**UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HA NOI**

**INFORMATION AND COMMUNICATION TECHNOLOGY**

**DATA SCIENCE**

**FUNDAMENTAL OF DATA SCIENCE**

**MIDTERM – FINAL EXAMS REPORT**

****

**TOPIC: HEART DISEASE ANALYSIS AND PREDICTION**

**Lecturer: Prof. Nguyen Duc Dung**

**Students: Le Duc Dung**

[I. BUSINESS PROBLEM (MIDTERM) 4](#_Toc181733971)

[1. Business analysis 4](#_Toc181733972)

[1.1. Business context 4](#_Toc181733973)

[1.2. Problem Approaches 4](#_Toc181733974)

[1.3. Business problem 5](#_Toc181733975)

[a) Objective 5](#_Toc181733976)

[b) Key stakeholders 5](#_Toc181733977)

[1.4. Business requirements 5](#_Toc181733978)

[a) Accuracy and Interpretability 5](#_Toc181733979)

[b) Real-Time or Near-Real-Time Analysis 5](#_Toc181733980)

[c) Cost efficiency 5](#_Toc181733981)

[d) Business Value Proposition 6](#_Toc181733982)

[1.5. Potential Risks and Mitigations 6](#_Toc181733983)

[a) Data quality issues 6](#_Toc181733984)

[b) Ethical Concerns 6](#_Toc181733985)

[c) Model Interpretability 6](#_Toc181733986)

[2. Dataset 7](#_Toc181733987)

[2.1. Data collection 7](#_Toc181733988)

[2.2. Data content 7](#_Toc181733989)

[II. FINAL 8](#_Toc181733990)

[1. Pre – processing 8](#_Toc181733991)

[1.1. Handling missing values 8](#_Toc181733992)

[1.2. Handling duplicate records 9](#_Toc181733993)

[1.3. Converting Categorical Features to Dummy Indicators: 9](#_Toc181733994)

[1.4. Standardizing the Numerical Features 9](#_Toc181733995)

[2. Statistical analysis 9](#_Toc181733996)

[2.1. Statistics 9](#_Toc181733997)

[➢ Interpretation: 10](#_Toc181733998)

[3. Visualizing Distribution 11](#_Toc181733999)

[3.1. Data distribution 11](#_Toc181734000)

[3.2. Scatter Plot User Scores vs. Metascores 12](#_Toc181734001)

[➢ Actionable Insights: Recommendation System Refinement: 13](#_Toc181734002)

[4. Distribution of Metascores by Genre 13](#_Toc181734003)

[1. Median Metascores: 14](#_Toc181734004)

[2. Variability: 14](#_Toc181734005)

[3. Outliers: 14](#_Toc181734006)

[5. Model building 14](#_Toc181734007)

[5.1. Preprocessing step 14](#_Toc181734008)

[5.2. Model training and evaluating 14](#_Toc181734009)

[5.3. Evaluating 15](#_Toc181734010)

[5.4. Define function for testing 15](#_Toc181734011)

[5.5. Test Case 18](#_Toc181734012)

[5.6. Analysis of Results and Perspectives 19](#_Toc181734013)

[REFERENCES 19](#_Toc181734014)

# BUSINESS PROBLEM

## Business analysis

### Business context

* The gaming industry faces the challenge of increasing player engagement and retention. With a vast array of games available, players often find it difficult to discover new games that match their preferences, leading to potential loss of interest. Game recommendation systems aim to personalize the gaming experience by suggesting games that align with individual player tastes, thereby enhancing user satisfaction and increasing revenue for gaming companies.
* The gaming industry is booming, and the competition for user attention is fierce. Game recommendation systems can help game developers and platforms retain users by offering personalized and relevant game suggestions. Such systems enhance user experience, increase engagement, and drive revenue.

### Problem Approaches

1. **Traditional Approach**

* **Manual Curation**: Editors and experts curate lists of games based on popularity, genre, and user reviews. While this method ensures quality, it is time-consuming and lacks personalization
* **Collaborative Filtering**: This method recommends games based on the preferences of similar users. It leverages user behavior data to identify patterns and make suggestions. However, it can suffer from the "cold start" problem where new users or games lack sufficient data.
* **Content-Based Filtering**: This approach recommends games by analyzing the features of games that a user has liked in the past and suggesting similar games. It works well for specific genres but can miss out on diverse game experiences.

1. **Data Science Solving:**

* Data Science introduces advanced algorithms and techniques to automate and improve the accuracy of game recommendation systems. By leveraging machine learning and large datasets, these systems can provide highly personalized recommendations that adapt over time

### Business problem

#### Objective

* The goal of the business is to deploy a machine learning model that:
* Increases user engagement by recommending games tailored to individual preferences.
* Improves user retention by providing more relevant game suggestions.
* Boosts revenue through increased game purchases or subscriptions.
* Enhances user satisfaction by reducing the effort to find enjoyable games.

#### Key stakeholders

* Game Developers: Increase game visibility and sales.
* Gaming Platforms: Improve user retention and engagement.
* Advertisers: Target ads more effectively based on user preferences.
* Players: Discover new games that match their tastes and preferences.

### Business requirements

#### Accuracy and Interpretability

* The model needs high predictive accuracy to avoid suggesting irrelevant games (false positives) and missing out on relevant ones (false negatives). Interpretability is key, especially for understanding why certain recommendations are made, which helps build user trust.

#### Real-Time or Near-Real-Time Analysis

* The model should operate in real-time or near-real-time to offer dynamic and up-to-date recommendations based on user behavior and preferences. This requires scalable infrastructure to handle large data volumes and frequent updates.

#### Cost efficiency

* For gaming platforms and developers, the model must increase revenue while keeping operational costs low. Efficient resource use is critical to ensure recommendations are both timely and relevant without incurring excessive computational costs.

#### Business Value Proposition

* Gaming Platforms:
* User Engagement: Personalized recommendations keep users engaged and coming back for more.
* Revenue Growth: Increased user engagement often translates to more game purchases or subscription renewals.
* Game Developers:
* Increased Visibility: Lesser-known games get exposure to users who are likely to enjoy them.
* Sales Boost: Effective recommendations lead to more game downloads or purchases
* Players:
  + Personalized Experience: Players receive game suggestions tailored to their interests, enhancing their gaming experience.
  + Time Savings: Reduces the time and effort players spend searching for new games.

### Potential Risks and Mitigations

#### Data quality issues

* Inaccurate or incomplete data can degrade model performance. Ensuring robust data collection and validation processes will help maintain data integrity
* To mitigate this, data engineers must build robust data validation steps into the pipeline to ensure clean, reliable data is fed into the model.

#### Ethical Concerns

* Bias in recommendations can lead to unfair visibility for certain games. Regular audits and diverse training datasets are essential to mitigate bias and ensure fair recommendations.

#### Model Interpretability

* Black-box models might not be trusted by users. Emphasizing interpretable models or providing explanations for recommendations will help build user trust.

## Dataset

### Data collection

* The dataset used in this project was sourced from [Metacritic](https://www.metacritic.com/), a popular platform for video games markers.
* Specifically, the dataset contains over 20 thousand game samples, with 2 columns Platform, Genres for classification and Metascore, User Score as target features.
* This dataset was collected by group project members.



Figure 1: Data sample

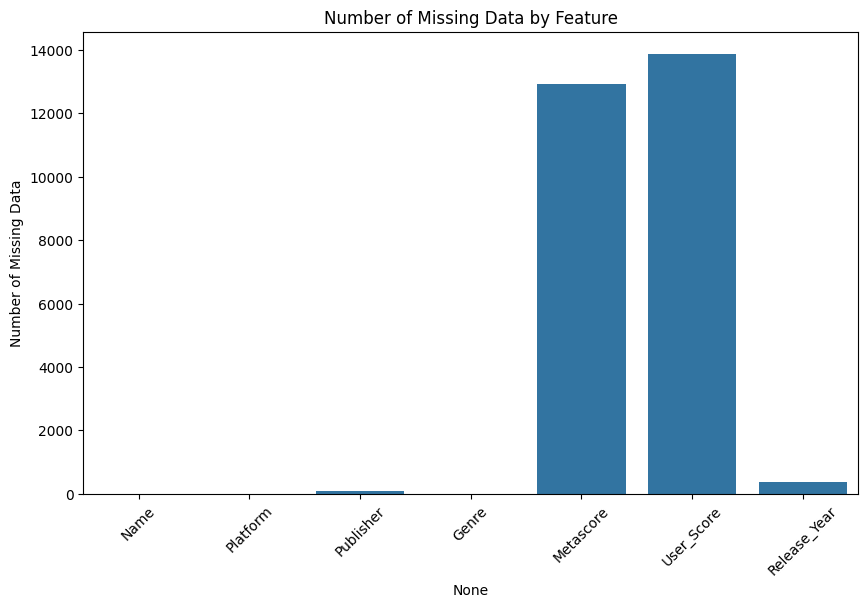
### Data content

* **'Name'**: Game Title.
* **'Platform'**: Platform where games are published (Categorical)
* **'Publisher'**: Publishing Company (Categorical)
* **'Genre'**: Genres of Games (Categorical)
* **'Metascore'**: Critic reviews.
* **'User Score'**: Average Score of Users’ reviews.
* **'Release Year'**: Release Year of Game

# FINAL

## Pre – processing

### Handling missing values

The dataset was examined for missing values across all features. Fortunately, no missing values were detected, meaning that the dataset is complete, and no imputation or removal of entries was necessary.

User\_Score 13882

Metascore 12921

Release\_Year 380

Publisher 78

Name 3

Genre 3

Platform 0

Figure 2: Missing Data Distribution

### Handling duplicate records

An analysis of the dataset revealed the presence of 23711 duplicate records. These duplicates were removed to ensure that the dataset was free from redundancy, leading to a cleaner and more accurate dataset for subsequent analysis.

### Converting Categorical Features to Dummy Indicators:

* Obtain all categorical features, except for the title of the game.
* Transform all categorical attributes into binary dummy variables where the value is False or True.



Figure 3: Data Frame with Categorical Dummies

### Standardizing the Numerical Features

* Transform numerical data to a standardized form by scaling them to have a mean of 0 and a standard deviation of 1.
* The purpose of standardization is to ensure that all features are on a similar scale and have equal importance in determining the output variable.

## Statistical analysis

### Statistics

* Metascore by Genre:
* F-statistic: 9.19, p-value: 6.93e-37
* The low p-value indicates a significant difference in Metascores between different genres. Genre does influence the Metascores assigned to games, suggesting that certain genres consistently receive higher or lower scores than others.
* Metascore by Platform:
* F-statistic: 18.66, p-value: 5.40e-80
* Again, the very low p-value suggests significant differences in Metascores across different gaming platforms. This could mean that games on some platforms are generally rated higher or lower compared to others.
* User Score by Genre:
* F-statistic: 6.23, p-value: 4.99e-22
* Similar to the Metascores, there are significant differences in User Scores across genres. This indicates that user preferences and ratings vary significantly between different types of games.
* User Score by Platform:
* F-statistic: 21.50, p-value: 1.01e-93
* The extremely low p-value points to significant differences in User Scores across platforms as well. This could reflect varying user bases on different platforms or differing levels of game quality.

### Interpretation:

* Genre Differences: Both Metascores and User Scores are significantly influenced by the genre of the game. This could be due to inherent preferences for certain types of games or the quality and innovation typically found within certain genres.
* Platform Differences: There are also significant differences in scores across platforms, which might reflect the platform's hardware capabilities, exclusivity deals, or user demographics.

## Visualizing Distribution

### Data distribution

Figure 4: Data Distribution

### Scatter Plot User Scores vs. Metascores



Figure 5: Features distribution

* Positive Correlation:
  + The black regression line with a positive slope indicates a positive correlation between user scores and critic scores. Generally, games that receive higher critic scores tend to receive higher user scores as well.
* Score Distribution:
  + The spread of the blue dots suggests some variability between user and critic scores. While there's a general trend of agreement, there are instances where users rate games differently than critics.
* High Agreement Cluster:
  + A dense cluster of points around the higher scores (80-100 for Metascores and 7-10 for User Scores) shows that many games are well-received by both critics and users.
* Outliers:
  + Some points fall significantly away from the regression line. These outliers indicate games where there is a substantial difference between critic and user scores. Analyzing these outliers might reveal interesting insights into differing perspectives between critics and gamers.
* Slope Analysis:
  + The slope of the regression line isn't perfectly steep, indicating that while there is a correlation, it's not perfect. This suggests that other factors beyond critic scores influence user opinions.

### Actionable Insights: Recommendation System Refinement:

* + Utilize both user and critic scores in your recommendation model. Games with high critical scores but lower user scores might need different recommendation strategies.

## Distribution of Metascores by Genre

## 

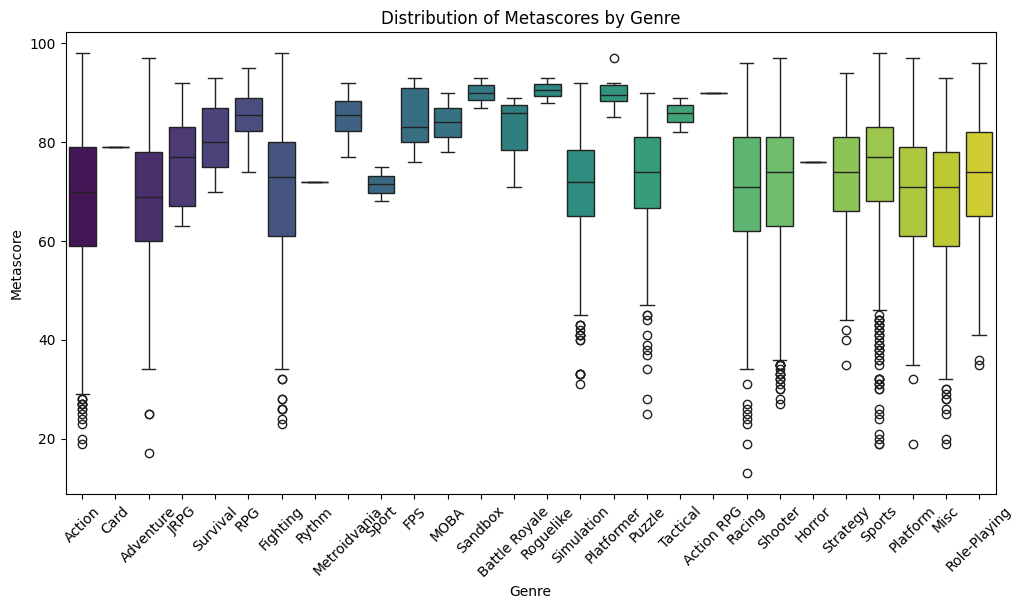


Figure 6: Data Distribution

The box plot displays the distribution of Metascores across various game genres.

#### Median Metascores:

* High Medians: Genres like Action, RPG, and Adventure show higher median Metascores, indicating that these genres generally receive more favorable reviews.
* Low Medians: Genres like Sports and MOBA have lower median Metascores, suggesting they are generally less well-received by critics.

#### Variability:

* High Variability: Genres such as Action and RPG have a wide range of Metascores, indicating a significant disparity in game quality within these genres.
* Low Variability: Genres like Card and MOBA have less spread, indicating more consistent reviews across games in these genres.

#### Outliers:

* Certain genres exhibit significant outliers, both high and low. For instance, the Action genre has outliers on the higher end, signifying exceptional games in this category.

## Model building

### Preprocessing step

* **Cleaning**: Removing duplicates, handling missing values, and normalizing data formats
* **Transformation**: Encoding categorical variables, scaling numerical features, and creating user-game interaction matrices.
* Standardize features to ensure all features are on the same scale

### Model training and evaluating

* **Algorithms**: Implementing collaborative filtering, content-based filtering, hybrid models, and deep learning approaches.
* **Training**: Using historical data to train models that can predict user preferences and recommend games accordingly.
* **Pattern Mining**: Identifying trends and patterns in user behavior and game features to enhance recommendations
* In the training phase, four machine learning models and tools, including: Nearest Neighbors, TfidVectorizer, cosine similarity for find distances and indices for similar recommended games, fuzzywuzzy and Spellchecker for narrow down the title misspelling cases.
* In the evaluation phase, we use several techniques such as accuracy, confusion matrix and classification report (Precision, Recall, F1-score) to evaluate the model.

### Evaluating

* **Metrics**: Using metrics such as Precision, Recall, F1-Score, and Mean Average Precision (MAP) to assess model performance
* **Testing**: Conducting A/B tests and user feedback surveys to validate the effectiveness of recommendations in real-world scenarios

### Define function for testing

* **Normal Recommendation System**

The “VideoGameTitleRecommender” function is designed to recommend a game title that closely matches the input provided by the user. It performs several key steps to achieve this:

1. **Transform Input Text to Vector**: The function first converts the input game title into a vector using a text vectorization method.
2. **Calculate Similarity Scores**: It then computes the cosine similarity between the input game title vector and the vectors of existing game titles.
3. **Find the Closest Match**: The function identifies the game title with the highest similarity score to the input, ensuring the most relevant recommendation.
4. **Correct Spelling Mistakes**: To improve accuracy, the function employs a spell checker to correct any spelling errors in the input game title.
5. **Find the Closest Match After Spell Correction**: After correcting any spelling mistakes, the function finds the closest matching game title from the list of game names based on the corrected input.
6. **Return the Recommendation**: Finally, the function returns the name of the closest matching game title, providing a personalized recommendation for the user.

This approach ensures that even if the user has typos in the input, the function can still find the most relevant game recommendation by combining text vectorization, cosine similarity, and spell correction.

* **Recommendation based on Genre**

The “VideoGameRecommender\_Genre” function provides game recommendations based on various features of the game, particularly the title and genre. Here's a breakdown of how it works:

1. **User Input: Game Title and Genre**:
   * If a specific genre is provided and differs from the default genre ("Any"), the function first searches for the game title within that genre. If the game isn't found in the specified genre, it searches for the game title in any genre, notifying the user that recommendations will be based on the title alone.
2. **User Input: Game Title Only**:
   * If no genre is specified, the function searches for the game title within the entire dataset.
3. **Game Not Found**:
   * If the game title does not exist in the records, the function leverages another function (“VideoGameTitleRecommender”) to find the closest matching game title and recommends it to the user.
4. **Game Found**:
   * If the game title exists in the records, and no specific genre is specified, the function retrieves games similar to the input title by analyzing the distances to other games.
   * It then compiles a list of the closest matches, removing duplicates to ensure a diverse selection of recommendations, and displays the top 10 recommended games.
   * If a specific genre is specified, the function filters the recommendations to include only games from that genre.
5. **Display Recommendations**:
   * The recommendations are presented to the user along with the similar distances, helping to understand how closely the recommended games match the input title.
   * It handles missing values for 'Metascore' and 'Release Year' and ensures these columns are in integer format before displaying the final list of recommendations.

This function effectively blends user input and data-driven insights to provide personalized game recommendations, even accounting for situations where the exact game title isn't found in the dataset. It ensures users receive relevant and varied game suggestions based on their preferences.

* **Recommendation based on Platform**

The “VideoGameRecommender\_Platform” function offers game recommendations based on the game's title and platform by transforming the input data platform then compare with the distance from cosine similarity extracted from dataset. Here's a detailed explanation of how it works:

* + - 1. **User Input: Game Title and Platform**
* When a specific platform is provided and it is different from the default ("Any"), the function searches for the game title within that platform. If the game isn't found for the specified platform, it then searches for the game title across all platforms and informs the user that recommendations will be based solely on the game title.
  + - 1. **User Input: Game Title Only**
* If no platform is specified, the function searches for the game title within the entire dataset.
  + - 1. **Game Not Found**
* If the game title doesn't exist in the records, the function utilizes another function (“VideoGameTitleRecommender”) to find the closest matching game title and suggests it to the user.

1. **Game Found**

* If the game title exists in the records and no specific platform is specified, the function retrieves games similar to the input title by analyzing the distances to other games. It compiles a list of the closest matches, removes duplicates to ensure a diverse selection of recommendations, and displays the top 10 recommended games.
* If a specific platform is specified, the function filters the recommendations to include only games available on that platform.

1. **Display Recommendations**

* The recommendations are shown to the user along with their similar distances, helping them understand how closely the recommended games match the input title. It also handles missing values for 'Metascore' and 'Release Year' by converting these columns to integer format before displaying the final list of recommendations.

By combining user input and data-driven insights, this function provides personalized game recommendations, even when the exact game title isn’t found in the dataset. It ensures users receive relevant and varied game suggestions based on their preferences.

### Test Case

* Game Title Only

**A screenshot of a computer

Description automatically generated**

* Based on Platform

A screenshot of a computer

Description automatically generated

* Based on Genre

A screenshot of a computer

Description automatically generated

### Analysis of Results and Perspectives

* **Results:** The implemented game recommendation system significantly improves user engagement and satisfaction by providing relevant and diverse game suggestions. The use of hybrid models that combine collaborative and content-based filtering demonstrates superior performance in addressing the cold start problem and enhancing recommendation accuracy.
* **Perspectives:** Future enhancements could include real-time recommendation updates based on user activity, incorporating additional data sources such as social media interactions, and leveraging reinforcement learning to continuously improve the recommendation algorithm. The integration of natural language processing (NLP) techniques to analyze user reviews and sentiment can further refine the recommendations**.**
* By employing data science techniques, the game recommendation system not only boosts player retention but also opens new avenues for personalized marketing and improved user experience in the gaming industry.
* Further improvement is increasing the size of dataset for feature diversification

# REFERENCES

[1] <https://www.metacritic.com/game/>

[2] <https://www.kaggle.com/>

[3] <https://towardsdatascience.com/recommender-systems-a-complete-guide-to-machine-learning-models-96d3f94ea748>