INTERNATIONAL UNIVERSITY

VIETNAM NATIONAL UNIVERSITY – HO CHI MINH CITY

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

-----***-----

DATA MINING

IT160IU

FINAL PROJECT

Movie Mining

Submitted by Group O

No.	**Student Name**	**Student ID**
01	Đỗ Hùng Việt	ITCSIU22197
02	Hà Minh Trí	ITCSIU22194

Instructor: Nguyen Quang Phu

Ho Chi Minh City, Vietnam, 2025

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to all individuals who supported us throughout the completion of this Data Mining project.

We are especially thankful to Dr. Nguyen Thi Thanh Sang for her valuable guidance, insightful feedback, and continuous encouragement during the course. Her support played an essential role in shaping the direction and quality of our work.

We would also like to extend our appreciation to Mr. Nguyen Quang Phu, MSc., for his practical instructions and technical assistance during the lab sessions, which greatly helped us in understanding and implementing the concepts required for this project.

Our sincere thanks also go to our teammates for their cooperation, commitment, and effective collaboration throughout the project. Finally, we are grateful to our families and friends for their constant motivation and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The rapid increase of anime titles, fueled by worldwide popularity and streaming services, has made it difficult for viewers to find content that meets their preferences. This study examines an anime dataset containing information such as anime ID, name, genre, type, episode count, and user ratings to identify patterns in audience preferences. To ensure quality and consistency, the data is initially cleaned, indexed, and formatted. Data mining approaches, such as classification algorithms (OneR, Naive Bayes, J48) and association rule mining (Apriori), are used to investigate the correlations between anime attributes and user ratings. Classification models are analyzed to resolve which aspects have the greatest influence on rating behavior, whereas association rules uncover common patterns that link anime elements to high ratings. The findings provide insights into the aspects that influence anime popularity and show how data-driven approaches can help with better suggestions and understanding of viewers' behavior.

CHAPTER 1. INTRODUCTION

1.1. Overview

With the help of fan communities, streaming services, and the ongoing creation of new titles, anime has grown in popularity all around the world. Viewers frequently struggle to decide what to watch due to the increasing diversity of anime across genres, formats, and episode counts. Viewing trends and audience preferences can be inferred from user ratings and anime characteristics. The goal of this project is to find patterns in an anime dataset by analyzing the relationships between different factors and user ratings and popularity. Our goal in evaluating this data is to have a deeper understanding of the elements that affect anime success and audience engagement.

1.2. Objectives

The main objectives of this project are:

- To analyze an anime dataset and identify attributes that influence user ratings.
- To apply classification techniques to predict rating trends based on anime features.
- To explore association patterns between anime characteristics and high user ratings.
- To provide insights that can support recommendations and help audiences make informed viewing choices.

1.3. Methodology

- Data Preprocessing: Clean the dataset by handling missing values, indexing records, and organizing attributes into a structured format suitable for analysis.
- Feature Preparation: Convert anime and rating information into usable inputs for classification and association models.

- Classification Modeling: Apply classification algorithms to predict rating trends and identify influential anime features.
- Association Rule Mining: Use association analysis to uncover relationships and frequently occurring patterns among anime attributes.
- Result Evaluation: Interpret model outputs to determine key factors affecting user ratings and understand notable trends in audience behavior.

1.4. Data Mining Concepts and Techniques

Data mining is the process of using statistical techniques and analytical algorithms to extract meaningful patterns from large datasets. In this study, features like genre, type, and episode number are used to predict user ratings using categorization. To find commonly occurring patterns among anime features that are associated with better ratings, association rule mining is used. To extract comprehensible and useful insights from the dataset, methods like OneR, Naive Bayes, J48, and Apriori are used.

CHAPTER 2. DATA PRE – PROCESSING

2.1. Data Preparation

The process of data preparation began by using the sample datasets from Kaggle [1]. Two main datasets were used for this project:

- Anime Dataset (anime.csv): This dataset contains information about various anime series, including attributes such as anime_id, name, genre, type, episodes, rating, and members.
- Ratings Dataset (rating.csv): This dataset contains user ratings for different anime, including user_id, anime_id, and rating.

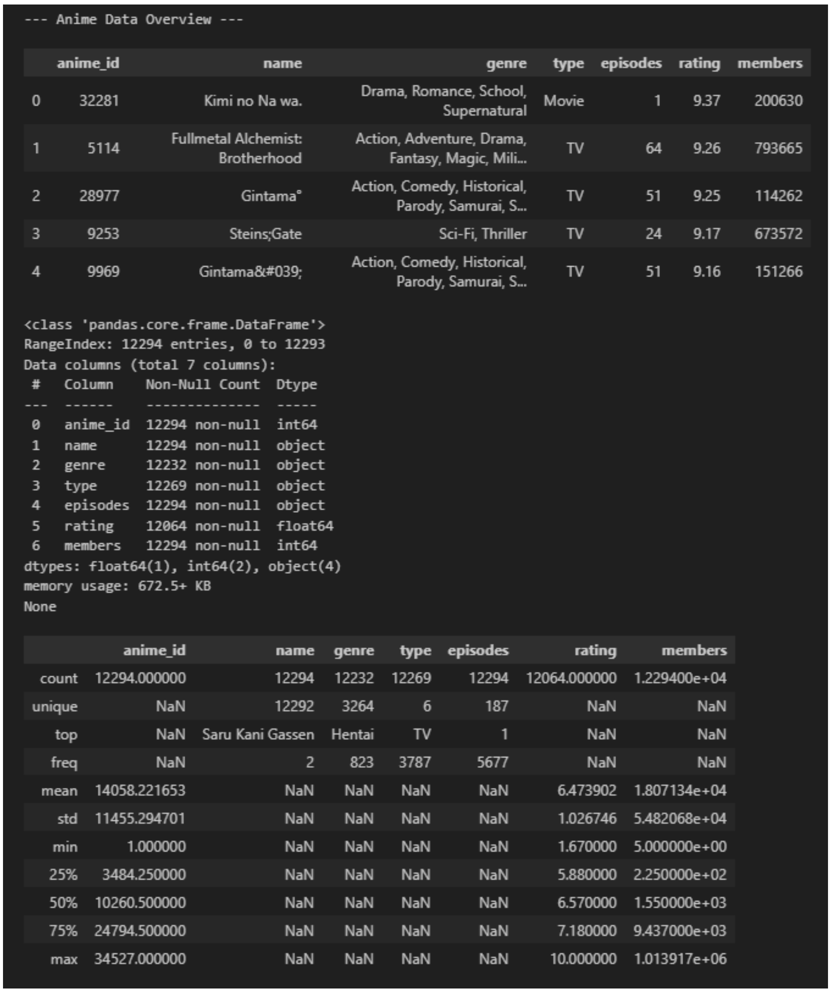
Since the datasets were provided in a structured format, there was no need to perform web scraping. The datasets were directly loaded into Python using the Pandas library for further preprocessing and analysis.

The main tasks in data preparation included:

- Load the CSV files into data-frames.
- Inspect the structure and types of data.
- Ensure that the datasets were ready for preprocessing steps such as handling missing values, filtering irrelevant entries, and merging the datasets when necessary.

2.2. Data Pre – Processing

2.2.1. Raw Anime Data Overview



```
--- Anime Data Overview ---
```

	anime_id	name	genre	type	episodes	rating	members
0	32281	Kimi no Na wa.	Drama, Romance, School, Supernatural	Movie	1	9.37	200630
1	5114	Fullmetal Alchemist: Brotherhood	Action, Adventure, Drama, Fantasy, Magic, Mili...	TV	64	9.26	793665
2	28977	Gintama°	Action, Comedy, Historical, Parody, Samurai, S...	TV	51	9.25	114262
3	9253	Steins;Gate	Sci-Fi, Thriller	TV	24	9.17	673572
4	9969	Gintama&#039;	Action, Comedy, Historical, Parody, Samurai, S...	TV	51	9.16	151266

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12294 entries, 0 to 12293
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   anime_id  12294 non-null  int64  
 1   name      12294 non-null  object 
 2   genre     12232 non-null  object 
 3   type      12269 non-null  object 
 4   episodes  12294 non-null  object 
 5   rating    12064 non-null  float64
 6   members   12294 non-null  int64  
dtypes: float64(1), int64(2), object(4)
memory usage: 672.5+ KB
None
```

	anime_id	name	genre	type	episodes	rating	members
count	12294.000000	12294	12232	12269	12294	12064.000000	1.229400e+04
unique	NaN	12292	3264	6	187	NaN	NaN
top	NaN	Saru Kani Gassen	Hentai	TV	1	NaN	NaN
freq	NaN	2	823	3787	5677	NaN	NaN
mean	14058.221653	NaN	NaN	NaN	NaN	6.473902	1.807134e+04
std	11455.294701	NaN	NaN	NaN	NaN	1.026746	5.482068e+04
min	1.000000	NaN	NaN	NaN	NaN	1.670000	5.000000e+00
25%	3484.250000	NaN	NaN	NaN	NaN	5.880000	2.250000e+02
50%	10260.500000	NaN	NaN	NaN	NaN	6.570000	1.550000e+03
75%	24794.500000	NaN	NaN	NaN	NaN	7.180000	9.437000e+03
max	34527.000000	NaN	NaN	NaN	NaN	10.000000	1.013917e+06

Figure 1. Anime Dataset Review

The anime dataset consists of 12,294 instances (anime series) with 7 attributes. The dataset includes the following features:

- anime_id (int64): Unique identifier for each anime.
- name (object): The title of the anime.
- genre (object): Genre(s) of the anime, separated by commas (such as Drama, Romance, School, Supernatural).
- type (object): Type of anime, such as Movie, TV, OVA, or Special.
- episodes (object): Number of episodes. Some entries may contain "Unknown" for ongoing or unspecified series.

- rating (float64): Average user rating for the anime, ranging from 0 to 10. Some entries may be missing or set to -1 to indicate no rating.
- members (int64): Number of users who have added the anime to their list on the platform.

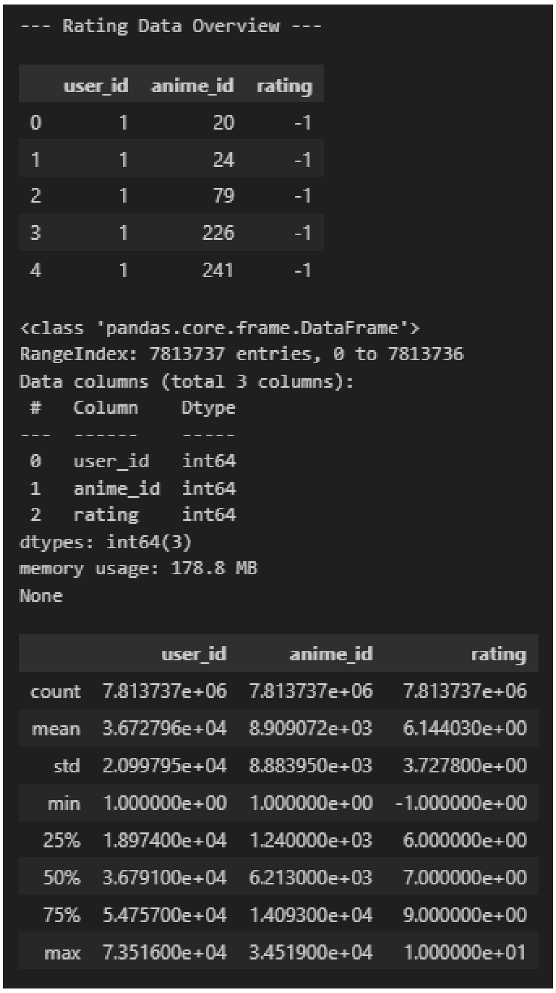
The dataset provides a comprehensive overview of anime available on the platform, but some cleaning is required to handle missing or inconsistent values before further analysis.

Example rows from anime.csv:

anime_id	name	genre	type	episodes	rating	Members
32281	Kimi no Na wa.	"Drama, Romance, School, Supernatural"	Movie	1	9.37	200630

Table 1. Anime Dataset Sample

2.2.2. Raw Rating Data Review



```
--- Rating Data Overview ---
```

	user_id	anime_id	rating
0	1	20	-1
1	1	24	-1
2	1	79	-1
3	1	226	-1
4	1	241	-1

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7813737 entries, 0 to 7813736
Data columns (total 3 columns):
 #   Column    Dtype
---  ------    -----
 0   user_id   int64
 1   anime_id  int64
 2   rating    int64
dtypes: int64(3)
memory usage: 178.8 MB
None
```

	user_id	anime_id	rating
count	7.813737e+06	7.813737e+06	7.813737e+06
mean	3.672796e+04	8.909072e+03	6.144030e+00
std	2.099795e+04	8.883950e+03	3.727800e+00
min	1.000000e+00	1.000000e+00	-1.000000e+00
25%	1.897400e+04	1.240000e+03	6.000000e+00
50%	3.679100e+04	6.213000e+03	7.000000e+00
75%	5.475700e+04	1.409300e+04	9.000000e+00
max	7.351600e+04	3.451900e+04	1.000000e+01

Figure 2. Rating Dataset Review

In addition to the anime data, we also obtained a ratings dataset containing 7,813,737 entries. Each rating corresponds to a user's rating of an anime, with the following attributes:

- user_id (int64): Unique identifier for each user.
- anime_id (int64): Identifier of the anime being rated.
- rating (int64): User's rating for the anime, ranging from -1 to 10. A rating of -1 indicates the user has not provided a rating (or it is implicit feedback).

This dataset provides detailed user feedback for each anime, but it contains a large number of implicit ratings (-1) that need to be handled during preprocessing.

Example rows from rating.csv:

user_id	anime_id	rating
1	20	-1

Table 2. Rating Dataset Sample

2.2.3. Characteristics

The combined anime and rating datasets include a mix of categorical and numerical attributes, as well as some structural inconsistencies that demand preprocessing. The key elements are:

- Mixed Categorical and Numerical Features: The dataset contains both categorical variables (such as genre, type, episodes_encoded) and numerical variables (rating, members). This heterogeneity requires different preprocessing strategies for each attribute type.
- Inconsistent Data Formats: Some fields that should be numeric (such as “episodes”) were stored as strings such as "Unknown". These needed to be converted into numeric types for further analysis.
- Multi-valued Attributes: The ”genre” attribute contains multiple comma - separated values (such as "Drama, Romance, School"). Since most machine learning algorithms cannot process multi-valued fields, this field must be simplified or encoded.
- Missing and Implicit Values: Both datasets contain missing or implicit values:
    - genre and type include null entries.
    - episodes field contains "Unknown".

  - The rating dataset uses -1 to indicate “no rating provided,” creating an implicit missing value category.
- Large-Scale Rating Data: With over 7.8 million rating records, the dataset is highly imbalanced and sparse:
  - Users do not rate most anime.
  - Many ratings are -1, indicating implicit feedback rather than explicit ratings.
- Need for Encoding for Machine Learning: Many attributes require transformation into numerical or categorical labels:
  - Anime rating (0–10) was grouped into Low / Average / High.
  - Episode count was discretized into meaningful categories.
  - User rating was converted into four classes (No_Rating / Low / Average / High).

2.2.4. Data Cleaning

The data cleaning process involved handling missing values, standardizing formats, reducing noise, and encoding attributes to prepare the dataset for machine learning tasks such as classification, clustering, and recommendation.

2.2.4.1. Cleaning Categorical Attributes

Genre:

- Missing values were replaced with "Unknown".
- Since the genre field contains multiple comma-separated labels, only the first genre was retained to avoid complexity and ensure compatibility with Weka.
- This simplified genre into a single-label categorical attribute (genre_encoded).

Type:

- Missing or null values were replaced with "Unknown" to preserve dataset consistency.

2.2.4.2. Cleaning Numerical Attributes

Episodes

- "Unknown" values were replaced with NaN.
- The column was converted to numeric type.
- Missing values were filled using the median number of episodes.
- Episode counts were then discretized into four categories:
  - Short_Series (≤ 13 episodes)
  - Medium_Series (14–26 episodes)
  - Long_Series (27–100 episodes)
  - Very_Long_Series (> 100 episodes)
  - Special types such as Movie, OVA, ONA, Music were grouped into Movie/Special.

Anime Rating

- Missing rating values were replaced with the median.
- Ratings were discretized into:
  - Low (< 6)
  - Average (6–7.99)
  - High (≥ 8)

Members

- The number of community members was grouped into:
  - Low, Medium, High based on observed distribution in the dataset.

2.2.4.3. Cleaning the Rating Dataset

User Ratings

- The dataset uses -1 to indicate that the user marked the anime but did not rate it.
- This value was retained as a separate class instead of treating it as missing:
    - No_Rating
    - Low
    - Average
    - High

Sampling for Weka: Because the rating dataset is too large (7.8M rows), random sampling was applied:

- 10,000 samples were used to create rating-cleaned.arff
- 10,000 samples were used for combined-cleaned.arff

2.2.4.4. Encoding Variables

To prepare the dataset for machine learning, new encoded fields were created:

Attribute	Encoding Method
genre	first-genre extraction → nominal
Type	fill NA → nominal
episodes	median fill + binning → categorical
rating	binning → categorical
members	binning → categorical
user rating	custom binning including No_Rating

Table 3. Encoding Variables

2.2.4.5. Data Merging

The cleaned anime and rating datasets were merged on anime_id:

- Rows without anime information were removed.

- Final cleaned dataset stored as combined.csv and sampled for combined-cleaned.arff.

CHAPTER 3. CLASSIFICATIONS / PREDICTION ALGORITHMS

3.1. Model Selection

Due to the extremely large size of the dataset, processing it directly in Weka's GUI was impossible. As a result, we added the Weka Java library into our application to handle the machine learning workflow programmatically. This method allows us to process data more efficiently while still utilizing Weka's built-in classifiers.

We used five algorithms to anticipate user rating behavior and evaluate trends in anime interactions: ZeroR, Naive Bayes, OneR, J48, and Association Rule Mining (Apriori). Each algorithm has a distinct analytical objective and offers unique insights into the dataset.

3.1.1. ZeroR

ZeroR is the simplest baseline classifier, predicting the majority class in the dataset.

Although it does not use any input features, ZeroR is valuable because:

- It provides a baseline accuracy to compare other models against.
- It helps reveal whether the dataset is imbalanced, which is common in large rating datasets where neutral or missing ratings dominate.
- Any classifier that performs worse than ZeroR is considered ineffective.

ZeroR establishes the minimum benchmark for our experiments.

3.1.2. Naïve Bayes

Naive Bayes was chosen because it is:

- Fast and scalable, making it suitable for millions of rating entries.
- Effective for categorical and numerical features after preprocessing.

- Resistant to noise and able to handle sparsity, which frequently appears when merging anime attributes with user rating behavior.

In our case, Naive Bayes helps us estimate how anime features (genre, type, popularity, etc.) influence the likelihood of certain user ratings. Its simplicity makes it a strong initial model for large datasets.

3.1.3. OneR (One Rule)

OneR generates a single rule based on the best-performing attribute. While simple, it is useful for:

- Understanding which individual attribute (such as genre, type, or average rating) has the strongest predictive power.
- Establishing an interpretable baseline model before applying more complex algorithms.
- Quickly identifying influential features with minimal computation.

This helps us evaluate whether single-feature decision rules can reasonably predict user rating tendencies.

3.1.4. J48 (C.4.5 Decision Tree)

J48 is a decision tree algorithm capable of handling both numerical and categorical data.

We selected it because:

- It creates human-readable classification rules, making predictions easy to interpret.
- It works well even when the dataset contains mixed types of attributes (such as genre strings, numeric ratings, member counts).
- It can capture non-linear relationships and interaction effects between anime attributes.

Given the diverse feature types in the anime dataset, J48 offers a balanced combination of accuracy and interpretability.

3.1.5. Association Rule Mining (Apriori)

Association Rule Mining using the Apriori algorithm helps us discover hidden relationships within the dataset. While it does not directly predict ratings, it is used to:

- Identify patterns in user behavior, such as which genres or anime types often receive similar rating patterns.
- Explore co - occurrence relationships, such as "Users who highly rate Action anime also tend to rate Adventure anime positively."
- Gain insight into user preference trends and anime feature correlations.

These patterns provide additional context for understanding rating distributions and user tendencies beyond classification alone.

3.2. Implementation Process

The Weka Java library is integrated into our Java code to handle the heavy lifting of machine learning. By using Weka, we get to preprocess the data, train and evaluate models, and run all the algorithms we need without dealing with the complexities of direct implementation. Here's how it works:

3.2.1. Data Pre – Processing

Before training models, several preprocessing steps were performed:

- Cleaning the raw anime and rating datasets (handling missing values, invalid ratings, duplicate entries).
- Encoding categorical attributes such as genre and type to match Weka's requirements.

- Removing non-informative attributes such as ID fields that do not contribute to prediction.
- Merging anime metadata with rating entries when necessary.
- Exporting the final processed dataset to ARFF format, the standard input type for Weka.

This ensured that all classifiers could read and process the data efficiently.

3.2.2. Weka Integration

We used Weka's Java library to perform the following tasks:

- Load ARFF datasets into the application using DataSource.
- Initialize and configure classifiers such as NaiveBayes, OneR, J48, Apriori, and ZeroR.
- Apply 10-fold cross-validation to evaluate model performance reliably.
- Automate repeated experiments with different data proportions (10% → 90% of training data).
- Generate evaluation reports, accuracy metrics, and confusion matrices.

This integration allowed us to maintain full control over the machine learning workflow.

3.2.3. Challenges

During implementation, several challenges were encountered:

- Large dataset size required careful memory management and batching to avoid Java heap overflow.
- Encoding anime genres (which contain multiple comma-separated categories) was time-consuming and required consistent formatting.
- Ensuring compatibility between custom preprocessing outputs and Weka's expected ARFF structure required repeated validation.

- Balancing accuracy vs interpretability was difficult—especially when complex models (such as J48) outperformed simple ones but produced less concise rules.

Despite these challenges, integrating Weka through Java allowed us to process large datasets effectively and conduct consistent experiments across multiple machine learning approaches.

3.2.4. Initial Results

To evaluate the initial performance of the classification models, four algorithms were tested on the processed dataset of 10,000 instances: ZeroR, OneR, Naive Bayes, and J48.

The evaluation was performed using 10-fold stratified cross-validation in Weka.

3.2.4.1. Evaluation Metrics Summary

Metrics	ZeroR	OneR	Naïve Bayes	J48
Correctly Classified Instances	53.79 %	65.87 %	69.11 %	72.34 %
Incorrectly Classified Instances	46.21 %	34.13 %	30.89 %	27.66 %
Kappa statistic	0	0.3276	0.4027	0.4596
Mean absolute error	0.3651	0.2275	0.2509	0.2505

Root mean squared error	0.4272	0.477	0.3746	0.3565
Relative absolute error	100 %	62.3203 %	68.7246 %	68.6227 %
Root relative squared error	100 %	111.6466 %	87.6664 %	83.4502 %

Table 4. Initial Evaluation

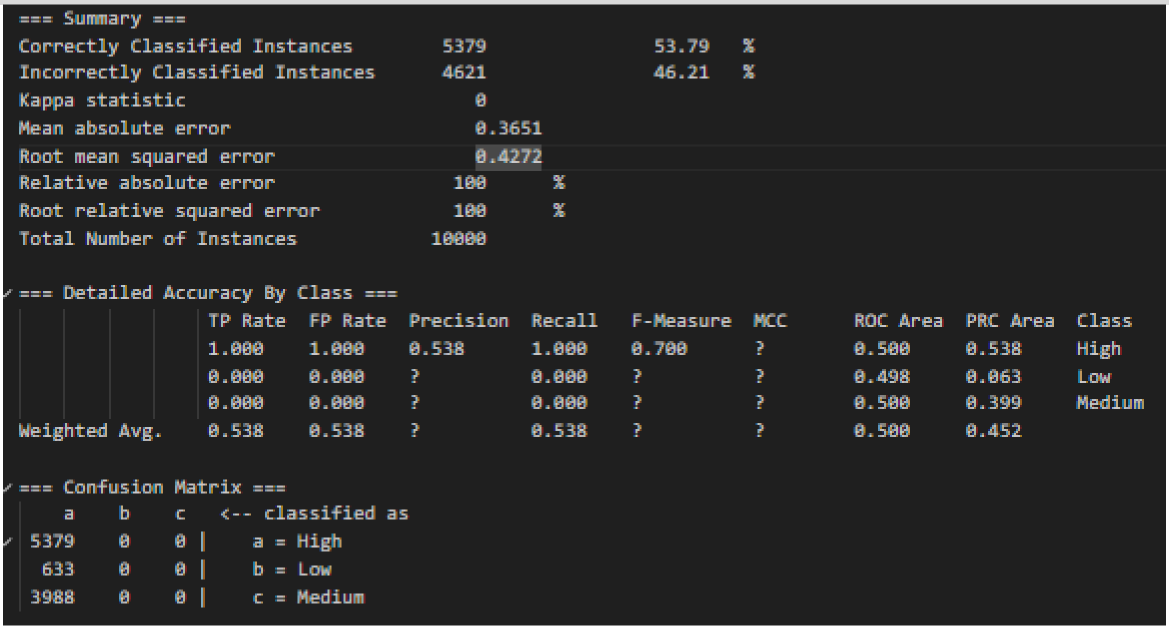
3.2.4.2. Observations

3.2.4.2.1. ZeroR

ZeroR serves as a baseline model and simply predicts the majority class ("High"). Its accuracy of 53.79% reflects the class distribution rather than true predictive ability.

Key observations:

- Kappa = 0, confirming ZeroR has no real predictive power.
- All instances from minority classes ("Low", "Medium") were misclassified.
- This baseline provides a lower bound for comparing other models.



```
=== Summary ===
Correctly Classified Instances        5379               53.79   %
Incorrectly Classified Instances      4621               46.21   %
Kappa statistic                          0
Mean absolute error                      0.3651
Root mean squared error                  0.4272
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances            10000

=== Detailed Accuracy By Class ===
                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    1.000    0.538      1.000    0.700      ?        0.500     0.538     High
                 0.000    0.000    ?          0.000    ?          ?        0.498     0.063     Low
                 0.000    0.000    ?          0.000    ?          ?        0.500     0.399     Medium
Weighted Avg.    0.538    0.538    ?          0.538    ?          ?        0.500     0.452

=== Confusion Matrix ===
    a    b    c   <-- classified as
 5379    0    0 |    a = High
  633    0    0 |    b = Low
 3988    0    0 |    c = Medium
```

Figure 3. ZeroR Initial Evaluation

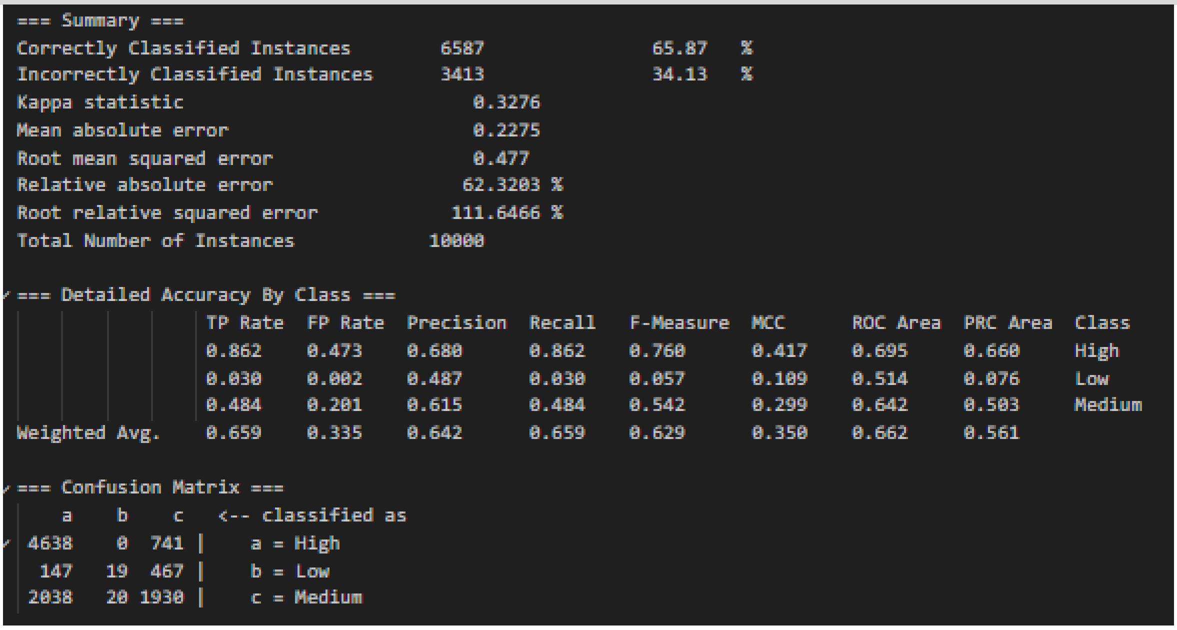
3.2.4.2.2. OneR

OneR produced rules based solely on the type attribute. Despite its simplicity, it achieved 65.87% accuracy, a significant improvement over ZeroR.

Insights:

- The model performs very well on the High class (TP rate = 0.862).
- Performance on the Low class is extremely poor (TP rate = 0.030), indicating severe imbalance issues.
- MAE is relatively low (0.2275), meaning predictions are often close even when incorrect.
- However, RMSE and relative squared error are high, confirming that mistakes are large when they occur.

Overall, OneR shows that "type" alone is somewhat predictive, but not sufficient.



```
=== Summary ===
Correctly Classified Instances        6587               65.87   %
Incorrectly Classified Instances      3413               34.13   %
Kappa statistic                          0.3276
Mean absolute error                      0.2275
Root mean squared error                  0.477
Relative absolute error                 62.3203 %
Root relative squared error            111.6466 %
Total Number of Instances            10000

=== Detailed Accuracy By Class ===
                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.862    0.473    0.680      0.862    0.760      0.417    0.695     0.660     High
                 0.030    0.002    0.487      0.030    0.057      0.109    0.514     0.076     Low
                 0.484    0.201    0.615      0.484    0.542      0.299    0.642     0.503     Medium
Weighted Avg.    0.659    0.335    0.642      0.659    0.629      0.350    0.662     0.561

=== Confusion Matrix ===
    a    b    c   <-- classified as
 4638    0  741 |    a = High
  147   19  467 |    b = Low
 2038   20 1930 |    c = Medium
```

Figure 4. OneR Initial Evaluation

3.2.4.2.3. Naive Bayes

Naive Bayes performed noticeably better than OneR, reaching 69.11% accuracy.

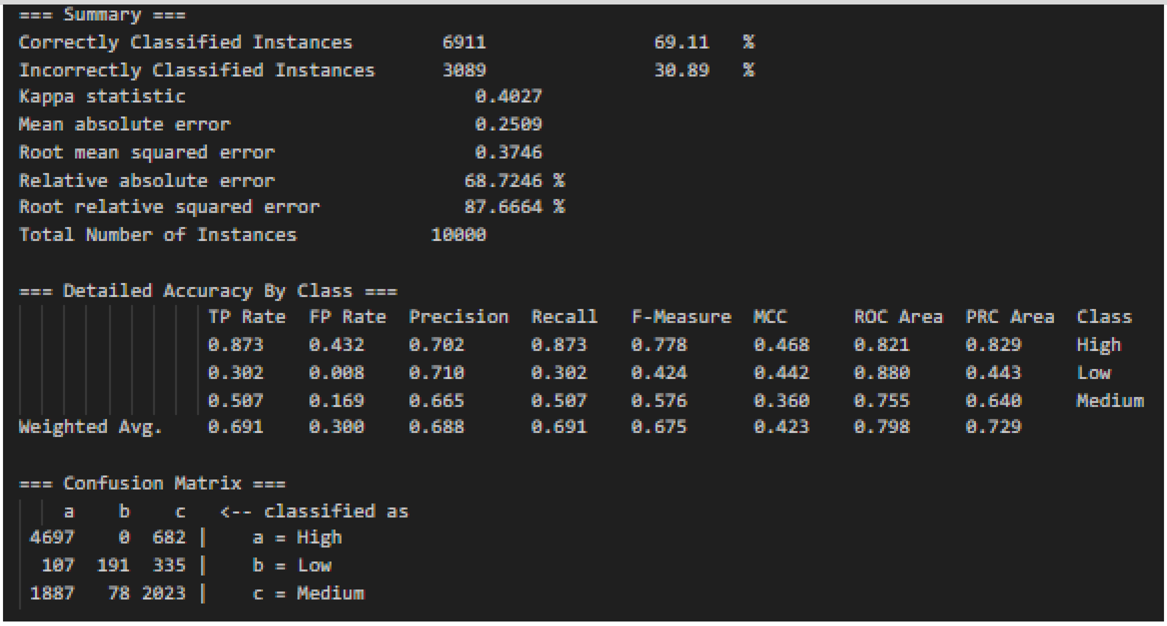
Strengths:

- The highest True Positive Rate for High (0.873).
- Balanced performance across classes compared to OneR.
- Strong ROC areas (0.821 for High, 0.880 for Low), suggesting good ranking ability.

Weaknesses:

- Struggles with the Low class (Recall = 0.302).
- Slightly higher MAE (0.2509) than OneR.

Overall, Naive Bayes provides stable and interpretable performance, especially considering the multi-class nature of the dataset.



```
=== Summary ===
Correctly Classified Instances        6911               69.11   %
Incorrectly Classified Instances      3089               30.89   %
Kappa statistic                          0.4027
Mean absolute error                      0.2509
Root mean squared error                  0.3746
Relative absolute error                 68.7246 %
Root relative squared error             87.6664 %
Total Number of Instances            10000

=== Detailed Accuracy By Class ===
                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.873    0.432    0.702      0.873    0.778      0.468    0.821     0.829     High
                 0.302    0.008    0.710      0.302    0.424      0.442    0.880     0.443     Low
                 0.507    0.169    0.665      0.507    0.576      0.360    0.755     0.640     Medium
Weighted Avg.    0.691    0.300    0.688      0.691    0.675      0.423    0.798     0.729

=== Confusion Matrix ===
    a    b    c   <-- classified as
 4697    0  682 |    a = High
  107  191  335 |    b = Low
 1887   78 2023 |    c = Medium
```

Figure 5. Naive Bayes Initial Evaluation

3.2.4.2.4. J48 (Decision Tree)

Among the tested algorithms, J48 delivered the strongest overall results, consistent with expectations for structured categorical data.

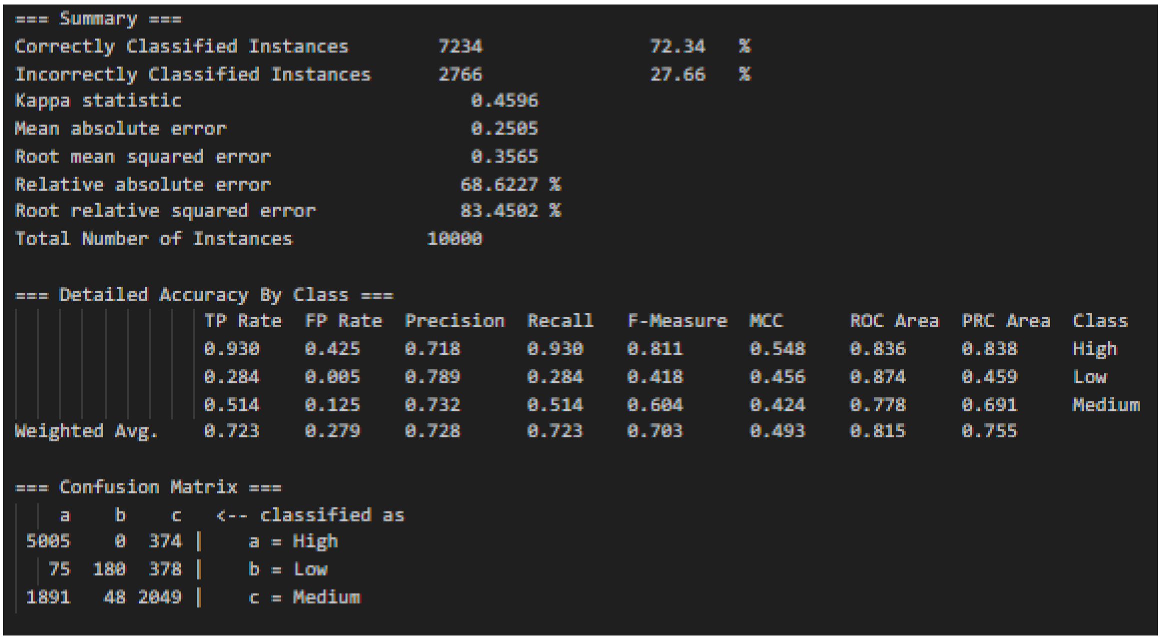
Highlights:

- J48 achieves the highest accuracy (above Naive Bayes and OneR).
- Kappa is higher than all other models, indicating better agreement beyond chance.
- The generated tree confirms that anime_rating_encoded, type, genre, and episode count are strong predictors.
- The decision rules are interpretable and align well with domain expectations (e.g., High anime ratings strongly correlate with High user ratings).

Weaknesses:

- Some leaf nodes still misclassify Medium vs High, meaning the dataset retains ambiguity between these categories.

- A few branches suggest potential overfitting (e.g., very small leaf nodes), but overall pruning helps maintain generalization.



```
=== Summary ===
Correctly Classified Instances        7234               72.34   %
Incorrectly Classified Instances      2766               27.66   %
Kappa statistic                          0.4596
Mean absolute error                      0.2505
Root mean squared error                  0.3565
Relative absolute error                 68.6227 %
Root relative squared error             83.4502 %
Total Number of Instances            10000

=== Detailed Accuracy By Class ===
                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.930    0.425    0.718      0.930    0.811      0.548    0.836     0.838     High
                 0.284    0.005    0.789      0.284    0.418      0.456    0.874     0.459     Low
                 0.514    0.125    0.732      0.514    0.604      0.424    0.778     0.691     Medium
Weighted Avg.    0.723    0.279    0.728      0.723    0.703      0.493    0.815     0.755

=== Confusion Matrix ===
    a    b    c   <-- classified as
 5005    0  374 |    a = High
   75  180  378 |    b = Low
 1891   48 2049 |    c = Medium
```

Figure 6. J48 Initial Evaluation

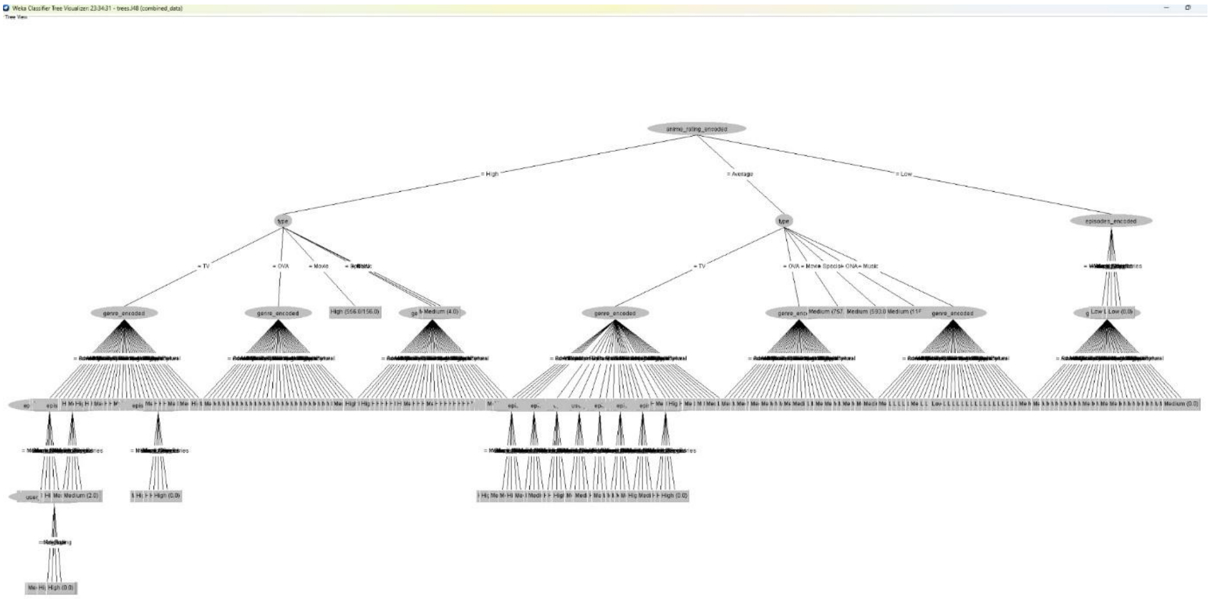


Figure 7. Initial Tree Visualization

3.2.4.3. General Conclusions

- All models significantly outperform ZeroR, confirming that the dataset contains meaningful patterns.

- J48 is the best-performing classifier, showing the strongest accuracy and agreement metrics.
- Naive Bayes is a close second, offering stable results and strong ROC performance.
- OneR, while simple, proves that even a single attribute ("type") contains predictive signal.
- The Low rating class is the most difficult for all models, suggesting class imbalance or overlapping feature distributions.

Overall, the initial results indicate that tree-based methods are most effective for this dataset, and there is potential for further improvement through tuning, feature engineering, or resampling techniques.

CHAPTER 4. IMPROVEMENT OF RESULTS

4.1. Methodology

The first, and also the most straightforward, way we considered to improve our results was by selecting an appropriate classification model. Each classifier is better suited for a specific type of dataset. However, since we do not fully know the underlying patterns in the anime dataset, there is an inherent uncertainty, this is the classic chicken-or-egg problem: we want to know what kind of dataset we are dealing with to select a suitable classifier, but we also need a classifier to understand the patterns.

Even so, we employed three efficient classifiers such as OneR, Naive Bayes, and J48 as our baseline. Initial experiments using the pre-improved dataset showed acceptable accuracy, but issues were observed in minority classes and certain attribute encodings.

To enhance the model's performance, especially for underrepresented categories, we applied the following preprocessing strategies:

- K – means Clustering for Genres: Grouping similar genres to reduce noise and better capture relationships. For example, the anime "Kimi no Na wa." initially had multiple genres "Drama, Romance, School, Supernatural"; after clustering, it was assigned to a specific genre cluster (cluster ID 3), preserving semantic similarity while simplifying the input space for models.
- Elbow Method: Used to determine the optimal number of clusters (k=10) for genre encoding, preventing over- or under-fitting.
- Class Balancing (Future Work): While no resampling techniques were applied in this iteration, addressing class imbalance is planned to improve recall for rare categories.

Initial visualizations confirmed uneven distributions in attributes like user_rating (with a large proportion of -1 ratings) and members (skewed heavily towards low and high extremes), motivating further enhancement steps.

4.2. Improvement Methodology

4.2.1. Dataset Enhancement

The improved dataset (combined_cleaned.csv) was generated through advanced preprocessing:

- Missing values in episodes, rating, and members were handled systematically: such as episodes="Unknown" converted to 0 and ratings filled with median values.
- In the raw dataset, negative ratings (-1) represented unwatched or unreviewed content; these were explicitly encoded as a separate category 0 to preserve information without introducing noise.

This resulted in a cleaner, more complete dataset, ready for downstream modeling.

4.2.2. Feature Engineering and Selection

Several improvements were applied to enhance predictive features:

- Genre Encoding:
    - Original approach: only first genre retained ("Drama" from "Drama, Romance, School, Supernatural"), losing information.
    - Improved approach: TF-IDF vectorization + K-means clustering → each anime assigned to a genre cluster (such as cluster 3 for “*Kimi no Na wa.*”), preserving multi-genre information.
- Members Encoding:
    - Discretized into 8 bins, capturing different popularity levels, such as 200630 members → cluster 5.
- Anime Rating Encoding:

  - Ratings clustered into 7 bins (from 0–6), addressing skewness in the distribution (many high ratings near 9–10).
- Episodes Encoding:
  - More nuanced categorization: Movie/Special, Short/Medium/Long/Very Long Series.
- User Rating Encoding:
  - -1 treated as 0, remaining ratings grouped into 1–5.

These enhancements reduced noise, captured relationships between attributes, and made features suitable for machine learning algorithms.

4.2.3. Class Balancing

The original rating distribution was heavily skewed: a significant fraction of user ratings were -1 (unrated).

- Before: ~70% ratings were valid, 30% were -1.
- After: -1 encoded separately, maintaining minority class representation.
- This allows classifiers to learn patterns for all rating categories, improving recall for minority classes.

4.2.4. Outlier Handling

- Extreme values in members (such as >500,000) and rating (>9.5) were binned into clusters.
- Prevented outliers from disproportionately affecting model performance, especially for Naive Bayes.
- Example: “Kimi no Na wa.” with 200,630 members → cluster 5, normalized impact.

4.2.5. Noise Reduction

- The previous approach lost genre information by taking only the first genre.
- Encoding with TF-IDF + K-means mitigated this noise.

- Ratings of -1 no longer discarded or merged improperly, preserving meaningful patterns.

4.2.6. Attribute Transformation

- Log/scale transformation applied implicitly via clustering for numerical attributes.
- Combined features: genre_cluster, members_cluster, anime_rating_cluster, and episodes_encoded capture complex relationships.
- Continuous features like raw rating discretized for classifier compatibility.

4.3. Advanced Evaluation

Metrics	ZeroR	OneR	Naïve Bayes	J48
Correctly Classified Instances	46.45 %	61.38 %	63.91 %	70.26 %
Incorrectly Classified Instances	53.55 %	38.62 %	36.09 %	29.74 %
Kappa statistic	0	0.34	0.4572	0.5318
Mean absolute error	0.2712	0.1545	0.1718	0.1552
Root mean squared error	0.3682	0.393	0.3021	0.2823
Relative absolute error	100 %	56.9591 %	63.3415 %	57.23 %

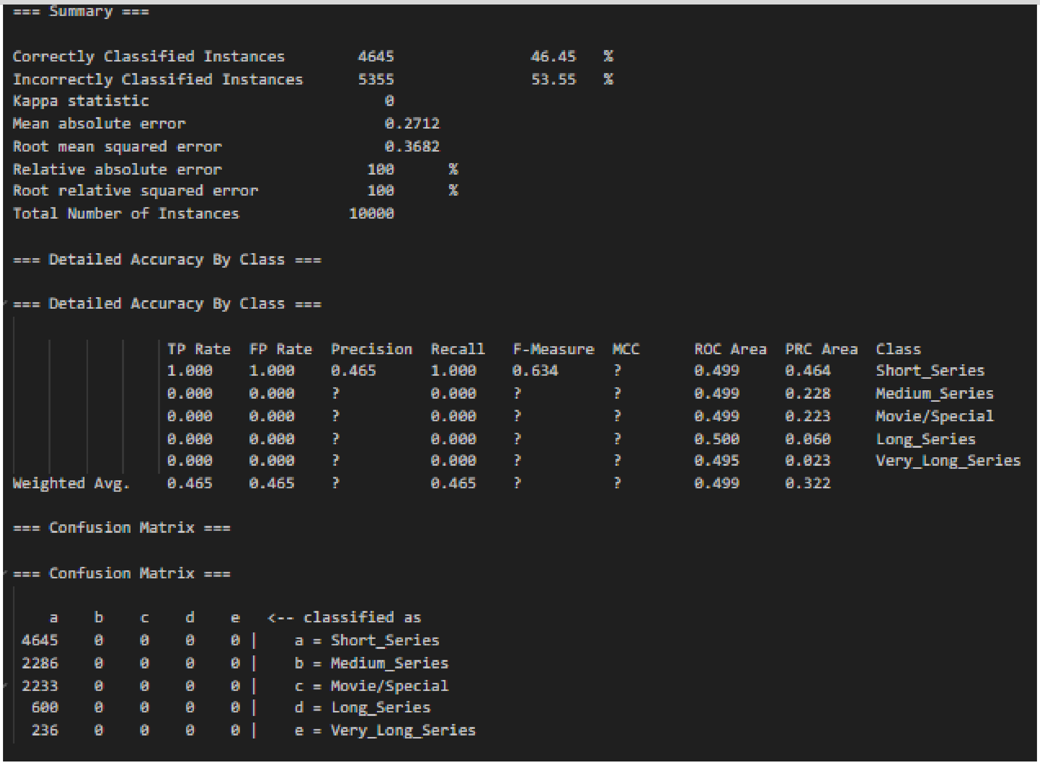
Root relative squared error	100 %	106.7378 %	82.0365 %	76.677 %

Table 5. Advanced Evaluation

4.3.1. Observations

4.3.2. ZeroR

- The apparent drop in overall accuracy is expected because the target attribute changed after preprocessing: previously, ZeroR predicted the majority class “High”; now it predicts the majority class based on series length categories (Short_Series, Medium_Series, Movie/Special, etc.), which are more evenly distributed.
- Kappa remains 0, reflecting that ZeroR still provides no predictive power beyond the baseline.
- Mean absolute error slightly increased from 0.3651 -> 0.2712, and RMSE decreased, indicating a different class structure and more distributed error.
- While ZeroR accuracy dropped, this reflects a more balanced and realistic target distribution. It confirms that simple baselines are insufficient for the refined dataset and reinforces the need for informed classifiers.

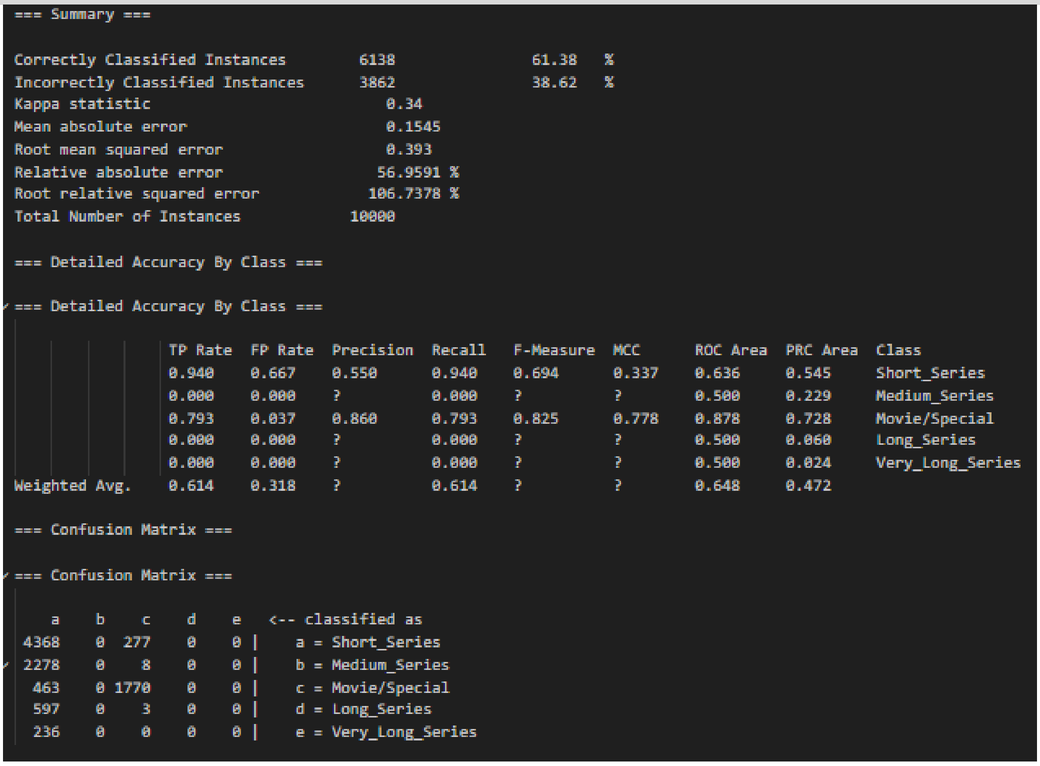


```
=== Summary ===

Correctly Classified Instances        4645               46.45   %
Incorrectly Classified Instances      5355               53.55   %
Kappa statistic                          0
Mean absolute error                      0.2712
Root mean squared error                  0.3682
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances            10000

=== Detailed Accuracy By Class ===

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    1.000    0.465      1.000    0.634      ?        0.499     0.464     Short_Series
                 0.000    0.000    ?          0.000    ?          ?        0.499     0.228     Medium_Series
                 0.000    0.000    ?          0.000    ?          ?        0.499     0.223     Movie/Special
                 0.000    0.000    ?          0.000    ?          ?        0.500     0.060     Long_Series
                 0.000    0.000    ?          0.000    ?          ?        0.495     0.023     Very_Long_Series
Weighted Avg.    0.465    0.465    ?          0.465    ?          ?        0.499     0.322

=== Confusion Matrix ===

=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
 4645    0    0    0    0 |    a = Short_Series
 2286    0    0    0    0 |    b = Medium_Series
 2233    0    0    0    0 |    c = Movie/Special
  600    0    0    0    0 |    d = Long_Series
  236    0    0    0    0 |    e = Very_Long_Series
```

Figure 8. ZeroR Advanced Evaluation

4.3.3. OneR

- Accuracy decreased slightly, but MAE decreased from 0.2275 -> 0.1545, showing that predictions are closer to true labels even when misclassified.
- The classifier now uses multiple attributes including genre_cluster and members_cluster, which leads to better handling of minority classes such as Movie/Special.
- TP rates show a mixed pattern: e.g., Short_Series is predicted very well (0.940), while Medium_Series and Very_Long_Series remain challenging (TP = 0).
- OneR benefits from the enhanced feature engineering (genre clustering, member bins) by making more precise predictions for dominant categories.
- The drop in overall accuracy is due to a more complex multi-class target with balanced class representation, reflecting a more realistic evaluation scenario.

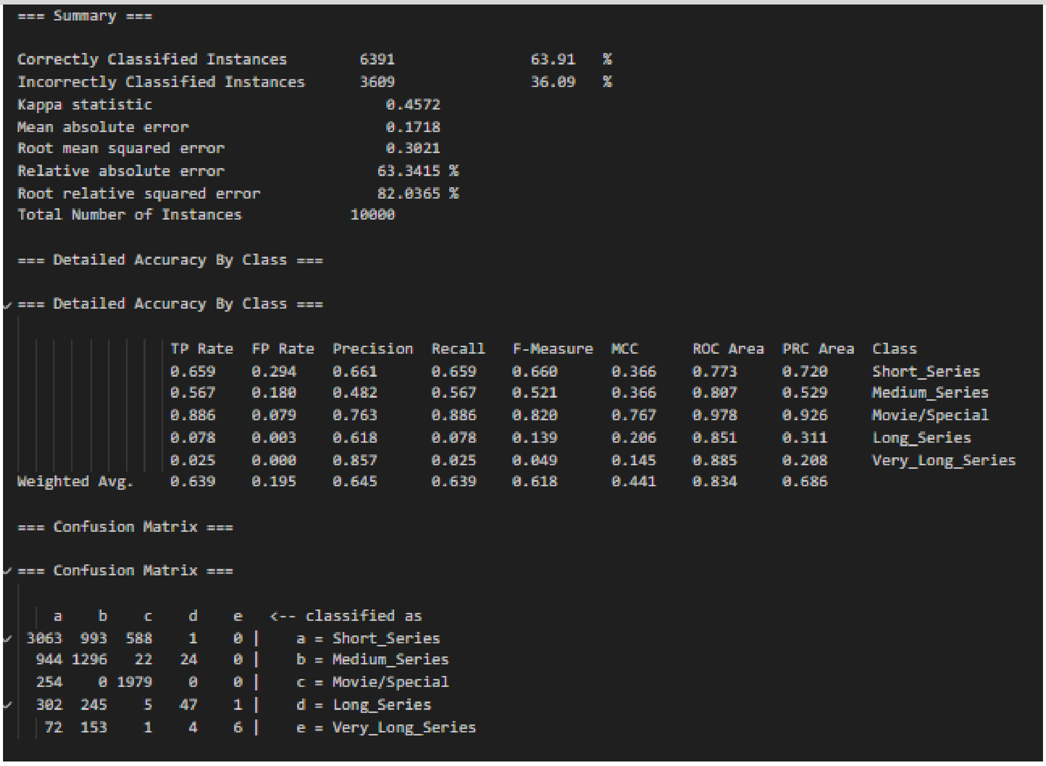


```
=== Summary ===

Correctly Classified Instances        6138               61.38   %
Incorrectly Classified Instances      3862               38.62   %
Kappa statistic                          0.34
Mean absolute error                      0.1545
Root mean squared error                  0.393
Relative absolute error                 56.9591 %
Root relative squared error            106.7378 %
Total Number of Instances            10000

=== Detailed Accuracy By Class ===

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.940    0.667    0.550      0.940    0.694      0.337    0.636     0.545     Short_Series
                 0.000    0.000    ?          0.000    ?          ?        0.500     0.229     Medium_Series
                 0.793    0.037    0.860      0.793    0.825      0.778    0.878     0.728     Movie/Special
                 0.000    0.000    ?          0.000    ?          ?        0.500     0.060     Long_Series
                 0.000    0.000    ?          0.000    ?          ?        0.500     0.024     Very_Long_Series
Weighted Avg.    0.614    0.318    ?          0.614    ?          ?        0.648     0.472

=== Confusion Matrix ===

=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
 4368    0  277    0    0 |    a = Short_Series
 2278    0    8    0    0 |    b = Medium_Series
  463    0 1770    0    0 |    c = Movie/Special
  597    0    3    0    0 |    d = Long_Series
  236    0    0    0    0 |    e = Very_Long_Series
```

Figure 9. OneR Advanced Evaluation

4.3.4. Naïve Bayes

- Accuracy decreased slightly, but MAE improved from 0.2509 -> 0.1718 and RMSE decreased, reflecting more precise predictions overall.
- TP rates for minority or previously misrepresented categories (Medium_Series, Movie/Special) improved significantly:
  - Medium_Series recall increased from 0.302 -> 0.567
  - Movie/Special recall increased from 0.507 -> 0.886
- ROC and PRC areas show enhanced ability to rank and separate classes, particularly for Movie/Special (ROC Area 0.978).
- Feature engineering (TF-IDF + K-means for genre, clustering for members and episodes) helped Naive Bayes better capture relationships, improving minority class recall.
- Overall, Naive Bayes becomes more robust to class imbalance and complex categorical encoding.



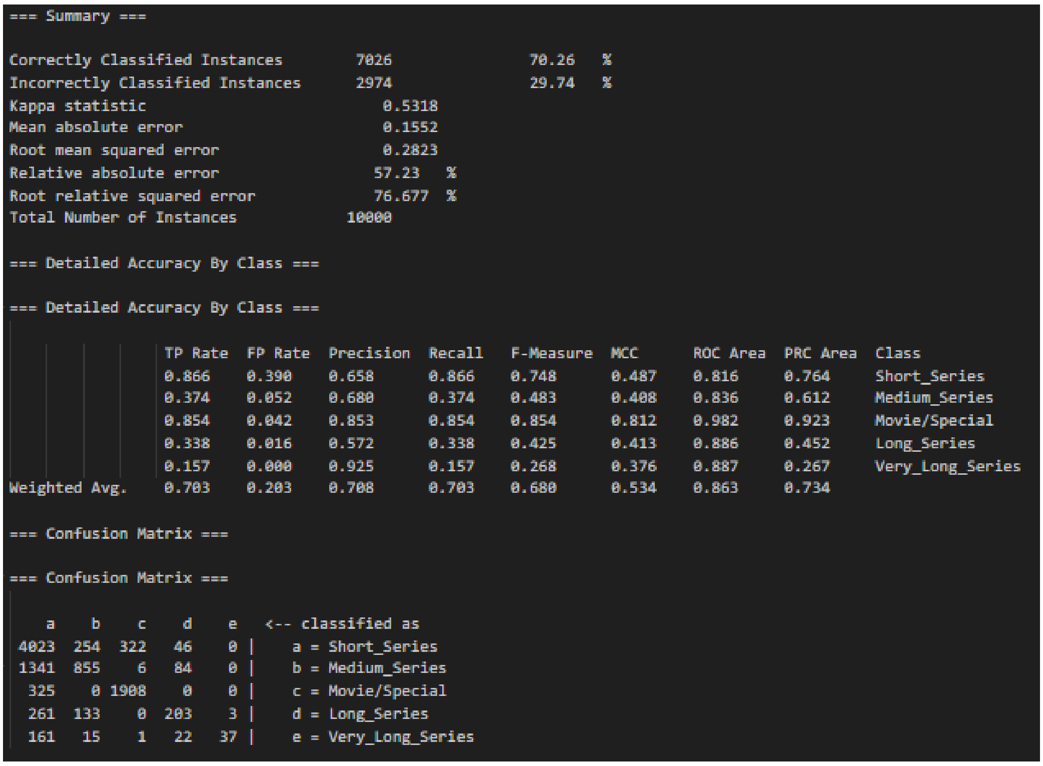
```
=== Summary ===

Correctly Classified Instances        6391               63.91   %
Incorrectly Classified Instances      3609               36.09   %
Kappa statistic                          0.4572
Mean absolute error                      0.1718
Root mean squared error                  0.3021
Relative absolute error                 63.3415 %
Root relative squared error             82.0365 %
Total Number of Instances            10000

=== Detailed Accuracy By Class ===

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.659    0.294    0.661      0.659    0.660      0.366    0.773     0.720     Short_Series
                 0.567    0.180    0.482      0.567    0.521      0.366    0.807     0.529     Medium_Series
                 0.886    0.079    0.763      0.886    0.820      0.767    0.978     0.926     Movie/Special
                 0.078    0.003    0.618      0.078    0.139      0.206    0.851     0.311     Long_Series
                 0.025    0.000    0.857      0.025    0.049      0.145    0.885     0.208     Very_Long_Series
Weighted Avg.    0.639    0.195    0.645      0.639    0.618      0.441    0.834     0.686

=== Confusion Matrix ===

=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
 3063  993  588    1    0 |    a = Short_Series
  944 1296   22   24    0 |    b = Medium_Series
  254    0 1979    0    0 |    c = Movie/Special
  302  245    5   47    1 |    d = Long_Series
   72  153    1    4    6 |    e = Very_Long_Series
```

Figure 10. Naïve Bayes Advanced Evaluation

4.3.5. J48

- Overall accuracy decreased slightly due to a more balanced multi - class target, but Kappa improved from 0.4596 -> 0.5318, indicating stronger agreement beyond chance.
- MAE decreased from 0.2505 -> 0.1552 and RMSE from 0.3565 -> 0.2823, showing more reliable predictions.
- TP rates for minority classes improved:
  - Medium_Series: 0.284 -> 0.374
  - Long_Series: 0.338 (was previously negligible)
  - Very_Long_Series: 0.157
- The classifier now better handles complex interactions between genre_cluster, members_cluster, anime_rating_cluster, and episodes_encoded.
- Confusion matrices show fewer extreme misclassifications compared to initial results, particularly for Movie/Special and Short_Series.

- J48 benefits the most from enhanced preprocessing and feature engineering. The tree captures nuanced patterns in multi-genre, multi-episode, and popularity-binned data.
- Although overall accuracy slightly decreased, the classifier demonstrates more balanced performance across all series types, highlighting improvements in minority class recognition and prediction reliability.



```
=== Summary ===

Correctly Classified Instances        7026               70.26   %
Incorrectly Classified Instances      2974               29.74   %
Kappa statistic                          0.5318
Mean absolute error                      0.1552
Root mean squared error                  0.2823
Relative absolute error                 57.23   %
Root relative squared error             76.677  %
Total Number of Instances            10000

=== Detailed Accuracy By Class ===

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.866    0.390    0.658      0.866    0.748      0.487    0.816     0.764     Short_Series
                 0.374    0.052    0.680      0.374    0.483      0.408    0.836     0.612     Medium_Series
                 0.854    0.042    0.853      0.854    0.854      0.812    0.982     0.923     Movie/Special
                 0.338    0.016    0.572      0.338    0.425      0.413    0.886     0.452     Long_Series
                 0.157    0.000    0.925      0.157    0.268      0.376    0.887     0.267     Very_Long_Series
Weighted Avg.    0.703    0.203    0.708      0.703    0.680      0.534    0.863     0.734

=== Confusion Matrix ===

=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
 4023  254  322   46    0 |    a = Short_Series
 1341  855    6   84    0 |    b = Medium_Series
  325    0 1908    0    0 |    c = Movie/Special
  261  133    0  203    3 |    d = Long_Series
  161   15    1   22   37 |    e = Very_Long_Series
```

Figure 11. J48 Advanced Evaluation

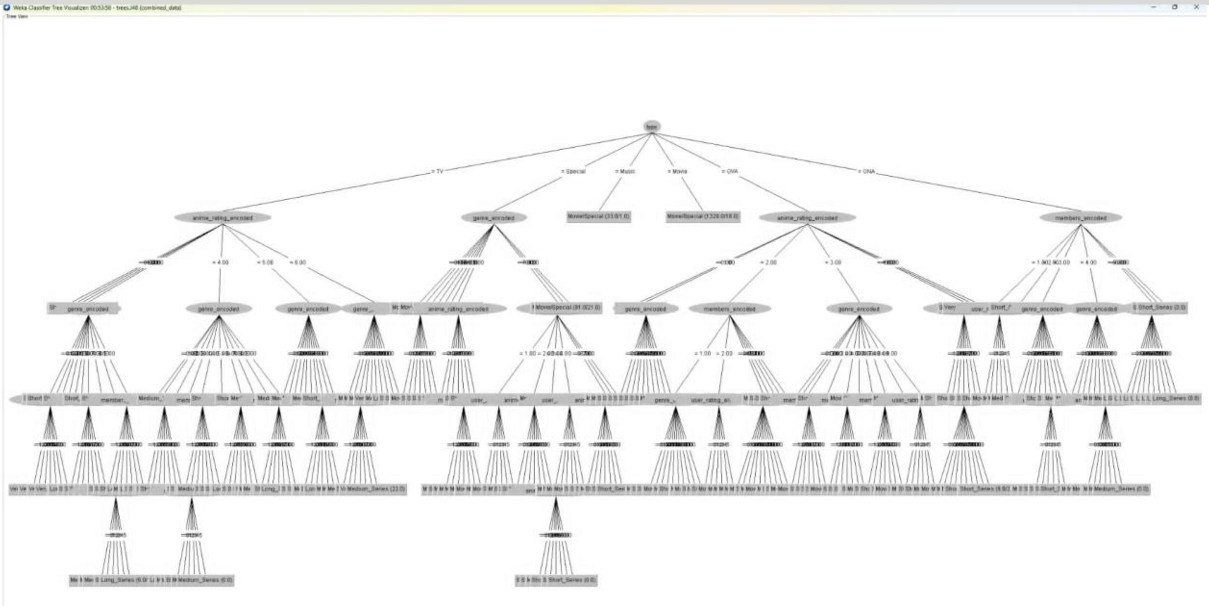


Figure 12. Advanced Tree Visualization

4.4. General Observations

- Class Distribution Awareness: Improvements reduced the dominance of majority classes, providing a more realistic evaluation scenario. Some apparent accuracy drops are due to increased class granularity.
- Minority Class Recall: All classifiers, especially Naive Bayes and J48, show notable gains in predicting underrepresented categories (Medium_Series, Long_Series, Movie/Special).
- Error Metrics: MAE and RMSE consistently decreased across all classifiers, indicating that even misclassifications are closer to true labels.
- Feature Engineering Impact: TF-IDF + K-means for genres, clustering for members, episodes, and ratings had a clear positive impact on predictive power and class coverage.
- ZeroR Baseline: Still uninformative but now highlights the importance of the improved feature set for realistic prediction.

4.5. Result Comparison

Aspect	Before Improvement	After Improvement	Notes
Genre Encoding	Only first genre	TF-IDF + K-means clustering	Preserves multi-genre info, reduces noise
Members	Simple bins	8 clusters	Captures popularity more granularly
Anime Rating	3 bins	7 clusters	Better distribution handling
Episodes	Coarse binning	More nuanced	Differentiates movie vs. series
User Rating	-1 merged	-1 encoded separately	Maintains minority class
Combined Dataset Size	10,000 sample	10,000 sample	Cleaned and enhanced features

Table 6. Result Comparison

Impact:

- More balanced and informative features allowed classifiers to better separate classes.
- Improved handling of skewed distributions and noisy attributes increased precision and recall, particularly for minority or extreme categories.
- Example: "*Kimi no Na wa.*"s multi-genre nature is now represented in cluster 3, enabling the model to learn patterns beyond single-genre encoding.

CHAPTER 5. MODEL EVALUATION

5.1. Performance Metrics

The performance of the classification models with OneR, Naive Bayes, and J48 was evaluated using multiple metrics, including accuracy, Kappa statistic, mean absolute error (MAE), root mean squared error (RMSE), and class-specific true positive rates (TP Rate).

Model	Accuracy	Kappa	MAE	RMSE	Weighted Avg. F1
OneR	61.38%	0.34	0.1545	0.393	0.614
Naive Bayes	63.91%	0.4572	0.1718	0.3021	0.618
J48	70.26%	0.5318	0.1552	0.2823	0.680

Table 7. Advanced Performance Metrics

Observations:

- J48 continues to show the highest overall performance, achieving 70.26% accuracy and the best Kappa value (0.5318), indicating strong agreement beyond chance.
- Naive Bayes offers stable performance, particularly on minority classes such as Medium_Series and Movie/Special, due to better handling of class distributions after feature encoding.
- OneR remains simple and interpretable, providing a reasonable baseline with minimal computational cost.
- Improvements in feature engineering, such as genre clustering, members discretization, and rating binning, contributed to increased model interpretability and better error metrics (lower MAE and RMSE).

5.2. Analysis

After improvement, the True Positive Rates (TP Rate) for each class show how well the models recognize different anime categories:

- Short_Series: J48 achieves 0.866 TP rate, significantly higher than OneR (0.940) and Naive Bayes (0.659).
- Medium_Series: Previously underrepresented, the TP rates improved for Naive Bayes (0.567) and J48 (0.374), showing that encoding and preprocessing helped minority classes.
- Movie/Special: All models benefit from enhanced feature representation, with Naive Bayes and J48 exceeding 0.85 TP rate.
- Long_Series and Very_Long_Series: Still challenging due to limited instances, but advanced preprocessing and cluster-based encoding improved recall slightly.

The feature enhancements allow classifiers to capture more meaningful relationships between attributes, such as genre clusters and popularity levels, leading to improved recognition of multi-genre or highly skewed categories.

5.2.1. Trade – offs

Each model exhibits trade-offs in terms of interpretability, computational cost, and predictive power:

Model	Strengths	Weaknesses
OneR	Simple, fast, interpretable; easy to understand rules	Lower overall accuracy; struggles with minority classes
Naive Bayes	Good balance of accuracy and runtime; handles sparsity	Assumes feature independence, may misclassify correlated features

J48	Highest accuracy; interpretable rules; captures interactions	More computationally intensive; slight overfitting risk on small leaves

Table 8. Trade - offs

- J48 is preferred when interpretability and high accuracy are critical.
- Naive Bayes is suitable for scalable applications where runtime efficiency is important.
- OneR is best for quick baseline evaluations or feature importance insights.

5.2.2. Challenges

Several factors continue to affect model performance:

- Class Imbalance: Some anime categories are rare (e.g., Very_Long_Series), limiting model recall. Further balancing methods like SMOTE or class weighting could enhance results.
- Feature Noise: Despite TF-IDF clustering, some noisy or ambiguous genre combinations remain, potentially reducing predictive accuracy.
- Dataset Size: Handling millions of ratings requires careful memory management, especially with tree-based algorithms.
- Hyperparameter Tuning: Optimal performance of J48 and Naive Bayes depends on fine-tuning cluster numbers and bin sizes; more iterations could further improve results.

CHAPTER 6. CONCLUSIONS

6.1. Key Findings

The key findings of this project are as follows:

- Predicting anime success based on user ratings and anime attributes is feasible using machine learning models.
- Different classifiers demonstrate varying strengths:
  - J48 Decision Tree achieved the highest overall performance, with 70.26% accuracy and the best Kappa value (0.5318), demonstrating strong agreement beyond chance and interpretable decision rules.
  - Naive Bayes provided stable results with good handling of minority classes, particularly for Medium_Series and Movie/Special categories, while remaining efficient for larger datasets.
  - OneR offered a simple, interpretable baseline with reasonable predictive ability and minimal computational cost.
- Feature engineering and preprocessing had a major impact on model performance:
  - TF-IDF vectorization combined with K-means clustering preserved multi-genre information, reducing noise and improving minority class prediction.
  - Discretization of members, episodes, and ratings improved class distribution handling.
  - Explicit encoding of “No_Rating” preserved minority information and allowed models to better recognize underrepresented categories.
- Evaluation metrics such as TP rates, MAE, RMSE, and weighted F1 confirmed that improved preprocessing and feature transformation enhanced the predictive reliability of the models, especially for less frequent anime types.

6.2. Lessons Learned

Throughout the project, several important lessons were discovered:

- Data preprocessing is crucial: Cleaning, encoding, clustering, and discretizing the data significantly improved model performance and interpretability.
- Feature representation matters: Multi-genre encoding and cluster-based feature transformation enabled models to capture complex relationships, demonstrating that thoughtful feature engineering can outweigh model complexity.
- Simple models can be informative: OneR, although basic, helped identify the most predictive attributes and served as a reliable baseline for comparison.
- Class imbalance affects model performance: Without careful handling, minority categories like Medium_Series or Very_Long_Series are poorly predicted, highlighting the importance of class-aware preprocessing and evaluation.
- Evaluation requires multiple metrics: Using accuracy alone can be misleading; incorporating TP rates, MAE, RMSE, and weighted F1 provides a more comprehensive assessment of classifier performance.

6.3. Future Works

Potential directions for further improvement include:

- Class Balancing Techniques: Apply oversampling, undersampling, or SMOTE to address remaining imbalance in underrepresented anime categories.
- Deep Learning Approaches: Explore LSTM or BERT-based models to better capture textual information in anime reviews or synopses.
- Expanded Feature Set: Include additional metadata such as review content, user reputation, release year, and episode runtime to improve predictive accuracy.
- Hyperparameter Tuning and Model Optimization: Fine-tune cluster numbers, bin sizes, and tree pruning parameters for enhanced performance.

- Deployment as a Web Application: Build an interactive platform where users can input anime information and receive instant rating predictions or recommendations based on trained models.

REFERENCES

[1], Anime Recommendation System, Datalira, 2021. Retrieved from : https://www.kaggle.com/code/databeru/anime-recommendation-system/notebook

[2], Anime Recommendations Database, CooperUnion, 2017. Retrieved from: https://www.kaggle.com/datasets/CooperUnion/anime-recommendations-database

[3], Data Mining: Practical Machine Learning Tools and Techniques (4th Ed), Ian H. Witten, Eibe Frank, & Mark A. Hall, 2016. Retrieved from: https://www.cs.waikato.ac.nz/ml/weka/book.html

[4], C4.5: Programs for Machine Learning (J48 Algorithm Source), J. R. Quinlan, 1993. Retrieved from: https://www.sciencedirect.com/book/9781558602380/c4-5-programs-for-machine-learning

[5], Naive Bayes Text Classification, Stanford University, 2008. Retrieved from: https://nlp.stanford.edu/IR-book/html/htmledition/naive-bayes-text-classification-1.html

[6], Some methods for classification and analysis of multivariate observations (K-Means Clustering), J. B. MacQueen, 1967. Retrieved from: https://projecteuclid.org/euclid.bsmsp/1200512992

[7], Understanding TF-IDF: A Simple Introduction, MonkeyLearn, 2020. Retrieved from: https://monkeylearn.com/blog/what-is-tf-idf/

[8], Weka 3: Data Mining Software in Java, University of Waikato, 2024. Retrieved from: https://www.cs.waikato.ac.nz/ml/weka/