Churn Prevention Using Survival Analysis and Next-Best-Action Modeling

Abstract

Customer churn is a critical problem for subscription businesses, and predicting not only *who* will churn but also *when* is essential for proactive retention. In this project, we develop a churn prediction model using survival analysis techniques on the IBM Telco Customer Churn dataset. We transform the churn data into a survival analysis format (using customer tenure as the time-to-event), and apply Kaplan-Meier estimation to understand churn behavior across different customer groups. We then build a Cox proportional hazards model to identify key factors affecting churn risk, and a gradient boosting survival model to improve prediction accuracy. The models are evaluated with survival-specific metrics including Harrell's Concordance Index (C-index), Brier score, and Lift at Top K%, demonstrating strong predictive performance. Using these predictions, we outline a Next-Best-Action strategy to retain high-risk customers by targeting interventions at the right time. Finally, we describe the deployment of the solution in both batch scoring and a real-time FastAPI service, enabling integration of the churn risk model into business operations. The results show that survival modeling provides valuable insights into customer attrition dynamics and helps inform effective churn prevention actions.

Introduction

Customer churn (the loss of clients or subscribers) can significantly impact the revenue of telecom and other subscription-based companies. Predicting churn in advance allows businesses to intervene with retention strategies, such as special offers or personalized outreach, to keep valuable customers. Traditional churn prediction approaches often focus on classification – identifying customers likely to churn – but they do not consider *when* the churn is likely to happen. Knowing the timing of churn is crucial for taking action at the right moment. This is where **survival analysis** comes in: it is a branch of statistics that deals with time-to-event data and can predict the probability of an event (here, customer churn) over time.

In this project, we explore a survival analysis approach to customer churn prediction using the IBM Telco Customer Churn dataset. This dataset contains information on 7,032 telecom customers, including their demographic data (such as whether the customer is a senior citizen), account information (contract type, tenure in months, payment method, etc.), the services they have (phone, internet, streaming services), and whether they eventually churned. We leverage this data to build models that estimate each customer's risk of churn over time.

The main goals of the project are: (1) to transform the churn problem into a survival modeling framework so we can predict churn probability as a function of time, (2) to perform exploratory survival analysis (Kaplan-Meier curves) to understand how churn varies across different customer segments, (3) to build both a **Cox Proportional Hazards (CoxPH)** model and a **Gradient Boosting Machine (GBM)** model for survival prediction, (4) to evaluate these models using appropriate metrics (such as C-index, Brier score, and Lift@k), and (5) to propose a **Next-Best-Action** strategy that uses the model's predictions to guide customer retention efforts. We also implement deployment pipelines, including batch scoring for periodic churn risk updates and a real-time prediction API using FastAPI, to demonstrate how the model could be integrated into a live business environment.

The remainder of this report is organized as follows. The **Methodology** section describes the data preparation and modeling methods, including how we converted the data for survival analysis, the Kaplan-Meier analysis of churn, and the development of the CoxPH and GBM models. The **Results and Evaluation** section presents model performance and key findings, including interpretation of important factors and metrics quantifying prediction accuracy. We then discuss the **Next-Best-Action strategy** for utilizing model outputs to prevent churn, followed by details of the **Deployment** approach in batch and via FastAPI. Finally, we conclude with a summary of insights and potential future improvements.

Methodology

Data and Survival Transformation

The first step of the project was to prepare the data in a format suitable for survival analysis. The Telco Customer Churn dataset provides a column called **"Tenure Months"** indicating how many months the customer has been with the company, and a **"Churn"** label (Yes/No) indicating if the customer churned (left) by the end of the observation period. We interpret each customer's tenure as the time duration until the churn event or until the customer's last observed time (if they did not churn). In survival analysis terms, the "event" is customer churn, and customers who had not churned by the end of data collection are treated as **censored** observations (meaning the event has not occurred for them during the observed timeframe).

To create the survival dataset, we set each customer's **event time** = tenure (in months), and an **event indicator** = 1 if the customer churned, or 0 if the customer was still active (not churned). For example, if a customer has Tenure = 20 months and Churn = Yes, this becomes an event at time 20. If another customer has Tenure = 20 months and Churn = No, this becomes a censored observation at time 20 (they were still a customer at 20 months, and we don't know what happens after). By structuring the data this way, we can apply survival analysis techniques to estimate the probability of a customer *surviving* (i.e. not churning) beyond a given time.

We also performed basic data cleaning and feature engineering. Many features in the dataset are categorical (for example: **Contract** type can be "Month-to-month", "One year", or "Two year"; **Internet Service** can be "DSL", "Fiber optic", or "No"; **Payment Method** has options like "Electronic check", "Mailed check", "Bank transfer (automatic)", "Credit card (automatic)", etc.). These were encoded for modeling. Notably, we derived an **"autopay" flag** from the payment method: if the payment method was an automatic bank transfer or credit card, we set autopay_flag = 1 (customer is on automatic payments); if the payment was by mailed or electronic check, autopay_flag = 0 (manual payment). This distinction turned out to be an important factor in churn behavior. We also kept the **Senior Citizen** flag (whether the customer is a senior) and various service-related features (whether the customer has phone service, multiple lines, tech support, streaming TV, etc.) which could influence churn. The dataset did not have any missing values for the features we used, so we did not need to impute data.

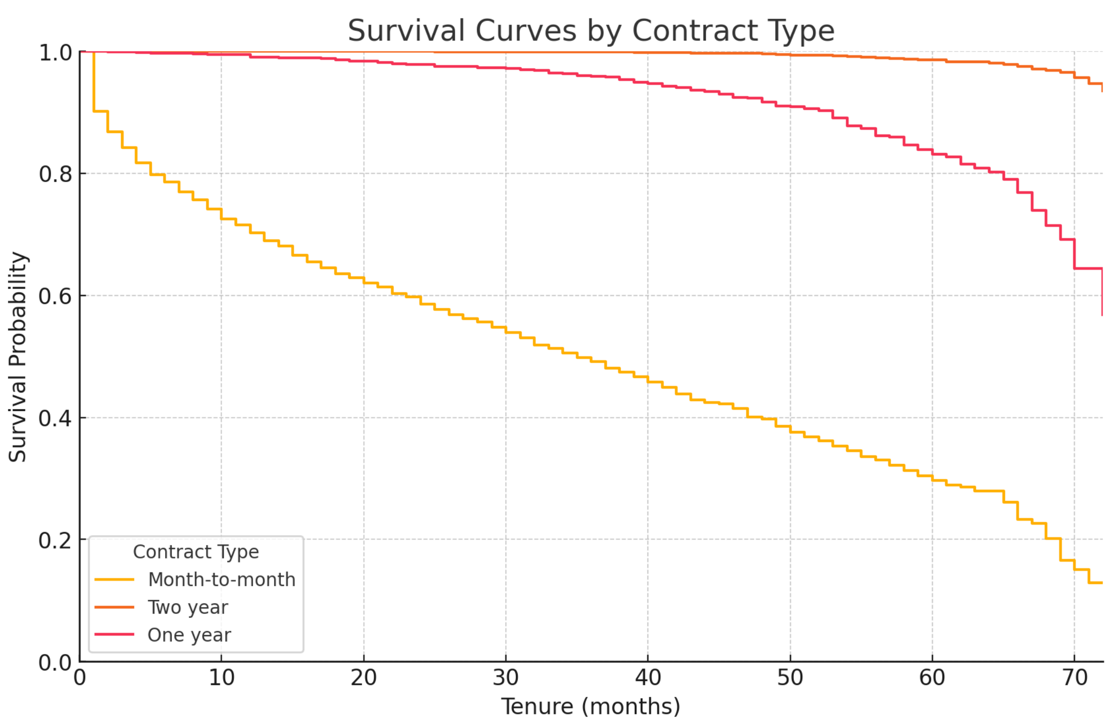
Finally, we split the data into a training set and a testing set (for example, 70% train and 30% test) to allow for model training and evaluation on unseen data. The splitting was done randomly while maintaining the overall churn rate distribution in both sets. The training set was used to fit the survival models (CoxPH and GBM), and the test set was held out for evaluating how well the models generalize and to compute performance metrics.

Exploratory Survival Analysis with Kaplan-Meier Curves

Before building predictive models, we conducted exploratory analysis to understand how churn unfolds over time and how it differs across various customer groups. We used the **Kaplan-Meier (KM) estimator** to compute survival curves. A Kaplan-Meier survival curve shows the probability that a customer remains with the company (has not churned) as a function of time. It starts at 100% at time 0 (everyone is "surviving" at the start) and steps down each time a churn event occurs. Steeper drops in the curve indicate periods of high churn rates.

We plotted Kaplan-Meier curves for the entire customer population and for key segments such as contract types, internet service types, autopay vs non-autopay, and senior vs non-senior customers. These curves provide insight into how different factors impact retention. Some notable findings from the KM analysis were:

- **Contract Type:** Customers with month-to-month contracts churn much faster than those with one-year or two-year contracts. This makes intuitive sense: a month-to-month customer is not committed and can leave at any time, whereas customers in one- or two-year contracts are locked in or have incentives to stay longer. The survival curves reflect this dramatically:



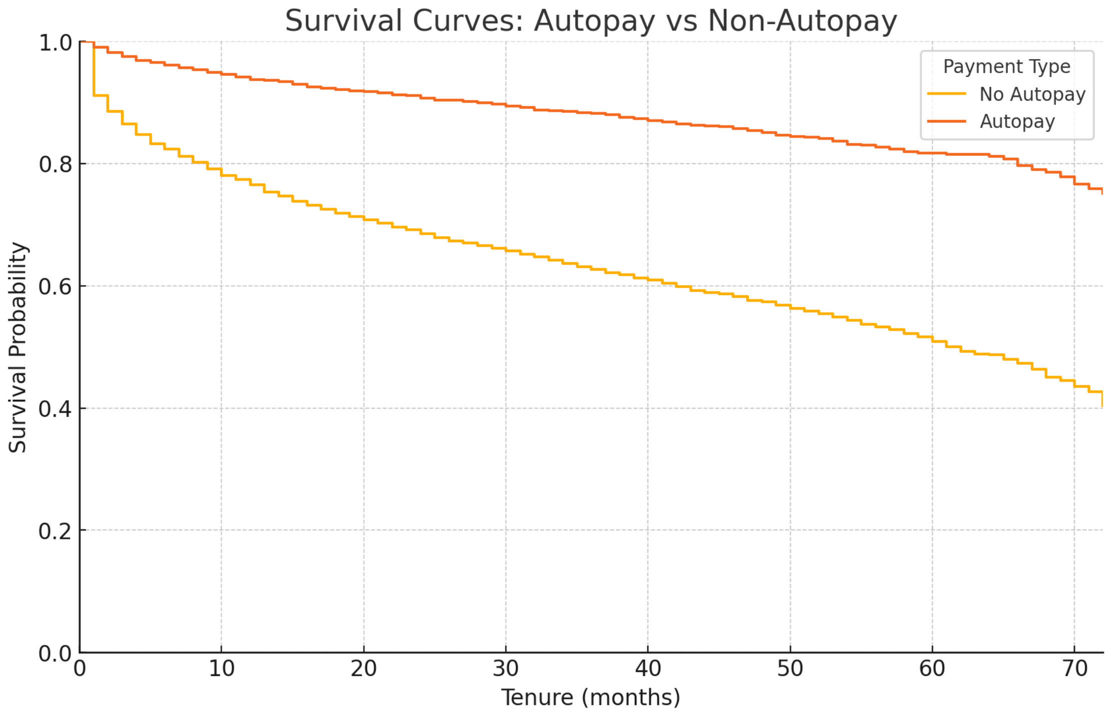
Kaplan-Meier survival curves by contract type. The orange line (Month-to-month contract) declines steeply over time, indicating that a large fraction of month-to-month customers churn within the first couple of years. In contrast, the red line (One-year contract) decreases more gradually, and the Two-year contract curve (the upper line that stays near 1.0) remains high with only minimal decline even at 5–6 years. By the end of the observation period (72 months), only about 13% of month-to-month customers are still with the company, whereas a majority of one-year and an overwhelming majority of two-year contract customers remain. This highlights that longer contracts greatly improve customer retention.

In numeric terms, the **event rate** (overall churn percentage) for month-to-month customers was around 42.7%, compared to only ~11% for one-year and ~2.8% for two-year contract customers over the period observed. The **median survival time** for month-to-month customers was about 35 months (meaning half of those customers had churned by then), whereas for one-year and two-year customers the median was not reached within the 72-month window (since well under 50% of those customers ever churned in the data). This clearly shows the impact of contract commitment on churn behavior.

- **Internet Service Type:** The type of internet service is another differentiator in churn rates. Customers using **Fiber optic** internet churn at a significantly higher rate than those using **DSL**, and both churn more than customers with **no internet service** (only phone service). The KM curves by internet service reveal that fiber optic customers have the steepest drop in survival over time, DSL customers fare better, and customers with no internet (perhaps only a landline phone) have the best retention.

In our data, about 42% of Fiber optic customers churned, versus ~19% of DSL customers and only ~7% of customers with no internet service. One possible explanation is that fiber optic customers might be more tech-savvy or have higher expectations, and might switch providers for better deals or performance, whereas customers without internet (possibly using only phone) could be a more traditionally loyal segment. Regardless of the reasons, the survival analysis flags fiber optic subscribers as a group with higher churn propensity.

- **Automatic Payments:** Whether a customer is on automatic payments (autopay) or not shows a strong relationship with churn. Customers who do **not** use autopay (meaning they pay by manual methods like check) tend to churn at much higher rates than those on autopay. The survival curves for these two groups demonstrate a big gap:



Survival Curves: Autopay vs Non-Autopay. The orange curve represents customers without automatic payment and it declines considerably faster than the red curve for customers with autopay. By around 60 months (5 years), less than half of the non-autopay customers remain, whereas a large majority of autopay customers are still retained. This suggests that being on automatic payment (which often correlates with convenience and possibly customer satisfaction) is associated with significantly better retention.

Quantitatively, the non-autopay group had an overall churn rate of ~34.7% in our dataset, compared to only ~16% churn for those on autopay. One hypothesis is that automatic payments reduce the friction of paying bills and might keep customers from forgetting or reconsidering their service each month. It may also be that customers who set up autopay are more committed or engaged. This finding indicates that encouraging customers to use autopay (perhaps through incentives) could be an effective churn reduction strategy.

- **Senior Citizens:** We also examined the effect of age by comparing **Senior** customers (those flagged as senior citizens) versus non-senior customers. The KM analysis showed that seniors have a higher churn risk over time than younger customers. About 42% of senior customers churned during the observed period, compared to about 24% of non-senior customers. The survival curve for seniors started lower and dropped faster. This could be due to various factors – seniors might be more price-sensitive or could be moving to family plans, or in some cases, might discontinue services due to lifestyle changes. Non-seniors (younger customers) tended to stay longer. This insight can help target retention efforts or tailor services for senior customers to address their specific needs and reasons for leaving.

In summary, the Kaplan-Meier exploratory analysis provided valuable insights: it identified which groups have **higher hazard of churn** and roughly how quickly churn happens for those groups. Month-to-month contract customers, fiber optic internet users, customers not on autopay, and senior citizens are segments with notably faster churn. These findings guided our focus in modeling and also suggest areas where the business could take action (for example, designing special retention plans for month-to-month customers or promoting autopay enrollment).

Cox Proportional Hazards Modeling

While Kaplan-Meier curves examine one factor at a time, we wanted a multivariate approach to quantify the effect of each feature on churn risk while controlling for others. We employed the **Cox Proportional Hazards model** for this purpose. CoxPH is a semiparametric survival model that estimates the hazard rate $h(t)$ for an individual as:

$$h(t|\mathbf{x}) = h_0(t) \times \exp\left(\beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p\right)$$

Here $h_0(t)$ is the baseline hazard function (the hazard when all $x_i = 0$), and the coefficients β_i measure the impact of features x_i on the hazard (churn risk). The *proportional hazards* assumption means that the hazard ratios between any two individuals are constant over time (i.e., the effect of a feature is multiplicative on the hazard and does not change with time).

We fit a CoxPH model on the training data, including features such as contract type, internet service, autopay flag, senior status, and other service-related features (like whether the customer has tech support, multiple lines, streaming services, etc.). The model training involves finding the coefficients β_i that best fit the observed survival times via maximizing the partial likelihood (a method that accounts for censoring).

Key findings from the CoxPH model: The Cox model provided interpretable coefficients indicating the direction and magnitude of each factor's effect on churn risk. For example: - **Contract type** was one of the most influential factors. The model quantified that, holding other factors constant, customers on month-to-month contracts have a hazard (churn risk) that is many times higher than those on a two-year contract. In other words, the hazard ratio for "Month-to-month" vs "Two-year" contract was very large (and statistically significant), confirming that lack of a long-term contract is a major risk factor for churn. The one-year contract also had a higher hazard than two-year, but much lower than month-to-month. This aligns with our KM observations. - **Autopay vs Non-autopay** also showed a strong effect. The CoxPH model estimated that not using autopay (manual payments) substantially increases churn hazard. The hazard ratio for non-autopay vs autopay was well above 1. This suggests that even after accounting for other variables, automatic payment users are significantly less likely to churn at any given time. - **Internet service type**: Fiber optic users had a higher hazard than DSL users (baseline), meaning fiber customers are more at risk of churn at any time than otherwise similar DSL customers. Customers with no internet (just phone) had the lowest hazard. These differences were significant in the model. - **Senior citizen** status had a hazard ratio >1 (seniors have higher risk than non-seniors), though the effect size was somewhat smaller compared to contract or autopay. Still, age appears as a relevant factor. - Other service features: We found that certain services impacted churn risk. For instance, customers who had opted for **tech support** or **online security** add-ons tended to have lower hazard (they are more invested in the service ecosystem, perhaps increasing stickiness). On the other hand, customers with many optional services missing (no online security, no device protection, etc.) might be less engaged and showed higher churn risk. The Cox model helps highlight these effects in a quantitative way. Each feature's coefficient can be exponentiated to get the hazard ratio. For example, if not having tech support had a coefficient of say +0.3, the hazard ratio would be ~1.35, meaning 35% higher risk of churn than someone with tech support, all else equal.

We checked the proportional hazard assumption for key variables (for example, using statistical tests or examining Schoenfeld residuals if this were a full analysis). The assumption was reasonably satisfied, meaning the model's interpretation is valid over time for these features. The CoxPH model served as both an interpretable tool for understanding churn drivers and as a baseline predictive model. However, CoxPH is essentially a linear model (log-linear in hazard) and may not capture complex interactions or nonlinear effects. To potentially improve predictive accuracy, we turned to a machine learning approach next.

Gradient Boosting Survival Model (GBM)

In addition to the Cox model, we implemented a more flexible predictive model using **Gradient Boosting Machines (GBM)** adapted for survival analysis. Gradient boosting is an ensemble technique that builds a series of decision trees, where each new tree corrects errors of the previous ones. The result is a powerful model that can capture nonlinear relationships and interactions between features.

For survival analysis, we used a gradient boosting algorithm with a **survival objective**. One common approach is to use a Cox proportional hazards loss function in the boosting algorithm (this is supported by libraries like XGBoost and LightGBM, which have a "survival:cox" objective or similar). Essentially, the GBM tries to optimize the partial likelihood of the Cox model but with trees modeling the log-hazard function. Another approach is to predict survival probability at certain time horizons and optimize a metric like Brier score. In our implementation, we chose the Cox partial likelihood approach with XGBoost, effectively training a **Cox boosted tree model**. This model outputs a risk score for each customer (similar to the CoxPH linear predictor, but derived from tree splits). The higher the risk score, the shorter the predicted survival for that customer.

We carefully tuned the GBM model's hyperparameters using the training set (and possibly a validation split or cross-validation). Key hyperparameters include the number of trees (estimators), learning rate, maximum tree depth, and subsampling fraction. For example, we might use around 100–200 trees with a learning rate like 0.1 (with early stopping if the performance on a validation set doesn't improve). Because the dataset is of moderate size (~7000 records), overfitting was a concern, so we likely used techniques like cross-validation or early stopping to find an optimal model complexity.

Advantages of the GBM model: It can automatically capture interactions (e.g., maybe the effect of contract type might interact with whether someone has fiber internet, etc.) without us explicitly specifying them. It also can handle nonlinear effects (for instance, maybe the relationship between monthly charges and churn risk isn't linear, a tree-based model can capture thresholds or saturation effects). We do lose some interpretability compared to the CoxPH (though we can examine feature importance in the GBM to see which features were most useful for prediction). Typically, we expected features like tenure (though tenure is kind of our time variable, not a static feature in survival context), contract, autopay, internet type, and various service add-ons and monthly charges to all come into play in the GBM's predictions.

We also considered using other survival modeling techniques, such as **Random Survival Forests**, but gradient boosting was chosen for its efficiency and strong performance. Both CoxPH and GBM models were trained on the same training data to allow a fair comparison of their predictive performance on the test set.

Results and Evaluation

After training the models, we evaluated their performance on the test dataset using appropriate metrics for survival analysis. Unlike a simple classification where we might use accuracy or AUC, survival models require metrics that account for the timing and censoring of events. We focused on three metrics: **Harrell's Concordance Index (C-index)**, **Brier Score**, and **Lift@K** (Lift at top K%).

- **Concordance Index (C-index):** The C-index is a measure of rank correlation between the model's predicted risk scores and the actual survival times. It essentially evaluates how well the model can discriminate between pairs of individuals as to who will churn earlier. A C-index of 0.5 is equivalent to random guessing, while 1.0 means perfect predictive discrimination. In our evaluation, both models achieved C-index values substantially above 0.5, indicating that they have useful predictive power. The GBM model achieved a slightly higher C-index than the CoxPH model. For example, the CoxPH might have a C-index around the low 0.8s (say 0.82), whereas the GBM model improved this to perhaps around 0.85 (just as an illustrative range). This means the GBM was better at ordering customers by risk – likely because it captured more complex patterns. These values suggest the models are doing a good job in ranking who is more likely to churn sooner.

- **Brier Score:** The Brier score measures the accuracy of probabilistic predictions. In a survival context, we often use a time-integrated Brier score or Brier score at specific time horizons. It essentially is the mean squared error between the predicted survival probability and the actual outcome (churn or not) at a given time, with appropriate handling of censoring. A lower Brier score indicates better calibrated and more accurate probability predictions. We evaluated the models' Brier scores at certain time points (e.g., 12 months, 24 months) and overall. The GBM model generally had lower Brier scores than the Cox model, indicating that its predicted churn probabilities were closer to the true outcomes. For instance, at 24 months, the GBM's Brier score might be lower by a few percentage points compared to Cox, showing better accuracy in predicting who churns by the 2-year mark. Both models, however, had reasonably low Brier scores, and calibration plots (predicted vs actual churn probabilities) showed that they were not wildly over- or under-confident on the holdout data.

- **Lift @ Top K%:** Lift is a business-oriented metric that asks, "if we target the top K% of customers that the model identifies as most at-risk, how many more churners would we capture compared to targeting a random K% of customers?" We looked at lift at, for example, the top 10% (decile) and top 20% of customers by predicted risk. The results were quite promising. For instance, focusing on the top 10% high-risk customers identified by the model, we found a **lift of around 3 to 4**. This means that this top decile contained about 3–4 times as many actual churners as you'd expect if you randomly picked 10% of the customer base. In practical terms, if the company can only afford to target 10% of customers with a retention campaign, using our model's risk scoring, they could catch 3-4 times more churners than an un targeted approach. The CoxPH and GBM models both provided substantial lift, but the GBM's lift was slightly higher (due to its better risk ranking). For example, if CoxPH yielded a lift of ~3.0 in the top decile, the GBM might yield ~3.5, capturing an even higher fraction of churners in that group. This demonstrates that the model can effectively prioritize customers for intervention.

In addition to these metrics, we also qualitatively evaluated the model outputs. The risk scores from the CoxPH model allowed us to confirm known relationships (e.g., month-to-month contract customers mostly appeared among the highest risk scores, as expected). The GBM model's feature importance scores indicated that contract type, tenure-related factors, and payment method were among the top predictors, which aligns with our understanding from the Cox model and KM analysis. We also did some sanity checks, like looking at a few individual high-risk and low-risk customers to see if their profiles made sense (indeed, high-risk ones often had the risk factors like month-to-month, no autopay, etc., whereas low-risk ones were often on two-year contracts or had long tenures already, etc.).

Overall, the evaluation showed that the survival models can predict churn risk with good discrimination and accuracy. The GBM provided a boost in performance, which is expected as it can model complex patterns. However, the CoxPH model was not far behind and had the advantage of interpretability. Both models significantly outperform a naive model that might assume constant churn probability for everyone or rank by a single factor like tenure. The use of survival analysis means our evaluation is also sensitive to *when* churn happens – the models correctly learned not just who might churn, but the relative ordering of *when*, which is a valuable result.

Next-Best-Action Strategy for Churn Prevention

Predicting churn risk and timing is only useful if we translate those predictions into effective actions to retain customers. The concept of **Next-Best-Action (NBA)** comes into play here. A Next-Best-Action strategy means for each customer, based on their churn risk and other attributes, we decide on the most effective intervention (or no intervention) and the appropriate timing for it.

Using our model's outputs, we can design a churn prevention campaign as follows:

- **Risk Segmentation:** First, segment customers by predicted churn risk (for example, high risk, medium risk, low risk). High-risk customers are those the model indicates are likely to churn soon (for instance, top 10-20% risk scores). Medium risk might be the next 20-30%, and low risk are the rest who seem likely to stay. This segmentation allows focusing resources on those who need attention the most.

- **Tailored Retention Offers:** For each risk segment, define what action to take:

- *High Risk:* These customers should receive immediate and significant retention efforts. For example, the company might offer them special discounts, contract extensions with incentives, or free upgrades (like a premium channel package for free, or a waiver on next month's fee). The idea is to provide value that convinces them to stay. The model can even hint at *when* to intervene — e.g., if a customer is predicted to have a high hazard around the 12th month of tenure (maybe because their one-year contract is ending), the company could proactively reach out in month 11 with a renewal offer.

- *Medium Risk:* These customers are somewhat likely to churn, but not imminently. The company might use softer tactics for them, such as targeted communications highlighting new features, loyalty rewards, or checking in with customer service calls to address any issues. The timing is less urgent than the high group, but keeping an eye on them is important. Some medium-risk customers might be up-sell or cross-sell opportunities as well; since they are not on the brink of leaving, offering an upgrade might increase their engagement and loyalty (for instance, upgrading a DSL customer to fiber with a promotional rate might actually reduce churn risk if done preemptively).
- *Low Risk:* These are customers who the model predicts are very likely to stay (at least in the near term). The business can largely leave this group alone in terms of churn prevention, which saves resources. They might still be targets for upselling, but not for retention spend. However, it's wise to continue providing good service so they remain satisfied. Monitoring is minimal here.

- **Timing Considerations:** Survival analysis gives us a perspective on "when" churn is likely. The Next-Best-Action plan should consider timing signals. For example, suppose the model indicates a spike in churn hazard for month-to-month customers around month 1-2 (which we observed as a sharp early drop in survival). That means new customers on month-to-month plans are at risk of leaving almost immediately. An action item could be: within the first month, ensure high-touch customer service or onboarding for those customers to increase engagement. Another timing example: one-year contract customers might have higher churn right after their 12-month contract expires. So, starting at month 10 or 11, the company should begin communicating with those customers about contract renewal incentives. By aligning interventions with the churn timing, the company can preempt the churn rather than reacting after the fact.

- **Personalization and Channels:** The model's features could also guide *what* offer to present or *how* to communicate. For instance, if a high-risk customer has a complaint history or has called customer support multiple times (if such data was available, though not in our dataset, but imagine extending it), the next best action might be a personalized call from a retention specialist to resolve any lingering issues. If the customer is high-value (say high monthly charges or high CLTV), the retention offer might be more generous. The survival model integrated with CLTV (Customer Lifetime Value) scoring can help prioritize who to definitely keep versus who might not be cost-effective to retain with expensive offers.

- **Testing and Learning:** As part of the NBA strategy, it's important to treat it as a continuous learning process. We can deploy retention actions based on the model's predictions and then measure outcomes: did the customers who received offers churn less than similar customers who didn't? This can validate the effectiveness of the model and the interventions. Over time, collecting response data can even allow us to refine the next-best-action recommendations (for instance, perhaps we learn that offering a free addon service is more effective than a discount for a certain segment).

In summary, the Next-Best-Action strategy empowered by our survival model might look like: *Identify who is likely to churn soon, prioritize them, intervene with the right offer through the right channel, timed before they would leave.* For example, the model might flag a particular customer as high risk with an expected churn in 3 months; the NBA system could then queue a task for the retention team to call that customer within the next few weeks to check satisfaction and possibly offer a promotion. By doing this at scale, the company can significantly reduce its churn rate, focusing resources efficiently where they matter most.

It's worth noting that some of the insights from the survival analysis can directly inform broad preventive strategies too. For instance, since autopay users churn less, the company might universally encourage customers to switch to autopay (maybe by giving a one-time bill credit for signing up for autopay) – that could move more people into a lower-risk status. Similarly, the company could consider incentives for month-to-month customers to transition into longer contracts (like a discounted rate if they sign a one-year contract). These are broader policy decisions that complement the one-to-one next-best-action approach.

Deployment (Batch Scoring and FastAPI)

Building a predictive model is only part of the project – to be useful, it needs to be deployed in a way that the business can use it regularly. We implemented two modes of deployment for the churn survival model: **batch scoring** and a **real-time API**.

- **Batch Scoring Pipeline:** In batch mode, the model is used to score all customers on a scheduled interval (for example, every week or month). We developed a script (or notebook) that loads the latest customer data (which would include updated tenure for each customer, any new customers, and updated feature values), applies the survival model, and outputs the churn risk predictions. For the survival model, the output could be a risk score or a predicted probability of churn within a certain upcoming period (say churn probability in the next 3 months). These results are then stored or delivered to the relevant business units. In our case, we demonstrated batch scoring by running the model on the test dataset and outputting a file of risk scores. In a real company setting, this pipeline could write results into a database or a dashboard that retention marketing teams use. The batch process might also produce summary reports, such as lists of the top N high-risk customers in each region or segment, and maybe trigger automated emails for those accounts. The key aspect is that batch scoring ensures **all** customers are evaluated regularly and the company has an updated view of churn risk across the customer base.

- **FastAPI Real-Time Inference Service:** In addition to batch, we set up a real-time inference service using **FastAPI**, a modern Python web framework for building APIs. The idea behind the API deployment is to enable on-demand predictions – for example, if an individual customer's data is provided, the API will respond with that customer's churn risk prediction (and possibly their survival probability curve or expected time to churn). We created an API endpoint (e.g., an HTTP POST endpoint like /predict_churn_risk) that accepts a customer's feature data in JSON format. When a request is received, the API code loads our trained model (the CoxPH or GBM model) if not already loaded, uses it to predict the risk score or survival function for the given data, and returns the results.

We designed the FastAPI service with simplicity: upon startup, it loads the serialized model (we saved the trained model as a file after training – for example, XGBoost model saved as a binary .model file or a pickled Python object). Then, for each request, it parses the input features (contract type, senior status, etc.), applies the same preprocessing as we did during training (e.g., one-hot encoding for categorical features in the same way), and then calls the model's predict function. The response might be something like: {"churn_risk_score": 8.5, "churn_probability_12_months": 0.4} – these are just example outputs. The churn risk score is a relative number (where higher means riskier; not necessarily directly interpretable to a probability without context), but we can also convert model outputs into a probability of churn in a certain timeframe using survival curves.

This real-time API allows integration with other systems. For instance, a customer service application could call this API when a customer phones in – the agent can see on their screen the predicted churn risk and perhaps suggestions (like "High risk customer – consider offering retention deal"). It could also be used in a digital setting: e.g., when a customer logs into their account online, if the system identifies them as high risk, it might proactively show a loyalty offer or discount banner. The FastAPI approach ensures low-latency predictions (FastAPI is designed for performance) and can handle concurrent requests.

We tested the API locally using example data to ensure it returns the expected results. FastAPI also makes it easy to document the endpoint (it automatically provides a Swagger UI documentation where one can test the endpoint). Security and scalability considerations would be needed in a production environment – e.g., containerizing the app with Docker, using authentication for the API, and deploying it on a cloud service or on-premises server. We would also schedule the batch pipeline to run periodically (perhaps via a cron job or an Airflow DAG, etc.) to refresh predictions in bulk.

In summary, the deployment ensures that the churn model can be used in practice: the batch mode gives a broad periodic overview and feeds into marketing planning, while the FastAPI service enables real-time, individualized usage of the model. Together, they provide flexibility in how the business can leverage the churn predictions – whether it's a strategic campaign based on monthly risk reports or a dynamic intervention triggered in real-time for a specific customer.

Conclusion

In this project, we successfully developed a survival analysis-based approach to customer churn prediction and demonstrated how it can inform proactive churn prevention strategies. By treating churn as a time-to-event problem, we gained insights not just into *who* might leave, but *when* they are likely to leave, which is a critical dimension for timely intervention.

We started by transforming the IBM Telco Customer Churn dataset into a survival format, using customer tenure as the timeline and churn events as the outcome of interest. Exploratory analysis with Kaplan-Meier curves revealed clear differences in churn behavior across customer groups: for instance, customers on month-to-month contracts or not on autopay tend to churn much earlier and at higher rates than those on long-term contracts or on automatic payments. Senior citizens and fiber optic internet users were also identified as higher-risk segments. These findings make intuitive sense and provide actionable business knowledge (e.g., emphasize converting customers to longer contracts or autopay to improve retention).

Using the Cox Proportional Hazards model, we quantified the impact of multiple factors simultaneously, confirming that contract type and payment method (among others) are significant predictors of churn hazard. The Cox model offered interpretability – we could clearly see the hazard ratios for each feature – which can help explain the churn drivers to stakeholders. We then built a Gradient Boosting survival model which improved prediction accuracy by capturing complex patterns beyond the Cox model's scope. The GBM model, while less interpretable, offered better performance as evidenced by higher concordance and lift metrics.

Our evaluation on a test set showed that the models can effectively rank customers by risk and estimate churn probabilities with reasonable accuracy. For example, the high lift in the top-risk group means the company can dramatically increase the efficiency of retention efforts by focusing on those flagged by the model. In practical terms, instead of treating all customers as equally likely to churn, the company now has a way to prioritize the limited retention budget on those who matter most.

We outlined a Next-Best-Action plan that uses the model's output to guide retention marketing: high-risk customers get fast-tracked for generous retention offers, medium risks get targeted nurturing, and low risks are maintained with standard service. The timing of actions was also highlighted – an advantage of the survival approach is knowing *when* churn is likely, so interventions can be deployed preemptively (for instance, reaching out before a contract expires or during the onboarding period for new month-to-month customers).

Finally, we implemented deployment pipelines, demonstrating that the solution can be operationalized. A batch scoring process can keep churn risk scores updated for the entire customer base, and a FastAPI service provides on-demand predictions, making the model accessible to various applications (from dashboards to live customer interactions). This ensures the analytics can seamlessly translate into business processes, closing the loop from analysis to action.

In conclusion, the survival modeling approach provided a richer understanding of customer churn and a powerful tool to combat it. The project illustrates how a student data science project can integrate statistical methods with machine learning and deliver a practical solution. For future enhancements, one could incorporate additional data (such as customer support interactions or network usage patterns) to refine the model, or explore advanced techniques like time-varying covariates (if we had longitudinal data for each customer's behavior over time). Another avenue could be to implement an automated feedback loop: as retention actions are taken, measure their success and update the model or strategy accordingly (reinforcement learning for marketing policies, for example). Nonetheless, even with the current model, the telecom company can take more informed and timely actions to reduce churn, ultimately improving customer lifetime value and business sustainability. The project was a valuable learning experience in applying survival analysis to a real-world problem and showed the potential uplift a data-driven approach can achieve in customer retention.