Q2 Report: Survival Analysis for Telco Customer Churn

Project 1 — Big Data Analysis

April 2026

Contents

1 Introduction

Survival Analysis is a collection of statistical methods used to model the time until an event of interest occurs. In this case study, the event is customer **churn** — the cancellation of a telecommunications subscription. The dataset comes from IBM's fictitious Telco company and contains 1,000 subscriber records with demographic, service-plan, and subscription-status information.

Two columns are central to every survival analysis model:

- **Tenure** — the number of months a customer has been (or was) with the company.
- **Churn** — a binary flag: 1 if the customer cancelled (*event observed*), 0 if still subscribed (*censored*).

Censoring is a key concept: customers who have not yet churned at the time of data collection are *right-censored*. Ignoring censored observations would underestimate survival probabilities; survival analysis handles them correctly.

This report covers four stages:

1. Data loading and preparation (bronze → silver curation)
2. Kaplan-Meier estimation and the log-rank test
3. Cox Proportional Hazards regression
4. Accelerated Failure Time (Log-Logistic) regression
5. Customer Lifetime Value (CLV) calculation

2 Data Preparation

2.1 Raw Data (Bronze Layer)

The raw CSV contains 1,000 rows and 21 columns. The `totalCharges` column is stored as a string and must be coerced to numeric. The target column is named `churnString` ("Yes" / "No") and is converted to a binary integer.

Listing 1: Load and clean raw data

```
import pandas as pd
import numpy as np

df = pd.read_csv('Telco-Customer-Churn.csv')
df['totalCharges'] = pd.to_numeric(df['totalCharges'], errors='
    coerce')
df['churn'] = (df['churnString'] == 'Yes').astype(int)
```

2.2 Curated Data (Silver Layer)

To keep the analysis focused on the highest-risk segment, we apply two filters that mirror the original Databricks tutorial:

- **Contract type**: month-to-month only (no long-term commitment, so churn is more likely).
- **Internet service**: exclude customers with no internet service.

Listing 2: Filter to silver layer

```
telco = (
    df.query("contract == 'Month-to-month'")
      .query("internetService != 'No'")
      .drop(columns=['churnString'])
      .reset_index(drop=True)
)
```

2.3 Silver Layer Statistics

After filtering, the silver dataset contains **422 records**.

Table 1: Silver layer descriptive statistics

Metric	Value
Total records	422
Churn rate	28.9%
Mean tenure	37.4 months
Median tenure	39 months
Max tenure	72 months

3 Kaplan-Meier Survival Analysis

3.1 Overview

Kaplan-Meier (KM) is a **non-parametric** method that estimates the survival function $S(t)$ — the probability that a customer is still subscribed at time t — without assuming any particular distribution for survival times. The estimator is:

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

where d_i is the number of churn events at time t_i and n_i is the number of customers at risk just before t_i.

3.2 Population-Level Results

Listing 3: Fit population-level KM model

```
from lifelines import KaplanMeierFitter

T = telco['tenure']
C = telco['churn'].astype(float)
```

```
kmf = KaplanMeierFitter()
kmf.fit(T, C)
print('Median survival time:', kmf.median_survival_time_)
```

Median survival time: 64 months. This means a customer has a 50% probability of remaining subscribed for at least 64 months.

Table 2: Population-level survival probabilities

Time (months)	$\hat{S}(t)$
6	0.9829
12	0.9675
24	0.9155
36	0.8337
48	0.7628
60	0.5818
72	0.1577

Interpretation: at 12 months, 96.75% of customers are still subscribed; by 60 months, only 58.18% remain.

3.3 Covariate-Level Analysis and Log-Rank Test

We split the KM curves by each covariate to assess whether different groups exhibit different survival behaviour. The **log-rank test** formally tests the null hypothesis H_0: the survival curves of all groups are identical. A p-value < 0.05 indicates statistically significant differences.

Listing 4: Covariate-level KM and log-rank test

```
from lifelines.statistics import pairwise_logrank_test

def plot_km_by_group(col):
    ax = plt.subplot(111)
    for val in telco[col].unique():
        mask = telco[col] == val
        kmf.fit(T[mask], C[mask], label=str(val))
        kmf.plot(ax=ax)
    plt.show()

def logrank_summary(col):
    result = pairwise_logrank_test(telco['tenure'], telco[col],
       telco['churn'])
    return result.summary
```

Table 3: Log-rank test results for selected covariates

Covariate	Comparison	p-value
Gender	Male vs Female	0.7659
Internet Service	DSL vs Fiber optic	0.1701
Online Security	No vs Yes	0.7079

Key findings:

- **Gender** ($p = 0.77$): the two survival curves are statistically equivalent. Gender is not a useful predictor of churn in this dataset.
- **Internet Service** ($p = 0.17$): DSL customers have a slightly higher median survival (65 months) than Fiber optic customers (63 months), but the difference is not statistically significant at $\alpha = 0.05$.
- **Online Security** ($p = 0.71$): no significant difference between customers with and without online security.

3.4 Survival Probability Extraction

KM survival probabilities can be extracted for downstream use (e.g., as input to a CLV model). For DSL internet customers:

Listing 5: Extract survival probabilities for DSL customers

```
mask_dsl = telco['internetService'] == 'DSL'
sp_dsl = kmf.fit(T[mask_dsl], C[mask_dsl], label='DSL')
pd.DataFrame(sp_dsl.survival_function_at_times(range(0, 10)),
             columns=['Survival Probability'])
```

4 Cox Proportional Hazards

4.1 Overview

Cox Proportional Hazards (CPH) is a **semi-parametric** model that estimates the *hazard ratio* for each covariate relative to a baseline. The hazard $h(t)$ is the instantaneous rate of churn at time t:

$$h(t \mid \mathbf{x}) = h_0(t) \cdot \exp\left(\boldsymbol{\beta}^\top \mathbf{x}\right)$$

where $h_0(t)$ is the unspecified baseline hazard and $\exp(\beta_j)$ is the hazard ratio for covariate j. A hazard ratio > 1 means the covariate *increases* churn risk; < 1 means it *decreases* churn risk.

4.2 One-Hot Encoding

Categorical variables must be one-hot encoded before fitting. We hand-select which dummy to drop so the baseline resembles the population average.

Listing 6: One-hot encode and select model columns

```
encode_cols = ['dependents','internetService','onlineBackup',
               'techSupport','paperlessBilling']
encoded = pd.get_dummies(telco, columns=encode_cols,
                         prefix=encode_cols, drop_first=False)
survival_pd = encoded[['churn','tenure',
    'dependents_Yes','internetService_DSL',
    'onlineBackup_Yes','techSupport_Yes']].copy()
survival_pd['churn'] = survival_pd['churn'].astype(float)
```

4.3 Model Results

Listing 7: Fit Cox PH model

```
from lifelines.fitters.coxph_fitter import CoxPHFitter

cph = CoxPHFitter(alpha=0.05)
cph.fit(survival_pd, duration_col='tenure', event_col='churn')
cph.print_summary()
```

Concordance index: 0.5675 (0.5 = random, 1.0 = perfect; 0.57 indicates modest predictive power.)

Table 4: Cox PH model summary

Covariate	Coef	exp(coef)	p-value	95% CI for exp(coef)
dependents_Yes	0.340	1.405	0.078	[0.963, 2.049]
internetService_DSL	-0.242	0.785	0.206	[0.540, 1.142]
onlineBackup_Yes	-0.107	0.898	0.579	[0.616, 1.311]
techSupport_Yes	-0.027	0.974	0.891	[0.665, 1.426]

Interpretation:

- **dependents_Yes** ($\exp(\beta) = 1.405$, $p = 0.078$): customers with dependents have a 40.5% higher hazard of churning compared to those without, though this is marginally significant.
- **internetService_DSL** ($\exp(\beta) = 0.785$, $p = 0.206$): DSL subscribers have a 21.5% lower hazard than Fiber optic subscribers, but the result is not statistically significant.
- **onlineBackup_Yes** and **techSupport_Yes**: both show slightly reduced hazard but are not statistically significant ($p > 0.5$).

Note: None of the covariates reach the conventional $\alpha = 0.05$ threshold. This is partly due to the small silver dataset (422 records) and the pre-filtering to month-to-month contracts, which reduces variance.

4.4 Proportional Hazards Assumption Check

The CPH model assumes that the hazard ratio between any two groups is *constant over time*. We verify this with three methods.

Listing 8: Check proportional hazards assumption

```
# Method 1: statistical test
cph.check_assumptions(survival_pd, p_value_threshold=0.05)

# Method 2: Schoenfeld residual plots (show_plots=True)
cph.check_assumptions(survival_pd, p_value_threshold=0.05,
   show_plots=True)

# Method 3: log-log KM plots
def plot_km_loglog(col):
    ax = plt.subplot(111)
    for val in telco[col].unique():
        mask = telco[col] == val
        kmf.fit(T[mask], C[mask], label=str(val))
        kmf.plot_loglogs(ax=ax)
    plt.show()
```

The statistical test and Schoenfeld residual plots indicate that the proportional hazards assumption is **violated** for several covariates (consistent with the original tutorial findings). This motivates the use of the Accelerated Failure Time model in the next section.

5 Accelerated Failure Time (AFT)

5.1 Overview

Accelerated Failure Time is a **fully parametric** model. Rather than estimating a hazard ratio, it models how covariates *accelerate or decelerate* the time until the event. The survival function is:

$$S_A(t) = S_B\left(\frac{t}{\phi}\right)$$

where ϕ is the *acceleration factor*. If $\phi > 1$, group A experiences the event faster than group B (accelerated failure).

We use the **Log-Logistic** distribution, which has the survival function:

$$S(t) = \frac{1}{1 + (\lambda t)^p}$$

5.2 Model Fitting

Listing 9: Fit Log-Logistic AFT model

```
from lifelines import LogLogisticAFTFitter

aft_encode_cols = ['partner','multipleLines','internetService',
```

```
    'onlineSecurity','onlineBackup','deviceProtection',
    'techSupport','paymentMethod']
aft_encoded = pd.get_dummies(telco, columns=aft_encode_cols,
                             prefix=aft_encode_cols, drop_first=
                                False)
aft_survival_pd = aft_encoded[['churn','tenure',
    'partner_Yes','multipleLines_Yes','internetService_DSL',
    'onlineSecurity_Yes','onlineBackup_Yes','deviceProtection_Yes',
    'techSupport_Yes','paymentMethod_Bank transfer (automatic)',
    'paymentMethod_Credit card (automatic)']].copy()
aft_survival_pd['churn'] = aft_survival_pd['churn'].astype(float)

aft = LogLogisticAFTFitter()
aft.fit(aft_survival_pd, duration_col='tenure', event_col='churn')
```

5.3 Model Results

AIC: 1396.73

Table 5: AFT (Log-Logistic) model summary — α parameters

Covariate	Coef	exp(coef)	p-value	95% CI
deviceProtection_Yes	-0.043	0.958	0.695	[0.775, 1.186]
internetService_DSL	0.102	1.108	0.333	[0.900, 1.362]
multipleLines_Yes	0.144	1.155	0.216	[0.919, 1.451]
onlineBackup_Yes	-0.001	0.999	0.993	[0.807, 1.237]
onlineSecurity_Yes	-0.066	0.936	0.540	[0.758, 1.156]
partner_Yes	0.039	1.040	0.709	[0.848, 1.274]
paymentMethod_Bank transfer (automatic)	0.066	1.068	0.589	[0.841, 1.357]
paymentMethod_Credit card (automatic)	0.037	1.038	0.770	[0.809, 1.331]
techSupport_Yes	0.064	1.066	0.550	[0.865, 1.314]
Intercept (α)	4.138	62.69	≈ 0	[48.47, 81.09]
Intercept (β)	0.789	2.200	≈ 0	[1.895, 2.554]

Interpretation:

- The Intercept (α) of 62.69 represents the baseline median survival time when all covariates are at their reference level.
- **multipleLines_Yes** ($\exp(\beta) = 1.155$): customers with multiple lines have a 15.5% longer expected survival time, though not statistically significant ($p = 0.216$).
- **onlineSecurity_Yes** ($\exp(\beta) = 0.936$): slightly shorter survival for customers with online security, but not significant.
- None of the covariates are statistically significant at $\alpha = 0.05$, consistent with the small dataset size.

5.4 Assumption Verification

Log-odds plots are used to verify two AFT assumptions:

1. **Proportional Odds**: lines should be parallel.
2. **Correct distribution**: lines should be straight (log-logistic is appropriate when the log-odds plot is linear).

Listing 10: Log-odds plots for AFT assumption verification

```
def plot_km_log_odds(col):
    ax = plt.subplot(111)
    for val in telco[col].unique():
        mask = telco[col] == val
        kmf.fit(T[mask], C[mask], label=str(val))
        sf = kmf.survival_function_.copy()
        sf['failureOdds'] = np.log((1 - sf.iloc[:,0]) / (sf.iloc
            [:,0] + 1e-9))
        sf['logTime'] = np.log(sf.index + 1e-9)
        ax.plot(sf['logTime'], sf['failureOdds'], label=str(val))
    plt.show()
```

The log-odds plots show that lines are mostly straight (supporting the log-logistic distribution choice) but not always parallel (indicating the proportional odds assumption is partially violated), consistent with the original tutorial findings.

6 Customer Lifetime Value (CLV)

6.1 Methodology

Using the fitted Cox PH model, we predict the survival probability curve for a specific customer profile and compute the **Net Present Value (NPV)** of expected monthly profit:

$$\text{NPV}_t = \frac{S(t) \times \text{Monthly Profit}}{(1+r)^t} \qquad \text{CLV} = \sum_{t=0}^{T} \text{NPV}_t$$

where r is the monthly internal rate of return (IRR).

6.2 Customer Profile

Table 6: Customer profile used for CLV calculation

Parameter	Value
dependents_Yes	0 (No)
internetService_DSL	1 (DSL)
onlineBackup_Yes	1 (Yes)
techSupport_Yes	0 (No)
Monthly Profit	\$30
Annual IRR	10%

6.3 CLV Results

Listing 11: CLV calculation

```
customer_profile = pd.DataFrame([{
    'dependents_Yes': 0, 'internetService_DSL': 1,
    'onlineBackup_Yes': 1, 'techSupport_Yes': 0,
}])
monthly_profit = 30
irr = 0.10 / 12

sf = cph.predict_survival_function(customer_profile)
clv_df['NPV'] = clv_df['sp'] * monthly_profit / ((1 + irr) **
   clv_df.index)
clv_df['Cumulative NPV'] = clv_df['NPV'].cumsum()
```

Table 7: Cumulative NPV at key milestones

Horizon	Cumulative NPV
12 months	\$366.88
24 months	\$662.15
36 months	\$915.40
60 months	\$1,301.87
Full horizon (72 months)	\$1,412.92

Interpretation: The maximum CLV for this customer profile is \$1,412.92 over the full 72-month horizon. This represents the maximum acquisition cost a business should be willing to pay for a customer with these characteristics (assuming a \$30/month profit margin and 10% annual discount rate). The payback period is approximately 12 months (\$366.88), meaning the business recoups its acquisition cost within the first year.

7 Summary and Conclusions

Table 8: Comparison of survival analysis methods

Method	Type	Key Output	Best Use
Kaplan-Meier	Non-parametric	$\hat{S}(t)$ curve	Univariate exploration
Cox PH	Semi-parametric	Hazard ratios	Multi-variate inference
AFT (Log-Logistic)	Fully parametric	Acceleration factors	When distribution is known

Key findings from this case study:

1. The median survival time for month-to-month internet subscribers is **64 months**. Survival drops sharply after month 60.

2. **Gender** is not a significant predictor of churn (log-rank p = 0.77). This is consistent with the original tutorial.

3. **Internet service type** (DSL vs Fiber optic) shows a slight difference in median survival (65 vs 63 months) but is not statistically significant in this 422-record subset.

4. The Cox PH model achieves a concordance index of **0.5675**, indicating modest discriminative ability. The proportional hazards assumption is violated for several covariates.

5. The AFT model (AIC = 1396.73) confirms that none of the selected covariates are statistically significant at $\alpha = 0.05$, suggesting that the small filtered dataset limits statistical power.

6. The CLV analysis shows a maximum lifetime value of **$1,412.92** for a DSL subscriber with online backup over a 72-month horizon.