Final exam Review

DS-6030 | Spring 2026

review-final.pdf

Table of contents

1 Ensembles and Stacking (HW 5, Quiz 5)

1.1 Ensemble Taxonomy

Method	Fitting	Base learners	Combining rule	Primary benefit
Bagging	Parallel	Same algorithm, bootstrap samples	Simple average	Variance ↓
Random Forest	Parallel	Trees + feature subsampling	Simple average	Variance ↓ (more)
CV Committee	Parallel	Same algorithm, CV folds	Simple average	Uses all data
BMA	Parallel	Different model specs	BIC/AIC weighted linear model	Model uncertainty
Linear Stacking	Parallel	Different families	Linear Model	Bias + variance
Non-linear Stacking	Parallel	Different families	Non-linear meta-learner	More flexible
Boosting	Sequential	Same families	Additive	Lower bias

1.2 Coverage Table

Legend: L = `ensembles.pdf` ·✓= covered · — = not covered

Topic	Lecture	HW5	Notes
Base learner variety: model families, tuning params, data splits	✓	✓	Three axes of diversity
Bagging: bootstrap resampling, average	✓	—	Variance reducer
CV Committee vs. model selection	✓	—	Average, don't select
BMA: posterior weights from BIC/AIC	✓	—	Lightly covered
Linear stacking form: $\hat{f}(x) = \sum_k a_k \hat{g}_k(x)$	✓	✓	
Single hold-out stacking algorithm	✓	—	Step-by-step in lecture
CV stacking algorithm	✓	✓	Applied in HW5
Stacking features: $Z_{ik} = \hat{g}_k(x_i \mid \mathcal{D}_{-\text{fold}})$	✓	—	Out-of-sample predictions as X for meta-learner
Non-linear stacking (RF, boosted trees as meta-learner)	✓	—	More flexible, more overfit risk
Why out-of-sample predictions matter	✓	—	Prevents leakage to meta-learner
Diversity among base learners and ensemble performance	✓	—	Core bias-variance insight

1.3 Key Ideas

Linear Stacking and Model Averaging:

$$\hat{f}(x) = \sum_{k=1}^{K} a_k \hat{g}_k(x) \qquad \text{weights } a_k \text{ fit by meta-learner on } Z_{ik}$$

- $Z_{ik} = \hat{g}_k(x_i)$ = out-of-sample prediction from base model k for observation i
- Meta-learner input: matrix of base-model predictions; meta-learner output: final prediction
- Simple meta-learner (constrained linear regression, ridge) often outperforms complex — why?

Why out-of-sample?

Diversity sources (three axes):

1. Base model family (trees vs. linear vs. SVM)
2. Base model complexity tuning (deep vs. shallow trees)
3. Data (bootstrap, different feature subsets)

Stacking vs. Calibration:

1.4 Quiz Questions (True/False — justify each)

1. Boosting is a type of ensemble where components are fit in **parallel**.

2. Random Forests can be considered a **boosting** model.

3. In boosted trees, the number of trees is an important tuning parameter.

4. Stacking is **guaranteed** to outperform the best individual base learner on held-out data if base learners are diverse.

5. Stacking can naturally combine fundamentally different model types (RF, SVM, neural net) within the same ensemble.

1.5 Discussion Questions

1. What properties of base learners lead a stacked ensemble to outperform the best individual model?

2. How does diversity affect the bias-variance trade-off in **stacking** differently than in **bagging**?

3. What is the stacking meta-learner actually learning: correcting bias, reducing variance, or both?

4. Why might a simple linear meta-learner sometimes outperform a complex non-linear one?

2 Gradient Boosting (HW 5, Quiz 5)

2.1 Boosting vs. Bagging

Feature	Bagging / RF	Gradient Boosting
Fitting order	Parallel (independent)	Sequential (each tree corrects last)
Primary effect	Variance ↓	Bias ↓
Base learner complexity	Can be deep	Usually shallow (depth 1–6)
Overfit risk	Low (more trees = better)	High (num trees must be tuned)
Key tuning params	`mtry`, `min.node.size`	`n.trees`, `shrinkage`, `interaction.depth`, `bag.fraction`

2.2 Coverage Table

Legend: L = `boosting.pdf` ·✓= covered · — = not covered

Topic	Lecture	HW5	Notes
Sequential vs. parallel ensembles	✓	—	Core distinction
Boosting as bias reducer	✓	—	Contrast with bagging
General form: $\hat{F}_M(x) = \sum_{m=1}^{M} \nu \hat{f}_m(x)$	✓	—	
L2 Boosting: pseudo-residuals $r_i = y_i - \hat{F}_{m-1}(x_i)$	✓	—	Step-by-step in lecture
Update rule: $\hat{F}_m = \hat{F}_{m-1} + \nu \hat{f}_m$	✓	—	
Shrinkage / learning rate ν: role and tradeoff	✓	✓	Smaller $\nu \rightarrow$ need more trees
`n.trees`: must be tuned (unlike RF)	✓	✓	Critical point
`interaction.depth`: controls tree complexity	✓	✓	Bias tuner
`bag.fraction` (subsampling): stochastic boosting	✓	✓	Variance + speed
XGBoost extras: `lambda` (L2), `alpha` (L1), `gamma` (pruning), `colsample`	✓	✓	
Taylor / Newton boosting: 2nd order approximation	✓	—	Key innovation for modern boosting models
AdaBoost / LogitBoost	lightly	—	**Not a main exam topic**

2.3 Key Ideas

L2 Boosting algorithm (3 steps per iteration m):

1. Compute pseudo-residuals: $r_i = y_i - \hat{F}_{m-1}(x_i)$
2. Fit tree $\hat{f}_m$ to the residuals
3. Update: $\hat{F}_m(x) = \hat{F}_{m-1}(x) + \nu \hat{f}_m(x)$

Learning rate (shrinkage) ν and `n.trees` — the key interaction:

What each tuning parameter controls:

Parameter	Controls	Bias/Variance
`n.trees`	Number of iterations	↑ trees → overfit if ν too large
`shrinkage` (ν)	Step size per iteration	Smaller = more stable, need more trees
`interaction.depth`	Tree depth (complexity of each step)	Deeper = more bias reduction per step
`bag.fraction`	Subsampling fraction	< 1 adds noise (can help generalization)
`lambda` / `alpha`	L2 / L1 penalty on leaf weights (XGBoost)	Regularization
`gamma`	Minimum gain to split (XGBoost)	Tree pruning

2.4 Quiz Question

True or False: When implementing gradient boosting, if your model is **underfitting**, decreasing the learning rate while holding `n.trees` fixed will generally improve performance.

2.5 Discussion Questions

1. How would you diagnose whether a boosted model is overfitting vs. underfitting using a validation curve?

2. A colleague sets `shrinkage = 0.001` and `n.trees = 10`. What do you predict will happen?

3. What parameter would you increase to give a GBM model more capacity to fit complex interactions?

4. You have an XGBoost model and want to reduce overfitting without adding data. Which parameters could help, and how?

3 Predictive Diagnostics (HW 6, Quiz 6)

3.1 Two Diagnostic Lenses

Diagnostic	Condition on	Detects	HW6 example
Residual analysis	Features X	Structural miss: missing variables, wrong functional form	Pearson residuals vs. `shot_distance`, `shooter_skill`, etc.
Calibration analysis	$\hat{p}(x)$	Scale miscalibration: predictions too extreme, wrong intercept	Observed rate vs. predicted rate plot
Discrimination	(rank-based)	Ability to separate high-risk from low-risk	AUC / C-index

Key insight: These lenses are independent. A model can pass one and fail the other.

3.2 Coverage Table

Legend: L = lecture notes ·✓= covered · — = not covered

Topic	Lecture	HW6	Notes
Overall EPE, CI for performance estimates	✓	—	
Performance heterogeneity across feature space	✓	P1–P3	Overall metrics mask subgroup failures
Pearson residuals: $r_i = (y_i - \hat{p}_i)/\sqrt{\hat{p}_i(1-\hat{p}_i)}$	✓	P1–P3	Used for all HW6 diagnostics
Residual analysis: plot r_i vs. each feature x_j	✓	P1–P3	Smoother reveals systematic pattern
Calibration plot: observed rate vs. $\hat{p}(x)$	✓	P1–P3	45° line = perfect calibration
Overfit signature: S-shaped calibration curve	✓	P2	Predictions too extreme
Underfit signature: flat or compressed calibration	✓	P3	Predictions not spread enough
Calibration vs. discrimination: good AUC + bad calibration	✓	P2, P3	Core conceptual distinction
Distribution shift: same model, new population (P4)	✓	P4	Residuals + calibration both shift
Recalibration / remediation (P5)	✓	P5	Refit or rescale predictions
Link to stacking: calibration = single-model stacking	✓	—	Lecture connection

Topic	Lecture	HW6	Notes
Link to boosting: residuals as target for next model	✓	—	L2 boosting motivation

3.3 Key Ideas

Pearson residuals (binary outcomes):

$$r_i = \frac{y_i - \hat{p}(x_i)}{\sqrt{\hat{p}(x_i)(1-\hat{p}(x_i))}} \qquad E[r_i] = 0,\ V[r_i] = 1 \text{ if model is correct}$$

Three failure modes and their signatures:

Model	Residual plot	Calibration plot	Implication
Good	Flat at 0 across features	Points on 45° line	No systematic error
Overfit	Noisy, may have local patterns	S-shaped: extremes too extreme	Shrink predictions toward mean
Underfit	Systematic trend vs. features	Compressed: predictions not spread	Need more complex model or more features

Estimate Performance $\neq$ True Performance :

- subgroup size

Distribution shift (P4):

Calibration $\neq$ Discrimination:

3.4 Quiz Questions

Q1. A calibration plot shows an **S-shaped** pattern (low predictions too low, high predictions too high). Which of the following are likely true?

- (a) The model is overfit.
- (b) Recalibrating (shrinking predictions toward the mean) could fix this.
- (c) The model's AUC is poor.

Q2. Residual analysis conditions on X; calibration conditions on $\hat{f}(x)$. These can give different conclusions. Why?

- (a) Residual analysis can look fine even when calibration is poor, because the miscalibration is spread evenly across feature space.
- (b) Calibration can look fine even when residuals show patterns, because feature-level errors cancel out when grouped by $\hat{p}$.

3.5 Discussion Questions

1. HW6 P2: The overfit model has good AUC but an S-shaped calibration curve. Explain why AUC is unaffected by scale miscalibration.

2. HW6 P4: You apply the "good" model to playoff data and diagnostics look bad. Is the model wrong, or has something changed? How would you decide?

3. How is residual analysis connected to the idea of L2 boosting?

4. How is calibration connected to the stacking?

4 Quantile Regression and Prediction Intervals (HW 7, Quiz 7)

4.1 Targets and Tools

Target	Method	Loss function	Output
$E[Y \mid X = x]$ (conditional mean)	OLS / regression	Squared error	Point prediction
$Q_\tau(Y \mid X = x)$ (conditional τ-quantile)	Quantile regression	Pinball loss	Quantile prediction
80% Prediction Interval	QR at $\tau = 0.10$ and $\tau = 0.90$	Pinball (each separately)	Interval $[\hat{q}_{0.10}(x),\ \hat{q}_{0.90}(x)]$
80% Confidence Interval	Bootstrap or parametric	—	Interval for $E[Y \mid X]$, not for Y

Key distinction: A PI covers a **new observation**; a CI covers the **conditional mean**.

4.2 Coverage Table

Legend: L = lecture notes ·✓= covered · — = not covered

Topic	Lecture	HW7	Notes
Quantile definition: $Q_\tau = \inf\{q : F(q) \geq \tau\}$	✓	—	
Pinball (check) loss: $L_\tau(u) = u(\tau - \mathbf{1}(u < 0))$	✓	—	$u = y - \hat{q}_\tau$
Median ($\tau = 0.5$) minimizes MAE	✓	P2b	Theoretical target
Quantile regression: linear and tree-based	✓	P2b,c	Applied in HW
PI from paired quantile regressions	✓	P2c	80% PI needs $\tau = 0.10$ and 0.90
CI vs. PI distinction	✓	—	CI $\neq$ PI
Empirical coverage assessment	✓	P2d	Compare nominal vs. actual
Quantile random forests	✓	—	Same leaf structure idea as survival forests

4.3 Key Ideas

Pinball loss for quantile τ:

$$L_\tau(u) = \begin{cases} u \cdot \tau & \text{if } u \geq 0 \\ u \cdot (\tau - 1) & \text{if } u < 0 \end{cases} \qquad u = y - \hat{q}_\tau(x)$$

Minimizing expected pinball loss gives $\hat{q}_\tau(x) = Q_\tau(Y \mid X = x)$.

Building an 80% PI:

1. Fit QR at $\tau = 0.10 \to \hat{q}_{0.10}(x)$
2. Fit QR at $\tau = 0.90 \to \hat{q}_{0.90}(x)$
3. PI for new observation: $[\hat{q}_{0.10}(x),\ \hat{q}_{0.90}(x)]$

Coverage assessment:

Empirical coverage = fraction of test observations where $y_i \in [\hat{q}_{0.10}(x_i),\ \hat{q}_{0.90}(x_i)]$

- Should be $\approx 80\%$ if well-calibrated
- Coverage $< 80\% \to$ intervals too **narrow**
- Coverage $> 80\% \to$ intervals too **wide**

Can QR produce a CI?

4.4 Quiz Questions

Q1. To construct an **80% prediction interval** using quantile regression, which pair of conditional quantiles is the natural choice?

- (a) 0.40 and 0.60
- (b) 0.20 and 0.80
- (c) 0.05 and 0.95
- (d) **0.10 and 0.90**

Q2. What theoretical target is optimized by minimizing **mean absolute error**?

- (a) **Conditional median**
- (b) Conditional mean
- (c) Conditional variance
- (d) Conditional mode

Q3. Empirical coverage on test data is **62%** for an intended 80% prediction interval. Most likely explanation?

- (a) The MAE must be small
- (b) **The intervals are too narrow**
- (c) The intervals are too wide
- (d) The median predictions are unbiased

4.5 Discussion Questions

1. HW7 P2: You fit a quantile regression model for housing prices. Your 80% PI has 62% empirical coverage. What does this mean and how would you fix it?

2. How does quantile regression for prediction intervals differ from a parametric approach (e.g., $\hat{y} \pm 1.28\hat{\sigma}$)?

3. When would you prefer the conditional median over the conditional mean as a prediction target?

5 Sampling and Weighting (HW 7, Quiz 7)

5.1 Weights Are Everywhere

Situation	Why weighting?	What weights represent	Effect on AUC
Balanced training on imbalanced data	Wrong base rate baked in	Inverse of sampling fraction	**None** (rankings unchanged)
Undersampling majority class	Same as above	Equivalent to weights	**None**
Weighted likelihood / IPW	Target a specific population	Inverse probability of selection	**None** (for base-rate fix)
IPCW (survival)	Censoring bias	Inverse probability of censoring	Changes estimates

Core insight: Base-rate correction changes calibration (the intercept), not ranking. AUC measures ranking, so AUC does not change after a pure base-rate fix.

5.2 Coverage Table

Legend: L = lecture notes ·✓= covered · — = not covered

Topic	Lecture	HW7	Notes
Why weighted fitting (class imbalance, sampling)	✓	P1	
Weighted log-loss: $-\sum_i w_i[y_i \log \hat{p}_i + (1-y_i)\log(1-\hat{p}_i)]$	✓	P1	HW AI images problem
What weighting does to the implied base rate	✓	P1a	
Base rate correction formula (log-odds adjustment)	✓	P1b,c	
AUC unaffected by base-rate correction	✓	P1d	Key insight
Log-loss and Brier score affected by base-rate correction	✓	P1d	Calibration metrics sensitive
Calibration plot to diagnose a base-rate mistake	✓	P1d	Shift in intercept visible
IPW for sampling correction	✓	—	Weights = 1/sampling probability
Connection to HW6 calibration: known vs. unknown correction	✓	—	Here we know the cause
Group-weighted modeling	✓	—	Targets specific subpopulation

5.3 Key Ideas

Base rate correction formula. Suppose the model was fit with training prevalence $\tilde{\pi}$ but the true prevalence is π. Correct the log-odds:

$$\text{logit}(\hat{p}_{\text{corrected}}) = \text{logit}(\hat{p}_{\text{model}}) + \underbrace{\log \frac{\pi}{1-\pi} - \log \frac{\tilde{\pi}}{1-\tilde{\pi}}}_{\text{log-odds correction}}$$

Only the **intercept** changes. Slopes (and therefore rankings) are unchanged.

HW7 AI Images scenario:

- True prevalence: 1/10 (10% AI images)
- Training prevalence with grad student's weights: $\approx$ 1/2 (equal weight to each class)
- Consequence: $\hat{p}(x)$ from model is too high for AI class
- AUC: unchanged after correction
- Log-loss / Brier: improve substantially after correction

Why unweighted is sometimes correct:

Connection to HW6 (calibration):

5.4 Quiz Questions

Q1. After a **correct** base rate adjustment, what should usually happen?

- (a) Log-loss and Brier score improve
- (b) AUC improves substantially
- (c) All metrics stay exactly the same
- (d) AUC worsens substantially

Q2. Why can a model have **strong AUC but poor log-loss** after class weighting or undersampling?

- (a) Because AUC depends on calibrated probabilities
- (b) **Because the model ranks observations well but outputs probabilities on the wrong scale**
- (c) Because weighting causes overfitting
- (d) Because log-loss ignores predicted probabilities

Q3. A logistic regression is fit after undersampling to 50/50. What is the main consequence of using predicted probabilities directly in deployment?

- (a) The model can no longer distinguish positive from negative cases
- (b) The estimated slopes will be close to zero
- (c) The ranking of test data will be poor
- (d) **The probabilities will be miscalibrated** (too high for positives)

5.5 Discussion Questions

1. HW7 P1: The grad student used weights $w_i = 1$ for AI, $w_i = 1/9$ for real. Walk through what this does to the effective base rate the model learns.

2. If we rank all test images from highest to lowest predicted probability of being AI, does the base-rate correction change the ranking? Why or why not?

3. How would you use a calibration plot to diagnose whether a base-rate correction is needed?

6 Forecasting (HW 8, Quiz 8)

6.1 Decomposition and Baselines

Component	Description	STL term
Trend	Long-run direction (up, down, flat)	T_t
Seasonality	Repeating calendar pattern (annual, weekly)	S_t
Remainder	What's left after trend + season removed	R_t

Additive: $Y_t = T_t + S_t + R_t$

Multiplicative: $Y_t = T_t \times S_t \times R_t$

Baseline Model	Forecast for h steps ahead	When appropriate
Naive	Last observed value y_T	No trend, no season
Seasonal naive	$y_{T-s+(h \mod s)}$ (same period last season)	All season, no trend
Mean	$\bar{y}$ (grand average)	Stationary
Linear trend	Fit line, extrapolate	Trend, no season

6.2 Coverage Table

Legend: L = lecture notes ·✓= covered · — = not covered

Topic	Lecture	HW8	Notes
Time series structure: sequential, non-iid	✓	context	
STL decomposition (additive/multiplicative)	✓	P1	Applied in HW
Baseline models (naive, seasonal naive, mean, linear trend)	✓	P2	3 baselines required
Why random CV is wrong for time series	✓	—	Core conceptual point
Rolling origin CV: growing vs. sliding windows	✓	P3	Used to tune in HW
Multi-horizon evaluation: performance as function of h	✓	P5	Evaluated at $h = 12$
Forecast error metrics: MAE, RMSE, sMAPE, MASE	✓	P5	
Prediction intervals for forecasts	✓	P4	Wider at longer h
ETS models (simple, double, triple / Holt-Winters)	✓	P3	One advanced model required
Feature engineering: lag features, calendar features	✓	—	For regression approach

Topic	Lecture	HW8	Notes
`daily_avg` vs. raw count: exposure correction	✓	P1	Months have different lengths

6.3 Key Ideas

Why rolling origin CV?

Why PI gets wider with horizon h?

Why `daily_avg` instead of raw monthly count?

6.4 Quiz Questions

Q1. What is the **primary reason** standard random k-fold CV is inappropriate for forecasting?

- (a) It cannot handle seasonal data
- (b) **Random splits can place future observations in the training set**
- (c) Only repeated k-fold is appropriate for time series
- (d) It is too slow for long time series

Q2. You forecast 12 months ahead and notice PI are much **wider at** $h = 12$ than at $h = 1$. Is this a problem?

Q3. A retail chain has weekly sales with a strong December spike every year for 10 years. Which STL component captures this?

Q4. True or False: If the **remainder** component of an STL decomposition looks like noise, the decomposition has successfully captured the structure in the data.

Q5. Naive vs. seasonal naive for a series with strong seasonality and no trend: what is the key difference?

7 Recommender Systems (HW 9, Quiz 9)

7.1 Content-Based vs. Collaborative Filtering

	Content-Based	Collaborative Filtering
Uses	Item features (genre, tags)	User rating history
Similarity	Between items (cosine of TF-IDF vectors)	Between users or between items
Cold-start: new user	Handles via feature preferences	**Fails** — no rating history
Cold-start: new item	**Fails** — no features for new items usually	Fails too
HW9	TF-IDF + cosine similarity	User-user or matrix factorization

7.2 Coverage Table

Legend: L = guest lecture slides ·✓= covered · — = not covered

Topic	Lecture	HW9	Notes
Utility matrix: users × items, mostly unobserved (sparse)	✓	context	610 users, 9700 movies, ~1.7% observed
TF-IDF: term frequency × inverse document frequency	✓	P1	High IDF = rare = informative
Cosine similarity for item-item comparison	✓	P1	$\text{sim}(u, v) = \frac{u \cdot v}{\|u\|\|v\|}$
Content-based recommendation pipeline	✓	P1	TF-IDF $\rightarrow$ cosine $\rightarrow$ top-k
User-user collaborative filtering	✓	P2	Find similar users, aggregate ratings
Item-item collaborative filtering	✓	—	More stable than user-user
Matrix factorization (at a high level)	✓	P2	Latent factor model
Cold-start problem	✓	—	New user / new item
Evaluation: Recall@k	✓	P2	$\|\text{recommended} \cap \text{held-out}\| / \|\text{held-out}\|$
Evaluation: MAE for rating prediction	✓	P2	
Not covered: explainable RS, veracity, novelty, human-in-the-loop	—	—	Explicitly out of scope

7.3 Key Ideas

TF-IDF:

$$\text{TF-IDF}(t,d) = \underbrace{\text{tf}(t,d)}_{\text{term freq in doc}} \times \underbrace{\log \frac{N}{\text{df}(t)}}_{\text{IDF: rarity across docs}}$$

Common terms (e.g., "Comedy") $\rightarrow$ low IDF $\rightarrow$ low weight. Rare terms (e.g., "pixar") $\rightarrow$ high IDF $\rightarrow$ high weight.

Recall@k:

$$\text{Recall@}k = \frac{|\text{top-}k \text{ recommendations} \cap \text{held-out items}|}{|\text{held-out items}|}$$

A user with 5 held-out movies where 2 appear in the top-10: Recall@10 = $2/5 = 0.40$.

Why is Recall@k often low even for a good recommender?

7.4 Quiz Questions

Q1. A ratings matrix: 610 users, 9,700 movies, 100,000 observed ratings. What fraction is observed, and what is this called?

Q2. A new user signs up with no rating history. Which method handles this more gracefully: content-based or collaborative filtering? Why?

Q3. In content-based filtering, what was cosine similarity used to measure in HW9?

Q4. Why does "Comedy" get a **lower** TF-IDF weight than "pixar"?

Q5. A user has 5 held-out movies; their top-10 list contains 2. Recall@10 = ?

8 Survival Analysis (HW 10, Quiz 10)

8.1 Decision Map

Target	Main question	Main tool	Main output	Notes
Population-level survival	What fraction of the cohort survives over time?	Kaplan-Meier	Group-level $\hat{S}(t)$ curve	Descriptive, not patient-specific
Fixed-horizon prediction	What is $P(\text{survive to } \tau \mid X)$?	IPCW-weighted binary model	Patient-specific risk at one horizon	Handles censoring via upweighting
Full survival curve	What is $\hat{S}(t \mid X)$ over many time points?	Survival forest or discrete-time hazard	Patient-specific survival curve + hazard shape	Multi-time prediction

8.2 Coverage Table

Legend: L1 = `survival.pdf` · L2 = `survival2.pdf` ·✓= covered · — = not covered

Topic	L1	L2	HW10
Censoring: $\tilde{T} = \min(T, C)$, $\delta = \mathbf{1}(T \leq C)$	✓	—	context
Independent censoring: $C \perp T \mid X$	✓	—	—
Two wrong approaches (drop / treat as failure) + bias direction	✓	—	—
KM: reading $\hat{S}(t)$, tick marks, jumpy tail	✓	✓	P1a,b
Estimating $\hat{G}(t)$: flip event indicator, run KM	✓	✓	P1c
IPCW: three groups at horizon τ	✓	✓	P2a
IPCW weight formula: $w_i = 1/\hat{G}(\min(\tilde{T}_i, \tau))$	✓	✓	P2a
IPCW-weighted binary classifier	✓	—	P2b
Brier score (0 = perfect, 0.25 = null)	✓	—	P2c
C-index (0.5 = random, 1.0 = perfect)	✓	—	P2c
Survival forest: log-rank split + KM leaf + `min.node.size`	—	✓	P3a
Discrete-time hazard: person-period reshape + binary classifier	—	✓	P3b
$\hat{S}(t) = \prod_{j=1}^{t}(1 - \hat{h}_j)$	—	✓	P3b iv

8.3 Key Formulas

IPCW weight:

$$w_i = \begin{cases} 0 & \tilde{T}_i < \tau \text{ and } \delta_i = 0 \quad \text{(censored before horizon)} \\ 1/\hat{G}(\tilde{T}_i) & \tilde{T}_i < \tau \text{ and } \delta_i = 1 \quad \text{(known failure)} \\ 1/\hat{G}(\tau) & \tilde{T}_i \geq \tau \quad \text{(observed past horizon)} \end{cases}$$

Survival from discrete-time hazards:

$$\hat{S}(t) = \prod_{j=1}^{t} (1 - \hat{h}_j)$$

8.4 Key Distinctions

KM vs. IPCW classifier vs. survival forest:

C-index vs. Brier score:

Survival forest vs. discrete-time hazard:

8.5 Discussion Questions

1. A patient is censored at 8 months in a 1-year study. What is their label and IPCW weight? Why?

2. What does the C-index measure? What does the Brier score measure? Can a model have a high C-index and a high (bad) Brier score?

3. Give one situation where you would prefer a survival forest, and one where you would prefer a discrete-time hazard model.