**Introduction**

In this project, we aim to predict video game sales by training machine learning models. Machine Learning, a subfield of Computer Science, enables computers to learn from data and make decisions. This project falls under supervised learning, where both features and labels are used to train the model. Specifically, it is a regression problem since the output (sales) is a numerical value.

We selected a dataset containing features such as game platform, publisher, genre, and release year, along with global sales data. The dataset was analyzed, cleaned, and preprocessed to prepare it for training. Various regression models were then trained and evaluated to determine their effectiveness in predicting future sales. The dataset was split into training and testing sets to ensure accurate model evaluation.

**The Dataset**

Our dataset contains the sales data of video games, comprising 16,598 rows with columns such as Rank, Name, Platform, Year, Genre, Publisher, NA\_Sales, JP\_Sales, Other\_Sales, and Global\_Sales. The source of the dataset is Kaggle.

* **Rank**: The ranking of overall sales.
* **Name**: The name of the video game.
* **Publisher**: The company that published the game.
* **Platform**: The system used for playing the game (e.g., PC, PS4).
* **Year**: The year the game was published.
* **Genre**: The category of the game (e.g., Action, Sport).
* **Sales**: Sales figures in North America, Europe, Japan, other regions, and globally (in millions).

The Rank and Name columns do not provide meaningful data for prediction, and NA\_Sales, JP\_Sales, and Other\_Sales are highly correlated with Global\_Sales, so they were removed. The features used for prediction are Platform, Publisher, Genre, and Year, with Global\_Sales as the predicted output. Platform, Publisher, and Genre are categorical variables, making this a regression problem with a scalar output value.

# Methodology

Our methodology involved several key steps: cleaning the data, analyzing the data, preparing the data, applying different machine learning models, recording the results, evaluating the models, and selecting the best model.

# Dataset Cleaning

We started by importing the dataset and using the Rank column as the index. We removed unnecessary columns such as Name, NA\_Sales, JP\_Sales, and Other\_Sales. We then checked for missing values, finding 271 rows missing the Year value and 58 rows missing the Publisher value. To handle the missing values, we removed rows with empty Publisher values and replaced missing Year values with the most common year using the mode function. Finally, we converted the Year values from float to integer.

# Dataset Analysis and Visualization

We analyzed the data in our columns, starting with the Genre column, which has 10 unique values. Action and Sport were the most common genres. The Platform column has 31 unique values, with DS and PS2 being the most common platforms. The Publisher column has 579 unique values, with Electronic Arts being the top publisher. The Year values range from 1980 to 2020, with 2008 and 2009 being the most common years.

# Data Preprocessing and Preparation

Before training the machine learning models, we preprocessed and prepared the data. The Publisher column had many unique values, with around 200 publishers occurring only once. We replaced these rare publishers with "Other". We also removed rows with outliers in Global\_Sales values and rescaled the Year column values to represent the number of years passed from the start year.

We then handled the categorical variables (Publisher, Platform, and Genre) by encoding them into numerical data using One Hot Encoding and Ordinal Encoding. One Hot Encoding adds dummy variables for each value of the categorical variable, while Ordinal Encoding assigns a number to each value. We experimented with both encoding types and normalized the ordinal encoded data.

# Machine Learning Regression Models

We evaluated the performance of each model using several key metrics:

* **Root Mean Square Error (RMSE)**: Measures the average magnitude of the errors. A lower RMSE indicates a better fit.
* **Mean Absolute Error (MAE)**: Measures the average magnitude of the errors in a set of predictions, without considering their direction. Lower values indicate better accuracy.
* **Mean Absolute Percentage Error (MAPE)**: Measures the accuracy of a forecasting method as a percentage. High values can indicate less reliable predictions.
* **R² (R-squared)**: Indicates how well the model's predictions match the actual data. A higher R² value represents a better fit.

### Linear Regression

* **Test RMSE**: 0.52
* **Test MAE**: 0.32
* **Test MAPE**: 315.79%
* **Test R²**: 0.20

### XGBoost

* **Test RMSE**: 0.52
* **Test MAE**: 0.31
* **Test MAPE**: 314.70%
* **Test R²**: 0.22

The evaluation metrics reveal that both models performed similarly in terms of RMSE. However, XGBoost had a slightly better MAE and R² compared to Linear Regression, suggesting it has a marginally better fit for the data. Despite the high MAPE values, which indicate some level of prediction error, the XGBoost model is more accurate overall.

# Conclusion

In conclusion, our analysis and modeling efforts demonstrated that the XGBoost model was the most effective for forecasting video game sales in this dataset. Its superior performance metrics, particularly in terms of MAE and R², indicate that it can provide valuable insights for inventory management and strategic planning.

Future work could involve further tuning of the models and exploring additional features to enhance prediction accuracy. Additionally, addressing the high MAPE values could improve the model's reliability for instances with lower sales figures. Overall, the XGBoost model's performance suggests it is a robust tool for predicting video game sales.