Game Sales Forecasting Engine

Proposal Document

George Pearson

Table of Contents

I. Executive Summary - Predictive Game Sales Forecasting Engine

The global games market has become increasingly hit-driven, competitive, and volatile. Yet forecasting practices used by many publishers remain rooted in simple comparative methods: past-title analogues, franchise averages, expert judgement and static ratios. These legacy approaches overlook the complex, multi-factor dynamics that truly govern game performance across launch, mid-tail, and long-tail phases.

Our proposed Predictive Game Sales Forecasting Engine replaces these heuristics with a modern, data-driven system modelled on techniques widely used in quantitative finance and large-scale forecasting environments. It provides accurate, explainable, and risk-aware sales forecasts for any new title, from indie launches to AAA blockbusters, even before release.

A fully reproducible version of this forecasting engine, including code, feature pipeline, and Power BI dashboards is available in the accompanying GitHub repository. Link can be found in the appendix.

Why a Modern Forecasting Engine Is Needed

Traditional forecasting methods fail in three critical areas:

1. Cold-Start Prediction

New releases often lack prior sales signals. Genre averages and franchise benchmarks fail to capture variability in:

- Marketing intensity
- Price and discount strategy
- Seasonality
- Competition
- Expected decay behaviour

2. Post-Launch Volatility

Sales paths can diverge sharply due to:

- Viral adoption
- DLC drops
- Discount campaigns
- Shift in reviews or marketing sentiment
- Competitor releases

3. Commercial Decision-Making Requires Risk Bands, Not Single Numbers

Publishing teams need:

- Best/worst-case ranges
- Uncertainty estimates
- Scenario simulations
- Tail-behaviour expectations

Our Approach - A Multi-Layer Predictive System

Our engine integrates four modern forecasting layers, each addressing a limitation found in traditional models:

1. Behavioural Prior Curve (Launch → Tail Lifecycle)

A robust decay structure based on real-market patterns, inspired by volatility modelling frameworks:

$$Sales(t) = Peak \times e^{-kt} \times Seasonal(t)$$

This provides a stable starting point for any new title and captures:

- Expected launch demand
- AAA/AA/Indie behavioural templates
- Seasonality patterns
- DLC/discount-free baseline behaviour

2. Residual Machine Learning (XGBoost)

Instead of predicting sales directly, the model predicts residuals on top of the prior curve. This stabilises training, reduces scale issues, and enables clean explainability.

The residual model learns:

- Price/discount effects
- DLC timing
- Competitor intensity
- Franchise strength
- Marketing uplift
- Regional effects

3. Cross-Sectional Transfer Learning (XSTL)

The engine compares the new game with hundreds of historical synthetic titles and uses this similarity score to:

- Adjust uncertainty
- Scale confidence
- Modify decay expectations
- Improve predictions for cold-start scenarios

This allows the model to generalise intelligently across genres, franchises, and release periods.

4. Forecast Reliability + Uncertainty Distribution

Each forecast includes:

- P10 / P50 / P90 sales scenarios
- Spread-based uncertainty index
- Regime confidence
- Drift score
- Similarity score
- A composite reliability score

This transforms the forecast from a single guess into a risk-aware prediction envelope.

What This System Delivers

1. Multi-Horizon Forecasts

Accurate predictions for:

- Week-1
- Month-1
- Quarter-1
- Year-1

Plus, complete 52-week post-launch curves.

2. Explainability & Transparency

Clear Power BI dashboard showing:

- Baseline vs blended forecast
- Cumulative trajectory with lifecycle phases
- Uncertainty bands (P10–P90)
- Promotion and DLC impact timing
- Marketing ROI vs no-marketing baseline
- Reliability scores over time

3. Scenario Planning & Stress Testing

Teams can simulate:

- High/medium/low marketing
- Discount timing changes
- DLC impacts
- Competitor launches
- Seasonal shifts

4. Conservative, Stable, Risk-Aware Guidance

The model emphasises:

- Realistic expectations
- Controlled smoothing
- Removal of artificial spikes
- Stable tail behaviour

Avoiding the inflated forecasts that often undermine publisher decision-making.

Commercial Value

This engine transforms forecasting from a heuristic exercise into a quantitative planning tool, offering:

- More accurate and consistent forecasts across genres and franchises
- Earlier insight into launch risk and upside potential
- Better marketing and budget allocation
- Improved inventory, demand planning, and revenue forecasting
- Auditable and explainable predictions, ready for internal and external stakeholders

In short, this system positions publishers to make confident, data-led decisions in an increasingly volatile market.

II. Business Problem

Modern game publishing operates under increasing uncertainty. Budgets are rising, release calendars are crowded and post-launch performance is shaped by rapidly shifting forces: platform algorithms, content updates, viral moments, competitor launches, discount cycles and fluctuations in player attention.

Despite this volatility, commercial teams are still expected to set revenue targets, plan marketing activity, and schedule promotions months in advance. The fundamental issue is not simply that forecasts are inaccurate; it is that publishers lack a structured, data-driven system that can describe and anticipate the full lifecycle of a game, before and after launch.

Below are the structural issues limiting current forecasting practices.

---

1. Cold-Start Forecasting Remains Highly Uncertain

New releases have little or no sales history, especially new IPs or titles launching into new regions or platforms. Publishers often resort to analogue-based forecasting or franchise averages, which fail to capture the wide variability in:

- Franchise strength
- Marketing intensity
- Genre and audience expectations
- Regional demand patterns
- Seasonal timing
- Competition in the release window

This produces forecasts that are frequently too optimistic, too conservative, or slow to adapt once real sales arrive.

2. Real-World Sales Patterns Are Complex and Non-Linear

Game sales rarely follow a smooth decay curve. They are shaped by mid- and long-tail behaviours that are difficult to model with traditional heuristics:

- Launch momentum
- DLC drops and content updates
- Discount and promotional campaigns
- Marketing bursts
- Review sentiment and social signals
- Platform-specific trends and featuring cycles

Conventional ratio-based models cannot integrate these diverse drivers or adjust dynamically as new signals emerge.

3. Commercial Teams Need More Than a Single Forecast Number

Modern decision-making requires forecasts that include uncertainty, scenarios and risk measures, not just a single deterministic value.

Teams require:

- High/medium/low sales scenarios
- Confidence intervals
- Risk-adjusted guidance
- Rapid reforecasting as data arrives
- Visibility into mid-tail and long-tail decay

Without these tools, teams struggle to:

- Plan marketing budgets
- Coordinate distribution and inventory
- Set realistic revenue expectations
- Schedule DLC releases
- Manage promotional calendars

A traditional “single number” forecast cannot support the full commercial lifecycle.

4. Existing Forecasting Models Lack Transparency and Diagnostic Capability

Most current methods provide no insight into why a forecast changes or which factors drive the result.

Static approaches lack:

- Driver attribution
- Reliability scoring
- Uplift decomposition
- Diagnostics for model shifts
- Scenario or what-if analysis

Without transparency, stakeholder buy-in is harder to achieve, and teams cannot confidently adjust strategy when conditions change.

Summary: Why a New Approach Is Needed

The industry requires a forecasting system that:

- Handles cold starts with consistent logic
- Adapts dynamically to real-world events
- Quantifies uncertainty rather than hiding it
- Provides transparent, interpretable forecast drivers
- Supports scenario planning and commercial decision-making
- Scales across genres, franchises, platforms, and regions
- Enables rapid reforecasting with each new week of data

Such a system moves forecasting away from subjective heuristics and toward a modern, data-driven predictive framework, similar to the sophisticated models used in quantitative finance, demand planning and large-scale forecasting environments.

III. Model Design and Data Inputs

Our forecasting engine is built as a structured, multi-stage pipeline designed to ensure that every prediction is deterministic, validated, feature-rich, risk-aware and explainable.

The model is intentionally modular, each layer is responsible for its own discipline:

Data generation → Validation → Feature learning → Prediction → Reliability → Output dashboards

---

Data Inputs and Feature Sources

To demonstrate the forecasting engine without requiring proprietary publisher data, we generate a fully deterministic synthetic game universe with 200 historical titles across 52 weeks. We then synthesise 3 new games with the same feature set but with sales data to be forecast, these three games are:

- NeonRift (AAA)
- Ashbound (AA)
- Pulsebreak (Indie)

1. Synthetic Training Universe

This dataset mirrors real-world market behaviour:

- Behavioural templates for AAA, AA and Indie
- Franchise systems (Mario, Pokémon, Zelda, Indie IP etc)
- Decay classes (Longtail, Standard, Fast)
- Region and platform (NA, EU, JP and fixed as Switch)
- Seasonality and holiday uplift
- DLC and discount events following realistic publisher patterns
- Marketing dynamics with AR(1) noise, genre differences, and decay
- Competitor intensity that evolves week-to-week
- Price and promotional depth
- Weekly rolling signals, change indicators, log features, and normalised marketing/competition

All randomness is seeded through a Seed Ledger, ensuring reproducibility. Each title ID is mapped to a fixed RNG state ensuring every run generates identical data.

This synthetic universe is used to train the model, demonstrate its behaviour, and allow clients to understand the framework prior to onboarding real data.

2. New Game Data

For our three new titles, the engine requires:

Title metadata

- Name and Dev type (AAA/AA/Indie)
- Franchise (if applicable)
- Platform and Region

Behavioural parameters

- Peak demand expectation
- Decay expectation
- Discount and DLC schedule

Marketing trajectory

- Pre-launch intensity
- Decay assumptions
- Post-launch campaign phasing

Calendar information

- Release date
- Seasonal effects (holidays, peak months)

The model can infer missing fields using behavioural templates or client priors.

3. Behavioural Templates (AAA, AA, Indie)

Each dev type contains a deterministic template with ranges for:

- Peak sales
- Decay rate
- DLC counts
- Discount frequency + depth
- Marketing peak + decay
- Price bands

This allows the engine to simulate realistic lifecycle curves even when client data is incomplete.

Architecture Overview

The forecasting engine uses a multi-layer architecture, each component addressing a specific modelling challenge.

1. Behavioural Prior Curve (Real Space)

The foundation of every forecast is a deterministic decay model:

$$Sales(t) = Peak \times e^{-kt}$$

Extended with:

- Seasonality
- Indie flattening after Week 20
- DLC uplift (with 2-week echo)
- Discount uplift (with 1-week echo)
- Dev-type timing multipliers

This forms the baseline expectation for each week independent of ML.

2. Residual XGBoost (Log Space)

To stabilise scale and improve interpretability:

- The model predicts log-residuals between actual log-sales and log-prior.
- This forces XGBoost to learn only deviations from expected behaviour (marketing bursts, competitor spikes, discount effects).
- Outputs are converted back to real space via:

$$\exp\left(\log(prior) + residual\right)$$

This reduces overfitting, removes scale issues, and produces clean uplift attribution.

Why XGBoost (Gradient-Boosted Trees) Is the Core ML Layer

XGBoost is a gradient-boosted decision-tree model designed for structured/tabular data, the exact format of weekly sales, prices, discounts, seasonality flags, competition and marketing features. It constructs an ensemble of many small regression trees, each trained to correct the residual errors left by the previous trees. This process is powered by gradient boosting, where the model computes the gradient of the loss function (e.g., log-RMSE) with respect to the current predictions and fits the next tree specifically to reduce that error. Over hundreds of boosting rounds, the ensemble gradually converges toward a highly accurate predictor.

To prevent overfitting — a critical requirement for commercial forecasting — XGBoost includes built-in regularisation:

- L1 regularisation shrinks or removes weak tree splits (feature sparsity).
- L2 regularisation penalises large leaf weights (smoother, more generalisable trees).
- Tree-level constraints (max depth, min child weight, learning rate) enforce conservative learning.

Together, these mechanisms create a model that captures nonlinear interactions (e.g., discount × marketing), dev-type differences, and sharp local patterns such as DLC spikes or competitor launches, without memorising noise.

It is an ideal choice for a commercial sales-forecasting engine because:

- It handles missingness and irregularities gracefully, unlike neural networks.
- It naturally captures nonlinear interactions that shape real game sales curves.
- It is resistant to overfitting when paired with residual learning, since the model only learns deviation from the prior decay curve.
- It is interpretable, providing feature importances and shapely explanations suitable for executive reporting.
- It outperforms deep learning on small-to-medium datasets, which is typical in game sales where only 200–500 titles exist.

In this architecture, XGBoost becomes a precision adjustment layer: the prior curve sets the lifecycle shape, and XGBoost fine-tunes weeks with marketing shocks, seasonal effects, franchise strength, or competition pressure, producing stable, industry-aligned forecasts.

3. Cross-Sectional Transfer Learning (XSTL)

The engine computes cross-sectional similarity between the new title and the entire historical universe via:

- Sales means / medians per week
- Dev-type cross-sectional stats
- Franchise-level weekly behaviour
- Mahalanobis distance in feature space

This gives:

- XSTL similarity score
- Cross-sectional features for both training and prediction

XSTL is crucial for cold starts because it allows titles to “borrow strength” from similar historical games.

4. Uncertainty Modelling

Each forecast includes a distribution, not just a point estimate:

- P10 (downside)
- P50 (base case)
- P90 (upside)
- Spread (forecast uncertainty index)

These quantify risk and help with scenario planning.

5. Anti-Spike Smoothing & Promo Safety Layer

Promo weeks and their echoes (DLC and discount) often produce unrealistic spikes.

The engine applies:

- Strict uplift caps
- Echo logic
- Promo-safe override that ensures predictions never exceed plausible uplift windows
- Small-prior protection (for curve stability)

This produces smooth, realistic lifecycle curves.

6. Reliability Scoring

The engine evaluates its own confidence using three distances:

- Regime Confidence
- Feature Drift Score
- XSTL Similarity

These are scaled 0–1 and combined geometrically:

$$\text{Reliability} = (\text{R} \times \text{D} \times \text{X})^{1/3}$$

The reliability score drives dynamic blending.

7. Ensemble Blending (Reliability-Weighted)

Final predictions blend:

- Prior curve
- Promo-safe XGB

With a minimum XGB weight of 0.30, rising to ~0.90 when reliability is high.

This ensures:

- Conservatism where data is weak
- Responsiveness where patterns are strong

Pipeline Walkthrough

Below is the full step-by-step model flow.

1. Feature Generation

The synthetic universe + new game pipelines generate:

- Raw behavioural features
- Events (DLC/discount)
- Marketing & competition signals
- Calendar effects
- Rolling windows
- Log transformations
- Cross-sectional signals

This produces a high-granularity dataset for training.

2. Data Validation

Institutional-grade validation checks include:

- Duplicate timestamp detection
- Negative sales checks
- Rolling z-score anomaly detection
- NaN flood ratio analysis
- Misaligned history detection
- Column-type validation
- Per-title and global health scores

Output:

- Per-title ASCII reports
- Global summary
- Consolidated health scores

3. Feature Pruning

The pruner performs a 7-stage cleaning & ranking process:

1. Numeric-only filtering
2. Missingness and zero-variance removal
3. High-correlation pruning
4. XGBoost gain scoring
5. Linear regression coefficients
6. Economic sign alignment
7. Composite Feature Quality Score (0–100)

Output:

- Selected_feature_list.txt (Top-N features + score threshold)

Ensures a lean, high-quality feature set.

4. Model Training

Pipeline includes:

- Building prior curve
- Constructing log-residual target
- Scaling time features
- Injecting XSTL cross-sectional statistics
- Computing CV SMAPE
- Training final XGB model

5. Prediction and Uncertainty

For each new title:

- Build prior curve
- Predict XGB residual
- Apply promo-safe override
- Blend based on reliability
- Compute P10/P90 distribution
- Compute marketing uplift & baseline curve

Output:

- Week-by-week forecasts
- Distribution range
- Reliability weights
- Diagnostic signals

6. Summary and Metadata

The runner produces:

- Detailed weekly forecast CSV
- Summary metrics (total, peak week, uplift, reliability)
- Metadata CSV
- Power BI-ready tables

IV. Comparative Advantage - Why This Approach Is Better Than Current Industry Practice

Most publishers rely on forecasting methods that were designed for simpler markets, where release calendars were less crowded, promotional strategies were predictable, and mid-tail performance mattered far less. These legacy techniques struggle to scale in today's environment, where marketing, DLC, seasonality, competition, and platform dynamics create week-to-week volatility.

Below we outline the common industry approaches and why they fall short. We then show how the proposed forecasting engine directly resolves each limitation.

Limitations of Current Industry Forecasting

1. Basic Regression on Historical Titles

Simple regressions (e.g., launch sales vs franchise strength) assume:

- Linear relationships
- Stable patterns across titles
- Minimal noise

But real game performance is non-linear, event-driven, and varies across genres. These models cannot handle:

- Promotional spikes
- Structural shifts
- Fat-tailed sales distributions
- Cold starts

2. Pure Curve Fitting

Traditional forecasting often fits a single decay curve to early sales.

However:

- Early data is noisy and unreliable
- Curve shape differs across AAA, AA, Indie
- DLC and discount events distort decay

3. Black-Box Machine Learning

Neural nets or generic ML models trained on raw sales:

- Require large proprietary datasets
- Overfit to noise
- Produce predictions with no explainability
- Cannot manage extreme spikes
- Do not distinguish structural behaviour vs residual noise

Without priors, they produce unstable curves and hallucinate in cold starts.

4. Manual Forecasting and Expert Tweaks

Human judgement remains valuable, but:

- It is inconsistent
- It is biased by recency and subjective impressions
- It does not scale
- It cannot provide risk bands or scenario simulations

Commercial teams end up debating opinions rather than aligning on structured intelligence.

5. No Transfer Learning Between Titles

Most methods consider each title independently.

This wastes insight from:

- Cross-genre patterns
- Dev-type behaviour
- Franchise sales trajectories
- Seasonal windows
- Competitor intensity dynamics

Without cross-sectional transfer learning, cold-start accuracy remains low.

6. No Uncertainty Ranges

Traditional forecasts output a single number.

Commercial planning needs:

- Downside risk
- Upside potential
- Confidence intervals
- Tail behaviour estimates

Operating without uncertainty leads to mis-budgeting, over-commitment and revenue volatility.

7. No Reliability or Stability Scoring

Executives need to know:

- How confident is the model this week?
- What changed?
- Is the prediction in or out-of-regime?

Legacy models cannot self-evaluate or quantify reliability.

How Our Approach Solves These Problems

Our forecasting engine integrates a set of modern techniques specifically chosen to overcome the limitations of existing practice.

1. Cold-Start Capability (Prior Curve + Residual Learning + XSTL)

The system does not rely on historical sales for the new title.

Instead it uses:

- Deterministic behavioural priors
- Cross-sectional similarity (XSTL)
- Feature-driven residual learning

This allows accurate forecasts even when no prior sales exist.

2. Realistic, Behaviourally Correct Priors

The prior curve encodes:

- Correct decay rates
- Dev-type differences
- Seasonality
- DLC/discount uplift windows
- Indie late-tail flattening

This ensures the model always begins with realistic, commercially grounded behaviour, not a blank slate.

3. Residual Learning Provides Stability and Interpretability

XGBoost learns only deviations from expected behaviour.

This solves:

- Scale issues
- Overfitting
- Variance between AAA/AA/Indie
- Lack of explainability

The XGB uplift is easy to visualise, making the model transparent.

4. Robust Decay Modelling with Safety Layers

The engine applies:

- Promo-safe smoothing
- Echo logic
- Small-prior protection
- Reliability-weighted blending

This prevents the unnatural spikes that plague pure ML models and curve-fitting systems.

5. Cross-Sectional Transfer Learning (XSTL)

XSTL improves predictions by comparing a new title to:

- All historical weeks
- Dev-type patterns
- Franchise patterns

It enables the model to borrow strength from structurally similar titles, lifting cold-start accuracy without overfitting.

6. Built-in Uncertainty Distribution (P10–P90)

Every forecast includes:

- A downside case
- A base case
- An upside case
- Weekly spread as a risk index

This provides a “hedge-fund style” view of forecast volatility, something traditional models lack entirely.

7. Reliability and Regime Scoring

The forecast adapts based on:

- Feature drift
- Regime similarity
- XSTL similarity

High reliability → XGB takes ~90% weight

Low reliability → prior dominates

This prevents model overconfidence and explains why predictions are changing.

Summary: A Step Change Over Industry Practice

Traditional models cannot:

- Handle cold starts
- Quantify uncertainty
- Diagnose model behaviour
- Integrate cross-title structure
- Manage promo/DLC volatility
- Provide stable long-tail curves

Our engine solves all of these using a modern, multi-layer, explainable architecture that combines the strengths of:

- Behavioural economics
- Time-series modelling
- Cross-sectional ML
- Risk-modelling from quantitative finance

The result is a highly stable, transparent, and commercially actionable forecasting system, materially more advanced than the tools used by most publishers today.

V. Reliability + Uncertainty

Forecasts are most useful when decision-makers understand not only what the model predicts, but how confident it is and how much risk surrounds each prediction.

This engine includes a full reliability and uncertainty framework, mirroring techniques used in quantitative finance to measure model stability, drift, and prediction volatility.

The system outputs two core diagnostics:

1. **Forecast Reliability** - how stable and trustworthy the model is at each week.
2. **Forecast Uncertainty** - how wide the upside/downside band is around the prediction.

Together, these provide a clear measure of prediction strength, risk, and volatility across the entire first year of a game's lifecycle.

Reliability Framework

Every weekly prediction is accompanied by three internal confidence scores:

1. Regime Confidence

Measures whether the new title is operating in a familiar market regime, based on:

- Marketing intensity
- Competitor pressure
- Price behaviour
- Seasonal factors

Titles whose conditions resemble known historical patterns receive higher regime confidence.

2. Drift Score

Quantifies how far the new title's features drift from the historical distribution of:

- Marketing
- Competitor intensity
- Price
- Seasonal behaviour
- Structural attributes

High drift → lower trust in pure ML predictions → heavier reliance on the prior curve.

3. XSTL Similarity

The cross-sectional distance metric measuring how similar the new title is to the entire historical dataset, using:

- Dev-type behaviour
- Franchise style
- Week-level cross-sectional sales patterns
- Global demand structure

High XSTL similarity means the model has strong historical anchors for this game.

Composite Reliability Score

These three components combine geometrically:

$$\text{Reliability} = \left(R_{\text{regime}} \times R_{\text{drift}} \times R_{\text{xstl}}\right)^{1/3}$$

This provides a smooth 0–1 score.

- 0.90+ → exceptionally strong signal
- 0.60–0.85 → healthy, stable regime
- 0.30–0.60 → caution, model leans toward prior
- <0.30 → highly uncertain, prior curve dominates

This reliability score is then used to weight the blend:

$$\text{XGB Weight} = \max\left(0.30, \text{Reliability(smoothed)}\right)$$

Meaning:

- The model cannot go below 30% XGB influence
- It can rise to ~90% when stable
- In low-confidence weeks, the prior curve stabilises the predictions

This prevents black-box behaviour and ensures smooth commercial curves even in noisy periods.

Uncertainty Framework

1. P10-P90 band

Each prediction includes a symmetric distribution band:

- P10: conservative, low-case expectation
- P90: optimistic, high-case expectation
- Spread (P90–P10): the weekly uncertainty range

This allows clear scenario planning:

- Upside (potential breakout weeks)
- Downside (risk of soft performance)
- Expected case (final blended forecast)

The distribution is calibrated historically and scales with:

- Marketing pressure
- Competitor volatility
- Decay-phase uncertainty
- Long-tail noise

2. Forecast Uncertainty Index

The spread between P10 and P90 becomes a direct weekly risk metric:

$$\text{Forecast Uncertainty Index} = \text{P90} - \text{P10}$$

Interpretation:

- High spread → volatile weeks (launch, DLC, deep discounts, competitive spikes)
- Medium spread → mid-tail stabilisation
- Low spread → long-tail predictable decay

This index makes forecast volatility measurable and intuitive.

Why This Matters Commercially

Reliability and uncertainty transform forecasting from a single guess into a risk-aware planning tool:

- Marketing can adjust spend based on confidence levels.
- Finance can plan around downside risk.
- Production and operations can time DLC and price drops strategically.

VI. Example Forecast Walkthrough (NeonRift - AAA Demo)

To illustrate how the engine operates in practice, we walk through the full forecasting pipeline for NeonRift, a flagship AAA release. This example demonstrates how the model moves from a structural prior, through residual learning and reliability weighting, to a final, risk-aware sales forecast for the full first year.

---

1. Behavioural Prior Curve (Baseline Structural Prediction)

The engine begins with the deterministic behavioural template.

NeonRift's real prior data:

- Peak parameter: 1,450,000
- Decay rate (k): 0.087
- Franchise strength: 0.88
- Dev-type: AAA
- Seasonal factor on launch: 1.448
- Launch week sales: 1,450,000 units

The prior provides a stable launch trajectory from:

$$Sales(t) = Peak \times e^{-kt}$$

Producing:

- Week 0 prior: 1,450,000 units
- Week 1 prior: 1,328,617 units
- Week 2 prior: 1,217,395 units
- Week 4 prior: 1,022,104 units

2. XGB Residual Prediction

Next, the model uses log-residual learning to capture behaviour the prior cannot model:

- Marketing effects
- Competitor pressure
- DLC/discount uplifts
- Seasonal fluctuations
- Cross-title interaction patterns

The model learns:

$$\text{Residual}(t) = \log(1 + \text{Sales}(t)) - \log(1 + \text{Prior}(t))$$

This residual is predicted using:

- Selected pruned features
- Time features
- XSTL cross-sectional features
- Marketing, competition, hype score
- Dev-type and franchise-normalised signals

Actual model residuals for NeonRift:

- Week 0 residual: +115,476.64
- Week 1 residual: +114,254.06
- Week 2 residual: +94,670.98
- Week 3 residual: +106,012.31
- Week 4 residual: +46,286.18

Predicted XGB residuals are then added to the prior forecast figures. The result is the XGB raw prediction.

XGB raw predictions:

- Week 0 raw: 1,565,476.64
- Week 1 raw: 1,442,871.06
- Week 2 raw: 1,312,066.25
- Week 3 raw: 1,221,496.49
- Week 4 raw: 1,068,390.49

These incorporate marketing, competitor intensity, seasonal interactions, and XSTL-based similarity.

3. Reliability-Weighted Blending

For each week, the engine computes:

- Regime Confidence
- Drift Score
- XSTL Similarity

These feed into a composite reliability that determines how much influence XGB should have.

The final blend is calculated using:

$$\text{Blended} = w_{\text{XGB}} \times \text{XGB} + (1 - w_{\text{XGB}}) \times \text{Prior}$$

With:

- minimum XGB weight = 0.30
- maximum ≈ 0.90 (when reliability is high)

NeonRift's reliability components:

Week 0:

- Regime Confidence: 0.2259
- Drift Score: 0.02150
- XSTL Similarity: 0.06380
- Composite Reliability: 0.06766
- XGB weight: 0.30 (minimum cap)

Week 7 (where reliability rises):

- Regime Confidence: 0.54135
- Drift Score: 0.03710
- XSTL Similarity: 0.09605
- Composite Reliability: 0.12449
- XGB weight: 0.32407

Across the year:

- XGB weight ranges from 0.30 → 0.59
- Prior stabilises low-confidence periods
- XGB dominates high-information weeks

4. Promo and DLC Smoothing

NeonRift includes a realistic promotional schedule, with DLC releases in Weeks 20, 32, and 44 and discounts in Weeks 26 and 40.

During these event weeks, the model automatically applies:

- Uplift adjustments based on DLC and discount depth
- Echo smoothing across the following 1–2 weeks
- Safety caps to prevent unrealistic spikes or collapses

These safety measures prevent the instability commonly seen in pure ML models and ensure smooth, credible mid-tail and long-tail behaviour aligned with real-world promotional patterns.

5. Final Blended Forecast

The **final blended forecast** represents the engine's best estimate of weekly sales, combining the stability of the prior curve with the responsiveness of the XGB residuals and the reliability-weighted blend. The output is structurally realistic yet dynamically adjusted for:

- Competitor intensity
- Seasonal peaks
- Marketing momentum
- DLC and discount uplifts

This blended curve is the primary forecast used by commercial teams for:

- Budgeting and revenue planning
- Supply chain and inventory forecasting
- Pricing and discount strategy
- Post-launch optimisation and DLC scheduling

NeonRift Final Blended Values (Weeks 0–9):

Week	0	1	2	3	4	5	6	7	8	9
Final Blend	1,484,642	1,362,893	1,245,796	1,147,287	1,035,990	943,259	842,645	781,019	722,754	654,866

These values come directly from the model runner's output. They demonstrate a smooth, credible decay trajectory for a AAA title with strong launch momentum and realistic mid-tail behaviour.

6. P10–P90 Distribution (Uncertainty Range)

To capture forecast risk and volatility, each weekly prediction includes an uncertainty distribution consisting of a downside case (P10), an upside case (P90), and the spread between them. The system applies a simple, robust rule:

- P10 = 85% of the blended forecast
- P90 = 115% of the blended forecast
- Spread = P90 – P10

This produces a clear and intuitive uncertainty range that scales with the sales level and responds to promotional events, DLC, and competitive pressure.

NeonRift Uncertainty Examples (Weeks 0–2):

Week	P10	P90	Spread
0	1,261,946	1,707,339	445,393
1	1,158,459	1,567,327	408,868
2	1,058,927	1,432,666	373,739

7. Marketing Uplift

The model generates two key signals that quantify the commercial impact of marketing activity:

- Baseline with no marketing - the expected sales trajectory if the game had zero marketing influence
- Marketing uplift - the incremental units attributable to marketing each week

These values allow teams to isolate the real contribution of pre-launch and post-launch campaigns, enabling clearer ROI analysis and more informed budget decisions.

Week	Baseline No Marketing	Uplift
0	965,017.94	519,625.05
1	885,880.60	477,012.63
2	835,008.37	410,788.20
3	779,605.30	367,682.57

8. Cumulative Forecast & Lifecycle Segmentation

Using real blended numbers:

- Launch (0–4 weeks): ~6.2M cumulative
- Mid-tail (5–20 weeks): ~10–14M cumulative
- Long-tail (21–52 weeks): ~17.6M total annual sales

Summary: Why the Walkthrough Matters

This example demonstrates the full behaviour of your engine:

- Prior curve carries structural decay
- Residual XGB learns deviations
- Reliability controls week-to-week weighting
- DLC/discount smoothing ensures stability
- P10–P90 bands quantify risk
- Marketing uplift becomes measurable ROI
- Cumulative lifecycle analysis supports commercial planning

VII. Power BI Visualisations

The forecasting engine is designed not only to generate accurate predictions, but also to present them in a visual, intuitive, and executive-friendly format. To achieve this, we provide a Power BI dashboard that transforms the model outputs into clear, actionable insights for commercial, marketing, finance, and production teams.

The dashboards are structured to communicate the full lifecycle of a title, from launch momentum to long-tail decay, along with the key drivers of performance, risk indicators, and promotional effects. Each visual reflects a direct component of the forecasting architecture, ensuring that decision-makers can see why the model behaves as it does.

All power BI images included within an accompanying PDF.

---

1. KPI Header – Title Metadata & Core Commercial Indicators

The dashboard begins with a clear KPI strip summarising attributes and performance metrics for the selected title. This section includes:

- Title selector (NeonRift, Ashbound, Pulsebreak)
- Dev Type
- Franchise and Franchise Strength
- Decay Class and Decay Rate
- Platform
- Region

Followed by the key forecast indicators:

- Total Annual Sales
- Marketing Uplift %
- Week-1 Sales
- Sales Half-Life Week
- Sales Plateau Week
- Forecast Uncertainty (spread)
- SMAPE (model error)

2. Baseline Curve, Blend Curve, and XGB Uplift

This line-plus-area chart shows how the model's three core components interact:

- Baseline Curve - the pure behavioural prior (no ML, no promotions)
- Final Blended Forecast - prior × XGB × reliability
- XGB Uplift - the residual correction learned by the model

The uplift curve visualises how much incremental value is added by marketing, competition, DLC/discounts and cross-sectional signals.

3. Cumulative Forecast with Lifecycle Segmentation

This visual shows cumulative annual sales with vertical segmentation lines for:

- Launch Phase (Weeks 0–4)
- Mid-Tail Phase (Weeks 5–20)
- Long-Tail Phase (Weeks 21–52)

The cumulative curve helps teams understand:

- Total expected annual sales
- How much demand is captured early
- Whether the title is front-loaded or tail-weighted
- Lifecycle shape vs. benchmarks

This is the core commercial planning visual.

4. Forecast Uncertainty Bands (P10–P90)

This chart overlays:

- Final Forecast (P50)
- Upper Band (P90)
- Lower Band (P10)

The shaded difference represents the Forecast Uncertainty Index.

Executives can quickly see:

- Where uncertainty is highest
- How upside/downside evolves over the lifecycle
- Whether risks cluster around launch, DLC weeks, or competitive pressure

5. Promotion Impact Timeline

This visual highlights how promotions shape the mid-tail:

- Green bars = Discount weeks
- Gold bars = DLC weeks
- Blue line = Final Forecast

The alignment between promo weeks and uplift allows teams to answer:

- Are promotions scheduled at commercially optimal times?
- How much uplift is attributable to DLC vs discounts?
- Are post-promo echoes behaving realistically?

This is a powerful tool for pricing and release-management teams.

6. Marketing ROI and No-Marketing Baseline Comparison

This dual-axis chart compares:

- Marketing ROI (bars) — the incremental units driven by marketing
- Final Forecast (blue line)
- Baseline No-Marketing Curve (orange line)

This reveals:

- How much marketing contributes week-by-week
- How rapidly marketing effects decay
- Whether spend intensity aligns with commercial return
- The counterfactual trajectory without marketing

A vital tool for budgeting and post-campaign analysis.

7. Model Reliability Over Time

This visual plots:

- Model Reliability Score (line)
- DLC and Discount markers (gold/green)

It shows how reliability evolves as the game progresses and how reliability interacts with:

- Feature drift
- Regime stability
- Cross-sectional similarity
- Promo and DLC windows

This chart answers:

- How much should we trust the forecast this week?
- Is reliability rising or falling?
- Are promotions occurring during stable or unstable periods?

Summary: Why Power BI Integration Matters

The Power BI suite converts the forecasting engine from a technical model into a commercial decision platform.

It ensures that stakeholders can see:

- How the forecast evolves
- What drivers are influencing performance
- Where uncertainty is high
- How promotions shape mid-tail behaviour
- What ROI marketing and pricing decisions generate

By combining quantitative forecasting with clear, intuitive visualisation, the system becomes immediately actionable across production, marketing, finance, and executive teams.

VIII. Limitations

No forecasting system can fully eliminate uncertainty, and responsible analytics requires transparency about where models may have constraints. The proposed engine is robust, explainable, and commercially aligned but it does have limitations that should be clearly understood at the outset.

---

1. Synthetic Dataset Used for Demonstration

The current model has been demonstrated on a synthetic training universe, designed to mimic realistic AAA/AA/Indie lifecycle behaviour. This allows stakeholders to see the full end-to-end workflow without requiring proprietary publisher data.

However:

- Real publisher data will differ in structure and nuance
- Franchise behaviours and regional patterns may need recalibration
- Feature distributions will shift during onboarding

A short calibration phase is required when migrating from synthetic data to production use.

2. The Model Cannot Anticipate Black-Swan Events

The engine can model expected volatility, DLC/discount windows, and competitor pressure, but it cannot forecast unpredictable shocks, such as:

- Viral social spikes
- Unexpected platform featuring
- Severe review events
- Sudden franchise revivals
- Macro shocks affecting spending

These events can materially alter sales curves. The system can respond quickly once their effects appear in the data, but cannot predict them beforehand

3. Requires Ongoing, High-Quality Data Streams

Accuracy depends on a stable pipeline of:

- Weekly sales
- Marketing spend or proxy index
- Discount flags and pricing
- DLC release information
- Competitor activity
- Seasonal calendar effects

If any of these inputs degrade (e.g., missing marketing signals, inconsistent competitor data), model performance will naturally weaken.

4. Marketing Index Is a Proxy and Needs Calibration

The model uses a marketing index that captures pre and post launch momentum.

For synthetic data, this behaves realistically but for production:

- The proxy must map to real spend patterns
- Campaigns vary significantly across publishers
- Different channels (digital, social, paid media) decay differently

The index must be calibrated against the client's own marketing workflows to maximise accuracy.

5. Some Behavioural Parameters Require Tuning

Certain parameters are designed to be flexible so that publishers can align the engine with their commercial philosophy:

- Peak sales priors
- Decay rate expectations
- DLC uplift shapes
- Discount response curves
- Franchise multipliers
- XSTL similarity thresholds

These defaults are grounded in industry patterns but may require adjustment to reflect specific franchises, genres, markets, or historical behaviour.

6. Uncertainty Bands Use a Simple, Transparent Structure

P10–P90 bands scale as fixed multipliers around the blended forecast.

This keeps uncertainty intuitive, but:

- Does not model asymmetric risks
- Does not account for volatility clustering at extreme outliers
- May understate uncertainty during abnormal market conditions

A more advanced probabilistic model can be added if required.

7. Reliability Score Is Diagnostic, Not Absolute

The composite reliability score (Regime + Drift + XSTL):

- Accurately signals model confidence
- Improves blending stability
- Highlights risky periods

But it remains a diagnostic measure rather than a statistical guarantee. It should guide interpretation, not serve as a definitive indicator of truth.

Summary

These limitations are normal for any modern forecasting system and are mostly related to data quality, external shocks, and calibration requirements. None of them undermine the engine's core strengths but acknowledging them allows for realistic expectations and sets the foundation for continuous improvement as more publisher data becomes available.

IX. Future Improvements

The proposed forecasting engine is a strong foundation for modern, data-driven commercial planning. However, as with any advanced analytics system, its value can be expanded further through an iterative roadmap. Below are the key enhancements that can be added as more real-world data becomes available and the system moves toward full production deployment.

---

1. Integration of Real-Market Publisher Dataset

The next major step is aligning the engine with real sales, marketing, and competitor data.

This will enable:

- Calibration of franchise- and genre-specific priors
- More accurate marketing uplift curves
- Real discount elasticity modelling
- Fine-tuned decay parameters per platform/region

This integration significantly increases forecast precision and reduces reliance on behavioural templates.

2. Bayesian Hierarchical Curve Priors

Introduce a Bayesian framework to refine priors using:

- Franchise-level historical patterns
- Similar-genre decay curves
- Platform-specific behaviour
- Cross-region variation

This allows priors to borrow strength from related titles, producing more stable forecasts and tighter uncertainty bands for cold starts.

3. LLM Metadata Extraction from Store Pages

Use large language models to automatically extract:

- Genre nuances
- Feature sets (multiplayer, open-world, roguelike, etc.)
- Audience descriptors
- Expected engagement depth
- Review sentiment
- Marketing descriptors
- Art style and tone cues

These metadata features become powerful inputs for XSTL and residual learning, improving predictions for new IPs or experimental titles.

4. Enhanced Competitor Modelling

Upgrade competitor intensity beyond the synthetic AR-smoothed variable by incorporating:

- Live store charts
- Wishlists and pre-orders
- Genre clash detection
- Franchise stacking (e.g., Pokémon vs Mario proximity)
- Cross-platform timing conflicts

This produces a dynamic competitor-pressure index grounded in real market data.

5. Automated Scenario Planning & Simulation Tools

Extend uncertainty modelling to generate dynamic scenarios such as:

- What if marketing spend increases by 20%?
- What if discounting starts at Week 10 instead of 20?
- What if a major competitor slips release date?

6. Advanced Probabilistic Uncertainty Modelling

Replace simple P10–P90 scaling with a full probabilistic model using:

- Quantile regression
- Bayesian posterior predictive simulations
- Volatility-scaled uncertainty
- Asymmetric tail modelling

This produces sharper, more accurate confidence intervals for decision-makers.

7. Automated Weekly Reforecasting Engine

Implement a live module that:

- Ingests latest sales/marketing data
- Re-evaluates drift and regime
- Updates cross-sectional similarity
- Re-blends the forecast
- Publishes new dashboards automatically

8. Integration with Internal BI, CRM, or Data Warehouses

Once in production, the engine can be connected to:

- Snowflake / BigQuery / Redshift
- Internal BI dashboards
- Marketing effectiveness tools
- Supply chain planning systems

This ensures the forecasting workflow is embedded directly into the publisher's commercial processes.

Summary

This roadmap demonstrates how the engine can evolve into a full enterprise platform for:

- Forecasting
- Scenario modelling
- Marketing ROI optimisation
- Cross-region intelligence

Each step deepens accuracy, automation, and strategic value, aligning the system with how modern publishers operate in increasingly complex markets.

X. Appendices

The appendices provide supporting technical detail for those who wish to understand the mechanics behind the forecasting engine. These materials are optional for executive readers but valuable for data, analytics, and engineering stakeholders involved in implementation.

---

1. GitHub Repository & Reproducibility

This proposal is accompanied by a complete GitHub repository containing all components required to reproduce the forecasting engine, including:

Your repository contains the following core folders and files:

src/

The source code for the forecasting engine.

data/

Holds example datasets, synthetic game-sales data, and new-game input files.

dashboard/

Contains Power BI Model – Game Forecaster.pbix and accompanying images.

docs/

Includes project documentation such as how_to_run.md.

Supporting Files

README.md

A concise introduction to the project, setup instructions, and links to key components.

requirements.txt

The full Python dependency specification for replicating the environment.

CHANGELOG.md

A transparent record of updates, enhancements, and version changes across the project lifecycle.

GitHub:

Github Repository (click to open)

(https://github.com/georgejp144/game-sales-forecasting-engine)

2. Feature Dictionary

A comprehensive catalogue of all features generated and consumed by the model, including:

Behavioural features

- Peak sales param, decay rate k, seasonal factor
- Dev type multipliers (AAA/AA/Indie)

Calendar features

- Week index, release week, holiday proximity
- Black Friday index, Christmas uplift factor

Marketing & promotional features

- Marketing index, post-launch decay
- Discount flag, discount depth
- DLC flag, DLC weeks since

Competition metrics

- Competitor intensity, competitor window score

Rolling sales features (for training universe)

- 4-week, 8-week, 13-week rolling means
- Week-on-week deltas
- Log1p transformations

Cross-sectional (XSTL) signals

- Franchise similarity, dev-type cross-means
- Week-level cross-title stats
- Mahalanobis distance scores

This dictionary allows a smooth transition from synthetic data to real publisher datasets.

3. Technical Pipeline Overview

This appendix summarises the full end-to-end workflow at a technical level:

1. Data ingestion
Training universe, new title metadata, marketing, DLC/discount, calendar inputs.

2. Feature generation
Behavioural templates, engineered signals, rolling windows, XSTL statistics.

3. Validation layer
NaN checks, type validation, anomaly detection, title-level health scoring.

4. Feature pruning
Missingness filter → variance filter → correlation pruning → XGB gain scores → linear regression coefficients → composite quality score.

5. Model training
Prior curve construction, log-residual target, XGBoost training, Bayesian tuning.

6. Prediction pipeline
Prior × promo × seasonal × dev-type effects; residual uplift; reliability-weighted blending.

7. Uncertainty and reliability scoring
P10–P90 bands; regime confidence; drift scoring; XSTL similarity.

8. Output publishing
Weekly forecast table, summary metrics, dashboard-ready layout.

This gives technical readers a clear understanding of how data flows through the system.

4. Model Mathematics

This appendix documents the key mathematical components underpinning the engine.

Behavioural Prior Curve

Extended multiplicatively by seasonal factors, promo/DLC uplifts, and dev-type behaviour.

$$\text{Sales}(t) = \text{Peak} \times e^{-kt}$$

Residual Regression

Residual prediction uses log space:

$$\text{Residual} = \log(1 + \text{Sales}_{\text{actual}}) - \log(1 + \text{Prior})$$

Predicted residuals are added back to the prior and transformed to real space:

$$\text{Sales}_{\text{pred}} = \exp(\log(\text{prior}) + \text{residual}) - 1$$

Cross-Sectional Transfer Learning (XSTL)

Similarity to historical titles is calculated using cross-title and cross-week statistics:

$$\text{XSTL} = e^{-D^2}$$

Where *D* is a Mahalanobis-style distance metric in feature space.

This influences both reliability scoring and residual behaviour.

Reliability Score

Smoothed and used to weight the ensemble blend.

$$\text{Reliability} = \left(R_{\text{regime}} \times R_{\text{drift}} \times R_{\text{xstl}}\right)^{1/3}$$

Uncertainty Distribution

$$P10 = 0.85 \times \text{Blended} \qquad P90 = 1.15 \times \text{Blended}$$

The spread is:

$$\text{Spread} = P90 - P10$$

Summary

The appendices supply the technical depth behind the engine while keeping the main proposal focused on strategy, accuracy, transparency, and commercial value. They ensure that data scientists, engineers, and technical stakeholders can fully understand the mechanics, mathematics, and implementation path of the forecasting system.