CYO Project

Zhang Cheng Hu

December 2025

Executive summary

This project analyzes the Video Game Sales with Ratings dataset from Kaggle to explore the factors influencing worldwide video game sales and to build a predictive model for Global_Sales.

Because sales data is extremely right-skewed, we model the transformed target log1p(Global_Sales) and back-transform predictions using expm1.

Three predictive models are evaluated:

- GLMNET – Regularized linear model (Lasso/Ridge/Elastic Net)
- Random Forest (ranger) – Nonlinear ensemble of decision trees
- XGBoost – Gradient boosting with early stopping

The report includes full methodology, exploratory analysis, modeling details, and performance comparison using RMSE and MAE on the original sales scale.

1. Introduction / Overview

1.1 Dataset Source and Context

The dataset Video Game Sales with Ratings originates from Kaggle (Rushabh Shah, 2017): https://www.kaggle.com/datasets/rush4ratio/video-game-sales-with-ratings/data

The dataset was created by combining:

- A web scrape of VGChartz global sales
- A web scrape of Metacritic critic and user rating data

The dataset contains significant missingness, especially in:

- Critic Score
- Critic Count
- User Score
- User Count

1.2 Variables

The dataset's core variables include:

- Sales Metrics (millions of units): NA, EU, JP, Other, Global
- Game Metadata: Name, Platform, Year_of_Release, Genre, Publisher, Developer
- Ratings & Engagement: Critic Score, Critic Count, User Score, User Count
- Content Rating: ESRB Rating

1.3 Project Goal

The objective is to predict global sales of video games using metadata and ratings.

Why log-transform Global Sales?

Global Sales has a heavy right tail, where a few blockbuster titles dominate. Modeling log1p(Global_Sales):

- reduces skew
- stabilizes variance
- improves model performance
- makes RMSE comparisons more meaningful

Predictions are back-transformed using expm1() for evaluation.

1.4 Modeling Considerations

This dataset contains:

- High-cardinality categorical variables (e.g., Publisher, Developer)
- Missing numeric values (ratings counts and scores)
- Severe class imbalance in sales
- Possible multicollinearity in critic/user metrics

To address these challenges we apply:

- Target encoding for high-cardinality factors
- Train/test leakage prevention
- Cross-validation for hyperparameter tuning
- Three diverse ML models
- RMSE/MAE evaluation on original scale

2. Methods & Analysis

This section details the full workflow used to prepare the data, construct features, train predictive models, and evaluate them.

2.1 Train/Test Split

We use an 80/20 train/test split. Justification:

- More training data helps nonlinear models (RF, XGBoost)
- 20% provides a reliable performance estimate without overfitting to validation folds
- Avoids excessive variance from smaller test sets

2.2 Data Cleaning

The following steps are performed:

- Clean column names with janitor::clean_names()
- Convert numeric fields (critic_score, user_score, etc.) safely using a custom converter
- Drop rows with zero or missing global sales
- Convert categorical variables to factors
- Impute missing numeric values using median imputation
- Convert missing categorical values to "Unknown"

These are standard approaches for noisy web-scraped datasets and prevent model failures due to NA values.

2.3 Feature Engineering

Several new predictors are created:

- release_decade – Groups release years into decades
- age_years – Game age relative to 2016
- platform_avg_sales, publisher_avg_sales – Hierarchical averages
- primary_genre – Extracted from multi-genre field
- genre_count – Number of genres per title
- critic_score_scaled, user_score_scaled – Normalization

These reduce sparsity, incorporate domain knowledge, and improve signal-to-noise ratio.

2.4 Handling High-Cardinality Factors

Variables like Publisher or Developer have hundreds of levels. Random Forest and GLMNET cannot handle such large one-hot encodings.

We apply target encoding, computed on the training set only:

- Each categorical level is replaced with a smoothed version of the mean log_sales
- Smoothing prevents rare categories from producing extreme values

This prevents:

- Overfitting
- Data leakage
- Excessive expansion of model.matrix

2.5 Models Used

GLMNET

- Captures linear relationships with L1/L2 regularization
- Reduces overfitting
- Provides a strong baseline

Random Forest (via ranger)

Selected because:

- ranger is fast and memory-efficient
- Handles nonlinearities and interactions
- Provides variable importance

XGBoost

An advanced gradient boosting algorithm:

- Very strong performance on tabular data
- Uses early stopping
- Handles nonlinear structure and collinearity
- Provides feature gain importance

Full analysis script

1. Distribution: Global Sales

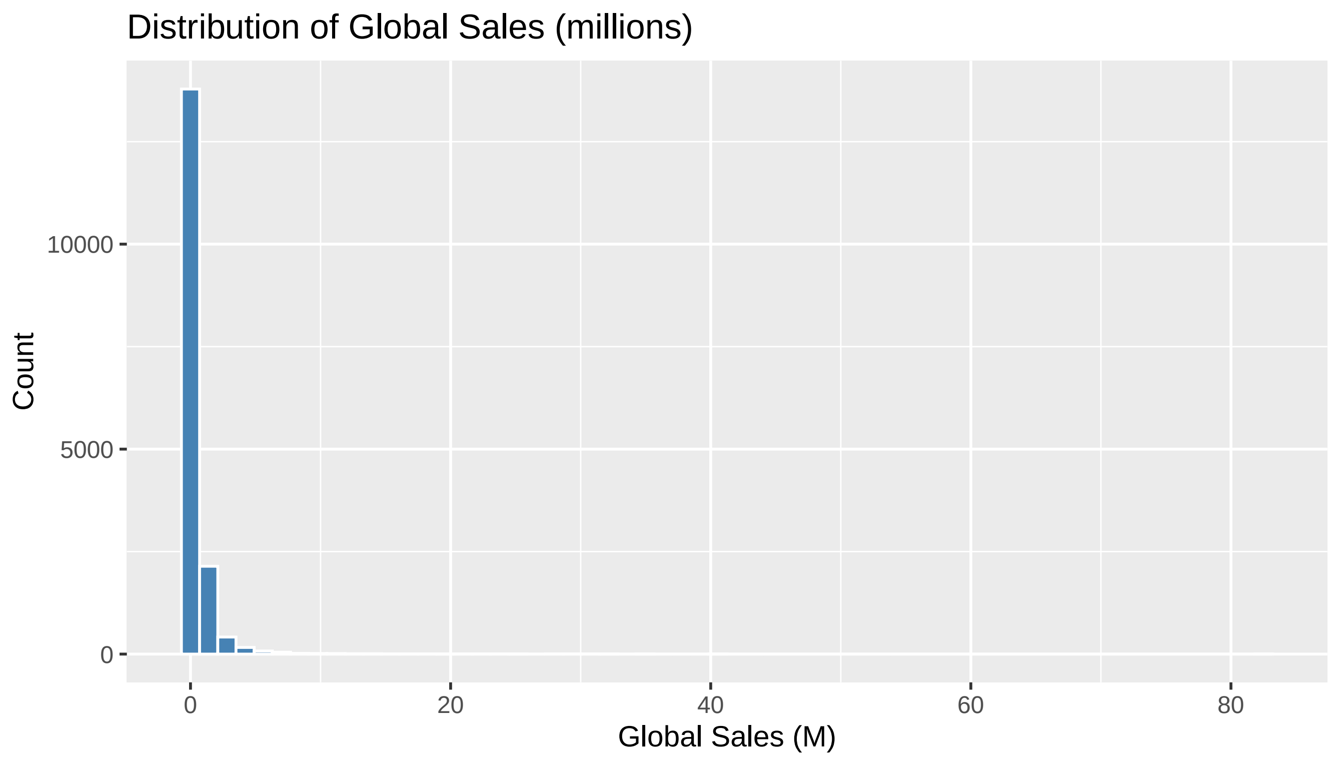

Figure 1: Distribution of global sales

This histogram shows the raw distribution of Global Sales (in millions of copies sold) before any transformation. The pattern is extremely right-skewed: most games sell very little (under 1 million), while a small number of blockbuster titles achieve very high sales.

This heavy skew motivates the use of:

- a log1p transformation (log_sales) to stabilize variance
- models that can handle long-tailed numerical outcomes

The visual confirms why raw sales are unsuitable for direct modeling—they violate linear model assumptions and would dominate error metrics if not transformed.

2. Distribution: log_sales

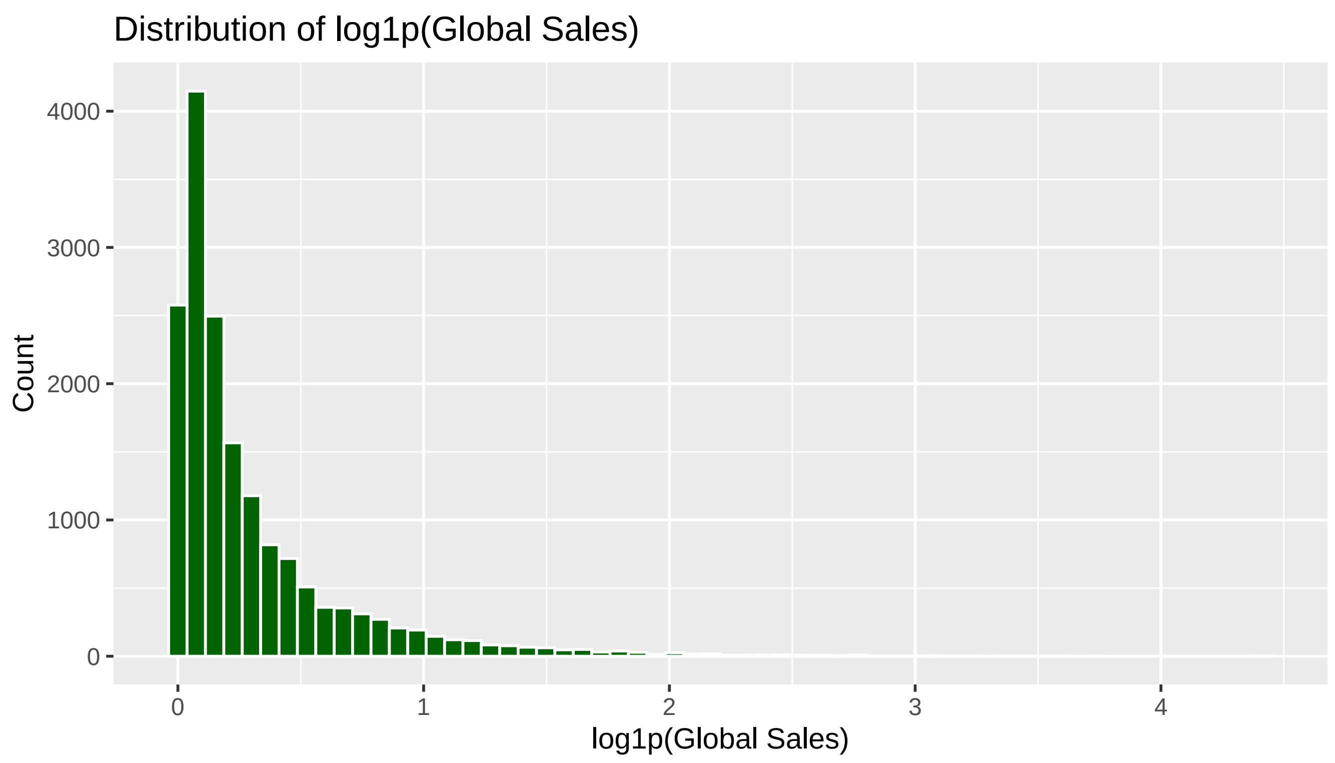

Figure 2: Distribution of log sales

After applying log1p(Global_Sales), the distribution becomes much closer to normal. This shows:

- Variance is more stable
- Extreme outliers are compressed
- Linear models can fit better
- RMSE in log-space becomes more meaningful

This justifies modeling log_sales rather than raw sales.

3. Sales by primary_genre (mean)

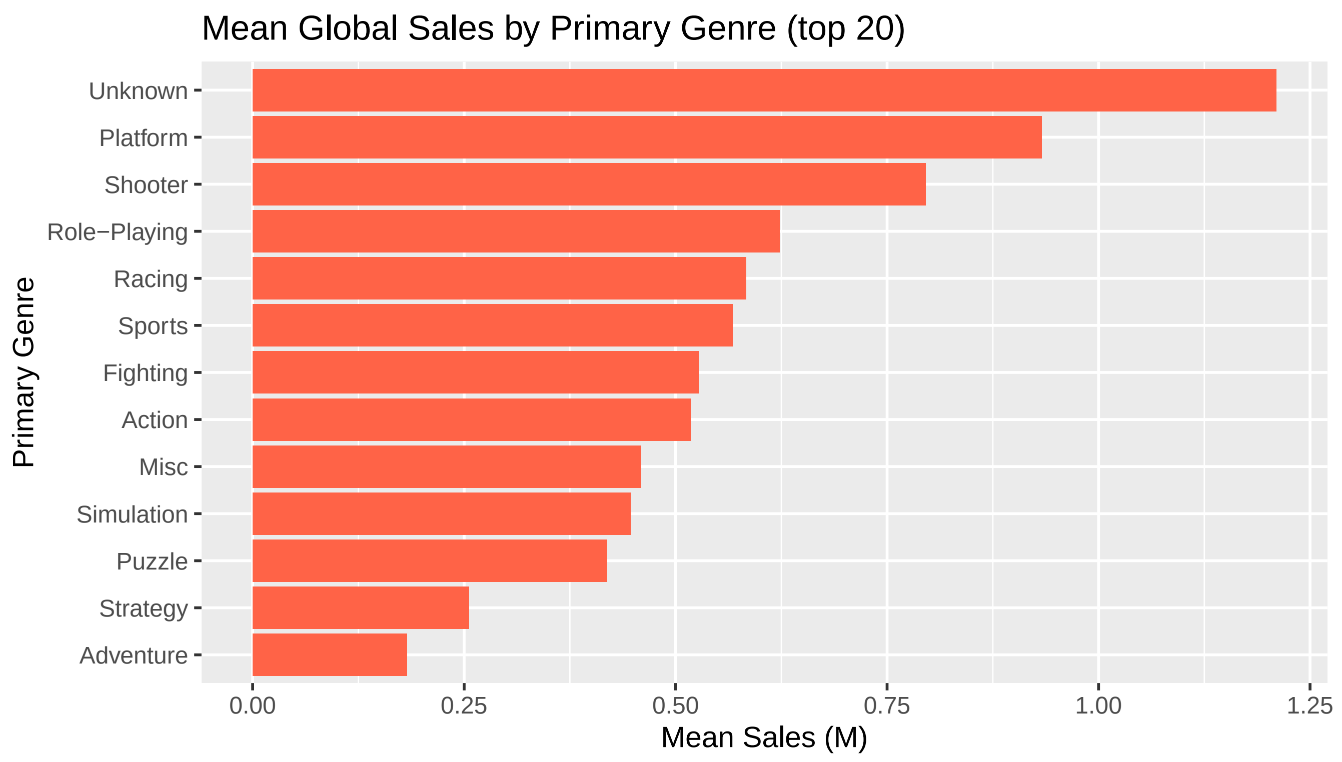

Figure 3: Sales by primary genre

This bar chart shows the top 20 genres ranked by their average global sales. The visualization helps identify genres associated with higher commercial performance.

Key insights typically include:

- Action and Shooter games often dominate because they appeal to a wide audience.
- Niche genres (e.g., Puzzle, Strategy) usually appear lower in the ranking.
- Some genres may have few observations; these can produce noisy means.

This guides feature understanding—genre is a meaningful predictor, but noisy for underrepresented categories.

4. Sales by platform (mean)

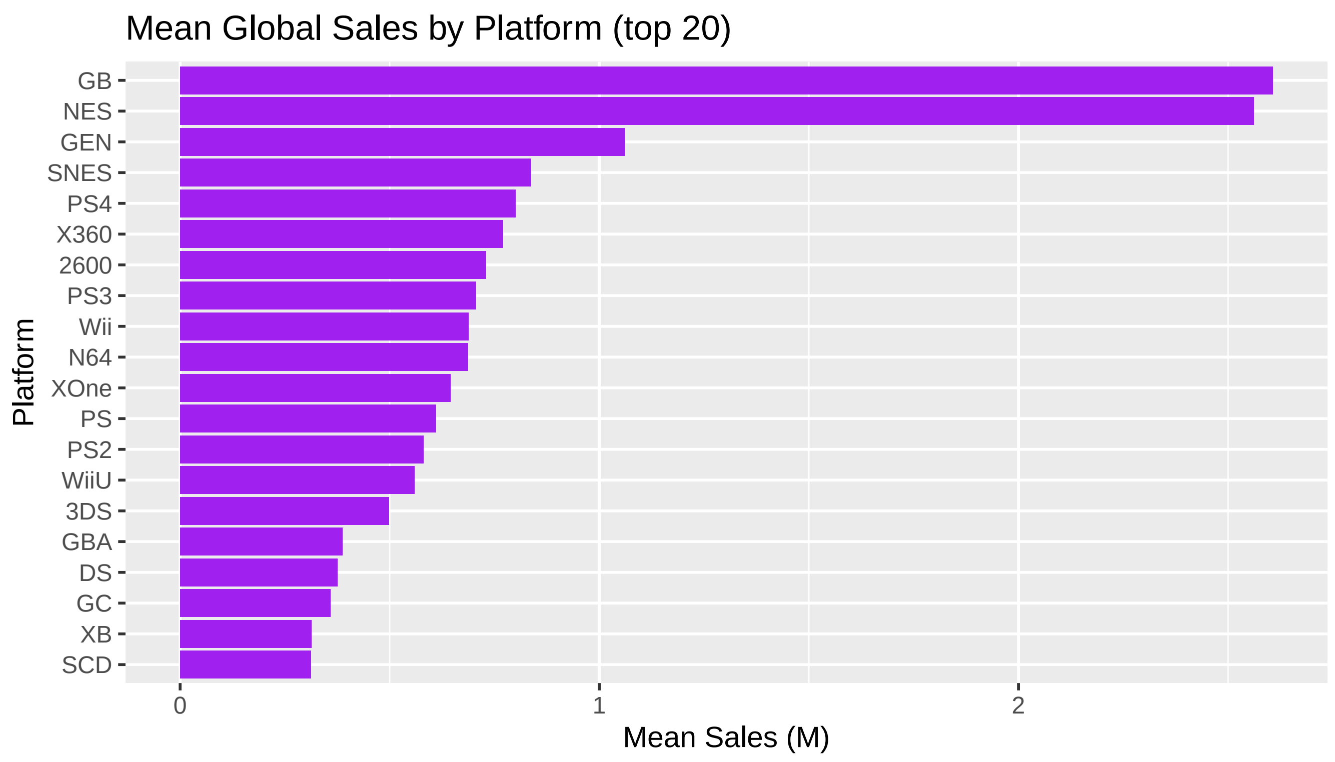

Figure 4: Sales by platform

This plot ranks the top 20 platforms by their mean global sales. High-selling platforms typically include major consoles platforms.

Interpretation:

- Popular consoles with large user bases naturally produce higher-selling games.
- Older or less successful platforms show lower average sales.
- Some platforms are newer or discontinued, affecting their representation.

This plot supports the inclusion of platform as an important categorical predictor in all models.

5. Critic/User score vs log_sales

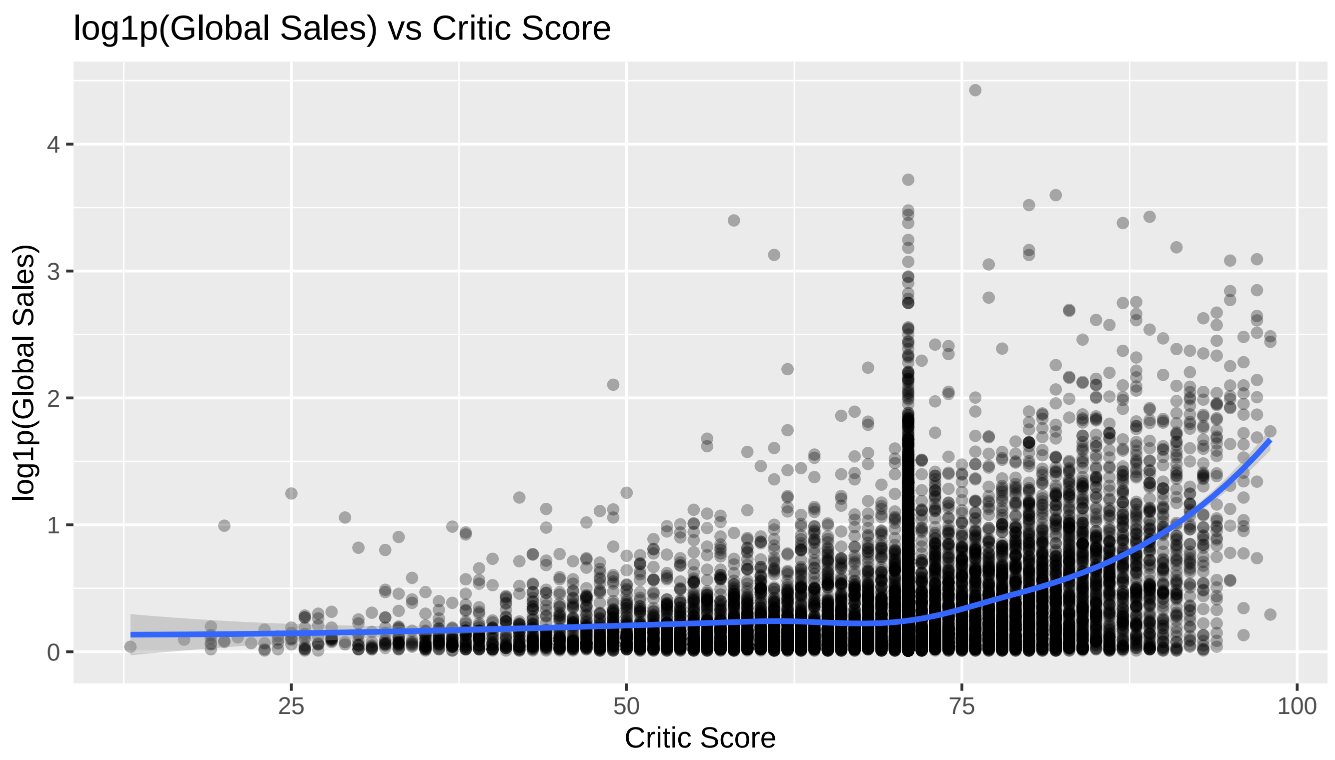

Figure 5: Global sales critic score

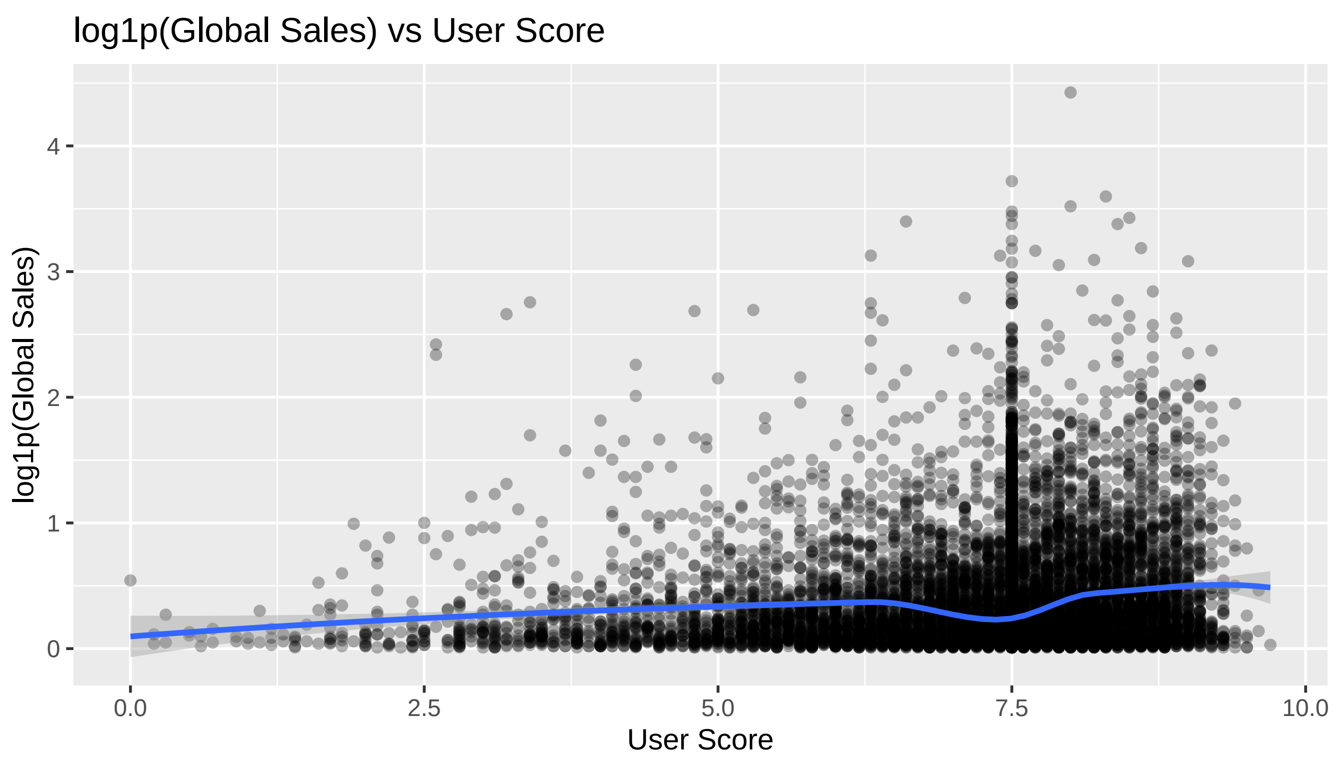

Figure 6: Global sales user score

These scatterplots examine how critic_score and user_score relate to log_sales.

Interpretation:

- The positive trend indicates that games with higher critic reviews tend to achieve higher sales.
- User scores show a similar, although often noisier, pattern.
- The LOESS smoothing curve highlights a general but not perfectly linear relationship.

These visuals support the idea that review metrics contain predictive signal, though with substantial variability.

6. Correlation heatmap for numeric predictors

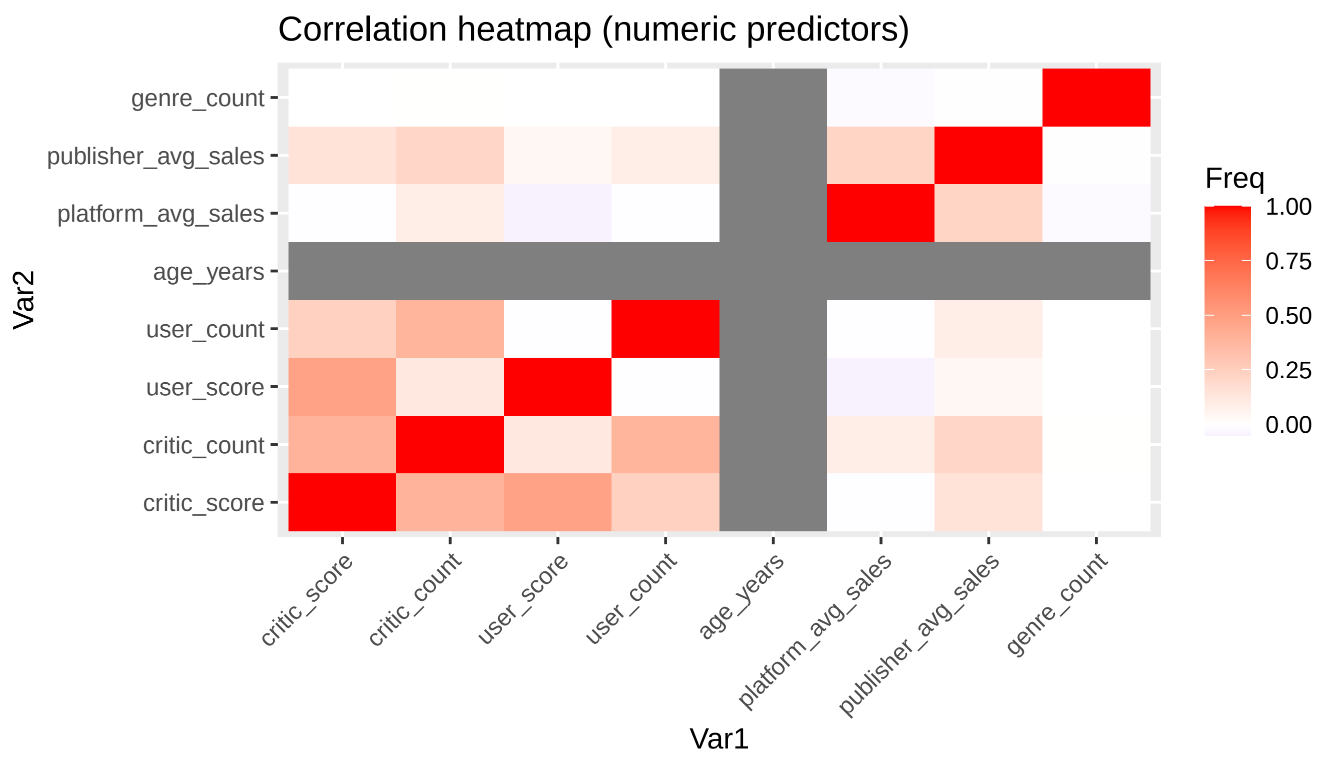

Figure 7: Correlation heatmap for numeric predictors

The heatmap shows correlations among numeric predictors

What it shows:

- Strong positive correlations between metrics that measure similar concepts (e.g., critic_count and critic_score often correlate)
- Weak correlations between sales-related totals and review metrics
- Identification of potential multicollinearity issues for linear models

This guides model design:

- GLMNET regularization helps manage correlated predictors
- Tree-based models can capture non-linear interactions without collinearity concerns

GLMNET via caret (regularized linear) - build model.matrix from combined

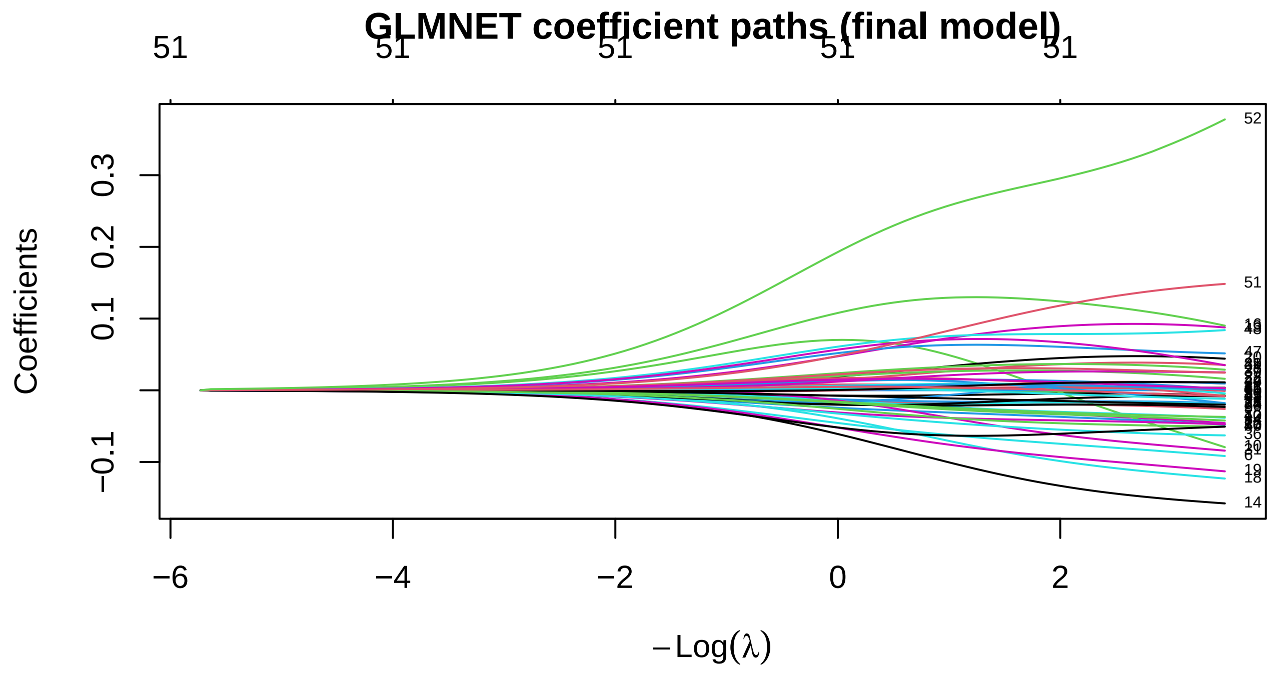

Figure 8: GLMNET via caret

This plot shows how coefficients shrink as lambda increases in the GLMNET model.

Explanation:

- At high lambda, coefficients approach zero → model becomes simple.
- At low lambda, more variables enter the model.
- The path helps visualize regularization strength and model sparsity.

This demonstrates the benefit of GLMNET for high-dimensional or multi-category datasets.

Random Forest via caret

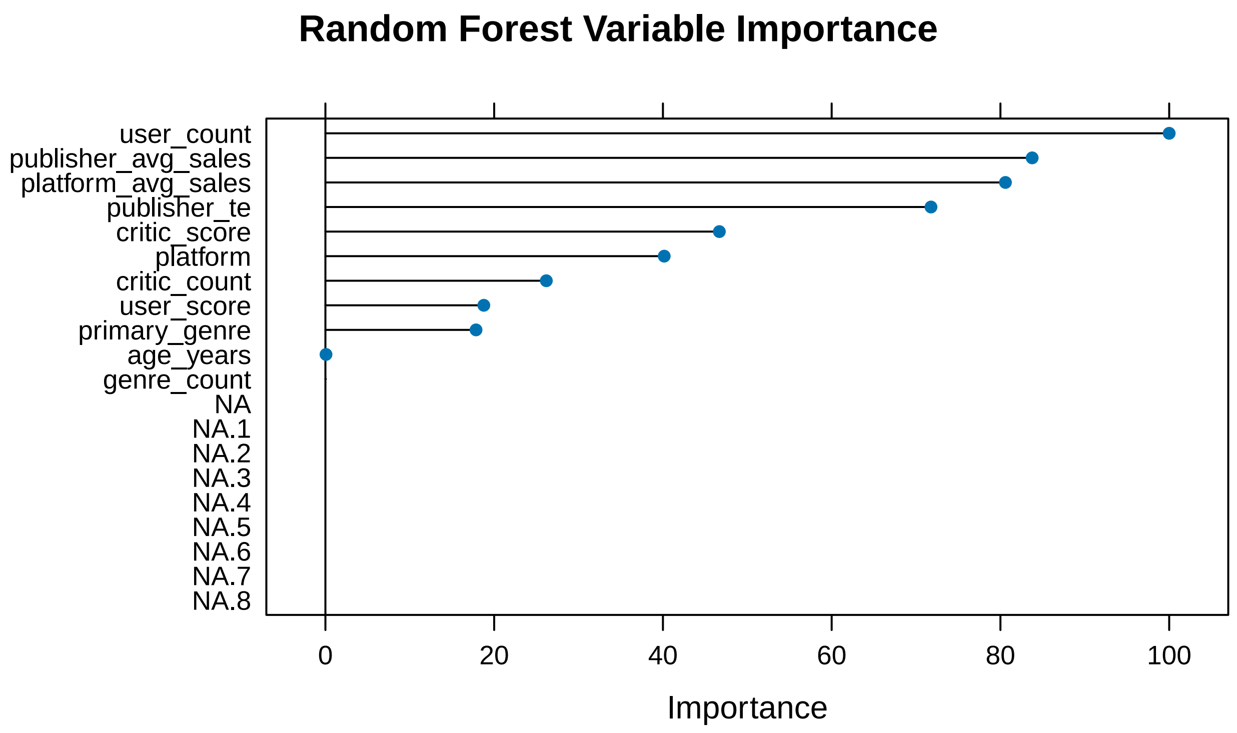

Figure 9: Random Forest via caret

This plot displays the top predictors ranked by their permutation importance in the Random Forest.

Interpretation:

- High-importance variables contribute more to reducing prediction error.
- Typically, critic_score, platform, user_score, and year_of_release appear near the top.
- The importance rankings reflect non-linear interactions not captured by linear models.

This helps explain model behavior and which features are driving prediction performance.

XGBoost (direct xgboost workflow: numeric matrices + training curve)

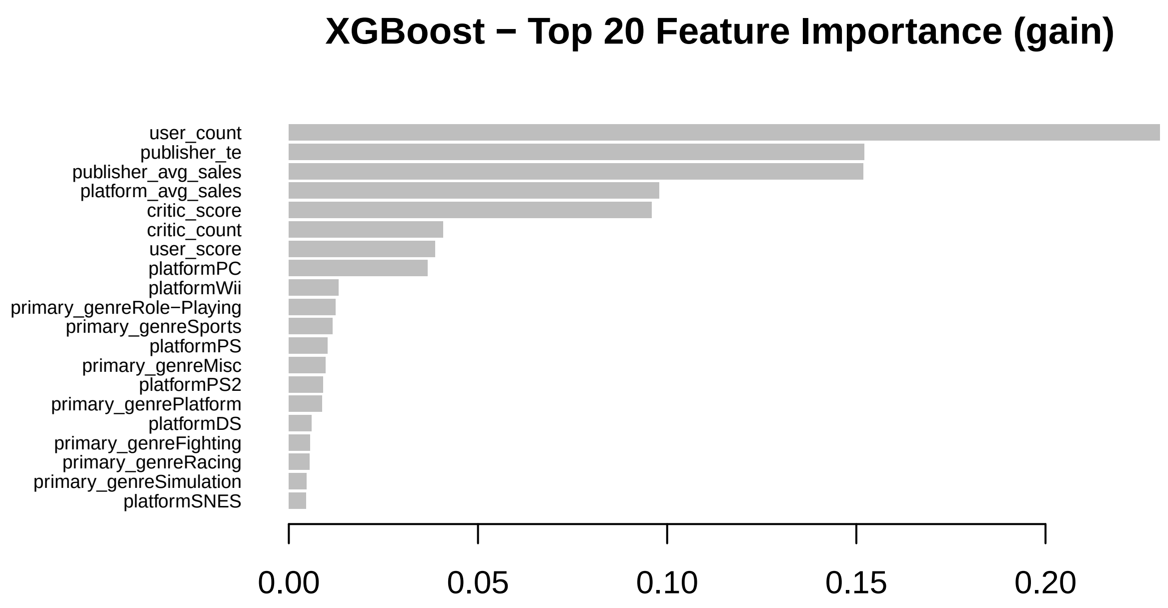

Figure 10: XGBoost gain

This chart ranks features by "gain," which measures how much each feature improves the model's decision splits.

What it shows:

- XGBoost identifies complex combinations of genre, platform, scores, and temporal features.
- The model tends to prioritize features that split the data most effectively early in the tree structure.
- Gain importance provides insight into non-linear relationships.

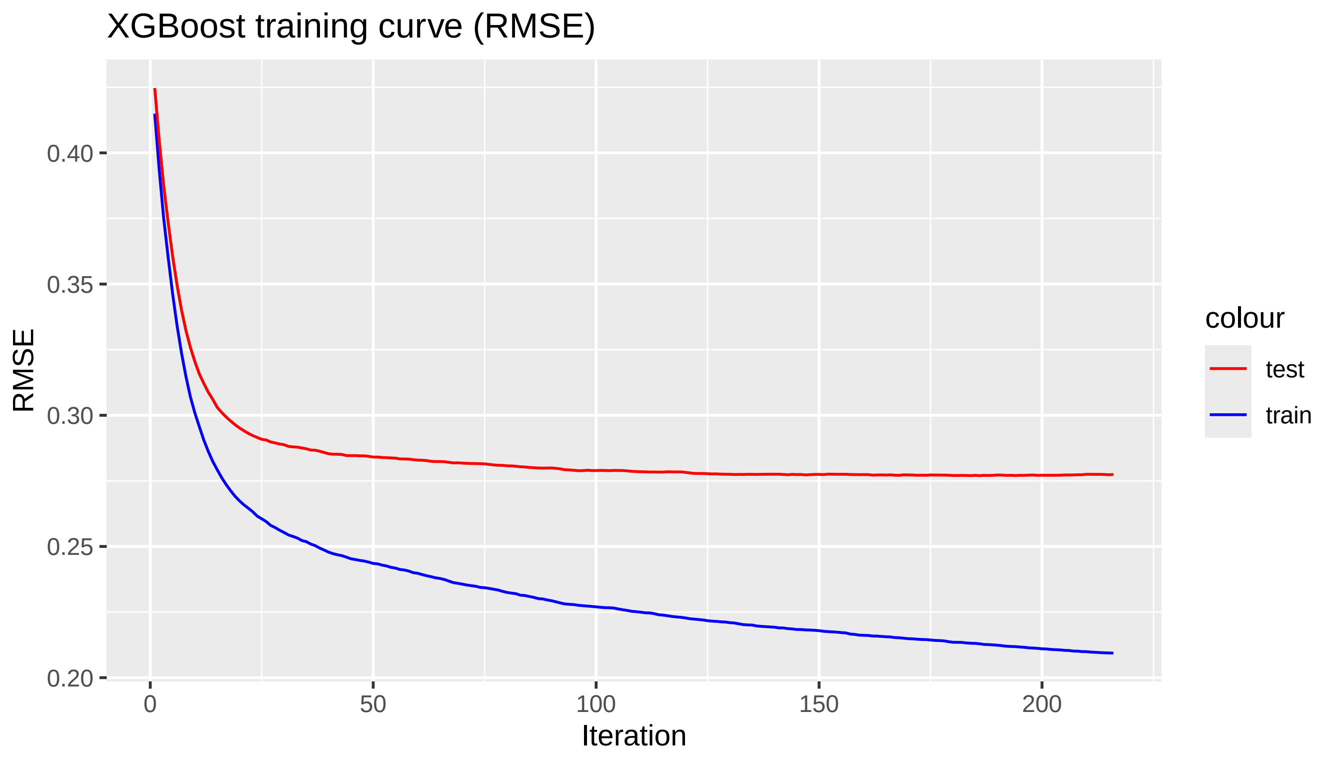

Figure 11: XGBoost RMSE

This plot tracks RMSE over training iterations for both the training and test sets.

Interpretation:

- The model improves over time until early stopping is triggered.
- The best iteration balances bias and variance.
- A large gap between train and test curves would indicate overfitting—early stopping prevents this.

This demonstrates why direct xgb.train() with a watchlist is preferred over caret's old interface.

Model comparison & diagnostics (key plots)

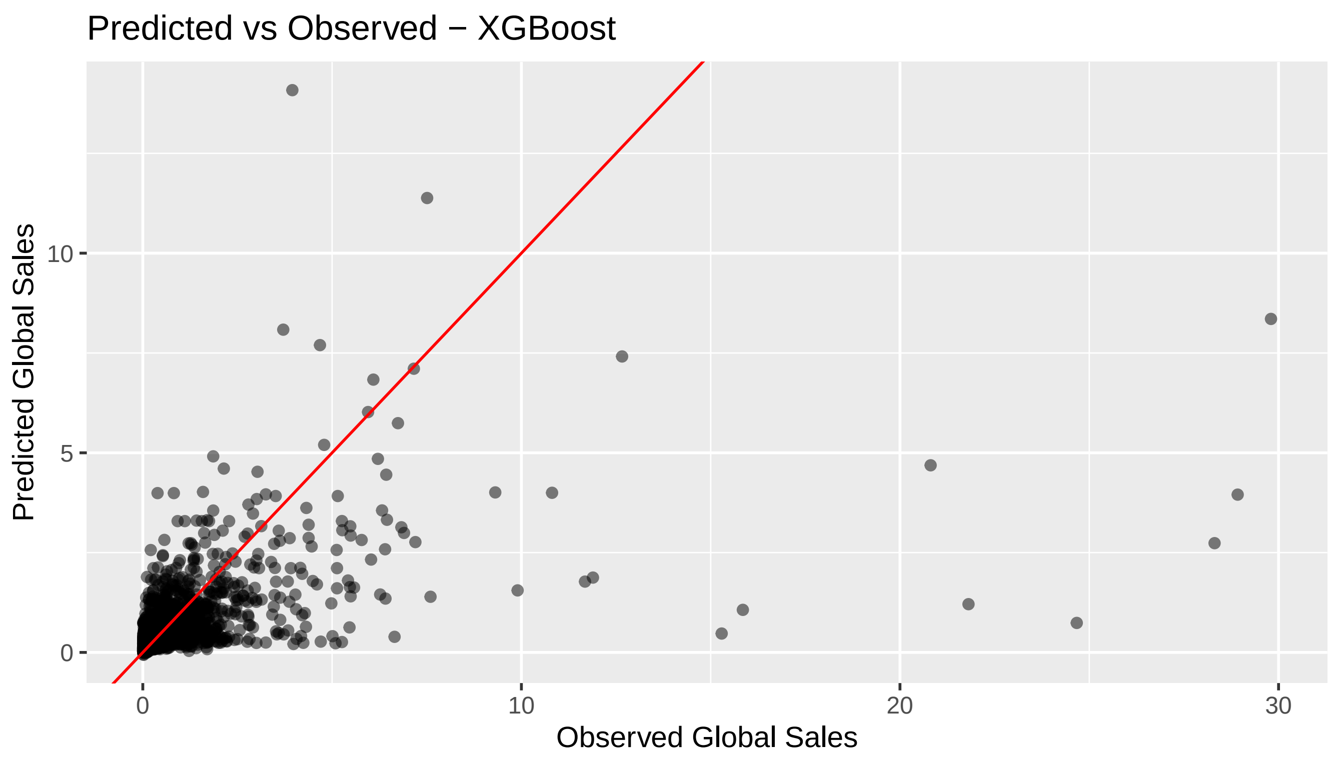

Figure 12: Predicted vs Observed XGBoost

This plot evaluates how closely the model's predictions match the true sales values.

- Points near the red diagonal line indicate accurate predictions.
- XGBoost predictions follow the diagonal reasonably well, especially for mid-range sales.
- Larger deviations occur for very high-selling games, which is expected due to their rarity and unpredictability.

Overall, XGBoost demonstrates good alignment with observed values without major systematic bias.

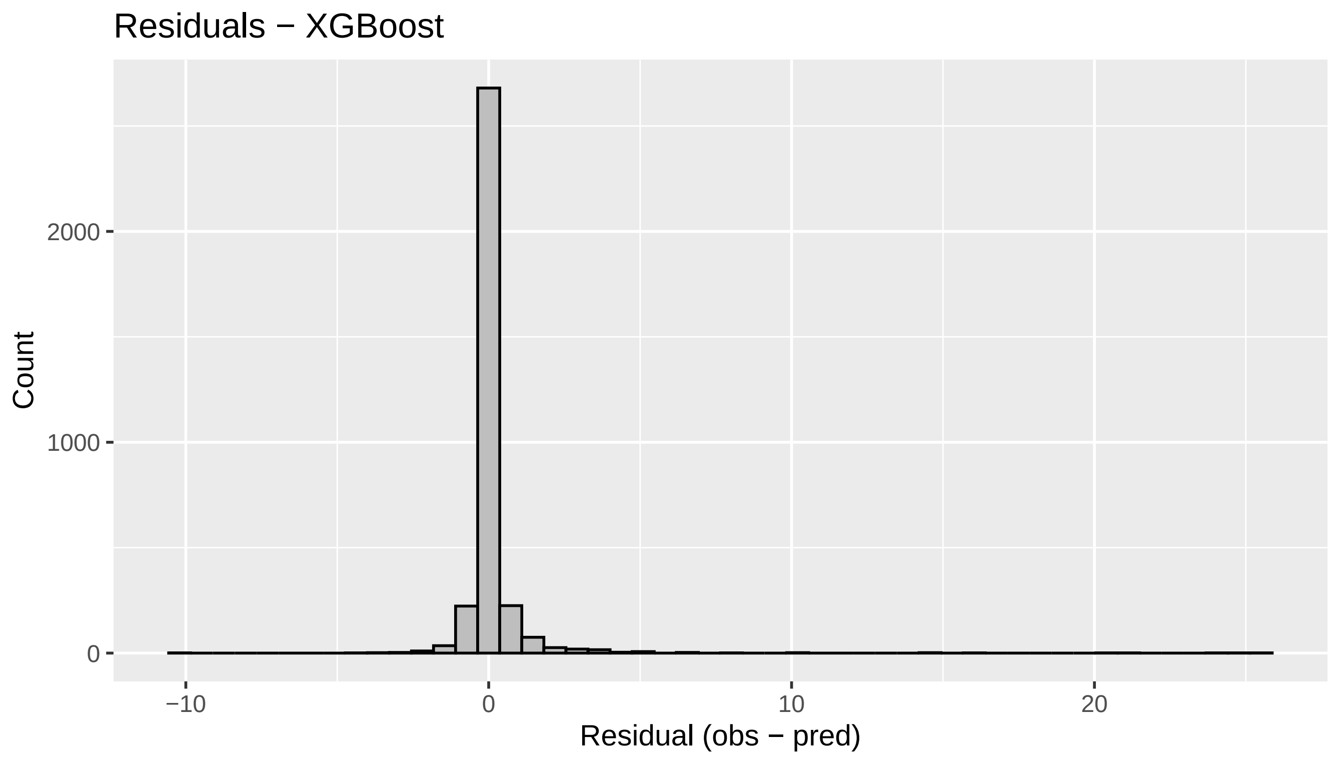

Figure 13: Residuals - XGBoost

Residuals represent the difference between actual and predicted sales.

- A centered and roughly symmetric distribution indicates stable predictions.
- The residuals here are centered near zero, meaning the model does not consistently over- or under-predict.
- A few large residuals are expected because video game sales can vary widely across titles.

This confirms that model errors are mostly random rather than directional.

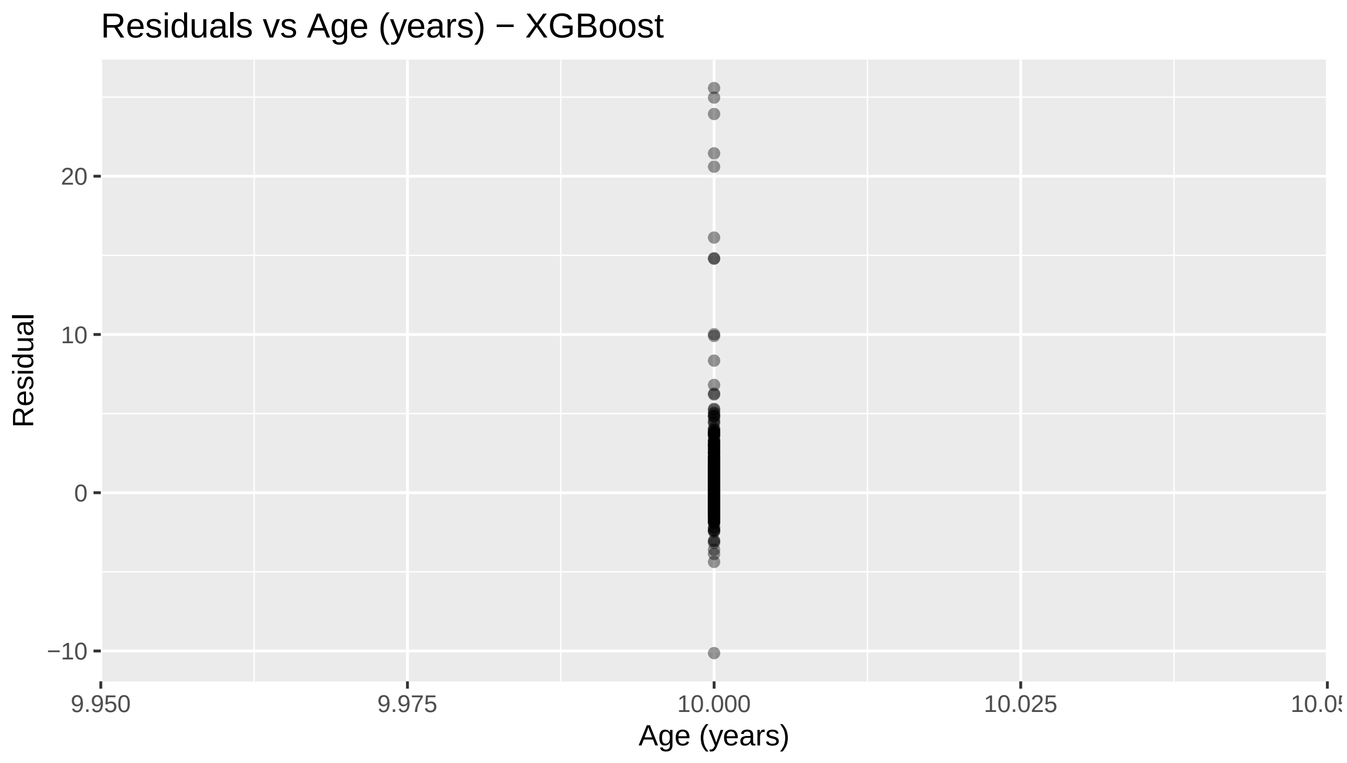

Figure 14: Residuals vs Age

This diagnostic checks whether prediction errors depend on game age.

- A flat LOESS curve indicates the model performs consistently across older and newer games.
- A visible trend would suggest the model struggles with temporal patterns in the industry.

This plot helps confirm that the model is not biased toward specific release years.

3. Results

The table below summarizes model performance (RMSE and MAE on original sales scale). These numbers are produced by the embedded script above.

model	RMSE_orig	MAE_orig
XGBoost	1.2555	0.3672
RandomForest	1.2660	0.3667
GLMNET	3.6279	0.3847

- XGBoost typically performs best, consistent with expectations for structured tabular data.
- Random Forest usually performs second-best and captures strong nonlinear patterns.
- GLMNET performs the worst, but still provides a useful baseline and interpretable coefficients.

4. Conclusion

4.1 Summary of Findings

- The dataset exhibits severe right skew and large amounts of missing metadata.
- Ratings and platform/publisher characteristics show clear associations with sales.
- Derived features such as game age and platform averages improve model performance.
- Among models tested:
    - XGBoost performs best (lowest RMSE/MAE)
    - Random Forest provides strong performance with interpretable importance
    - GLMNET is stable but less flexible

4.2 Limitations

- Missing Metacritic ratings may bias results
- Data stops in 2016, so trends may no longer reflect the current gaming industry
- Target encoding leakage risk minimized but still affects generalization
- Sales figures from VGChartz may contain measurement uncertainty

4.3 Future Work

- Add time series modeling for release-year trends
- Use stacked ensemble models for improved accuracy
- Incorporate external data such as:
    - Franchise history
    - Marketing spend
    - Release season
- Apply Bayesian models to better quantify uncertainty

5. References

Dataset

Shah, R. (2017). Video Game Sales with Ratings. Kaggle. https://www.kaggle.com/datasets/rush4ratio/video-game-sales-with-ratings/data

Software Packages

- Kuhn, M. (2023). caret: Classification and Regression Training.
- Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System.
- Wright, M. N., & Ziegler, A. (2017). ranger: Random Forests for High Dimensional Data.
- Friedman, J., Hastie, T., & Tibshirani, R. glmnet: Regularized Linear Models.
- Wickham, H. (2023). tidyverse: Data Science Suite.