56]:

```
customerID                               0
tenure                                   0
PhoneService                             0
PaperlessBilling                         0
MonthlyCharges                           0
TotalCharges                            11
Churn                                    0
SeniorCitizen                            0
Partner                                  0
Dependents                               0
Contract_One year                        0
Contract_Two year                        0
PaymentMethod_Credit card (automatic)    0
PaymentMethod_Electronic check           0
PaymentMethod_Mailed check               0
gender_Male                              0
```

```
InternetService_Fiber optic               0
InternetService_No                        0
MultipleLines_No                          0
MultipleLines_Yes                         0
OnlineSecurity_No                         0
OnlineSecurity_Yes                        0
OnlineBackup_No                           0
OnlineBackup_Yes                          0
DeviceProtection_No                       0
DeviceProtection_Yes                      0
TechSupport_No                            0
TechSupport_Yes                           0
StreamingTV_No                            0
StreamingTV_Yes                           0
StreamingMovies_No                        0
StreamingMovies_Yes                       0
dtype: int64
```

[57]:

```
# Checking the percentage of missing values
round(100*(telecom_df.isnull().sum()/len(telecom_df.index)), 2)
```

[57]:

```
customerID                               0.00
tenure                                   0.00
PhoneService                             0.00
PaperlessBilling                         0.00
MonthlyCharges                           0.00
TotalCharges                             0.16
Churn                                    0.00
SeniorCitizen                            0.00
Partner                                  0.00
Dependents                               0.00
Contract_One year                        0.00
Contract_Two year                        0.00
PaymentMethod_Credit card (automatic)    0.00
PaymentMethod_Electronic check           0.00
PaymentMethod_Mailed check               0.00
gender_Male                              0.00
InternetService_Fiber optic              0.00
InternetService_No                       0.00
MultipleLines_No                         0.00
MultipleLines_Yes                        0.00
OnlineSecurity_No                        0.00
OnlineSecurity_Yes                       0.00
OnlineBackup_No                          0.00
OnlineBackup_Yes                         0.00
DeviceProtection_No                      0.00
DeviceProtection_Yes                     0.00
```

```
TechSupport_No                           0.00
TechSupport_Yes                          0.00
StreamingTV_No                           0.00
StreamingTV_Yes                          0.00
StreamingMovies_No                       0.00
StreamingMovies_Yes                      0.00
dtype: float64
```

[58]:

```
# Removing NaN TotalCharges rows
telecom_df = telecom_df[~np.isnan(telecom_df['TotalCharges'])]
```

[59]:

```
# Checking percentage of missing values after removing the missing values
round(100*(telecom_df.isnull().sum()/len(telecom_df.index)), 2)
```

[59]:

```
customerID                               0.0
tenure                                   0.0
PhoneService                             0.0
PaperlessBilling                         0.0
MonthlyCharges                           0.0
TotalCharges                             0.0
Churn                                    0.0
SeniorCitizen                            0.0
Partner                                  0.0
Dependents                               0.0
Contract_One year                        0.0
Contract_Two year                        0.0
PaymentMethod_Credit card (automatic)    0.0
PaymentMethod_Electronic check           0.0
PaymentMethod_Mailed check               0.0
gender_Male                              0.0
InternetService_Fiber optic              0.0
InternetService_No                       0.0
MultipleLines_No                         0.0
MultipleLines_Yes                        0.0
OnlineSecurity_No                        0.0
OnlineSecurity_Yes                       0.0
OnlineBackup_No                          0.0
OnlineBackup_Yes                         0.0
DeviceProtection_No                      0.0
DeviceProtection_Yes                     0.0
TechSupport_No                           0.0
TechSupport_Yes                          0.0
StreamingTV_No                           0.0
StreamingTV_Yes                          0.0
StreamingMovies_No                       0.0
StreamingMovies_Yes                      0.0
dtype: float64
```

[60]:

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

[62]:

```
# Putting feature variable to X
X = telecom_df.drop(['Churn','customerID'], axis=1)

X.head()
```

[62]:

```
   tenure  PhoneService  PaperlessBilling  MonthlyCharges  TotalCharges  \
0       1             0                 1           29.85         29.85
1      34             1                 0           56.95       1889.50
2       2             1                 1           53.85        108.15
3      45             0                 0           42.30       1840.75
4       2             1                 1           70.70        151.65

   SeniorCitizen  Partner  Dependents  Contract_One year  Contract_Two year  \
0              0        1           0                  0                  0
1              0        0           0                  1                  0
2              0        0           0                  0                  0
3              0        0           0                  1                  0
4              0        0           0                  0                  0

   …  OnlineBackup_No  OnlineBackup_Yes  DeviceProtection_No  \
0  …                0                 1                    1
1  …                1                 0                    0
2  …                0                 1                    1
3  …                1                 0                    0
4  …                1                 0                    1

   DeviceProtection_Yes  TechSupport_No  TechSupport_Yes  StreamingTV_No  \
0                     0               1                0               1
1                     1               1                0               1
2                     0               1                0               1
3                     1               0                1               1
4                     0               1                0               1

   StreamingTV_Yes  StreamingMovies_No  StreamingMovies_Yes
0                0                   1                    0
1                0                   1                    0
2                0                   1                    0
3                0                   1                    0
4                0                   1                    0

[5 rows x 30 columns]
```

[63]:

```
# Putting response variable to y
y = telecom_df['Churn']
```

```
y.head()
```

```
[63]: 0    0
      1    0
      2    1
      3    0
      4    1
      Name: Churn, dtype: int64
```

```
[64]: # Splitting the data into train and test
      X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7,␣
       ↪test_size=0.3, random_state=100)
```

```
[65]:     from sklearn.preprocessing import StandardScaler
```

```
[66]: scaler = StandardScaler()

      X_train[['tenure','MonthlyCharges','TotalCharges']] = scaler.
       ↪fit_transform(X_train[['tenure','MonthlyCharges','TotalCharges']])

      X_train.head()
```

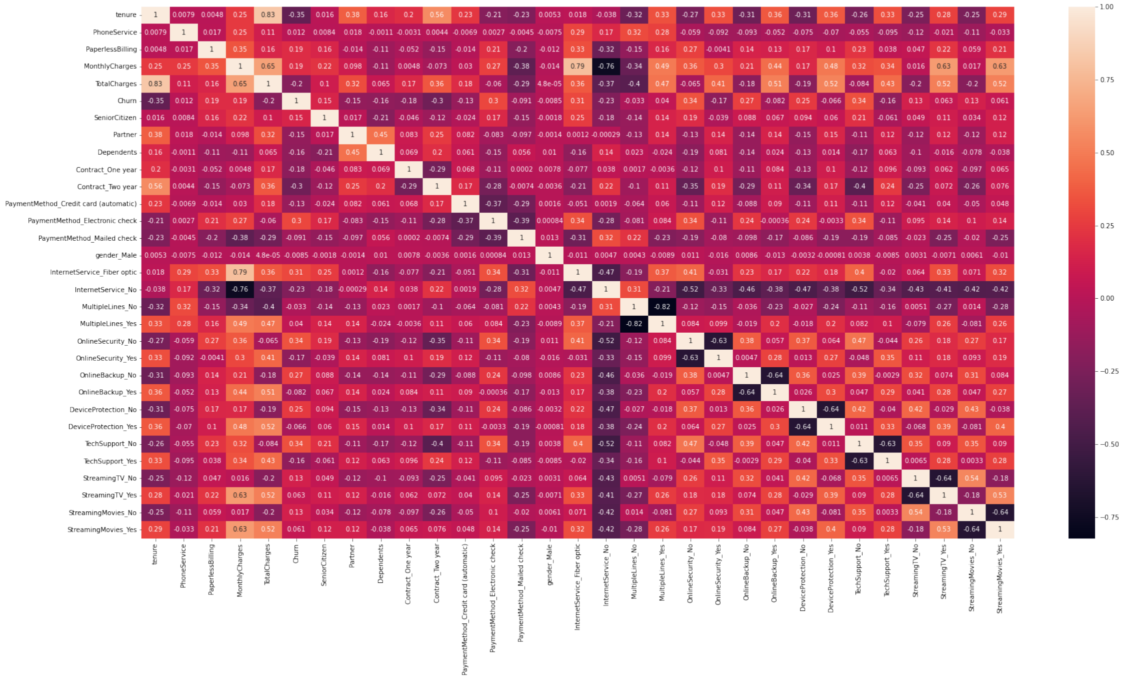
```
[66]:         tenure  PhoneService  PaperlessBilling  MonthlyCharges  TotalCharges  \
      879   0.019693             1                 1       -0.338074     -0.276449
      5790  0.305384             0                 1       -0.464443     -0.112702
      6498 -1.286319             1                 1        0.581425     -0.974430
      880  -0.919003             1                 1        1.505913     -0.550676
      2784 -1.163880             1                 1        1.106854     -0.835971

            SeniorCitizen  Partner  Dependents  Contract_One year  \
      879               0        0           0                  0
      5790              0        1           1                  0
      6498              0        0           0                  0
      880               0        0           0                  0
      2784              0        0           1                  0

            Contract_Two year  …  OnlineBackup_No  OnlineBackup_Yes  \
      879                   0  …                0                 1
      5790                  0  …                0                 1
      6498                  0  …                0                 1
      880                   0  …                0                 1
      2784                  0  …                1                 0

            DeviceProtection_No  DeviceProtection_Yes  TechSupport_No  \
      879                     1                     0               1
      5790                    1                     0               1
      6498                    0                     1               1
```

```
880                     0                     1               0
2784                    0                     1               0

      TechSupport_Yes  StreamingTV_No  StreamingTV_Yes  StreamingMovies_No  \
879                 0               1                0                   1
5790                0               0                1                   0
6498                0               1                0                   1
880                 1               0                1                   0
2784                1               0                1                   0

      StreamingMovies_Yes
879                     0
5790                    1
6498                    0
880                     1
2784                    1

[5 rows x 30 columns]
```

[67]:
```
### Checking the Churn Rate
churn = (sum(telecom_df['Churn'])/len(telecom_df['Churn'].index))*100
churn
```

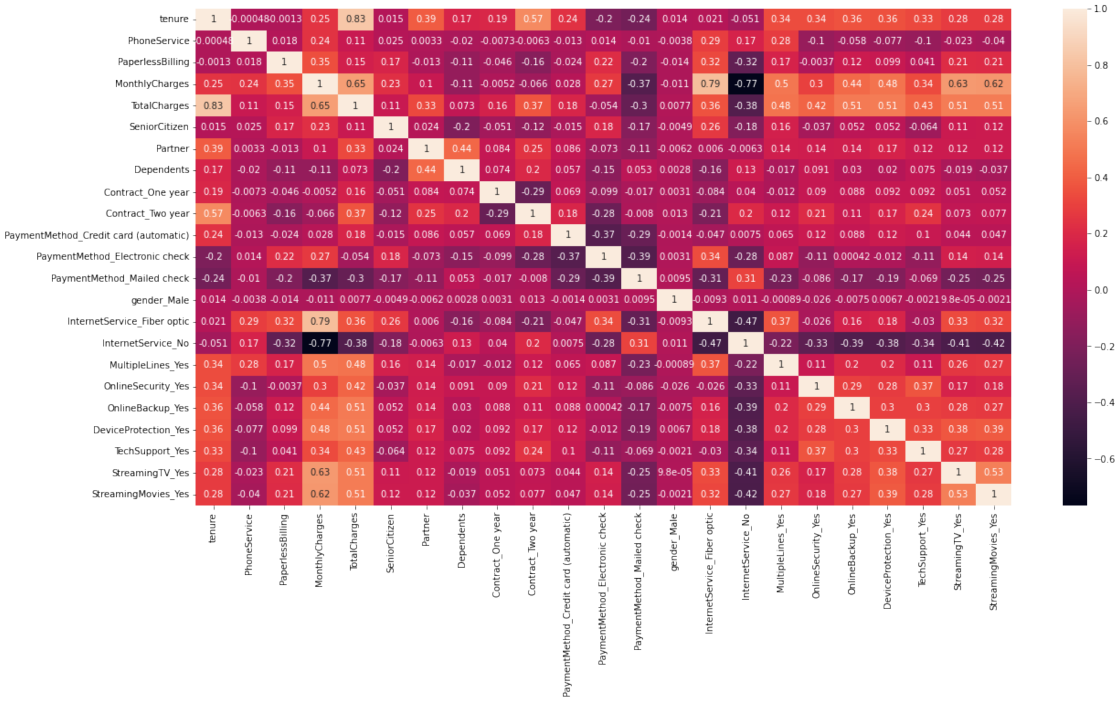
[67]:
```
26.578498293515356
```

[68]:
```
# Let's see the correlation matrix
plt.figure(figsize = (30,15))        # Size of the figure
sns.heatmap(telecom_df.corr(),annot = True)
plt.show()
```



[69]:
```
X_test = X_test.
↪drop(['MultipleLines_No','OnlineSecurity_No','OnlineBackup_No','DeviceProtection_No','TechS
                       'StreamingTV_No','StreamingMovies_No'], 1)
X_train = X_train.
↪drop(['MultipleLines_No','OnlineSecurity_No','OnlineBackup_No','DeviceProtection_No','TechS
                         'StreamingTV_No','StreamingMovies_No'], 1)
```

[70]:
```
plt.figure(figsize = (20,10))
sns.heatmap(X_train.corr(),annot = True)
plt.show()
```



[71]:
```
jovian.commit
```

[71]:
```
<function jovian.utils.commit.commit(message=None, files=[], outputs=[],
environment=None, privacy='auto', filename=None, project=None, new_project=None,
git_commit=False, git_message='auto', require_write_access=False, **kwargs)>
```

[ ]:
```

```