The **Kaplan-Meier test**, more commonly referred to as the **Kaplan-Meier estimator**, is a non-parametric statistic used to estimate the survival function from lifetime data. The Kaplan-Meier estimator provides a way to measure the fraction of subjects surviving for a certain amount of time after treatment or intervention. The Kaplan-Meier estimator produces a step function that visualizes the survival probability over time.
The survival function, represents the probability that a subject will survive beyond time. It is defined as: $\mathbf{S(t)=P(T>t)}$ where T is the time until the event occurs.

$$S(t) = \prod_{t_i \leq t} \left(1 - \frac{d_i}{n_i}\right)$$

, where t_i = time at which an event occurs, d_i = number of events at time t_i, n_i = number of subjects at risk just before time t_i

The curve drops at each time point where an event occurs, and the size of the drop depends on the number of events and the number of subjects at risk at that time.

The **Kaplan-Meier estimator (KMF)** is a non-parametric method used to estimate the survival function from lifetime data. In the context of customer churn analysis, it helps model the probability of customers staying with a company over time. This analysis evaluates the KMF model using two key metrics:

- **Concordance Index (C-Index)** – measures the model's predictive accuracy.
- **Brier Score** – assesses the calibration of predicted probabilities.

The dataset is split into **training (70%)** and **testing (30%)** subsets to assess the model's generalization performance. This ensures that evaluation metrics are computed on unseen data.
The **Kaplan-Meier Fitter** is trained on the tenure (time until churn) and churn status (event occurrence). Predictions are made for the test dataset, providing the probability of a customer staying subscribed at given time points.
The **churn probability** is then computed as **(1 - survival probability)**.

The **C-Index** is a measure of predictive accuracy for survival models. It evaluates whether the model correctly ranks survival times. A **C-Index of 0.5** indicates random predictions, while **values closer to 1** indicate strong predictive ability.

$$\text{Brier Score} = \frac{1}{N}\sum_{i=1}^{N}(\hat{S}_i - O_i)^2$$

.

S^i = Predicted survival probability from KM.
Oi = Actual outcome (1 if churned, 0 if not).
N = Number of test samples.

Next Steps :

How to evaluate accuracy of this model?

- How to use this survival curve to predict probabilty of customer leaving after n months (e.g. 3 months)?

```python
from lifelines import KaplanMeierFitter
from lifelines.utils import concordance_index
from sklearn.metrics import brier_score_loss
from sklearn.model_selection import train_test_split

# Split data into train/test sets
train, test = train_test_split(data, test_size=0.3, random_state=42)

# Fit Kaplan-Meier model on training data
kmf_train = KaplanMeierFitter()
kmf_train.fit(train["tenure"], event_observed=train["Churn"])

# Generate predictions for test set
test_times = test["tenure"].values
test_events = test["Churn"].values

# Predict survival probabilities for each customer in the test set
survival_probs = kmf_train.predict(test_times)

# Convert survival probabilities to risk scores (1 - survival probability)
risk_scores = 1 - survival_probs

# Compute Concordance Index (using risk scores)
c_index = concordance_index(
    test_times,
    risk_scores,
    test_events
)

print(f"Kaplan-Meier Concordance Index: {c_index:.4f}")
```

The **Brier Score** measures the accuracy of probability predictions, reflecting the mean squared difference between predicted probabilities and actual outcomes.
A lower **Brier Score (0 to 1)** indicates better calibration.
It penalizes both **misclassification** and **uncertainty in predictions**.

Metrics evaluated:

```
Kaplan-Meier Concordance Index (Test Set): 0.9886
Kaplan-Meier Brier Score (Test Set): 0.2500
Full Dataset Concordance Index: 0.9888
Kaplan-Meier Brier Score (Full Dataset): 0.2435
```

```
# Calculate Brier Score for the test set
churn_probs = 1 - survival_probs
brier_test = brier_score_loss(test_events, churn_probs)
print(f"Kaplan-Meier Brier Score (Test Set): {brier_test:.4f}")

# Evaluate on the entire dataset for comparison
kmf_full = KaplanMeierFitter()
kmf_full.fit(data["tenure"], event_observed=data["Churn"])

# Predict survival probabilities for entire dataset
full_survival_probs = kmf_full.predict(data["tenure"].values)

# Convert survival probabilities to risk scores for the full dataset
full_risk_scores = 1 - full_survival_probs

# Compute C-Index for the full dataset
full_c_index = concordance_index(
    data["tenure"],
    full_risk_scores,
    data["Churn"]
)

print(f"Full Dataset Concordance Index: {full_c_index:.4f}")

# Calculate Brier Score for the full dataset
full_churn_probs = 1 - full_survival_probs
brier_full = brier_score_loss(data["Churn"], full_churn_probs)
print(f"Kaplan-Meier Brier Score (Full Dataset): {brier_full:.4f}")
```

Probability:

The fitted Kaplan-Meier model estimates the **survival function**, which describes the probability that a customer remains active at any given time.
This calculates the probability of a customer **still being subscribed** at selected months (1, 3, 6, 12, and 24).

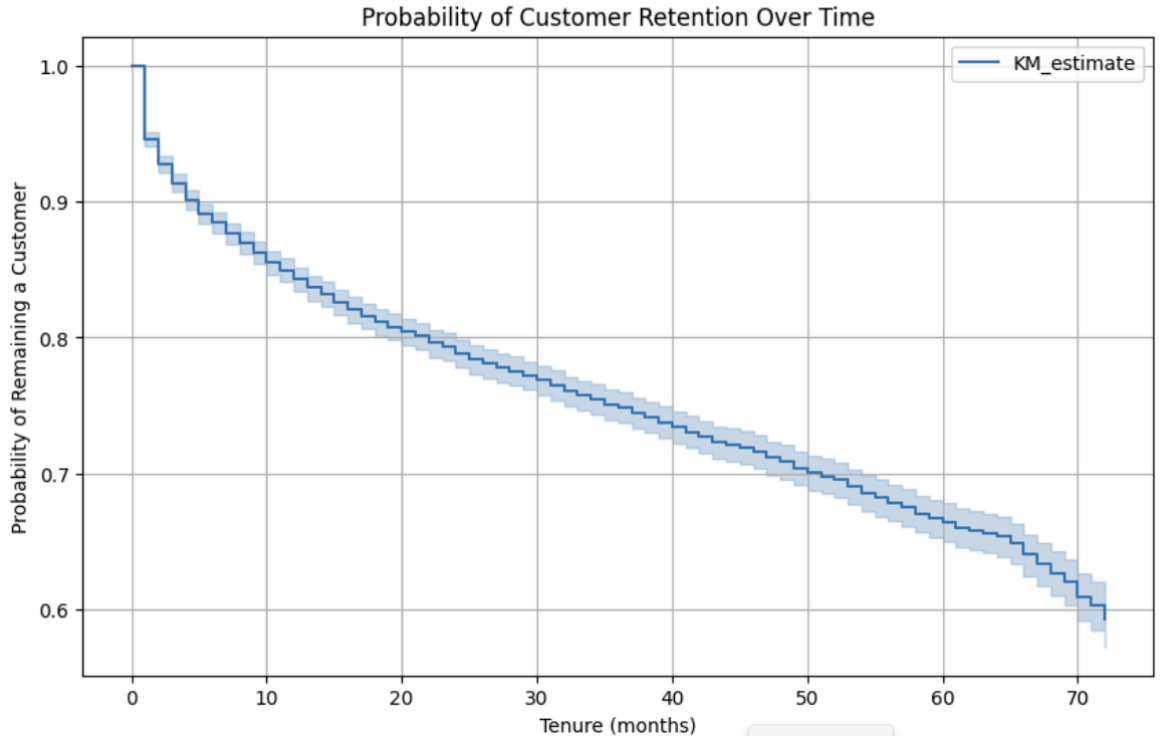
kmf.predict(t) provides the survival probability at time *t*.
*The **churn probability** is calculated as: **P(Churn by t) = 1−P(Retention by t).***

```
# Fit the Kaplan-Meier model
kmf = KaplanMeierFitter()
kmf.fit(data['tenure'], data['Churn'])



# 2. Predict for multiple time points
time_points = [1, 3, 6, 12, 24]
for t in time_points:
    churn_prob = 1 - kmf.predict(t)
    print(f"Probability of churning by month {t}: {churn_prob:.4f}"

# 3. Generate full survival curve
survival_curve = kmf.survival_function_
plt.figure(figsize=(10, 6))
kmf.plot()
plt.title("Probability of Customer Retention Over Time")
plt.xlabel("Tenure (months)")
plt.ylabel("Probability of Remaining a Customer")
plt.grid(True)
plt.show()
```

This means 8.63% expect to churn by 3 months, 11.52% of customers are expected to have churned within 6 months and by 24 months, about 21.13% of customers are expected to have left.

Probability of churning by month 1: 0.0540
Probability of churning by month 3: 0.0863
Probability of churning by month 6: 0.1152
Probability of churning by month 12: 0.1568
Probability of churning by month 24: 0.2113

```
Probability of churning by month 1: 0.0540
Probability of churning by month 3: 0.0863
Probability of churning by month 6: 0.1152
Probability of churning by month 12: 0.1568
Probability of churning by month 24: 0.2113
```



Cox PH Analysis (Approach-1)

The Cox Proportional Hazards (**Cox-PH**) model is a widely used survival analysis technique that helps estimate the probability of an event (customer churn) occurring over time. The **Cox-PH model** estimates the effect of multiple factors on the **hazard function** (the risk of churn at a given time). It assumes the form:

$$h(t|X) = h_0(t) \times e^{(\beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n)}$$

Where:

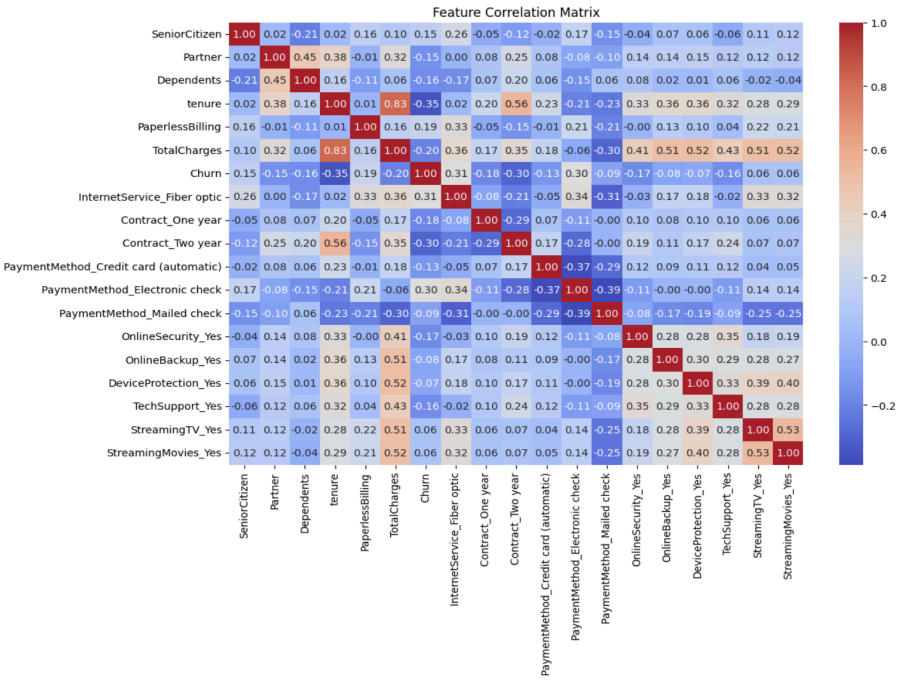
- h(t | X) => **Hazard rate** (risk of churn at time t).
- h0(t) => **Baseline hazard function** (risk when all predictors are zero).
- Xi => **Predictor variables** (customer attributes like contract type, monthly charges, etc.).
- Bi {i = n} => **Coefficients** that represent the effect of each predictor on churn risk.

Contrast with Kaplan Meier: The Cox-PH model is a **semi-parametric** method that estimates the **effect** of multiple **variables on the hazard of an event occurring whereas, Kaplan Meier is a non-parametric method.**

Kaplan Meier provides a survival curve but does not explain why events occur.
Cox PH estimates the impact of variables (e.g., contract type, monthly charges) on churn.
Incorporates independent variables – Unlike KM which does not incorporate independent variables, it accounts for multiple factors influencing churn risk.
Assumes proportional hazards – The effect of each variable on churn remains constant over time.

Before fitting Cox PH, these things should be taken into account:

Perfect Multicollinearity: Some features might be highly correlated or even duplicates. [Check VIF > **15**]
Very Low/Low Variance Columns: Features with the same value for all rows (constant columns) can cause singularity. (Variance Threshold = **0.1**)
Missing Values (NaN/Inf): Cox-PH **cannot** handle missing values, which can make the matrix singular.



Too Many One-Hot Encoded Categories: If a categorical feature has many categories, the matrix might become nearly singular. To Remove redundant columns from "`pd.get_dummies()`".

Excluding columns with VIF > 15, we try to fit the Cox-PH.

Results:

```python
from lifelines import CoxPHFitter
from lifelines.utils import concordance_index

# Fit the Cox Proportional Hazards model
cph = CoxPHFitter()
cph.fit(churn_hazard_2, duration_col="tenure", event_col="Churn")

# Calculate Concordance Index
c_index = concordance_index(churn_hazard_1["tenure"], -cph.predict_partial_hazard(churn_hazard_1)
print(f"Cox-PH Model Concordance Index: {c_index:.4f}")
```

```python
from sklearn.metrics import brier_score_loss

# Predict probabilities
pred_probs = 1 - cph.predict_survival_function(churn_hazard_2).T

# Compute Brier Score
brier_score = brier_score_loss(churn_hazard_2["Churn"], pred_probs.mean(axis=1))
print(f"Brier Score: {brier_score:.4f}")
```

Cox-PH Model Concordance Index: **0.9265**, Brier Score: **0.1908**

```python
# Predict survival function
survival_curve = cph.predict_survival_function(churn_hazard_2)

# Probability of surviving after 3 months
survival_3_months = survival_curve.loc[3].values[0]
print(f"Probability of customer staying after 3 months: {survival_3_months:.4f}")

# Probability of churning after 3 months
churn_prob = 1 - survival_3_months
print(f"Probability of churning within 3 months: {churn_prob:.4f}")
```

Probability of customer staying after 3 months: **0.9153**
Probability of churning within 3 months: **0.0847**

model	lifelines.CoxPHFitter
duration col	'tenure'
event col	'Churn'
baseline estimation	breslow
umber of observations	7043
mber of events observed	1869
partial log-likelihood	-12841.68
time fit was run	2025-02-23 21:38:39 UTC

	coef	exp(coef)	se(coef)	coef lower 95%	coef upper 95%	exp(coef) lower 95%	exp(coef) upper 95%
SeniorCitizen	0.06	1.06	0.06	-0.05	0.17	0.95	1.18
Partner	-0.23	0.80	0.06	-0.33	-0.12	0.72	0.89
Dependents	-0.14	0.87	0.07	-0.28	-0.01	0.75	0.99
PaperlessBilling	0.22	1.25	0.06	0.11	0.33	1.12	1.40
TotalCharges	-3.02	0.05	0.08	-3.17	-2.87	0.04	0.06

Examples :-

Dependents (-0.14, HR = 0.87, p = 0.04)

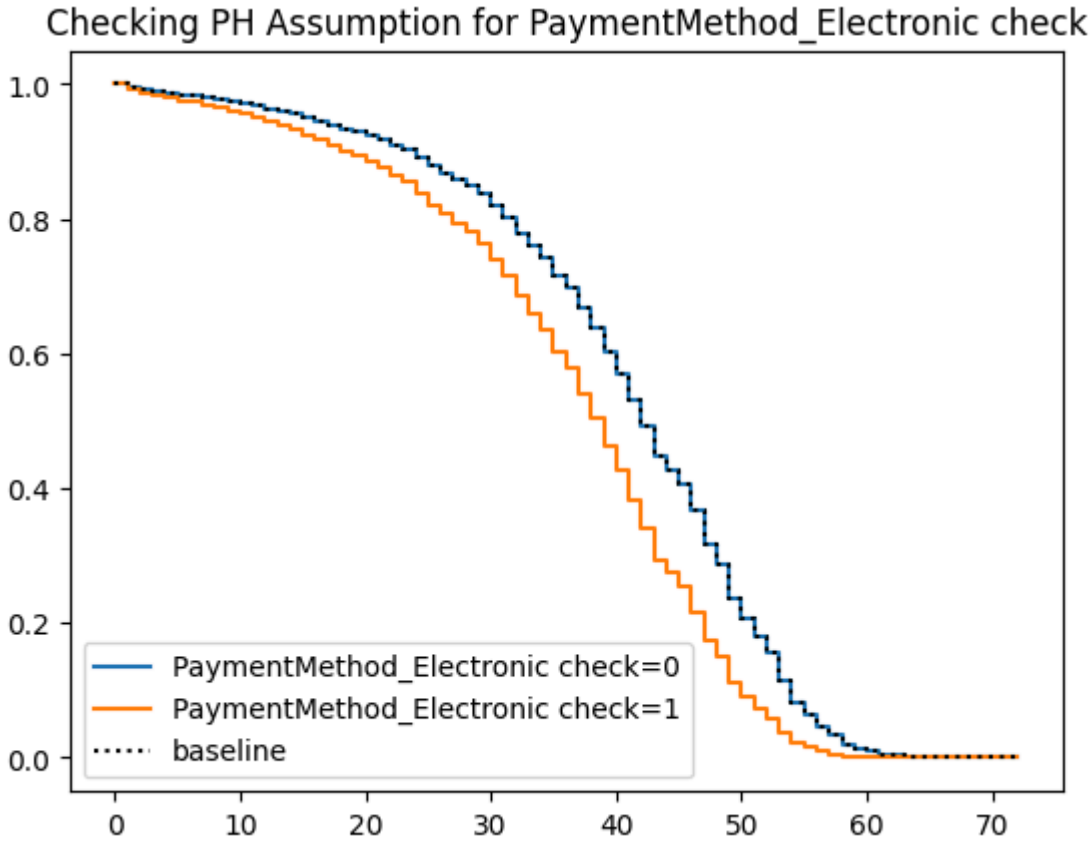
- **Customers with dependents have 13% lower risk of churn.**
- **Interpretation: Customers with dependents are less likely to leave.**

🔹 **PaperlessBilling (0.22, HR = 1.25, p < 0.005)**

- **Customers with paperless billing have 25% higher risk of churn.**
- **Interpretation: Paperless billing users may be more likely to leave (perhaps less commitment).**

Other Interpretations:

These plots checks the **Proportional Hazards (PH) assumption for the covariate [PaymentMethod]** in a Cox Proportional Hazards model. If the proportional hazards assumption holds, the survival curves for different values of TotalCharges should be proportional over time (**i.e., they should not cross and should maintain a similar shape**).



The Proportional Hazards (PH) assumption is a fundamental assumption in Cox Proportional Hazards (Cox PH) models, which are commonly used in survival analysis. It states that:

> **The ratio of hazard functions between any two individuals remains constant over time.**

This means that the effect of a covariate on the hazard rate (risk of the event occurring) does not change over time.

The Cox model expresses the hazard function as:

$h(t|X) = h_0(t) \cdot e^{(\beta X)}$

where:

- $h(t|X)$ is the hazard function at time t given covariates X,
- $h_0(t)$ is the **baseline hazard function,**
- β represents the regression coefficients for covariates XXX,
- $e^{(\beta X)}$ scales the baseline hazard but **does not depend on time t.**

If the PH assumption holds, the **hazard ratio (HR)** between two individuals with covariates X1X_1X1 and X2X_2X2 remains constant:

$h(t|X1)/h(t|X2) = e^{\beta(X1-X2)}$

This means the relative risk between the two individuals **remains the same at all time points.**

Schoenfeld Test : If the **PH assumption holds**, the residuals should be **randomly scattered** and show **no systematic trend over time.** If they show a trend, it means the covariate's effect changes over time, violating the PH assumption

The **Grambsch-Therneau test** (based on Schoenfeld residuals) checks whether there is a significant time dependency.

🔹 **Null Hypothesis (H0):** The PH assumption holds (no time dependency).
🔹 **Alternative Hypothesis (Ha):** The PH assumption is violated (time dependency exists).

Test Steps:

1. Compute Schoenfeld residuals for each covariate.
2. Regress the residuals against time.

3. If there is a significant trend (**p-value < 0.05**), the **PH assumption is violated.**

Future Tasks (**when PH assumption violated**):

- Using **time-dependent covariates** (e.g., interaction with **log(time)**).
- Consider a **stratified Cox model** (**stratify on the violating variable**).
- Try **parametric survival models** (e.g., **Weibull, Gompertz**) instead of Cox.