11) Upsell Propensity — GBM/Logit with profit guardrail

Goal & decision

Offer the **next-tier / add-on** only when it **increases net GM per order without** hurting base checkout conversion.

Feature sources

- **carts** (online): `items[]` , `subtotal_cents` , `hero_product_id`
- **upsell_catalog** (mapping): `hero → {candidates, price_delta, margin_cents}`
- **orders / offers_log**: past upsell **shows/accepts**, base conversion outcomes
- **products**: `margin_cents` , compatibility flags

Compact formula

$\hat{p}_u = \sigma(\mathbf{w}^\top \mathbf{x})$. **Show upsell iff**

$$\hat{p}_u\, m_u \;-\; \lambda\, \Delta p_{\text{base}}\, \text{baseGM} \;>\; 0$$

Where: m_u= upsell **margin** (cents); Δp_{base}= predicted **drop** in base conversion; $\lambda \geq 1$= risk aversion; baseGM= expected base order GM (cents).

Full expansion (worked)

Given $\hat{p}_u = 0.15$, $m_u = 1{,}800\,\text{cents}$, $\Delta p_{\text{base}} = 0.004$, $\text{baseGM} = 10{,}000\,\text{cents}$, $\lambda = 1.0$:

- **Expected upsell GM:** $\hat{p}_u\, m_u = 0.15 \times 1{,}800 = \mathbf{270}\,\text{cents}$
- **Penalty (base risk):** $\lambda\, \Delta p_{\text{base}}\, \text{baseGM} = 1.0 \times 0.004 \times 10{,}000 = \mathbf{40}\,\text{cents}$
- **Decision score:** $270 - 40 = \mathbf{230}\,\text{cents} > 0 \Rightarrow \mathbf{SHOW}$

(If multiple candidates exist, pick the max positive score subject to compatibility & category caps.)

Result & steering

- **Show only** candidates with **positive net** decision score.
- If base conversion dips in a canary, **autopause** or **raise** λ.
- Enforce **compatibility**, **min-margin**, and **no-substitute** rules.

▼ **metrics: upsell_propensity_example (with feature_sources)** — click to expand

```
metrics:
  upsell_propensity_example:
    inputs:
      p_accept_hat: 0.15
      upsell_margin_cents: 1800
      base_conversion_drop: 0.004
      base_order_gm_cents: 10000
      lambda_risk_aversion: 1.0
    calculations:
      expected_upsell_gm_cents: 270.0        # 0.15 * 1800
      penalty_cents: 40.0                     # 1.0 * 0.004 * 10000
      decision_score_cents: 230.0             # 270 - 40
    decision: "SHOW"                          # > 0

feature_sources:
  - mongo.carts
  - mongo.upsell_catalog
  - mongo.orders
  - mongo.products
  </details>
```


12) Cross-sell — Graph-embedding complements with margin/stock re-rank

Goal & decision

Recommend **complements** (not substitutes) that **attach with profit** for the current **hero** item (PDP/Cart).

Feature sources

- **product_graph** (derived): item embeddings v_i, co-buy **lift**
- **products**: `margin_cents` , `stock_cover_days` , `category_id` , `role`
- **carts** (online): `hero_product_id`

Compact formula

Convert signals → attach probability → **net value**:

- Latent score: $z_i = \beta_0 + \beta_1 \cos\big(v_H, v_i\big) + \beta_2\, \mathrm{lift}_{H,i} - \beta_3\, \mathbf{1}\{\mathrm{subst}(i,H)\}$
- Probability: $\hat{p}_{\mathrm{attach},i} = \sigma(z_i)$
- Net value: $\mathrm{Net}_i = \hat{p}_{\mathrm{attach},i}\, m_i - \alpha\, \mathrm{LSP}_i$
- Low-stock penalty: $\mathrm{LSP}_i = \max\big(0, \frac{7-\mathrm{cover}_i}{7}\big)\, m_0$ (e.g., $m_0 = 1000$ cents)

Show **top-N** complements by Net_i under caps; **exclude substitutes**.

Full expansion (worked)

Weights: $\beta_0 = -2.0,\ \beta_1 = 2.0,\ \beta_2 = 0.8,\ \beta_3 = 5.0,\ \alpha = 1.0,\ m_0 = 1000\ \text{cents.}$
Hero: H. **Candidates:** C1, C2, C3.

Inputs

- C1: $\cos = 0.72,\ \text{lift} = 1.8,\ m = 1200\ \text{c},\ \text{cover} = 10,\ \text{subst} = 0$
- C2: $\cos = 0.85,\ \text{lift} = 1.2,\ m = 1400\ \text{c},\ \text{cover} = 5,\ \text{subst} = 1$
- C3: $\cos = 0.60,\ \text{lift} = 1.5,\ m = 900\ \text{c},\ \text{cover} = 8,\ \text{subst} = 0$

Compute z, $\hat{p}$, LSP, Net (using $\sigma(z) = 1/(1+e^{-z})$)

$$\begin{aligned}
\textbf{C1:}\quad & z = -2 + 2(0.72) + 0.8(1.8) - 5(0)\ =\ 0.88, \\
& \hat{p} = \sigma(0.88) = \mathbf{0.70682222}, \qquad \text{LSP} = \max\!\left(0, \tfrac{7-10}{7}\right) 1000 = 0, \\
& \text{Net} = 0.70682222 \cdot 1200 - 0\ =\ \mathbf{848.19}\ \text{cents.}
\end{aligned}$$

$$\begin{aligned}
\textbf{C2:}\quad & z = -2 + 2(0.85) + 0.8(1.2) - 5(1)\ =\ -4.34, \\
& \hat{p} = \sigma(-4.34) = \mathbf{0.01286876}, \qquad \text{LSP} = \max\!\left(0, \tfrac{7-5}{7}\right) 1000\ =\ \mathbf{285.7143}\ \text{cents}, \\
& \text{Net} = 0.01286876 \cdot 1400 - 285.7143\ =\ -\mathbf{267.70}\ \text{cents}\ \ \text{(exclude).}
\end{aligned}$$

$$\begin{aligned}
\textbf{C3:}\quad & z = -2 + 2(0.60) + 0.8(1.5) - 5(0)\ =\ 0.40, \\
& \hat{p} = \sigma(0.40) = \mathbf{0.59868766}, \qquad \text{LSP} = 0, \\
& \text{Net} = 0.59868766 \cdot 900\ =\ \mathbf{538.82}\ \text{cents.}
\end{aligned}$$

Select: top positive nets under caps ⇒ **[C1, C3]**.

Result & steering

- **Exclude substitutes** (e.g., $\cos \geq 0.80$ in same role/category).
- **Penalize low stock** via LSP; enforce **category diversity** in top-N.
- If embeddings are **sparse**, fall back to **association rules** / attribute similarity.

▼ **metrics: cross_sell_example (with feature_sources)** — click to expand

```yaml
metrics:
  cross_sell_example:
    weights:
      beta0: -2.0
      beta1_cosine: 2.0
      beta2_lift: 0.8
      beta3_substitute_penalty: 5.0
      alpha_low_stock: 1.0
      m0_low_stock_penalty_cents: 1000
    hero: H1
    candidates:
      - { id: C1, cosine: 0.72, lift: 1.8, margin_cents: 1200, stock_cover_days: 10, is_substitute: 0 }
      - { id: C2, cosine: 0.85, lift: 1.2, margin_cents: 1400, stock_cover_days: 5,  is_substitute: 1 }
      - { id: C3, cosine: 0.60, lift: 1.5, margin_cents: 900,  stock_cover_days: 8,  is_substitute: 0 }
    calculations:
      C1: { z: 0.88, p_attach: 0.70682266, lsp_cents: 0.0,     net_cents: 848.19 }
      C2: { z: -4.34, p_attach: 0.01286876, lsp_cents: 285.71, net_cents: -267.70 }
      C3: { z: 0.40, p_attach: 0.59868766, lsp_cents: 0.0,     net_cents: 538.82 }
    selected: [ C1, C3 ]

feature_sources:
  - mongo.product_graph
  - mongo.products
  - mongo.carts
</details>
```


13) Bundle Optimizer — Knapsack/MILP (slow-mover + hero)

Goal & decision

Create bundles pairing **slow-movers** with **hero** SKUs to raise **sell-through** and **net GM**, while **bounding cannibalization**.

Feature sources

- **inventory**: slow-mover list, `stock_units`
- **products**: `margin_cents` , brand rules
- **attach_priors**: bundle **attach rates** (π^{attach}), **cannibalization** rates (π^{cann})
- **Output**: `discounts.bundle_definitions[]` (pairs + constraints)

Compact formula

Score (expected net value) for pair (s, h):
$\text{Score}_{s,h} = \pi^{\text{attach}}_{s,h}(m_s + m_h) - \pi^{\text{cann}}_{s,h}\, m_h - \text{discCost}_{s,h}$

Decision (knapsack/MILP): choose $y_{s,h} \in \{0,1\}$ to
$\max \sum_{(s,h)} y_{s,h}\, \text{Score}_{s,h}$
s.t. **stock**, **brand**, and **max-bundles-per-hero** constraints.

Full expansion (worked)

Given (margins in cents):
slow movers: $S1 = 600,\ S2 = 800,\ S3 = 500$
heroes: $H1 = 1500,\ H2 = 1200$

Priors & penalties (attach, cannibal; $\mathrm{discCost} = 0$ in toy):

- $S1{-}H1:\ \pi^{\text{attach}} = 0.06,\ \pi^{\text{cann}} = 0.01$
  $\text{Score} = 0.06(600 + 1500) - 0.01 \cdot 1500 = 126 - 15 = \mathbf{111}$
- $S2{-}H1:\ \pi^{\text{attach}} = 0.08,\ \pi^{\text{cann}} = 0.02$
  $\text{Score} = 0.08(800 + 1500) - 0.02 \cdot 1500 = 184 - 30 = \mathbf{154}$
- $S3{-}H2:\ \pi^{\text{attach}} = 0.07,\ \pi^{\text{cann}} = 0.01$
  $\text{Score} = 0.07(500 + 1200) - 0.01 \cdot 1200 = 119 - 12 = \mathbf{107}$
- $S2{-}H2:\ \pi^{\text{attach}} = 0.05,\ \pi^{\text{cann}} = 0.015$
  $\text{Score} = 0.05(800 + 1200) - 0.015 \cdot 1200 = 100 - 18 = \mathbf{82}$
- $S1{-}H2:\ \pi^{\text{attach}} = 0.04,\ \pi^{\text{cann}} = 0.008$
  $\text{Score} = 0.04(600 + 1200) - 0.008 \cdot 1200 = 72 - 9.6 = \mathbf{62.4}$

Constraints (toy): max **1 bundle per hero**, $\mathrm{stock}(S1, S2, S3) = (50, 30, 40)$.
Optimal pick: S2–H1 (154) and **S3–H2 (107)**.

Result & steering

- Emit those **bundle definitions**; enforce **brand rules** and **min hero GM**.
- **Re-estimate priors weekly**; drop bundles whose realized **net GM** falls below floor.

▼ **metrics: bundle_optimizer_example (with feature_sources)** — click to expand

```yaml
metrics:
  bundle_optimizer_example:
    margins_cents:
      slow_movers: { S1: 600, S2: 800, S3: 500 }
      heroes:      { H1: 1500, H2: 1200 }
    priors:
      attach_rate:
        S1_H1: 0.06
        S2_H1: 0.08
        S3_H2: 0.07
        S2_H2: 0.05
        S1_H2: 0.04
      cannibal_rate:
        S1_H1: 0.01
        S2_H1: 0.02
        S3_H2: 0.01
        S2_H2: 0.015
        S1_H2: 0.008
    discount_cost_cents: { S1_H1: 0, S2_H1: 0, S3_H2: 0, S2_H2: 0, S1_H2: 0 }
    scores_cents:
      S1_H1: 111.0
      S2_H1: 154.0
      S3_H2: 107.0
      S2_H2:  82.0
      S1_H2:  62.4
    constraints:
      max_bundles_per_hero: 1
      stock_units: { S1: 50, S2: 30, S3: 40 }
    selected_bundles: [ S2_H1, S3_H2 ]

feature_sources:
  - mongo.inventory
  - mongo.products
  - mongo.attach_priors

</details>
```


14) Churn Propensity (risk scoring) — GBM/Logit

Goal & decision

Flag **at-risk customers** for **targeted retention** when the **expected net** of the action is **positive**.

Feature sources

- **customers**: tenure, first/last purchase, engagement (active days)
- **orders**: recent frequency & margin (e.g., last 90 d)
- **events**: visits/email opens (to build E)
- **support** *(optional)*: service events
- **Output**: `users.churn_risk`

Compact formula

Binary churn probability $\hat{p}_{\text{churn}} = \sigma(\mathbf{w}^\top \mathbf{x})$. **Treat iff**

$$\hat{p}_{\text{churn}} \cdot g \;-\; c \;>\; 0$$

Where: g= expected **margin saved** by action (cents); c= **action cost** (cents).

Full expansion (worked)

Excel-friendly features:
$R_w = 10$ (recency in weeks), $F_{90} = 2$ (orders last 90 d), $\ln M_{90} = \ln(30{,}000) \approx 10.31$, $T_m = 12$ (tenure months), $E_{30} = 6$ (active days last 30 d).

Weights: $w_0 = -1.2,\ w_R = 0.10,\ w_F = -0.30,\ w_M = -0.08,\ w_T = -0.02,\ w_E = 0.05.$

Linear score and probability

$$\begin{aligned} z &= w_0 + w_R R_w + w_F F_{90} + w_M \ln M_{90} + w_T T_m + w_E E_{30} \\ &= -1.2 + 0.10 \cdot 10 - 0.30 \cdot 2 - 0.08 \cdot 10.31 - 0.02 \cdot 12 + 0.05 \cdot 6 \\ &= \mathbf{-1.5648} \\ \hat{p}_{\text{churn}} = \sigma(z) &= \frac{1}{1+e^{-z}} = \frac{1}{1+e^{1.5648}} \approx \mathbf{0.1729} \end{aligned}$$

Intervention economics (cents): $g = 1{,}500,\ c = 200$

$$\text{Decision value} = \hat{p}_{\text{churn}} \cdot g - c = 0.1729 \cdot 1500 - 200 = \mathbf{59.35} \;>\; 0 \;\Rightarrow\; \textbf{TREAT}$$

Threshold intuition: treat if $\hat{p}_{\text{churn}} > c/g = 200/1500 = \mathbf{0.133}$.

Result & steering

- **Rank** customers by **net value**; **cap daily contacts**; **suppress** if recent negative response.
- **Re-calibrate quarterly** (seasonality/cohorts); audit precision-recall and cost curves.

▼ **metrics: churn_propensity_example (with feature_sources)** — click to expand

```yaml
metrics:
  churn_propensity_example:
    features:
      recency_weeks: 10
      freq_last_90d: 2
      ln_margin_90d: 10.31
      tenure_months: 12
      active_days_30d: 6
    weights:
      w0: -1.2
      w_recency: 0.10
      w_freq: -0.30
      w_ln_margin: -0.08
      w_tenure: -0.02
      w_engagement: 0.05
    calculations:
      z_score: -1.5648
      p_churn: 0.1729
      action_gain_cents: 1500
      action_cost_cents: 200
      decision_value_cents: 59.35
    decision: "TREAT"

feature_sources:
  - mongo.customers
  - mongo.orders
  - mongo.events
  - mongo.support
  </details>
```


15) Lifecycle Timing — HSMM (explicit state durations) + optional Hawkes

Goal & decision

Detect lifecycle **state transitions** (**Active** → **At-Risk** → **Lapsed**) to **time win-backs** and **suppress waste**.

Feature sources

- **events** (weekly aggregates): visits, email opens/clicks
- **orders** (weekly): purchases, spend
- **Output**: `users.lifecycle_state` , `users.state_probs[]`

Compact formula (HSMM forward)

States $s \in \{A,\ AR,\ L\}$ with **duration pmf** $P(D = d \mid s)$ and **emission** $f_s(o_t)$. The **forward** mass at time t in state s with **current duration** d:

$$\alpha_t(s,d) = \left[\sum_{s'} \alpha_{t-d}(s')\, A_{s'\to s}\right]\, P(D{=}d \mid s) \prod_{k=t-d+1}^{t} f_s(o_k)$$

Posterior (requires normalization):

$$P(S_t = s) = \frac{\sum_{d=1}^{t} \alpha_t(s,d)}{\sum_{u\in\{A,AR,L\}} \sum_{d=1}^{t} \alpha_t(u,d)}$$

Optional Hawkes (time-to-next purchase): $\lambda(t) = \mu + \sum_{t_i<t} \alpha\, e^{-\beta(t-t_i)}$.

Full expansion (worked)

Toy HSMM settings (weekly): Durations (Poisson means): $\mu_A = 8,\ \mu_{AR} = 3,\ \mu_L = 12$.
Emissions (visits/week, Poisson): $A : \lambda = 3,\ AR : \lambda = 1,\ L : \lambda = 0.2$.
Observed visits (weeks 1–4): $[3, 2, 1, 0]$.

We compute **two AR durations** that could end at **week 4**.

1) AR with $d = 2$ (weeks 3–4 under AR)

- Emissions: $P(1;1) = e^{-1} \cdot 1 = 0.3679,\ P(0;1) = e^{-1} = 0.3679$ → product $= \mathbf{0.1353}$
- Duration pmf: $P(D{=}2 \mid AR) = e^{-3} \cdot 3^2/2!\ =\ \mathbf{0.2240}$
- Transition mass into AR at $t - d = 2$: $\sum_{s'} \alpha_2(s')A_{s'\to AR} = \mathbf{0.4}$ (toy)
- Forward mass: $\alpha_4(AR, 2) = 0.4 \times 0.2240 \times 0.1353 = \mathbf{0.0121}$

2) AR with $d = 3$ (weeks 2–4 under AR)

- Emissions: $P(2;1) = 0.1839,\ P(1;1) = 0.3679,\ P(0;1) = 0.3679$ → product $= \mathbf{0.0249}$
- Duration pmf: $P(D{=}3 \mid AR) = \mathbf{0.2240}$
- Transition mass into AR at week 1: $\mathbf{0.5}$ (toy)
- Forward mass: $\alpha_4(AR, 3) = 0.5 \times 0.2240 \times 0.0249 = \mathbf{0.0028}$

Aggregate AR forward mass at week 4: $\alpha_4(AR) = 0.0121 + 0.0028 = \mathbf{0.0149}$.

> **Note:** To claim a trigger at θ (e.g., 0.60), we must compute the **normalized posterior**
> $P(S_4 = AR) = \alpha_4(AR)/\sum_{s\in\{A,AR,L\}} \alpha_4(s)$.
> Without $\alpha_4(A)$ and $\alpha_4(L)$, the trigger **cannot** be asserted.

Trigger rule (operational): fire **At-Risk win-back** when $P(S_t = \mathrm{AR}) \ge \theta$ for at least d^* consecutive weeks (e.g., $\theta = 0.60,\ d^* = 2$). Suppress if $S_t = \mathrm{L}$ and no win-back is active.

Result & steering

- Use **state** and **time-in-state** to schedule messages; integrate with **promotion uplift** for offer choice.
- If HSMM is heavy in Excel, **approximate** with a **logistic gate** on recency + visit-slope; adopt HSMM in code later.

▼ **metrics: lifecycle_hsmm_example (with feature_sources)** — click to expand

```yaml
metrics:
  lifecycle_hsmm_example:
    states: [A, AR, L]
    duration_poisson_means:
      A: 8
      AR: 3
      L: 12
    emission_poisson_lambdas_visits:
      A: 3.0
      AR: 1.0
      L: 0.2
    observations_weekly_visits: [3, 2, 1, 0]   # weeks 1..4
    toy_transition_mass_into_AR:
      at_week_2: 0.4
      at_week_1: 0.5
    durations_considered:
      - { state: AR, d: 2, emission_product: 0.1353, P_D: 0.2240, contrib_alpha: 0.0121 }
      - { state: AR, d: 3, emission_product: 0.0249, P_D: 0.2240, contrib_alpha: 0.0028 }
    alpha_AR_week4_total: 0.0149
    trigger_rule:
      threshold_theta: 0.60
      min_consecutive_periods: 2
    outcome: "NEEDS_POSTERIOR_NORMALIZATION"   # compute alpha_A and alpha_L to decide

feature_sources:
  - mongo.events
  - mongo.orders
```


</details>