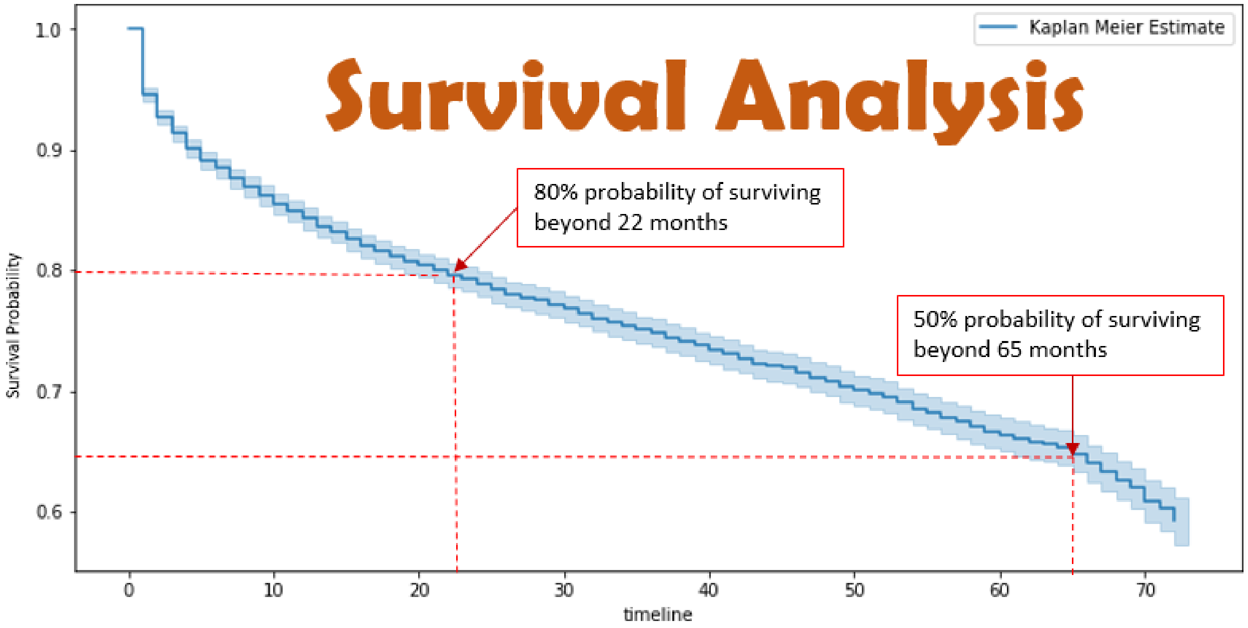
Survival Analysis: Intuition & Implementation in Python

Anurag Pandey Jan 6, 2019 · 15 min read ★



There is a statistical technique which can answer business questions as follows:

- How long will a particular customer remain with your business? In other words, after how much time this customer will churn?
- 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?

If you find any of the above questions (or even the questions remotely related to them) interesting then read on.

The purpose of this article is to build an intuition, so that we can apply this technique in different business settings.

Table of Contents

Introduction

Survival Analysis is a set of statistical tools, which addresses questions such as ‘how long would it be, before a particular event occurs’; in other words we can also call it as a ‘time to event’ analysis. This technique is called survival analysis because this method was primarily developed by medical researchers and they were more interested in finding expected lifetime of patients in different cohorts (ex: Cohort 1- treated with Drug A, & Cohort 2- treated with Drug B). This analysis can be further applied to not just traditional death events, but to many different types of events of interest in different business domains. We will discuss more on the definition of events and time to events in the next section.

Definitions

As mentioned above that the Survival Analysis is also known as Time to Event analysis. Thus, from the name itself, it is evident that the definition of Event of interest and the Time is vital for the Survival Analysis. In order to understand the definition of time and event, we will define the time and event for various use cases in industry.

1. **Predictive Maintenance in Mechanical Operations**: Survival Analysis applies to mechanical parts/ machines to answer about ‘how long will the machine last?’. Predictive Maintenance is one of its applications.

Here, **Event** *is defined as the time at which the machine breaks down.* **Time** *of origin is defined as the time of start of machine for the continuous operations.* Along with the definition of time we should also define **time scale** (time scale could be weeks, days, hours..). The difference between the time of event and the time origin gives us the time to event.

2. **Customer Analytics** (Customer Retention): With the help of Survival Analysis we can focus on churn prevention efforts of high-value customers with low survival time. This analysis also helps us to calculate Customer Life Time Value. In this use case, **Event** *is defined as the time at which the customer churns / unsubscribe.* **Time** *of origin* is defined as the time at which the customer starts the service/subscription with a company. **Time scale** could be months, or weeks. The difference between the time of event and the time origin gives us the time to event.

3. **Marketing Analytics** (Cohort Analysis): Survival Analysis evaluates the retention rates of each marketing channel. In this use case, **Event** *is defined as the time at which the customer unsubscribe a marketing channel.* **Time** *of origin* is defined as the time at which the customer starts the service / subscription of a marketing channel. **Time scale** could be months, or weeks.

4. **Actuaries**: Given the risks of a population, survival analysis evaluates the probability of the population to die in a particular time range. This

analysis helps the insurance companies to evaluate the insurance premiums. Guess, the **event** and **time** definition for this use case!!!

I hope the definition of a event, time origin, and time to event is clear from the above discussion. Now its time to delve a bit deeper into the mathematical formulation of the analysis.

Mathematical Intuition

Lets assume a **non-negative continuous random variable T**, representing the time until some event of interest. For example, T might denote:

- the time from the customer's subscription to the customer churn.
- the time from start of a machine to its breakdown.
- the time from diagnosis of a disease until death.

Since we have assumed a random variable T (a random variable is generally represented in capital letter), so we should also talk about some of its attributes.

T is a random variable, 'what is random here ?'. To understand this we will again use our earlier examples as follows.

- T is the time from customer’s(a **randomly selected** customer) subscription to the customer churn.
- T is the time from start of a **randomly selected** machine to its breakdown.
- T is the time from diagnosis of a disease until death of a **randomly selected** patient.

T is continuous random variable, therefore it can take any real value. **T is non-negative**, therefore it can only take positive real values (0 included).

For such random variables, **probability density function (pdf)** and **cumulative distribution function (cdf)** are commonly used to characterize their distribution.

Thus, we will assume that this random variable has a probability density function **f(t)** , and cumulative distribution function **F(t)** .

pdf : f(t)

cdf : F(t) : As per the definition of cdf from a given pdf, we can define cdf as **F(t) = P (T< t) ;** here , **F(t)** gives us the probability that the event has occurred by duration **t.** In simple words, **F(t)** gives us the proportion of population with the time to event value less than **t.**

$$\int_0^t f(x)dx$$

cdf as the integral form of pdf

Survival Function: S(t) = 1 - F(t)= P(T ≥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.**

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

Survival Function in integral form of pdf

Hazard Function : h(t) : 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 time t. In medical terms, we can define it as “out of the people who survived at time t, what is the rate of dying of those people”.

Lets make it even more simpler:

1. Lets write it in the form of its definition:

$h(t) = [(S(t) - S(t + dt))/dt] / S(t)$

limit $dt \to 0$

2. From its formulation above we can see that it has two parts. Lets understand each part

Instantaneous rate of event: $(S(t) - S(t + dt))/dt$; this can also be seen as the slope at any point t of the Survival Curve, or the rate of dying at any time t.

Also lets assume the total population as P.

Here, $S(t) - S(t + dt)$, this difference gives proportion of people died in time dt, out of the people who survived at time t. Number of people surviving at t is S(t)*P and the number of people surviving at t+dt is S(t+dt)*P. Number of people died during dt is (S(t) -S(t + dt))*P. Instantaneous rate of people dying at time t is (S(t) -S(t + dt))*P/dt.

Proportion Surviving at time t: S(t); We also know the surviving population at time t, S(t)*P.

Thus dividing number of people died in time dt, by the number of people survived at any time t, gives us the hazard function as measure of RISK of the people dying, which survived at the time t.

The hazard function is not a density or a probability. However, we can think of it as the probability of failure in an infinitesimally small time period between (t) and (t+ dt) given that the subject has survived up till time t. In this sense, **the hazard is a measure of risk: the greater the hazard between times t1 and t2, the greater the risk of failure in this time interval.**

We have : h(t) = f(t)/S(t) ; [Since we know that (S(t) -S(t + dt))/dt = f(t)] This is a very important derivation. The beauty of this function is that Survival function can be derived from Hazard function and vice versa. The utility of this will be more evident while deriving a survival function from a given hazard function in Cox Proportional Model (Last segment of the article).

These were the most important mathematical definitions and the formulations required to understand the survival analysis. We will end our mathematical formulation here and move forward towards estimation of survival curve.

Kaplan-Meier Estimate

In the Mathematical formulation above we assumed the pdf function and thereby derived Survival function from the assumed pdf function. Since we don't have the true survival curve of the population, thus we will estimate the survival curve from the data.

There are two main methods to estimate the survival curve. The first method is a parametric approach. This method assumes a parametric model, which is based on certain distribution such as exponential distribution, then we estimate the parameter, and then finally form the estimator of the survival function. A second approach is a powerful **non-parametric method called the Kaplan-Meier estimator.** We will discuss it in this section. In this section we will also try to create the Kaplan-Meier curve manually as well as by using the Python library (lifelines).

$$\hat{S}(t) = \prod_{i:\ t_i \leq t} \frac{n_i - d_i}{n_i}$$

Here, **ni** is defined as the population at risk at time just prior to time **ti;** and **di** is defined as number of events occurred at time **ti**. This, will become more clear with the example below.

We will discuss an arbitrary example from a very small self created data, to understand the creation of Kaplan Meier Estimate curve, manually as well as using a python package.

Event, Time and Time Scale Definition for the Example:

The example below(Refer Fig 1) shows the data of 6 users of a website. These users visit the website and leaves that website after few minutes. Thus, **event of interest** is the time in which a user leaves the website. **Time of origin** is defined as the time of opening the website by a user and the

time scale is in minutes. The study starts at time t=0 and ends at time t=6 minutes.

Censorship:

Point worth noting here is that during the study period , event happened with 4 out of 6 users(shown in red), while two users (shown in green) continued and the event didn't happened till the end of the study; such data is called the **Censored data.**

In case of **censorship**, as here in case of user 4 and user 5, we don't know at what time the event will occur, but still we are using that data to estimate the probability of survival. If we choose not to include the censored data, then it is highly likely that our estimates would be highly biased and under-estimated. The inclusion of censored data to calculate the estimates, makes the Survival Analysis very powerful, and it stands out as compared to many other statistical techniques.

Calculations for KM Curve and the interpretation:

Now, lets talk about the calculations done to create the KM Curve below (Refer Fig 1). In figure 1, Kaplan Meier Estimate curve, x axis is the time of event and y axis is the estimated survival probability.

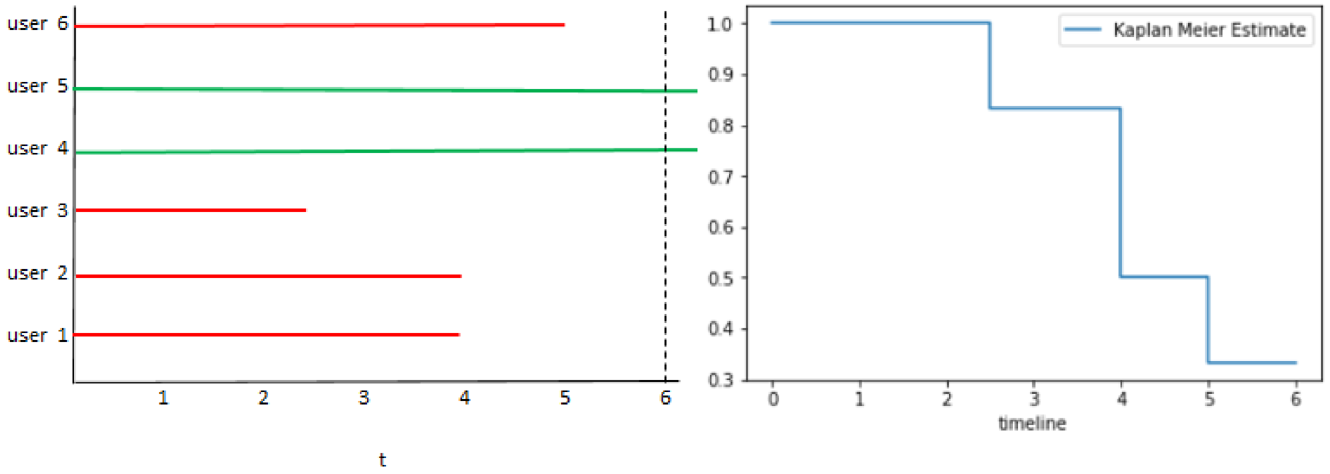
From t=0 till t<2.5 or t∈[0 , 2.5), number of users at risk(ni) at time t=0 is 6 and number of events occurred(di) at time t=0 is 0, therefore for all t in this interval, estimated S(t) = 1. From the definition of the event we can say that 100% is the probability that the time between a user opens the website and exit the website is greater than 2.499* minutes.

From t=2.5 till t<4 or t ∈ [2.5 , 4), number of users at risk(ni) at time just before time 2.5 minutes (2.4999* mins) is 6 and number of events occurred(di) at time t=2.5 minutes is 1, therefore therefore for all t in this interval, estimated S(t)= 0.83. From the definition of the event we can say that 83% is the probability that the time between a user opens the website and exit the website is greater than 3.999* minutes.

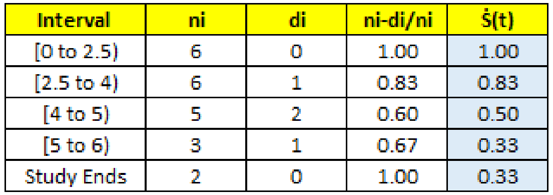
From t=4 till t<5 or t ∈[4 , 5), number of users at risk(ni) at time just before time 4 minutes (3.999* mins) is 5 and number of events occurred(di) at time t=4 minutes is 2, therefore for all t in this interval, estimated S(t) = 0.5. *This result can also be verified by simple mathematics of relative frequency. For any t∈[4,5), lets say t=4.5, total number of users at the start were 6, total number remaining at t are 3. Therefore, the probability of* *the users spending more than 4.5 (or any time t ∈[4,5)) minutes on website is (3/6), which is 50%.*

Similarly, we can estimate the probability for other time intervals (refer table calculations in fig 1)

Mathematically, for any time t ∈ [t1, t2), we have
S(t) = P(survive in [0, t1)) × P(survive in [t1, t] | survive in [0, t1))



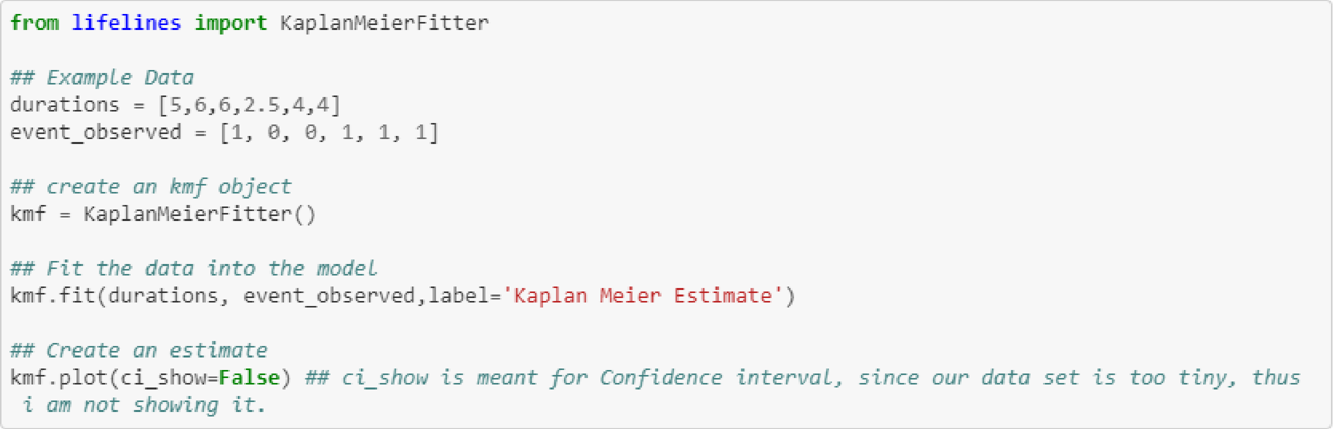
$$\hat{S}(t) = \prod_{i:\, t_i \le t} \frac{n_i - d_i}{n_i}$$



Interval	ni	di	ni-di/ni	Ŝ(t)
[0 to 2.5)	6	0	1.00	1.00
[2.5 to 4)	6	1	0.83	0.83
[4 to 5)	5	2	0.60	0.50
[5 to 6)	3	1	0.67	0.33
Study Ends	2	0	1.00	0.33

fig 1: a. Shows the user level time data in color.; b. Shows Kaplan Meier (KM)Estimate Curve; c. Formula for estimation of KM curve; d. Table showing the calculations

```
# Python code to create the above Kaplan Meier curve

from lifelines import KaplanMeierFitter

## Example Data
durations = [5,6,6,2.5,4,4]
event_observed = [1, 0, 0, 1, 1, 1]

## create a kmf object
kmf = KaplanMeierFitter()

## Fit the data into the model
kmf.fit(durations, event_observed,label='Kaplan Meier Estimate')

## Create an estimate
kmf.plot(ci_show=False) ## ci_show is meant for Confidence interval,
since our data set is too tiny, thus i am not showing it.
```



```
from lifelines import KaplanMeierFitter

## Example Data
durations = [5,6,6,2.5,4,4]
event_observed = [1, 0, 0, 1, 1, 1]

## create an kmf object
kmf = KaplanMeierFitter()

## Fit the data into the model
kmf.fit(durations, event_observed,label='Kaplan Meier Estimate')

## Create an estimate
kmf.plot(ci_show=False) ## ci_show is meant for Confidence interval, since our data set is too tiny, thus
 i am not showing it.
```

Real World Example:

As mentioned earlier that Survival Analysis can be used for the cohort analysis, to gain insights. So, here we will be using the Telco-Customer-Churn data set, to gain insight about the lifelines of customers in different cohorts.

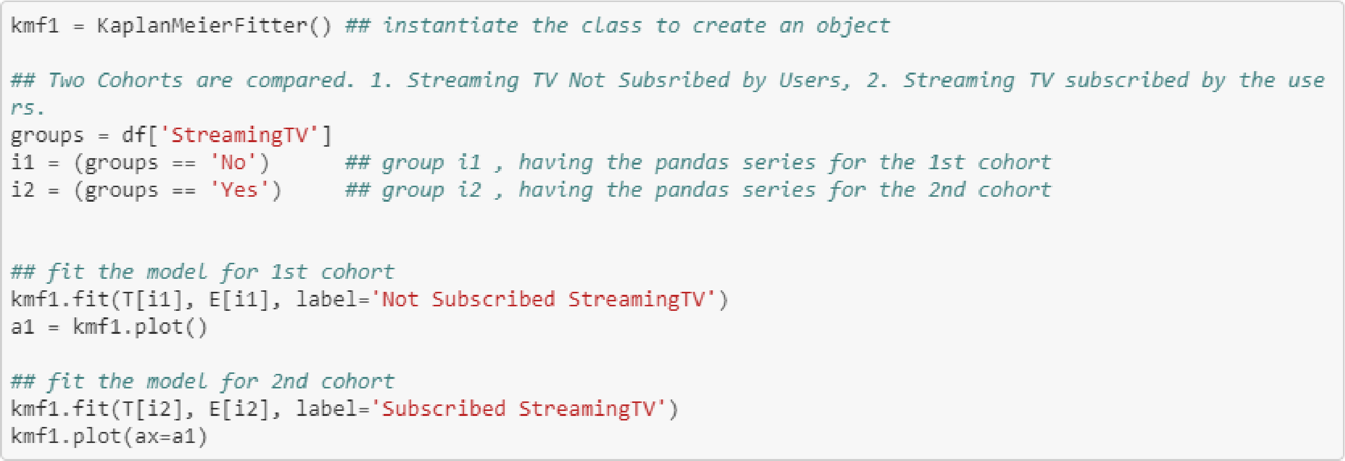
Github link for the code: Link

Lets create two cohorts of customers based on whether a customer has subscribed for Streaming TV or not. We want to know that which cohort has the better customer retention.

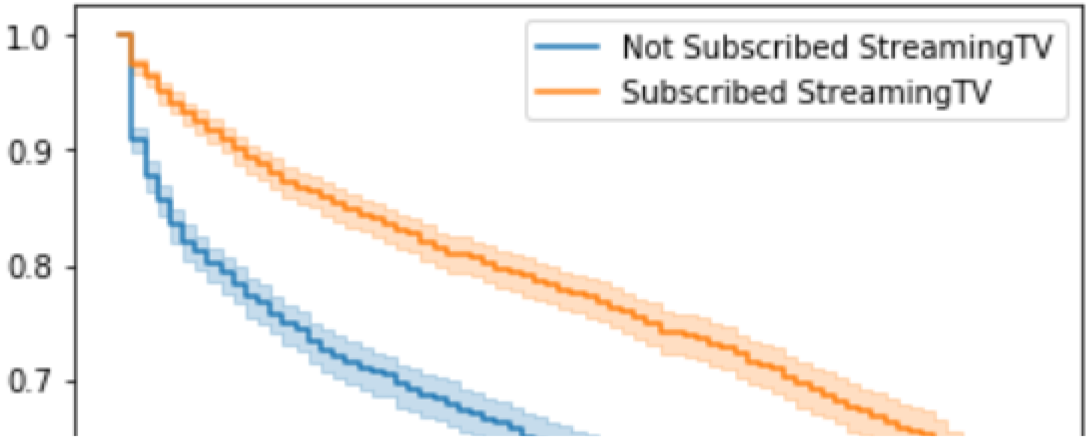
The required code for plotting the Survival Estimates is given below.

```
kmf1 = KaplanMeierFitter() ## instantiate the class to create an
object

## Two Cohorts are compared. Cohort 1. Streaming TV Not Subscribed by
users, and Cohort  2. Streaming TV subscribed by the users.


groups = df['StreamingTV']
i1 = (groups == 'No')      ## group i1 , having the pandas series
for the 1st cohort
i2 = (groups == 'Yes')     ## group i2 , having the pandas series
for the 2nd cohort


## fit the model for 1st cohort
```

```
kmf1.fit(T[i1], E[i1], label='Not Subscribed StreamingTV')
a1 = kmf1.plot()

## fit the model for 2nd cohort
kmf1.fit(T[i2], E[i2], label='Subscribed StreamingTV')
kmf1.plot(ax=a1)
```



```
kmf1 = KaplanMeierFitter() ## instantiate the class to create an object

## Two Cohorts are compared. 1. Streaming TV Not Subsribed by Users, 2. Streaming TV subscribed by the users.
groups = df['StreamingTV']
i1 = (groups == 'No')      ## group i1 , having the pandas series for the 1st cohort
i2 = (groups == 'Yes')     ## group i2 , having the pandas series for the 2nd cohort


## fit the model for 1st cohort
kmf1.fit(T[i1], E[i1], label='Not Subscribed StreamingTV')
a1 = kmf1.plot()

## fit the model for 2nd cohort
kmf1.fit(T[i2], E[i2], label='Subscribed StreamingTV')
kmf1.plot(ax=a1)
```



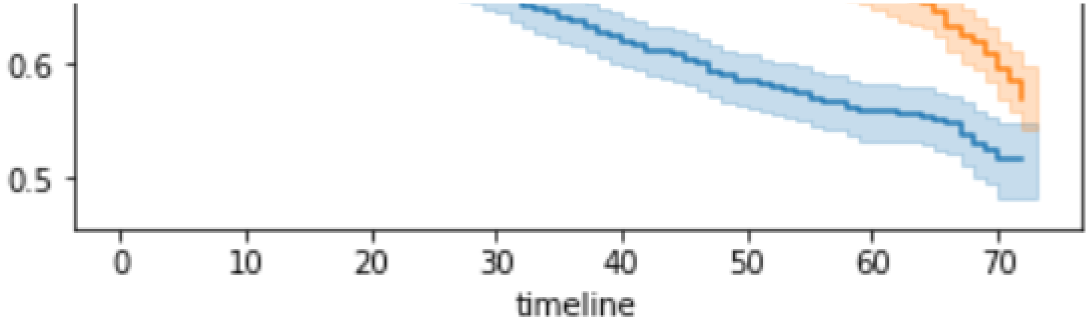


Fig 2: Kaplan Meier Curve of the two cohorts.

We have two survival curves , one for each cohort. From the curves, it is evident that the customers, who have subscribed for the Streaming TV, have better customer retention as compared to the customers, who have not subscribed for the Streaming TV. At any point t across the timeline, we can see that the survival probability of the cohort in blue is less than the cohort in red. For the cohort in blue, the survival probability is decreasing with high rate in first 10 months and it gets relatively better after that; however, for the red cohort, the rate of decrease in survival rate is fairly constant. Therefore, for the cohort , which has not subscribed for the Streaming TV, efforts should be made to retain the customers in first 10 volatile months.

We can do more such cohort analysis from the survival curves of the different cohorts. This cohort analysis represents the limited use case of the potential of the survival analysis because we are using it for the aggregated level of the data. We can create the Survival Curves for even the individual users based on the effects of covariates on the baseline Survival Curves. This will be our focal point of the next section of this article.

Cox Proportional Hazard Model

The time to event for an individual in the population is very important for the survival curves at the aggregate level; however, in real life situations along with the event data we also have the covariates (features) of that individual. In such cases, it is very important to know about the impact of covariates on the survival curve. This would help us in predicting the survival probability of an individual, if we know the associated covariates values.

For example, in the telco-churn example discussed above, we have each customer's tenure when they churned (the event time T) and the customer's Gender, MonthlyCharges, Dependants, Partner, PhoneService etc. The other variables are the covariates in this example. We are often interested in how these covariates impacts the survival probability function.

In such cases, it is the conditional survival function $S(t|x) = P(T > t|x)$. Here x denotes the covariates. In our example, we are interested in S(tenure

> t|(Gender, MonthlyCharges, Dependants, Partner, PhoneService etc)).

The Cox (proportional hazard) model is one of the most popular model combining the covariates and the survival function. It starts with modeling the hazard function.

$$h(t|X = x) = h_0(t) \exp(x^T \beta)$$

Here, β is the vector of coefficients of each covariate. The function h*o*(t) is called the baseline hazard function.

The Cox model assumes that the covariates have a linear multiplication effect on the hazard function and the effect stays the same across time.

The idea behind the model is that the log-hazard of an individual is a linear function of their static covariates, *and* a population-level baseline hazard that changes over time. [Source: lifelines documentation]

From the above equation we can also derive cumulative conditional hazard function as below:

$$H(t|x) = \exp(x^T\beta)\int_0^t h_0(s)ds = \exp(x^T\beta)H_0(t)$$

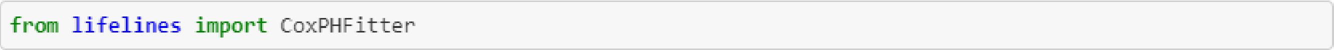
As we are already aware that we can derive survival function from the hazard function with the help of expression derived in above section. Thus, we can get the survival function for each subject/individual/customer.

Basic implementation in python:

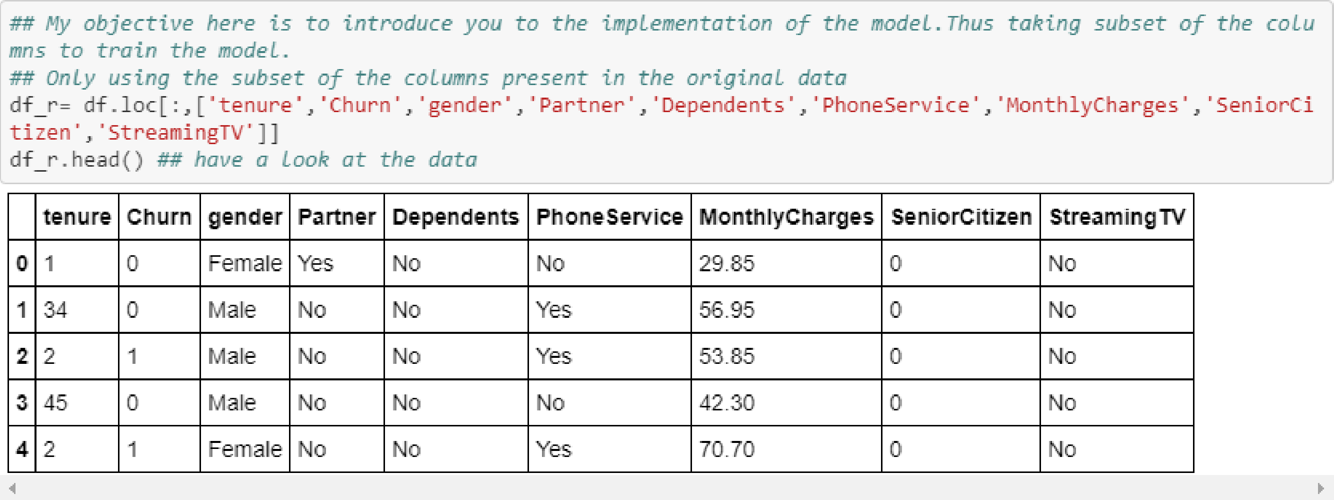
We will now discuss about its basic implementation in python with the help of lifelines package. We have used the same telco-customer-churn data-set, which we have been using in the above sections. We will run a python code for predicting the survival function at customer level.

```
from lifelines import CoxPHFitter

## My objective here is to introduce you to the implementation of the
model.Thus taking subset of the columns to train the model.
## Only using the subset of the columns present in the original data
```

```
df_r= df.loc[:['tenure', 'Churn', 'gender', 'Partner', 'Dependents',
'PhoneService','MonthlyCharges','SeniorCitizen','StreamingTV']]


df_r.head() ## have a look at the data
```



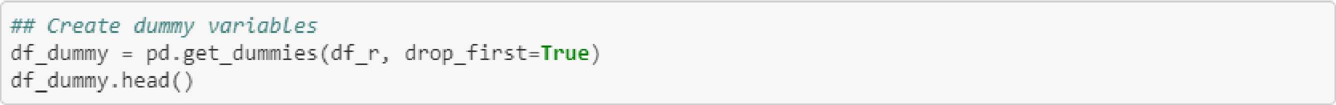
```
from lifelines import CoxPHFitter
```



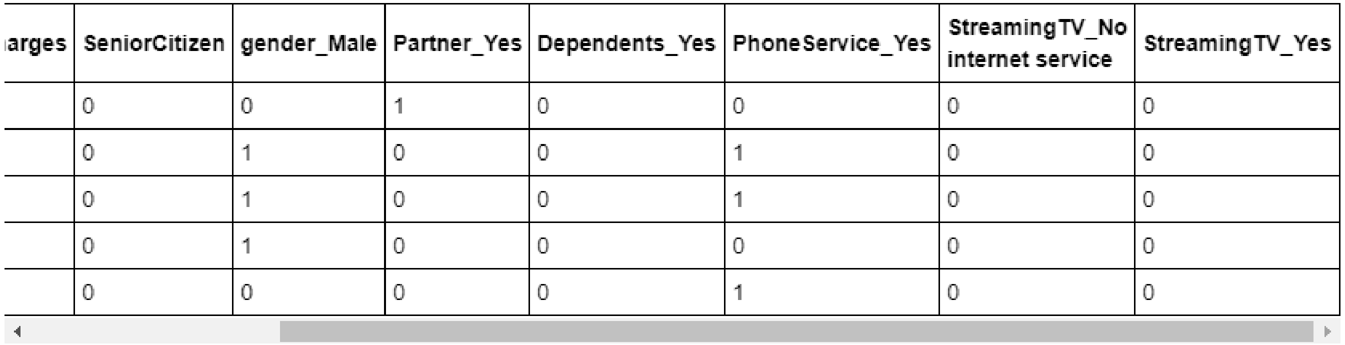
```
## My objective here is to introduce you to the implementation of the model.Thus taking subset of the colu
mns to train the model.
## Only using the subset of the columns present in the original data
df_r= df.loc[:,['tenure','Churn','gender','Partner','Dependents','PhoneService','MonthlyCharges','SeniorCi
tizen','StreamingTV']]
df_r.head() ## have a look at the data
```

	tenure	Churn	gender	Partner	Dependents	PhoneService	MonthlyCharges	SeniorCitizen	StreamingTV
0	1	0	Female	Yes	No	No	29.85	0	No
1	34	0	Male	No	No	Yes	56.95	0	No
2	2	1	Male	No	No	Yes	53.85	0	No
3	45	0	Male	No	No	No	42.30	0	No
4	2	1	Female	No	No	Yes	70.70	0	No

```
## Create dummy variables
df_dummy = pd.get_dummies(df_r, drop_first=True)
df_dummy.head()
```

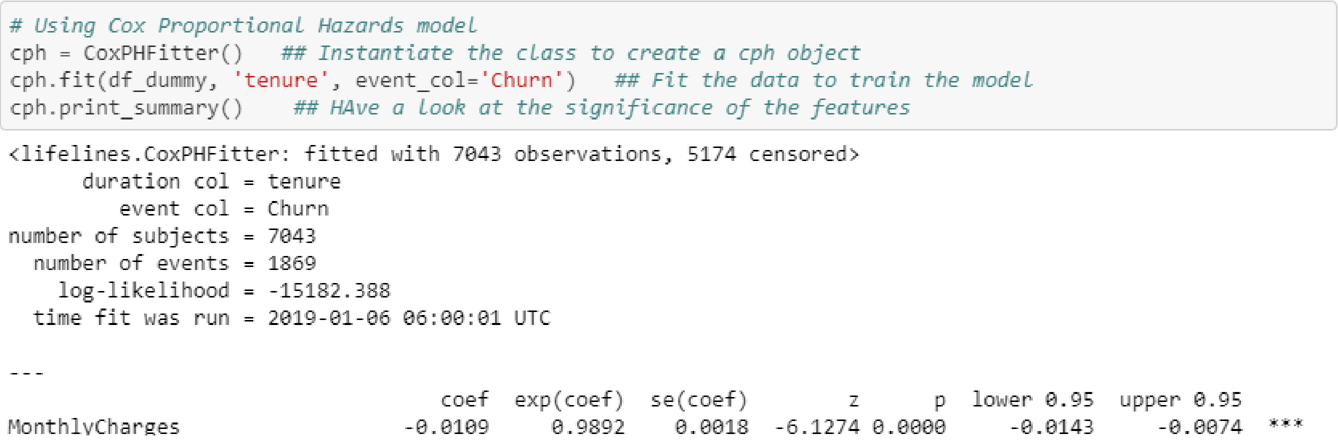


```
## Create dummy variables
df_dummy = pd.get_dummies(df_r, drop_first=True)
df_dummy.head()
```



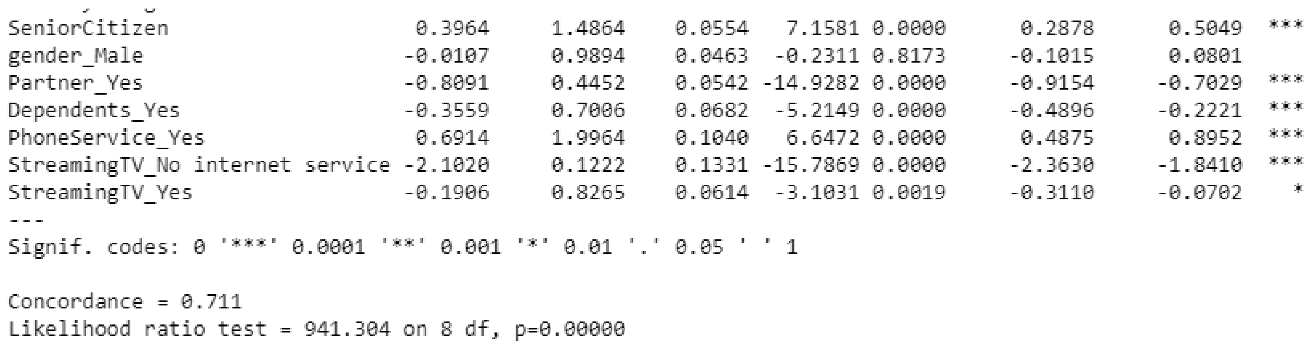
arges	SeniorCitizen	gender_Male	Partner_Yes	Dependents_Yes	PhoneService_Yes	StreamingTV_No internet service	StreamingTV_Yes
	0	0	1	0	0	0	0
	0	1	0	0	1	0	0
	0	1	0	0	1	0	0
	0	1	0	0	0	0	0
	0	0	0	0	1	0	0

```
# Using Cox Proportional Hazards model
cph = CoxPHFitter()   ## Instantiate the class to create a cph object

cph.fit(df_dummy, 'tenure', event_col='Churn')   ## Fit the data to
train the model

cph.print_summary()    ## HAve a look at the significance of the
features
```

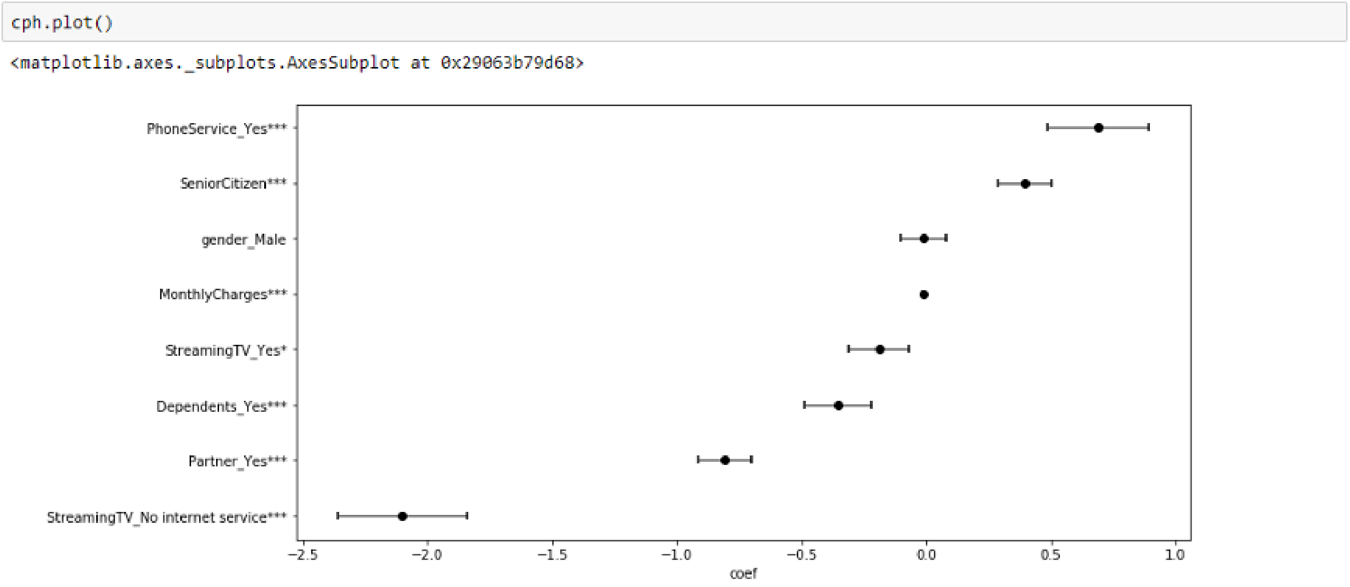


```
# Using Cox Proportional Hazards model
cph = CoxPHFitter()   ## Instantiate the class to create a cph object
cph.fit(df_dummy, 'tenure', event_col='Churn')   ## Fit the data to train the model
cph.print_summary()    ## HAve a look at the significance of the features
```

```
<lifelines.CoxPHFitter: fitted with 7043 observations, 5174 censored>
      duration col = tenure
         event col = Churn
number of subjects = 7043
  number of events = 1869
    log-likelihood = -15182.388
  time fit was run = 2019-01-06 06:00:01 UTC

---
                                  coef  exp(coef)  se(coef)       z      p  lower 0.95  upper 0.95
MonthlyCharges                 -0.0109     0.9892    0.0018 -6.1274 0.0000     -0.0143     -0.0074  ***
```



```
SeniorCitizen                      0.3964     1.4864    0.0554   7.1581 0.0000      0.2878      0.5049  ***
gender_Male                       -0.0107     0.9894    0.0463  -0.2311 0.8173     -0.1015      0.0801
Partner_Yes                       -0.8091     0.4452    0.0542 -14.9282 0.0000     -0.9154     -0.7029  ***
Dependents_Yes                    -0.3559     0.7006    0.0682  -5.2149 0.0000     -0.4896     -0.2221  ***
PhoneService_Yes                   0.6914     1.9964    0.1040   6.6472 0.0000      0.4875      0.8952  ***
StreamingTV_No internet service -2.1020     0.1222    0.1331 -15.7869 0.0000     -2.3630     -1.8410  ***
StreamingTV_Yes                   -0.1906     0.8265    0.0614  -3.1031 0.0019     -0.3110     -0.0702    *
---
Signif. codes: 0 '***' 0.0001 '**' 0.001 '*' 0.01 '.' 0.05 ' ' 1

Concordance = 0.711
Likelihood ratio test = 941.304 on 8 df, p=0.00000
```



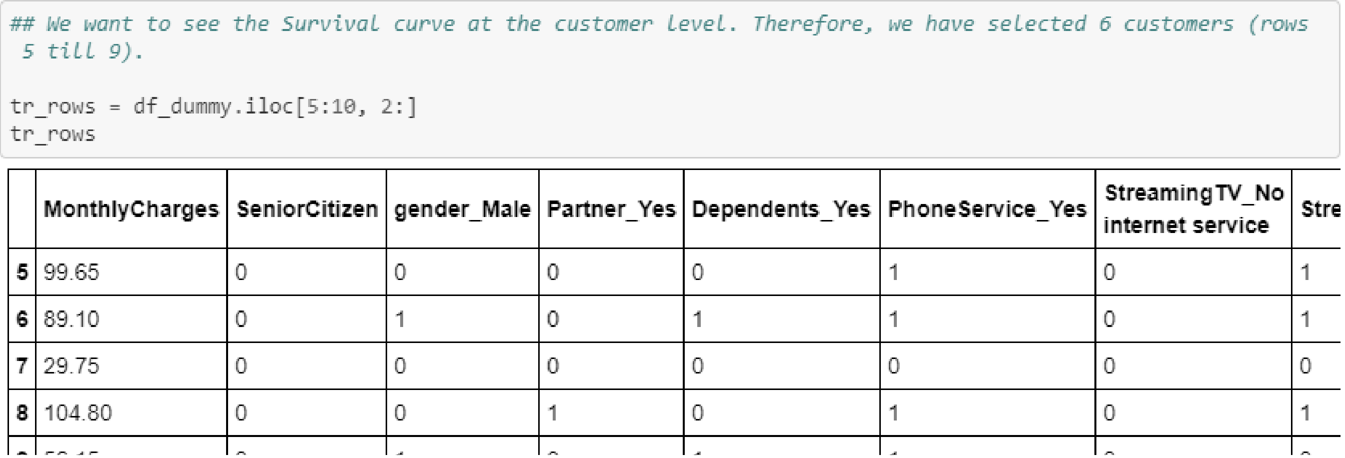
```
cph.plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x29063b79d68>
```

The summary statistics above indicates the significance of the covariates in predicting the churn risk. Gender doesn't play any significant role in predicting the churn, whereas all the other covariates are significant.

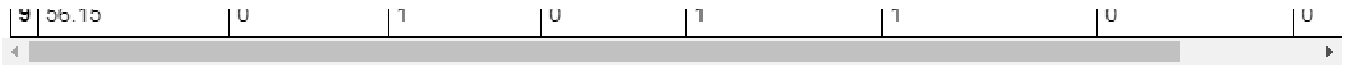
Interesting point *to note here is that , the β (coef) values in case of covariates MonthlyCharges and gender_Male is approximately zero (~-0.01), but still the MonthlyCharges plays a significant role in predicting churn , while the latter is insignificant. The reason is that the MonthlyCharges is continuous value and it can vary from the order of tens, hundreds to thousands, when multiplied by the small coef (β=-0.01), it becomes significant. On the other hand, the covariate gender can only take the value 0 or 1, and in both the cases [exp(-0.01 * 0), exp(-0.01*1)] it will be insignificant.*

```
## We want to see the Survival curve at the customer level.
Therefore, we have selected 6 customers (rows 5 till 9).

tr_rows = df_dummy.iloc[5:10, 2:]
tr_rows
```



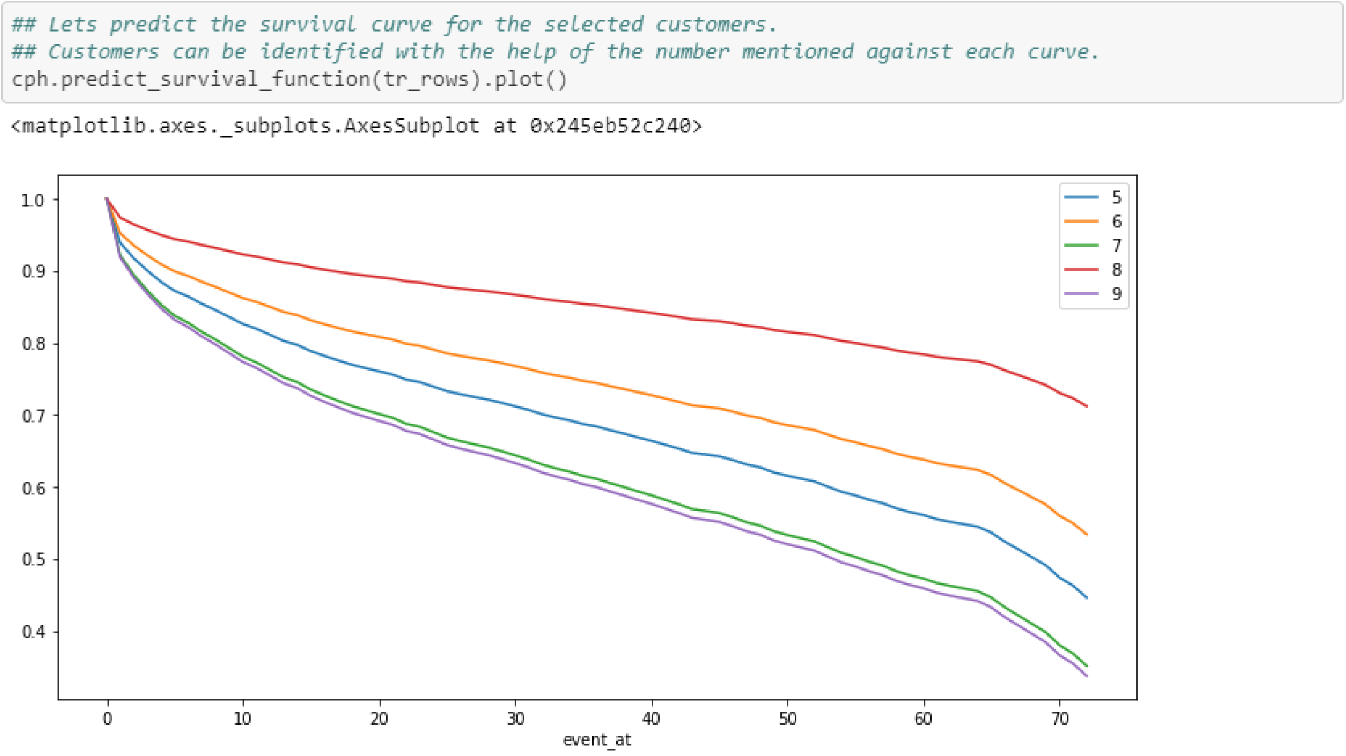
```
## We want to see the Survival curve at the customer level. Therefore, we have selected 6 customers (rows
 5 till 9).

tr_rows = df_dummy.iloc[5:10, 2:]
tr_rows
```

	MonthlyCharges	SeniorCitizen	gender_Male	Partner_Yes	Dependents_Yes	PhoneService_Yes	StreamingTV_No internet service	Stre
5	99.65	0	0	0	0	1	0	1
6	89.10	0	1	0	1	1	0	1
7	29.75	0	0	0	0	0	0	0
8	104.80	0	0	1	0	1	0	1



9	56.15	0	1	0	1	1	0	0

```
## Lets predict the survival curve for the selected customers.
## Customers can be identified with the help of the number mentioned
against each curve.

cph.predict_survival_function(tr_rows).plot()
```



```
## Lets predict the survival curve for the selected customers.
## Customers can be identified with the help of the number mentioned against each curve.
cph.predict_survival_function(tr_rows).plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x245eb52c240>
```

fig 2. It shows the Survival Curves at customer level of customer number 5,6,7,8, and 9

Fig 2 . shows the survival curves at customer level. It shows the survival curves for customer number 5,6,7,8, & 9.

Creating the survival curves at each customer level helps us in proactively creating a tailor made strategy for high-valued customers for different survival risk segments along the timeline.

End Note

Though, there are many other things which are still remaining to be covered in survival analysis such as 'checking proportionality assumption', & 'model selection' ; however, with a basic understanding of the mathematics behind the analysis, and the basic implementation of the survival analysis (using the lifelines package in python) will help us in implementing this model in any pertinent business use case.

Additional Resources

The following resources were extremely helpful not only in motivating me to study the survival analysis but also in making this article. Check them out for more on survival analysis.

1. Lifelines Python Documentation

2. SciPy 2015 lecture by Allen Downey

3. IPPCR 2015: Conceptual Approach to Survival Analysis

4. Nonparametric Statistics by Yen-Chi Chen

5. Princeton University Lecture Notes: Survival Models

Python Data Science Probability Analytics Statistical Analysis