data_analysis1-Copy1

April 1, 2024

```
In [2]: # [U+8B80] [U+5165] [U+7FD2] [U+6163] [U+7684] [U+8CC7] [U+6599] [U+79D1] [U+5B78] [U+5957] [U+4
        import numpy as np
        import pandas as pd
        import os
```

1 1.[U+6570] [U+636E] [U+6982] [U+89C8]

```
In [3]: import matplotlib.pyplot as plt
        import seaborn as sns


        df = pd.read_csv('loss.csv')
        df.head()
```

```
Out[3]:    customerID  gender  SeniorCitizen Partner Dependents  tenure PhoneService  \
        0  7590-VHVEG  Female              0     Yes         No       1           No
        1  5575-GNVDE    Male              0      No         No      34          Yes
        2  3668-QPYBK    Male              0      No         No       2          Yes
        3  7795-CFOCW    Male              0      No         No      45           No
        4  9237-HQITU  Female              0      No         No       2          Yes

              MultipleLines InternetService OnlineSecurity  ... DeviceProtection  \
        0  No phone service             DSL             No  ...               No
        1                No             DSL            Yes  ...              Yes
        2                No             DSL            Yes  ...               No
        3  No phone service             DSL            Yes  ...              Yes
        4                No     Fiber optic             No  ...               No

          TechSupport StreamingTV StreamingMovies        Contract PaperlessBilling  \
        0          No          No              No  Month-to-month              Yes
        1          No          No              No        One year               No
        2          No          No              No  Month-to-month              Yes
        3         Yes          No              No        One year               No
        4          No          No              No  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

[5 rows x 21 columns]
```

```
In [4]: df.shape[0]
```

```
Out[4]: 7043
```

```
In [5]: # View data variables, total number, missing data, variable measurement(dimension)
        def data_overview():
            print("Rows :  " , df.shape[0])
            print("Columns:  " , df.shape[1] )
            print('Missing Value number : ' , df.isnull().sum().values.sum()) #isnull.sum() will
            print('\nUnique values' , df.nunique())
        data_overview()
```

```
Rows :   7043
Columns:   21
Missing Value number :  0

Unique values customerID          7043
gender                 2
SeniorCitizen          2
Partner                2
Dependents             2
tenure                73
PhoneService           2
MultipleLines          3
InternetService        3
OnlineSecurity         3
OnlineBackup           3
DeviceProtection       3
TechSupport            3
StreamingTV            3
StreamingMovies        3
Contract               3
PaperlessBilling       2
PaymentMethod          4
MonthlyCharges      1585
TotalCharges        6531
Churn                  2
dtype: int64
```

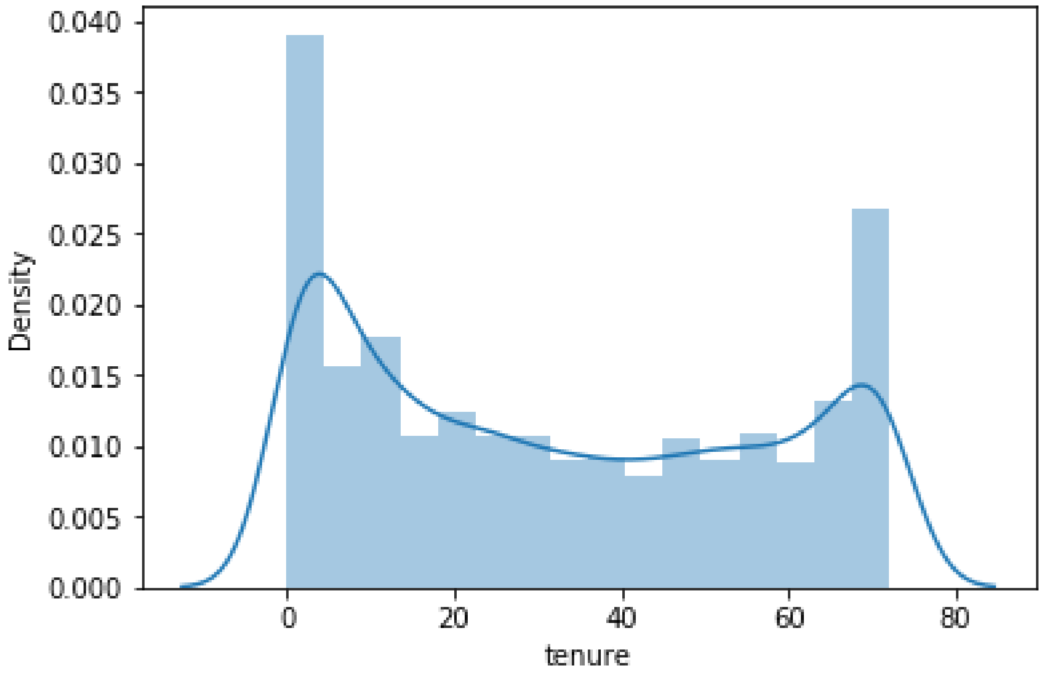
```
In [6]: df.isnull().sum()
```

```
Out[6]: customerID          0
        gender              0
        SeniorCitizen       0
        Partner             0
        Dependents          0
        tenure              0
        PhoneService        0
        MultipleLines       0
        InternetService     0
        OnlineSecurity      0
        OnlineBackup        0
        DeviceProtection    0
        TechSupport         0
        StreamingTV         0
        StreamingMovies     0
        Contract            0
        PaperlessBilling    0
        PaymentMethod       0
        MonthlyCharges      0
        TotalCharges        0
        Churn               0
        dtype: int64
```

```
In [7]:
        sns.distplot(df.tenure)
```

```
  y = y[:, np.newaxis]
```

```
Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x28093dfc748>
```



```
In [ ]: sns.distplot(df.MonthlyCharges)
```

```
In [ ]: df.TotalCharges
```

```
In [ ]: sns.distplot(df.TotalCharges)
```

2 2. [U+6570] [U+636E] [U+9884] [U+5904] [U+7406]

2.0.1 2.1 [U+68C0] [U+67E5] [U+5F02] [U+5E38] [U+503C]

2.0.2 [U+7B2C] [U+4E00] [U+56DB] [U+5206] [U+4F4D] [U+6570]-1.5*IQR* < *[U+6B63] [U+5E38] [U+503C]* < *[U+7B2C] [U+4E09] [U+56DB] [U+5206] [U+4F4D] [U+6570]*+1.5IQR

2.0.3 -1,1,1,1,2,2,2,3,3,3,4,4,4

```
In [ ]: #    [U+4F7F] [U+7528] [U+4E86] [U+56DB] [U+5206] [U+4F4D] [U+6570] [U+8303] [U+56F4] [U+FF08] IQ
        #    IQR [U+662F] [U+7B2C] [U+4E09] [U+56DB] [U+5206] [U+4F4D] [U+6570] [U+FF08] 75% [U+5206] [U+4
        #    [U+5728] [U+8FD9] [U+4E2A] [U+4EE3] [U+7801] [U+4E2D] [U+FF0C] [U+4EFB] [U+4F55] [U+5C0F] [U-
        #
        #    [U+8FD9] [U+662F] [U+4E00] [U+79CD] [U+5E38] [U+7528] [U+7684] [U+5F02] [U+5E38] [U+503C] [U-
        def smells(df):
            summary = df.describe(include='all')
            for column in summary.columns:
                if df[column].dtype in ['float64', 'int64']:
                    IQR = summary.at['75%', column] - summary.at['25%', column]
                    lower_bound = summary.at['25%', column] - 1.5 * IQR
                    upper_bound = summary.at['75%', column] + 1.5 * IQR
                    if df[column].lt(lower_bound).any() or df[column].gt(upper_bound).any():
                        print(f"Column {column} may have outliers.")
                elif df[column].dtype == 'object':
                    if df[column].str.len().max() > 255:
                        print(f"Column {column} may have strings that are too long.")
```

```
In [ ]: smells(df)
```

2.1 [U+7ED3] [U+8BBA] [U+FF1A] **SeniorCitizen** [U+5B58] [U+5728] [U+5F02] [U+5E38] [U+503C] [U+FF0C
[U+6216] **1** [U+FF0C] [U+56E0] [U+6B64] [U+53EF] [U+80FD] [U+662F] [U+8BA1] [U+7B97] [U+8BEF] [U-

2.2 **2.2** [U+5904] [U+7406] [U+7A7A] [U+5B57] [U+7B26] [U+4E32]

```
In [ ]: df.isnull().sum()
```

```
In [ ]: df= df.replace(' ',np.nan)
```

```
In [ ]: df.isnull().sum()
```

2.3 [U+89C2] [U+5BDF] [U+5F02] [U+5E38] [U+503C] [U+6240] [U+5728] [U+7684] [U+8D26] [U+6237]

```
In [ ]: # [U+53D1] [U+73B0] 11 [U+4E2A] [U+7A7A] [U+503C]

        print('LOST[U+FF1A]')
        print(df.TotalCharges.isnull().sum())
```

```
In [ ]:
        df[df.TotalCharges.isnull()]
```

```
In [ ]:
        plt.figure(figsize = (12,4))
        sns.distplot(df.TotalCharges.notnull().astype(float))
```

```
In [ ]: sns.distplot(df.TotalCharges)
```

2.4 **2.3** [U+7F3A] [U+5931] [U+503C] [U+5904] [U+7406]

```
In [ ]: print('[U+6E05] [U+9664] [U+7F3A] [U+5931] [U+503C] ')
        df = df[df.TotalCharges.notnull()]
        df = df.reset_index()
        #[U+518D] [U+8F49] [U+63DB] [U+4E00] [U+6B21] [U+578B] [U+614B]
        df.TotalCharges = df.TotalCharges.astype(float)
```

2.5 [U+5B57] [U+7B26] [U+4E32] [U+8F6C] [U+6570] [U+5B57]

```
In [467]:
          df = df.replace({'Yes':1 , 'No' :0})
          df.head()
```

```
Out[467]:    index  customerID  gender  SeniorCitizen  Partner  Dependents  tenure  \
          0      0  7590-VHVEG  Female              0        1           0       1
          1      1  5575-GNVDE    Male              0        0           0      34
          2      2  3668-QPYBK    Male              0        0           0       2
          3      3  7795-CFOCW    Male              0        0           0      45
          4      4  9237-HQITU  Female              0        0           0       2

             PhoneService     MultipleLines InternetService  ... DeviceProtection  \
          0             0  No phone service             DSL  ...                0
          1             1                 0             DSL  ...                1
          2             1                 0             DSL  ...                0
          3             0  No phone service             DSL  ...                1
          4             1                 0     Fiber optic  ...                0

            TechSupport StreamingTV StreamingMovies        Contract PaperlessBilling  \
          0           0           0               0  Month-to-month                1
          1           0           0               0        One year                0
          2           0           0               0  Month-to-month                1
          3           1           0               0        One year                0
          4           0           0               0  Month-to-month                1

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

          [5 rows x 22 columns]
```

```
In [468]:
          df = df.replace({'No phone service':0})
          df.head()
```

```
Out[468]:    index  customerID  gender  SeniorCitizen  Partner  Dependents  tenure  \
          0      0  7590-VHVEG  Female              0        1           0       1
          1      1  5575-GNVDE    Male              0        0           0      34
          2      2  3668-QPYBK    Male              0        0           0       2
          3      3  7795-CFOCW    Male              0        0           0      45
          4      4  9237-HQITU  Female              0        0           0       2

             PhoneService  MultipleLines InternetService  ... DeviceProtection  \
          0             0              0             DSL  ...                0
          1             1              0             DSL  ...                1
          2             1              0             DSL  ...                0
          3             0              0             DSL  ...                1
          4             1              0     Fiber optic  ...                0
```

```
  TechSupport StreamingTV StreamingMovies        Contract PaperlessBilling  \
0           0           0               0  Month-to-month                1
1           0           0               0        One year                0
2           0           0               0  Month-to-month                1
3           1           0               0        One year                0
4           0           0               0  Month-to-month                1

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

[5 rows x 22 columns]
```

2.6 2.4 [U+8FDE][U+7EED][U+578B][U+53D8][U+91CF][U+5904][U+7406]

```
In [469]: #[U+9023][U+7E8C][U+578B][U+8B8A][U+91CF][U+6211][U+5011][U+53EF][U+4EE5][U+8F49][U+6
          print(df.tenure.describe())
          #[U+6211][U+5011][U+767C][U+73FE][U+6578][U+64DA][U+9084][U+883B][U+5E73][U+8861][U+7
          #[U+64B0][U+5BEB][U+4E00][U+500B][U+5C07]tenure[U+8F49][U+8B8A][U+70BA][U+96E2][U+656
          def tenure_to_bins(series):
              labels = [1,2,3,4,5]
              bins = pd.cut(series , bins = 5 , labels = labels)
              return bins
          temp_tenure = df.tenure
          df['tenure_group'] = tenure_to_bins(temp_tenure)
          df.head()
```

```
count    7032.000000
mean       32.421786
std        24.545260
min         1.000000
25%         9.000000
50%        29.000000
75%        55.000000
max        72.000000
Name: tenure, dtype: float64
```

```
Out[469]:    index  customerID  gender  SeniorCitizen  Partner  Dependents  tenure  \
          0      0  7590-VHVEG  Female              0        1           0       1
          1      1  5575-GNVDE    Male              0        0           0      34
          2      2  3668-QPYBK    Male              0        0           0       2
          3      3  7795-CFOCW    Male              0        0           0      45
          4      4  9237-HQITU  Female              0        0           0       2
```

```
  PhoneService  MultipleLines InternetService  ... TechSupport StreamingTV  \
0            0              0             DSL  ...           0           0
1            1              0             DSL  ...           0           0
2            1              0             DSL  ...           0           0
3            0              0             DSL  ...           1           0
4            1              0     Fiber optic  ...           0           0

  StreamingMovies        Contract PaperlessBilling              PaymentMethod  \
0               0  Month-to-month                1           Electronic check
1               0        One year                0               Mailed check
2               0  Month-to-month                1               Mailed check
3               0        One year                0  Bank transfer (automatic)
4               0  Month-to-month                1           Electronic check

  MonthlyCharges  TotalCharges Churn  tenure_group
0          29.85         29.85     0             1
1          56.95       1889.50     0             3
2          53.85        108.15     1             1
3          42.30       1840.75     0             4
4          70.70        151.65     1             1

[5 rows x 23 columns]
```

```
In [470]: # [U+5C07] [U+5169] [U+985E] [U+6578] [U+64DA] [U+5206] [U+958B]
          churn = df[df.Churn == 1]
          not_churn = df[df.Churn == 0]
          # [U+5C07] [U+985E] [U+5225] [U+8B8A] [U+6578] [U+8207] [U+9023] [U+7E8C] [U+8B8A] [U+6578] [U+
          Id_col = ['customerID']
          target_col = ['Churn']
          cat_cols = df.nunique()[df.nunique() < 6].keys().tolist() #[U+53D6] [U+51FA]Series.inde
          cat_cols = [col for col in cat_cols if col not in target_col]
          num_cols = [x for x in df.columns if x not in Id_col + target_col + cat_cols]
```

3 3.EDA(exploratory data analysis [U+FF09]

```
In [471]: # [U+5148] [U+5C0E] [U+5165] [U+76F8] [U+95DC] [U+5957] [U+4EF6]
          import plotly.offline as py
          py.init_notebook_mode(connected=True) #[U+70BA] [U+4E86] [U+80FD] [U+5728] [U+672C] [U+573
          import plotly.graph_objs as go
          import plotly.tools as tls
          import plotly.figure_factory as ff
```

3.1 3.1 [U+6982] [U+89C8] [U+76EE] [U+6807] [U+53D8] [U+91CF]

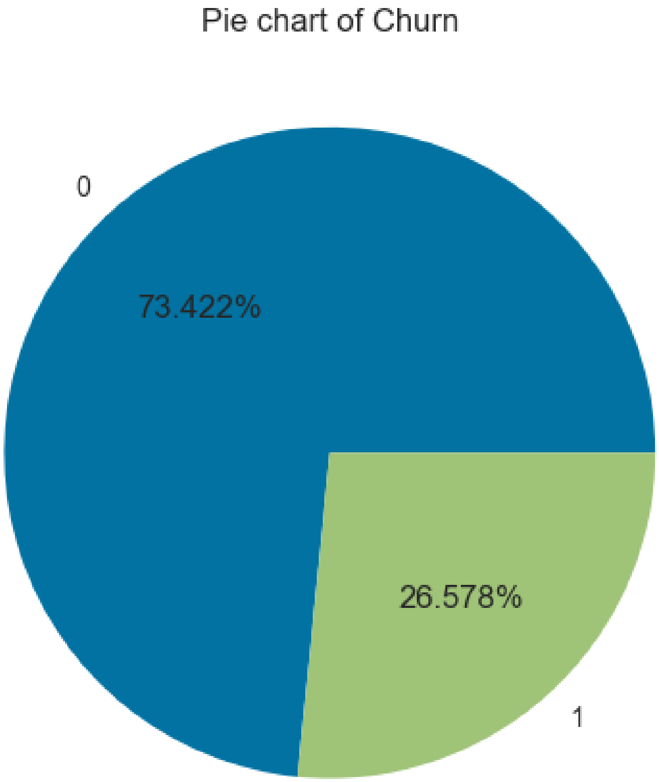
[U+4E0B] [U+9762] [U+FF0C] [U+4F7F] [U+7528] **plt** [U+5E93] [U+FF08] [U+4E00] [U+4E2A] [U+7528] [U+4E8E] [U+5

```
In [472]: import matplotlib.pyplot as plt
          def plot_pie(df, column):
```

```
    # [U+8BA1][U+7B97][U+5404][U+4E2A][U+503C][U+7684][U+6570][U+91CF]
    counts = df[column].value_counts()
    print(counts)
    # [U+7ED8][U+5236][U+997C][U+56FE]
    plt.pie(counts, labels=counts.index, autopct='%1.3f%%')
    plt.title(f'Pie chart of {column}')
    plt.show()
```

```
In [473]: plot_pie(df, 'Churn')
```

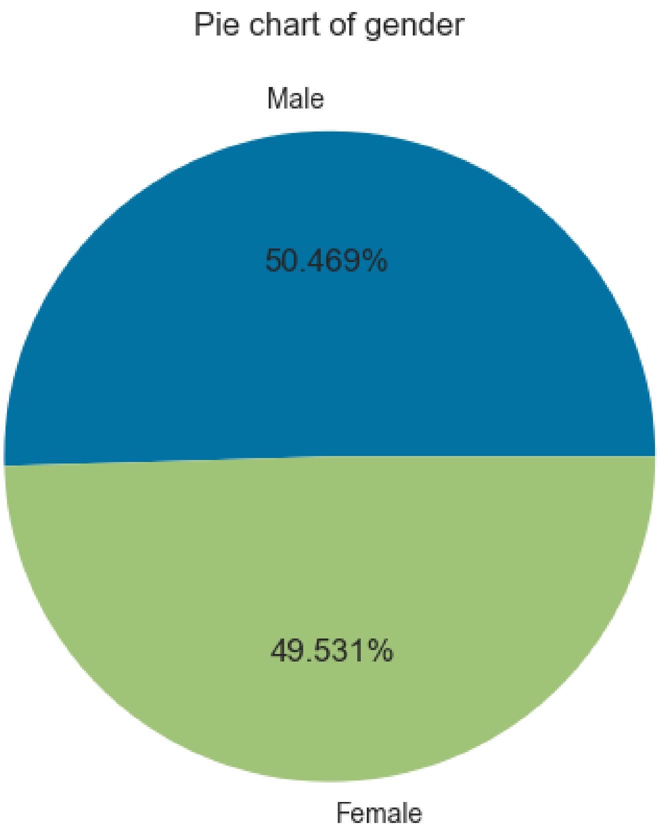
```
Churn
0    5163
1    1869
Name: count, dtype: int64
```



```
In [474]: plot_pie(df, 'gender')
```

```
gender
Male      3549
```

```
Female    3483
Name: count, dtype: int64
```



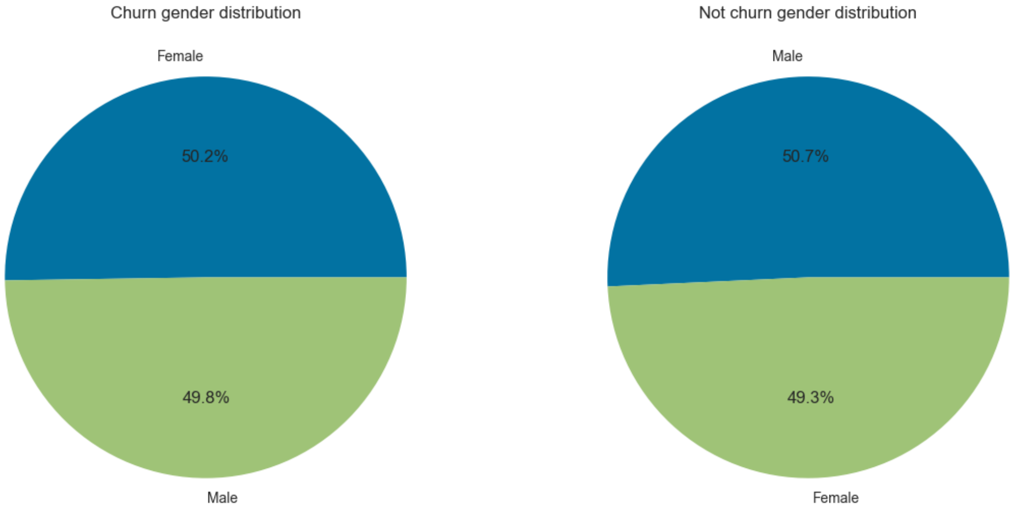
3.1.1 [U+997C] [U+56FE]

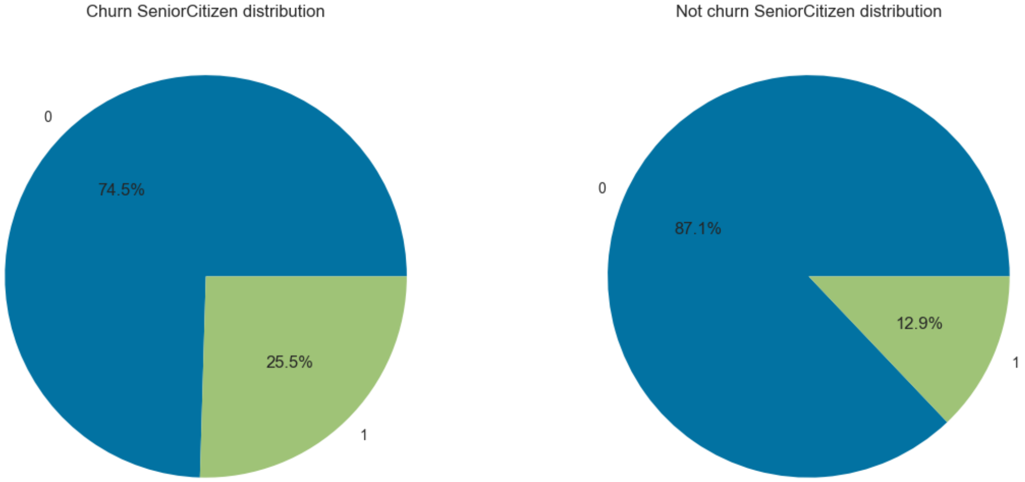
def pie_draw(df, column): plt.figure(figsize=(6,6)) df.groupby('Churn')[column].value_counts(normalize=True).u
subplots=True, autopct='%1.1f%%') plt.title(column + " distribution in Churn") plt.show()

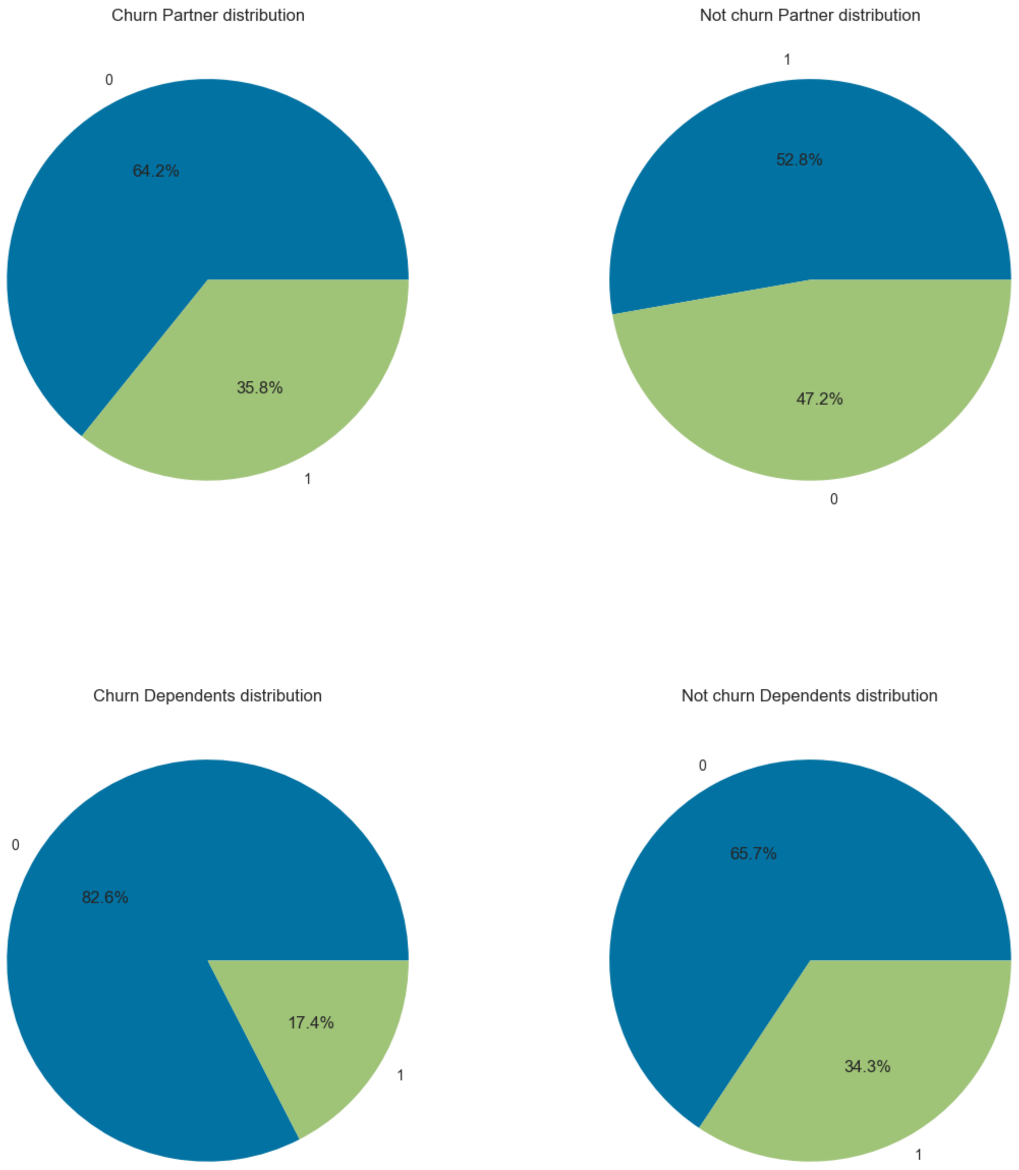
```
In [479]: import matplotlib.pyplot as plt

          def draw_pie(df, column):
              # [U+5206] [U+522B] [U+83B7] [U+53D6] [U+6D41] [U+5931] [U+5BA2] [U+6237] [U+548C] [U+975E
              churn_df = df[df['Churn'] == 1]
              not_churn_df = df[df['Churn'] == 0]
              # [U+521B] [U+5EFA] [U+4E00] [U+4E2A] [U+65B0] [U+7684] [U+56FE] [U+5F62] [U+FF0C] [U+5305
              fig, axs = plt.subplots(1, 2, figsize=(14, 7))
              # [U+5728] [U+7B2C] [U+4E00] [U+4E2A] [U+5B50] [U+56FE] [U+4E2D] [U+7ED8] [U+5236] [U+6D4
              axs[0].pie(churn_df[column].value_counts(), labels=churn_df[column].value_counts()
              axs[0].set_title('Churn ' + column + ' distribution')
```

```
    # [U+5728] [U+7B2C] [U+4E8C] [U+4E2A] [U+5B50] [U+56FE] [U+4E2D] [U+7ED8] [U+5236] [U+975E
    axs[1].pie(not_churn_df[column].value_counts(), labels=not_churn_df[column].value_
    axs[1].set_title('Not churn ' + column + ' distribution')

    # [U+663E] [U+793A] [U+56FE] [U+5F62]
    plt.show()
```

```
In [480]: for col in cat_cols:
              draw_pie(df,col)
```





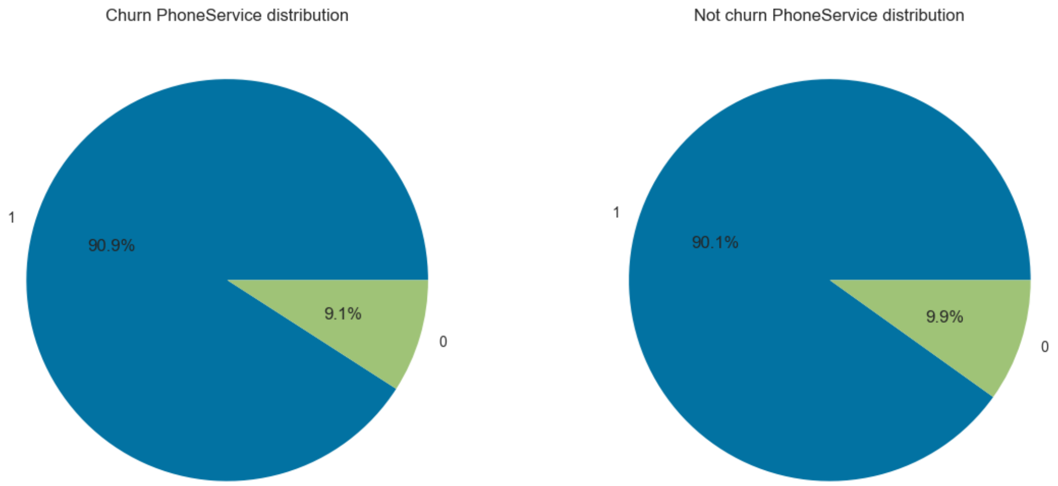


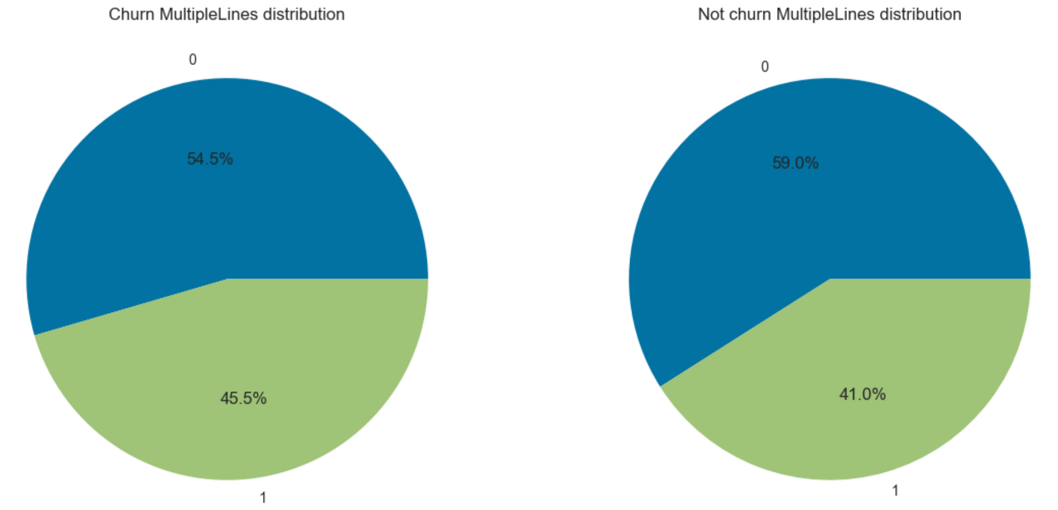
Churn Partner distribution

Not churn Partner distribution

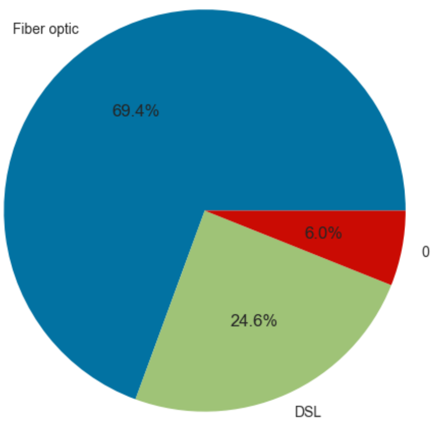
Churn Dependents distribution

Not churn Dependents distribution

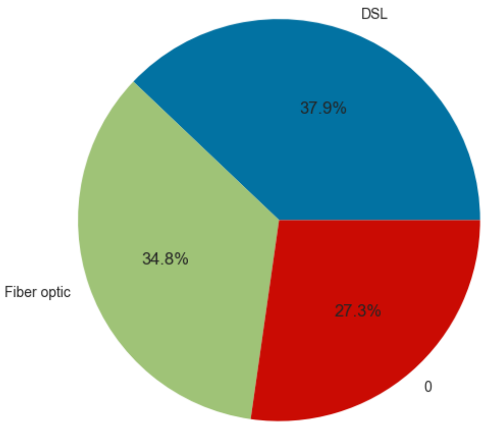

Churn PhoneService distribution
1
90.9%
9.1%
0
Not churn PhoneService distribution
1
90.1%
9.9%
0


Churn MultipleLines distribution
0
54.5%
45.5%
1
Not churn MultipleLines distribution
0
59.0%
41.0%
1

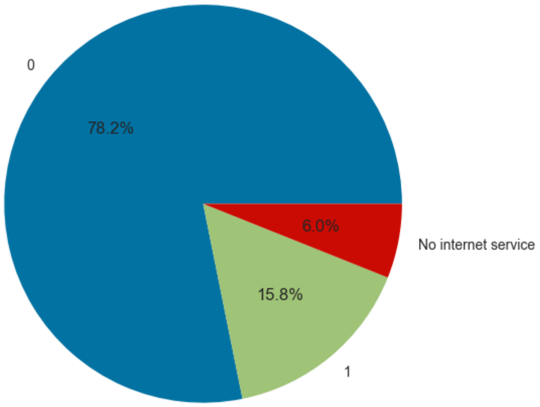
Churn InternetService distribution



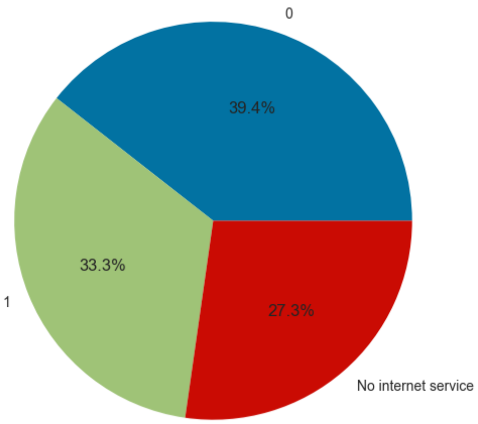
Not churn InternetService distribution

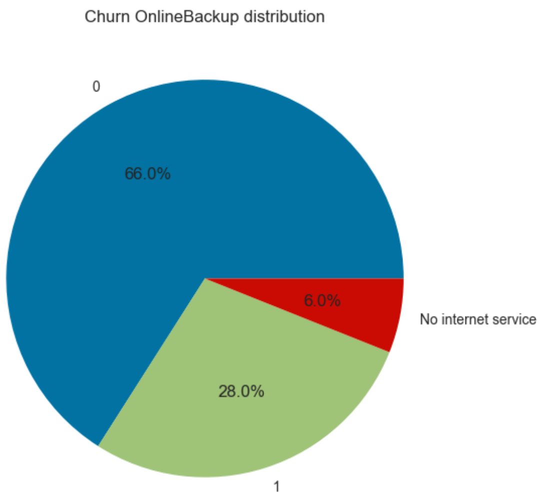


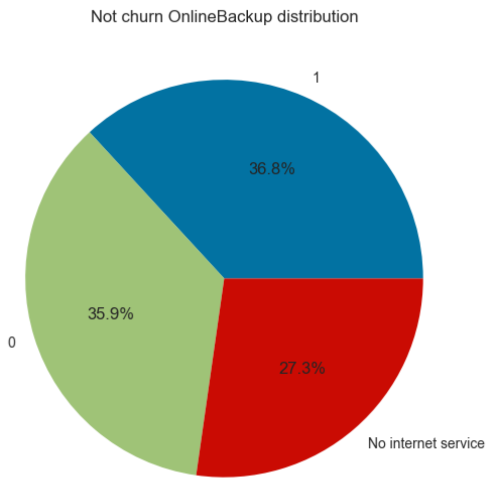
Churn OnlineSecurity distribution

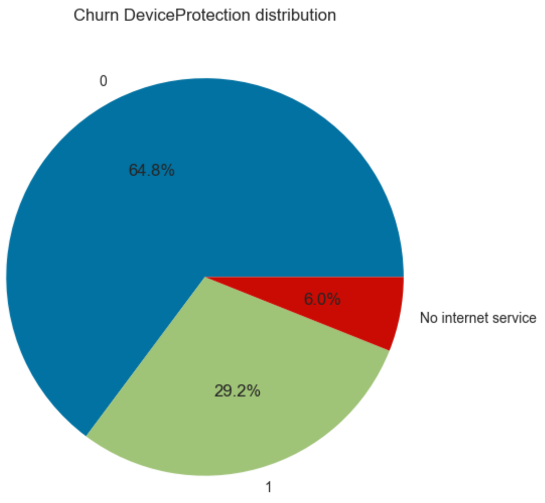


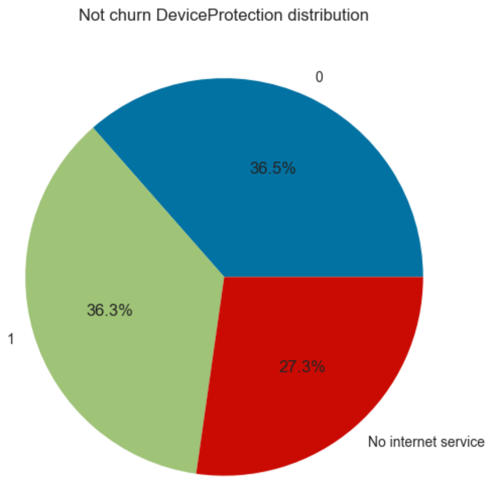
Not churn OnlineSecurity distribution

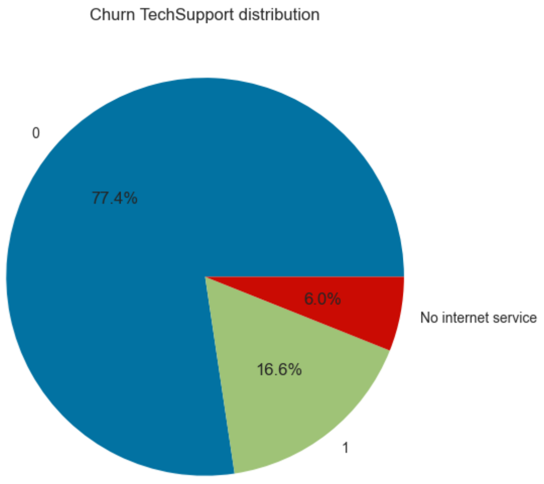



Churn OnlineBackup distribution
0
66.0%
6.0%
No internet service
28.0%
1

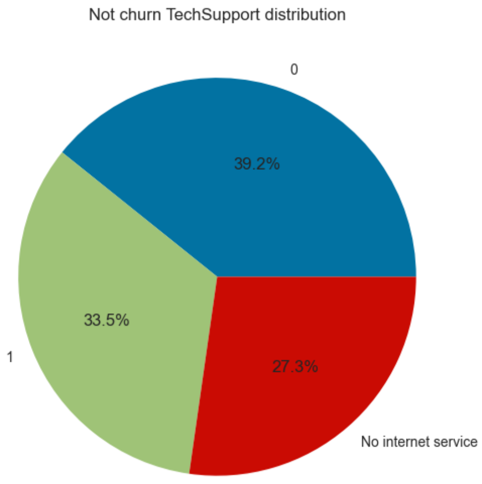

Not churn OnlineBackup distribution
1
36.8%
35.9%
0
27.3%
No internet service


Churn DeviceProtection distribution
0
64.8%
6.0%
No internet service
29.2%
1

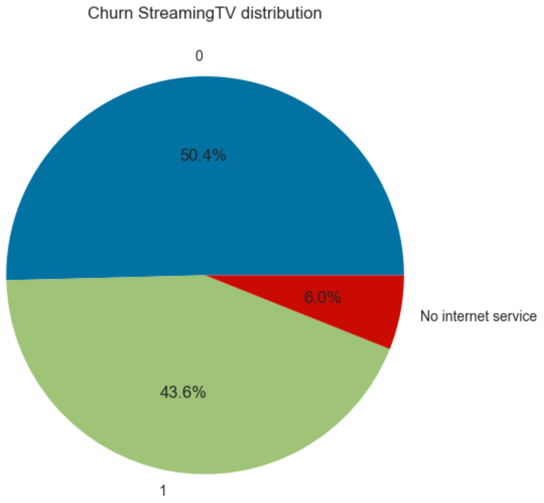

Not churn DeviceProtection distribution
0
36.5%
36.3%
1
27.3%
No internet service



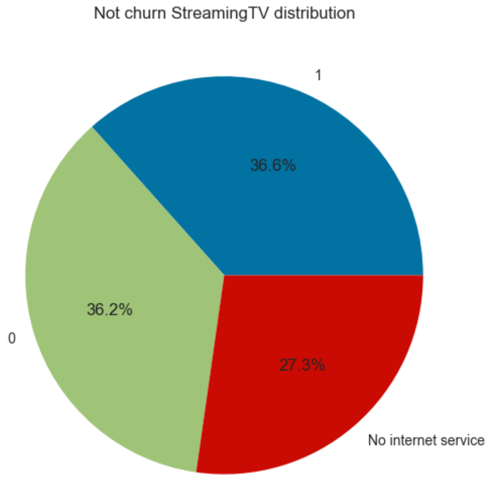
Churn TechSupport distribution



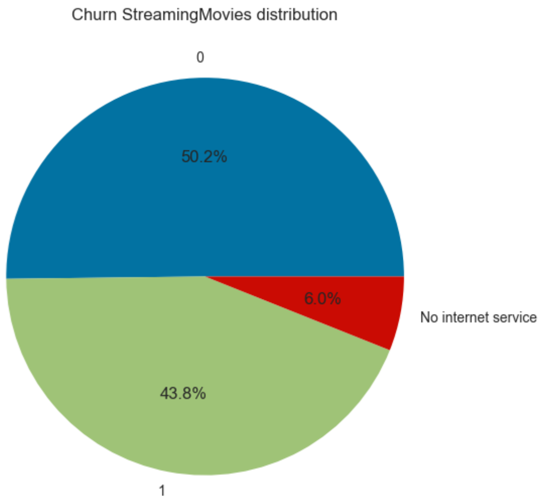
Not churn TechSupport distribution



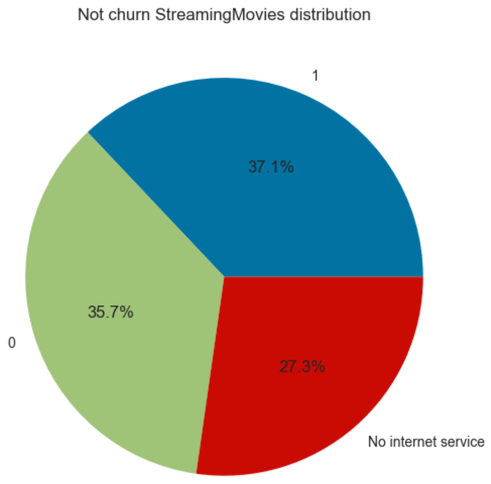
Churn StreamingTV distribution



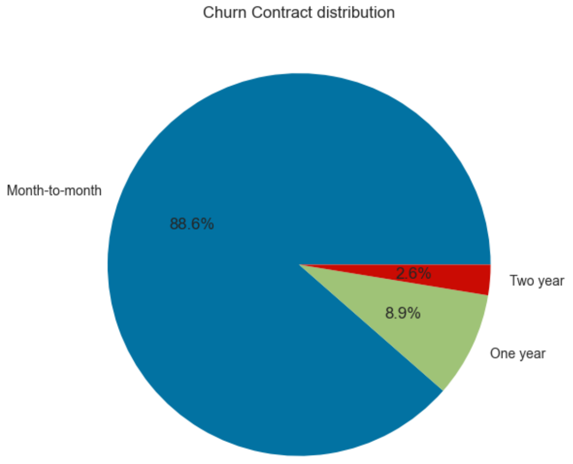
Not churn StreamingTV distribution



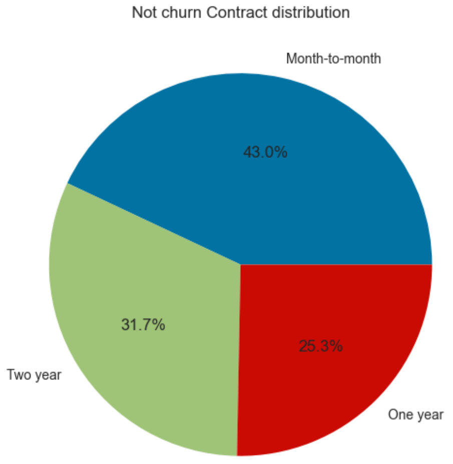
Churn StreamingMovies distribution



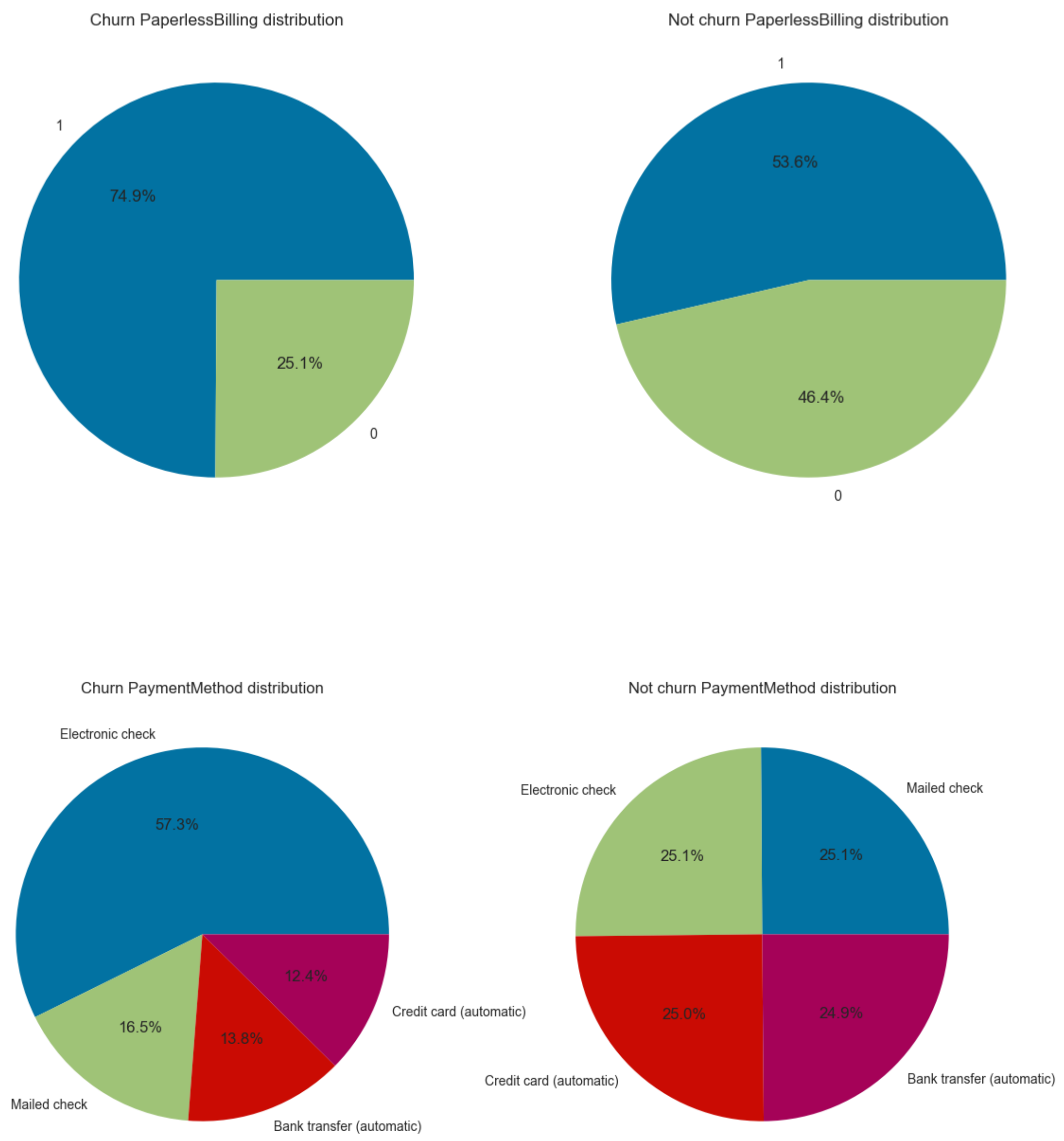
Not churn StreamingMovies distribution

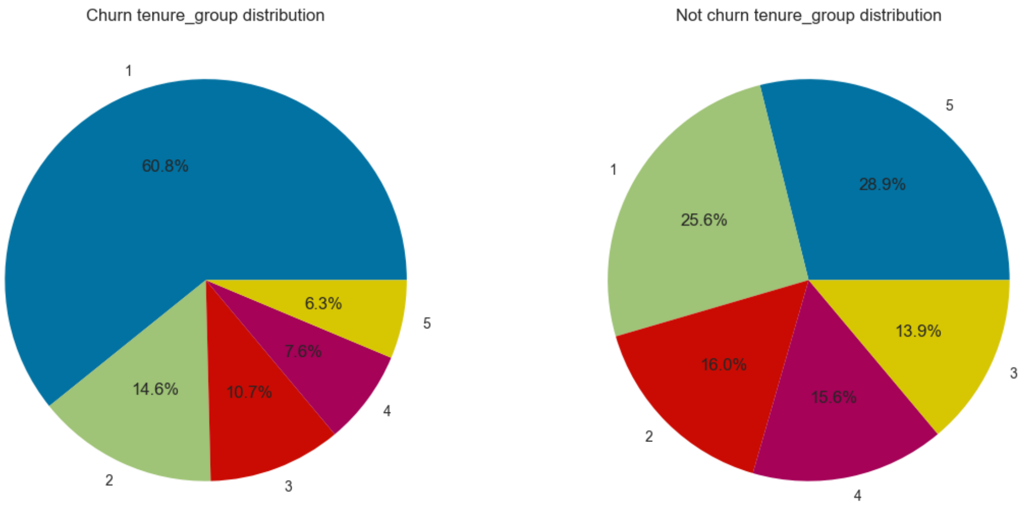


Churn Contract distribution



Not churn Contract distribution


Churn PaperlessBilling distribution
1
74.9%
25.1%
0
Not churn PaperlessBilling distribution
1
53.6%
46.4%
0
Churn PaymentMethod distribution
Electronic check
57.3%
16.5%
13.8%
12.4%
Credit card (automatic)
Mailed check
Bank transfer (automatic)
Not churn PaymentMethod distribution
Electronic check
Mailed check
25.1%
25.1%
25.0%
24.9%
Credit card (automatic)
Bank transfer (automatic)



[U+7537] [U+6027] [U+548C] [U+5973] [U+6027] [U+7684] [U+635F] [U+5931] [U+6BD4] [U+4F8B] [U+76F8] [U+4

[U+5728] [U+6D41] [U+5931] [U+7684] [U+4EBA] [U+53E3] [U+4E2D] [U+FF0C] [U+8001] [U+5E74] [U+4EBA] [U+

[U+6CA1] [U+6709] [U+4F34] [U+4FA3] [U+7684] [U+4EBA] [U+6BD4] [U+4F8B] [U+66F4] [U+9AD8] [U+FF0C] [U+

[U+6CA1] [U+6709] [U+5B69] [U+5B50] [U+7684] [U+4EBA] [U+7684] [U+635F] [U+5931] [U+66F4] [U+9AD8] [U+

[U+662F] [U+5426] [U+66FE] [U+7ECF] [U+4F7F] [U+7528] [U+8FC7] [U+4E0D] [U+76F8] [U+5173] [U+7684] [U+

[U+4E0D] [U+6B62] [U+4E00] [U+4E2A] [U+7535] [U+8BDD] [U+670D] [U+52A1]**?**[U+603B] [U+4E4B] [U+FF0C] [U+

[U+4F7F] [U+7528] **No** [U+7F51] [U+7EDC] [U+670D] [U+52A1] [U+7684] [U+4EBA] [U+5F88] [U+5C11] [U+4E22] [

[U+4EC5] [U+4F7F] [U+7528] [U+4E00] [U+4E2A] [U+6708] **(**[U+4E00] [U+4E2A] [U+6708] [U+5408] [U+540C]**)**[U

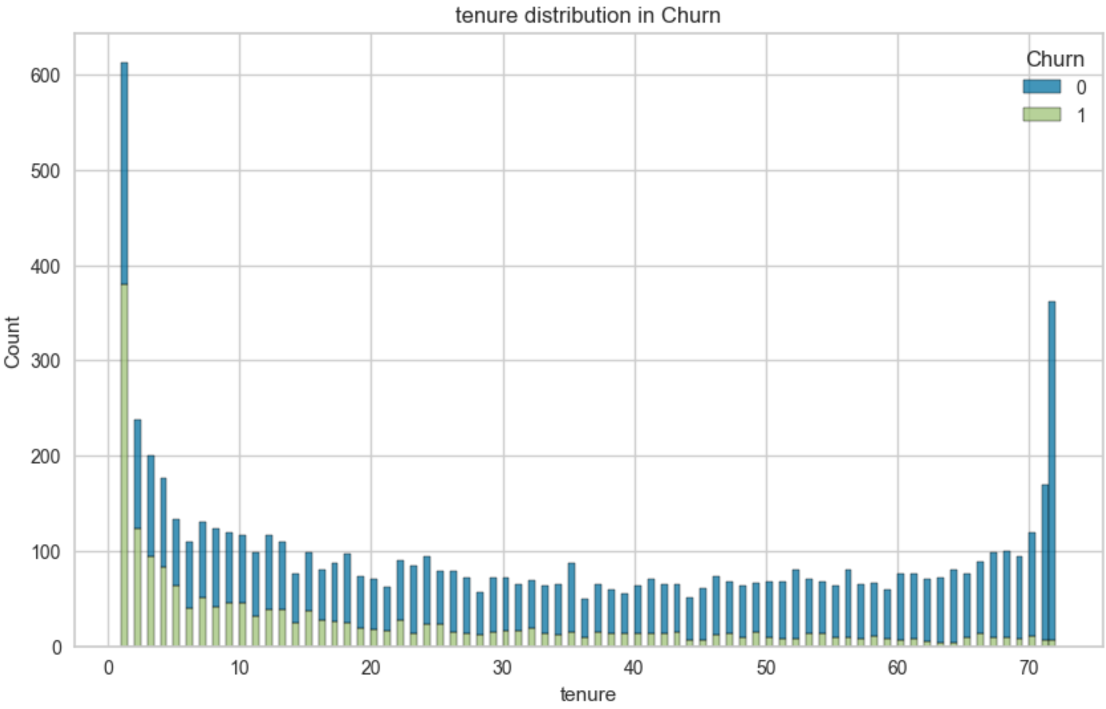
[U+65E0] [U+7EB8] [U+5316] [U+8BA1] [U+8D39] [U+7684] [U+635F] [U+5931] [U+7387] [U+8F83] [U+9AD8] [U+

```
In [ ]: plot_pie(df)
        # [U+6CA1] [U+6709] [U+5BB6] [U+5C5E] [U+7684] [U+7528] [U+6237] [U+53EF] [U+80FD] [U+66F4] [U+50
```

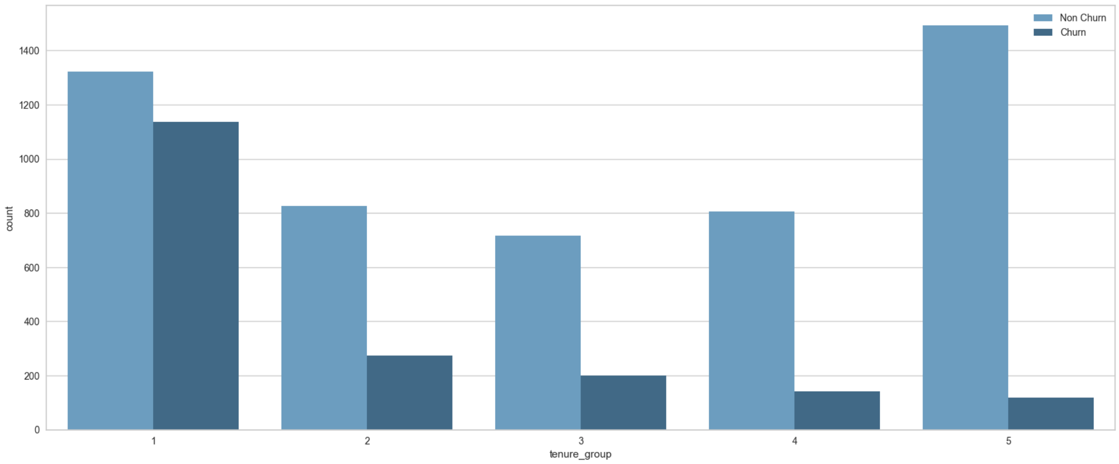
```
In [482]: # [U+76F4] [U+65B9] [U+56FE]
          def zf_Draw(df, column):
              plt.figure(figsize=(10,6))
              sns.histplot(data=df, x=column, hue="Churn", multiple="stack", binwidth=0.5)
              plt.title(column + " distribution in Churn")
              plt.show()
```

```
# [U+6563][U+70B9][U+56FE][U+77E9][U+9635]
def plot_scatter(df, columns):
    sns.pairplot(df[columns], hue="Churn")
    plt.show()
```

```
In [483]: zf_Draw(df,'tenure')
```



```
In [484]: plt.figure(figsize = (20,8))
          sns.countplot(x = df.tenure_group , hue = df.Churn,palette=("Blues_d"))
          plt.legend(['Non Churn' , 'Churn'])

Out[484]: <matplotlib.legend.Legend at 0x13bf08340>
```



4 4.Data modeling

4.0.1 4.1 [U+6570] [U+636E] [U+5904] [U+7406] [U+FF0C] [U+5C06] [U+975E] [U+6570] [U+503C] [U+5F62] [U+5F0

4.1.1 [U+4E8C] [U+5143] [U+53D8] [U+91CF] [U+4F7F] [U+7528] **lable** [U+7F16] [U+7801] **0 / 1** [U+FF1B] [U+591A] [U+5143] [U+53D8] [U+91CF] [U+FF0C] [U+4E3A] [U+4E86] [U+907F] [U+514D] [U+5F15] [U+516 **hot** [U+7F16] [U+7801] [U+6210] [U+5411] [U+91CF] [U+5F62] [U+5F0F]

```
In [485]: #[U+4E8C] [U+5143] [U+8B8A] [U+6578]
          bin_cols = df.nunique()[df.nunique()==2].keys().tolist()
```

```
In [486]: bin_cols
```

```
Out[486]: ['gender',
           'SeniorCitizen',
           'Partner',
           'Dependents',
           'PhoneService',
           'MultipleLines',
           'PaperlessBilling',
           'Churn']
```

```
In [487]: #[U+591A] [U+5143] [U+8B8A] [U+6578]
          multi_cols = [col for col in cat_cols if col not in bin_cols]
          multi_cols
```

```
Out[487]: ['InternetService',
           'OnlineSecurity',
           'OnlineBackup',
           'DeviceProtection',
           'TechSupport',
           'StreamingTV',
           'StreamingMovies',
           'Contract',
           'PaymentMethod',
           'tenure_group']
```

In [488]:
```
test_cols = [1,2,3,4,5]
for number in test_cols:
    # do some thing
```

```
  Cell In[488], line 3
    # do some thing
                   ^
SyntaxError: unexpected EOF while parsing
```

In [489]:
```
# Read in the required kits
# We use label to process the category coding, and logistic must be standardized, and
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
#[U+4E8C][U+5143][U+8B8A][U+6578]
bin_cols = df.nunique()[df.nunique()==2].keys().tolist()
#[U+591A][U+5143][U+8B8A][U+6578]
multi_cols = [col for col in cat_cols if col not in bin_cols]
#[U+5C07][U+4E8C][U+5143][U+6578][U+503C][U+7DE8][U+78BC]
# cato = df.tenure_group.cat.codes
# df.tenure_group = cat
le = LabelEncoder()
# df[multi_cols] = df[multi_cols].replace({0:'No' , 1:'Yes'})
# [U+4ECE] bincols[U+4E00][U+4E2A][U+4E2A][U+53D6][U+51FA][U+503C][U+8FDB][U+884C][U+
for col in bin_cols:
    df[col] = le.fit_transform(df[col])
```

In [490]:
```
df_show = pd.read_csv('/Users/Desktop/code/loss.csv')
multi_cols
```

Out[490]:
```
['InternetService',
 'OnlineSecurity',
 'OnlineBackup',
 'DeviceProtection',
 'TechSupport',
 'StreamingTV',
 'StreamingMovies',
 'Contract',
 'PaymentMethod',
 'tenure_group']
```

In [491]:
```
df_show['Contract']
```

```
Out[491]: 0       Month-to-month
          1             One year
          2       Month-to-month
          3             One year
          4       Month-to-month
                       ...
          7038          One year
          7039          One year
          7040    Month-to-month
          7041    Month-to-month
          7042          Two year
          Name: Contract, Length: 7043, dtype: object
```

[U+72EC] [U+70ED] [U+7F16] [U+7801] [U+FF0C] [U+5047] [U+5982] [U+4E00] [U+4E2A] [U+5C5E] [U+6027] [U+670
[0 0 … 1 … 0] [U+7684] [U+5F62] [U+5F0F] [U+3002] [U+8FD9] [U+4E2A] [U+5411] [U+91CF] [U+957F] [U+5EA6] [U+

[U+4E0D] [U+540C] [U+7684] [U+4F4D] [U+7F6E] [U+53D6] **1** [U+FF0C] [U+5C31] [U+8868] [U+793A] [U+4E0D] [U+540

[U+5982] [U+8001] [U+864E] [U+FF0C] [U+72EE] [U+5B50] [U+FF0C] [U+957F] [U+9888] [U+9E7F]
=> [1 0 0] [U+8001] [U+864E] [U+FF0C] **[0 1 0] =>** [U+72EE] [U+5B50]

[U+7B2C] [U+4E00] [U+4F4D] [U+4EE3] [U+8868] [U+662F] [U+5426] [U+662F] [U+8001] [U+864E] [U+FF0C] [U+7B2

```
In [492]: test = df_show[['Contract']]
```

```
In [493]: test
```

```
Out[493]:             Contract
          0     Month-to-month
          1           One year
          2     Month-to-month
          3           One year
          4     Month-to-month
          ...              ...
          7038        One year
          7039        One year
          7040  Month-to-month
          7041  Month-to-month
          7042        Two year

          [7043 rows x 1 columns]
```

```
In [494]: tt = pd.get_dummies(data = test , columns=['Contract']).astype('int')
```

```
In [495]: tt
          # True [U+548C] False [U+4E0E] 1 / 0 [U+7B49] [U+4EF7] [U+4E0D] [U+7528] [U+8F6C] [U+6362]
```

```
Out[495]:       Contract_Month-to-month  Contract_One year  Contract_Two year
          0                           1                  0                  0
          1                           0                  1                  0
          2                           1                  0                  0
          3                           0                  1                  0
          4                           1                  0                  0
          ...                       ...                ...                ...
          7038                        0                  1                  0
          7039                        0                  1                  0
          7040                        1                  0                  0
          7041                        1                  0                  0
          7042                        0                  0                  1

          [7043 rows x 3 columns]
```

```
In [496]: df_show[['tenure','MonthlyCharges','TotalCharges']]
```

```
Out[496]:       tenure  MonthlyCharges TotalCharges
          0          1           29.85        29.85
          1         34           56.95       1889.5
          2          2           53.85       108.15
          3         45           42.30      1840.75
          4          2           70.70       151.65
          ...      ...             ...          ...
          7038      24           84.80       1990.5
          7039      72          103.20       7362.9
          7040      11           29.60       346.45
          7041       4           74.40        306.6
          7042      66          105.65       6844.5

          [7043 rows x 3 columns]
```

```
In [497]: scaled
```

```
Out[497]:          index    tenure  MonthlyCharges  TotalCharges
          0    -1.732466 -1.280248       -1.161694     -0.994194
          1    -1.731974  0.064303       -0.260878     -0.173740
          2    -1.731482 -1.239504       -0.363923     -0.959649
          3    -1.730990  0.512486       -0.747850     -0.195248
          4    -1.730498 -1.239504        0.196178     -0.940457
          ...        ...       ...             ...           ...
          7027  1.729945 -0.343137        0.664868     -0.129180
          7028  1.730437  1.612573        1.276493      2.241056
          7029  1.730929 -0.872808       -1.170004     -0.854514
          7030  1.731421 -1.158016        0.319168     -0.872095
          7031  1.731913  1.368109        1.357932      2.012344

          [7032 rows x 4 columns]
```

```
In [498]: # [U+4E3A] [U+4E86] [U+907F] [U+514D] [U+5F15] [U+5165] [U+5927] [U+5C0F] [U+5173] [U+7CFB] [U+
          df = pd.get_dummies(data = df , columns=multi_cols)

          # Handle continuous variables
          std = StandardScaler()
          scaled = std.fit_transform(df[num_cols])
          scaled = pd.DataFrame(scaled,columns=num_cols)

          df_origin =  df.copy()
          df = df.drop(columns=num_cols , axis = 1)
          df = df.merge(scaled , left_index=True , right_index=True , how = 'left')
```

```
In [499]: df
```

```
Out[499]:       customerID  gender  SeniorCitizen  Partner  Dependents  PhoneService  \
          0     7590-VHVEG       0              0        1           0             0
          1     5575-GNVDE       1              0        0           0             1
          2     3668-QPYBK       1              0        0           0             1
          3     7795-CFOCW       1              0        0           0             0
          4     9237-HQITU       0              0        0           0             1
          ...          ...     ...            ...      ...         ...           ...
          7027  6840-RESVB       1              0        1           1             1
          7028  2234-XADUH       0              0        1           1             1
          7029  4801-JZAZL       0              0        1           1             0
          7030  8361-LTMKD       1              1        1           0             1
          7031  3186-AJIEK       1              0        0           0             1

                MultipleLines  PaperlessBilling  Churn  InternetService_0  ...  \
          0                 0                 1      0              False  ...
          1                 0                 0      0              False  ...
          2                 0                 1      1              False  ...
          3                 0                 0      0              False  ...
          4                 0                 1      1              False  ...
          ...             ...               ...    ...                ...  ...
          7027              1                 1      0              False  ...
          7028              1                 1      0              False  ...
          7029              0                 1      0              False  ...
          7030              1                 1      1              False  ...
          7031              0                 1      0              False  ...

                PaymentMethod_Mailed check  tenure_group_1  tenure_group_2  \
          0                          False            True           False
          1                           True           False           False
          2                           True            True           False
          3                          False           False           False
          4                          False            True           False
          ...                          ...             ...             ...
          7027                        True           False            True
```

```
7028                       False           False           False
7029                       False            True           False
7030                        True            True           False
7031                       False           False           False

      tenure_group_3  tenure_group_4  tenure_group_5     index    tenure  \
0              False           False           False -1.732466 -1.280248
1               True           False           False -1.731974  0.064303
2              False           False           False -1.731482 -1.239504
3              False            True           False -1.730990  0.512486
4              False           False           False -1.730498 -1.239504
...              ...             ...             ...       ...       ...
7027           False           False           False  1.729945 -0.343137
7028           False           False            True  1.730437  1.612573
7029           False           False           False  1.730929 -0.872808
7030           False           False           False  1.731421 -1.158016
7031           False           False            True  1.731913  1.368109

      MonthlyCharges  TotalCharges
0          -1.161694     -0.994194
1          -0.260878     -0.173740
2          -0.363923     -0.959649
3          -0.747850     -0.195248
4           0.196178     -0.940457
...              ...           ...
7027        0.664868     -0.129180
7028        1.276493      2.241056
7029       -1.170004     -0.854514
7030        0.319168     -0.872095
7031        1.357932      2.012344

[7032 rows x 46 columns]
```

PCA [U+4E3B] [U+6210] [U+5206] [U+5206] [U+6790] [U+7684] [U+76EE] [U+7684] [U+FF1A]
[U+4F7F] [U+7528] [U+66F4] [U+5C11] [U+7684] [U+7EF4] [U+5EA6] [U+FF08] [U+5C5E] [U+6027] [U+FF09] [U+8868

[U+4E5F] [U+79F0] [U+4E4B] [U+4E3A] [U+6295] [U+5F71] [U+FF08] [U+628A] [U+9AD8] [U+7EF4] [U+7684] [U+6570
[U+4F7F] [U+5F97] [U+964D] [U+7EF4] [U+540E] [U+7684] [U+6570] [U+636E] [U+4E4B] [U+95F4] [U+7684] [U+65B9

[U+8868] [U+660E] [U+8FD9] [U+79CD] [U+6295] [U+5F71] [U+65B9] [U+5F0F] [U+65E0] [U+6CD5] [U+5B8C] [U+6574

https://www.zhihu.com/question/41120789/answer/2918798394

```
In [500]: from sklearn.decomposition import  PCA
          # [U+6574] [U+6570] [U+8868] [U+793A] [U+964D] [U+5230] [U+7684] [U+7EF4] [U+6570]
          # [U+5C0F] [U+6570] [U+8868] [U+793A] [U+9700] [U+8981] [U+4FDD] [U+6301] [U+7684] [U+4FE1] [U+
          pca = PCA(n_components = 2)
          X = df[[col for col in df.columns if col not in Id_col + target_col]]
          Y = df[target_col + Id_col]

          pc = pca.fit_transform(X)

          # [U+4F7F] [U+7528] [U+9006] [U+53D8] [U+6362] [U+91CD] [U+6784] [U+6570] [U+636E]
          X_reconstructed = pca.inverse_transform(pc)
          # [U+8BA1] [U+7B97] [U+91CD] [U+6784] [U+8BEF] [U+5DEE]
          from sklearn.metrics import mean_squared_error
          reconstruction_error = mean_squared_error(X, X_reconstructed)

          print(reconstruction_error)
```

```
0.1446012083410596
```

In [501]:
```
Y
```

Out[501]:
```
      Churn  customerID
0         0  7590-VHVEG
1         0  5575-GNVDE
2         1  3668-QPYBK
3         0  7795-CFOCW
4         1  9237-HQITU
...     ...         ...
7027      0  6840-RESVB
7028      0  2234-XADUH
7029      0  4801-JZAZL
7030      1  8361-LTMKD
7031      0  3186-AJIEK

[7032 rows x 2 columns]
```

In [502]:
```
pca_data
```

Out[502]:
```
           PC1       PC2      Churn  customerID
0    -1.601119 -1.651747  Not Churn  7590-VHVEG
1    -0.225030 -0.175515  Not Churn  5575-GNVDE
2    -1.318313 -1.489117      Churn  3668-QPYBK
3    -0.084991  0.442937  Not Churn  7795-CFOCW
4    -0.980941 -2.376076      Churn  9237-HQITU
...        ...       ...        ...         ...
7027  0.775548  0.063506  Not Churn  6840-RESVB
7028  3.350979  1.118409  Not Churn  2234-XADUH
7029 -1.470925 -1.299471  Not Churn  4801-JZAZL
7030 -0.745052 -2.087808      Churn  8361-LTMKD
7031  3.003348  1.140132  Not Churn  3186-AJIEK
```

```
[7032 rows x 4 columns]
```

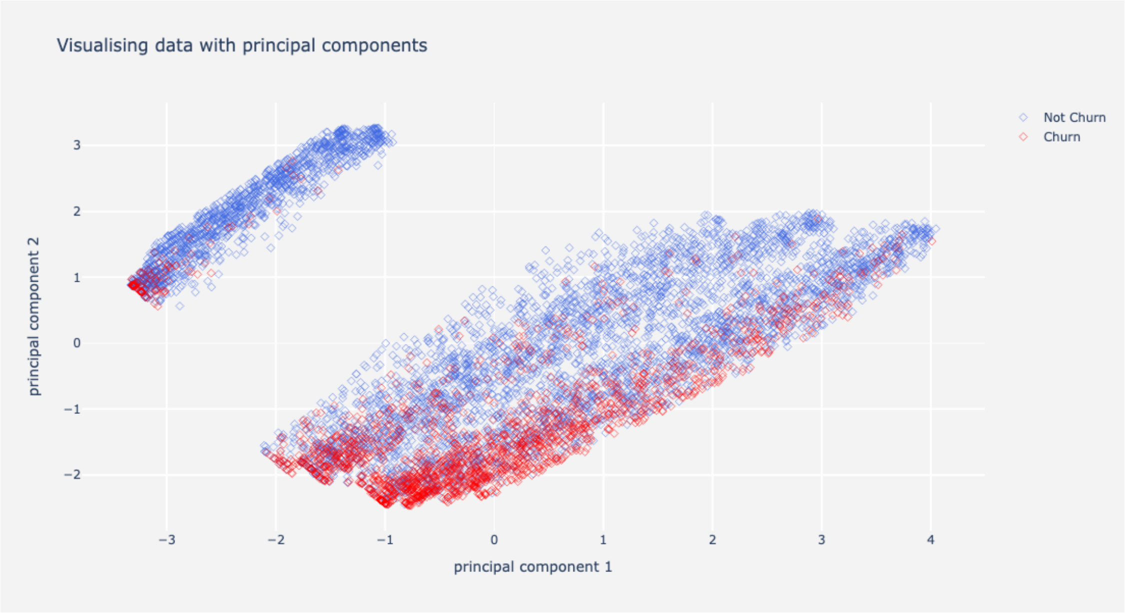
```
In [503]: pca_data  = pd.DataFrame(pc , columns=['PC1' , 'PC2'])
          pca_data = pca_data.merge(Y , left_index = True , right_index = True , how = 'left')
          pca_data = pca_data.replace({1:'Churn' , 0: 'Not Churn'})
```

```
In [504]: pca_data
```

```
Out[504]:            PC1       PC2      Churn  customerID
          0    -1.601119 -1.651747  Not Churn  7590-VHVEG
          1    -0.225030 -0.175515  Not Churn  5575-GNVDE
          2    -1.318313 -1.489117      Churn  3668-QPYBK
          3    -0.084991  0.442937  Not Churn  7795-CFOCW
          4    -0.980941 -2.376076      Churn  9237-HQITU
          ...        ...       ...        ...         ...
          7027  0.775548  0.063506  Not Churn  6840-RESVB
          7028  3.350979  1.118409  Not Churn  2234-XADUH
          7029 -1.470925 -1.299471  Not Churn  4801-JZAZL
          7030 -0.745052 -2.087808      Churn  8361-LTMKD
          7031  3.003348  1.140132  Not Churn  3186-AJIEK

          [7032 rows x 4 columns]
```

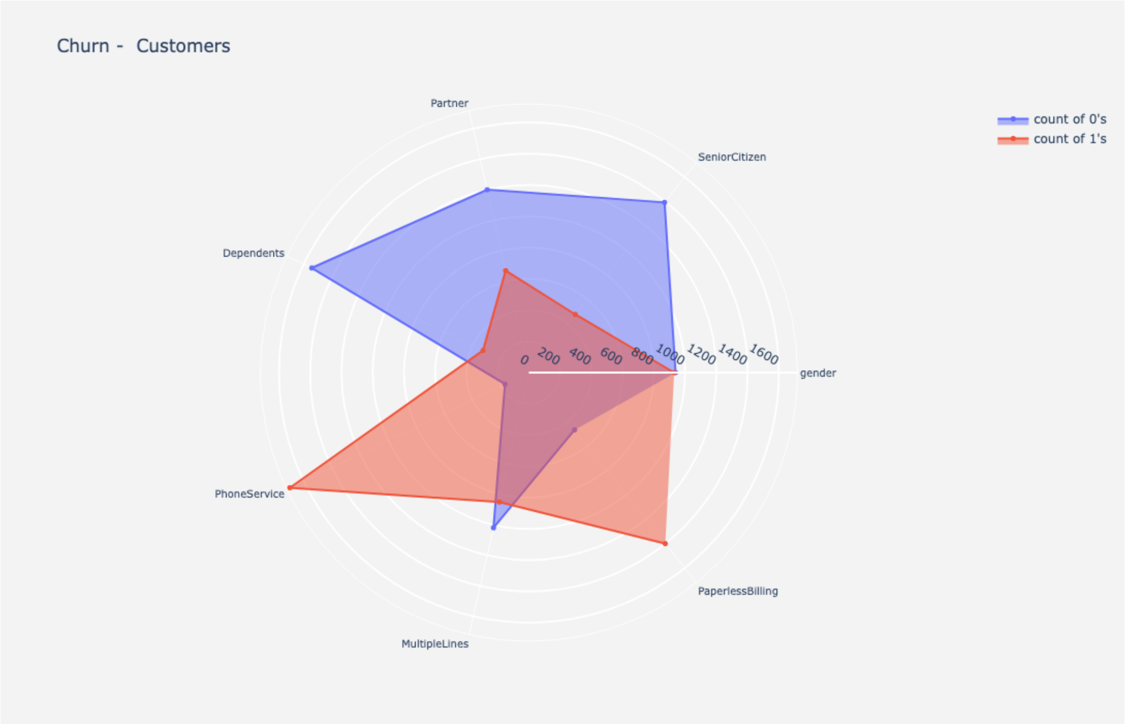
```
In [505]: def pca_scatter(target,color):
              tracer = go.Scatter(x = pca_data[pca_data["Churn"] == target]["PC1"] ,
                                  y = pca_data[pca_data["Churn"] == target]["PC2"],
                                  name = target,mode = "markers",
                                  marker = dict(color = color,
                                                line = dict(width = .5),
                                                symbol =  "diamond-open"),
                                  text = ("Customer Id : " +
                                          pca_data[pca_data["Churn"] == target]['customerID'])
                                 )
              return tracer
          layout = go.Layout(dict(title = "Visualising data with principal components",
                                  plot_bgcolor  = "rgb(243,243,243)",
                                  paper_bgcolor = "rgb(243,243,243)",
                                  xaxis = dict(gridcolor = 'rgb(255, 255, 255)',
                                               title = "principal component 1",
                                               zerolinewidth=1,ticklen=5,gridwidth=2),
                                  yaxis = dict(gridcolor = 'rgb(255, 255, 255)',
                                               title = "principal component 2",
                                               zerolinewidth=1,ticklen=5,gridwidth=2),
                                  height = 600
                                 )
                            )
          trace1 = pca_scatter("Churn",'red')
          trace2 = pca_scatter("Not Churn",'royalblue')
```

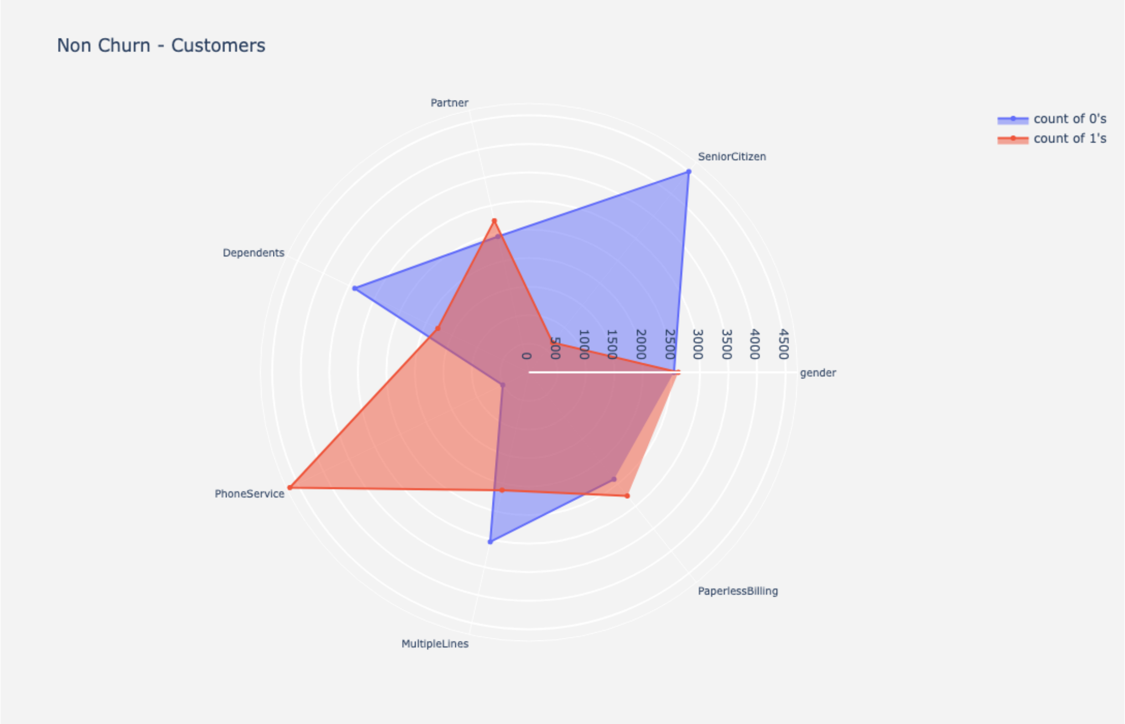
```
data = [trace2,trace1]
fig = go.Figure(data=data,layout=layout)
py.iplot(fig)
```



In [506]:

```
bi_cs = bin_cols
dat_rad = df[bin_cols]
#[U+756B][U+51FA][U+96F7][U+9054][U+5716]
def plot_radar(df,aggregate,title) :
    data_frame = df[df["Churn"] == aggregate]

    data_frame_x = data_frame[bi_cs].sum().reset_index()
    data_frame_x.columns  = ["feature","yes"]
    data_frame_x["no"]    = data_frame.shape[0]  - data_frame_x["yes"]
    data_frame_x  = data_frame_x[data_frame_x["feature"] != "Churn"]

    #count of 1's(yes)
    trace1 = go.Scatterpolar(r = data_frame_x["yes"].values.tolist(),
                             theta = data_frame_x["feature"].tolist(),
                             fill  = "toself",name = "count of 1's",
                             mode = "markers+lines",
                             marker = dict(size = 5)
                            )
    #count of 0's(No)
    trace2 = go.Scatterpolar(r = data_frame_x["no"].values.tolist(),
                             theta = data_frame_x["feature"].tolist(),
                             fill  = "toself",name = "count of 0's",
                             mode = "markers+lines",
                             marker = dict(size = 5)
                            )
    layout = go.Layout(dict(polar = dict(radialaxis = dict(visible = True,
                                                           side = "counterclockwise",
                                                           showline = True,
                                                           linewidth = 2,
                                                           tickwidth = 2,
                                                           gridcolor = "white",
                                                           gridwidth = 2),
                                         angularaxis = dict(tickfont = dict(size = 10),
                                                            layer = "below traces"
                                                           ),
                                         bgcolor  = "rgb(243,243,243)",
                                        ),
                            paper_bgcolor = "rgb(243,243,243)",
                            title = title,height = 700))

    data = [trace2,trace1]
    fig = go.Figure(data=data,layout=layout)
    py.iplot(fig)
#plot
plot_radar(dat_rad,1,"Churn -  Customers")
plot_radar(dat_rad,0,"Non Churn - Customers")
```





5 [U+4E8C] [U+5143] => [U+8F93] [U+51FA] [U+6982] [U+7387]

6 [U+51B3] [U+7B56] [U+9608] [U+503C] $\beta = 0.5$

In [8]:
```
from sklearn.model_selection import  train_test_split
from sklearn.linear_model import  LogisticRegression
from sklearn.metrics import confusion_matrix , accuracy_score , classification_report
from sklearn.metrics import  roc_auc_score , roc_curve
from sklearn.metrics import f1_score
import statsmodels.api as sm
from sklearn.metrics import  precision_score ,recall_score
from yellowbrick.classifier import DiscriminationThreshold
# splitting train and test data
#  [U+8BAD][U+7EC3][U+6570][U+636E]  :  [U+9884][U+6D4B][U+6570][U+636E]  [U+6837][U+672C]
train , test = train_test_split(df , test_size = 0.25 , random_state = 3 )

train2 , test2 = train_test_split(df , test_size = 0.25 , random_state = 4 )


cols = [col for col in df.columns if col not in Id_col + target_col]
train_X =train[cols]
train_Y = train[target_col]
```

```
    test_X = test[cols]
    test_Y = test[target_col]
```

```
      File "<ipython-input-8-e9b93b7fcb9c>", line 15
    1 2 3 4 5 6 7  8 9 0 = > 0 9 8 7 6 5 4 | 3 2 1
      ^
SyntaxError: invalid syntax
```

```
In [ ]: #[U+5EFA] [U+6A21] [U+7684] [U+6642] [U+5019] [U+901A] [U+5E38] [U+6703] [U+7528] [U+4E0D] [U+53E
        def select_model_prediction(algorithm,training_x,testing_x,
                                     training_y,testing_y,cols,cf,threshold_plot) :
            #model
            algorithm.fit(training_x,training_y)
            predictions   = algorithm.predict(testing_x)

            #[U+5206] [U+985E] [U+6A21] [U+578B] [U+7684] [U+6A5F] [U+7387] [U+6211] [U+5011] [U+8981] [U
            probabilities = algorithm.predict_proba(testing_x)
            #coeffs
            if   cf == "coefficients" :
                coefficients  = pd.DataFrame(algorithm.coef_.ravel())
            elif cf == "features" :
                coefficients  = pd.DataFrame(algorithm.feature_importances_)

            column_df     = pd.DataFrame(cols)
            coef_sumry    = (pd.merge(coefficients,column_df,left_index= True,
                                      right_index= True, how = "left"))
            coef_sumry.columns = ["coefficients","features"]
            coef_sumry    = coef_sumry.sort_values(by = "coefficients",ascending = False)

            print(algorithm)
            print("\n Classification report : \n",classification_report(testing_y,predictions))
            print("Accuracy   Score : ",accuracy_score(testing_y,predictions))
            #confusion matrix
            conf_matrix = confusion_matrix(testing_y,predictions)
            #roc_auc_score
            model_roc_auc = roc_auc_score(testing_y,predictions)
            print("Area under curve : ",model_roc_auc,"\n")
            fpr,tpr,thresholds = roc_curve(testing_y,probabilities[:,1])

            #plot confusion matrix
            trace1 = go.Heatmap(z = conf_matrix ,
                                x = ["Not churn","Churn"],
                                y = ["Not churn","Churn"],
                                showscale  = False,colorscale = "Picnic",
                                name = "matrix")
```

```
    #plot roc curve
    trace2 = go.Scatter(x = fpr,y = tpr,
                        name = "Roc : " + str(model_roc_auc),
                        line = dict(color = ('rgb(22, 96, 167)'),width = 2))
    trace3 = go.Scatter(x = [0,1],y=[0,1],
                        line = dict(color = ('rgb(205, 12, 24)'),width = 2,
                        dash = 'dot'))

    #plot coeffs
    trace4 = go.Bar(x = coef_sumry["features"],y = coef_sumry["coefficients"],
                    name = "coefficients",
                    marker = dict(color = coef_sumry["coefficients"],
                                  colorscale = "Picnic",
                                  line = dict(width = .6,color = "black")))

    #subplots
    fig = tls.make_subplots(rows=2, cols=2, specs=[[{}, {}], [{'colspan': 2}, None]],
                            subplot_titles=('Confusion Matrix',
                                            'Receiver operating characteristic',
                                            'Feature Importances'))

    fig.append_trace(trace1,1,1)
    fig.append_trace(trace2,1,2)
    fig.append_trace(trace3,1,2)
    fig.append_trace(trace4,2,1)

    fig['layout'].update(showlegend=False, title="Model performance" ,
                         autosize = False,height = 900,width = 800,
                         plot_bgcolor = 'rgba(240,240,240, 0.95)',
                         paper_bgcolor = 'rgba(240,240,240, 0.95)',
                         margin = dict(b = 195))
    fig["layout"]["xaxis2"].update(dict(title = "false positive rate"))
    fig["layout"]["yaxis2"].update(dict(title = "true positive rate"))
    fig["layout"]["xaxis3"].update(dict(showgrid = True,tickfont = dict(size = 10),
                                        tickangle = 90))
    py.iplot(fig)

    #[U+7528]yellow_brick[U+5E6B][U+6211][U+5011][U+53EF][U+8996][U+5316][U+5716][U+724

    if threshold_plot == True :
        visualizer = DiscriminationThreshold(algorithm)
        visualizer.fit(training_x,training_y)
        visualizer.poof()
```

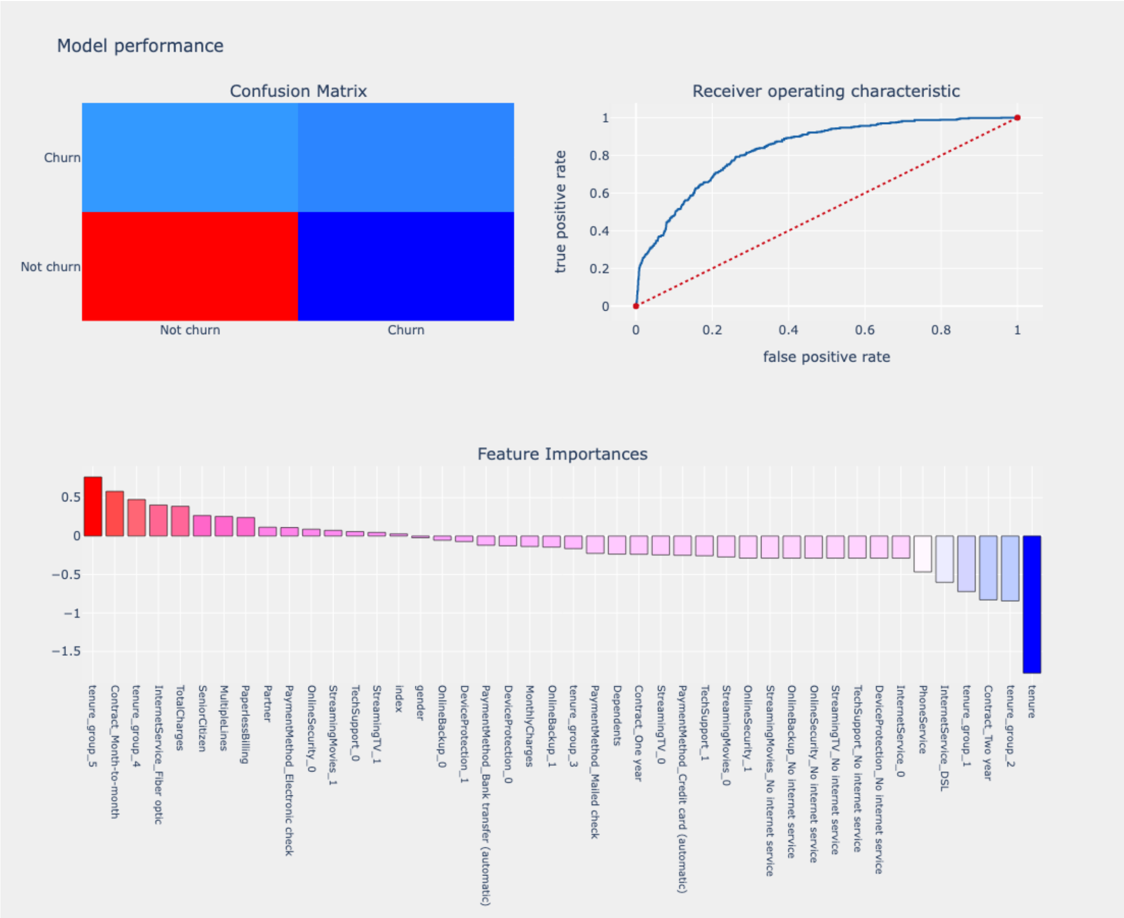
```
In [ ]: # [U+51B3][U+7B56][U+9608][U+503C] => 0.5
        # [U+51B3][U+7B56][U+9608][U+503C][U+662F][U+5426][U+5E94][U+8BE5][U+662F]0.5

        # [U+51B3][U+7B56][U+9608][U+503C][U+8BBE][U+7F6E][U+7684][U+503C] [U+5F71][U+54CD] [U
        #               [ 0    0    0     0     1    1 ]
        # prediction = [ 0.3, 0.4, 0.4, 0.51, 0.8, 0.9]

        # => 0.5 => RESULT = 0 , 0 , 0 ,  1,    1  ,  1 ] 5/6
        # => 0.6           = 0   0   0    0     1     1   6/6
```

```
In [508]: #[U+5BEB][U+597D]logistic[U+7684][U+6F14][U+7B97][U+6CD5][U+FF0C][U+6B63][U+5247][U+6
          logit  = LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
            intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
            penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
            verbose=0, warm_start=False)
          #[U+8DD1]model
          select_model_prediction (logit,train_X,test_X,train_Y,test_Y,
                                   cols,"coefficients",threshold_plot = True)
```

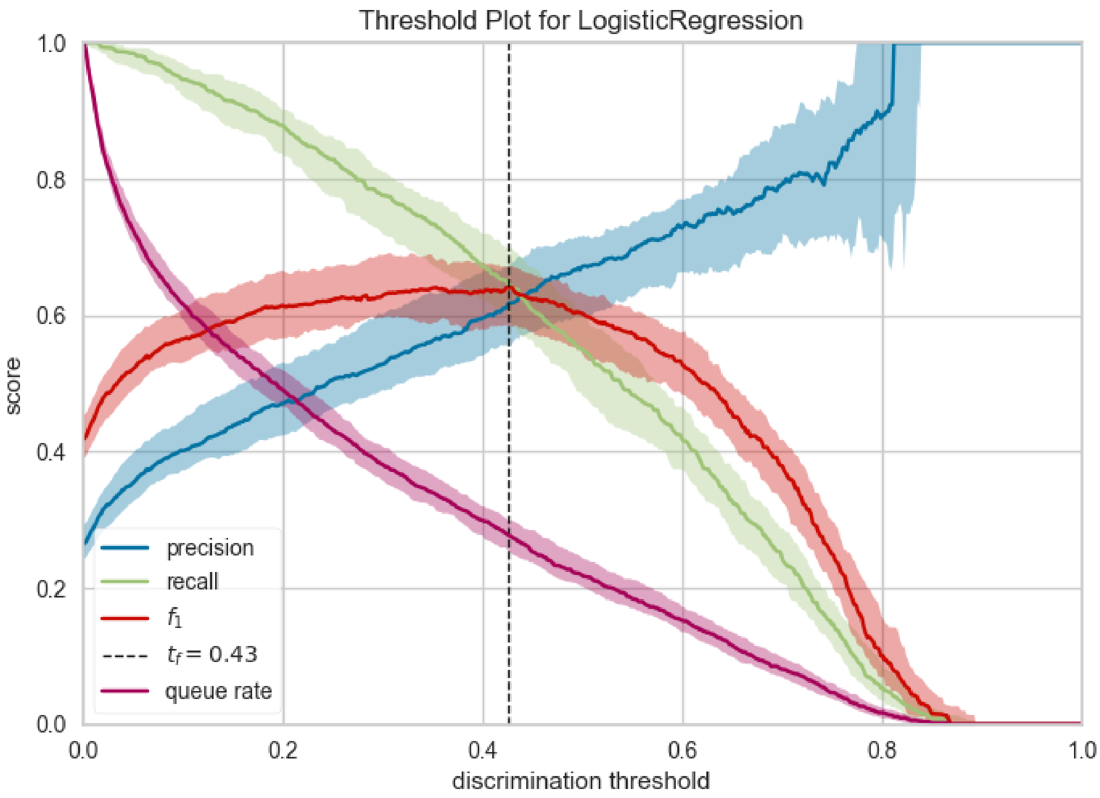
```
A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_sa
```

```
plotly.tools.make_subplots is deprecated, please use plotly.subplots.make_subplots instead
```

```
LogisticRegression(multi_class='ovr', n_jobs=1, solver='liblinear')

 Classification report :
               precision    recall  f1-score   support

           0       0.83      0.90      0.86      1300
           1       0.63      0.48      0.55       458

    accuracy                           0.79      1758
   macro avg       0.73      0.69      0.71      1758
weighted avg       0.78      0.79      0.78      1758

Accuracy   Score :  0.7906712172923777
Area under curve :  0.6915888478333891
```



```
X does not have valid feature names, but LogisticRegression was fitted with feature names
```



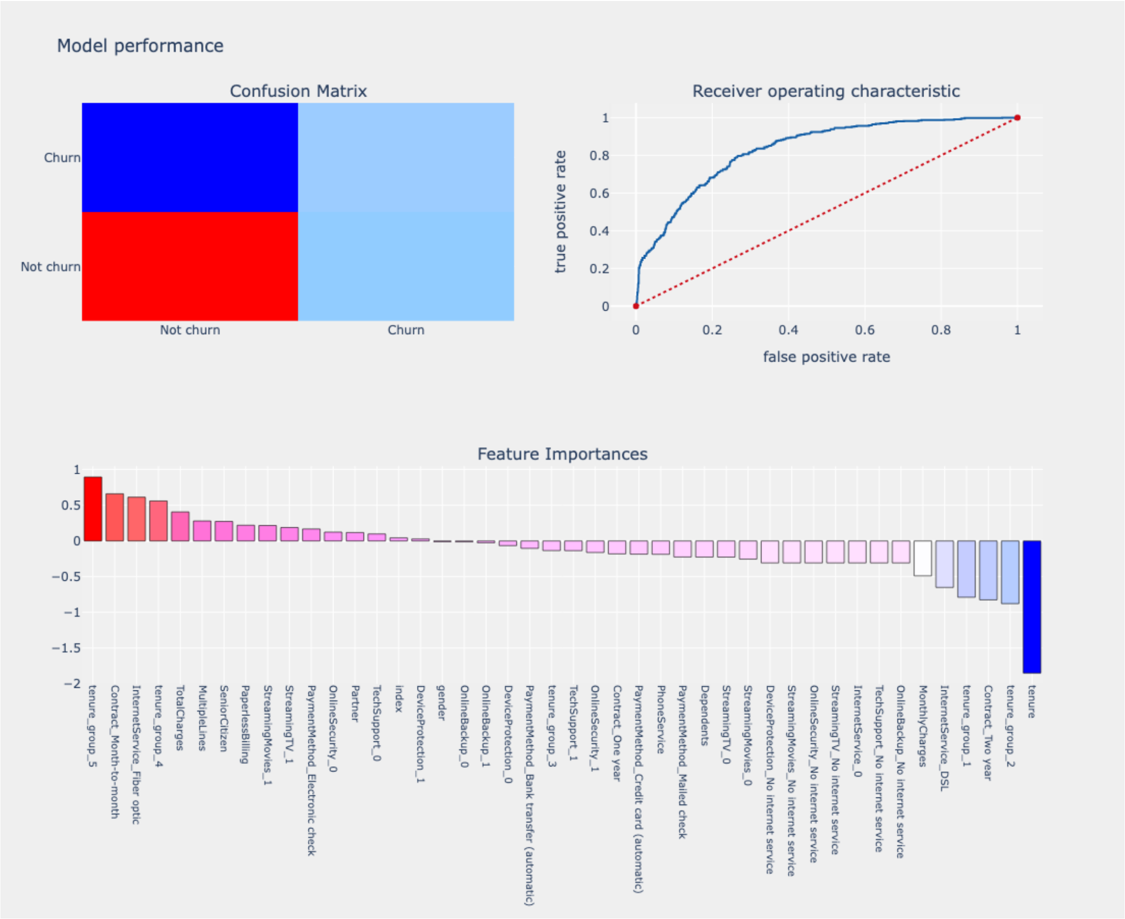
```
In [411]: # [U+4F7F][U+7528][U+6A21][U+578B][U+8FDB][U+884C][U+9884][U+6D4B]
          predicted_values = logit.predict(test_X)

          # [U+6253][U+5370][U+9884][U+6D4B][U+503C]
          print(predicted_values)
```

```
[0 0 1 ... 0 0 0]
```

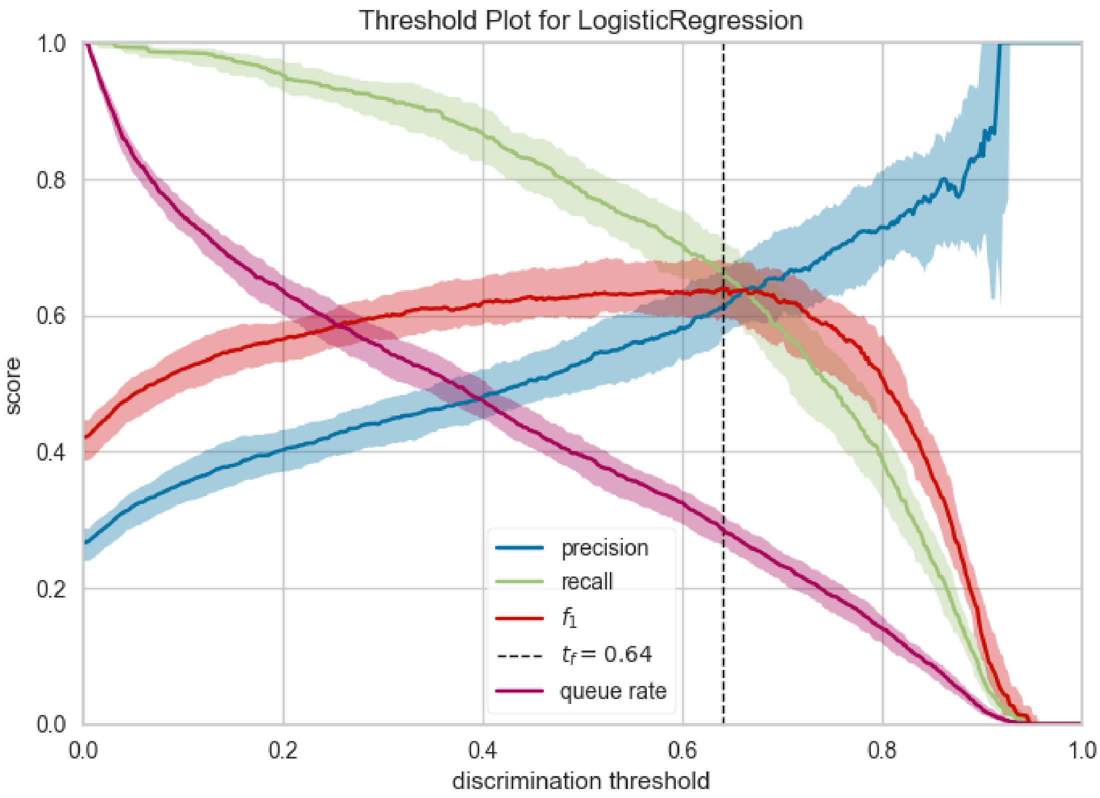
```
In [509]: #[U+5BEB][U+597D]logistic[U+7684][U+6F14][U+7B97][U+6CD5][U+FF0C][U+6B63][U+5247][U+6
          class_weight1 = {0:1 , 1:2.5}
          logit  = LogisticRegression(C=1.0, class_weight=class_weight1, dual=False, fit_interce
            intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
            penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
            verbose=0, warm_start=False)
          #[U+8DD1]model
          select_model_prediction (logit,train_X,test_X,train_Y,test_Y,
                                   cols,"coefficients",threshold_plot = True)
```

```
A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_sa
plotly.tools.make_subplots is deprecated, please use plotly.subplots.make_subplots instead
```

```
LogisticRegression(class_weight={0: 1, 1: 2.5}, multi_class='ovr', n_jobs=1,
                   solver='liblinear')

 Classification report :
              precision    recall  f1-score   support

           0       0.91      0.74      0.81      1300
           1       0.51      0.79      0.62       458

    accuracy                           0.75      1758
   macro avg       0.71      0.76      0.72      1758
weighted avg       0.81      0.75      0.76      1758

Accuracy   Score :  0.7502844141069397
Area under curve :  0.7625663419549883
```



Model performance
Confusion Matrix
Churn
Not churn
Not churn
Churn
Receiver operating characteristic
true positive rate
false positive rate
0
0.2
0.4
0.6
0.8
1
Feature Importances
1
0.5
0
−0.5
−1
−1.5
−2
tenure_group_5
Contract_Month-to-month
InternetService_Fiber optic
tenure_group_4
TotalCharges
MultipleLines
SeniorCitizen
PaperlessBilling
StreamingMovies_1
StreamingTV_1
PaymentMethod_Electronic check
OnlineSecurity_0
Partner
TechSupport_0
index
DeviceProtection_1
gender
OnlineBackup_0
OnlineBackup_1
DeviceProtection_0
PaymentMethod_Bank transfer (automatic)
tenure_group_3
TechSupport_1
OnlineSecurity_1
Contract_One year
PaymentMethod_Credit card (automatic)
PhoneService
PaymentMethod_Mailed check
Dependents
StreamingTV_0
StreamingMovies_0
DeviceProtection_No internet service
StreamingMovies_No internet service
OnlineSecurity_No internet service
StreamingTV_No internet service
InternetService_0
TechSupport_No internet service
OnlineBackup_No internet service
MonthlyCharges
InternetService_DSL
tenure_group_1
Contract_Two year
tenure_group_2
tenure


Threshold Plot for LogisticRegression
score
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
discrimination threshold
precision
recall
f_1
$t_f = 0.64$
queue rate