INST414 Sprint 3 Report - Seamus Sullivan

Project Title: Video Game Ratings, Sales, and Genre Analysis
Research Question: How do critic and user review scores relate to video game sales across different genres?
Track: ADSP
Link to GitHub Repository:
https://github.com/seamusgsullivan/Video-Game-Ratings-Sales-and-Genre-Analysis

Code Note:
All code for Sprint 3 can be found in Sprint_3_Code.ipynb, which is located in the "notebooks" folder of my GitHub repository. The code is organized to match the sections and subsections in this document and in the Sprint 3 assignment description. All outputs should already be included in the notebook, but you can rerun the cell to reproduce them if needed.

Modeling Strategy & Implementation

Model Selection Justification

For this project, I chose to use tree-based models - mainly Random Forest and Gradient Boosting - because my exploratory data analysis showed that the relationship between critic/user scores and global sales is weak and nonlinear. Linear regression struggled to capture this pattern, so models that can learn nonlinear relationships were a better fit. Random Forest and Gradient Boosting also handle interactions between features naturally, which is helpful because genre may change how review scores relate to sales. I still kept a simple linear regression model as a baseline so I could compare a traditional, interpretable method to more flexible models.

There are trade-offs between the algorithms. Linear regression is very interpretable and fast, but it performs poorly when the patterns are nonlinear. Random Forest is more powerful and stable, but it is less interpretable and takes longer to train. Gradient Boosting often provides even higher predictive performance than Random Forest, but it can overfit more easily and depends heavily on hyperparameter choices. Overall, I selected these models because they balance performance and interpretability, and they are widely recommended for structured datasets like this one.

Before choosing the models, I looked at best practices for similar problems. Prior research on video-game sales prediction found that tree-based models often outperform linear models in datasets with weak signals and nonlinear patterns (Keerthana & Rao, 2019). More general work on ensemble regression shows that ensemble learning methods such as boosting and bagging usually perform better than single models when dealing with real-world data (Mendes-Moreira et al., 2012). These sources helped confirm that my choices were appropriate for this project.

Hyperparameter & Design Decisions

The key hyperparameter choices in this project included settings such as the number of trees, maximum depth, and learning rate for the tree-based models. I started by running each model with mostly default parameters from scikit-learn, because the goal of my project is to explore patterns rather than to optimize predictive accuracy. I made small adjustments to a few important hyperparameters, such as limiting tree depth to avoid overfitting and testing different numbers of estimators to compare stability. For Gradient Boosting, I kept the learning rate small and used a reasonable number of boosting stages so the model would learn gradually without overfitting.

I decided on these settings by following common best practices, checking documentation, and looking at how other similar projects approached tuning. I did not run a full grid search because the dataset is large enough that an extensive search would take time and may not yield meaningful improvements, especially given the weak predictive power of the features. Being transparent: I used mostly default parameters because they provide a fair baseline, are stable for exploratory work, and fit the goals of this project, which focuses more on understanding patterns and less on maximizing performance scores.

Data Partitioning Strategy

I split the dataset into an 80% training set and a 20% test set. This gives enough data to train the models while keeping a separate portion of the data for honest evaluation. I used a fixed random seed to keep the split reproducible. Since the main task in this part of the project is predicting global sales as a continuous value, the models are performing regression, so stratification does not apply. Later, when I create a classification model using the sales tiers (Low, Medium, High),

I will use stratified sampling so each tier is represented in both the training and test sets in similar proportions.

Cross-validation is part of my validation strategy, especially for comparing or tuning models. A simple 5-fold cross-validation approach will help reduce randomness from a single split and give a more stable sense of model performance. Since the dataset is not a time series and each row represents an independent game/platform entry, there are no temporal dependencies that require a time-based split. Overall, this partitioning strategy supports both model training and reliable evaluation without violating any structural assumptions about the data.

Model Development & Training

Baseline Model

For this project, my baseline model is a simple "mean predictor" that ignores all features and predicts the average global sales value from the training set for every game in the test set. This gives a neutral benchmark that represents "no learning." The baseline produced an R^2 of -0.0001, an RMSE of 1.5593, and an MAE of 0.7920. These results show that the baseline explains essentially none of the variation in global sales, which makes sense because it gives the same prediction for every game regardless of its review scores or genre. Establishing this baseline is important because any useful model needs to perform noticeably better than simply predicting the mean. The baseline gives me a clear point of comparison to judge whether the machine learning models in the next steps actually add value.

Primary Model(s)

My primary models are Linear Regression, Random Forest Regressor, and Gradient Boosting Regressor. Linear Regression provides a straightforward baseline among machine learning models because it assumes a linear relationship between critic scores, user scores, and genre. It trains quickly and is easy to interpret, making it a good starting point. Random Forest is an ensemble of decision trees that averages many tree predictions, helping capture nonlinear patterns and interactions in the data. Gradient Boosting builds trees one at a time, with each new

tree learning from the errors of the previous ones. This model often performs better on structured data but requires more tuning and training time.

All three models were trained using the same 80/20 train-test split. I kept most parameters at their default values and only made light adjustments, such as using 200 trees for Random Forest and Gradient Boosting and setting a small learning rate for Gradient Boosting to reduce the risk of overfitting. Training time was also recorded to understand the computational cost. After training, the results showed clear differences across the models. Linear Regression performed the weakest among the main models, with an R^2 of 0.1183, RMSE of 1.4641, and MAE of 0.7422. Random Forest improved on this with an R^2 of 0.1988, RMSE of 1.3956, and MAE of 0.6947, showing that tree-based methods can capture more structure in the data. Gradient Boosting performed the best overall, reaching an R^2 of 0.3227, RMSE of 1.2832, and MAE of 0.6327, although it took the longest to train.

Here is the code I used to train and evaluate each model:

```
# --------------------------------------
# Primary Model(s) + Model Comparison
# --------------------------------------

# Helper function to train and evaluate a model
def train_and_evaluate(model, X_train, X_test, y_train, y_test):
    start_time = time.time()
    model.fit(X_train, y_train)
    end_time = time.time()
    train_time = end_time - start_time

    y_pred = model.predict(X_test)

    r2 = r2_score(y_test, y_pred)

    mse = mean_squared_error(y_test, y_pred)
    rmse = np.sqrt(mse)

    mae = mean_absolute_error(y_test, y_pred)

    return {
        "model": model,
        "r2": r2,
        "rmse": rmse,
        "mae": mae,
        "train_time": train_time
    }

# Define models
lin_reg = LinearRegression()

rf = RandomForestRegressor(n_estimators=200, max_depth=None, random_state=42, n_jobs=-1)

gbr = GradientBoostingRegressor(n_estimators=200, learning_rate=0.05, max_depth=3, random_state=42)

# Train and evaluate each model
results = []
for name, model in [
    ("Linear Regression", lin_reg),
    ("Random Forest", rf),
    ("Gradient Boosting", gbr)
]:
    metrics = train_and_evaluate(model, X_train, X_test, y_train, y_test)
    metrics["name"] = name
    results.append(metrics)

# Show results as a small table
results_df = pd.DataFrame(results)[["name", "r2", "rmse", "mae", "train_time"]]
print("\nModel performance comparison table (R², RMSE, MAE, and training time):\n")
print(results_df)
```

Model Comparison

To compare performance across models, I used regression metrics (R^2, RMSE, and MAE) along with training time. Accuracy, precision, recall, and AUC do not apply because this is a regression task. The baseline model had an R^2 very close to zero, which shows it cannot explain any meaningful variation in sales. Linear Regression performed somewhat better but still struggled to capture the weak and nonlinear relationships in the dataset. Random Forest improved on Linear Regression by modeling nonlinear effects, but it still performed moderately. Gradient Boosting clearly performed the best, with the highest R^2 and the lowest error values, although it trained slower than the other models. The model comparison table is included below (the interpretability of each model is discussed in the next paragraph):

Model performance comparison table (R^2, RMSE, MAE, and training time):

	name	r2	rmse	mae	train_time
0	Linear Regression	0.118250	1.464113	0.742237	0.024868
1	Random Forest	0.198792	1.395643	0.694681	0.248130
2	Gradient Boosting	0.322727	1.283169	0.632691	0.467443

Interpretability also differed across models. Linear Regression was the easiest to interpret because each coefficient directly shows how a feature affects predicted sales. Random Forest had medium interpretability, since it offers feature importance scores, but individual trees are more difficult to understand. Gradient Boosting was the least interpretable because of its sequential learning process, even though it achieved the strongest performance. These differences matter because they highlight the trade-off between understanding the model and getting higher predictive accuracy.

Overall, Gradient Boosting was the strongest model, suggesting that boosting is able to extract more of the weak signal between scores, genre, and sales than Linear Regression or Random Forest. The biggest surprise was how much better Gradient Boosting performed compared to Random Forest, despite using the same features. This suggests that the dataset benefits from the incremental learning style of boosting, which can capture subtle patterns that Random Forest averages away. Another interesting result is that Linear Regression still performed noticeably

better than the baseline, showing that even a simple linear model captures some useful information from critic and user scores.

Feature Engineering (Advanced)

To test whether additional features could improve performance, I created two new variables based on the review scores. The first was Avg_Score, which averages the critic and user scores to create a single combined quality measure. The second was Score_Diff, which captures the difference between critic and user views of a game. These features were added to a new version of the feature set and used to train another Random Forest model.

The results showed that adding these engineered features did not improve performance. The Random Forest with the original features had an R^2 of 0.1988, an RMSE of 1.3956, and an MAE of 0.6947. After adding the engineered features, performance dropped slightly, with an R^2 of 0.1793, an RMSE of 1.4126, and an MAE of 0.7034. This small decline suggests that the engineered features did not introduce helpful new information and may have added noise or redundancy. Since critic and user scores are already included as separate features, combining them did not give the model anything substantially different to learn from. This result also matches the weak relationships observed during EDA, where both scores showed only limited predictive power. Because of this, the engineered features were not used in the final model.

Model Evaluation & Diagnostics

Regression Diagnostics

I focused my diagnostic analysis on the Gradient Boosting model because it performed the best out of all the models I tested. To evaluate how well it fits the data, I examined several diagnostic plots. The residual plot showed that most predictions cluster between 0 and about 2 million sales, and most residuals fall between roughly -2 and 3 million. The points looked randomly scattered around zero, but there was still noticeable spread, especially for games with higher predicted sales. This suggests that the model captures some general patterns but still makes large errors for certain games. The histogram of residuals looked close to a normal distribution, with most errors

near zero and far fewer large positive or negative errors. This is a good sign because it means the model is not making biased mistakes in one direction.

The Q-Q plot mostly followed the reference line, although it curved away at the upper end. This pattern shows that the model handles typical cases well but struggles with extreme values. These extreme points represent the highest-selling games, which the model consistently underpredicts. The Actual vs. Predicted plot showed the same pattern. Most games with low or moderate sales were predicted fairly well, but almost all high-selling games (over 5 million copies) were predicted far below their true values. Because sales are measured in millions, even a difference of one value means one million units. This makes the underpredictions for the highest-selling games especially large.

I reported both RMSE and MAE because they capture different aspects of error. RMSE penalizes large mistakes more heavily, which matters in this dataset because the biggest errors come from high-selling games. MAE gives the average size of mistakes and is less sensitive to extreme values. Using both metrics gives a more complete picture of model performance.

Cross-Validation & Generalization

To measure generalization, I ran 5-fold cross-validation on all three models. The Gradient Boosting model had the strongest average performance, with a mean R^2 of about 0.16 across folds. This is low, but it was still better than Linear Regression and Random Forest. The Random Forest model sometimes performed very poorly, as shown by its negative mean R^2 and high standard deviation. These results show that Random Forest was unstable for this dataset, possibly because the relationship between the features and sales is weak and noisy. The cross-validated RMSE for Gradient Boosting was lower than the other models, which further supports that it generalizes slightly better.

The learning curve for Gradient Boosting showed that the training R^2 stayed at zero, which is expected for this style of plot because it averages scores from subsets of the data. The validation R^2 started very negative with small training sizes, then steadily improved as more training data was used. As the training set grew, the curve flattened out, showing diminishing improvement.

This suggests that adding more data would probably not dramatically improve performance. The overall pattern indicates that the model generalizes reasonably but still struggles due to weak signals in the data.

Feature Importance & Interpretability

To understand which features were most important, I examined both tree-based importance and permutation importance for the Gradient Boosting model. In both methods, Critic Score was the most influential feature by a large margin. It had an importance score of about 0.62 in the tree-based method and about 0.73 in the permutation results. User Score was the second most important feature, but far below Critic Score. These two review-based variables contributed most of the predictive power in the model.

Genre features had very small importance values. The top genres were Shooter and Racing, but even these contributed only about 0.04 each in the tree-based importance. Many genres had almost no predictive influence, and some even had slightly negative permutation importance values, meaning that removing them actually improved performance. This suggests that genre does not add much predictive signal beyond the review scores, even though genres differ in average sales. Overall, the importance results indicate that critic reviews matter the most for predicting global sales, and user reviews matter to a lesser extent. Genre has very little direct predictive influence in this model.

Error Analysis & Failure Modes

To understand where the model fails, I looked at the ten largest prediction errors. All of the biggest mistakes came from games with very high sales. For example, one game sold about 21.8 million copies but was predicted to sell only around 1.3 million, resulting in an error of over 20 million units. Many other high-selling games showed similar patterns. This confirms that the model systematically underpredicts the highest-selling games. These cases drive up the RMSE and show that the model cannot capture the factors that make certain games extremely successful, such as franchise popularity, marketing budgets, or platform dominance - none of which are included in the dataset I used.

I also looked at residuals by genre. The average errors for most genres were close to zero, meaning the model did not consistently overpredict or underpredict specific categories. However, some genres, such as Puzzle and Shooter, had much higher variability (larger standard deviations), meaning the model was less stable for those types of games. Genres like Miscellaneous and Puzzle also showed especially high spread, which may reflect that these categories include unusual or clustered sales patterns. Overall, the main failure mode is underpredicting extremely successful games, not errors tied to specific genres.

In Sprint 4, I plan to address these issues by exploring alternative ways to model high-selling games, such as adding new features if possible, adjusting the model to focus more on tail behavior, or analyzing sales tiers instead of raw sales values. Even though the model is not highly accurate, these failure patterns provide useful insight into the limits of prediction when important real-world factors are missing.

Results & Interpretation

Key Findings

Overall, I found that critic review scores are the strongest predictor of global video game sales, with user scores also contributing but to a smaller degree. Genre has much less predictive power in the model, even though different genres do have different average sales levels. The Gradient Boosting model did the best job of using these features, but it still only explains a modest part of the variation in sales.

In terms of metrics, the baseline mean predictor had an R^2 of about 0, meaning it could not explain any variation in sales. Linear Regression improved this slightly, while Random Forest did better, but Gradient Boosting performed the best, with an R^2 of about 0.32 (Figure 2) and the lowest RMSE and MAE. This means the best model improves meaningfully over the baseline, but most of the variation in sales is still unexplained. Feature importance plots (Figure 1) show that Critic_Score has the highest importance by a large margin, followed by User_Score, while genre dummy variables contribute only small amounts.

One surprising result is that even with a flexible tree-based ensemble, the overall predictive power remains fairly low. In Sprint 2, I already saw that correlations between review scores and game sales were weak, but I expected that nonlinear models might uncover stronger patterns. Instead, the models confirm that review scores and genre only provide a limited signal. Another surprise is how badly the model performs on the highest-selling games: the error analysis and the actual-versus-predicted plot (Figure 3) show that many of the highest-selling games are heavily underpredicted, even when they have good review scores.

Interpretation in Domain Context

For the original problem - understanding how critic and user review scores relate to global sales across genres - the results suggest that reviews do matter, but only up to a point. Critic scores are consistently the most informative feature, so games with higher critic scores tend to sell more on average, and user scores also provide some useful signal. However, the relatively low R^2 values show that these factors alone are not enough to reliably predict how well a game will sell. In practice, this means that a publisher could use critic and user scores as one input when thinking about sales potential, but should not expect them to accurately predict which games will become major hits.

The size of the effects is also important. Gradient Boosting reduces error compared to the baseline and other models, but it still leaves a lot of unexplained variation (Figure 2). The residual and error plots show that the model is reasonably accurate for low- and medium-selling games, but performs poorly for the highest-selling games (Figures 3 and 4). From a practical point of view, this limits how useful the model would be for high-stakes decisions like budgeting or predicting big releases. The model is more helpful for describing general patterns - such as "higher critic scores are associated with higher sales" and "some genres sell more on average than others" - than for making precise predictions for individual games.

There are also important limits on what these results can tell us. The model is based on observational data, so it cannot answer causal questions, such as whether improving review scores would directly cause an increase in sales. Many important variables are missing, including marketing budget, franchise size, platform-specific factors, and timing of release. The dataset

also covers a particular time period and, after cleaning, only includes games with Metacritic critic and user scores. This likely biases the sample toward more prominent titles that received enough attention to be reviewed. Given these limitations, the results should be interpreted as correlations in this dataset, not as general rules for the entire video game industry or future releases.

Comparison to Baselines & Prior Work

Compared to the baseline mean predictor, all three machine learning models - Linear Regression, Random Forest, and Gradient Boosting - offer clear improvements. The baseline has an R^2 close to zero, while Gradient Boosting reaches around 0.32 (Figure 2), with lower RMSE and MAE. This shows that using critic score, user score, and genre gives a real, measurable advantage over making no use of the features at all. However, the modest R^2 also makes it clear that even the best model cannot capture most of the variation in game sales with these features alone.

In relation to prior work, some studies on video game sales prediction have used larger feature sets, including platform, publisher, release year, and other metadata, and sometimes report stronger performance. My project intentionally focused on a smaller, more interpretable set of features - critic score, user score, and genre - so it is reasonable that my R^2 values are lower. Still, the general pattern is consistent with industry knowledge and earlier research: review scores provide some predictive value, and tree-based ensemble methods like Gradient Boosting can outperform simpler linear models, but there are many other factors that influence whether a game becomes a major commercial success. These results support that view rather than challenging it.

Limitations, Assumptions & Threats to Validity

Data Limitations

A major limitation of my project is that the dataset I used does not include many variables that likely influence video game sales. Important factors - such as marketing budget, franchise popularity, advertising campaigns, platform install base (how many consoles or devices are out in the market), release timing within the year, and cross-platform differences - are not available. Without these, the model can only capture a small part of what actually influences sales. Another

limitation is that the cleaned dataset includes 7,017 game/platform entries, but only those that had complete critic and user scores. This means the dataset may skew toward more visible or popular titles that were reviewed on Metacritic, leaving out smaller or older games that lacked reviews. The temporal range is also limited, since most games in the cleaned dataset were released between 1996 and 2016. As a result, the findings of my analysis might not apply to more recent games or to industry changes that have happened after 2016. Finally, even though the dataset was cleaned carefully, some quality issues - like inconsistent review counts or possible scraping errors - may still exist in the background.

Methodological Limitations

Because my analysis uses observational data, the models cannot make any causal claims. Even if critic scores correlate with higher sales, this does not mean that raising a critic score would directly increase a game's revenue. The models also rely on assumptions that may not fully hold. For example, tree-based models can still be sensitive to hyperparameters, and the performance of Gradient Boosting depends on choices like learning rate and number of estimators. Although cross-validation was used, the weak predictive power suggests that the relationship between the features and sales is limited or noisy. There is also a risk that the Gradient Boosting model (which was used as the primary model in my analysis because it performed the strongest) underfits or oversimplifies the nonlinear factors behind the success of the highest-selling games. Since my analysis focused on a small set of features, the model may be missing important interactions that could not be tested.

Generalizability & Bias Concerns

The model is trained only on games with complete critic and user review data, most of which were released between 1996 and 2016. This limits how well it can generalize to games released today or to games in markets that do not rely on Metacritic-style reviews. Because the dataset overrepresents well-known titles - those that received enough attention to be reviewed - it may be biased toward established publishers and major franchises. This means the model likely performs worse on smaller indie titles, niche genres, or games released with little marketing. There may also be genre-related biases, since some genres have fewer samples (like Adventure or Puzzle) which makes predictions for those categories less reliable. Although the model does

not directly use demographic information, it still reflects the broader biases of the gaming industry and Metacritic's reviewer base.

Threats to Internal Validity

There are several possible threats to internal validity. Selection bias is present because only games with complete review data were kept in the cleaned dataset, which may remove lower-profile games and distort the overall distribution. Measurement error is also possible, since both VGChartz sales numbers and Metacritic review scores can be noisy or inconsistent. Confounding variables are another major concern. For example, franchise size could influence both review scores and sales, but it is not included in the dataset, meaning the model might incorrectly attribute effects to critic or user scores. While data cleaning removed missing values for the key variables used in my model (critic score, user score, genre, and global sales), it could not address deeper issues, such as how review counts or platform-specific differences might affect sales, because these factors were not included as features in my analysis.

Honest Statement on Feasibility

Overall, my approach works for identifying general patterns - like the fact that critic scores matter more than user scores - but it does not produce a highly accurate predictive model. The weak R^2 values across all models show that review scores and genre alone cannot explain most of the variation in global sales. If the goal were to build a practical video game sales prediction system, the current feature set would not be enough. For Sprint 4, I plan to shift from focusing on prediction accuracy to interpreting what the model's patterns say about the relationship between review scores and sales across genres. If stronger predictive performance were required, I would need a richer dataset with variables capturing marketing effects, franchise recognition, seasonality, and platform characteristics. For now, my original problem statement still makes sense, but the findings of my analysis should be viewed as descriptive rather than predictive.

Sprint 4 Plan & Refinement Strategy

Model Refinement Roadmap

For Sprint 4, my goal is to refine my analysis rather than focus on maximizing predictive accuracy. Because the Gradient Boosting model performed the best out of the models I tested, I plan to focus on improving its interpretability and exploring why it behaves the way it does. I do not plan to ensemble additional models or introduce deep learning because the current feature set is too limited to justify more complex approaches. Instead, I will emphasize two areas: feature expansion and targeted model tuning.

One refinement step is to engineer a few additional features that may strengthen the model, such as review score ratios, binned score categories (e.g., low/medium/high critic score), or interaction terms between critic and user scores. These features could make nonlinear patterns easier for the model to detect. I also plan to test a small set of modified hyperparameters for Gradient Boosting - such as reducing the learning rate or increasing tree depth - to see whether these adjustments help reduce the underprediction of high-selling games.

Another part of the roadmap is to explore whether smaller subsets of the data, like individual genres or time periods, reveal clearer relationships. Since the global model has weak predictive power, breaking the dataset into more meaningful segments may help uncover patterns that are hidden when everything is combined. Overall, my main measurable goals for Sprint 4 are:

1. Create and test at least three new features
2. Run a focused hyperparameter search for Gradient Boosting
3. Generate clearer, more interpretable insights about how review scores relate to sales

Addressing Findings

Sprint 3 revealed several challenges that I plan to address directly. The first challenge was the model's weak predictive performance, especially with high-selling games. To address this, I will experiment with new features that better capture nonlinear patterns and differences between game types, and I will test adjusted hyperparameters to reduce underfitting for the highest-selling games.

A second major challenge was the limited feature set. The current dataset does not include important variables like franchise recognition or marketing strength, so the model can only detect part of the story. While I cannot fully fix this without new data, I can partially mitigate the issue by creating proxy features or grouping games into categories that may reflect some of these effects (e.g., identifying long-running franchises by looking for recurring franchise names such as “FIFA” or “Call of Duty” across different years or platforms).

The third challenge was the mismatch between model results and real-world knowledge. Even though some genres sell well on average, genre was not an important predictor in the model. In Sprint 4, I will look more closely at genre-specific models and partial dependence plots to understand whether genre interacts with review scores in ways that are not captured by global feature importance values.

Looking Ahead for Deliverables for Sprint 4

Sprint 4 will focus on producing a clear and well-structured final analysis. My main deliverables for Sprint 4 will include a written interpretation of the model’s insights for a non-technical audience, along with refined visualizations and a final narrative that explains what my results suggest about the relationship between review scores, genre, and video game sales. The goal is not to maximize predictive accuracy but to clearly communicate what patterns the model revealed and what they mean in practice.

There are still some risks as I move forward. New feature engineering may not meaningfully improve model performance, and the dataset’s limitations could continue to restrict what the model can learn. The lack of additional variables may also make it difficult to offer concrete recommendations. Even with these challenges, the remaining work is manageable, and I do not expect to need additional compute resources or new data. The main constraint will be time - especially if new features require more testing or if additional exploratory analysis becomes necessary.

Overall, Sprint 4 will focus on interpreting results clearly, refining the Gradient Boosting model’s insights, and producing a final report that communicates what the dataset I used can and

cannot tell us about video game sales. The final deliverables will be presentation-ready and designed to summarize the most meaningful findings, explain their limitations, and offer high-level insights about how review information relates to the commercial performance of video games.

Self-Assessment

At this point in the project, I am on track with the overall project timeline. I have completed all of the Sprint 3 goals, which puts me in a good position for Sprint 4, and I don't feel behind in any major area.

My biggest win from this sprint was developing a stronger understanding of how the Gradient Boosting model works and why it outperformed the other models, even though its overall predictive power was still modest. Producing the diagnostic visualizations also went better than expected - they made the Gradient Boosting model's strengths and weaknesses much clearer and helped me interpret its results more effectively.

My biggest challenge was dealing with the weak predictive performance across all models. No matter which model I used, the features I selected for my analysis - critic score, user score, and genre - could not explain much of the variation in global sales. The hardest part was figuring out why the models were struggling and determining whether the issue came from the algorithms themselves or from the limited feature set I used. This uncertainty also made it difficult to decide which refinement strategies were worth pursuing for Sprint 4 and which ones were unlikely to meaningfully improve the results.

On a scale of 1 to 10, I would rate my confidence in the model's results at about a 6. I trust the diagnostics and believe the model accurately reflects what the dataset I used can show, but I also know that the limited feature set I used naturally restricts the model's predictive ability. I am more confident in the general patterns the model identifies than in its ability to accurately predict video game sales.

For support, I do not need anything specific from instructors or TAs at this time. The remaining work for Sprint 4 seems manageable, and I feel prepared to move forward with refining my analysis and writing the final interpretation.

References

Keerthana, B., & Rao, K. V. (2019). Sales prediction on video games using machine learning. *Journal of Emerging Technologies and Innovative Research (JETIR), 6*(6), 482-487. https://www.jetir.org/papers/JETIR1907H50.pdf

Mendes-Moreira, J., Soares, C., Jorge, A. M., & De Sousa, J. F. (2012). Ensemble approaches for regression: A survey. *ACM Computing Surveys, 45*(1), 1-40. https://doi.org/10.1145/2379776.2379786