THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

TIME SERIES MODELING & ANALYSIS

Instructor Name: Reza Jafari

Lab#: 11

Submitted by Dinesh Kumar Padmanabhan

Date: 03-Dec-2020

ABSTRACT

The main purpose of this LAB is to estimate and plot the survival function for real data set using “lifelines” library in Python. Using survival analysis we can study the fundamental questions like How long will particular customer remains with your business? How long will this machine last, after successfully running for a year? What is the relative retention rate of different marketing channels? What is the likelihood that a patient will survive after being diagnosed ? All of above questions can studied under Survival Analysis topic.

INTRODUCTION

Theory

Survival analysis is a set of statistical tools which answer the following question: How long would it be, before a particular event occurs or a particular event occurs? In other words we can also call it as a ”time to event analysis”. This method is called survival analysis because it was developed by medical researches and they were more interested in finding expected lifetime of patients in different cohort. The method can be further applied to not just traditional death events, but to many different types of events of interest in different businesses domains like

Predictive maintenance

Customer Analytics:(customer retention)

Marketing Analytics:(Cohort Analysis)

Mathematical Intuition

Survival function:

$$S(t) = 1 - F(t) = P(T \geq t)$$

S(t) gives us the probability that the event has not occurred by the time t.

In simple words, S(t) gives us the proportion of population with the time to event value more than t.

$$\int_t^{\infty} f(x)dx$$

Along with the survival function, we are also interested in the rate at which event is taking place, out of the surviving population at any given t.

Hazard Function: h(t):

$$h(t) = \lim_{dt \to 0} \frac{S(t) - S(t+dt)}{dt} \times \frac{1}{S(t)}$$
$$= \frac{S(t)'}{S(t)}$$

METHOD, THEORY & PROCEDURES

Method:

1. Programming Language: Python

Libraries used: Some basic libraries used for analysis & model building are mentioned below

library(Numpy) - large collection of high-level mathematical functions to operate on these arrays.

library (Pandas) – For Data manipulation and analysis

library(Matplotlib) – is a system for declaratively creating graphics

library(Math) –To Compute mathematical calculations

library (statsmodels) – Import statistical models

library (scipy) – Scientific Computations

library lifelines – statistical computations

Theory:

To estimate and plot the survival function for real data set.

Procedure:

I shall be looking at the results for WA_Fn-UseC_-Telco-Customer-Churn.csv using survival function. Perform various plots and infer about it in my analysis. And through my exploration I shall try to identify which methods perform better and draw inferences.

The Dataset will be explored in following stages:

1. **Data Exploration (EDA)** – looking at the models and making inferences about the data.
2. **Data Visualization** – Plotting different time series plots for the regression method and forecast accuracy.
3. **Testing** – Running Autocorrelation, Pearson correlation test to identify the correlation between errors.

ANSWERS TO QUESTIONS

File - unknown

```
C:\ProgramData\Anaconda3\python.exe "C:\Program Files\
JetBrains\PyCharm 2019.3.1\plugins\python\helpers\pydev\
pydevconsole.py" --mode=client --port=53135

import sys; print('Python %s on %s' % (sys.version, sys.
platform))
sys.path.extend(['C:\\Users\\nsree_000\\Desktop\\Python-
Quiz', 'C:/Users/nsree_000/Desktop/Python-Quiz'])

Python 3.7.4 (default, Aug  9 2019, 18:34:13) [MSC v.1915
64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more
information
IPython 7.8.0 -- An enhanced Interactive Python. Type '?'
for help.
PyDev console: using IPython 7.8.0

Python 3.7.4 (default, Aug  9 2019, 18:34:13) [MSC v.1915
64 bit (AMD64)] on win32
In[2]: runfile('C:/Users/nsree_000/Desktop/Python-Quiz/TIME
 SERIES/labs-911/LAB11.py', wdir='C:/Users/nsree_000/
Desktop/Python-Quiz/TIME SERIES/labs-911')
Loading csv as dataframe df:
      customerID  gender  SeniorCitizen  ...
MonthlyCharges TotalCharges  Churn
0     7590-VHVEG  Female              0  ...          29.85
        29.85     No
1     5575-GNVDE    Male              0  ...          56.95
        1889.5    No
2     3668-QPYBK    Male              0  ...          53.85
        108.15    Yes
3     7795-CFOCW    Male              0  ...          42.30
        1840.75   No
4     9237-HQITU  Female              0  ...          70.70
        151.65    Yes
          ...     ...             ...  ...           ...
        ...    ...
7038  6840-RESVB    Male              0  ...          84.80
        1990.5    No
7039  2234-XADUH  Female              0  ...         103.20
        7362.9    No
7040  4801-JZAZL  Female              0  ...          29.60
        346.45    No
7041  8361-LTMKD    Male              1  ...          74.40
        306.6     Yes
7042  3186-AJIEK    Male              0  ...         105.65
```

Page 1 of 2

File - unknown

```
25        6844.5     No
26
27 [7043 rows x 21 columns]
28    customerID  gender  SeniorCitizen  ... MonthlyCharges
   TotalCharges  Churn
29 0  7590-VHVEG  Female              0  ...          29.85
          29.85     No
30 1  5575-GNVDE    Male              0  ...          56.95
         1889.5     No
31 2  3668-QPYBK    Male              0  ...          53.85
         108.15    Yes
32 3  7795-CFOCW    Male              0  ...          42.30
        1840.75     No
33 4  9237-HQITU  Female              0  ...          70.70
         151.65    Yes
34
35 [5 rows x 21 columns]
36 <class 'pandas.core.frame.DataFrame'>
37 RangeIndex: 7043 entries, 0 to 7042
38 Data columns (total 21 columns):
39 customerID          7043 non-null object
40 gender              7043 non-null object
41 SeniorCitizen       7043 non-null int64
42 Partner             7043 non-null object
43 Dependents          7043 non-null object
44 tenure              7043 non-null int64
45 PhoneService        7043 non-null object
46 MultipleLines       7043 non-null object
47 InternetService     7043 non-null object
48 OnlineSecurity      7043 non-null object
49 OnlineBackup        7043 non-null object
50 DeviceProtection    7043 non-null object
51 TechSupport         7043 non-null object
52 StreamingTV         7043 non-null object
53 StreamingMovies     7043 non-null object
54 Contract            7043 non-null object
55 PaperlessBilling    7043 non-null object
56 PaymentMethod       7043 non-null object
57 MonthlyCharges      7043 non-null float64
58 TotalCharges        7043 non-null object
59 Churn               7043 non-null object
60 dtypes: float64(1), int64(2), object(18)
61 memory usage: 1.1+ MB
62 None
63
```

Page 2 of 2

1. Using Pandas library .read_csv function load the “WA_Fn-UseC_-Telco-Customer-Churn.csv”, as a Dataframe call it df.

2. Plot the first few rows of data set to get a feeling about the dataset. This can be done using df.head()

```
Lading csv as dataframe df:
      customerID  gender  SeniorCitizen  ... MonthlyCharges TotalCharges  Churn
0     7590-VHVEG  Female              0  ...          29.85        29.85     No
1     5575-GNVDE    Male              0  ...          56.95       1889.5     No
2     3668-QPYBK    Male              0  ...          53.85       108.15    Yes
3     7795-CFOCW    Male              0  ...          42.30      1840.75     No
4     9237-HQITU  Female              0  ...          70.70       151.65    Yes
```

3. Get more information about Dataframe, i.e. data type and missing values using df.info().

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
customerID          7043 non-null object
gender              7043 non-null object
SeniorCitizen       7043 non-null int64
Partner             7043 non-null object
Dependents          7043 non-null object
tenure              7043 non-null int64
PhoneService        7043 non-null object
MultipleLines       7043 non-null object
InternetService     7043 non-null object
OnlineSecurity      7043 non-null object
OnlineBackup        7043 non-null object
DeviceProtection    7043 non-null object
TechSupport         7043 non-null object
StreamingTV         7043 non-null object
StreamingMovies     7043 non-null object
Contract            7043 non-null object
PaperlessBilling    7043 non-null object
PaymentMethod       7043 non-null object
MonthlyCharges      7043 non-null float64
TotalCharges        7043 non-null object
Churn               7043 non-null object
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
None
```

4. Convert “Total Charges” to numeric using the following function:
5. Replace yes and no in the churn column to 1 and 0. This can be done as follows:

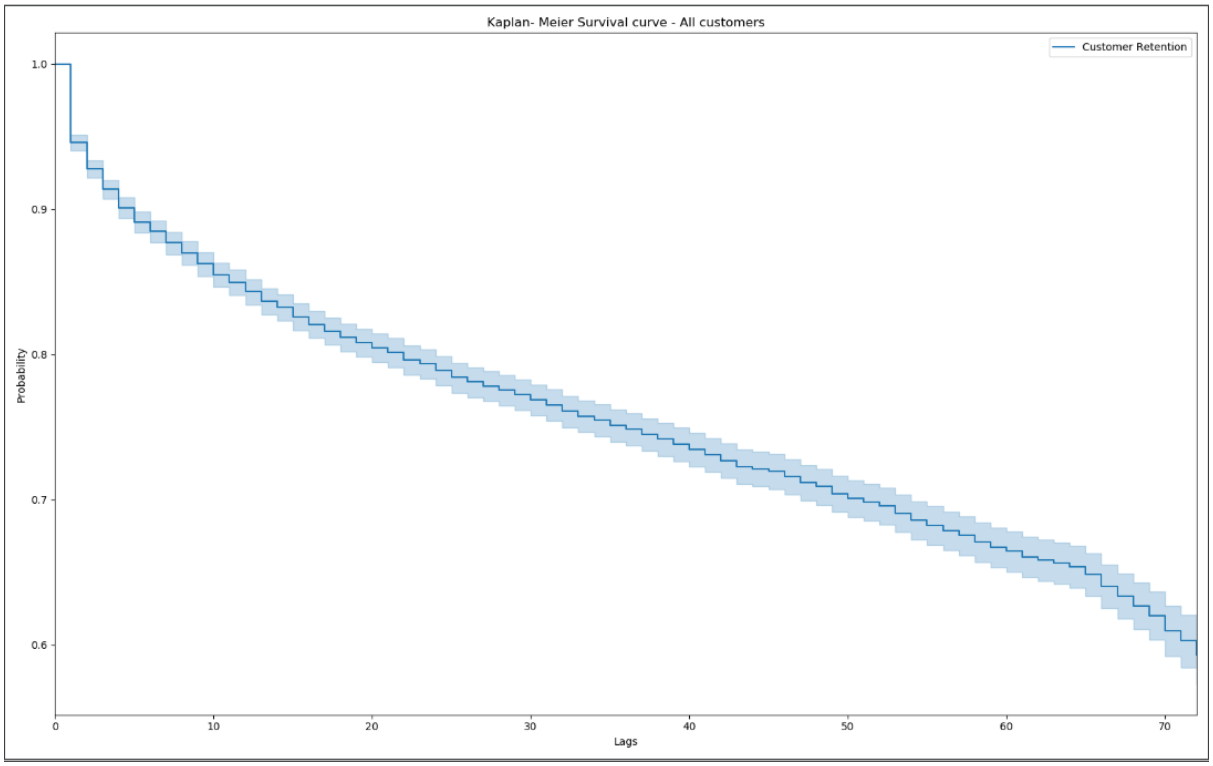
```
      customerID  gender  SeniorCitizen  ... MonthlyCharges TotalCharges  Churn
0     7590-VHVEG  Female              0  ...          29.85        29.85      0
1     5575-GNVDE    Male              0  ...          56.95      1889.50      0
2     3668-QPYBK    Male              0  ...          53.85       108.15      1
3     7795-CFOCW    Male              0  ...          42.30      1840.75      0
4     9237-HQITU  Female              0  ...          70.70       151.65      1
          ...     ...             ...  ...            ...          ...    ...
7038  6840-RESVB    Male              0  ...          84.80      1990.50      0
7039  2234-XADUH  Female              0  ...         103.20      7362.90      0
7040  4801-JZAZL  Female              0  ...          29.60       346.45      0
7041  8361-LTMKD    Male              1  ...          74.40       306.60      1
7042  3186-AJIEK    Male              0  ...         105.65      6844.50      0

[7043 rows x 21 columns]
```

6. Impute the null value of total charges with the median value using the following function:

```
Before Imputing the null value for total charges
: TotalCharges        11
```

```
After Imputing the null value for total charges
: Churn               0
OnlineSecurity      0
gender              0
SeniorCitizen       0
Partner             0
Dependents          0
tenure              0
PhoneService        0
MultipleLines       0
InternetService     0
OnlineBackup        0
TotalCharges        0
DeviceProtection    0
```

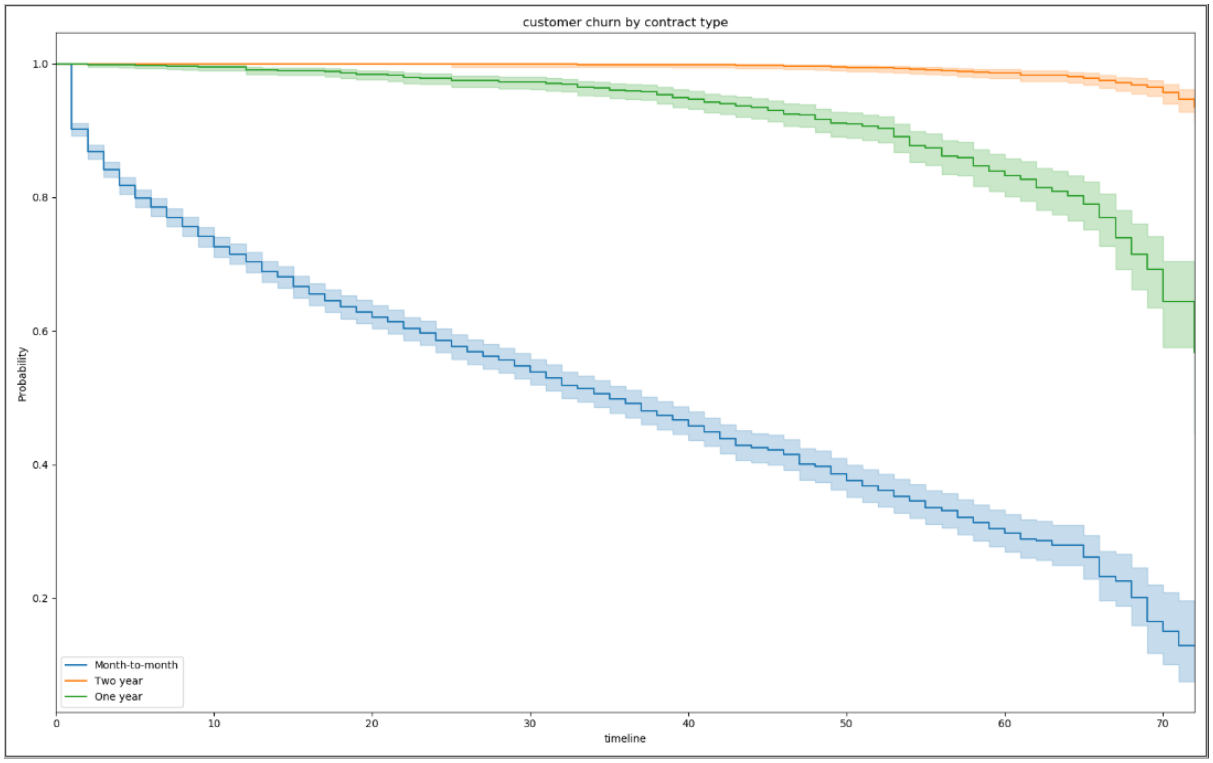
10. Plot the estimated survival curve using:



11. Interpret the plot created in the previous step.

The above should give us some basic intuition about the customers. As we would expect for telecom, churn is relatively low. Even after 72 months, the company is able to retain 60% or more of their customers.

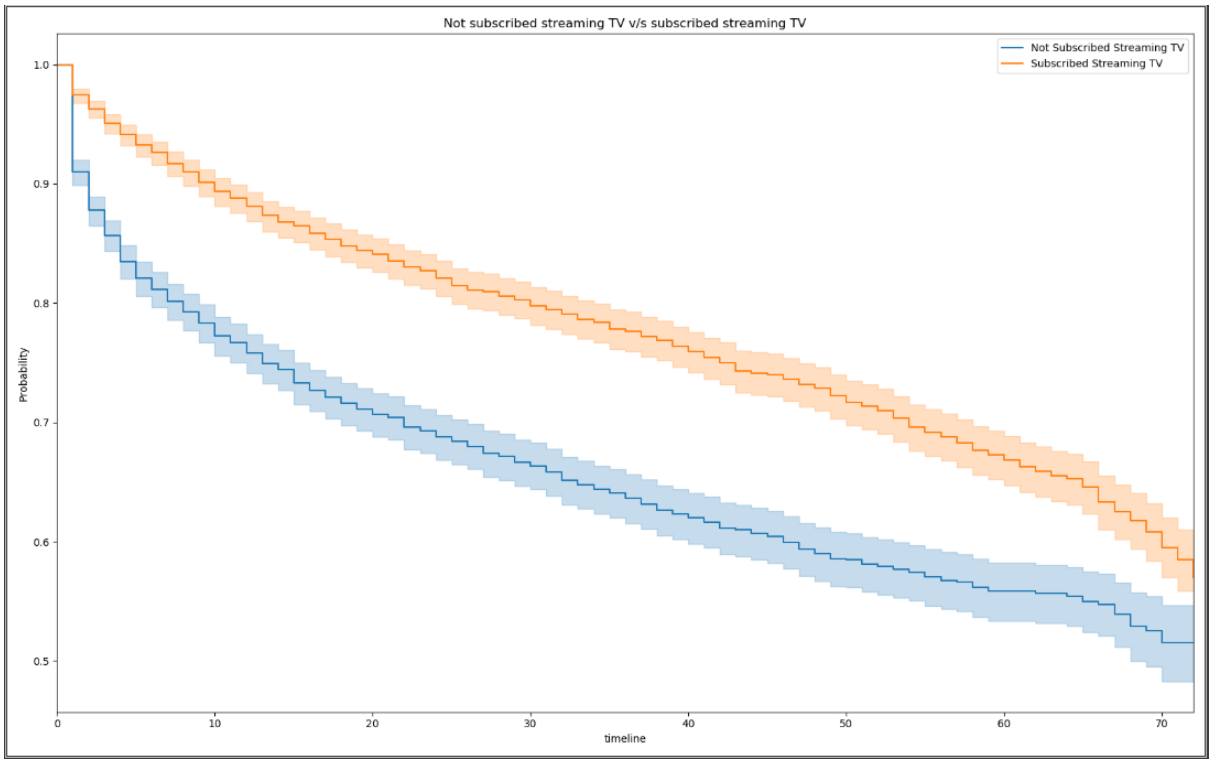
13. - Fit the cohort 1, 2 and 3 data and plot the survival curve using the following commands:



14. - Interpret the plot created in the previous step. How does the length of contract affect retention?

The above should give us some basic intuition about the customer churns by contract type. churn is relatively low for month-month type the company is able to retain about less than 0.2%probability of their customers. The most important feature, by far, is the presence of a 1 or 2 year contract. Customers are .65 and .9, respectively, times as likely to retain their service if they are in 1 and 2 year contract.

17. Repeat the procedures in step 13 to fit the cohorts created in the previous step and plot the estimated survival curve. Make sure to assign the correct labels.



18. Interpret the plot created in the previous step. How is the streaming TV affect retention?

The above should give us some basic intuition about the customers churn who are subscribed to streaming TV and with those who are not subscribed to streaming TV. It looks like the rate is relatively low for those customers that are not subscribed streaming TV vs to those with the subscription.

CONCLUSION

Thus it was estimated and ploted the survival function for real data set using “lifelines” library in Python. Using survival analysis studied the fundamental question How can our telecom company reduce customer churn? We can make recommendations along three dimensions: contract specification, customer selection, and payment systems. To visualize some of our findings, we will fit categorically based Kaplan-Meier curves and plot them, allowing us to see difference in churn rate between customer categories.

CHALLENGE

None

APPENDIX

```
from lifelines import KaplanMeierFitter

import pandas as pd

import lifelines

import matplotlib.pyplot as plt

from pandas.plotting import register_matplotlib_converters


import warnings

warnings.filterwarnings("ignore")

register_matplotlib_converters()


#%%=============================================================================

# 1. Using Pandas library .read_csv function load the “WA_Fn-UseC_-Telco-Customer-Churn.csv”, as

# a Dataframe call it df

# %%-------------------------------------------------------------------------------------------------

df = pd.read_csv('WA_Fn-UseC_-Telco-Customer-Churn.csv')

print("Loading csv as dataframe df:\n", df)


#%%=============================================================================

# 2. Plot the first few rows of data set to get a feeling about the dataset. This can be done using

# df.head()

# %%-------------------------------------------------------------------------------------------------

print(df.head(5))
```

```
#%%========================================================================================

# 3. Get more information about Dataframe, i.e. data type and missing values using df.info()

# %%-------------------------------------------------------------------------------------------------------------

print(df.info())


#%%========================================================================================

# 4. Convert “Total Charges” to numeric using the following function:

# %%-------------------------------------------------------------------------------------------------------------

df['TotalCharges']=pd.to_numeric(df['TotalCharges'],errors='coerce')

#%%========================================================================================

# 5. Replace yes and no in the churn column to 1 and 0. This can be done as follows:

# %%-------------------------------------------------------------------------------------------------------------

df['Churn']=df['Churn'].apply(lambda x: 1 if x == 'Yes' else 0 )

# print(df)


#%%========================================================================================

# 6. Impute the null value of total charges with the median value using the following function:

# %%-------------------------------------------------------------------------------------------------------------

# df = df.isnull().sum().sort_values(ascending = False)

# print('Before Imputing the null value for total charges\n:', df)

df.TotalCharges.fillna(value=df['TotalCharges'].median(),inplace=True)

# df = df.isnull().sum().sort_values(ascending = False)

# print('After Imputing the null value for total charges\n:', df)
```

```
#%%=============================================================================
# 7. Create an overall Kaplan Meier curve, without breaking it into groups of covariates (groups will
# be created in the future steps). For this purpose, you need to create Time to event of censored
# and event data. You also need to create event observed data for customer who has churned (1)
# and censored (0). This can be done as follows:
# %%--------------------------------------------------------------------------------------------------
durations = df['tenure']
event_observed = df['Churn']
#%%=============================================================================
# 8. Create a kmf object as km
# %%--------------------------------------------------------------------------------------------------
km = KaplanMeierFitter()
#%%=============================================================================
# 9. Fit the data into the model
#  km.fit(durations, event_observed,label='Customer Retention')
# %%--------------------------------------------------------------------------------------------------
km.fit(durations, event_observed,label='Customer Retention')
#%%=============================================================================
# 10. Plot the estimated survival curve using:
# %%--------------------------------------------------------------------------------------------------
fig, ax = plt.subplots(figsize=(16,10))
km.plot()
plt.ylabel('Probability')
plt.title("Kaplan- Meier Survival curve - All customers")
plt.show()
```

```
#%%==========================================================================

# 11. Interpret the plot created in the previous step.

# %%-------------------------------------------------------------------------------------------------

''' The above should give us some basic intuition about the customers.

As we would expect for telecom, churn is relatively low. Even after 72 months, the company is able to retain 60% or

more of their customers.

'''

#%%==========================================================================

# 12. Create Kalan Meier curves for three cohorts

# %%-------------------------------------------------------------------------------------------------

kmf = KaplanMeierFitter()

T = df['tenure'] ## time to event

E = df['Churn'] ## event occurred or censored

groups = df['Contract'] ## Create the cohorts from the 'Contract' column

ix1 = (groups == 'Month-to-month') ## Cohort 1

ix2 = (groups == 'Two year') ## Cohort 2

ix3 = (groups == 'One year') ## Cohort 3

#%%==========================================================================

# 13. - Fit the cohort 1, 2 and 3 data and plot the survival curve using the following commands:

# %%-------------------------------------------------------------------------------------------------

fig, ax = plt.subplots(figsize=(16,10))

kmf.fit(T[ix1], E[ix1], label='Month-to-month')

ax = kmf.plot()

kmf.fit(T[ix2], E[ix2], label='Two year')

ax1 = kmf.plot(ax=ax)

kmf.fit(T[ix3], E[ix3], label='One year')

kmf.plot(ax=ax1)
```

```
plt.ylabel('Probability')

plt.title("customer churn by contract type")

plt.show()

#%%==============================================================================

# 14. - Interpret the plot created in the previous step. How does the length of contract affect

# retention?

# %%---------------------------------------------------------------------------------------------

'''

The above should give us some basic intuition about the customer churns by contract type.

churn is relatively low for month-month type the company is able to retain 18% of their customers.

The most important feature, by far, is the presence of a 1 or 2 year contract. Customers are .25 and .02, respectively,

times as likely to cancel their service if they are under contract. Cancellation fees are a possible underlying cause.

As long as these fees do not prohibit new sales, we would recommend continuing to put them into as many contracts as

possible.

'''

#%%==============================================================================

# 15. Add the appropriate legend and title to the graph created in the previous step.

# %%---------------------------------------------------------------------------------------------

'''
Added titles and legends

'''

#%%==============================================================================

# 16. Define two new cohorts based whether a subscriber “StreamingTV” or not “StreamingTV”. We

# would like to know how the streaming TV option affect retention. You can create the cohorts as

# follow:

# %%---------------------------------------------------------------------------------------------
```

```
kmf1 = KaplanMeierFitter()

groups = df['StreamingTV']

i1 = (groups == 'No')

i2 = (groups == 'Yes')

#%%=============================================================================

# 17. Repeat the procedures in step 13 to fit the cohorts created in the previous step and plot the

# estimated survival curve. Make sure to assign the correct labels.

# %%-------------------------------------------------------------------------------------------------

fig, ax = plt.subplots(figsize=(16,10))

kmf1.fit(T[i1], E[i1], label='Not Subscribed Streaming TV')

ax = kmf1.plot()

kmf1.fit(T[i2], E[i2], label='Subscribed Streaming TV')

ax1 = kmf1.plot(ax=ax)

plt.ylabel('Probability')

plt.title("Not subscribed streaming TV v/s subscribed streaming TV")

# plt.figure(figsize=(16,10))

plt.show()

#%%=============================================================================

# 18. Interpret the plot created in the previous step. How is the streaming TV affect retention?

# %%-------------------------------------------------------------------------------------------------

'''

Customers with a partner or dependents are .82 and .91 times as likely to cancel as normal customers.

Families and other large households seem to be less likely to change providers.

This could be due to higher incomes, less time to consider options, or another combination of factors.

'''
```

REFERENCES

https://otexts.com/fpp2/#