Contents

College Football Model — Under the Hood (v9.5)

A Clear, Honest Account of What It Is, Where It's Been Wrong, and Where Its Edge Lives

Clarity and honesty over flash

Citation discipline, stated once and binding everywhere below: the only performance numbers this project verifiably offers are **73.7% accuracy, Brier 0.1712, AUC 0.816** — the production v9.5 artifact's clean one-shot on the 2025 season (763 games, the pre-registered "Learned logit" variant, design frozen before the season was ever scored). Every number this project published before June 10, 2026 — 85.1%, 83.9%, 83.6%, Brier 0.1200 — was inflated by roughly ten percentage points by a data leak we introduced and later caught. Part VII explains it in full, because understanding how the leak happened matters as much as understanding the architecture.

Part I — Orientation

This part is the whole document in summary: the legal footing, what the product actually is, and a map of where this document leads. Read Part I alone and you will know what the model is, how to bet with it, and how much to trust it. Everything after Part I is the proof.

I.I Standing Notice

This is an unofficial fan and analysis project. It is not affiliated with, endorsed by, or sponsored by the NCAA, any college or university, any athletic conference, the NFL, ESPN, or any data provider named here.

Everything in this document is for entertainment and educational purposes only. It is not financial, investment, or betting advice, and nothing in it recommends any wager. Every prediction, probability, ROI figure, and backtest is historical, modeled, or illustrative — and past or simulated performance guarantees nothing about the coming Saturday. To be explicit: all performance figures this project published before June 10, 2026 are retracted across every surface (see VII.I); only the v9.5 numbers above may be cited.

Sports wagering carries a real risk of financial loss. If you choose to bet, bet only what you can afford to lose, be of legal age (21 or older in most U.S. jurisdictions), and never bet where it is not legal. If gambling is or becomes a problem, call 1-800-GAMBLER for free, confidential help.

Data is used in compliance with each provider's terms of service — the College Football Data API, ESPN's public scoreboard, and TheOddsAPI. SP+ ratings are © Bill Connelly and are attributed to him wherever they appear.

I.II Our Product

Strip away the mechanisms and the product reduces to the following: **for every game: a projected margin and a win probability...for each week: a short list of bets locked publicly before kickoff and graded honestly afterward.**

That single output is read two ways, because a margin and a probability are two products of one number (the conversion is exact, and III.II does the arithmetic). The probability is the moneyline read: "who wins", and how sure are we. The margin is the spread read: "by how much", against the number the market is offering. Same forecast, two languages.

This model's edge is *not* raw accuracy. In transparency, leak-free testing the model is about as accurate as the Las Vegas closing line. It does *not* beat the market on average, and any model claming to "beat Vegas" every week should trigger skepticism.

The entire edge is **selectivity**: of the ~60 games on a card per week, it shrugs at almost all of them and acts only on the few where its number disagrees sharply with the line. Our one-line betting rule, *bet only when our number and the market's number separate by more than a set threshold*, is stated here as the "what", the "how", the thresholds, the staking, and the hard truth that a real edge dies on execution rather than model error: Part IV.

Lastly, it's vital to note that *trust is not uniform across every product-type we publish.* A graded, locked pick carries more weight than a single-game prediction, which carries more than a power ranking, which carries more than a full-season simulation. Confidence degrades the further an output sits from a bet that gets graded. Part V ranks every product surface and shares the reasoning.

I.III What's to Follow

Upon completion of the document, the reader will understand four things: (1) what the model predicts and how to bet it; (2) how we leaked future information into our own results, why it survived for months, and how we caught and fixed it; (3) the defense behind every headline number: accuracy, Brier, AUC, ROI; and the model's inevitable blind spots.

The document is designed in crawl, walk, run format. **Part I** is the crawl: the entire story in a few pages. **Parts II–V** are the walk — the vocabulary (II), what the model is *for* and why a real edge can exist at all (III), how to use it on gameday (IV), and exactly how much to trust each thing it produces (V). **Parts VI–XI** are the run, for readers seeking the proof: the data and target (VI), the leak (VII), the architecture and how it was validated (VIII), how signals were chosen and why chasing more carries diminishing returns (IX), how to read every number (X), and the limits as well as the road to the next version (XI). The **Appendix** directly answers the sharpest questions, offers a script for explaining the model to various fields, and indexes every term.

A note on navigation. Sections carry both a visible Roman label (e.g. "V.II") and a linked anchor, so you can either click a cross-reference or search the text for the label. When a term is defined elsewhere, it is hyperlinked to reduce redundancy. The footnote markers `*`, `**`, and `***` correspond to asides that are useful but would interrupt the flow if left inline.

Part II — Terms and Methods

This Part is the document's vocabulary. While not particularly riveting, its the groundwork behind everything that follows. The architecture, the leak, the betting edge, the metrics — they're all derived from of a couple dozen ideas. Rather than re-explain them each time they recur, we pin every one down here, once, with a worked example with real numbers. Read it carefully and the rest of the book reads on its own terms. None of this requires math beyond multiplication and division.

The Part splits in two. The first half is the betting and probability grammar: what a price is, what a probability means, what makes a bet worth making. The second half is the modeling toolkit: the specific statistical machinery this project used (and, in several cases, deliberately rejected), each tagged with what it's for and where in the book it actually appears.

II.I Betting and Probability

Probability and Variance

When the model says "Georgia 70% to win," it is not saying Georgia *will* win, and it is not saying Georgia will win 70% of *this one game*. Intuitively, a game happens once and someone wins. It's making a claim about the long run: out of all the games where the model says "70%," the favored team should actually win about 7 out of every 10. That is the only honest meaning a single percentage can carry — a frequency promised over many repetitions, attached to a singular event.

This leads to the single most important mental adjustment for reading this whole document: a 70% pick that loses is not the model being wrong. If you collect every 70% pick and they win exactly 70% of the time, the model is *perfect*, to include 3 out of every 10 of those picks losing. A model whose 70% picks won 100% of the time would be a *miscalibrated* model. So a confident pick busting on Saturday is no more evidence against the model than a coin landing tails is evidence the coin isn't fair. The only way to judge a probability is to gather a pile of same-confidence picks and verify it against the hit rate. One singular game can never confirm or refute a percentage that is not "0" or "100".

The flip side of that coin is *variance*, the emotionally hard truth in betting that even a flawless bettor has losing weeks due to chance. Take a coin you know is rigged to land heads 65% of the time and bet heads every flip (a real, provable edge). Flip a coin 10 times averaging 6.5 heads. Any given ten flips might give you 4 heads (a losing session) or 9 (a heater). The 65% is a promise about thousands of flips; it says little about the next ten. Those ten will average about 6–7 heads, but in any single batch you'd land somewhere between 4 and 9 about 95% of the time (occasionally

even outside that). The fewer the flips, the wider that "wobble"; only over the long run does it settle near 65% (the law of large numbers in action). A football prediction model adheres to the same law. Over a single Saturday of eight picks, an average of "5.2 wins" hides a wide spread, and going 3 for 8 on a weekend doesn't verify whether the edge is real. This is why you judge a model over a season, not a Saturday. A handful of bets is influenced by luck; the edge is only revealed once you have enough bets for the variance to average out. A bad week is not a broken model; it's the expected outcome of a 65% process. A good week is not a vindicated one either. (This is also why staking is done with fractional Kelly, defined in II.II and applied in IV.IV: bet small enough that an ordinary cold streak can't deplete the bankroll before the long run arrives.)

Odds, the moneyline, and the vig

Bookmakers don't quote probabilities; they quote prices. American odds are the format you'll see and are built around a $100 reference. A negative number (a favorite) like −250 means you risk $250 to win $100 (more negative means heavier favorite: −800 is a near-lock, −120 a slight favorite). A positive number (an underdog) like +180 means you risk $100 to win $180 (larger number means more of a longshot).

Every price has a probability embedded (the win rate at which the bet would exactly break even). Converting is just arithmetic. For a favorite, take the absolute value and compute `odds / (odds + 100)`:

```
−250  →  250 / (250 + 100)  =  250 / 350  =  0.714  =  71.4%
```

So −250 is the market saying "this team wins about 71.4% of the time." (Bottom line: win more than 71.4% (break-even probability being 0.714) and you profit; less and you don't. For an underdog, compute `100 / (odds + 100)`:

```
+180  →  100 / (180 + 100)  =  100 / 280  =  0.357  =  35.7%
```

Now the catch that makes betting difficult. Convert *both sides* of a typical game where Vegas prices each side of the spread at −110 (risk $110 to win $100):

```
−110  →  110 / 210  =  0.524  =  52.4%   (each side)
52.4% + 52.4% = 104.8%
```

The two implied probabilities sum to more than 100% (here 104.8%). But a game has exactly one winner; the true probabilities must sum to 100%. That extra 4.8% is "the vig" (also called juice or the hold); the bookmaker's built-in commission that is baked into the odds. Before the vig (both sides sum to 100%), you could bet both

sides and net zero. After the vig, betting both sides will lose you $10 (at -110, winner only recieves $210 of the combined $220 they'd have put down on both sides).

De-vigging is the act of stripping away that commission to recover what the market actually thinks. Crudely, you divide each side's implied probability by the total:

```
fair prob of one side  ≈  52.4% / 104.8%  =  50.0%
```

A −110/−110 game de-vigs to a true 50/50 coin flip, that's a pick'em. The vig-inflated number is not the market's real opinion. Whenever this document compares "our probability" to "the market's probability" or asks whether a bet is +EV (positive expected value), it uses is the *de-vigged* market number. Misquote the raw −110 as "52.4% to win" and you've credited the book's house edge to the team. (The structural consequence - tying the de-vigged market is a slow guaranteed loss, so the vig must be *overcome*, not matched — is the argument of III.II.)

The spread and covering (ATS)

The moneyline is a bet on who wins, straight up. The point spread is a different bet that handicaps the favorite with a points head start to turn a lopsided game into a coin flip. Write it "Alabama −7": the minus belongs to the favorite, and it's points subtracted from Alabama's final score before the bet settles. Bet Alabama −7 and you win only if Alabama wins by 8+ points (winning by exactly 7 is a "push," bet refunded). Alabama winning 30–20 covers (margin 10 > 7); winning 24–20 does not (margin 4 < 7), even though Alabama won the game (the "moneyline"). Bet the underdog at +7 and you win if they lose by up to 6 points or win outright.

Betting the spread is called betting ATS, "against the spread." It is a genuinely different question from "who wins". A heavy favorite can win the game in dominant fashion and still lose you the bet. Conversely, a doomed underdog can lose the game spectacularly and win you the bet by covering. This is why the engine has two products derived from the same predicted margin: one for "who wins" (moneyline) and one for "will they cover" (ATS); see III.I.

Spread bets are almost always priced at −110 each side, which means each side's break-even is 52.4%. That number is essential for the rest of the document. At standard −110 vig you must win 52.4% of your spread bets just to *break even*. Win exactly half and you slowly lose because the 4.8% vig grinds you down. Anything above 52.4% is real profit. 59.5% ATS isn't "9.5 points above a coin flip" — the coin flip you actually have to beat is 52.4%, not 50%. Clearing the real, vig-inclusive bar by ~7 points is the entire edge, and in ATS terms that gap is still sizeable. (Our ATS hit rates by edge size are reported in X.IV.)

Expected value

Expected value (EV) is the singular concept that separates betting from gambling. It is the average dollar outcome of a bet if you could make it repeatedly (your *probability-weighted profit*). The formula is just ['what you win' times 'how often you win'], minus ['what you lose' times 'how often you lose']:

```
EV = (profit if you win x prob of winning) − (amount risked x prob of losing)
```

Worked example: you put $100 on a +150 underdog and you believe the true win probability is 45%. At +150 odds, a win returns $150 profit, and your loss is the $100 you risked. The win probability 45%, so the loss probability is 55%.

```
EV = (0.45 × $150) − (0.55 × $100)
   = $67.50 − $55.00
   = +$12.50
```

Positive. On average, every time you place this $100 bet you expect to come out $12.50 ahead, even though you'll lose it 55% of the time. The wins pay off more than enough to cover the more-frequent losses. Now notice *where* the edge came from. Convert that +150 price to its implied probability: `100 / 250 = 40%`. You think the team has a 45% shot while the market prices the team at having a 40% shot. That gap of +5% is exactly what makes the bet +EV. Stated as a law: a bet is +EV if and only if your true win probability is higher than the probability implied by the price. Beat the price's number and you profit on average; fall short of it and you bleed on average (independent of win frequency; dependent only on the *gap between price and probability*).

This is why a *calibrated* model is a betting tool, not just a prediction tool. You de-vig the market price into its probability. If your probability is meaningfully higher, the bet is +EV and you profit on average. If yours is lower or equal, you pass on the bet even if you think the team will win, because "will probably win" and "is mispriced" carry different meaning. A −800 favorite breaks even only if it wins 88.9% of the time (8/9 times) since that is exactly what the price charges. A −800 favorite that truly wins 92% is a *winning* bet (on average), while one that wins only 85% is a *losing* bet (despite winning over five out of six instances; 85% < 88.9%). EV doesn't care how the game "feels"; only whether your number cleared the price's number. (The full EV-gated betting policy is laid out in IV.IV; its near-miss results in X.III.)

The normal curve and standard deviation

When you predict a game will land at "Alabama by 7," reality won't deliver exactly 7 — it'll come in at 3, or 10, or Alabama loses by 4. Collect the misses (predicted margin minus actual margin) across thousands of games and they pile up in a particular

shape: the bell curve (or "normal distribution"). The distribution shares an intuitive insight: most outcomes land near the prediction. The farther out you go, the rarer outcomes become - symmetrically - tapering on both sides. Picture a pile of sand poured onto your prediction — tall in the middle, thinning smoothly toward both edges.

The number describing *how wide* that pile is is the standard deviation (SD), read as "the typical distance an outcome lands from the center." A small SD means a tight, tall pile where outcomes cluster hard and creating a predictable quantity. A large SD means a flat, wide pile where outcomes scatter creating a noisy quantity. The curve has a famous rule of thumb: about 68% of outcomes land within one SD of the center, about 95% within two.

The concrete football fact the rest of the book leans on: college-football final margins scatter widely and roughly normally around even a good prediction. The model uses a working spread of about 13.5 points to translate between predicted margins and win probabilities, and real outcomes scatter at least that much — in truth a bit wider, about 15.6 points around the sharpest predictions (the market's) and about 19.8 around the model's own. Why there are several such numbers, and why 13.5 is a fixed display dial rather than a measured error, is untangled in VIII.II. For the illustration that follows, read 13.5 as the model's working curve. So if the model predicts "Alabama by 7," the center sits at +7. One SD (±13.5) puts about 68% of outcomes between Alabama −6.5 and Alabama +20.5; two SDs (±27) puts about 95% between Alabama losing by 20 to winning by 34. It's worth noting how wide that is: a prediction of "by 7" routinely produces anything from a two-touchdown loss to a four-touchdown win. Two touchdowns of slop each game is normal football and does no discredit the model. That's also why a confident pick still loses about 30% of the time. Even when the center of the curve is dead-on accurate, part of the bell (in this case 30%) still sits on the wrong side of zero. The prediction can be sound and the night can still break the other way. That's irreducible variance or simply put, noise - *not* model error.

That working spread of 13.5 is what turns a predicted margin into a win probability, which is simply the fraction of the bell curve sitting above zero. A bigger predicted margin slides more of the curve past zero, so a stronger favorite carries a higher probability. Compare two games: one the model pegs at +3 has about 59% of its curve above zero, while one it pegs at +7 sits around 70%. The margin always comes first, from the matchup, and the probability is just that margin translated. The tool that does this conversion — turning a predicted margin into a win probability — is called `pnorm` (which measures how much of the bell curve sits on the winning side of zero). The conversion (3 pts ≈ 59%, 7 ≈ 70%, 14 ≈ 85%) is performed in VIII.II.

Calibration

Everything above converges on the one property that decides whether a model's probabilities are usable or merely decorative: calibration. A model is calibrated if its stated probabilities match reality in the long run — gather every game where it said 70%, and if the favored team actually won about 70% of those (and the 80%-claims won ~80%, the *55%*-claims *~55%*, all the way up and down), the model is calibrated. Its numbers mean what they say.

A model can be a great picker and still be a liar about probabilities, and the failure has two flavors. Overconfident: it says 90% on games that win 75% of the time — it sounds authoritative and bets like a maniac, computing fat +EV on bets that are actually break-even or worse and oversizing them. Underconfident: it says 60% on games that win 80% — it leaves money on the table, passing real +EV bets because its understated probability never clears the price.

Why does calibration sit at the very top of what we care about? Because EV is computed *from* the probability. If the probability is a lie, every downstream number is a lie too: the EV is fictional, the bet sizing wrong, and a beautiful-looking backtest can be quietly built on probabilities that don't hold. You cannot bet your way out of a miscalibrated model; you can only lose more efficiently. Accuracy doesn't catch this at all — a model can be 73% accurate while wildly overstating its confidence — which is why the project's first metric is the Brier score (defined in II.II, read deeply in X.I), built to reward calibration and punish bravado. The whole through-line: a probability is a long-run frequency; a bet is +EV only when your probability beats the price's; therefore your probabilities have to be true, or the entire edge is imaginary. (The honest limits of our own calibration check are discussed in X.V.)

II.II Modeling Toolkit

Models, features, training rows

Strip away the word "model" and what you have is a rule that turns facts known before kickoff into a guess about the outcome. The facts come in a table where every row is one past game and the columns split two ways. A *feature* is an input column — something you knew before the game: the two teams' ratings, their records so far, who Vegas favored, home or away. The *target* is the one thing you're predicting, recorded after the game: in our case, the margin of victory. A single *training row* for "Alabama at Texas, week 5 of 2023" might read: *Alabama rating 24.1, Texas rating 19.8, Alabama favored by 6, game in Austin → Texas won by 5* — features left of the arrow, target on the right.

Training is the process of looking at thousands of completed rows and finding the weights (multipliers) that best connect features to targets — it nudges the weights to

miss less across all games at once and stops when it can't reduce the misses further; the frozen set of weights it lands on *is* the model. *Predicting* is then trivial: plug a new, unplayed game's features into the frozen rule and read off the guess. The model never sees the new game's result, which is the entire point — a model tested only on games it trained on is a student grading their own exam with the key open. This project's whole data foundation (7,179 graded training games, 2016–2025) is described in VI.I; the target choice in VI.II.

OLS / least squares (the "GLM" head)

One of the two prediction engines is a linear model: the simplest honest forecaster there is, which assigns each input a fixed weight and adds them up ("take the rating gap times *this* number, the recruiting gap times *that* number, add the home-field bump, total it"). You can print the rule on an index card. It is fit by *ordinary least squares* (OLS), the workhorse method for choosing those weights. "Least squares" describes the goal: pick the weights that make the squared misses as small as possible, where a miss is the gap between predicted and real margin. Squaring does two jobs — it makes every error count as badness regardless of sign (a +5 and a −5 miss don't cancel), and it punishes big misses far more than small ones (a 10-point miss counts 100, two 5-point misses count 25 each). "Ordinary" just means the plain version with no extra penalty bolted on.

An honest terminology note, because it recurs: this head is labeled "GLM" throughout the code and diagrams, but in the current version it is literally R's `lm()` — ordinary least squares — *not* a logistic generalized linear model. The name is a fossil from an earlier era when this head really did predict win/lose probabilities directly; we kept the label so old comments still match. When the book says "GLM head," read "the ordinary-least-squares straight-line rule." *Used*, as one of the two heads; applied in VIII.I.

Gradient-boosted trees (XGBoost)

The other engine is a committee of hundreds of tiny if-then rules. A *decision tree* is a flowchart of yes/no questions ("is the rating gap above 10? if yes go left…") with a small predicted margin at the bottom of every path. A single shallow tree is crude, but XGBoost (the software; the name is short for "eXtreme Gradient Boosting") builds hundreds in sequence — that's *boosting*. Tree #1 makes a rough guess; we look at its leftover errors (the *residuals*, the part of the real margin not yet explained); tree #2 is trained to predict *those errors*; we add a small slice of its correction; repeat hundreds of times, each tree a specialist in the mistakes its predecessors couldn't shake.

Why a second, harder-to-read machine? Because the straight-line rule assumes every factor's effect is fixed and additive — one point of rating gap worth the same in week 1 as week 12. Football has *interactions*: places where the effect of one thing

depends on another (rating gaps mean less early in the season; a real betting line tends to swamp other signals). A flat rule structurally cannot represent "this factor matters more in that situation"; a tree does it for free, since every branch is "the answer down here depends on the answers above." Trees are also natively robust to overlapping features — they grab whichever feature most reduces error at a split and ignore its redundant twin — which a linear model is not (see the multicollinearity discussion in IX.I). The cost: trees are hungrier for data and you can't read them off a card. Pairing the two — stable interpretable backbone plus situational nuance, wrong in different ways so blending beats either — is the whole "two heads" idea. *Used*, as the second head; applied in VIII.I.

ANOVA and the F-test

To decide which features earn a seat, each candidate goes on trial. ANOVA — analysis of variance — asks one question in plain English: "when I add this feature, does the leftover unexplained variation drop by more than random chance would have dropped it anyway?" That last clause is the whole game, because adding *any* column — even one of random numbers — shrinks leftover variance a little by accident. The tool ANOVA uses is the *F-test*, a ratio of "variance this feature explains" over "variance still left unexplained, per slot used up." If the feature explains a lot relative to what's left, F is large and chance is an implausible explanation, so the feature is "significant." The p-value is just F translated to a 0-to-1 scale: "if this feature were truly worthless, how often would chance alone fake a reduction this large?" A tiny p (0.001) means real; a fat p (0.97) means chance fakes this all the time. *Used*, as the feature-selection screen; applied in IX.I.

A one-line warning that cost this project an afternoon of confusion: the Type-I (sequential) ANOVA trap. Type-I ANOVA credits explained variance *in the order the variables enter the model*, so a strong-but-overlapping feature entered last can score p ≈ 0.97 ("useless") simply because earlier, correlated features already claimed the variance it would have explained — the test answers "does this add anything *after everything else*?", a different question from "is this related to the outcome?" The full war story (the spread flagged as noise) is told in IX.I.

Bonferroni correction

Running many significance tests at once invites false positives. The usual bar of $p < 0.05$ lets through a 5% false-positive rate by construction — even a worthless feature has a 1-in-20 chance of looking significant by luck. Across 45 candidate features that's `45 × 0.05 ≈ 2.25`, about two fake "significant" features expected by chance alone. The *Bonferroni correction* fixes this by tightening the bar in proportion to the number of tests: divide the target false-positive rate by the number of tests, here

0.05 / 45 ≈ 0.0011, so a feature must clear that much sharper threshold to count. *Used*, as the keep bar in feature selection; applied in IX.I.

Penalized regression (ridge / lasso / elastic net)

These three are all *penalized regressions*, and the one-sentence version is: penalized regression is ordinary least squares plus a tax on big coefficients. OLS fits coefficients with no constraint; penalized regression adds a fine, so the model only "spends" a big coefficient on a feature that really earns it. *Ridge* (L2) taxes the square of each coefficient, shrinking them all toward zero but never quite reaching it (everybody takes a pay cut; nobody gets fired). *Lasso* (L1) taxes absolute size, snapping small coefficients to exactly zero — automatic feature selection. *Elastic net* blends both ($\alpha = 0.5$ is a 50/50 mix). These cure two specific diseases: too many features for your sample size, and multicollinearity destabilizing coefficients.

Tested and rejected. Elastic net was run properly ($\alpha = 0.5$, penalty strength chosen by in-fold cross-validation, scored on the same folds as everything else) and came in 0.0009 Brier *worse* than the plain ANOVA-selected OLS — the penalty shrank the model to just 5–11 surviving features and threw away real signal the Bonferroni screen had correctly kept. The reason the textbook tool loses here is that it solves a problem this dataset doesn't have: penalties earn their keep when the feature count p rivals the row count n (the genomics case, 20,000 genes on 200 patients). Here $p \ll n$ — roughly 50 candidate features against 7,179 games, about 1 feature per 140 rows — so OLS coefficients are already stable and the tax just confiscates good signal. The full post-mortem is in the graveyard, IX.II.

Walk-forward cross-validation, OOF, pre-registration, and the spent one-shot

A model that looks brilliant on the data it was trained on has told you nothing — it could be memorizing. The only meaningful test is performance on games it has *never seen*, and getting that test right is the discipline that earns this project's numbers their trust. Four ideas do the work, all *used* and applied in VIII.IV.

Walk-forward cross-validation: always train on the past, test on the future, never the reverse. Football happens in time, so the test must respect time — slide a training window forward through the seasons and always test on a season strictly *after* the training data. A "fold" is one such train-then-test split (train 2016–2020, test 2021; train 2016–2021, test 2022; and so on), each one mimicking how the model will be used live. *Out-of-fold (OOF)* predictions are the leak-proof kind used to tune downstream pieces without cheating: a prediction for a game made by a version of the model that was *not* trained on that game. If a model graded itself on games it studied, the grade would be a memorization score, not a forecasting score.

Pre-registration is the single most important anti-self-deception device: commit to the decision rule *before* you see the answer. When you choose *after* peeking at the test result ("let me also try variant X and re-check, ooh that's better, keep it"), you are quietly fitting the test set — every peek-then-adjust bends the model toward that holdout's noise, and your "test score" becomes a rehearsal you've polished. Pre-registration forces the rule to be written down and locked before the sealed test is touched. *The spent one-shot* is the consequence: a sealed holdout works *once*. The first time you evaluate on it you learn something about it, and any further decision influenced by that knowledge silently converts it into a validation set you're now fitting. So the moment the 2025 season was scored, it was retired; every future modeling decision runs only on the earlier folds, and the next genuinely clean test is the live 2026 season, which cannot leak because it hasn't happened.

The Kelly criterion

Knowing which bets are good leaves the question of *how much* to put on each. Bet too little and you leave money on the table; bet too much and you can be wiped out even when every individual bet is in your favor. The *Kelly criterion* is the mathematically correct answer: a formula that takes your *edge* (how much better your true win probability is than the price implies) and the *odds* (how much the bet pays) and returns the fraction of your bankroll that maximizes long-run *compounding growth* — not expected profit on one bet, but growth over hundreds. For an even-money bet the Kelly fraction simplifies to your edge: if you win 55% versus a 50% break-even, `f = 0.55 − 0.45 = 0.10`, bet 10% of bankroll. The general form is `f = edge / odds`. Kelly bets proportional to how good the bet is, which a flat "same stake every time" rule lacks.

But full Kelly is only optimal if your probabilities are *exactly* right, and no model's are. Kelly is savagely unforgiving of overconfidence — think you're 55% when you're really 52% and full Kelly over-bets every game, compounding losses instead of wins, with drawdowns of 50%+ and a real risk of ruin. The fix is *fractional Kelly*: bet a fraction of the Kelly stake. This project uses *quarter-Kelly*, because growth falls off slowly as you back off (quarter-Kelly keeps roughly half of full Kelly's long-run growth) while risk falls off fast (drawdowns shrink to a small fraction). You sacrifice a little growth for a lot of protection — insurance against the one thing we know is true: our probabilities are approximately right, never exactly. *Used*, as the staking rule; applied in IV.IV. Kelly is downstream of calibration: feed it dishonest probabilities and it will faithfully, mechanically over-bet and lose money with perfect discipline, which is why calibration (and the Brier score) sits above it in the metric hierarchy (III.III).

Brier score and AUC

Two scoring metrics recur throughout, defined here and read deeply in X.I. The *Brier score* is the mean squared error of the probabilities themselves: take `(p − outcome)²` for each game, where outcome is 1 if home won and 0 if it lost, and average. It scores the *number* we published, not just the call, which is why it's the metric experiments are run against — say 90% and win, you pay `(0.90 − 1)² = 0.01`; say 60% and lose, you pay `(0.60 − 0)² = 0.36`, more than if you'd humbly said 40% and missed (`0.16`). Lower is better; it rewards probability honesty and punishes bravado. *AUC* (area under the ROC curve) measures *ranking* quality and never picks a single winner: across every pairing of a game the home team won with one it lost, AUC is the fraction of pairs where the model gave the won game the higher probability. A perfect ranker scores 1.0 even when it was technically "wrong" on a 60% favorite that lost, because it still ranked that game below its more-confident calls. Both *used*, as headline metrics; the full reading, baselines, and market benchmark are in X.I and X.II.

Part III — Objective and Edge

Before a single feature or formula, a blunt question: what is this model actually *for*? "Predicting football games" is not an answer. It's an activity, and one at which thousands of handicappers, several billion-dollar companies, and one extremely efficient betting market are already very good. If the goal were merely "be right about who wins," the honest advice would be: go read the closing line and stop. So the goal has to be sharper than that, and getting it exactly right is what this part is about. It also answers the question every fair skeptic asks once they hear the headline numbers — *if you're basically tied with the casino, why use this at all?* — and the answer turns out to be the most important strategic fact about the whole project.

III.I What the Model Is For

Here is the objective, stated in one sentence, with every word essential:

> Pick a small number of moneyline and against-the-spread bets each week, lock them publicly before kickoff, and be profitable over a season.

Read it slowly.

A small number. Not all sixty-odd games on a Saturday. Five or so. The entire strategy lives in the word *small*, for reasons that will take until III.IV to fully unpack — but

the seed is this: we are never obligated to have an opinion on a game, and almost all of our discipline is in the games we decline.

Moneyline and against-the-spread. Two bet types (both defined in II.I). We'll see in III.II that they're not two products at all but two readings of one underlying number.

Lock them publicly before kickoff. Written down, timestamped, frozen, ungameable after the fact. This is the part that separates a track record from a story. Anyone can tell you, in December, which Septembers they nailed. Far fewer will show you the ticket they wrote in September and let you grade it.

Profitable over a season. Not "accurate," not "impressive," not "beats Vegas on average." Profitable — which, as III.III will show, is a different and harder thing than being accurate. A model can be *less* accurate than the market and still make money; a model can be *more* accurate than the market and still lose money. Profit is its own quantity, and it is the one that counts.

The locked pick log is the brand. Everything else — the app, the rankings, the charts, the map, the recruiting pipelines, even this document — is scaffolding around that log. If you remember one thing from this whole explainer, remember that the product is not "a smart model." The product is a public, self-grading record of a few honest bets a week. The model is simply how we decide which bets to make. (The mechanics of *what* gets logged and how it's read live in IV.I; here we're fixing the objective, not the user manual.)

That framing matters because it changes what "good" means. A model judged as a forecaster wants to be right about all 800 games. A model judged as a bettor wants to be right about the five it chooses to bet — and is free to be wrong, or silent, about the other 795. Those are different jobs with different scoreboards, and conflating them is the single most common way people misread what a betting model is doing. We'll keep coming back to that distinction, because nearly every apparent paradox in this part dissolves once you hold it firmly.

III.II Two Sides of the Same Coin

The two bet types we just named raise an immediate question: are moneyline and against-the-spread (ATS) two independent products, or is one derived from the other? In this model the answer is concrete, not philosophical — one number underlies both, and the relationship is literal.

The model predicts one thing: the *margin of victory* — how many points one team is expected to win or lose by. Both bet types are derived readings of that single margin signal.

```
                predicted margin (one signal)
                /                           \
read #1: probability                read #2: vs the line
pnorm(margin / σ)                   margin − closing spread
     ↓                                    ↓
MONEYLINE pick                      ATS pick
(who wins, how sure)                (where we disagree with Vegas)
```

Think of the predicted margin as a single beam of light, and the two products as two lenses held up to it. The beam doesn't change; what you read off it does.

The moneyline lens asks: *how likely is a win?* It converts the margin into a win probability through the S-shaped conversion described in II.II — bigger expected margin, higher win probability, on a curve. If that probability clears a confidence threshold, it becomes a moneyline pick. This lens ignores the betting line entirely; it cares only whether the team wins, not by how much relative to any handicap.

The ATS lens asks: *do we disagree with the casino?* It compares our *market-blind* margin — the model run with the betting line deliberately hidden from it — against the spread Vegas actually posted. The gap between the two is the bet. (Why we run a market-blind version at all, and why that blindness is essential rather than a quirk, is the heart of III.IV and III.V.)

A worked example, start to finish

Suppose the model, looking only at football inputs, expects Alabama to win by 10. And suppose Vegas has posted the line as Alabama −3 — Alabama favored by 3. Watch how the *same* +10 produces two entirely different actions.

The moneyline read. Convert +10 to a win probability using the points-scale conversion (σ = 13.5, the conversion is laid out in II.II):

```
P(Alabama wins) = pnorm(10 / 13.5) = pnorm(0.74) ≈ 77%
```

So the moneyline read is: Alabama is about 77% to win the game outright. Whether that becomes a logged pick depends on the confidence tiers (see IV.I); 77% lands in the credible-but-not-elite middle tier. Notice this read ignored the line completely — it only asked whether Alabama wins, not by how much against a handicap.

The ATS read. Now bring in Vegas. We say Alabama by 10; Vegas says Alabama by 3:

```
edge = our margin − the line = 10 − 3 = 7 points of disagreement
```

We think Alabama is *seven points better than the casino does*. That seven-point gap is the bet, and the bet is Alabama to cover the −3 spread: if Alabama is really a ten-

point-better team, they should comfortably clear a three-point handicap. Note carefully that the ATS bet is *not* "Alabama wins" — the moneyline read already said that, and at −3 the market mostly agrees Alabama wins. The ATS bet is specifically *"the market has Alabama too cheap."* It is a bet on the disagreement, not on the team.

Same +10 margin. One number, two reads, two completely different statements: "Alabama is 77% to win" and "Alabama is 7 points underpriced." The moneyline read is a statement about the *game*; the ATS read is a statement about the *line*.

> When the two reads point opposite ways. They can, and that's a feature, not a bug. Suppose we still expect Alabama by 10, but Vegas says Alabama −17. Moneyline read: Alabama ~77% to win, a fine favorite. ATS read: edge = 10 − 17 = −7, so we think Alabama is seven points *worse* than the line, and the ATS bet is on the underdog to cover. We would back Alabama to win *and* back their opponent to cover, simultaneously, with no contradiction whatsoever. The moneyline is about the outcome; the spread is about the price. A great team can be a terrible bet, and a mediocre team a great one, purely on price.

This unification is the reason the model is built to predict margin in the first place. An earlier design that predicted win/loss directly — trying to output a 1 or a 0 — would have to treat the ATS product as a bolted-on afterthought, backing out an implied margin from a win probability through a lossy, awkward translation. By training on margin directly, the quantity the model is *literally* optimized for becomes the native currency of both products. Moneyline and ATS stop being two models that must be reconciled and become two readings of one honest number. (The architecture that makes this work — and the version history behind it — belongs to a later part; here the point is only that the single-signal design is what lets one prediction serve both bets cleanly.)

III.III Metric Hierarchy

When you build a model you must decide what "better" means — and what to do when two definitions of better disagree. We use three measures, in a strict priority order, and the order *is* the point. The three measures themselves (Brier score, accuracy, and risk-adjusted return via tiers and Kelly staking) are all defined in II.II; here we are arranging them into a hierarchy, not re-teaching them.

The order is: be honest, then be right, then be paid.

Be honest comes first — Brier score. The Brier score asks whether our probabilities are *truthful*: when we say "70%," does the thing happen about 70% of the time? This sits on top for three reasons. First, every dollar of betting expected value is computed *from* a probability — if the probabilities are lies, the EV math is built on lies, and you

will confidently bet games you should pass and pass games you should hammer. A model with dazzling accuracy but dishonest probabilities will mis-size its confidence and bleed money even while looking brilliant. Calibration is the foundation the betting rests on, so it has to be checked first. Second, Brier is ungameable: there is no threshold to tune, no knob to twist to flatter it, and it punishes overconfidence *and* underconfidence — and it punishes overconfidence *quadratically*, so being confidently wrong costs far more than being timidly wrong. That asymmetry is exactly what you want guarding the gate. Third, it is the experiment-decision metric: every "should we adopt this change?" call in the project is denominated in Brier, so it has to be the measure we trust most.

Be right comes second — accuracy. Accuracy is the share of games called correctly. It is the headline everyone understands and demands, and it is a genuinely useful sanity check — if your threshold behavior is broken, accuracy will catch it. But it cannot sit on top, for one fatal reason: accuracy treats a 51% call and a 95% call identically. Both are just "right" or "wrong." A model could be 74% accurate while being wildly overconfident on the games it got right and barely-there on the ones it missed, and accuracy would never notice. It's a blunt instrument: perfect for a headline, useless for deciding how much to bet. So it ranks below Brier — reported honestly, but never the thing we optimize.

Be paid comes third — risk-adjusted return. This is where "right about football" finally turns into money in the account. Calibrated probabilities get sorted into confidence tiers, each with a known historical hit rate, and the stake on each bet is sized with fractional Kelly (the conservative quarter-Kelly variant; both the tiers and the staking math live in IV.I and IV.IV). This is the layer that produces actual ROI numbers, and it is deliberately *last*, because — and this is the deep point — *you cannot tier-and-stake your way out of a miscalibrated model.* If the probabilities are dishonest (bad Brier), then every tier is mislabeled and every Kelly stake is the wrong size, and no amount of clever bankroll management repairs it. The money layer is only ever as good as the honesty layer beneath it.

So: be honest (Brier) → be right (accuracy) → be paid (tiers/Kelly). Each layer assumes the one above it is solid. Reversing the order — optimizing ROI first, checking honesty later — is the classic way to build a gorgeous backtest sitting on probabilities that were lying the whole time. You tune until the ROI curve looks beautiful, ship it, and discover in live betting that the "edge" was an artifact of fitting your own test data. We put Brier first specifically so that cannot happen here.

III.IV Selectivity: Picking Our Spots

This is the single most important strategic fact about the project, and it is the answer to the question every honest skeptic asks the moment they see the headline

numbers.

> Vegas must post a sharp line on every one of roughly 800 FBS games a season, against the whole world, all the time. We only have to find about five games a week where we're right and the line isn't.

The book *can't pass*. When Akron plays Kent State on a Tuesday night in November, somebody in a sportsbook has to put a number on it, and that number has to survive every sharp bettor on earth trying to pick it off. Vegas is obligated to have an opinion on all 800 games and to make all 800 opinions good enough that nobody can systematically beat them. That is an extraordinary, nearly inhuman standard, and it is the reason the closing line is the sharpest public forecast that exists.

We are under no such obligation. We are allowed to look at sixty games on a given week, shrug at fifty-five of them ("the line looks right, no thanks"), and act only on the handful where our margin head disagrees with the market by a meaningful threshold. We do not have to be smarter than Vegas across all 800 games. We just have to find the few where, *this week, on this game,* we happen to have it more right than the line does.

This is the lever that resolves the central tension in the headline numbers, so it's worth stating the numbers precisely. On the sealed 2025 one-shot (763 games), the model was 73.7% accurate; the market favorite was 74.2% on the same games — the market edges us by half a point. On the 2021–2024 folds (2,928 lined games), the model was 74.0% accurate to the spread-implied market's 72.4% — here *we* edge the market by a point and a half. Put the two together and the honest summary is: across all games, *we are market-equal.* Not better. Not reliably worse. Tied with the best public forecast on Earth, which sounds like a disappointing place to stop — and would be, if betting everything were the plan. It isn't.

Selectivity is what converts a market-equal forecaster into a profitable bettor. Restrict attention to the games where our blind margin disagrees with the closing line by a meaningful amount, and the ATS hit rate climbs well above break-even. Break-even against the standard −110 juice the book charges is 52.4% (the −110/52.4% relationship is derived in II.I). Our hit rate at 3-to-5 points of disagreement is 59.5%. Push to 8-or-more points of disagreement and it is 66.3%. Those are not market-average numbers; they are *selected-subset* numbers, and the selection is the product. The bet thresholds follow directly from this curve: we act at an absolute edge of 3 or more points in weeks 1–7, and tighten to 7 or more from week 8 onward, when the season's data has settled enough that smaller disagreements are more often noise than signal (the week-by-week confidence story is IV.III).

The crucial thing to see is that the headline accuracy and the betting hit rate are *different statistics about different samples.* The 73.7% is the model's accuracy across every game it's asked to call. The 59.5% and 66.3% are its hit rate on the small, deliberately-chosen subset where it disagrees with the market loudly enough to bet. A .250 hitter who somehow only ever swings at fastballs down the middle can post a great average on the pitches he actually swings at; being mediocre across all pitches and lethal on the ones you choose are perfectly compatible. We are tied with the market across all games and ahead of it on the games we choose to bet. Conflating those two samples is the exact mistake the skeptic's question makes.

There's a natural objection: maybe the selected games are just favorites in a chalky era, and favorites win — so the 60% is favorite-bias dressed up, not real discrimination. There is a clean way to kill that, and the model passes it (the full test is reported in X.III): split the same pool by what the EV filter *does*, the picks it keeps versus the picks it rejects. If the filter were just favorite-bias, the rejected favorites would win too. They don't — on the sealed 2025 holdout, kept picks returned +8.5% while rejected picks returned −3.2%, and the rejected pool *loses money*. The gate is genuinely separating mispriced lines from fair ones, not riding chalk. That discrimination — kept wins, rejected loses — is the one finding in this whole document worth staking credibility on, because the favorite-bias story cannot produce it.

So when someone asks "if you're tied with the market, why use this at all?", the honest answer is not "we secretly beat Vegas" — we don't, and you should distrust anyone who claims they do. The answer is that *market-equality is the license to act on the disagreements.* If the model were worse than the market overall, every disagreement would carry the prior "the model is the one that's wrong," and the gaps would be noise. Because we're market-equal, a disagreement is a genuine difference of opinion between two forecasters of roughly equal skill — and *that* is what makes the disagreement tail worth betting. Market-equality isn't the disappointing result. It's the whole foundation.

III.V Why the Mispriced Tail Survives

The previous subsection made a strong claim: the book has to price all 800 games, we get to act on five, and our money lives in the small tail where our margin head and the closing line disagree loudly. A sharp reader should push back, in three escalating ways. First: if a tail of mispriced games is just lying around, why hasn't the army of professional bettors already bet it flat and erased it? Markets are supposed to eat free money. Second: who, exactly, is on the other side of these soft lines — somebody has to be wrong for us to be right, and "Vegas is wrong" sits awkwardly next to the hymn we just sang to the sharpness of the closing line. Third: doesn't the

thesis talk out of both sides of its mouth — is the line "the sharpest public forecast on Earth" or "a price that balances action"?

These are the right questions, and the thesis only gets stronger once they're answered. None of what follows requires believing we out-think Vegas. It requires understanding that a betting market is not a pure forecasting engine; it's a forecasting engine bolted to a commission business that also has to take recreational money. The seams between those three jobs are where the edge lives.

The vig is a structural tax

Start with the most uncomfortable arithmetic in the project, because it reframes everything. A standard −110/−110 game implies 52.4% + 52.4% = 104.8%; the extra 4.8% is the hold — the book's commission, baked into the price (the vig and the de-vig method that strips it back out are both in II.I).

Now a thought experiment. Suppose we built a forecaster whose de-vigged probabilities exactly *equaled* the market's on every game — a perfect mimic, no sharper, no duller. What's our return? On a fair, commission-free price, an exact mimic has an expected edge of precisely zero, because we and the price agree on the true odds. But we don't bet at the fair price. We bet *through the vig*. So our net baseline isn't zero; it's roughly −4.8% before any disagreement, ground down on every bet by a commission we pay whether we win or lose.

Sit with that. Merely tying the de-vigged market is not break-even; it is a slow, guaranteed loss. This is precisely why "we're market-equal on accuracy (73.7% vs the favorite's 74.2%)" is *not* the same as "betting this is break-even." A forecaster tied with the market who bets *everything* is a customer paying the casino 4.8% to play and matching the house on skill. The 4.8% has to be *overcome*, and there is only one place to overcome it: in the disagreement tail, on the specific games where our number is genuinely far enough from the de-vigged line that the gap clears the vig with room to spare.

So selectivity (III.IV) is not a stylistic preference or a risk-management nicety; it is *forced by the arithmetic.* If "tied with the market" is a −4.8% starting line, the only path to a positive return is to refuse the games where we're merely tied and bet only the ones where we're measurably ahead. The body of the distribution — the games where we and the line agree — is structurally a money-loser at any honest skill level short of beating the closing line outright, which we've said plainly we cannot do. The vig doesn't weaken the spot-picking thesis. It *is* the thesis. There is no honest version of this product that bets a full slate.

Limits to arbitrage: why the tail isn't bet to zero

Fine — but if the tail is profitable, efficient markets are supposed to be self-correcting. The moment a soft line appears, sharp money should pour in, move the line to where it belongs, and pocket the difference. After enough of that, the mispricing is gone. So how can a *durable* tail exist?

The answer is a forty-year-old idea in finance: limits to arbitrage. The mechanism that's supposed to erase mispricings is itself constrained, friction-laden, and partial. It works *toward* efficiency without ever fully arriving. Three frictions operate in sports betting, and all three point at exactly the games we claim.

Books cap stakes and limit or ban winners. A sharp who spots a soft line can't simply lever into it until it corrects. Sportsbooks impose maximum bet sizes and — the essential fact — they *limit or close the accounts of people who win.* A bettor who beats the line consistently gets his max bet cut to lunch money or gets shown the door. So the very people best equipped to police mispricings are systematically prevented from sizing into them. The arbitrage force exists, but it's throttled at the source: money that *would* flow in to correct a soft line can't flow in at scale, so the line stays soft longer than a frictionless theory predicts.

Thin markets are the softest, and they're disproportionately ours. A Saturday-afternoon SEC marquee has tens of millions of dollars and every syndicate on Earth pushing on its line; by kickoff it's as close to truth as a number gets. A Tuesday-night Group-of-Five game between two teams nobody outside two towns is watching has a fraction of that money policing it. Less sharp attention means less correction means a wider band of survivable mispricing — and the games where little sharp money does the policing are precisely the games where our model most often claims an edge. The closing line is sharpest exactly where the action is thickest and softest exactly where it's thin, and a market-equal model finds its biggest disagreements in the thin, lightly-policed corners, not in the spotlight games where the world's money has already ground the number to a point. We're not competing for the well-lit lines; we're foraging in the dim ones.

The vig itself is an arbitrage barrier. A sharp correcting a mispriced line *also* pays the 4.8% hold. So a mispricing isn't worth correcting until it exceeds the vig: a line that's "only" 3% off the truth is *unprofitable to bet even though it's wrong,* because the commission eats the 3%. That leaves a permanent band of sub-vig mispricings no rational sharp will ever touch — too wrong to be fair, not wrong enough to beat the commission. The market is efficient *up to the cost of trading,* and below that threshold it simply cannot self-correct. Our job is to live at the edge of that band: take the mispricings big enough to clear the vig, which the throttled, account-limited, thin-market arbitrage force has left on the table.

So the tail isn't bet to zero because the thing that's supposed to bet it to zero is capped, banned, absent from thin markets, and itself taxed. Efficiency is real and powerful; it is also incomplete by construction.

Who's on the other side

Mispricings persisting is necessary but not sufficient. For a *systematic* tail to exist — soft in a predictable direction we can model — someone has to reliably *fund* it. Random noise would wash out; we need a standing source of money that pushes lines off truth in a repeatable way. There is one: recreational bettors.

A sportsbook is not running a pure forecasting contest. It's running a business that *also* must take bets from the public — fans who bet their favorite team, who hammer overs because rooting for points is more fun than rooting for a defensive struggle, who pile onto marquee favorites on national TV because betting the famous team feels safer. This public money has a *direction*, and it's not random. It shades lines toward popular teams, toward favorites in high-profile spots, toward the over. A book that wants balanced action — and books partly do — shades the price to account for where the recreational money is going, which means the posted line is a blend of the sharpest forecast *and* a price calibrated to absorb predictable public flow.

The textbook name for the most famous of these distortions is the favorite–longshot bias: across many betting markets, bettors systematically overpay for longshots (the lottery-ticket thrill of a big payout) and misprice heavy favorites in various spots. The tails of the probability distribution get bent by *how betting feels* rather than *what's likely*, in a direction documented for decades. A calibrated, market-aware model can sometimes *see* that bend, because its de-vigged probability sits where the football says it should while the line sits where the public money dragged it.

So when our model flags a disagreement, part of what it's detecting — not all, but a real part — is the footprint of recreational money. The edge isn't "we forecast better than the syndicates." It's "we can sometimes spot where the line has been shaded away from truth to absorb predictable public bias, and bet the other side." That's a humbler and far more defensible claim, and it carries a built-in expiration warning: it fades as books get better at neutralizing recreational bias without over-shading the line — separating the recreational book from the sharp number, taking the public's money without leaving as much on the table for someone watching the seam. As they improve, the footprint we feed on shrinks. This is the mechanism *underneath* the fade we report honestly elsewhere (the season-by-season decline in kept-pick ROI; see X.III). The edge isn't fading because our model is decaying; it's fading because the inefficiency it harvests is being engineered out of the market.

Two true halves, not a contradiction

Now resolve the tension the sharp reader flagged. Is the line "a price that balances action" or "the sharpest public forecast on Earth"? Both are true, and stated together they stop contradicting:

> The closing line is simultaneously the sharpest publicly available forecast of the outcome *and* a price defended against informed money while absorbing recreational money.

It is not a naive "balance the action" book that just sets the number where equal money lands on each side — that book would be destroyed by sharps and doesn't survive. Modern closing lines are genuinely forecast-sharp, ground toward truth by professional flow. But they are *also* prices, set by a commission business that must take public bets, and so they carry a small, structured residue of where that public money leans. The forecast half is why we can't beat the line on average and would be fools to try. The price half — the recreational shading, the favorite–longshot bend, the thin-market softness the sharps can't fully police — is why a calibrated, selective, market-aware model can find a modest, fading edge in the disagreement tail anyway. Two true halves of one object. The skill is in knowing which half you're exploiting (the price residue, never the forecast) and never confusing the two.

Our own footprint, and the wrong side of the bet

Two final honesties, because the mechanism cuts against us as well as for us.

We move the lines we target, which caps us. The inefficiency we harvest lives in thin, lightly-policed markets — and a thin market is, by definition, one a modest amount of money can move. We publish a free, public pick log designed for people to act on. If that log gains a following, the money it directs at a soft Tuesday-night line is exactly the kind of flow that *corrects* the soft line; we'd be feeding the very arbitrage force we depend on staying weak. This is a self-erasing capacity limit: the strategy works *because* the market is thin and succeeds *only* to the degree it stays small enough not to thicken the market it feeds on. It is a real edge that cannot be scaled into a money machine, by its own mechanism. The honest framing — bet it flat and small, never press it (IV.IV) — isn't only risk control; it's an acknowledgment that the edge has a ceiling baked into its own success.

Systematic disagreement can be the wrong side of the bet. The flip side of "we spot where the public shaded the line" is the case where *we're* the one who's wrong. When we systematically disagree with a sharp, well-policed line, the base-rate-correct prior is often that the market knows something we don't — a late injury, a weather turn, a sharp information edge our frozen feature set can't see (V.IV inventories

these blind spots). Disagreement is not self-justifying; sometimes the reason a line looks soft to us is that we're missing the very fact that makes it correct. This is exactly why the most dangerous cell in the tier table is the one flagged later (see X.III): our highest-confidence picks that collide head-on with the market favorite — a tiny sample where the hit rate drops to a coin flip, the market having fully erased our supposed edge in precisely the spot where we were most sure of it. That sobering cell is the mechanism turned against us, made visible. The tail we look in is real, but the boundary between "the line is soft" and "we're the one being picked off" is thin, and the right defense is the discipline this whole document is built on: bet the residue, respect the forecast, log everything, and let the season grade it.

Part IV — Using It on Gameday

The previous parts argued what the model is for and why a sliver of edge can survive in a market as sharp as this one. This part is the operator's manual: what a pick on the board actually tells you, when in the week to act on it, how to read a thin week against a fat one, how much to stake, and — the part most documents leave out — why a real edge on paper so often dies on the way to a real wager. None of this re-derives the machinery; it tells you how to drive it.

IV.I Reading a Pick

Every entry on the board carries two numbers that, between them, tell you almost everything: a side, and a separation. The side is whoever the model makes the better bet — sometimes a straight win probability (the moneyline read), sometimes a side against the spread. The separation is the gap between our number and the market's number on that same game. That gap is the entire product. We are tied with the market across all games (the head-to-head is in X.II); the only place we expect to make money is where our number and the book's number disagree loudly, and the separation is the size of that disagreement.

So the first thing to read is not whether we like a team. It's *how far* our line sits from the book's. A pick where our margin and the closing line agree to within a point is a pick we will not log, no matter how confident the win probability looks, because there is no daylight to bet into — the market already says what we say. A pick where our margin disagrees with the line by a touchdown is the kind of spot the whole apparatus exists to surface. The selectivity logic behind this — why we shrug at most of the board and act on a handful — is laid out in III.IV and III.V; here, just internalize the operating rule: read the separation, not the affection.

The reason separation matters is that it maps, historically, onto how often the bet won. Against the spread, the hit rate climbs with the size of the disagreement, on a ladder worth memorizing:

- At 3 to 5 points of separation from the line, the historical hit rate is about 59.5%.
- At 8 or more points, it climbs to about 66.3%.

Set those against the only number that decides whether a bet makes money: the break-even hit rate at standard −110 juice, which is 52.4% (the arithmetic is in II.I). A coin flip is not the thing to beat; 52.4% is, because the book takes its cut whether you win or lose. Read that way, 59.5% is not "barely better than a coin flip" — it is seven full points over the only line that counts, and 66.3% is enormous. The ladder is why we set thresholds where we do: at least 3 points of separation in weeks 1 through 7, at least 7 from week 8 on. Early in the season the market is softer, so a smaller gap is genuine value; by midseason the line has sharpened, so we demand a wider gap before we believe the disagreement is ours and not the market's superior information. The shifting threshold is the ladder applied to a market that gets harder to beat as the season ages.

One caution that the ladder cannot show you on its own: a separation that is enormous can be enormous for a reason that has nothing to do with us being smart. When our number screams that the book has a team badly mispriced, one possibility is that we found a soft line — and the other is that the market priced in a fact we are structurally blind to. That distinction is the entire subject of V.IV and a recurring theme of IV.V; for now, hold the thought that the biggest separations are simultaneously the most attractive and the most suspect, and the board flags the worst-offending cell of all — our highest-confidence picks that disagree with the market favorite — for exactly that reason.

IV.II Timing

A betting line is not a fixed price but a living one, and *when* you act changes what you get. Lines open early — usually Sunday night or Monday for the coming Saturday — and sharpen as the week goes on, as money pours in and injury and weather news lands. The number the book sits at right before kickoff is the closing line, and decades of evidence across every sport say it is the single most accurate publicly available estimate of a game's true probability — it has absorbed everything the market could find, including the bets of people far sharper than us. (The closing line's role as the model's external grader is the heart of the falsification plan in XI.I.)

The practical consequence is that the soft numbers — the big gaps in your favor — show up disproportionately *early*, before the line has sharpened, and frequently

vanish by Saturday. So the timing rule is simple: when you see a price better than our projection for our side, take it sooner rather than later. We do not forecast where the line will close and will never tell you "wait, it'll move to −10." We give you one thing: our fair number. Any price better than that, for our side, is value the moment you see it — Tuesday morning or five hours before kickoff. The closing price is the market's job to discover, not ours to predict.

Acting early has a clean payoff that doubles as confirmation. If you bet our side Tuesday and the line later drifts toward your pick, that is the sharp money arriving at the conclusion you reached first. You "beat the close," and beating the close is, over a long run of bets, almost mechanically equivalent to having an edge: you bought the same outcome cheaper than its eventual fair price. It is the cleanest early-warning instrument a bettor has, which is why the 2026 grading plan (XI.I) leans on it. The best sign a bet was right is not that it won on Saturday — one game tells you almost nothing (X.I) — it is that the market moved toward you after you got your number down.

IV.III Confidence Varies by Week

The board always shows a ranked "best bets" list, and the ranking is the most seductive and most misleading thing on it. The trap is to read this week's top five as if they carry the same weight as last week's top five. They do not. The ranking is relative; it sorts *this week's* opportunities against each other, and some weeks the slate simply does not contain a strong spot.

The honest read is the absolute one: the size of the edge and the tier, not the rank. Some weeks our top picks carry fat separations — 8 or more points, high confidence — and sit high on the ladder from IV.I. Those are genuinely strong spots. Other weeks the best the slate offers is a cluster of thin 3-point leans. Those are still our best estimate, and they still clear the early-season threshold, but they sit at the bottom of the ladder, much closer to the break-even line. A "top pick" with an 8-point edge in October is a real spot. A "top pick" with a 3-point edge in November — where the threshold is higher precisely because the market has sharpened — is barely off even, and labeling it number one on the week does not make it stronger. It only means it was the strongest of a weak field.

This is why every pick is tagged twice: by tier (A is most confident, then B, then C — the tiers are defined and priced in IV.IV) and by the raw size of the edge. The two tags exist so that you never have to infer strength from rank. A bettor who flat-bets "this week's top five" every week is implicitly assuming the fifth-best pick in a strong week equals the fifth-best in a weak one, and over a season that assumption quietly bleeds money on the thin weeks. Read the edge. Read the tier. The rank is just the sort order.

IV.IV Staking

Knowing *which* bets are good is half the job; knowing *how much* to put on each is the other half, and getting it wrong can ruin you even when every individual bet is in your favor. The staking rules here are deliberately conservative, for reasons that come straight from the metric hierarchy in III.III: you cannot stake your way out of a model whose probabilities are even slightly off, so the sizing must assume they are.

The picks sort into three confidence tiers, cut along how far the game sits from a coin flip — Tier A for the model's most lopsided games, Tier B for clear-but-real leans, Tier C for near-toss-ups (the full tier construction, the median prices that make A a trap and B the value, and the parlay-vs-single logic live in X.IV). The one-line operating summary: Tier B straight singles are where a probability edge converts cleanest to profit; Tier A's gaudy hit rate is mostly already priced into a steep favorite; Tier C is for tracking calibration, not betting.

The stake on each bet is sized by quarter-Kelly. The Kelly criterion is the mathematically correct answer to "what fraction of my bankroll maximizes long-run growth, given my edge and the odds?" — but full Kelly is optimal only if your probabilities are *exactly* right, and ours, like every model's, are only approximately right. Full Kelly punishes that overconfidence savagely, with stomach-churning drawdowns and a real risk of betting your way to zero despite a genuine edge. The fix is to bet a fraction of the Kelly stake. We use one quarter, which keeps roughly half the long-run growth while cutting the worst-case swings to a small fraction of full Kelly's — insurance against the one thing we know is true, that our numbers are close, never exact. (The full Kelly derivation is in X.IV.)

On top of fractional Kelly sit two hard caps, and they are the part to obey even when a spot looks irresistible:

- Never more than 5% of your bankroll on a single bet.
- Never more than 25% of your bankroll exposed across a single week.

These caps exist because Kelly, even quartered, can occasionally suggest a stake larger than is prudent on one game — and because a single week can stack several correlated bets that, taken together, expose far more of your bankroll than any one of them suggests. The caps are a backstop against both. And the temperament to go with them: expect to lose roughly 40% of *weeks* even when everything is working exactly as designed. A losing Saturday is not evidence the model is broken — that confusion is exactly what X.I exists to cure — it is the ordinary texture of a small, real edge. Bet flat, bet small, obey the caps, and judge the season, not the week.

IV.V Execution Reality

Suppose you grant everything above — that the model edge is real, modest, and validated. There is still one cliff between that edge and a dollar in your account. The number we measured is a *measured* edge; a *bettable* edge is a different thing, and it is smaller, sometimes all the way to zero. This is the single highest-value thing a professional bettor knows that a backtest does not: the backtest assumes a frictionless world — perfect prices, infinite limits, nobody watching — and the moment you try to place the bet, every one of those assumptions charges rent. The validated edge sits on a lower bound of about +0.44% (the LB95 from X.III). At that altitude, friction is not a rounding error; friction is the whole conversation.

The friction arrives in a few predictable forms, and it is worth knowing them in the order they cost you money.

First, line shopping is not optional; it is essential. A backtest pretends there is one price for a game. There is not: a −200 favorite might be −195 at one book, −205 at a second, −210 at a third. The difference between laying −105 and −115 to win the same payout is, in expected-value terms, *larger than the entire +0.44% proven floor.* A bettor who takes the first price he sees is, on these numbers, a bettor with no edge at all — the model can be right about the game and still lose because the bettor paid ten cents too much for the privilege of being right. So shop every bet across several books, take the best number, and treat line shopping as a wall the strategy rests on, not an optimization at the margin.

Second, the limits are coming, and getting booted is a feather in the cap. A sportsbook profiles its customers, and one who consistently beats the closing line gets flagged fast — often within weeks — then limited (your maximum bet quietly slides from $500 to $50 to $5) or banned outright. The cruel geometry is that our edge lives exactly in the moderate favorites the books watch hardest, because those markets are liquid and where sharp money lives. So the live sequence is predictable: you bet the gated picks, they work, the book flags you, and your stake gets harvested fifty dollars at a time while the bets you can still get down at full size are the ones the book is *happy* to take. The reassuring inversion: if a book ever limits you to pennies or closes your account, that is not the model failing — it is the market confirming the edge was real enough to defend against. The goal is not to avoid it forever; it is to harvest as much as you can, across as many books as you can, before it happens. Spread your action, take the limits as a compliment, and only ever bet money you can afford to lose.

Third, there is a subtler tax that has nothing to do with the books and everything to do with *which bets look most attractive*. The bigger a separation the model flags, the higher the chance the disagreement exists because the market priced in information

the model cannot see — a Thursday quarterback scratch, a weather front, a suspension that hit the wire after our features froze. The very bets that look most attractive to a naive filter are *selected* to be the ones where you are most likely to be the sucker. This is the deep reason our highest-confidence-against-the-favorite spots get flagged and ejected from parlays; it is treated in full in V.IV and is the structural counterweight to the hit-rate ladder.

Where all of this lands is a single distinction, the most important takeaway in this part. The model edge — the accuracy, the gate's keep-versus-reject separation, the return that survived a sealed one-shot — is validated. But the *bettable* edge — what survives line shopping, account limits, the public decay of any posted pick, adverse selection, and a stake-sizing pipeline not yet fully guarded — is not validated, and is necessarily smaller. Every form of friction subtracts from a gap that started at +0.44%. So do not judge the strategy by the backtest. The backtest answers "was the model right about football?" — a careful yes. The only thing that answers "is there money in it for a real person at a real book?" is the live, public, timestamped log graded across the 2026 season (XI.I). That log is the only court with jurisdiction over the bettable edge, and the honest expectation going in is that it throws out some of the measured edge, maybe most of it. "Promising, not proven" was always a statement about the model. The bettable edge has not yet earned even that.

Part V — Inputs, Outputs, and Trust

This part draws the map of the whole machine — what goes in, what comes out, and, crucially, how much faith to place in each thing that comes out. The single most important idea here is that not all of the model's outputs deserve equal trust. A bet that gets graded against a real outcome is a fundamentally more trustworthy object than a season-long simulation that compounds a dozen guesses, even though both wear the same confident decimal places. By the end you should be able to look at any number the app shows you and place it correctly on a trust ladder.

V.I Where the Engine Feeds

Start with the plain accounting: what the model reads, and what it produces. The inputs are a handful of public, settled facts about each team and game:

- Bill Connelly's SP+ ratings — a respected efficiency measure of overall team quality.

- Elo-style power ratings — a rating that rises and falls with wins and losses, weighted by opponent.
- Recruiting — a multi-year composite of high-school recruiting rankings, a proxy for raw talent.
- Returning production — how much of last year's roster (and its on-field value) is back.
- Form — recent scoring, captured as an in-season blend of points per game that updates as games are played.
- The betting line — the market's own number on the game, which the model treats as a feature to disagree with, not a thing to copy.
- Situational signals — home field, rivalry, and the like.

Every one of these is point-in-time: the model reads only what was settled *before* the game it is predicting. That discipline is what keeps the track record auditable, and it is also the source of every blind spot in V.IV. The mechanics of the point-in-time design, and the leak that taught us to enforce it, are in VII.I; the full feature roster and how it was chosen are in IX.I.

From those inputs the engine produces four distinct kinds of output, surfaced across the app:

- The locked pick log (under the Edge tab). A small number of bets, written to an immutable record before kickoff and graded after against what actually happened. This is the only output that gets checked against reality.
- Single-game predictions (Predict and Matchups). For any one game, a win probability and a projected margin.
- Power rankings and team compare. The model's overall ordering of teams and its side-by-side breakdowns, driven by the same engine.
- The season and playoff simulator. A Monte Carlo engine that plays the rest of the schedule thousands of times to produce things like "86% to make the playoff."

All four are powered by the same underlying machine. They are not, however, equally trustworthy — and that is the subject of the next section.

V.II Trust Levels by Output

Here is the single principle to carry out of this entire document: trust degrades the further an output sits from a directly-graded bet. A number that has been checked against real outcomes is the gold standard. A number that has been sealed-tested but isn't itself a graded bet is next. A number that reflects our methodology but has never been validated against any outcome is directional only. And a number that com-

pounds many such un-validated guesses across a whole season is the softest of all — a vibe with a decimal point. Ranked from most to least trustworthy:

1. The locked pick log — most trustworthy, because it is graded against real outcomes. Every entry is timestamped before kickoff and scored afterward against what happened, win or lose, push excluded. It is the only output that cannot flatter itself: the games either came in or they didn't. This is the court of record for the whole strategy, and the thing to judge the model by. (How it is graded and what would falsify it: XI.I.)

2. Single-game win probability and margin — second, because they are sealed-validated. The model's per-game accuracy, calibration, and discrimination were measured once on a holdout season the design never touched: 73.7% accuracy, Brier 0.1712, AUC 0.816 on 763 games. That is an honest, out-of-sample grade, which is why these carry weight. They sit just below the pick log only because a single-game prediction is a probability you have to *interpret*, not an outcome already on the books. (Each number is unpacked in X.I.)

3. Power rankings and team compare — third, and directional only. These are our methodology applied honestly, but they have never been validated against an outcome, because a power ranking has none — there is no "the rankings were right" event the way a bet wins or loses. They are an internally consistent ordering under our chosen weights, useful for orientation and argument, but a *view*, not a verified forecast. Treat a ranking as "this is how the engine sees it," not a settled fact. (The engine and its weights: IX.I.)

4. The season and playoff simulator — least trustworthy, and explicitly directional, not precise. A Monte Carlo simulation plays the remaining schedule thousands of times, compounding the uncertainty of twelve or more games, each already carrying the per-game error from level 2. Errors do not cancel as they stack; they multiply. So "86% to make the playoff" should be read as a vibe, not a guarantee — it means "the engine, run forward many times under its own assumptions, lands there more often than not," a genuinely weaker claim than "73.7% accurate on graded games." Read the simulator for shape and relative standing, never for the third decimal place.

The ladder ranks not how *useful* these outputs are — all four are useful — but how much each has earned the right to be believed. The further down, the more the number is a projection of our methodology and the less it has been checked against the world. When two outputs disagree — the simulator loving a team the pick log has been fading — trust the one nearer the top.

V.III Error Bars and Sample Sizes

Every figure in this document is an estimate, not a fact carved in stone, and the honest question about any of them is never "what's the number?" but "how far off could it reasonably be?" The tool that answers it is the error bar, and the rule that governs it is short: a number without its sample size is half a number. The full derivation of standard error and confidence intervals lives in II.II; here we apply it to the headline figures so you can see which ones to lean on and which are noise that only looks like signal.

Take the headline accuracy first: 73.7% on 763 games. The standard error of a hit rate is the square root of p(1−p)/n, here sqrt(0.737 × 0.263 / 763) ≈ 0.016, about 1.6 points. The 95% confidence interval is the rate plus or minus 1.96 standard errors: 0.737 ± 0.031, running from roughly 70.6% to 76.8%. So the honest reading of the headline is not "the model is 73.7% accurate" but "the true accuracy is very likely between about 70.6% and 76.8%, and 73.7% is the best single guess in that band." That band is almost six points wide — which is exactly why we do not over-read the market favorite's 74.2% on those same games. The two bands overlap almost completely; a half-point gap on 763 games is well inside the noise. The defensible statement is that we are statistically indistinguishable from the market on 2025 accuracy, not that the market beat us.

Now the opposite end of the trust spectrum. Tier A — the model's highest-confidence picks — hit 88.4% across 1,251 picks on the folds. Its standard error is sqrt(0.884 × 0.116 / 1,251) ≈ 0.009, under a point. The confidence interval is 0.884 ± 0.018, roughly 86.6% to 90.2%: a band under four points wide, sitting comfortably high, with even its pessimistic edge an excellent hit rate. Two things made it tight — a big n drowning out luck, and a rate far from 50%, where there is simply less inherent randomness in the p(1−p) term. This is what a number you can actually lean on looks like.

Set against those, the n=18 trap. Buried in the betting analysis is a cell that looks like a finding: games where our highest-confidence picks disagreed with the market favorite, on which the model hit 50%. Put the error bar on it before acting: sqrt(0.50 × 0.50 / 18) ≈ 0.118, about 11.8 points. The interval is 0.50 ± 0.231, stretching from roughly 27% — catastrophically worse than a coin — to 73% — crushing it. The data cannot tell "we're terrible at fading the market" from "we're great at it." The cell tells us *nothing*; it is not a 50% finding, it is an n=18 shrug. This matters beyond one cell, because small samples appear the moment you slice the season finely — "the model on road dogs in November," "Tier A in rivalry games" — and each thin slice reads like a precise insight while carrying a ±15-to-25-point error bar. The numbers that carry weight here are the aggregate ones over thousands of games; the thin slices are the noise they are.

Finally the betting headline, which is trickier because it is a return, not a hit rate, and returns swing harder than win rates — winning bets pay different amounts, so the dollar outcome bounces more than a simple win/loss would. For the EV-gated policy on the sealed 2025 season — +8.46% over 165 bets — the right tool is the one-sided 95% lower bound: the pessimistic edge of the interval, the answer to "if luck broke against us, how low could the true edge plausibly be?" That lower bound is +0.44%. Read it carefully, because it is the most honest number in the betting story: the best guess is +8.46%, but the bottom of the plausible range is *just barely* above zero. The edge is very likely real, but it is small, and 165 bets cannot prove a fat one. That single fact — point estimate handsome, lower bound a whisker over break-even — is the entire reason the verdict is "promising, not proven." (The full betting-number accounting, including the real-prices-only cut, is in X.III.)

The portable version of all of this fits on a card. Ask "out of how many?" — no n, no trust. Eyeball the width: n in the teens gives ±15-to-25 points (noise); n in the hundreds gives ±3-to-5 (suggestive to solid); n in the thousands gives ±1-to-2 (trustworthy). For a return, check the lower bound, not the headline. And remember that a single season is a single draw from the bell curve: one-shot numbers are honest *and* wobbly, so don't crown or bury the model on one sample.

V.IV Model's Blindspots

The honest way to describe a model is not to list what it can see — that is a sales sheet — but to list, plainly, what it cannot see and never will by design. The thread through all of it is the point-in-time discipline from V.I: to guarantee that a week-5 prediction was built only from information that existed before week 5 kicked off, the design reads only settled history. That is what makes the track record auditable, and it is also, unavoidably, what blinds the model to anything that happens *this week*. Here is what lives in that blind spot. None of these are bugs; each is a disclosed trade-off.

Confirmed quarterback availability, and in-week QB injuries. This is the big one — the single largest week-to-week swing in the sport. A starting quarterback ruled out Thursday, or knocked out in the first quarter on Saturday, can move a line ten points or more; nothing else moves it like that. The model carries quarterback signal through prior-week production gaps, but it knows only who the quarterback *was* — not a Thursday scratch, a game-time decision, or a mid-game hook to the backup. It walks in assuming the depth chart it last saw still holds, and in the portal-and-injury era that breaks loudly several times a season. When the model confidently rates a team it has no idea is starting its third-stringer, this is why.

Transfer-portal roster divergence. The recruiting feature is a multi-year composite of high-school rankings — a sensible talent proxy in the era it was trained on. But

the portal has severed the link between who a school *recruited* and who actually *suits up*: a roster can be half rebuilt in an offseason, stars leaving and replacements arriving that the composite never credited to this team. So the recruiting average and the returning-production estimate can diverge sharply from the real roster on the field. There is a subtler cost, too — it makes the model's early-season priors noisier than they were in the pre-portal seasons that anchor the training data, a genuine concern that the world the model learned on is drifting under its feet. (The training window and why it skews pre-portal are in VI.III.)

Game-day weather. Wind, rain, snow, and cold at kickoff suppress passing and depress scoring, and the sharpest bettors refresh the radar for hours before a game. The market prices it; our settled-inputs feature set does not contain it, because kickoff weather is by definition information that does not yet exist when the model reads. We bring a forecast-free worldview to a game that may be decided by a 25-mph crosswind — one of the clearest cases of the market holding a card we chose not to draw.

Scheme and coordinator mismatch. The coaching feature is a head-coach win-percentage gap — a measure of track record, not of style. It has no representation of scheme: an option attack meeting a defense that can't simulate it, an air raid against a run-stopping secondary, a defensive coordinator who left in the offseason and took the unit's identity with him. Stylistic matchups — the reason a "worse" team on paper routinely troubles a "better" one every year — are invisible to a win-percentage number, as is coordinator turnover that quietly remakes a team's ceiling.

Playoff-era motivation and stakes. People play differently when the stakes change, and the stakes now change constantly. The twelve-team playoff has created a thicket of bid-chasing scenarios; bowl season is a minefield of NFL-bound opt-outs, interim coaches, and wildly uneven motivation between a team playing for a title and one that checked out in November. Rivalry intensity, a senior day, a coach on the hot seat — these are human variables that move outcomes and that no settled-stats feature can encode. The model reads talent and form; it cannot read *why this game matters to these players today.*

These blind spots are less a reason to distrust the model than a guide to *where* its outputs are weakest and what the next real improvement looks like. They are exactly why the betting edge lives in sides and not totals — the very inputs that drive a game's point total (quarterback availability, weather, opt-outs, pace) are the ones in this list, so the model brings its worst hand to the totals market and its best hand to the sides market (the totals decision is settled in X.IV). And they map where future gains live: the road to a smarter model is not a fancier machine but new inputs — confirmed inactives at lock time, live kickoff weather, a scheme signal, a stakes flag — each attacking one named blind spot above, and each held to the same pre-regis-

tered keep rule every other idea faced (XI.III). The bargain is plain: we gave up game-time information to get a track record you can actually check. A model that tells you what it cannot see is the only kind you can responsibly bet behind.

❦

Part VI — Data and Target

Before we touch architecture, blends, or betting tiers, we have to be honest about the two things every model is built from: what it learns from, and what it is trying to guess. Get either of these wrong and nothing downstream can save you — a flawless engine pointed at the wrong destination just gets you to the wrong place faster. So this Part moves slowly. It describes the rows the model studies, the exact quantity it is asked to predict, and the window of history we let it look at. The methods used here — ordinary least squares, regression versus classification, the natural logarithm — are defined in II.II; here we apply them rather than re-teach them.

VI.I The Data

Every row the model trains on is one game. There are 7,179 of them, all graded — meaning the game has been played and the final score is known — spanning the 2016 through 2025 seasons. The full database is larger than that, because it also holds the 2026 schedule with no scores attached yet and other not-yet-played rows, but only the graded games train the model. You cannot learn from a flashcard whose back is blank.

For each game we record the obvious facts — the two teams, the final score, the season, the week, the conference — and then a few dozen computed numbers describing what was knowable about the two teams going in: efficiency ratings, recruiting strength, scoring rates, the betting spread where one exists, and so on. The full roster of those numbers, and how we decided which ones earned a seat, is the subject of Part IX. For now the only thing that matters is the shape of a row: a front (the descriptive numbers) and a back (the result).

A single training row for "Alabama at Texas, week 5 of 2023" reads, in spirit: Alabama's rating sits here, Texas's rating sits there, Alabama is favored by a few points, the game is in Austin — and on the back, Texas won by ten. The model never learns that this row was "the famous Texas upset." It only ever sees the numbers on the front and the answer on the back, and across 7,179 such flashcards it discovers a single thing: when the fronts look like this, the backs tend to look like that. That learned mapping — fronts to backs — is the model. Nothing mystical lives inside it.

It is the set of multipliers that made the smallest total error across games whose answers we already knew.

The data comes from three public sources, named once here and used throughout: the College Football Data API (the backbone — ratings, results, schedules), ESPN's public scoreboard (live and supplementary scores), and TheOddsAPI (betting lines). The provenance details and reproducibility notes live in XI.II.

The choice of where the row count comes from is worth one clarifying line, because the number 7,179 differs from other game counts elsewhere in this document. Those 7,179 are *all* graded games used in training. When we later report a market benchmark (the closing-line favorite versus our model), that comparison can only run on games that actually *had* a betting line — a smaller pool, since many low-profile games never get a widely posted spread. So "7,179 training games" and "2,928 lined fold games" are not in conflict; they are two different slices, and we will always say which slice a number came from. Keeping those pools labeled is not pedantry — confusing them is exactly the kind of slip that lets a flattering number hide.

VI.II Our Target

Now the other half of every flashcard: what, precisely, is the model asked to guess?

The instinctive answer — "did the home team win?" — is the wrong one, and replacing it was the single largest improvement in this model's history. The distinction is between classification and regression, defined in II.II: a classifier answers a yes/no question and outputs a probability of "yes"; a regression predicts a number on a scale. Through v8.6 our model was a classifier: home win, yes or no. From v9.2 on it became a regression. It predicts the *margin of victory* — how many points one team wins or loses by.

Why margin instead of win/lose? Because a yes/no target throws away enormous information. To a classifier, a three-point escape and a thirty-five-point annihilation are *both just "1"* — both are "the home team won," utterly indistinguishable. But those two games say wildly different things about how good the teams are. Squeaking by tells you the teams are close; blowing the doors off tells you one team is far superior. Regressing the margin lets *every game teach the model how much better one team was*, not merely that it was better. Each flashcard now carries a rich answer (+35) instead of a single bit (yes).

But we do not regress the raw margin. We run it through a dampening transform first:

```
target = sign(margin) × log(|margin| + 1)
```

Here `log` is the natural logarithm (see II.II), `|margin|` is the margin's size ignoring sign, and `sign(margin)` is +1 for a home win and −1 for a home loss. The reason for the transform is that raw margin over-rewards blowouts, and blowout points are mostly noise. Once a game is 42–7 in the third quarter, the starters come out, the losing team stops trying, and the remaining garbage-time points tell you almost nothing new about true team strength. A raw-margin regression treats the 49th point of margin as exactly as informative as the 1st. We do not want the model obsessing over whether a rout finished at 45 or 52; we want it to care intensely about the gap between a 3-point game and a 14-point game, which is where the real information about team quality lives.

The logarithm compresses large numbers far more than small ones, which is exactly the behavior we want. Here is the transform on five representative margins, with the arithmetic spelled out so there is no sleight of hand:

Real margin	`log(margin + 1)`	The arithmetic
3	1.39	`log(3 + 1) = log(4) = 1.386…`
7	2.08	`log(7 + 1) = log(8) = 2.079…`
14	2.71	`log(14 + 1) = log(15) = 2.708…`
28	3.37	`log(28 + 1) = log(29) = 3.367…`
49	3.91	`log(49 + 1) = log(50) = 3.912…`

The `+ 1` inside the logarithm is a small technical nicety: it keeps a zero-point margin mapping to `log(1) = 0` instead of plunging to negative infinity, which `log(0)` would do. And `sign(margin)` out front simply restores direction, so a 14-point *loss* becomes −2.71, not +2.71, and the model still knows who won.

Now watch what the compression *does*, by reading the gaps between consecutive rows:

- Going from a 3-point win to a 14-point win moves the target 1.32 units (2.71 − 1.39). A big jump — and it should be, because that is the difference between "basically a coin flip" and "clearly the better team."
- Going from a 28-point win to a 49-point win moves the target only 0.54 units (3.91 − 3.37). A small jump — and it should be, because that is merely the difference between "blowout" and "bigger blowout," which tells us nothing extra about who is good.

So the same 21-point change in raw score is worth less and less the further out you go. The everyday analogy is temperature: the difference between a 60°F day and a 75°F day changes how you dress completely, but the difference between a 95°F day

and a 110°F day is just "miserably hot, stay inside either way" — the practical information saturates. Log scaling captures exactly that. Early differences are meaningful; extreme differences blur together. This is the same insight behind the margin-of-victory dampening in chess and football rating systems, and it is the heart of why the model treats a close game as a far more informative teacher than a rout.

There are receipts. Measured by out-of-fold Brier — the project's decision metric, defined in VIII.IV and treated fully in X.I, read for now as "lower is better, measured honestly on games the model never trained on" — the choice of target ranked as follows in the experiment harness (all-games pool): log-dampened margin came in best, capped-margin and raw margin close behind, and binary win/loss a clear distance worse, at 0.17254 against the dampened family's ≈0.17085. The entire margin family beats the classifier, and dampening wins within the family. That gap over the binary classifier was the largest single improvement of the whole v9.1 experiment campaign — larger than any architectural cleverness we tried afterward (Part IX). The lesson worth carrying out of this Part is blunt: choosing *what* to predict mattered more than any later choice about *how* to predict it.

VI.III The Training Window

The last decision about the data is which years to include, and how much to weight each one. A 2016 game and a 2024 game are not worth the same. College football changes — rosters turn over, rules shift, the transfer portal rewired the sport. A pattern that held in 2016 may be stale; a pattern from last season is probably still live. So we do not treat all the data equally. We tell the model to trust recent games more, by attaching to each training row a weight — a multiplier on how much that flashcard counts during learning.

The weight follows an exponential-decay rule:

```
weight = exp(−0.2 × age)
```

where `age` is how many seasons ago the game was played, relative to the most recent full season in the data. The exponential is the same curve that describes radioactive decay or compound interest running in reverse: each additional year of age multiplies the weight by the same fraction, so importance fades smoothly rather than dropping off a cliff. Turn the crank on two games:

- A current-era game (`age = 0`): weight `= exp(−0.2 × 0) = exp(0) = 1`. Full strength.
- A 2016 game eight seasons back (`age = 8`): weight `= exp(−0.2 × 8) = exp(−1.6) ≈ 0.20`.

So a 2016 game enters the model's study session at roughly one-fifth the importance of a current game — equivalently, a recent game is worth about 5× an eight-year-old one ($1 \div 0.20 = 5$).

Two natural questions: why start at 2016 at all, and why the decay rate 0.2 rather than 0.1 or 0.4? Both were settled by experiment, not taste. We swept a grid of training-start years (2016 / 2018 / 2021) crossed with decay rates (0 / 0.2 / 0.4) and graded every cell by out-of-fold Brier. The 2016 start with a decay of 0.2 won. Truncating the window to a 2018 start moved the metric by less than 0.0006 — noise. A 2021 start was the worst arm everywhere, and it left the earliest cross-validation folds with almost no training data to learn from. Pushing the window the *other* direction, into 2014–15, would only add rows that enter at about 13% effective weight under the decay anyway, on top of degraded supporting data (no cached ratings for those years); if *removing* two recent fully-weighted seasons barely moves the needle, adding two ancient, heavily-discounted ones cannot move it either. The year 2016 also happens to mark the point where the underlying data coverage stabilizes and the modern playoff era matures.

The deeper point is that the decay weighting already encodes the legitimate kernel of every "just use recent years" argument — trust the present more than the past — but does it smoothly, on a dial, rather than with a hard cutoff that throws away the faint-but-real signal still living in older games. The same `exp(−0.2 · age)` weights reappear inside the validation folds (VIII.IV), so within any training window the recency tilt is consistent everywhere the model learns.

Part VII — The Leak

This is the most important Part in the document, and the one to re-read once a year. It is the story of how a model can be honest in every line of its code, validated with textbook-correct procedure, audited repeatedly — and still be lying to you by ten percentage points. We built that lie ourselves, shipped it four times, defended it against our own suspicions, and finally caught it. Understanding *how we got it wrong* is as essential as understanding the architecture, because the trap that caught us is the single most common way sports and finance models go wrong, and it is nearly invisible from the inside.

VII.I The Leak

A *data leak* is when a feature secretly contains information that would not actually be available at prediction time — like a flashcard whose front has a hidden peek at the back. The model learns to read that peek, looks brilliant on historical flashcards, then falls apart in the real world where the peek does not exist. Ours was a specific, sneaky variety called *season-aggregate leakage*. Here is exactly how it worked.

Most of our features were built from season-level tables: a team's rating *for 2023*, its points-per-game *for 2023*, its win% *for 2023*. These are season *totals* — they summarize the whole year. The problem is that a whole-year summary necessarily includes every game in that year, including the very game we are trying to predict, and including games that happened weeks *after* it.

The worked example makes it concrete. Take Alabama versus Texas, week 5 of 2023. We want to train the model on this game, so we build its flashcard. On the front, among other features, we attach Alabama's 2023 points-per-game, Alabama's 2023 win%, and Alabama's 2023 rating.

Now stop and look at what we just did. It is week 5. But "Alabama's 2023 points-per-game" is the average across *all* of Alabama's 2023 games — including the Texas game in this very row, and including the eight games Alabama had not yet played as of that Saturday. We handed the model, on the front of the flashcard, a number that was secretly computed from the back of the flashcard, with a chunk of the future stacked on top.

Texas won that game, 34–24. That upset dragged Alabama's *final* 2023 numbers down — its season win% finished lower, its season point margin thinner, its season rating dinged — relative to where they would have landed had Alabama won. So when the model studied this flashcard, Alabama's "form" features were *already faintly stained by the loss the model was supposed to predict*. The model, doing exactly its job, learned a rule: "when the favorite's season-summary numbers look a touch soft, lean toward the upset." It looked like genius. It was reading the answer key.

The stain is not dramatic in any single feature. It is a faint smear, because one game is only about one-twelfth of a season average. But it is smeared across *many* features at once — the rating gap, the scoring-margin gaps, the win% gap, the momentum and turnover gaps, the opponent-adjusted ratings, the advanced stats — and a model is brilliant at summing faint correlated whispers into a confident shout. Added across all of them, that whisper was worth about ten percentage points of fake accuracy.

The geometry of the mistake is the part to internalize, because it is what made the leak invisible. There are two kinds of time-leakage, and they live on different axes.

Picture a calendar with seasons running left to right:

```
  2021         2022         2023                                2024
[season] ---> [season] ---> [========  S E A S O N  ========] ---> [season]
                                 ↑week5          ↑week13
                              the game         end of year
                             we predict        (its own future)
```

An *across-season* leak would be training on 2024 to predict 2023 — using a later season to forecast an earlier one. We guarded that boundary perfectly. The leak we missed was *within-season*: the week-5 game's features reached rightward *along its own season* to week 13 and the end of the year. The across-season wall was airtight; the within-season door was wide open, and we never thought to check it because all our attention was on the wall.

VII.II Why We Missed It

Our validation protocol was walk-forward cross-validation, defined in II.II and detailed in VIII.IV. It is the correct technique, and it did its one job flawlessly: it guaranteed we never trained on a future season to predict a past one. We always trained on, say, 2016–2022 to predict 2023, never the reverse. That seals the across-season boundary, and we were rightly proud of it.

But walk-forward CV operates at the level of *whole seasons*. It draws its train/test line *between* years and never once looks *inside* a year to ask whether a week-5 row's features were built from week-13 data. It was a security guard who checks everyone entering the building and never notices the problem is coming from inside the office. The protocol was correct, and blind, at the same time. The single wrong assumption was a level-of-abstraction error: "we use walk-forward CV, therefore there is no temporal leakage." The truth was narrower — walk-forward CV guarantees *across-season* integrity, full stop. It says nothing about within-season leakage, which is precisely where ours lived.

That one wrong assumption was reinforced by three mistakes in how we read evidence, and each is worth naming because each is a trap any modeler can fall into.

First, every audit found a *real* leak — a different one each time — and fixing it barely moved the headline number. An earlier version fixed a genuine player-feature leak and a genuine cross-validation contamination; accuracy ticked down a point or so and we read that as reassurance: "we audit hard, the number survives audits, so the number is robust." The correct reading was the opposite: the dominant leak was still in there, dwarfing everything we kept finding. A number that survives repeated small corrections is not necessarily clean; it may simply have one big problem in-

stead of many small ones. We were bailing a boat with a teacup while ignoring the open seacock. Every cup of water was real water removed, and the boat kept sinking.

Second — and this is the one that should have been a fire alarm — we never computed the one benchmark that could not be fooled. The market favorite, meaning "just pick whoever the closing line favors," scores 74.2% on 2025. Any model claiming 83.9% is claiming to soundly beat the most efficient prediction market on earth by nearly ten points, an edge that would be worth millions of dollars a season to anyone who actually held it. That sentence, written down, is absurd on its face. We did not write it until June 2026, because we had been benchmarking against *naive* baselines — the home team wins about 59.5% of the time, a raw rating system gets into the low 70s — which we beat plausibly, instead of against the *sharp* one, which would have called our number impossible. The lesson: always benchmark against the strongest available predictor, because only it has the standing to tell you your number is fantasy.

Third, suspicion without a mechanism does not act. We were always a little uneasy about 83% — old internal notes literally said "say 80–82% in conversation," a hedge that admits the discomfort. But the unease sat idle because we had no specific thing to investigate. The break came only when the audit question changed from "is this number too good?" (a vibe) to "reconstruct exactly what information existed on the morning of each game, for every feature" (a mechanism). The second question found the leak in an afternoon. Good hygiene aimed in the wrong direction is indistinguishable from rigor, right up until the moment it isn't.

VII.III The Fix

The cure has a name: *point-in-time features*. The principle is one sentence — every feature must be reconstructed using only information that existed before the opening kickoff of the game it describes. No reaching rightward along the calendar. No season totals that include the game itself. You rebuild each feature as a photograph of what was true that morning, not a summary written the following January.

In code, the feature builder now runs in point-in-time mode as its only mode, for both training and serving, and both read the same flag from the saved model artifact so the two paths match (a past bug was exactly this kind of train/serve drift — see VIII.III). In candor, this is a best-effort mirror, not a hard guarantee: an artifact that somehow lacked the flag would silently fall back to the old leaky path rather than refuse to run. A serve-time guard that hard-refuses any artifact missing its point-in-time, variant, and architecture flags is recommended hardening, staged for a dedicated engine session. We name that gap rather than paper over it.

For a game in season S, week W, here is what each feature is now allowed to see:

- Ratings and advanced stats: the *prior* season's values (S−1), with a weighted variant blending mostly S−1 and a little of S−2. During season S we never touch season-S published ratings at all, because a published season rating is itself a season-aggregate — the exact thing that bit us.
- Win%: a progressive in-season blend against the prior *two* seasons.
- Scoring rates (points scored and allowed): a progressive in-season blend, described next.
- Momentum: current-season win% minus prior-season win%, which by construction can only use games already played.
- Opponent-adjusted ratings: recomputed for each (season, week) using *only* games with week earlier than W; early in the season it falls back to the prior season's full ratings.

The trickiest of these is the in-season scoring blend, and it is worth making fully concrete because it captures a genuinely elegant idea: trust this season's numbers exactly as fast as this season earns the right to be trusted. Early on, you have almost no current data — after one game, "this team averages 38 points" is nearly meaningless — so you lean on last season's known quantities. As the weeks accumulate and the current sample grows, you shift trust toward what is happening now. The formula:

```
weight on THIS season's games = min(1, n_games / 6)
weight on LAST season         = 1 − (that weight)
```

where `n_games` counts games this team has *already played* before the game being predicted. The `min(1, …)` caps the weight at 1 — once full trust is earned, more games cannot push past 100%. The "6" is the ramp length: it takes about six games for the current season to fully take over. Two worked weeks:

- Week 3 (about 2 games played): weight on this season `= min(1, 2/6) = 0.33`. A week-3 prediction blends roughly one-third this season, two-thirds last. Reasonable — two games is a hint, not a verdict.
- Week 10 (about 8 games played): weight on this season `= min(1, 8/6) = 1.0`, the cap. A week-10 prediction is essentially all current season. Also reasonable — eight games is a real sample, and last year's roster is increasingly ancient history.

The beauty is that this is automatic and leak-free. It never peeks rightward; it only ever counts games already played, and it dials its own confidence up smoothly as evidence accumulates, the way a careful human handicapper's mind works across a season. Win% uses the same machinery with a slower ramp (8 instead of 6) against

the prior *two* seasons, because win% is noisier than scoring and deserves a longer memory.

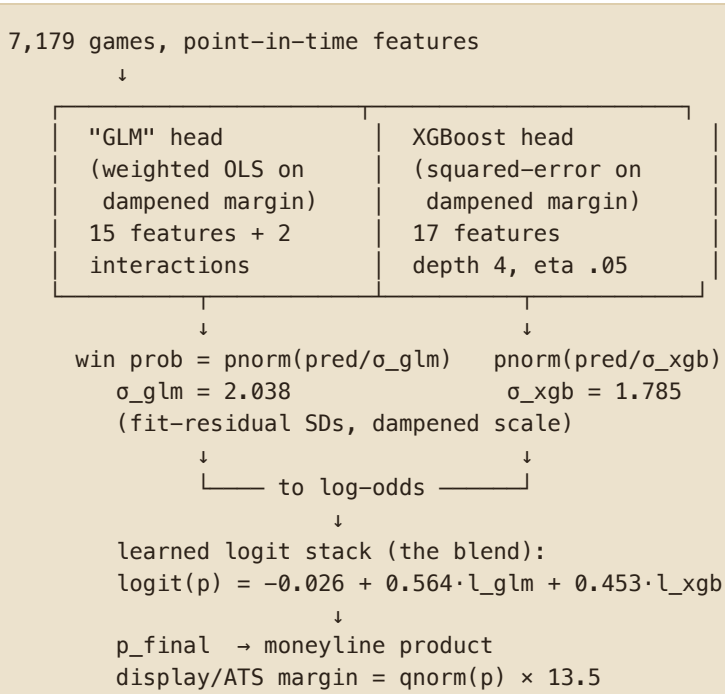
Now the honest cost, stated as plainly as it can be. Closing the leak dropped measured accuracy from 83.9% to 73.7%. We did not lose ten points of skill. We gave back ten points of fiction. Those ten points were never real — they were the model reciting outcomes it had been secretly shown on the front of the flashcard. When we slammed that window shut, accuracy fell to what the model had *always* actually been capable of on information available before kickoff: 73.7%, with a Brier of 0.1712 and an AUC of 0.816, on the sealed 2025 one-shot of 763 games. (An earlier development run of the same frozen design scored 73.9% / 0.1711 on the identical protocol; the production artifact that actually serves scored 73.7% / 0.1712. The two-game difference is run-to-run variance, and we cite the artifact that serves. Every number this project published before June 2026 — 85.1%, 83.9%, 83.6%, Brier 0.1200 — is retracted, citable only as this warning.)

The smaller number is the reassuring one. A model that genuinely beat Vegas by ten points would not be a hobby project; it would be a hedge fund. The 73.7% sits right alongside the sharpest forecaster on earth — the market favorite scored 74.2% on the very same games — which is exactly where an honest, market-aware college-football model *should* land. Every strategic decision the project makes downstream (Part III) flows from accepting that 73.7%, not 83.9%, is the real starting line. The leak did not merely inflate a stat; believing it would have aimed the entire betting strategy at a fantasy edge that was not there.

One last note, in the spirit of the rest of this Part: a residual situational-feature leak has been identified and its fix is staged in code. The fix is serve-neutral — it does not change anything the production model serves today — and it will be realized at the next retrain, when the artifact is rebuilt. We flag it here rather than wait to be asked. Catching leaks is not a one-time event; it is a standing practice, and there is always one more rock to look under.

Part VIII — Architecture and Validation

This is the engine room. The previous Parts explained *what* the model predicts (a log-dampened margin, VI.II) and *how we stopped lying to ourselves* (Part VII). Now we open the box and walk every wire, then prove the whole thing earns its numbers. The pipeline has five stages; the diagram below is the map, and the subsections that follow take it one stage at a time.



VIII.I Two Heads, One Target

The model does not have one brain; it has two, chosen *because they think differently*. The whole point is that they make different kinds of mistakes, and when you average two forecasters who err in unrelated ways, the errors partly cancel and the average is steadier than either alone — the same reason a panel of dissimilar judges scores a competition more reliably than any single judge: their biases do not line up.

The two heads are an ordinary-least-squares linear model and a gradient-boosted tree ensemble (both defined in II.II). Each is given the identical job — predict the dampened margin from VI.II — and each goes about it in an opposite temperament.

The linear head is a straight-line rule you could print on an index card: take each input, multiply it by a fixed weight, add them up, and that sum is the predicted margin. Every input pulls the answer in one direction by a fixed amount, no matter the situation. Its virtues are transparency (you can read exactly how much each factor counts) and data-efficiency (a simple rule needs few examples to pin down, so it stays stable when data is thin). It is fit by weighted OLS — the time-decay weights from VI.III make recent seasons count more — on 15 features plus 2 hand-built interaction terms. One terminology fossil to retire: this head is labeled “GLM” in the code and the diagram, a leftover from the classifier era when it really did predict win/lose probabilities directly. In v9.5 it is literally an `lm()` — ordinary least squares

on the dampened margin, nothing more. When you see "GLM head," read "the straight-line rule."

The tree head is the opposite kind of machine: a committee of hundreds of small if-then flowcharts, built by XGBoost in sequence, each new tree trained to clean up the leftover errors of the ones before it (the boosting mechanism is defined in II.II). It runs shallow trees (depth 4), takes small cautious steps (learning rate 0.05), and works from 17 features. Why bother with this harder-to-read second machine? Because the straight-line rule has a built-in blind spot: it assumes every factor's effect is fixed and additive — one extra point of rating gap is worth the same in week 1 as week 12, for Alabama as for Akron, whether or not a betting line exists. That assumption is *mostly* true, which is why the linear head is good. But football has *interactions* — places where the effect of one thing depends on another. Rating gaps mean less in week 1 when the sample is small and ratings are stale; when a real betting line exists, it tends to swamp the other signals. A flat rule structurally cannot represent "this factor matters more in that situation"; a tree does it for free, because every branch is already "the answer down here depends on the answers above." The trees natively capture the bends and conditional effects the line misses. The cost is that trees are hungrier for data and you cannot read them off a card.

Put the two together and you get the best of both temperaments: the linear head supplies a stable, interpretable backbone, the tree head supplies situational nuance, and because they are wrong in *different* ways, blending them beats either alone. That is why two heads, not one. (Which features each head carries, and why 15 versus 17, is the subject of Part IX.)

VIII.II From Margin to Probability

Each head outputs a *margin* guess on the dampened scale. But the moneyline product needs a *probability* — how sure are we this team wins? The bridge from "expected margin" to "chance of winning" is the normal distribution, applied here exactly as set up in II.II.

The idea in one line: predictions are never exact, so the actual result scatters around the predicted margin in a bell curve, and the win probability is simply the share of that bell curve sitting above zero. The wider the scatter, the more of the bell pokes below zero, and the less sure we are. The function that reads "what fraction of a bell curve centered at M sits above zero" is `pnorm`:

```
P(win) = pnorm(M / σ)
```

The input `M / σ` is "how many standard deviations above zero is our predicted margin?" — and `pnorm` turns that distance into a probability. On the raw-points scale, us-

ing the model's working spread of about 13.5 points (a display dial, not the empirical error SD — real margins scatter wider, ≈15.6 around the market and ≈19.8 around us, reconciled below), the conversions are: +3 → about 59%, +7 → about 70%, +14 → about 85% (these specific values are worked in II.II).

Now the part that trips everyone up, named loudly: there are two different σ's in this system, on two different scales, doing two different jobs.

The first σ is a *per-head residual standard deviation*, used going *in* to turn each head's margin guess into a probability. There are two of them, one per head: σ_glm = 2.038 and σ_xgb = 1.785. Each is measured from its own head's leftover errors during fitting — literally "across all training games, how big is this head's typical miss?" The trees fit a hair tighter, which is why their number is smaller. Both live on the dampened (log-transformed) scale, because that is the scale the heads actually predict on. So each head computes its probability as `pnorm(its prediction / its own σ)`: a head that is usually only a little wrong gets to be more confident off the same prediction.

One honesty caveat, flagged by the engineering review and worth stating plainly: these two σ's are *in-sample fit residuals*. They are measured on the same games the heads were fit to, and a fit has already chased those points, so the residuals come out a touch smaller than they would on fresh data. The practical consequence is that each head, taken alone, is slightly over-confident *before* the blend. Two things rescue calibration downstream: the learned blend (VIII.III), whose weights were fit on honest out-of-fold predictions and which dampens that optimism, and the calibration check in X.V, which confirms the final probabilities land honest. Measuring these σ's out-of-fold is a noted improvement for the next retrain. We do not hide that the inputs are slightly optimistic; we show where the system corrects for it.

The second σ is a *fixed presentation constant*, σ = 13.5, used going *out* to turn the final probability back into points. After the blend produces a final win probability p, we run the bridge backwards — `qnorm` is the inverse of `pnorm`, turning a probability back into a "how many σ's above zero" distance (see II.II) — and multiply by 13.5: `display margin = qnorm(p) × 13.5`. This σ lives on the raw-points scale, is not measured from the model's errors, and is not either head's residual SD. It is a *fixed, frozen display constant*. The original rationale was that 13.5 matched the points-scale scatter the *market's* spreads imply, so our margin and their line would be read on one shared yardstick — the ATS product (X.IV) compares our points margin against Vegas's points margin, and if we expressed our margin with a different σ than the one implicit in reading their spread, every ATS "edge" would be a measurement artifact rather than a real disagreement. A robustness check has since sharpened that story: the market's spreads actually imply a *wider* scatter, closer to 15.6 points, and the model's own margin errors are wider still (about 19.8). So 13.5 is *tighter* than the market-matched value, and we state the consequence plainly rather than bury it —

reading our probability back through a 13.5 dial compresses our displayed margins roughly 13% toward zero versus a 15.6 dial, so our against-the-spread edges, if anything, *understate* the disagreement rather than inflate it. The constant is conservative, not flattering. It stays frozen at 13.5 for the season to keep the locked pick log internally consistent, and re-deriving the display σ from the market's implied scatter is a noted improvement for the next retrain.

The trap in one line: 2.038 and 1.785 are "how wrong is each head, on the dampened scale," used going in to make probabilities; 13.5 is "the standard points-scale yardstick," used coming out to make a comparable margin. They are all called sigma only because they are all standard deviations of *something*. Keep straight *which something*, and the confusion evaporates.

VIII.III Our Blend

Now we have two probabilities, one per head, and we need a single number. We do not average them naively, and we do not use the old "75/25" rule some early notes mention. We combine them in log-odds space — the logit scale defined in II.I — with weights the data chose for us.

Why log-odds? Probabilities cannot be safely added or weighted directly: they bunch up against the hard walls at 0 and 1, so averaging two near the edges misbehaves (you cannot push "90% plus a bit more confidence" past 100%). Log-odds have no walls — they run from minus-infinity to plus-infinity, centered at zero for a coin flip — so you can add, weight, and combine them like ordinary numbers and then convert back at the end, always landing in a sensible 0-to-1 probability. Log-odds is also the natural scale for stacking independent opinions, because adding log-odds corresponds to multiplying odds.

The blend is a tiny learned logistic stack: "logistic" because it works in log-odds space, "learned" because its weights were fit from data rather than hand-picked, and "stack" because it sits on top of the two head outputs. It was fit on out-of-fold predictions (the leak-proof kind, defined in VIII.IV), and the weights it found are:

```
logit(p) = −0.026 + 0.564·l_glm + 0.453·l_xgb
```

where `l_glm` and `l_xgb` are the two heads' probabilities expressed as log-odds. Reading each piece: −0.026 is the intercept, a baseline nudge so close to zero that the blend carries no built-in lean toward home or away. The 0.564 and 0.453 are the data's verdict on how much to trust each head — near-equal partners with a slight lean to the linear head, exactly what you would expect at this data size, where the stable straight-line rule has a small edge over the data-hungrier trees. They do not

sum to 1 and do not need to; log-odds weights are not a percentage split. Their *ratio*, about 1.24 to 1, is the meaningful thing: lean a bit toward the linear head.

A worked example so the formula is not abstract. Suppose both heads independently call the home team a 70% favorite. The log-odds of 70% is `ln(0.70 / 0.30) = ln(2.333) ≈ 0.847`, so both `l_glm` and `l_xgb` are ≈ 0.847. Plug in:

```
logit(p) = −0.026 + 0.564·(0.847) + 0.453·(0.847)
         = −0.026 + (1.017)·0.847
         = −0.026 + 0.861
         ≈ 0.835
```

Convert that log-odds back to a probability: `p = 1 / (1 + e^(−0.835)) ≈ 0.697`, which rounds to ≈ 70%. When both heads agree on 70%, the blend returns ~70%, exactly as it should — it passes a consensus through cleanly, neither inventing confidence nor throwing it away. Its real work shows up when the heads *disagree*: there it produces a weighted-in-log-odds compromise that leans slightly toward the head it learned to trust more. The near-equal weights mean this is a genuine partnership, not one head riding the other.

A note on what is *not* in this blend, because an earlier classifier pipeline needed two extra calibration layers (Platt scaling and isotonic regression — after-the-fact corrections that re-map raw probabilities to honest ones) to counteract the way log-loss-trained classifiers compress their probabilities toward 0.5. The margin architecture dissolves that problem at the root: a squared-error margin regression has no incentive to shade its margins toward zero, and the `pnorm` conversion inherits that honesty, so the probabilities come out roughly calibrated without a rescue layer. The pre-registration in VIII.IV confirmed this empirically — among six candidate variants, the plain "learned logit" blend above beat both calibrated alternatives. The calibration machinery still ships inside the artifact for diagnostics, but production does not apply it. There is one resolved bug worth recording for the discipline it teaches: serving code once applied the Platt layer *unconditionally* whenever a Platt model was present in the artifact, briefly serving a never-pre-registered, never-measured variant. It was fixed by gating that layer on the artifact's recorded chosen-variant flag, so serving now matches the exact configuration that was validated — and the lesson logged for the next retrain is to persist each game's out-of-fold predictions inside the artifact, so "what would serving have done?" is answerable straight from disk.

VIII.IV Validation

A model that looks brilliant on the data it trained on has told you nothing — of course it fits the games it already saw. The only test that means anything is performance on games it has *never seen*, and getting that test right, especially after Part VII

showed how easily we fool ourselves, is the discipline that earns the v9.5 numbers their trust. Three ideas, all introduced in II.II, do the work here; we apply them.

Walk-forward folds train on the past and test strictly on the future, never the reverse. We slide the training window forward through the seasons and always test on a year that comes after it:

Fold	Train on	Test on (never seen in training)
1	2016–2020	2021
2	2016–2021	2022
3	2016–2022	2023
4	2016–2023	2024
Final one-shot	2016–2024	2025

Each fold mimics live use exactly: you can only ever know the past. Training rows inside each window still carry the `exp(−0.2 · age)` decay weights from VI.III, so recent football counts more even within a fold. This protocol guards the *across-season* boundary. It does not, by itself, guard the *within-season* boundary — that was the exact blind spot of Part VII, fixed separately by point-in-time features. Both guards are needed; neither substitutes for the other.

Out-of-fold (OOF) predictions are how we tune the downstream pieces — the blend and any calibrators — without cheating: every prediction used for tuning is made by a version of the model that was *not* trained on that game. If a model graded itself on games it studied, the grade would be a memorization score, not a forecasting score.

Pre-registration is the anti-self-deception device, and it is the most important one. When you choose *after* peeking at the test result — "let me try one more variant and re-check 2025, oh that's better, keep it" — you are quietly fitting the test set, and your final "test score" is no longer a test but a rehearsal you have been polishing. Pre-registration breaks the loop by forcing the decision rule to be written down and locked *before* the sealed test is touched. You declare in advance which metric on which fold decides, then you look, and whatever it says, you are bound to it.

Put together, the contamination-proof protocol ran in this order:

1. Fit the blend (and the calibrators kept for diagnostics) on folds 1–3 OOF only — honest, never-trained-on predictions.
2. Pre-register the production variant by Brier on fold 4 (2024), choosing among six candidates {GLM only, XGB only, fixed 75/25, learned logit, learned+Platt,

learned+Platt+isotonic} *before ever touching 2025*. The winner was the plain learned logit.

3. Refit the calibrators on folds 1–4 now that the variant is locked.
4. Score the pre-registered variant exactly once on 2025.

That single, untouched evaluation is the headline: 73.7% accuracy, Brier 0.1712, AUC 0.816, on 763 games. The folds themselves (2021–2024, on the 2,928 games that carried a betting line) tell the complementary story used throughout the betting Parts: the model hit 74.0% with a Brier of 0.1714, while the spread-implied market favorite hit 72.4% with a worse Brier of 0.1783. So on the folds we edge the market; on the spent 2025 one-shot the market edges us back (74.2% to our 73.7%). Net, across both: market-equal. (The full reading of these numbers, and why market-equality is the *enabling* result rather than the disappointing one, is Part X.)

The one-shot is now spent, and that is a feature, not a regret. A sealed holdout is a chemical reagent: it works once. The first time you evaluate on 2025 you learn something about 2025, and any further decision influenced by that knowledge silently converts the holdout into a validation set you are now fitting. So the moment we scored it, 2025 was retired; every future modeling decision runs only on folds 1–4, and the next genuinely clean test is the live 2026 season, which cannot leak because it has not happened. After validation closed and the variant locked, the two heads were refit one final time on all of 2016–2025 to produce the shipping artifact, the design was frozen, and in-season retraining was disabled — non-negotiable for the brand, because every locked 2026 pick must trace to one fixed, auditable model. A model that could quietly change mid-season would make its own track record unfalsifiable, and an unfalsifiable track record is worth nothing.

VIII.V Worked Example: One Game, Start to Finish

Everything above is easier to hold once you have pushed a single game through the whole pipeline by hand. Here is one, end to end. The input numbers are *illustrative* — chosen round for clarity, not lifted from a real game — but every operation on them is the real machinery from the diagram. Picture a home team the model genuinely likes against a visitor it does not, with a betting line already posted.

Stage 1 — the two heads each predict a dampened margin. Each head reads its features and outputs a number on the log-dampened scale of VI.II, not in points. Suppose the linear head lands at 0.62 and the tree head at 0.68 — close but not identical, because they reason differently. (To sanity-check the scale: a dampened-scale value near 0.7 corresponds, after everything below, to a mid-single-digit favorite. The heads are not yet speaking in points.)

Stage 2 — each head converts its margin into a probability. Using its own residual SD from VIII.II:

```
GLM head:  pnorm(0.62 / 2.038) = pnorm(0.304) ≈ 0.620   → about 62%
XGB head:  pnorm(0.68 / 1.785) = pnorm(0.381) ≈ 0.648   → about 65%
```

The tree head ends up a touch more confident off a similar prediction, because its typical miss (1.785) is smaller than the linear head's (2.038) — the smaller σ buys more confidence per unit of predicted margin.

Stage 3 — blend the two in log-odds space. Convert each probability to log-odds, then run the learned stack from VIII.III:

```
l_glm = ln(0.620 / 0.380) = ln(1.632) ≈ 0.490
l_xgb = ln(0.648 / 0.352) = ln(1.841) ≈ 0.610

logit(p) = −0.026 + 0.564·(0.490) + 0.453·(0.610)
         = −0.026 + 0.276 + 0.276
         ≈ 0.526
```

Convert back to a probability: `p = 1 / (1 + e^(−0.526)) ≈ 0.629`. The blended win probability is about **63%**. Note it sits between the two heads' inputs and leans, by the weights, slightly toward the linear head's view — exactly the genuine-partnership behavior described in VIII.III.

Stage 4 — read the two products off that one number. The moneyline read is the probability itself: the home team is about a 63% favorite. Whether that clears a confidence tier and becomes a logged pick is the staking question of Part IV; 63% is a credible but not a premium-tier number.

The ATS read converts the probability back to points using the fixed presentation σ = 13.5 from VIII.II:

```
display margin = qnorm(0.629) × 13.5 = 0.329 × 13.5 ≈ 4.4 points
```

So the model's market-blind line is "home team by about 4.4." If Vegas had posted this game at home −1.5, the edge is 4.4 − 1.5 ≈ 3 points of disagreement in the home team's favor — enough to flag under the week-1-to-7 threshold of Part IV, and the ATS bet would be the home team to cover. If Vegas had instead posted home −7, the edge is 4.4 − 7 ≈ −2.6, meaning the market likes the home team *more* than we do, and there is no bet (or, on a large enough gap, a lean to the underdog to cover).

One game, one dampened-margin signal from each head, blended once, then read two ways: a 63% moneyline favorite and a ~4.4-point market-blind line. That is the

entire machine, start to finish. Every number it ever publishes is some version of this trace — and every number it publishes carries the honesty, the leak history, and the validation discipline of the Parts above it.

Part IX — Choosing the Signals

A model is only ever as good as the columns you feed it. The architecture of Part VIII — two heads, a learned blend, a margin turned into a probability — is the machine; this Part is about the fuel. What are the inputs, how did we decide which ones earn a seat, and once the selection is done as honestly as we know how, what is the ceiling on what *any* selection can buy us?

That last question is the important one, so it gets its own subsection. The short version: we screened forty-five candidate inputs to a disciplined handful, ran seventeen experiments to do better and kept exactly one, and the pattern is not "we haven't found the magic feature yet." It is "the magic feature does not exist, because in an efficient market it cannot." The next real gain is not a cleverer curve through the same data — it is a new piece of data.

IX.I Feature Selection Process

We started with forty-five candidate features — forty-five possible input columns, each computed for both teams and differenced (home minus away) so the model sees one signed gap per game. The job: keep the columns that carry real, *new* information about who wins, reject the ones that only look useful by luck or that merely echo a column already in the room.

Two tools do the work, both defined in II.II: an ANOVA F-test, which asks whether adding a feature shrinks the model's leftover error by more than chance alone would, and the Bonferroni correction, which tightens the bar for the fact that we run forty-five such tests at once. The arithmetic is the spine of the screen: at the usual $p < 0.05$, each test lets through worthless features at a 5% rate, so across forty-five tests the expected count of pure-noise sneak-throughs is $45 \times 0.05 \approx 2.25$ — about two fakes for free, every time. Bonferroni divides the bar by the number of tests to hold the *total* false-positive budget at 5%:

```
keep bar  =  0.05 / 45  ≈  0.0011
```

A candidate now has to clear $p < 0.0011$ — roughly fifty times stricter than 0.05 — to earn a slot. The harshness follows the document's spine: we would rather wrongly drop a marginal real feature (a sliver of signal we can live without) than keep a fake

one (which quietly teaches the model to trust noise, the exact mechanism that inflated the pre-v9 numbers, Part VII).

One trap nearly cost us our best feature. The *sequential* (Type I) flavor of ANOVA credits explained variance in the order features enter the model. Run that way, the closing spread — the strongest market signal on earth, partial correlation ~0.50 — scored p = 0.97, screaming "useless." The reason was order, not worthlessness: the power ratings entered first, and because the spread is built largely from those same ratings (multicollinearity, II.II), by the time it "walked in" nearly all the variance it could explain had already been claimed. The fix is to judge each feature on the variance it explains *given all the others present*. Read that way, the spread is the most valuable column we have.

The survivors. The linear (GLM) head carries fifteen features plus two interaction terms:

> sp_diff, sp_weighted_diff, ppg_diff, papg_diff, recruit_diff, coach_wp_diff, elo_diff, opp_adj_off_diff, opp_adj_def_diff, returning_ppa_diff, sp_early, letdown_diff, lookahead_diff, qb_advantage_diff, spread_diff
>
> interactions: home_field : conf_matchup and sp_diff : conf_matchup

The XGBoost head carries those same fifteen, plus two more — has_spread and revenge_diff — for seventeen. Two facts here correct older versions of our own documentation. First, spread_diff lives in *both* heads in v9.5. The old "spread is tree-only, to avoid multicollinearity" rule belonged to the v8 classifier era; under the margin target the ANOVA screen kept the spread in the linear head too, and the fold metric agreed the signal was worth the collinearity cost. Second, the situational trio (letdown / lookahead / revenge) earns its slot *conditionally* — it is included only when it improves the folds-1–4 out-of-fold Brier by more than 0.0005 in that run's ablation, a bar it cleared in the shipped artifact.

A tight tour of what each column carries, grouped by kind:

- *Market.* spread_diff is the closing point spread itself, our single strongest column ($r \approx +0.66$ with the outcome — see X.II); where no line exists it is set to 0, and the model was *trained* with that same zero-placeholder convention so it degrades gracefully rather than breaking. has_spread is a 1/0 "is a real line present?" flag — an interaction-shaped signal that lets the trees switch behavior between lined and unlined games, the conditional a tree handles better than a linear coefficient, which is why it is tree-only.
- *Ratings.* sp_diff and sp_weighted_diff are gaps in SP+, Connelly's opponent-adjusted efficiency rating (the weighted variant blends the prior two seasons),

always on *prior*-season values to stay point-in-time clean (Part VII). elo_diff is the gap in our own self-computed Elo (100% coverage, since we compute it). opp_adj_off_diff and opp_adj_def_diff are our iterative opponent-adjustments of offense and defense, refreshed weekly so they use only games already played.

- *Form.* ppg_diff and papg_diff are the gaps in points scored and allowed per game, built as the progressive in-season blend from Part VII (early-season lean on last year, ramping toward this year as the sample earns trust).
- *Roster.* recruit_diff is the gap in a four-year rolling 247Sports Composite score; returning_ppa_diff the gap in returning production (in predicted points added); qb_advantage_diff the quarterback situation, built from the prior week so it is knowable before kickoff.
- *Situational.* sp_early down-weights the ratings in weeks 1–4; letdown_diff flags a team off a big win over a ranked opponent (hangover risk); lookahead_diff flags a marquee game *next* week (the looking-past trap); revenge_diff flags a rematch the team lost last season (tree-only).
- *Interactions.* home_field : conf_matchup lets home advantage bend by matchup (a cross-country road trip beats an in-conference bus ride); sp_diff : conf_matchup lets a talent gap mean something different inside a conference than across the P4/G5 divide.

Fifteen-to-seventeen columns, every one a measure of quality, form, talent, situation, or the market's own number — each screened past a bar fifty times stricter than the textbook default.

IX.II The Graveyard

A model defined only by what it *is* invites endless second-guessing; a model that can show you the headstones of everything it *isn't*, and the fair, pre-committed rule that killed each, has answered most questions before they are asked. Seventeen experiments ran. One survived.

Every candidate change was scored on folds-1–4 out-of-fold Brier (II.II, measured on games the model never trained on), with 2025 kept physically untouchable — the harness filters `season <= 2024`, so the sealed year can't be peeked at even by accident. The adoption bar was set *before* the experiments ran: improve that Brier by at least 0.0010 to ship. The reason for a hard threshold rather than "adopt whatever helps" is the multiple-comparisons trap (II.II): with seventeen attempts the best-looking result is partly just the luckiest, and the harness's noise floor is about ±0.0005, so a −0.0005 "improvement" is indistinguishable from a coin landing your way. The 0.0010 bar sits above that floor, so only non-chance effects get through. (Two baselines anchor the deltas, both on the harness's all-games pool — a different, slightly

higher pool than the lined-games headline of X.II: 0.17254 before the margin architecture, 0.17085 after it.)

The ledger, top to bottom, deltas in out-of-fold Brier against that pre-registered −0.0010 bar:

Experiment	Δ Brier	Verdict
Margin architecture (log-dampened target)	**−0.0015**	KEEP — shipped as v9.5
Ridge + tuned-hyperparameter combo	−0.0008	best loser, sub-bar
Symmetry augmentation (mirrored rows)	−0.0007	redundant once margin head exists
Hyperparameter sweep, 70 configs	−0.0006	v9.0 params already near-optimal
Ridge opponent-adjustment (λ shrinkage)	−0.0006	sub-bar
Training-window × decay grid (9 cells)	−0.0006	confirms 2016 / λ=0.2 stands
Elo parameter grid (12 configs)	−0.0004	Elo insensitive to its knobs
ml_implied_diff (de-vigged moneyline as feature)	−0.0001	spread already carries it
GLM-share blend grid	−0.0001	learned blend already optimal
ret_x_sp interaction	−0.0001	no new signal
Talent composite (recency-weighted recruiting)	−0.0001	saturated by recruit_diff
PPA efficiency ratings (ridge-validated)	±0.0000	saturated by Elo + form + spread
Elastic-net GLM (α=0.5, full pool, λ.1se)	+0.0009 (worse)	over-shrinks; see below
Preseason SP+ backfill	—	infeasible (data exists only for 2026)
Totals (O/U) product	—	dead as a product; see below
(v8-era) spread_diff in the classifier GLM	worsened	died on classifier-era multicollinearity

Read that Δ column top to bottom and one humbling pattern jumps out, the lesson of the whole campaign: *every architectural cleverness moved the needle less than just predicting the right quantity in the first place.* The margin target alone bought −0.0015; everything else clustered between −0.0008 and ±0.0000, short of the bar. What you predict mattered more than how you predict it.

The deaths worth a sentence apiece: the margin architecture is the one keep, the reason Part VIII predicts a log-dampened margin instead of a binary win/loss. The ridge variants were the *best losers* (−0.0008, −0.0006) — exactly the trap, since "close" tempts you to rationalize a sub-threshold change as "basically the bar," which is how forking-path rot begins; ANOVA-selected OLS stands because seventeen attempts to beat it all failed under a fair, pre-committed rule. ml_implied_diff (the de-vigged moneyline) and the PPA ratings died as *redundant* — the spread and the Elo/form features already carry their information. The training-window grid confirms the 2016-start, λ=0.2 decay of Part VI; the Elo grid shows our Elo is insensitive to its knobs here.

Two killed ideas need a touch more, because they are the ones people ask about most.

Elastic net. The penalized regressions (ridge, lasso, elastic net, defined in II.II) cure two specific diseases: too many features for your sample size, and multicollinearity destabilizing coefficients. We tested elastic net properly — α=0.5 over the full pool, λ chosen by in-fold cross-validation with the conservative "1se" rule, on the same folds as everything else — and it came in 0.0009 Brier *worse* than the plain ANOVA-selected OLS. The penalty was too eager, shrinking to just five-to-eleven surviving features and throwing away real signal the Bonferroni screen had correctly kept. The cure made the patient sicker because the patient wasn't sick: penalization earns its keep when features rival or exceed rows (genomics: 20,000 genes on 200 patients), and we are the opposite — roughly fifty candidates against seven thousand games, one feature per 140 rows. At that ratio the OLS coefficients are already stable, the tax just confiscates good signal for nothing, and the pool was *already* selected by a domain-aware, leak-audited, Bonferroni-screened pipeline. Asking lasso to re-select is asking a tax form to second-guess a scout.

Totals. We did not reject over/unders on a hunch — we built a blind totals model (combined-scoring features: PPG sums, PPA, pace proxies, weather) and let it lose. Its typical miss, in RMSE (the average size of the miss in points, II.II), was 16.32 against the market's 15.87 — the market's miss is *smaller* than ours, so we are flatly worse at totals than our feature set allows. And the edge-bucket pattern fails the lie-detector test of II.II: our bigger disagreements did *not* hit better than our small ones (non-monotone), where the sides product behaves exactly as a real edge should (59.5% at 3–5 points of disagreement, 66.3% at 8+). When your biggest disagreements

with a sharp counterparty are not your best bets, the disagreements are about *you*, not the line. The football reason: totals turn on pace, kickoff weather, and skill-position availability — the game-time information our point-in-time features deliberately do not carry and the market *does*. Sides are the edge; totals are a donation. (A v10 with real game-time inputs earns a fresh re-test, not one minute sooner.)

Two method families never reached the ledger, ruled out on the same principle that built it. A genetic algorithm evolves thousands of model configurations and keeps the fittest each generation — a maximally aggressive multiple-comparisons machine with no Bonferroni analog; on our row count and ±0.0005 noise floor it would "discover" fold-fitting combinations all day, each a forking-path artifact. It does on purpose the exact thing our methodology exists to prevent. Neural networks need either enormous n or strong inherent structure (an image's pixel grid, a sentence's word sequence); tabular sports data with no such structure is the textbook setting where gradient-boosted trees match or beat them — and we already have the trees in the XGBoost head.

IX.III The Information Ceiling

Here the ledger stops being a list of failures and becomes a *map*. Look at the magnitudes, not the verdicts: a seventy-config hyperparameter sweep moved Brier −0.0006; the margin target moved it −0.0015, once; everything else clustered near zero. The biggest lever in seventeen swings was −0.0015, the second-biggest a −0.0008 that didn't clear the bar.

That distribution is the most important strategic fact in this Part: we are bumping against the information ceiling of this feature set. The bottleneck is not the cleverness of the curve we draw through the data — it is the information content of the data itself. Squeezing the function-approximator harder (a fancier blend, a deeper tree, a genetic search, a neural net) cannot extract information the columns never contained. When the biggest improvement in seventeen disciplined attempts is fifteen ten-thousandths of a Brier point, the honest inference is not "keep searching for the −0.01 idea"; it is "the −0.01 idea is not in these columns."

There is a deeper reason to believe that, and it goes beyond our ledger to the structure of the thing we bet into. Suppose a genuinely golden signal *did* exist — some feature, computable from public data before kickoff, that reliably beat the closing line. It would not stay golden. The betting market is the aggregated, real-money forecast of everyone willing to risk capital on these games, syndicates included (the full argument is in Part III, the market's strength quantified in X.II). The instant a reliable edge becomes visible in public data, money floods toward it, the line moves to absorb it, and the edge is arbitraged toward zero. A signal both genuinely predictive *and* sitting unused in public data is a contradiction in an efficient market: if it existed

and worked, everyone would already have it. Its very availability is evidence it has already been competed away.

So the ledger and the market tell the same story from two directions — the ledger empirically ("seventeen ways to wring more out of these columns and the well is nearly dry"), the market theoretically ("a golden public signal can't persist"). The search for a cleverer curve through the same data is the wrong search. (One honest boundary: the ledger bounds what the strategies *we tried* could find, not the theoretical best of the feature set — read it as "we have not found more, with strong structural reason to doubt more is there," not as proof of impossibility.)

What, then, is the right search? A new *input* — a column the market hasn't fully digested or that our point-in-time set is currently blind to: quarterback status confirmed at lock time, injury and inactives reports, kickoff weather, late line movement. These are not refinements of the curve; they are new information, the only thing that can move a forecast already sitting at the market's shoulder. This is why the project hunts data, not architectures — and why Part XI's road to v10 is a list of inputs to acquire, not models to try. The cleverness is not where the next gain lives. The data is.

Part X — Reading the Numbers

Every model eventually has to put a number on the table and say "this is how good it is." This Part is that number — or rather the several it takes to tell the truth, because no single one is safe alone. We report three headline metrics, benchmark them against the sharpest forecaster available, lay the betting layer's numbers on one honest table, and close with the fine print a careful reader is owed. The discipline throughout: the only numbers cited are ones we can point to a computation for; where one wasn't computed we say so. The citation rule from Part I holds — the production v9.5 sealed one-shot is 73.7% / Brier 0.1712 / AUC 0.816, and nothing published before June 10, 2026 (85.1%, 83.9%, 0.1200) is anything but the warning of Part VII.

X.I Accuracy, AUC, Brier

Three numbers carry the model, all from a single sealed evaluation on the 763 played games of 2025: accuracy 73.7%, AUC 0.816, Brier 0.1712. Accuracy asks "how often is the favorite right?"; AUC asks "can the model rank games correctly, most-confident to least?" (a pure ordering, never picking a winner); Brier asks "are the

probabilities themselves honest?" A model can be strong on one and weak on another, which is why we report the whole trio (all three defined in II.II).

Accuracy, and the crowded neighborhood at ~74%. For each game, call home if the model's home-win probability is above 50%, else away, then count matches. On 2025 the call was right 73.7%. That means nothing without a baseline, so here is the neighborhood, all on comparable games: always pick home, 59.5% (home advantage alone, the floor every handicapper must clear); raw SP+ favorite, ~74%; the market favorite, 74.2% on the same 763 games; our model, 73.7%.

Everyone competent clusters around 74%, within a single point of each other, and that is the *expected* result, not a disappointment. Most college games aren't close calls — when a powerhouse plays a directional school, SP+, the market, and we all call it correctly — so the convergence near 74% is the signature of the genuine toss-up tail, roughly a quarter of games where no information moves you off 50/50 and being "right" is just landing on the lucky side of variance. The ceiling on pickable games is low because football is noisy, not because the models are weak. Being inside the ~74% pack is the mark of a correctly built model; a model claiming 84% isn't smarter than the market, it is leaking the answer (Part VII).

AUC — ranking, not calling. AUC is 0.816, defined pairwise (II.II): over every pairing of a home win with a home loss, the fraction where the model gave the won game the higher probability. It rewards ordering and ignores where you draw the 50% line, which is why 0.816 (ranking) and 73.7% (calling) coexist comfortably: the ordering is strong, and the gap down to accuracy is almost entirely well-ranked-but-upset games — the 60% favorites the model correctly flagged as *only* 60% that landed on the 40 side. A 60% favorite is supposed to lose four times in ten; if it never did, the 60% was a lie (II.I). AUC credits honest uncertainty; accuracy can't see it.

Brier — the honesty metric, and its ladder. Brier is 0.1712, the mean of `(p − outcome)²` across games (outcome 1 for a home win, 0 for a loss). It scores the number we published, which is why it is the metric every experiment in IX.II was decided on, and it punishes overconfidence quadratically — say 90% and win costs `0.01`, say 60% and lose costs `0.36`, far more than the `0.16` of saying 40% and being wrong the gentle way. The reference ladder, worst to best:

```
0.25 (coin flip)  →  0.241 (know only the home rate)  →  0.1783 (Vegas)  →  ~0.171 (us)
```

The 0.241 rung is one line of arithmetic: a constant "59.5% home" forecast scores $0.595 \times 0.405 \approx 0.241$, the score of knowing the home rate and nothing else. The fall from 0.241 to ~0.171 is the model's *skill*, and precision matters here. Brier decomposes (Murphy) as `Uncertainty − Resolution + Reliability`: uncertainty is the 0.241 weather; resolution, *subtracted*, is sorting games into buckets whose realized rates

genuinely differ from the base rate; reliability, *added back*, is the calibration penalty. So the ~0.07 we bought back is `resolution − reliability`, skill net of whatever calibration tax we pay — if calibration is flawless the whole 0.07 is resolution, if we're a touch overconfident our true resolution is larger with a penalty eating part of it. We do not publish the split, so the honest phrasing is: the improvement is skill, and how much is each is taken up in X.V.

Why three metrics and not one: accuracy can be gamed by memorizing "favorites win" and inflated by leakage; AUC ignores whether probabilities are honest, only their order; Brier catches dishonest probabilities but blurs ranking and calling together. No single number is safe alone. When all three move the same way you have found something real; when one moves and the others don't, you have usually found a way to fool yourself.

X.II The Market Benchmark

A metric only means something against the strongest available yardstick, and here that is the closing line — after a week of sharp money pushes it, the single strongest publicly available estimate of game outcomes on earth, the aggregated real-money forecast of everyone willing to bet, corrected the instant it's wrong (the full argument is in Part III). We convert a spread to a probability with the same `pnorm(spread/13.5)` machinery as our own margin (3 points ≈ 59%, 7 ≈ 70%, 14 ≈ 85%, from Part VIII).

This is why the closing spread is our single strongest feature — spread_diff correlates with the outcome at r ≈ +0.66 against ~+0.47 for the next-best (SP+ differential). In this domain +0.66 is enormous: the most useful thing the model knows about a game is what the market already thinks, and everything else is refinement on that anchor. We do not out-think Vegas from scratch; we stand on its shoulders and hunt the small residual it hasn't fully priced.

Now the honest scoreboard, and the one place where the old draft of this document had its numbers crossed. There are *two* comparisons, on two different game sets, and they point in opposite directions:

- On the walk-forward folds (2021–2024, 2,928 lined games), we edge the spread-implied market: accuracy 74.0% vs 72.4%, Brier 0.1714 vs 0.1783. On these games, on both metrics, we are ahead.
- On the sealed 2025 one-shot, the raw market favorite nosed ahead of us on accuracy: 74.2% vs 73.7%. (We deliberately do *not* quote a 2025 Vegas Brier — that specific number was never computed in the sealed evaluation, and the rule of this document is that no figure appears we can't point to.)

An earlier draft mistakenly reported the market at 74.2% on the *folds*; that 74.2% is the one-shot number, and on the folds the spread-implied market is 72.4%. The corrected pair is the whole point: we edge the market on the folds, the market edges us on the one-shot, net *market-equal* — essentially tied, with a defensible residual edge in the disagreement tail and no edge where we and the line agree. That is exactly where an honest, market-aware model should land, and why the old "we beat Vegas's Brier" claim should have triggered the leak hunt a season early (Part VII): if your strongest feature *is* the market's own forecast, you cannot routinely crush its Brier — at best you nudge past it where you've found a real residual. The value lives in that small tail, and even there it is modest. That is not a hedge; it is the only claim a calibrated, market-aware model has any business making.

X.III The Betting Layer's Near-Miss

The model was audited thoroughly before we trusted it; the betting layer — the machine that turns probabilities into "bet this, skip that" — had never been audited at all, and in June 2026 it led us toward the same kind of mistake the leak had taught us to fear, in a new form. This is the second lesson, and it matters because the dangerous failures don't crash — they produce a plausible, wrong number that trips no alarm because the code "ran fine." Five findings fell out of giving the betting layer the model's treatment, each a general trap, not a college-football quirk.

B1 — a market-equal model cannot have a big betting edge. The first backtest of the EV-gated tiers came back at +11% to +23% ROI, and we got nervous rather than celebrating. A betting edge is the gap between your probability and the market's (II.I); no gap, no edge, and that gap caps your sustainable profit. But the audited fact is that we and the market are essentially tied (74.0% vs 72.4% on folds, 73.7% vs 74.2% on the one-shot — net, market-equal). If your probabilities are on average no sharper than the market's, your average edge is ~zero. A +20% return from a market-equal model is not a triumph; it is a contradiction — it is telling you about a flaw in your test, not a flaw in the market.

B2 — hit rate is a safety stat; EV is the edge stat. Tier A, the bucket that hits 88.4%, *loses money bet flat*, because a hit rate means nothing without the price beside it. Tier A picks are heavy favorites at a median price near −850, which needs ~89.5% just to break even (II.I). The 88.4% is real and it is not an edge — it is a survival stat, perfect for a parlay leg, wrong as a flat single. The edge lives where the model's probability beats the price (the +EV picks), and those concentrate in moderate −200 to −110 favorites, where there's daylight between model and market and room to get paid. So the policy isn't "bet the highest tier"; it is "bet only the picks, in any tier, where the model beats the price." That filter — the EV gate — is the entire strategy.

B3 — decompose every average, then drop the best bets. A blend can hide that almost all the profit came from one corner. Breaking the +20% apart, that's what we found: roughly sixteen-to-twenty-one underdog and pick'em bets returning +71% to +83%, while the robust large-sample bucket of moderate favorites returned a believable +3% to +8%. Twenty-one bets at +71% is not a strategy; it's a story about which way the dice fell. The fix is to report the large-*n* cell, then run leave-N-out: the honest 2025 policy survived it — drop the best bet and +8.46% becomes +7.75%, drop the best three and it's +6.66%, still clearly positive. That survival is the gap between signal and noise.

B4 — bad prices manufacture fake edge, and an EV filter hunts for them. The EV gate's job is to find prices that look too generous relative to the model — which is also, word for word, the description of a *corrupted* price, so the filter is magnetically attracted to bad data. The historical odds file had roughly 53 games (about 2.3%) with moneylines swapped relative to the spread — a 5.5-point favorite priced like a dog at +210 — and the filter was sucking them up as its juiciest plays. The fix is a price-sanity guard: compare moneyline-implied to spread-implied probability and throw the bet out if they disagree by more than 0.10. It protects the backtest. The open risk: the *live* system reads the same feed and the live guard is not yet shipped — and because quarter-Kelly sizes the stake *with* the apparent edge, a single swapped live price becomes the largest-staked, most confident recommendation on the board. The live guard must ship before any live Kelly stake is displayed; that order of operations is non-negotiable.

B5 — pre-register the policy, freeze it, spend the holdout once. You can tune a betting strategy forever against your test data until it scores beautifully on that data's noise — the garden of forking paths (II.II), one layer up from the model. So we did it properly: developed the policy on 2021–2024 only (in-sample +13.1%, refused as a forecast), committed the exact rule to git before touching 2025, then scored it once on the sealed season and retired it. The result is trustworthy not because it is high but because it could not have been tuned after the fact. It landed at +8.46% (the in-sample +13.1% regressing downward as honest tests do), survived leave-3-out, concentrated in the moderate-favorite bucket, passed a winner's-curse check (selected-pool average probability 0.810 against a realized 0.818, no overconfidence in the bets we actually made), and showed weakly positive closing-line value. The full table follows in X.IV; the long-form arithmetic lives in Part IV. The verdict is deliberately undramatic: tiers bet flat are chalk; the EV gate is a real but modest edge — promising, not proven — and the only ungameable judge is the live 2026 log.

X.IV What Each Betting Number Measures

By now the betting edge has appeared as "+8.46%," "+8.5%," "+13.1%," and a per-season "fade," and a careful reader should ask whether those are four findings or one finding counted four ways. They are not the same number, not measured on the same games, and do not deserve the same trust. This is the reconciliation table.

The trap to disarm first: the frozen policy returned +8.46% on 2025 and the keep-vs-reject diagnostic returned +8.5% on 2025 — they look like the same number rounded differently, but they run on different sets of games. The policy (n=165) is the actual product: only Tier A or B picks (confidence ≥ 0.65), only EV > 0, only price-sane, only no missing features. The discrimination test (n=268 kept against n=380 rejected) is a *diagnostic*: it splits the whole pool of 2025 model-vs-market comparisons into kept versus rejected, with a *wider* kept pool because it drops the tier and degraded-feature filters. Its job is the *gap* between kept and rejected, not its own ROI. So +8.46% (policy, n=165) and +8.5% (kept, n=268) are two measurements of the same edge through two sieves — mildly reassuring that they land close, but not two independent confirmations. Citing both as such would be double-counting.

Figure	Value	n	Game universe	In / Out	How hard to lean
Frozen policy ROI (EV-gated A+B singles)	+8.46% (LB95 +0.44%)	165	Sealed 2025; A/B + EV>0	Out, examined	the headline
Discrimination — kept	+8.5%	268	Sealed 2025; all +EV	Out, examined	not a 2nd win
Discrimination — rejected	−3.2%	380	Sealed 2025; EV ≤ 0	Out, examined	essential
In-sample policy ROI	+13.1%	dev pool	2021–24 dev, EV-gated A+B	In-sample	discount heavily
In-sample discrimination	+16.0% (kept) vs −9.5% (rejected)	521 / 603	2021–24 dev, full +EV pool	In-sample	shape only
Per-season "fade"	+15.6% → +17.3% → +8.5%	kept pool / year	2023–24 in, 2025 out	Mixed	not a trend
Flat (non-EV) tiers	A +3.91%, B +3.91%, C −2.00%	A/B/C 2025	Sealed 2025, flat, no gate	Out, examined	counter-example

Reading the essential rows: the most important single fact in the betting case is the discrimination row — the picks the gate *rejected* (EV ≤ 0) *lost money* (−3.2%) while the kept picks made it (+8.5%), an 11.7-point gap proving the gate sorts mispriced lines from fair ones rather than riding chalk. The flat-tiers row is the counter-example: betting the tiers blindly is chalk (only flat Tier A clears zero, barely), and the EV gate roughly doubles the edge. The in-sample rows exist only to show the regression — +13.1% shrinking to +8.46%, the 25.5-point in-sample gap shrinking to 11.7 out-of-sample — which is what honest tests do; quote them for the shrinkage, never as a forward estimate.

That leaves the "fade," which earlier drafts oversold. The per-season kept-pool ROI ran +15.6% (2023) → +17.3% (2024) → +8.5% (2025), and calling that a "decline that shrinks but never to zero" claims far more than three data points can carry. It is not a trend — a trend needs years it hasn't seen — and it is not even monotone: it goes *up* from 2023 to 2024 before dropping, where a clean "the market is sharpening" story would step down every year. The only out-of-sample year is 2025, no longer a virgin holdout, with a lower bound (+0.44%) that scrapes break-even — statistically indistinguishable from "the edge regressed to a small positive mean and is bouncing around." So the pattern is *consistent with* a decaying edge and equally consistent with regression to a small positive mean; the sample cannot tell which. The trustworthy claims are the two that don't depend on a flattering year: the kept-vs-rejected gap, and the direction of the in-sample-to-out-of-sample shrinkage. The year-by-year magnitude is decoration on those.

One caveat sits under the whole table: no betting number here is perfectly clean out-of-sample. The 2025 figures were sealed against the *model's* selection but examined enough since that a precise estimate is partly survivorship; the 2021–24 figures are flatly in-sample. The cleanest evidence isn't any single ROI but the *shape* — the rejected pool losing money, the in-sample numbers shrinking as they should. The only number that will arrive uncontaminated is the one not on this table yet: the live 2026 log.

X.V Calibration and CLV

The uncertainty discipline of this document — put an error bar on every number — has to be turned on the document itself. Three claims here are stated a notch more confidently than the math strictly licenses. None is wrong; all are *sharper* said precisely, and saying them precisely pushes every estimate in the conservative direction, which only reinforces the verdict the document already reaches.

The Brier improvement is skill, and we owe you the split. X.I decomposed Brier as `Uncertainty − Resolution + Reliability` and flagged that the ~0.07 we bought back is skill — resolution net of a reliability penalty — not pure resolution. This is more

than pedantry: two forecasters can post the *identical* Brier with completely different splits, one with huge resolution dragged down by overconfidence, another with modest resolution and perfect calibration — same Brier, very different machines, very different betting consequences, because the EV math is poisoned by reliability and indifferent to resolution. So "we're tied with Vegas on Brier (0.1714 vs 0.1783 on the folds)" would be far more convincing as a decomposition: the tie could hide the market having more resolution while we have better calibration, or the reverse. The interesting question — better-sorted or more honest? — lives entirely in the split, which a single Brier averages away. That decomposition is a to-be-published artifact; we will not fabricate it here.

Calibration is asserted, not shown, and the blind spot is where the money is. The document elsewhere calls our fold calibration "near-perfect across bins," but nowhere displays the bins. Softened to what we can defend: we see no detectable miscalibration at the bin resolution we can measure — weaker and truer. Each calibration bucket's error bar follows the `sqrt(p(1−p)/n)` rule, and the high-confidence buckets that *define Tier A and carry the betting weight* are the smallest, so they have the *widest* error bars. The one region where a calibration flaw would cost real money — overconfidence in the high bins — is precisely where our measurement is least able to catch it. Saying "near-perfect" while the 85% bin might hold forty games claims a precision the data doesn't grant. The artifact that would settle it is a reliability diagram with the count n printed on every point, and it is the single most important thing this document still owes the reader. (The winner's-curse check from X.III — selected-pool 0.810 predicted against 0.818 realized — is reassuring but is a single point on a curated subset, not the whole diagonal; it answers "did we fool ourselves on the bets we made?", not "are the probabilities honest everywhere?")

CLV, against a named sharp close, with a numeric kill-threshold. Closing-line value — did the market move *toward* your pick after you committed? — is the leak-proof, weekly, out-of-sample heartbeat, since no feature leakage can retroactively bend an external market's number. Three tightenings. First, "beating the close" is meaningless until you name *whose* close: a soft book's stale number proves nothing, so the benchmark must be a named sharp close — Pinnacle or Circa — or it is a vanity metric. Second, the early read (Tier A picks closing +0.91 points in our direction, Tier B +0.35) has no sample size attached, so it is half a number — weakly positive in sign, silent in magnitude, illustration not evidence. Third, the kill-condition needs a number: made numeric, *kill-condition 3 fires if mean CLV is negative with a one-sided 95% upper bound below roughly +0.2 points over at least $n \geq 150$ picks, against the named sharp close.* (One thing CLV does *not* protect against is price-data corruption: it is immune to the feature leak of Part VII, but a swapped open or close in the feed makes CLV garbage exactly as it made the EV math garbage — same swapped-line risk, same pending live guard.)

The independence assumption — the crack under every error bar. This is the deepest. Every standard error, confidence interval, LB95, and Kelly stake here uses a formula of the `sqrt(p(1−p)/n)` family (or, for ROI, a per-bet return SD over `sqrt(n)`), and every one assumes each game is an independent draw — which a Saturday slate is not. A cold front depresses scoring in five games at once; a market mispricing tempo offenses replicates the same error across every tempo team we bet; the same team in a moneyline pick and an ATS pick rises and falls on one afternoon. When draws are positively correlated the *effective* sample is smaller than the raw count, so the true error bars are wider than we quoted. The direction is certain — correlation only ever widens bars — so the 73.7% might really be ±4 not ±3.1, and the LB95 +0.44% might scrape below zero.

The betting consequence is sharper: per-bet quarter-Kelly *over-bets a correlated portfolio*. Kelly (Part IV) sizes one independent bet; stake eight correlated picks each at its own quarter-Kelly and your total Saturday exposure exceeds quarter-Kelly on the real portfolio, because the diversification Kelly assumes isn't there. The fix is a weekly aggregate-exposure cap below the sum of the individual stakes, not a smaller fraction picked from a hat. And every lower bound should travel with its estimator's name: "LB95 +0.44%" here is a one-sided normal approximation that assumes independence; a bootstrap resampling *whole weeks* respects the correlation and is the more honest estimator. (The backtest's bankroll Monte-Carlo already bootstrapped games within weeks, so its ruin and drawdown figures are more honest than the headline SE — but the LB95 is still a per-bet-independence number.)

None of this changes the verdict the document already lands on: small, real-but-modest, promising-not-proven. If anything it reinforces it, because every correction points the same conservative way. The truth is a little fuzzier, a little wider, and a little more modest than the cleanest version of each number suggests — which is, by now, the only kind of conclusion this document knows how to reach.

Part XI — Limits and the Road Ahead

Everything before this part argued that v9.5 is sound: calibrated, honestly validated, carrying a small but real betting edge. This part is the humility clause. It states, in advance, exactly what would prove that wrong; it shows that a stranger could rebuild every number from scratch and catch us if we lied; and it lays out where the model goes next and why we are nonetheless freezing it untouched for the whole 2026 season. The three subsections answer three questions a careful reader is right to ask: *how would you know if you were wrong, how do I check your work, and why aren't you still working on it?*

XI.I How We'll Know If We're Wrong

The most credible thing a model document can contain is not a list of what the model gets right. It is a list, written before the season starts, of exactly what would make us kill or demote it — and a commitment to act on that evidence when it shows up. A kill-condition you invent after seeing the results is not a kill-condition; it is an excuse. These are written first.

The mechanism that makes this binding is the locked pick log (the brand, described in I.II; the integrity rule in XI.II). Every 2026 pick is timestamped before kickoff and written to an immutable record. We cannot quietly forget the bad weeks and remember the good ones; the log forbids it. So the question is never *whether* we'll be graded — the schedule and the log guarantee that — but *what grade fails*, decided now, while we still have no idea what the grade will be.

Closing-line value: the free weekly signal

There is a deep problem with judging a betting model by whether its bets win: you have to wait for the games, and even then the sample is tiny. One game can never confirm or refute a probability (the point made in II.I and V.III), and the same trap scales up to a whole Saturday — going 4-6 tells you almost nothing, because variance alone produces 4-6 weekends constantly even from a genuinely winning model.

Closing-line value (CLV), defined in II.I, is the way out. It asks one question: did the betting line move *toward* our pick after the moment we locked it? When a game opens at home −6 and we log a market-blind pick on the home side, and by kickoff the line has drifted to home −7.5, the sharpest money in the world ended up *more* convinced of exactly what we said — we "beat the close" by 1.5 points. That is positive CLV. If the line drifts to −4.5 instead, the market moved against us: negative CLV.

This makes CLV the best falsification instrument we have, for three reasons.

It arrives weekly and in volume. Every pick in the locked log generates a CLV reading whether it wins or loses, and the reading lands the moment lines close — before a single ball is snapped. We accumulate evidence roughly an order of magnitude faster than by waiting for win/loss outcomes.

It is out-of-sample by construction. CLV is measured against where an external, independent market lands — a market we do not control and that does not know our model exists. There is no way to overfit to it.

It is leak-proof in the way our old accuracy numbers were not. Part VII is the autopsy of how a season-aggregate leak inflated our accuracy by ten points. CLV cannot

be faked that way: the line moved or it didn't, and no feature engineering can make a closing line retroactively agree with a Wednesday pick. CLV is genuinely immune to the leakage class of Part VII. It is *not* immune to price-data corruption — the swapped-line problem of X.III — because CLV computed off a corrupted close is garbage in exactly the way the EV math was. CLV closes the leak door and leaves the corruption door exactly as open as it is everywhere else in the pipeline.

The early read on 2025 is faintly encouraging and nothing more: Tier A picks closed +0.91 points in our direction, Tier B +0.35. Weakly positive — genuine corroboration, not a victory lap.

A cold start means nothing; a season underwater means the thesis is dead

The most dangerous moment for any public model is the first bad stretch, because that is when the temptation to "fix" it — and quietly re-open every overfitting trap the experiment ledger (IX.II) was built to close — is strongest. So we have to be precise, in advance, about the line between noise and signal.

The arithmetic is the same standard-error math from V.III, run forward. Our policy placed 165 bets across the 2025 season, roughly a dozen a week. A "2–4 week cold start" is therefore about 25–50 bets. Suppose the true hit rate is the 77% the 2025 one-shot suggested. Over a 30-bet stretch the standard error on the observed rate is

```
SE ≈ sqrt( p(1−p) / n ) = sqrt( 0.77 × 0.23 / 30 ) ≈ sqrt(0.0059) ≈ 0.077  →  about 7.7
```

A rough 95% range is plus-or-minus two of those, about ±15 points. A genuinely 77%-true model will, over a random 30-bet window, routinely *show* anything from the low 60s to the low 90s. A 30-bet stretch where the model goes 17-13 (57%) feels awful and means nothing — the window is simply too small to tell a 77% model having a bad month apart from a coin flip being itself.

A full season is a different animal. Across ~165 bets the standard error tightens to about 3.3 points (sqrt(0.77 × 0.23 / 165) ≈ 0.033), so the 95% range narrows to roughly ±6.5 points and a result is no longer drownable in noise. The asymmetry is the whole point: small samples can only ever exonerate, never convict. So the commitment, made now: a 2–4 week cold start is presumed variance, we do not touch the model, and we watch CLV (which converges faster, being continuous rather than binary). A full season underwater, with negative CLV to match, has falsified the central thesis, and the honest move at that point is to kill or rebuild, not to defend.

The pre-stated kill-conditions

Each tripwire below is stated as a falsifiable condition with a number, because a kill-condition without a number is just a feeling.

Kill-condition 1 — the EV-gate stops discriminating. The entire betting thesis rests on the gate separating good bets from bad. The 2025 one-shot measured this cleanly: the bets the gate *kept* returned +8.5% (n=268), the bets it *rejected* returned −3.2% (n=380). That gap is the edge (X.IV). Across 2026 the kept pool must continue to beat the rejected pool; if the rejected pool matches or beats it over a full season, the gate is a coin flip dressed up as a filter and the policy is dead, even if total ROI happens to look fine on a lucky run. This is sharper than overall profit because it is internal and self-controlled: both pools face the same season, the same variance, the same market — the only difference is whether our math flagged them.

Kill-condition 2 — calibration drifts out of its bins. Calibration (defined in II.II) is the property the probabilities live or die on: 70%-picks should win about 70% of the time. On the folds it was near-perfect, and the winner's-curse check on the selected pool was clean (kept bets carried model probability 0.810 against a realized 0.818). The 2026 test buckets the season's picks by stated confidence; a 65–70% bucket that wins *55%* over a full season means the model is systematically overconfident and every EV and Kelly number built on it (Part IV) is poisoned at the root. Per-bucket samples are smaller than the whole, so we judge calibration on the season's accumulation, not on any single week's bin.

Kill-condition 3 — CLV negative all season. This is the leak-proof one. The rule, stated as a tripwire with a threshold, an estimator, and a minimum sample: kill-condition 3 fires if mean CLV is negative with a one-sided 95% upper bound below roughly +0.2 points, over at least $n \geq 150$ picks, all measured against a named sharp closing line. In plain English, it is not enough for CLV to dip negative on a bad week; the *season's accumulated* CLV must be negative, and its plausible range must not even reach a fifth of a point of positive value, over a sample at least as large as the 165-bet season the original claim was staked on. Season-long negative CLV is a kill *even if we made money*, because profit on negative CLV is luck the next season is expected to give back — the most dangerous illusion in betting. The symmetric promise holds too: season-long positive CLV with a losing record is treated as "edge confirmed, results unlucky, keep going."

Kill-condition 4 — the policy stops beating its own flat baseline. The EV-gate earns its complexity only by roughly doubling the edge over flat-tier betting (+8.46% versus +3.91% for flat Tier A or B in 2025). If over 2026 the gated policy fails to beat flat tiers, the gate is machinery without edge and the simpler approach should replace it. This kills the *policy*, not the model — worth stating separately, because the model can be fine while the cleverness on top of it turns out to be noise.

What is deliberately *not* on the list: "we lost money this week," "our top pick blew a lead," "we went 0-4 Saturday." None carry information at their sample size, and pretending they do is exactly the failure of discipline these tripwires exist to prevent.

Every condition is season-scale or CLV-scale, because those are the only scales at which the evidence can convict.

The open failure mode we're already watching

Honesty requires naming the one place we know there is live risk and the fix is not fully shipped. The swapped-moneyline corruption of X.III — about 53 historical games (2.3%) where the moneyline was priced against the spread — is neutralized in the backtest by the price-sanity guard (drop any bet where moneyline-implied and spread-implied probabilities disagree by more than 0.10). But the same corruption risk lives in the live pipeline: the real-time EV and Kelly calculations that will drive 2026 betting could be fed a swapped live price and over-select a phantom-value bet, because an EV filter is *built* to chase exactly that signature. The live guard is pending approval, not yet shipped. So it goes on the falsification dashboard explicitly: any live pick flagged with an implausibly large EV edge gets manually inspected for a price swap before it is trusted. That is a data-integrity failure, not a model failure, and it gets the same treatment as a kill-condition — stop, inspect, distrust the affected picks until the guard ships.

There is a deliberate asymmetry across all of this, and it is the right one. The conditions that would make us *confident* are demanding and slow (a full season, beating the close, holding calibration). The conditions that would make us *suspicious* are cheap and fast (negative CLV early, a phantom-value flag, a drifting bin). A model quick to doubt itself and slow to congratulate itself is the only kind worth betting real money behind.

XI.II Provenance and Reproducibility

Every number in this document — every accuracy figure, every feature, every betting result — traces to a specific public data source and a specific script that turns that data into the figure. The standard this section holds itself to is unforgiving:

> A stranger with the code, the API keys, and a free weekend could rebuild every data file from scratch, re-run the validation, and get the same figures we published — or catch us if we lied.

That is the difference between a model you are asked to trust and a model you can check. We want the second one.

Three data sources

The model eats facts, and the facts come from exactly three outside sources — all public, all factual (scores, ratings, prices), never logos or anything copyrighted. The

legal and scientific reason this matters is the same one: scores, ratings, and prices are facts, and facts are not copyrightable (the *Feist v. Rural Telephone* principle). A model built on proprietary, unshareable data can never be independently re-checked. Ours can.

CFBD (the CollegeFootballData API) is the analytics backbone — the source for almost everything the model itself learns from. A free key unlocks SP+ ratings (Bill Connelly's opponent-adjusted team quality, always attributed to him), every final score back to 2016 (the targets, see VI.II), team and advanced stats, the 247Sports recruiting composite, and records, rosters, and coaches. If CFBD vanished, the model would have nothing to learn from.

ESPN's public scoreboard, which needs no key, supplies the 2026 schedule (who plays whom, when, where — CFBD's lags) and live/final scores for grading locked picks. Two documented quirks: ESPN spells some teams differently than CFBD, reconciled by a name-normalization map, and it omits North Dakota State from its FBS feed until mid-2026, so NDSU's eight Mountain West games are hardcoded from the published schedule. Patches, not silent fudges.

TheOddsAPI supplies the moneylines and spreads the entire betting layer is measured against — historical lines (used to build the `spread_diff` feature, the single strongest feature in the model at bivariate r ≈ +0.66, see IX.I, and to backtest the policy) and live lines (this week's prices for finding EV bets). The blunt honesty note belongs here too: the swapped-moneyline corruption of X.III lives in the historical lines file, it is guarded in the backtest, and the live guard is still pending — a named, open risk, not a solved problem.

Frozen artifact

One rule the entire credibility of the 2026 picks rests on: `data/win_model.rds`, the trained model, is immutable for the whole season — not retrained, re-tuned, or re-blended, not once. The locked pick log is only honest if every pick in it came from the same model. Imagine retraining in week 7 after a rough September: the season-long hit rate becomes a blend of two models, the second trained on data that includes games it is about to be judged on, and the log silently becomes the self-graded exam that Part VII confesses to. Worse, every "improvement" is most tempting precisely when we are losing — which is exactly when variance, not the model, is to blame (XI.I). Freezing is not caution; it is integrity infrastructure, the model locked the way a fighter's weight is locked at the weigh-in so the contest is real. Retraining happens only between seasons, on explicit request, with deliberate data-quality verification.

The script map

Every figure comes from one of a small set of scripts. Run the script, get the number.

`scripts/fetch_data.R` rebuilds the core data files (teams, games, records, stats, recruiting, rosters, coaches) from CFBD and ESPN; it carries the most safety guards, because CFBD occasionally drops a handful of FBS teams from its classification list. `scripts/fetch_live_odds.R` rebuilds the live betting lines from TheOddsAPI; a missing file degrades gracefully to a neutral spread signal rather than crashing. `scripts/train_model.R` regenerates the model and its headline numbers — 73.7% accuracy, Brier 0.1712, AUC 0.816 — by training both heads (VIII.I) on 7,179 games (2016–2025), fitting the learned blend, and running the sealed validation, all with the point-in-time reconstruction whose absence inflated every pre-v9 number. It is run only on explicit request, never in-season.

`scripts/experiment_v91.R` is the sealed harness behind why you can trust the *design*. Every "should we change the model?" question — the 17 experiments, 1 kept ledger of IX.II — is answered here, and one line in the code makes the discipline verifiable:

```
fdf <- fdf[fdf$season <= 2024, ]   # 2025 NEVER enters the harness
```

The harness physically cannot see 2025. It runs walk-forward CV on folds 1–4 (test seasons 2021–2024) against a Brier keep-bar of −0.0010; the 2025 holdout was spent once, on the frozen design, and is now off-limits. That one line is what turns "we pre-registered before touching the holdout" from a promise into a fact anyone can read. `scripts/build_betting_edge_cache.R` pre-computes the EV-gated edges shown on the Picks tab. `scripts/log_predictions.R` writes the locked pick log: run by cron in-season, it LOCKs (predicts every upcoming game, writing rows immutable once written, keyed by season/week/teams/model version) and GRADEs (fills in actual winners once scores exist). Because rows are keyed by model version, the frozen-artifact rule is enforced at the data level, not just by good intentions.

There is a second document, `ENGINE_SUMMARY.md` — the engineering source of truth, terse and internal, kept in lockstep with the actual code. When this explainer and the engine summary ever disagree on a number, the engine summary wins; this explainer is written *from* it. The figures here are not a marketing gloss; they are the same numbers the engineers hold themselves to, rendered into plain English.

A stranger could rebuild it

Three public data sources. A handful of scripts. One frozen model. One immutable pick log. One engineering source of truth. The claim at the top of this subsection is therefore not a boast but a checklist: a stranger could rebuild and re-check every fig-

ure in this document, and the things they *couldn't* yet check — the live 2026 results, the still-pending live price-guard — are exactly the things we have named as open, because the only honest provenance is the one that tells you where the trail ends, too.

XI.III The Road to v10

This subsection answers two questions that sound like opposites but are the same question from two ends: where does the model go next, and why are we comfortable not touching it for three months while the season plays out? Both run through one fact established in IX.III — the thing holding this model back is not the cleverness of its math but the information in its columns. Accept that and the roadmap writes itself (go get new information) and the freeze justifies itself (nothing left to gain by re-fitting the same columns, and much to lose).

New inputs, not a new architecture

The most important finding of the whole experiment campaign is in the *shape* of the losers' column (IX.II). The most exhaustive "tune the existing machine harder" effort, a 70-config hyperparameter sweep, moved Brier −0.0006 — below the −0.0010 bar, dead. The one change that cleared the bar, the margin architecture (VIII.I), moved Brier −0.0015, and it worked precisely because it changed *what we asked the model to learn* rather than how hard we fit the same target. The lesson is the information ceiling (IX.III): when the largest gain anyone can wring from these columns is fifteen ten-thousandths of a Brier point, no optimizer alive — not a neural net, not AutoML, not a smarter blend — will conjure a tenth of a point out of them. You cannot draw a better curve through data that does not have the answer in it.

So v10 runs through new inputs, specifically the class of information the point-in-time discipline of Part VII deliberately throws away. That discipline rebuilt every feature to contain only what was knowable before kickoff, which is why the model is honest — and it is also why the model walks into every game carrying yesterday's information (last week's ratings, the season's progressive form, the spread as of lock time) while the market carries today's (the inactives report, the weather radar, the late line moves). The gap between yesterday and today is exactly where the market still beats us, and exactly where v10's gains have to come from. Four candidates, each a new column, each validated on folds 1–4 under the same pre-registered −0.0010 keep rule, with 2025 left untouched:

1. *Game-time quarterback status, confirmed at lock time* — the big one. A team with its starter and the same team with its third-stringer are not the same team, and our point-in-time features have no clean way to know which is taking the field; the market knows, and a confirmed ruling moves a line several points in min-

utes. Tested by building a QB-availability feature on the folds and running it through the sealed harness against the bar.

2. *Injury and availability feeds beyond the QB* — offensive line, secondary, a star edge. Lower per-game leverage, but it accumulates. The catch is honest: availability data is noisy and historically spotty, so the real test is whether we can even reconstruct clean, point-in-time-correct availability for 2021–2024. A feature you can't rebuild cleanly for the folds may end up validated only on live 2026.
3. *Weather at kickoff* — wind, precipitation, temperature. This is the input that could re-open totals as a product: IX.II rejected totals (RMSE 16.32 versus the market's 15.87) largely on a weather-and-pace story, and committed in writing to a fresh re-test of totals if a future version ever adds real game-time inputs. Historical weather is one of the easier inputs to backfill cleanly, which makes it an unusually fair fold test.
4. *Closing-line timing and movement* — not just the spread but when and how it moved. The most delicate of the four, because a feature built on the closing line risks smuggling in information you would not have at lock time. The fold test must be ruthless about timestamp discipline: the feature may use only line data available at the moment we would actually bet, never the close itself. Tested sloppily, this becomes a new leak; tested correctly, it is a legitimate input.

The discipline does not relax for v10. The garden of forking paths (IX.I) is just as dangerous when the candidates are new inputs as when they were new architectures — maybe more, because "we found a great new feature" is more seductive than "we found a great new penalty term." And the one honest limit on all of it: the 2025 holdout is spent (VIII.IV). Folds 1–4 can screen candidates; the genuinely clean, never-peeked test of any v10 input is the live 2026 season itself. That is not a weakness of the plan — it is the plan.

Why we freeze v9.5 from strength

Freezing for three months is not flying blind; it is the disciplined move you make once the audit surface is covered and the only remaining test is live play. Two reasons converge.

First, the artifact must be immutable or the pick log means nothing (XI.II). In-season retraining would smear the season across a sequence of models, and any bad stretch could be blamed on "the old version" while the wins are credited to "the improved one." That is the tout behavior this whole project is defined against (I.II). One model, one season, every pick attributable, the losses displayed first-class next to the wins.

Second, we have categorically checked the *known* failure surface. Freezing is confidence, not hope, because each known way-to-fail has a documented check:

- *Data leakage* — the class that already burned us (Part VII). Guarded now by per-feature point-in-time reconstruction plus a written post-mortem naming the exact mechanism. The most-checked surface in the project, because it cost the most.
- *Serving-vs-eval drift* — production quietly running a configuration you never measured. Fixed June 11: every artifact stores `chosen_variant` and every serve branch checks it, so production serves the validated "Learned logit" exactly (the Platt step is gated on the flag, see VIII.III). Honest qualifier: this holds because the production artifact is well-formed, not because the loader refuses a malformed one — hardening it is staged.
- *Calibration honesty* — checked by Brier and bin-by-bin calibration; passed (Brier 0.1712 sealed 2025, 0.1714 on the folds, near-perfect across bins). The methodology puts Brier first (III.III) precisely so this class can't slip through.
- *Betting-layer self-deception* — the most recently closed class (X.III): the self-graded tier table, the impossible +20% backtest, the hit-rate-versus-EV confusion, the swapped prices, all taken apart and the policy pre-registered and spent-once. What survived is +8.46% (LB95 +0.44%), labeled in writing as promising, not proven.
- *Data integrity* — bad inputs producing confident garbage. Guarded at shared chokepoints: an FCS-eligibility filter, the fetch-drops-teams guards, and an integration test plus render gate that fail loudly before every deploy.
- *Garden of forking paths* — the meta-class. Defended four ways (IX.I): pre-registered metric, pre-registered threshold, sealed test set, and the public ledger of every experiment tried, not just the survivor.

Six classes, six documented checks. But "categorically checked the *known* failure surface" is not "nothing can go wrong," and the word *known* is doing honest work. The unknown unknowns are, by definition, not on the list, and there are almost certainly some — the same way the season-aggregate leak was an unknown unknown right up until the morning it became the most obvious thing in the world. The named, still-open item is the live price-guard (XI.I).

So how do we square "there are unknowns" with "freeze it anyway"? The remaining unknowns are not resolvable by more tinkering — the information ceiling says there is nothing left to find in these columns, and the holdout is spent so there is nothing left to honestly test against. Tinkering now would actively make things worse: re-opening the forking-paths risk and breaking pick-log integrity. The disciplined move and the safe move are the same move. Freezing v9.5 is not the end of

the work; it is the start of the only experiment that can settle it — a full season, one frozen model, graded in public against tripwires written in advance.

❦

Appendix — Critiques Addressed

A.I Burning Questions

Short, sharp answers to the questions a skeptical reader is right to ask. Each points to its home section for the full argument.

Why trust a model only tied with Vegas to beat Vegas? Because beating Vegas on average is not the product — selectivity is. The model is market-equal on overall accuracy, but it doesn't bet every game; it bets the ~5 spots a week where it most disagrees with the price, and a market-equal model can still hold a real edge in a thin, mispriced tail even while matching the average everywhere else. See III.IV and III.V.

Isn't +8.46% on one season just luck? It might be small, but it isn't only luck. The number is a pre-registered, frozen, scored-once result, not a figure tuned until it looked good; it survived leave-3-out (dropping the best three bets still leaves +6.66%); and the EV-gate's *kept* pool beat its *rejected* pool by a wide margin on the same season — an internal control luck doesn't easily fake. The honest caveat stays: the lower bound only just clears zero. See V.III and X.IV.

If there's a real edge, why publish it? Because a publicly posted +EV pick is an instruction to a crowd to bet the same side, which moves the line against the next person and erodes the edge — the more public and accurate the log, the faster its own picks decay. We keep it anyway: the log's *evidentiary* value (proving the edge was real) and its *actionable* value (letting you capture it) point in opposite directions, and the first is why it exists. See IV.V.

Why is the playoff simulator less trustworthy than the picks? Because they are graded on completely different scales. A pick is a single calibrated probability that gets logged before kickoff and checked against reality; a season-long simulation compounds many uncertain estimates across months, and small errors multiply. The picks carry a falsifiable track record; the simulator is a directional illustration, not a forecast you should stake on. See V.II.

Didn't you say 83.9% before? Yes — and it was wrong, which we proved ourselves. The old numbers (83.9% / 85.1% / 83.6%, Brier 0.1200) were inflated by about ten points by a season-aggregate leak that let features contain each game's own outcome. We

found it, documented the exact mechanism, and re-validated under pre-registration; the honest number is 73.7%. See Part VII.

Why not just add more data and signals until it's great? Because the ceiling here is the information in the columns, not the cleverness of the fit. Seventeen experiments showed the biggest "tune harder" gain was −0.0006 Brier, below the keep bar, while the only win came from changing the target. No optimizer can extract information the features never contained, so the next gains must come from genuinely new inputs (QB, weather, injuries), not from squeezing the existing ones. See IX.III.

Why bet quarter-Kelly instead of full Kelly? Because Kelly is growth-optimal only if your probabilities are exactly right, and ours are estimates with error bars (V.III). Quarter-Kelly is the insurance premium against our own overconfidence — it sacrifices some theoretical growth to survive a stretch where the true edge is smaller than we measured. See IV.IV.

A 70%-confidence pick lost — was the model wrong? No. A single game can never confirm or refute a probability; a 70% call is *supposed* to lose three times in ten. The model is judged on calibration over the pile (do the 70%-picks win ~70% of the time?), not on any one result. See II.I and V.III.

A.II Breaking It Down by Field

The same case, translated for four audiences. Tight and quotable on purpose.

To a statistician. "Two-head margin regression — OLS and XGBoost on sign(m)·log(|m|+1) — blended by a learned logit stack; probabilities via pnorm against fitted residual SDs. Point-in-time features, Bonferroni-screened (0.05/45). Walk-forward CV; calibrators fit folds 1–3, variant pre-registered on fold 4, single evaluation on a sealed 2025: 73.7% / Brier 0.1712 / AUC 0.816, against a 74.2% market favorite. Previous published numbers were ~10pp inflated by within-season aggregate leakage; we found it, documented it, and re-validated under pre-registration." Only people who have never audited anything think confessing a leak weakens you.

To a data scientist. "Seventeen pre-registered experiments on a sealed-holdout harness, one adoption — the margin target. Elastic net lost to ANOVA-selected OLS by +0.0009; penalization solves p≈n and we're at p≪n with a pre-screened pool. The information ceiling is empirically mapped: next gains are new inputs (game-time QB/injury/weather), not architecture."

To a bettor. "Roughly market-equal on headline accuracy — nobody honest claims otherwise without inside information. The product is selectivity: ~5 disagreements a week. The one finding I'd stake credibility on is the discrimination test — the prices the model flagged as +EV returned +8.5% on the sealed 2025 holdout while the ones

it rejected lost 3.2%, so it's sorting mispriced lines from fair ones, not riding favorites. It's a real but modest edge, the bottom of its range barely clears zero, so I'd bet it flat and small, not press it. Mid-confidence singles at fair prices and big ATS disagreements (66% at 8+ points of edge) are the spots; the chalk tier is for parlay legs, not EV. Every pick locks publicly before kickoff and grades itself. Judge the log, not the pitch."

To the skeptic (including the one in the mirror). "You're right that 83.9% was too good to be true — it was, we proved it ourselves, and the proof is published in the repo with the exact mechanism. The current number survived a protocol specifically designed to prevent us from fooling ourselves again: sealed holdout, pre-registered variant, public dead-experiment ledger. And the 2026 season is the only validation that can't be gamed even in principle: the picks are timestamped before the games are played."

A.III Glossary

One-line definitions, loosely in order of appearance. Most terms are defined in full where they first do real work — chiefly Part II.

- *Probability vs odds* — probability is a share of outcomes (0–1); odds are the same belief re-expressed as a payout ratio. (II.I)
- *Moneyline / American odds* — a price quoted as −150 (risk $150 to win $100) or +130 (risk $100 to win $130). (II.I)
- *Implied probability* — the win-rate a moneyline breaks even at; −150 implies 150/250 = 60%. (II.I)
- *The vig (juice)* — the book's built-in margin; implied probabilities across a game sum to more than 100%. (II.I)
- *De-vigging* — stripping the vig back out so implied probabilities sum to 100% and compare to a model. (II.I)
- *Point spread / covering* — the handicap on the underdog; covering is beating the spread, not just winning. (II.I)
- *Expected value (EV)* — average dollar result of a bet replayed forever: p·(win) − (1−p)·(stake); the only reason to bet. (II.I)
- *Closing-line value (CLV)* — whether the line moved toward your pick after you locked it; the leak-proof weekly signal. (II.I, used in XI.I)
- *Variance* — the spread of outcomes around the average; why a 70% bet still loses ~3 in 10. (II.I)
- *Normal distribution / bell curve* — the symmetric distribution we assume game margins follow. (II.II)

- *Standard deviation (σ)* — typical distance from the average; σ = 13.5 points on the margin scale. (II.II)
- *Calibration* — when you say 70%, it happens ~70% of the time; honesty of the probability, separate from accuracy. (II.II)
- *Model / feature / training row* — the fitted rule; one input column; one past game it learned from. (II.II)
- *Margin architecture* — regress the dampened point margin, derive win probability from it; one signal, both products. (VIII.I)
- *Log-dampened margin* — sign(m)·log(|m|+1); blowout points carry shrinking information. (VIII.I)
- *Residual* — leftover error after a prediction (actual minus predicted). (II.II)
- *OLS (ordinary least squares)* — plain linear regression minimizing squared error; the GLM head's engine. (VIII.I)
- *Boosting / XGBoost* — many small decision trees added in sequence, each fixing the last's errors; the second head. (VIII.I)
- *RMSE (root mean squared error)* — average prediction error in original units (points); 16.32 us vs 15.87 market on the rejected totals model. (IX.II)
- *Logit / log-odds* — probability on a (−∞, +∞) scale where blending is linear; how the two heads combine. (VIII.III)
- *Point-in-time features* — every feature reconstructed from only what was knowable before kickoff; the discipline whose absence cost 10 fake points. (Part VII)
- *Season-aggregate leakage* — features built from full-season tables contain each game's own outcome. (VII.I)
- *Walk-forward CV* — always train past, test future; guards across-season leakage and only that. (VII.II)
- *OOF (out-of-fold)* — a prediction made by a model that never trained on that game. (VIII.IV)
- *Pre-registration* — committing to the decision rule before seeing the test result. (VIII.IV)
- *One-shot / spent holdout* — a sealed test set may be evaluated once; afterward it's a fitting target, not a test. (VIII.IV)
- *Bivariate correlation* — feature vs outcome, nothing controlled; spread_diff's is strongest at r ≈ +0.66. (IX.I)
- *Multicollinearity* — features correlated with each other; destabilizes linear coefficients, harmless to trees. (IX.I)
- *Type I (sequential) ANOVA* — order-dependent variance attribution; can score a vital-but-redundant feature as p≈1. (IX.I)

- *Bonferroni* — divide the significance bar by the number of tests run (0.05/45 here). (IX.I)
- *Elastic net / lasso / ridge* — penalized regressions for p≈n and collinearity; tested, lost (+0.0009). (IX.II)
- *Garden of forking paths* — iterative choices that silently fit the validation data; the keep-rule plus ledger is the antidote. (IX.I)
- *Brier score* — mean squared error of probabilities; the honesty metric and the experiment currency. (X.I)
- *Accuracy* — share of games where the >50% favorite was correct; a thresholded yes/no per game. (X.I)
- *AUC* — probability the model ranks a random winner above a random loser; threshold-free ordering quality. (X.I)
- *pnorm / qnorm* — the normal CDF and its inverse; the bridge between margins and probabilities (σ = 13.5). (VIII.II)
- *Tier A/B/C* — confidence bands (≥0.80 / ≥0.65 / rest); A = survival chalk, B = EV, C = calibration tracking. (IV.III)
- *ATS edge* — market-blind model margin minus the closing line; bet at ≥3 (weeks 1–7) / ≥7 (week 8+). (X.IV)
- *Monotone edge buckets* — bigger disagreement should mean higher hit rate; its absence (totals) signals self-error. (IX.II)
- *Kelly criterion / Quarter-Kelly* — the growth-optimal stake formula; we bet a fourth of it as insurance against our own probabilities. (IV.IV)
- *Locked pick log* — immutable pre-kickoff picks, script-graded; the brand, and the only validation left that cannot be fooled. (I.II, XI.II)

Last updated: June 19, 2026 | Model v9.5 (frozen for the 2026 season) | cfb.terranalytics.com | Companion docs: ENGINE_SUMMARY.md (engineering source of truth), cache/experiments_v91.csv (raw experiment ledger)