Alfonso Hidalgo – Quantitative Treasury ALM Analyst

Understanding the Margin Call Recovery Probability Model

Abstract

This model estimates the probability that each pending margin call will be collected (fully paid) within a given time horizon. It uses dynamic historical data — tracking how exposures evolve through time — to identify when clients are likely to meet payment obligations. By combining timing information with account features, it provides a forward-looking view of recovery likelihoods.

---

1. The Big Picture

The **Margin Call Recovery Probability Model** predicts how likely each outstanding margin call is to be collected in the future. It focuses on the remaining amount owed (**GBP Outstanding**) rather than the total initially issued. Every record in the model represents a specific MC's situation at a given observation date, allowing the model to simulate real-world monitoring.

The model's outputs — probabilities of collection in 30, 60, 90, or 180 days — help risk managers anticipate which accounts will likely resolve their exposure soon, and which may require intervention.

---

2. Key Dates Explained

Term	Meaning	Role in Model
issued_date	When the margin call was first issued	Defines age_since_issue
event_date	When the account fully pays or resolves the outstanding amount	Defines the "event" (collection)
censor_date	The last date the account was observed without full payment	Marks the observation as censored (still unpaid)
cutoff_date	Training boundary for historical data	Ensures we only use data available before this point
eod_book (observation_date)	The snapshot date used to simulate a historical "look" at the account	Represents when we take a model reading
age_since_issue	Days between issue date and observation date	Measures how long the margin call has been active
observed_duration	Days from observation to the event or censor date	Defines the time until resolution or censoring

---

3. How the Model Learns and Predicts

The model is trained on a **landmark dataset**, which reconstructs historical "snapshots" of each margin call at different times. For every snapshot, it tracks whether the call was eventually paid within a future window (e.g., 365 days). Each row in this dataset represents a snapshot of a specific margin call on a given day, containing both its current status and its historical progress since issuance. For every snapshot, the model looks forward in time and checks whether the margin call was fully collected (paid) within the next 365 days:

- If it was, that record is marked as **is_event** = 1 (a collection event occurred).
- If it was not, the record is marked as **is_event** = 0 (censored — meaning still unpaid at the end of the observation window).

The time elapsed between the snapshot date and the event (or censoring) date is stored in **observed_duration**, which becomes the ***time-to-event*** label that the model learns from.

3.1 Model Inputs (Predictors)

The model uses a specific set of variables as predictors, defined in the code as:

```
features = ["risk_weight", "amount_band", "progress_ratio", "remaining_ratio",
"age_since_issue", "risk_x_amount"]
```

Feature	Meaning	Why It Matters
risk_weight	Quantifies the client's internal credit or risk category	Higher risk clients tend to have slower or less complete recoveries.
amount_band	Categorical band of the margin call size (e.g., Small, Medium, Large)	Larger calls are typically harder to collect quickly.
progress_ratio	Portion of the call already paid at the snapshot date	Accounts with higher progress are closer to completion.
remaining_ratio	Share of the call still outstanding (1 - progress_ratio)	Indicates remaining exposure and payment difficulty.
age_since_issue	Days elapsed since the margin call was issued	Older calls have different payment dynamics than newer ones.
risk_x_amount	Interaction term between risk and amount	Captures compounded effects – e.g., large risky accounts behave differently than small low-risk ones.

Categorical variables (like amount_band) are expanded into dummy columns (pd.get_dummies) so that the Cox model can assign separate coefficients to each category.

3.2 How the Model Learns

As we have already mentioned, the model is trained to estimate the instantaneous likelihood that a MC transitions from "unpaid" to "paid" at any given point in time, conditional on having survived (remained unpaid) until that moment. During fitting, the model learns the relative contribution (hazard ratio) of each predictor to the payment rate over time.

For example:

- A higher progress_ratio typically increases the likelihood of payment (faster collection).
- A higher risk_weight or remaining_ratio may decrease that likelihood (slower recovery).

3.3 How Predictions Are Produced

Once the model is trained on the landmark dataset, it can be applied to current live margin calls from *int_4_cl_mc_score*, which contain the same predictor fields.

For each active account, the model calculates a survival curve, S(t), describing the probability that the call remains unpaid up to time t.

Collection probabilities are then derived as: P(*collected by t*) = 1 - S(t)

This gives interpretable results such as:

- prob_30d → Probability the call will be collected within the next 30 days.
- prob_150d → Probability of collection within 150 days.
- prob_300d → Probability of collection within 300 days.

These probabilities are the foundation for all the business metrics, age-bucket charts, and expected recovery analyses displayed in your dashboard.

3.4 Future Model Enhancements

Although the current model already provides strong predictive insight, further refinements could improve performance and interpretability:

- Additional client behavioral indicators — such as number of previous margin calls.
- Market context variables — e.g., volatility indices or liquidity stress indicators at issuance time.
- Portfolio-level exposure measures — share of total exposure represented by each client.

3.5 Difference between observed_duration and age_since_issue

Variable	What It Represents	How It's Used
observed_duration	The label or outcome — how long it took (in days) from the snapshot date until the event (collection) or censoring.	Used by the model as the dependent variable (Y) in training.
age_since_issue	The feature — how old the margin call was (in days) at the time of the snapshot.	Used by the model as an independent variable (X) to explain differences in collection risk.

age_since_issue tells the model where in the life cycle the margin call currently is. ("This call has been open for 20 days so far.")

observed_duration tells the model how much longer it took before something happened. ("It was paid 40 days after this snapshot — so total duration from this point to event = 40 days.")

They are related but not the same:

- One describes the current state (input).
- The other describes the future outcome (target).

3.5.1 A Real Example

Let's take an actual account timeline to visualize it:

Date	Event	Derived Values
Jan 1	Margin call issued	age_since_issue = 0
Feb 1	Snapshot (model observation date)	age_since_issue = 31
Mar 15	Margin call fully collected	event_observed = 1, observed_duration = 42

So, for the snapshot on Feb 1, the model sees:

- Inputs (features):
    - age_since_issue = 31
    - risk_weight, amount_band, progress_ratio, etc.
- Label (target):
    - observed_duration = 42 (it took 42 more days until collection)
    - is_event = 1 (event occurred)

That's how the Cox model learns from many similar examples:

"Given that a call was already 31 days old and had these financial features, it was collected 42 days later — let's learn that pattern."

3.5.2 Why We Need Both

- If we only used observed_duration, the model wouldn't know where the account currently stands, it would just know how long it took to pay, without context.

- If you only used age_since_issue, we'd know how long it's been open, but not what happens afterward.

By using both:

- The model understands when in the call's life cycle it is observing the case (via age_since_issue), and
- It learns how much longer until payment occurs (via observed_duration).

This is what makes the landmark Cox model powerful — it learns conditional risk: "Given that the call has survived unpaid for "X" days already, what is the chance it will be paid soon?"

3.5.3 Why This Separation Matters in Survival Analysis

In survival modeling, we distinguish between:

- The survival time (observed_duration), outcome of interest (how long until event/censor).

- The covariates/features/predictors (age_since_issue, risk metrics, etc.), observed at the time of measurement.

That separation is essential because the model needs to handle time-dependent information: **The risk of payment changes as the margin call gets older**.

Including **age_since_issue** as a feature allows the model to explicitly capture those aging effects:

- Younger calls tend to be paid faster.
- Older ones might have slowed payment rates or become problematic.

Meanwhile, **observed_duration** remains the supervised signal that tells the model whether its assumptions about timing are correct.

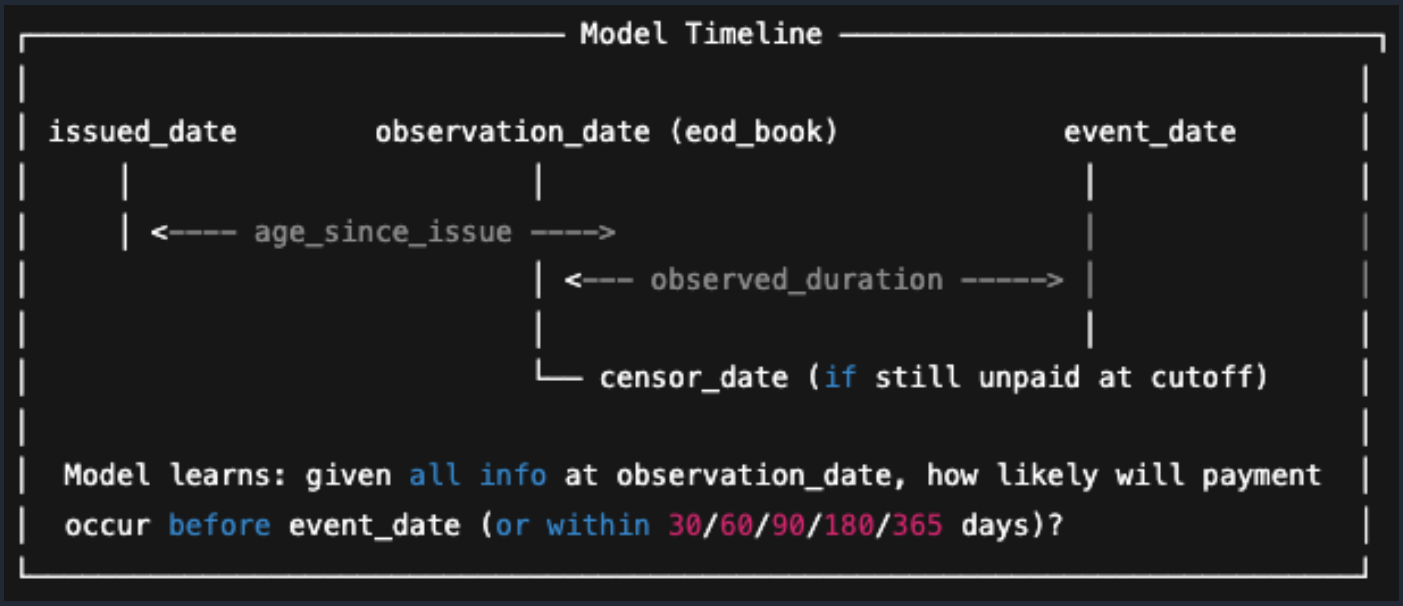
---

4. Why the Landmark Approach Matters

Unlike a static model that only looks once per account, as we saw previously, the landmark approach reuses every time slice of data. It recognizes that payment risk **changes over time** – for example, a margin call unpaid for 10 days behaves differently from one that's 200 days overdue. This approach captures that dynamic evolution and improves prediction realism.

---

5. Annex: Model Timeline Diagram



6. The Core Idea with Examples

The model is tracking, for every snapshot of an account, whether the margin call is paid (collected) within the next 365 days after that snapshot date. This is how we translate historical data into a predictive survival framework.

6.1 The Landmark Snapshot Logic

Each observation (row) in the landmark dataset represents a situation like this:

"On date X (the snapshot date, eod_book), account "A" had "Y" GBP still outstanding and "Z" days since issue."

From that point in time, we look forward in history and check what happened next.

We ask: "Did this margin call get fully collected within the next 365 days?"

- If yes → we record this snapshot as an event = 1 (it failed/succeeded within the window).
- If no → it becomes event = 0, censored = 1 (it was still open after that window ended).

This gives us pairs of:

- observed_duration (the number of days until the event or censoring).
- event_observed (1 if collected, 0 if still unpaid).

So each row answers: "From this point in time, how long until collection (if it happens at all)?"

6.2 How This Is Used in the Cox Model

The Cox Proportional Hazards model is not directly predicting "probability in 30 days." Instead, it estimates a hazard rate: i.e., how "likely" it is that the event (collection) will occur at any given time, conditional on the account not having been collected yet.

It does this by learning relationships between:

- the features (e.g. outstanding amount, progress ratio, age since issue, risk profile, etc.), and
- the timing of observed events.

Conceptually, the model learns something like: "Given that this margin call has these characteristics today, what is the relative likelihood it will be collected tomorrow (compared to others)?"

That's the hazard. Then, using math, the Cox model turns those hazards into survival probabilities: i.e., the probability the margin call remains unpaid up to a given day.

6.3 From Survival to Collection Probability

Once trained, we can query the model for each account in the live score table (*int_4_cl_mc_score*):

"Given today's account state, what's the chance it will still be unpaid after 30 / 60 / 90 / 180 / 365 days?"

The Cox model provides the survival function, S(t), which tells us: Probability (MC still unpaid at time t).

To get the collection probability, we take: P(*collected by t*) = 1–S(t)

So for example:

- 1–S(30) → probability of collection within 30 days
- 1–S(150) → probability of collection within 150 days
- 1–S(300) → probability of collection within 300 days

That's exactly how we get those columns: prob_30d, prob_150d, prob_300d, etc.

6.4 Why 365 Days Matter (Training Horizon)

When we built the training dataset, we limited the future "look-ahead" window to 365 days. That means the model never tries to infer collection beyond that horizon: it learns relationships up to a one-year window.

So during training, each snapshot's "label" (event vs. censor) is determined by what happens within that 365-day look-ahead period. When scoring, we can then safely compute predicted probabilities for 30, 60, 90, 180, and 365 days because the model has seen data covering that entire range.

6.5 Putting It All Together (Conceptual Summary)

Step	What the Model Sees	What It Learns	What You Get
1	"Account/MC "A", snapshot on "Feb 1st" – 20 days since issue, £500k outstanding."	Did it get collected within the next 365 days?	Learns mapping between features & collection timing.
2	Repeats for thousands of similar snapshots across time and accounts.	Learns how hazard (collection risk) changes with features and age.	Builds an internal time-based risk function.
3	You give it today's live data.	Predicts survival curve for each account.	Converts survival into collection probabilities per time horizon.

6.6 Intuitive Analogy

We are watching many margin calls evolve through time.

Each day, we note: "At this point, it's "X" days old, "Y%" paid, "Z" risk score."

We record whether each one eventually gets paid soon after or not.

After seeing thousands of examples, the model learns what combinations of features usually lead to collection soon versus staying unpaid longer.

When we give it today's pending cases, it recognizes similar patterns and tells us: "Based on its current state and age, this one looks like the type that usually gets paid within the next 90 days."

That's what the "probability of collection in the next "X" days" really means.