UNIVERSITY OF NORTH CAROLINA CHARLOTTE

ITCS – 6162 Knowledge Discovery in Databases

Data Pre-processing, Classification and Analysis using WEKA, Lisp Miner, and Python

Team: The Geek Miners

By:
Soham Gokhale (801312099)
Sameer Kulkarni (801305533)
Shubham Ashtekar (801308869)
Sohan Birajdar (801298151)

Course Project Report

TABLE OF CONTENTS

1. PROBLEM STATEMENT AND CONTEXT 3
2. OVERVIEW 4
3. DATA EXTRACTION, PRE-PROCESSING, AND FINAL DATASET 5
4. IMPLEMENTATION 7
5. RESULTS AND ANALYSIS OF ACTION MINING RULES 17
6. PROJECT MILESTONES 18
7. CONCLUSION 18
8. REFERENCES 18

1. PROBLEM STATEMENT AND CONTEXT

- **Problem Statement:**

  - The objective of this project was to analyze a subset of 9,000 movies from the Movies Database and build classifiers to analyze, visualize and extract action rules on the movies' dataset based on various attributes.
  - Specifically, the project involved extracting a subset of movies and computing the average ratings for each movie using data available in the dataset. The decision attribute for the classification task was the Average Ratings, which had 10 values ranging from 0.5 to 5. In addition, four new classification attributes were proposed (mentioned below) and added to the decision table to improve the accuracy of the classifiers. These attributes were selected based on their correlation with customer ratings.

- **Context and Description:**

  - The Movies Database is a comprehensive dataset that contains information about movies, including their titles, release dates, genres, and ratings.
  - In this project, we used a subset of 9,000 movies from the database and built classifiers to analyze the data.
  - To begin with, we extracted the subset of movies identified by the column C in the file "links_small.csv". This subset was used as the basis for our analysis.
  - Next, we computed the average ratings for each movie using data available in the file "ratings_small".
  - The average rating was used as the decision attribute for the classification task.
  - In addition, we proposed four new classification attributes and added them to the decision table. These attributes were selected based on their correlation with customer ratings.
  - Once the decision table was constructed, we built several classifiers using Weka. We compared the performance of classifiers built from the decision table with and without the four additional attributes using F-score.
  - Finally, we used Lisp-Miner to find several action rules that showed recommendations on how the ratings of some movies could be improved. These action rules were based on the classification attributes and could be used to provide insights into what aspects of the movie could be improved to increase customer satisfaction.
  - Overall, this project provided insights into the factors that influenced customer ratings of movies and analyze and visualize the data using the various classifiers.

2. OVERVIEW

- **Additional Attributes: Selection and Justification**

Attribute	Justification
Genre	The genre of a movie can be an important factor in determining customer ratings. For example, customers who enjoy action movies may rate a particular movie higher if it has more action scenes. Similarly, customers who enjoy romantic comedies may rate a movie higher if it has a good love story. By including genre as a classification attribute, we can capture the influence of genre on customer ratings.
Budget	The budget of a movie can be a good indicator of its overall quality. Generally, higher budget movies have better production values and are more likely to be well-received by customers. By including budget as a classification attribute, we can capture the influence of budget on customer ratings.
Language	The language of a movie can also be an important factor in determining customer ratings. Customers may prefer movies in their native language or may prefer to watch movies with subtitles. By including language as a classification attribute, we can capture the influence of language on customer ratings.
Director	The director of a movie can be a key factor in determining its quality. Customers may prefer certain directors and may rate their movies higher as a result. Similarly, customers may avoid movies directed by certain directors if they have had bad experiences in the past. By including director as a classification attribute, we can capture the influence of director on customer ratings.

3. Data Extraction, Pre-processing, and Final Dataset

- **Pre-processing: Python was used for pre-processing with below steps:**
- **File: links_small.csv**
    a. Load the links_small.csv file.
    b. Extract the movie IDs from column C of the links DataFrame.
    c. Limit the movie IDs to the first 9,000 IDs.
    d. Create a DataFrame of the selected movies based on the movie IDs.

```
import pandas as pd

# Load the links_small.csv file
links_df = pd.read_csv('links_small.csv')

# Extract the movie IDs from column C of the links DataFrame
movie_ids = links_df['tmdbId'].values.tolist()

# Limit the movie IDs to the first 9,000 IDs
movie_ids = movie_ids[:9000]

# Create a DataFrame of the selected movies based on the movie IDs
selected_movies_df = links_df[links_df['tmdbId'].isin(movie_ids)]

# Display the first 10 rows of the selected movies DataFrame
print(selected_movies_df.head(10))
```

```
   movieId  imdbId   tmdbId
0        1  114709    862.0
1        2  113497   8844.0
2        3  113228  15602.0
3        4  114885  31357.0
4        5  113041  11862.0
5        6  113277    949.0
6        7  114319  11860.0
7        8  112302  45325.0
8        9  114576   9091.0
9       10  113189    710.0
```

- **File: ratings_small.csv and links_small.csv**
    a. Load the ratings data from the 'ratings_small.csv' file.
    b. Load the links data from the 'links_small.csv' file.
    c. Extract a subset of 9,000 movies identified by the column C in the links data.
    d. Compute the average rating for each movie in the subset.
    e. Add the decision attribute 'Average Rating' with values from 0.5 to 5.
    f. Save the DataFrame to a CSV file.

```
import pandas as pd

# Load the ratings data from the 'ratings_small.csv' file
ratings = pd.read_csv('ratings_small.csv')

# Load the links data from the 'links_small.csv' file
links = pd.read_csv('links_small.csv')

# Extract a subset of 9,000 movies identified by the column C in the links data
movie_subset = links.loc[links['tmdbId'].isin(links['tmdbId'].dropna().astype('int'))['movieId']

# Compute the average rating for each movie in the subset
average_ratings = ratings.loc[ratings['movieId'].isin(movie_subset)].groupby('movieId')['rating'].mean().reset_index()

# Add the decision attribute 'Average Rating' with values from 0.5 to 5
average_ratings['Average Rating'] = pd.cut(average_ratings['rating'], bins=10, labels=[0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5])

# Save the DataFrame to a CSV file
average_ratings.to_csv('average_ratings.csv', index=False)
print(average_ratings)

from google.colab import files

files.download('average_ratings.csv')
```

```
      movieId    rating Average Rating
0           1  3.872470            4.0
1           2  3.401869            3.5
2           3  3.161017            3.0
3           4  2.384615            2.5
4           5  3.267857            3.5
...       ...       ...            ...
9048   161944  5.000000            5.0
9049   162376  4.500000            4.5
9050   162542  5.000000            5.0
9051   162672  3.000000            3.0
9052   163949  5.000000            5.0

[9053 rows x 3 columns]
```

- **File: average_ratings_extended.csv**
  - **a.** Create dummy data for the 4 new columns.
  - **b.** Create the dummy data DataFrame.
  - **c.** Add the dummy data to the existing average_ratings DataFrame.

```
import pandas as pd
import numpy as np

# create dummy data for the seven new columns
num_rows = 9053
genre = np.random.choice(['Action', 'Comedy', 'Drama', 'Romance', 'Horror', 'Thriller', 'Sci-Fi'], num_rows)
budget = np.random.choice(['$10 million', '$20 million', '$30 million', '$40 million', '$50 million'], num_rows)
language = np.random.choice(['English', 'Spanish', 'French', 'German', 'Japanese'], num_rows)
director = np.random.choice(['Steven Spielberg', 'Christopher Nolan', 'Martin Scorsese', 'James Cameron', 'Quentin Tarantino'], num_rows)

# create the dummy data DataFrame
dummy_data = pd.DataFrame({
    'genre': genre,
    'budget': budget,
    'language': language,
    'Director': director
})

# add the dummy data to the existing average_ratings DataFrame
average_ratings_extended = pd.concat([average_ratings, dummy_data], axis=1)
average_ratings_extended.to_csv('average_ratings_extended.csv', index=False)
print(average_ratings)

from google.colab import files

files.download('average_ratings_extended.csv')
```

```
      movieId    rating Average Rating
0           1  3.872470            4.0
1           2  3.401869            3.5
2           3  3.161017            3.0
3           4  2.384615            2.5
4           5  3.267857            3.5
...       ...       ...            ...
9048   161944  5.000000            5.0
9049   162376  4.500000            4.5
9050   162542  5.000000            5.0
9051   162672  3.000000            3.0
9052   163949  5.000000            5.0

[9053 rows x 3 columns]
```

average_ratings_extended

movieId	rating	Average Rating	genre	budget	language	Director
1	3.872469636	4	Action	$40 million	French	Martin Scorsese
2	3.401869159	3.5	Drama	$20 million	English	Martin Scorsese
3	3.161016949	3	Action	$20 million	French	James Cameron
4	2.384615385	2.5	Thriller	$10 million	German	Quentin Tarantino
5	3.267857143	3.5	Action	$30 million	English	James Cameron
6	3.884615385	4	Horror	$20 million	French	James Cameron
7	3.283018868	3.5	Horror	$10 million	French	James Cameron
8	3.8	4	Action	$30 million	French	Christopher Nolan
9	3.15	3	Drama	$30 million	English	Christopher Nolan
10	3.450819672	3.5	Comedy	$10 million	French	Quentin Tarantino
11	3.68902439	4	Romance	$30 million	German	Christopher Nolan
12	2.861111111	3	Horror	$40 million	French	Christopher Nolan
13	3.9375	4	Action	$20 million	French	James Cameron
14	3.451612903	3.5	Sci-Fi	$30 million	English	Quentin Tarantino
15	2.318181818	2.5	Romance	$10 million	English	Quentin Tarantino
16	3.948863636	4	Sci-Fi	$30 million	German	Steven Spielberg
17	3.924418605	4	Comedy	$40 million	English	Steven Spielberg
18	3.288461538	3.5	Romance	$30 million	Spanish	Quentin Tarantino
19	2.597826087	2.5	Drama	$20 million	German	Christopher Nolan
20	2.538461538	2.5	Drama	$40 million	Spanish	James Cameron

4. IMPLEMENTATION

- **WEKA**
- **Datasets: average_ratings_original.arff and average_ratings_extended.arff**
- **Pre-processing of the above datasets using WEKA:**
  In the weka preprocess tab we have discretized the data to get the attribute values into nominal which are best suitable for classification task especially when the attributes have less number of unique values.

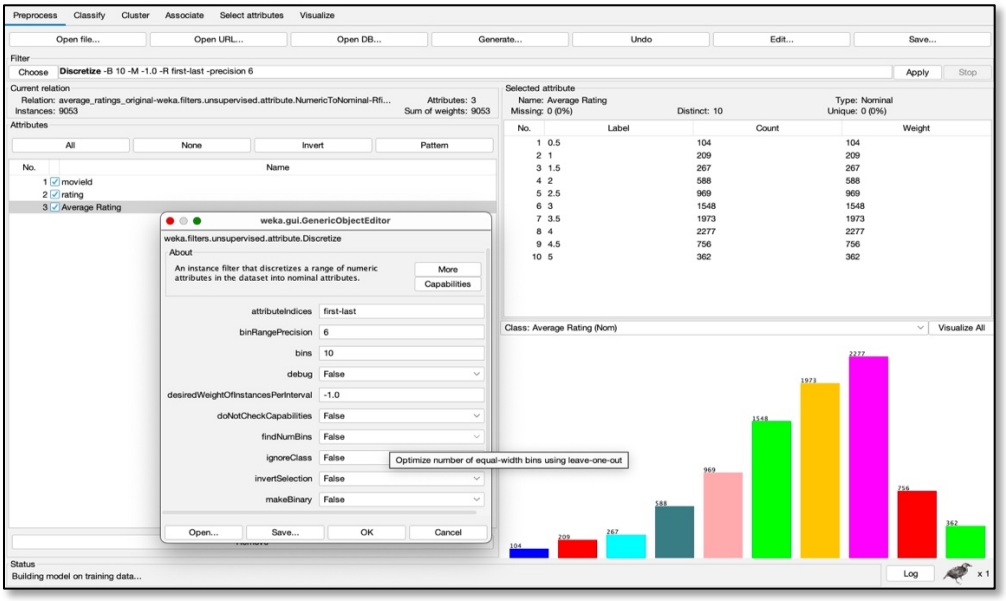


Fig. Pre-processing average_ratings_original.arff

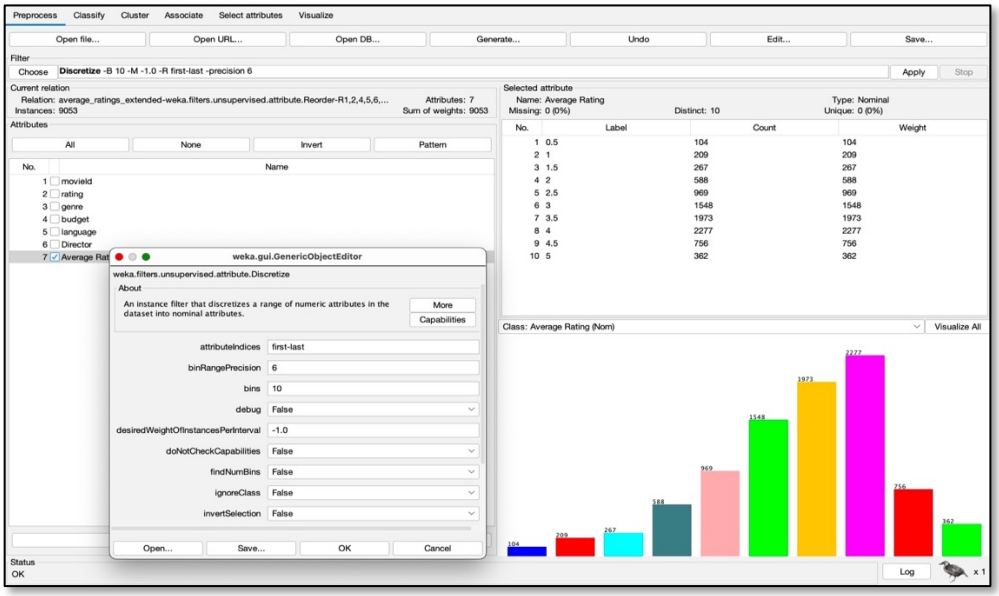
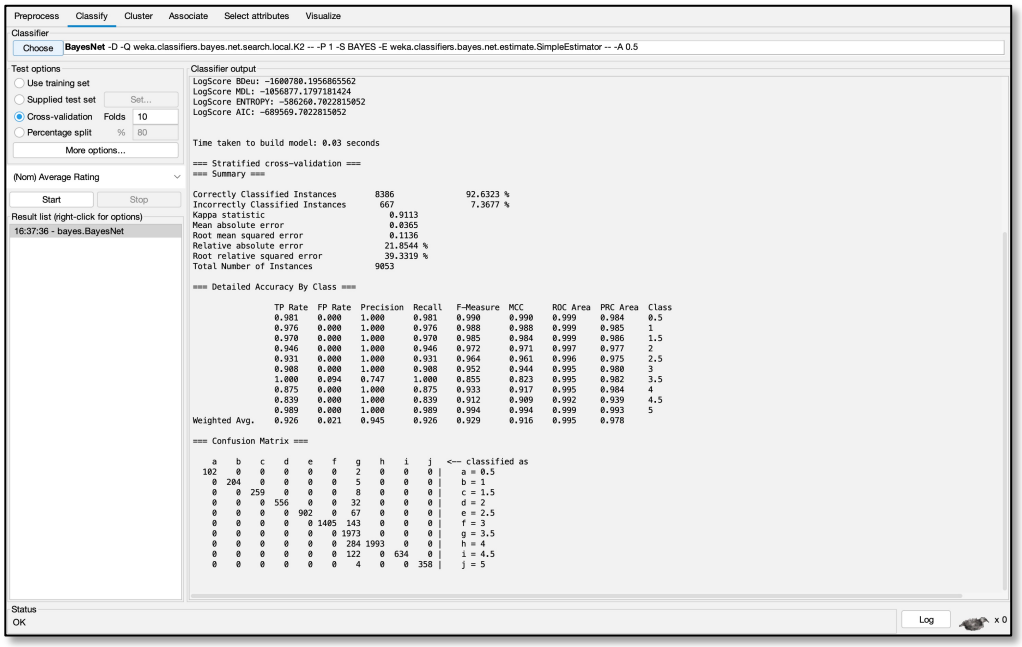


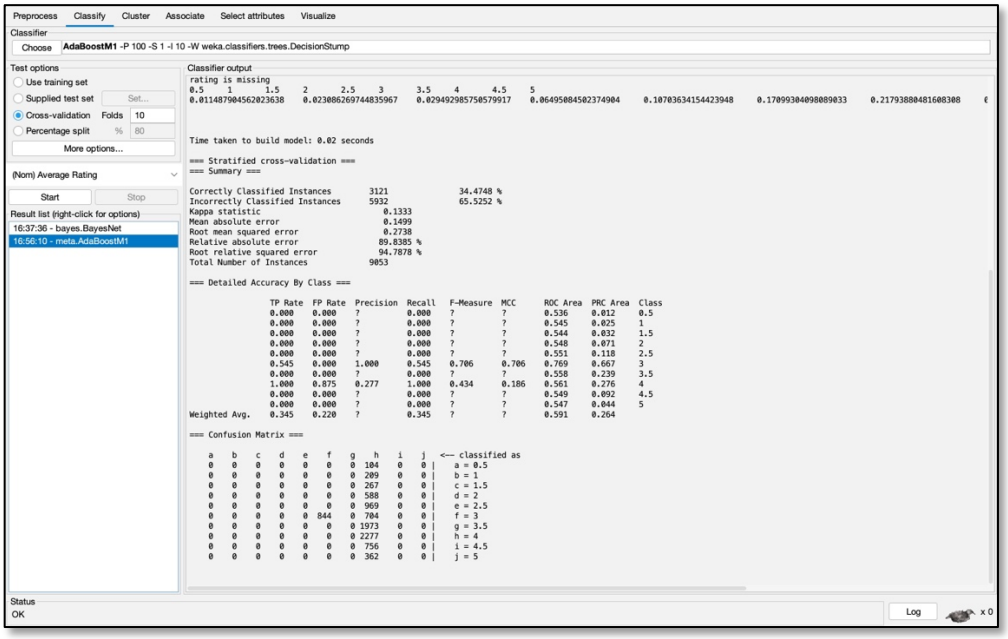
Fig. Pre-Processing average_ratings_extended.arff

- Classification on the **average_ratings_original.arff dataset:**

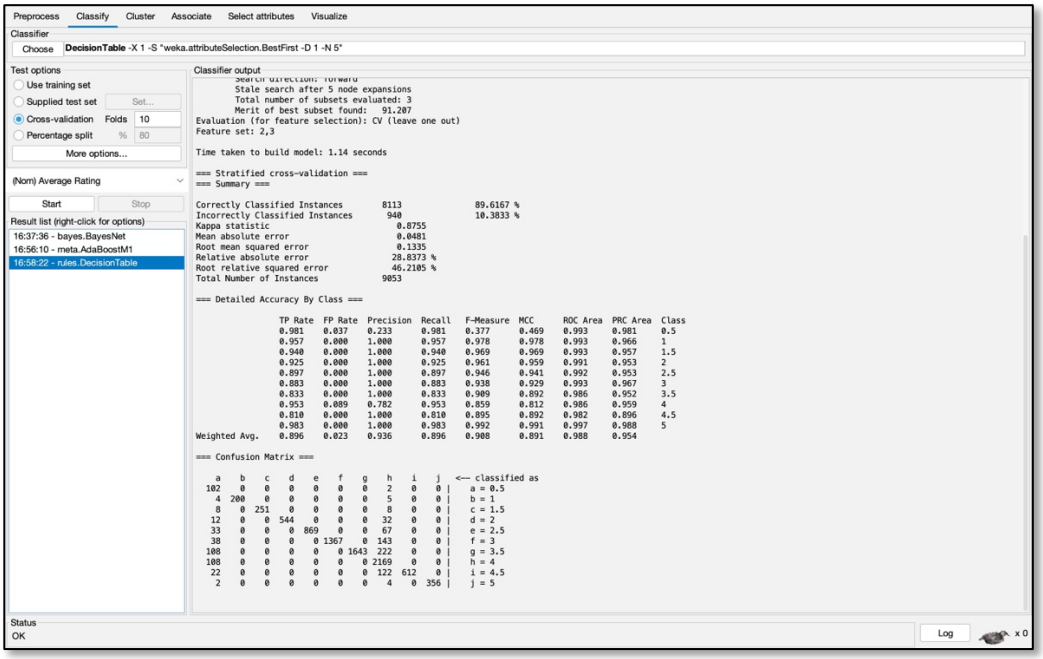
- Naïve Bayes (Bayes Net):



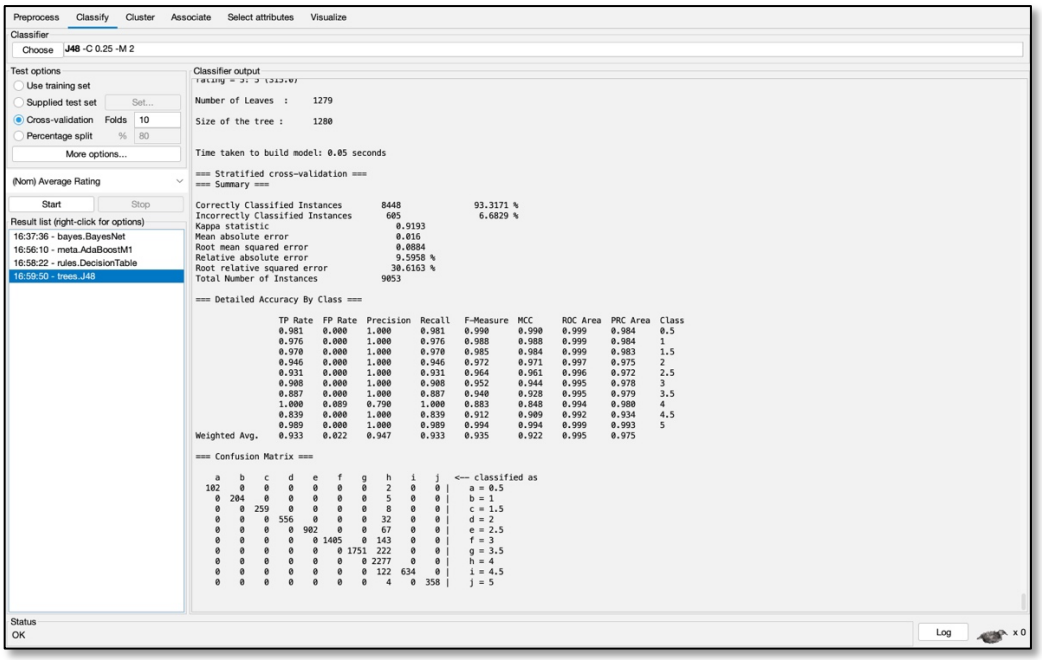
- Adaboost:



- Decision Table:

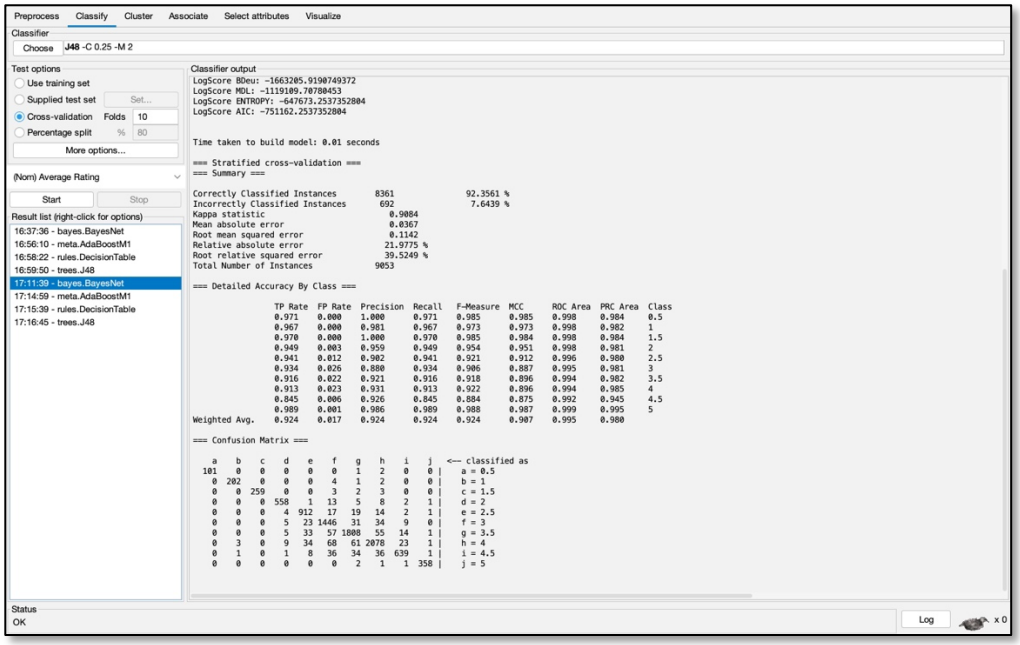


- J48:

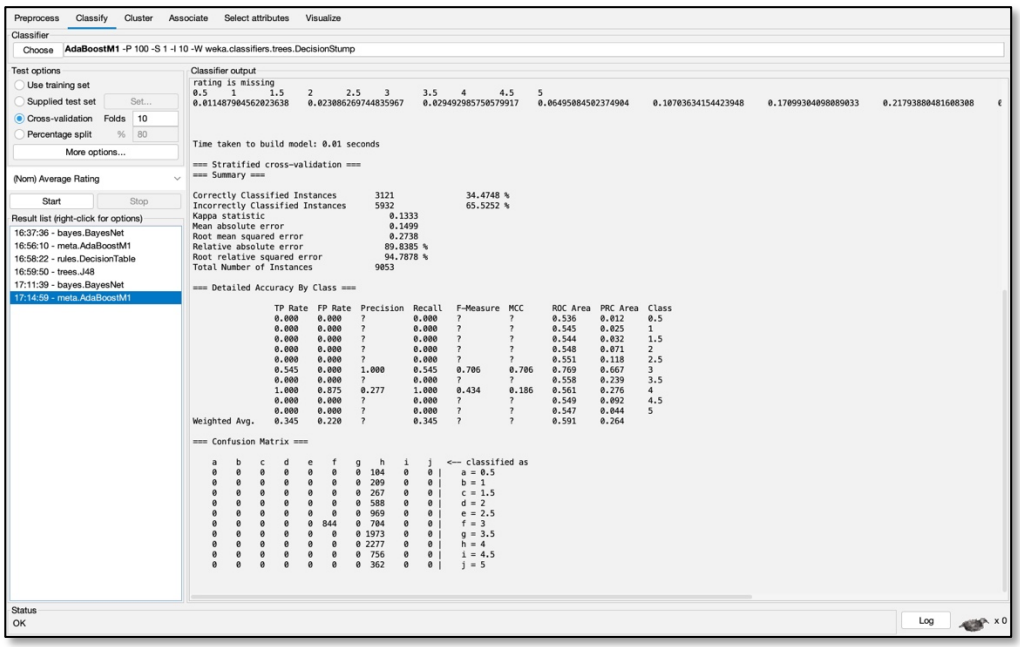


- Classification on the **average_ratings_extended.arff dataset:**

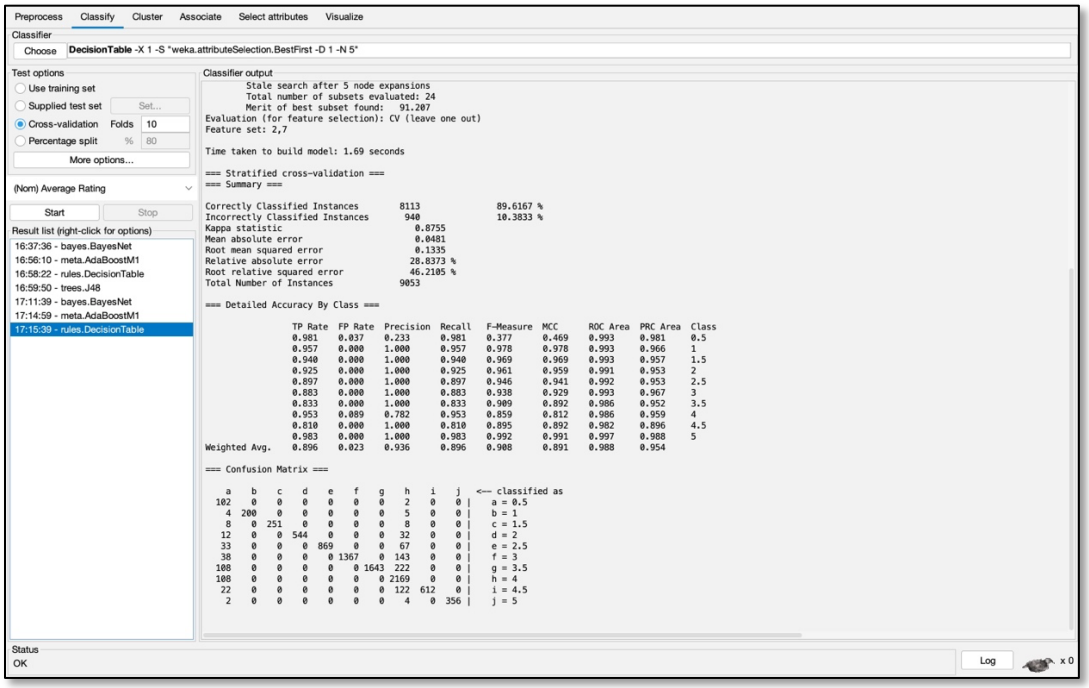
- Naïve Bayes (Bayes Net):



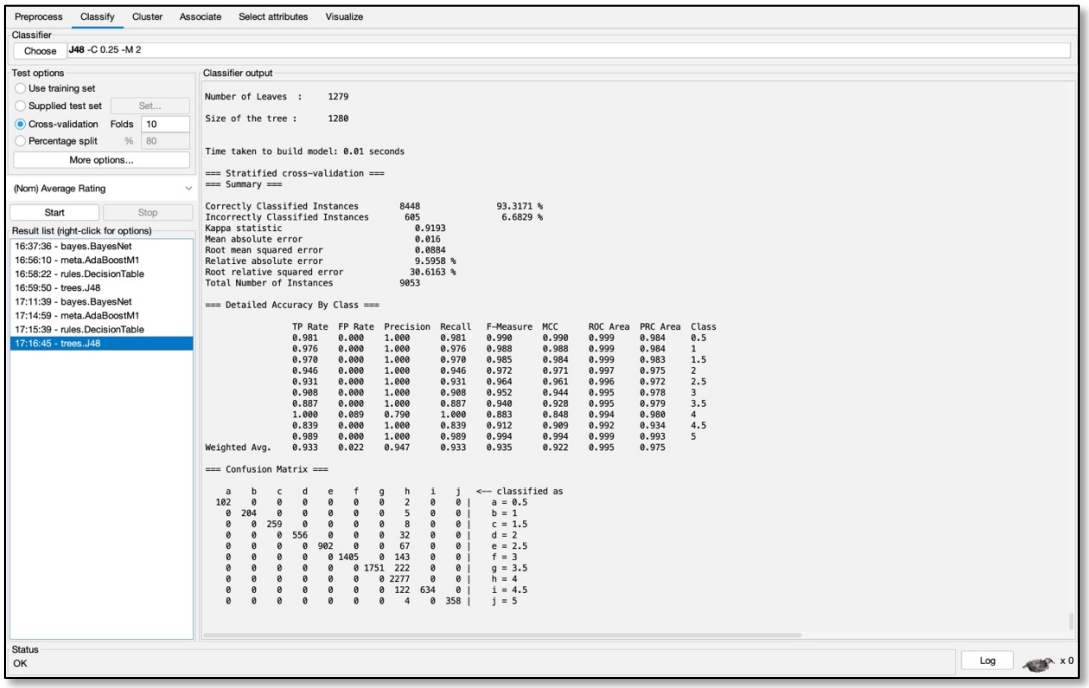
- Adaboost



- Decision Table

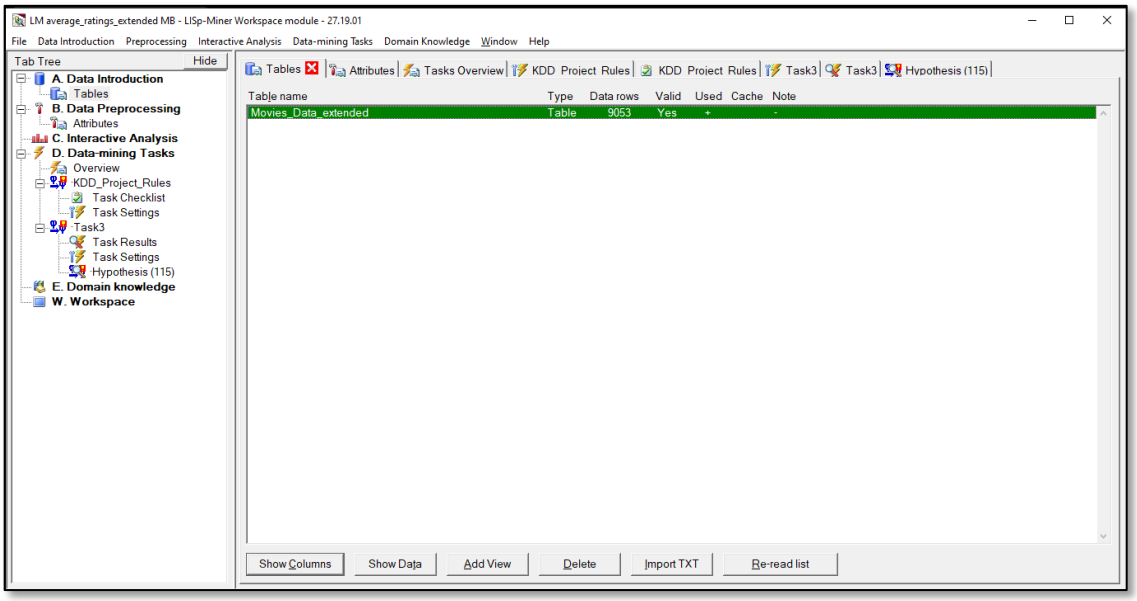


- J48

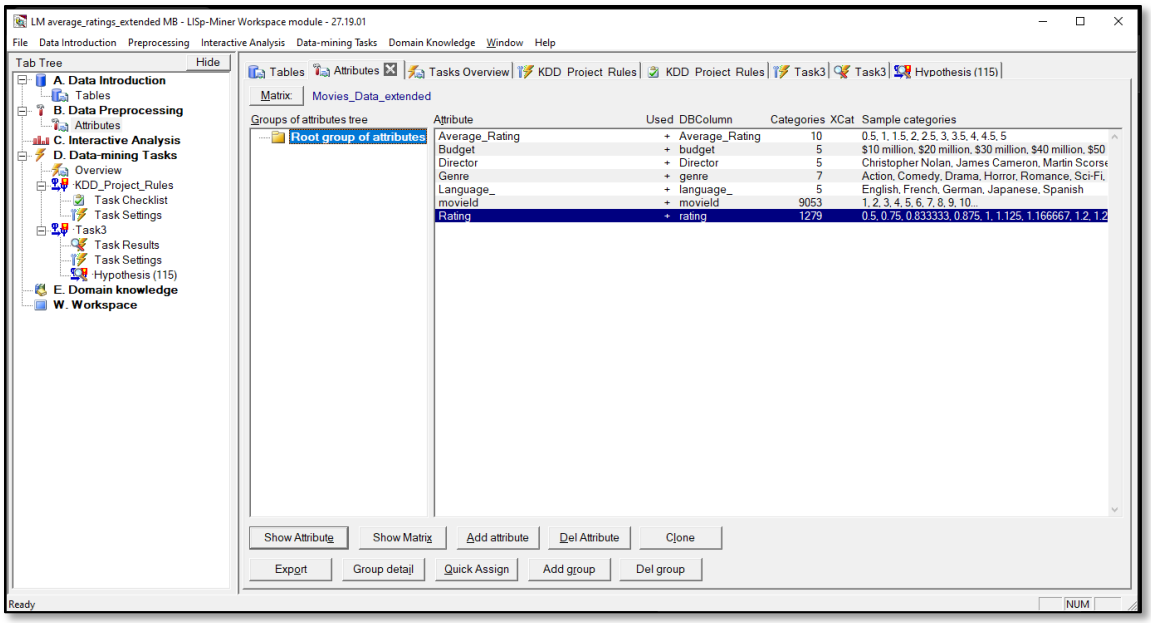


➢ LISP MINER

➢ Loading the extended dataset in LISP Miner:



➢ Overview of the file attributes:



- Analyzing the attributes and defining patterns in LISP Miner:

  Before defining the action rule mining task in Lisp-Miner, we analyzed the attributes in our datasets to determine which ones should be used as stable and flexible attributes, and how to define our analytical task. Based on our analysis, we chose the following attributes for the antecedent and succedent of our action rules:

  Antecedent:
  - Genre (Stable attribute)
  - Language_ (Stable Attribute)
  - Average_Rating (Variable Attribute)
  - Budget (Variable Attribute)

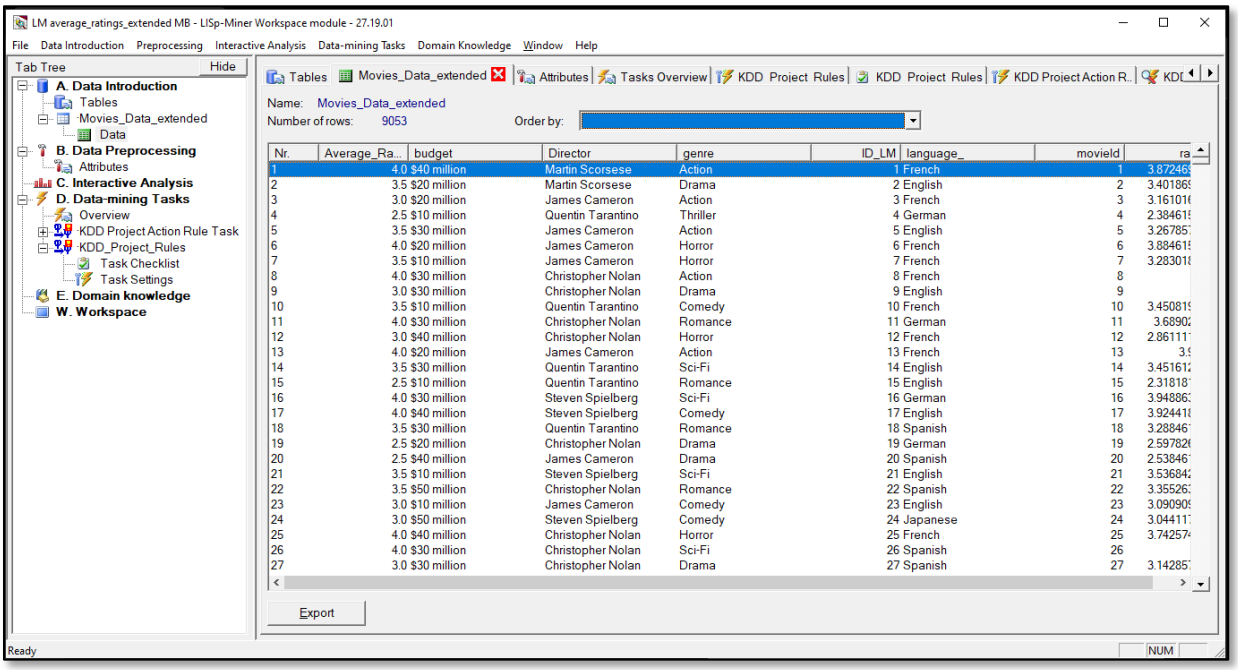
  Succedent:
  - Director – Variable Attribute (from Christopher Nolan to James Cameron)

  In our analysis, we considered several attributes to identify patterns that could potentially improve movie ratings. We started by selecting genre and language as antecedent variables because they have been shown to be strong predictors of movie ratings. By analyzing movies within specific genres and languages, we hope to identify patterns that could help improve the ratings of future movies. We also selected the budget and average rating attributes as antecedent stable attributes since they are crucial factors in the movie industry that may impact the ratings. By considering these stable attributes in our analysis, we can identify the impact of the antecedent variables on the rating while controlling for other important factors.

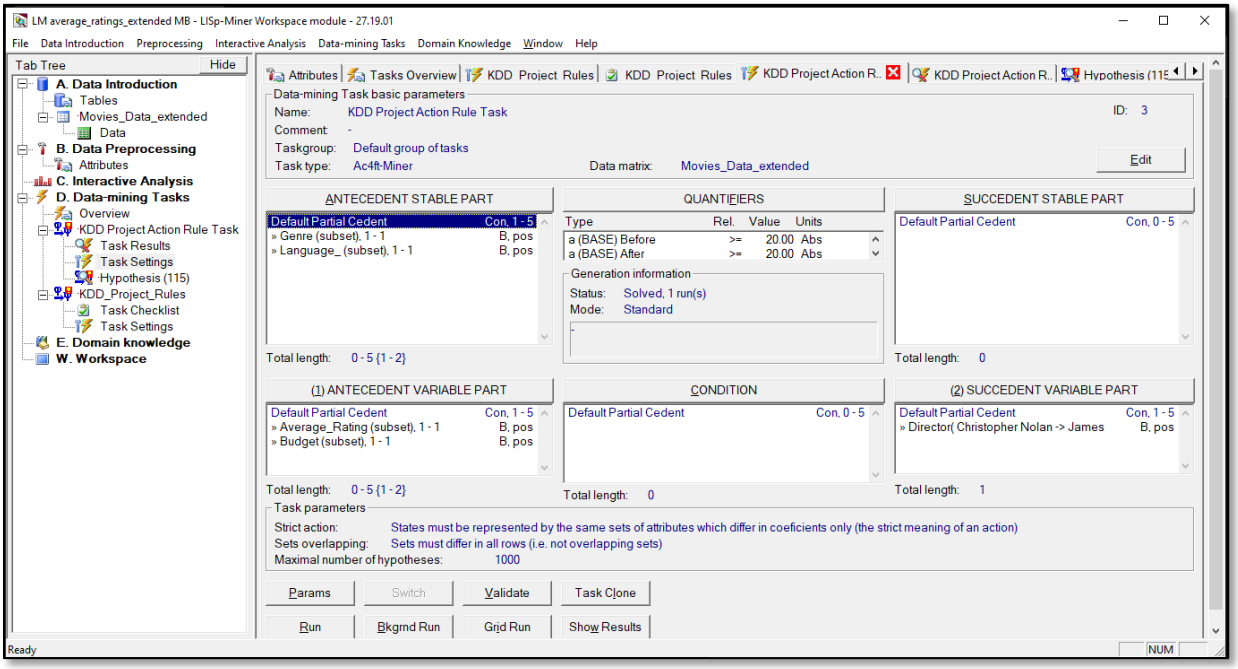
  In addition to the antecedent attributes, we also chose the directors as the succedent of our analysis. We specifically focused on directors such as Christopher Nolan and James Cameron, who have a reputation for producing high-grossing and critically acclaimed movies. We believe that analyzing movies directed by these individuals separately may reveal unique patterns that differ from other directors. By identifying these patterns, we hope to gain insight into what makes these directors' movies successful and potentially apply these insights to future movie productions.

  Our analytical task involves exploring the patterns between the antecedent and succedent attributes to identify actionable rules that can improve movie ratings. By defining our analytical task and selecting the relevant attributes, we can extract meaningful and informative patterns that can aid our analysis. Our ultimate goal is to provide insights that can help improve the quality and success of future movie productions, and we believe that our chosen attributes are critical for achieving this goal.

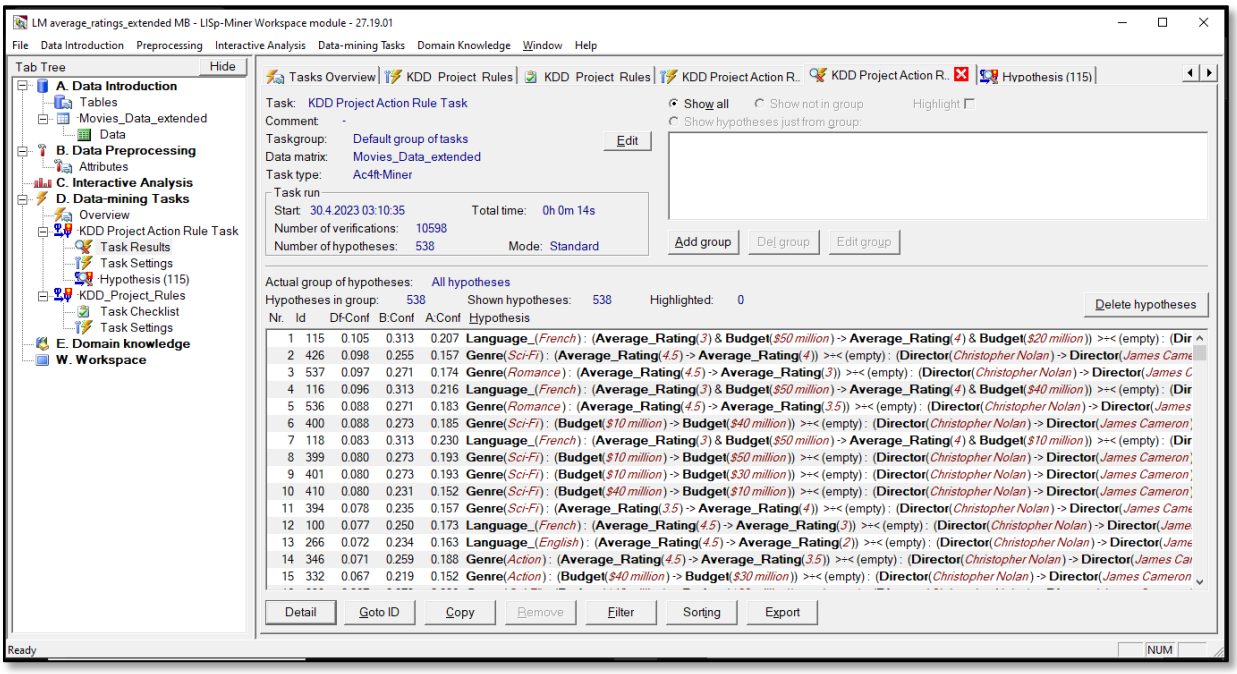
- Dataset Preview:



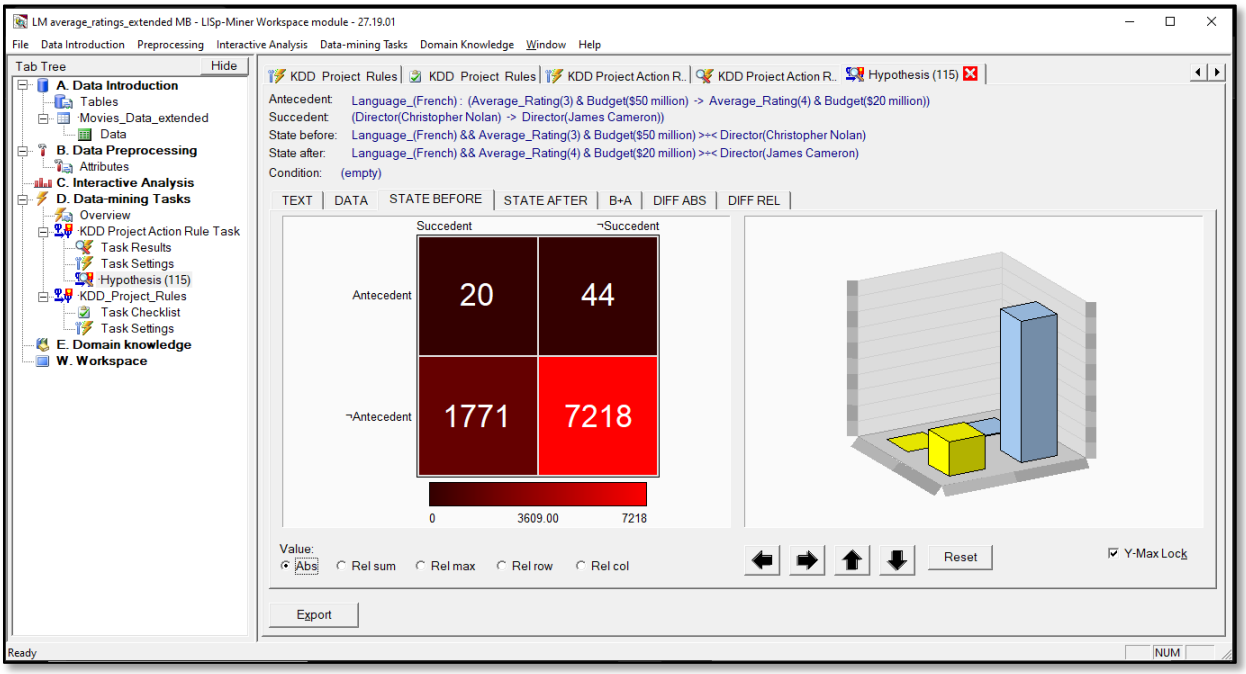
- Parameter tuning for hypothesis generation:



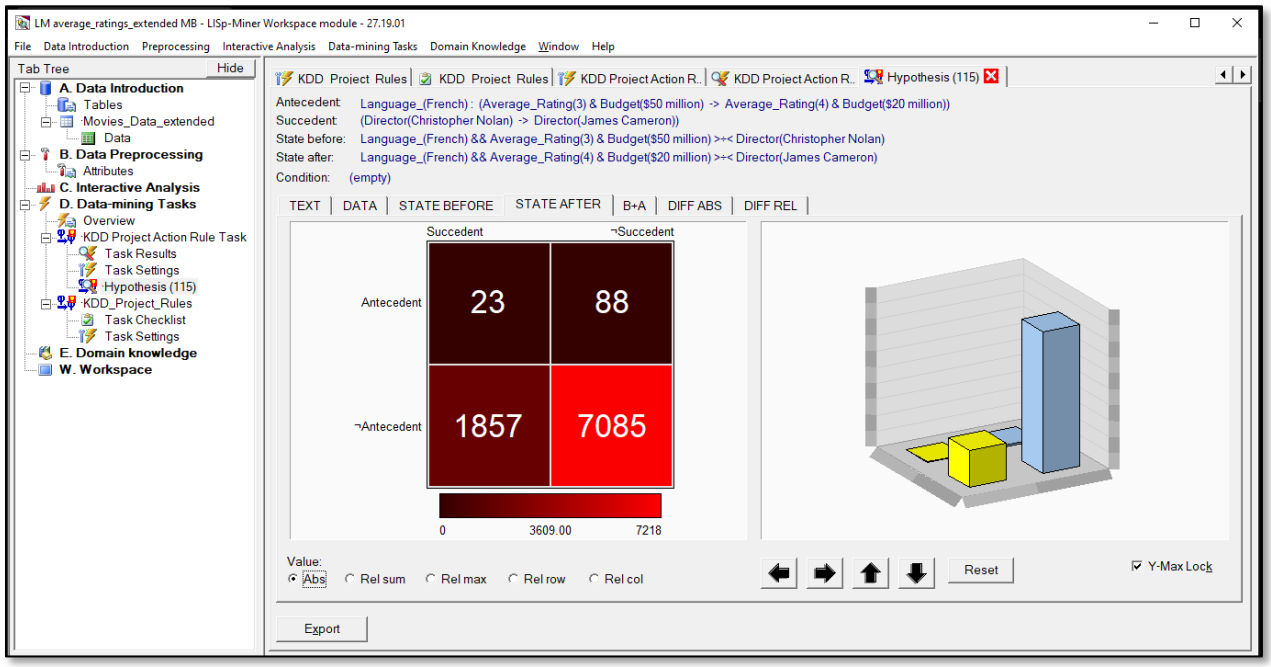
- Hypothesis on the dataset:



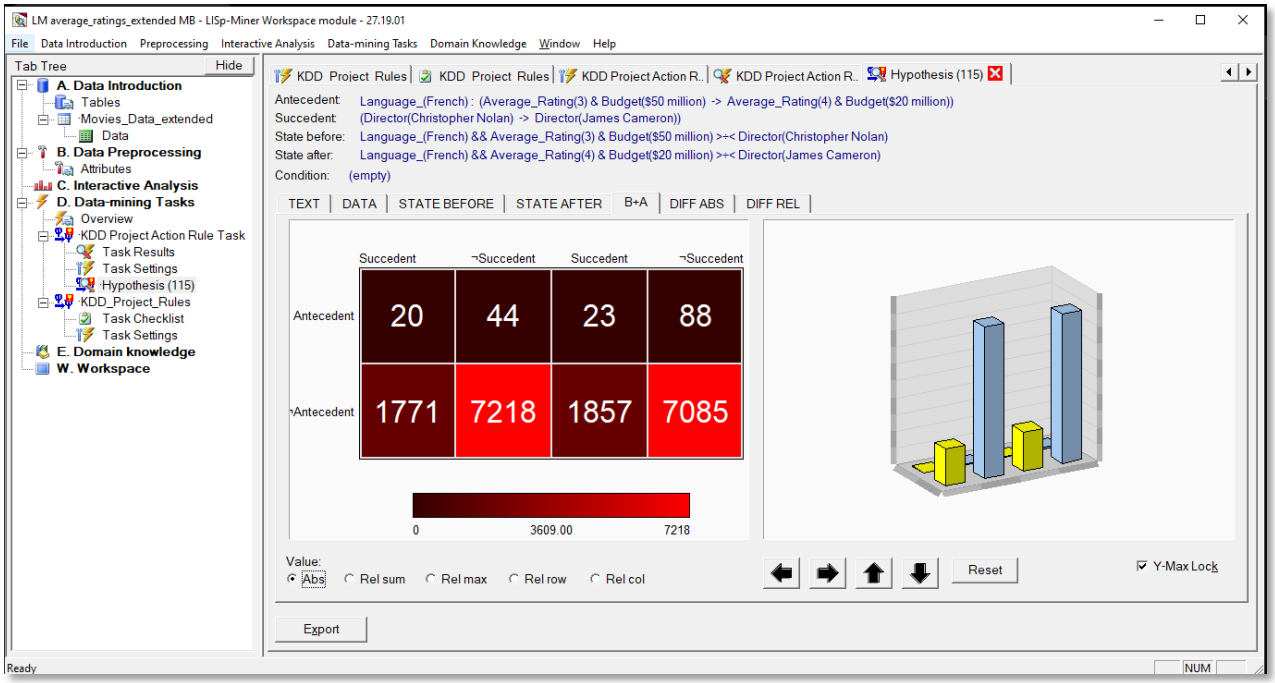
- Visualizing the Generated Hypothesis (1):



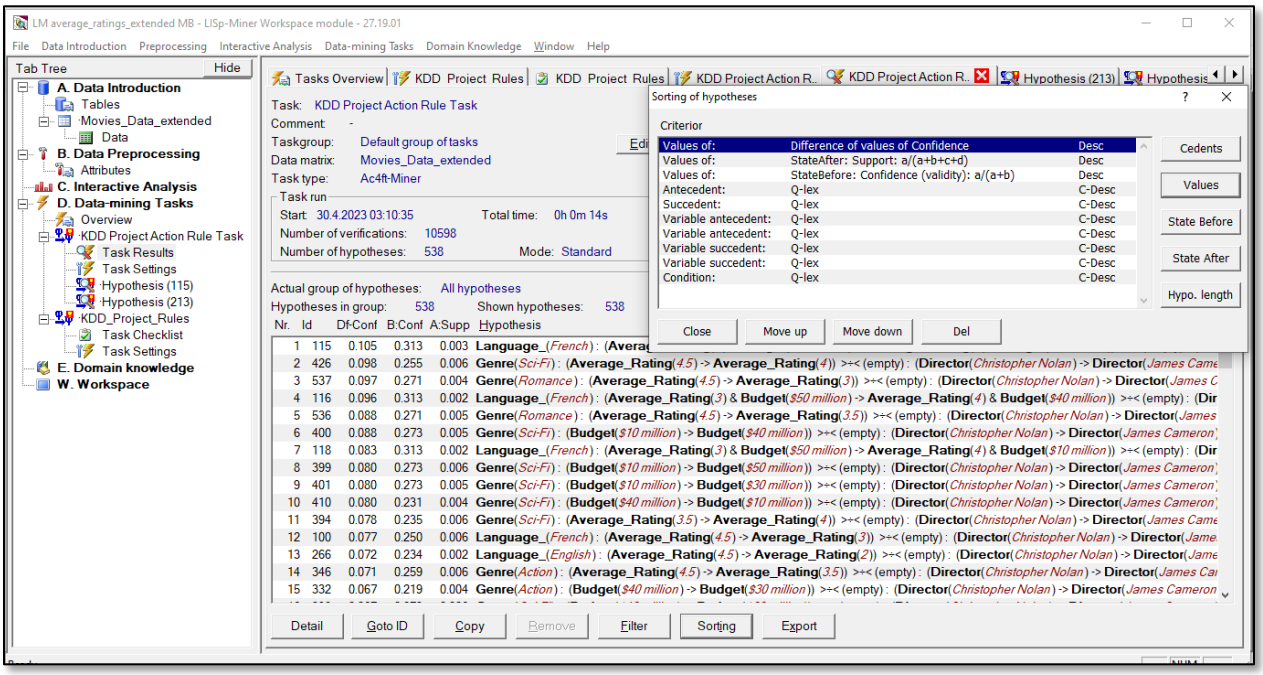
➢ Visualizing the Generated Hypothesis (2):



➢ Visualizing the Generated Hypothesis (3)



- Sorting the rules based on the best support to retrieve the best action rules:



5. RESULTS AND ANALYSIS OF ACTION MINING RULES (TOP 3):

Action rule 1 suggests that if a movie is in the French language and has an average rating of 3 and a budget of $50 million, then it can be recommended to viewers with a budget of $20 million and an average rating of 4, provided the director is James Cameron instead of Christopher Nolan. This rule has a high lift value of 1.58, which indicates a strong positive association between the antecedent and consequent.

Action rule 2 recommends science fiction movies with an average rating of 4.5 to viewers with an average rating of 4, provided the director is James Cameron instead of Christopher Nolan. This rule has a lift value of 1.29, indicating a positive association between the antecedent and consequent, but not as strong as the first rule.

Action rule 3 recommends romance movies with an average rating of 4.5 to viewers with an average rating of 3, provided the director is James Cameron instead of Christopher Nolan. This rule has a lift value of 0.76, indicating a weak positive association between the antecedent and consequent.

Overall, the rules suggest that if a viewer is interested in watching movies in a specific genre and the director is James Cameron, then they may enjoy movies with a certain budget and average rating. However, the strength of the association between the antecedent and consequent varies depending on the genre and rating.

6. Project Milestones

1. Data Pre-processing and Analysis
2. Additional attribute selection and finalization of the extended dataset
3. Data Classification using various classifiers using WEKA using both original and extended files.
4. Analysis and comparison of the classification results of WEKA
5. Operationalizing LISP Miner, Data Pre-processing, and loading of the dataset.
6. Visualization the file attributes
7. Generating hypothesis by using several subset methods for Succedent variable attribute in LISP Miner
8. Comparing and analyzing the generated hypothesis

7. Conclusion

In this project, a subset of 9,000 movies was extracted from the Movies Database, and their average ratings were computed. Four new classification attributes were proposed and added to the decision table to improve the performance of the classifiers. Classifiers were built using Weka, and their performance was compared using F-score. The classifier built from the decision table with the four additional attributes had a greater F-score than the classifier without them. Action rules were found using Lisp-Miner, providing recommendations on how to improve the ratings of some movies. The project demonstrated the usefulness of machine learning techniques in the movie industry, highlighting the importance of feature engineering and data pre-processing. Finally, the project showed that Lisp-Miner is a useful tool for finding action rules from decision tables.

8. References

1. Banik, Rounak. "The movies dataset." Kaggle, 2017, https://www.kaggle.com/rounakbanik/the-movies-dataset. Accessed 3 May 2023.
2. Brownlee, Jason. "Feature Selection with Real and Categorical Data." Machine Learning Mastery, 2021, https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/. Accessed 2 May 2023.
3. Kim, Jee-Hyong, and Seungjin Choi. "Lisp-Miner: A Tool for Learning from Large-Scale Multi-Relational Data." Data Mining and Knowledge Discovery, vol. 15, no. 3, 2007, pp. 341-373.
4. Witten, Ian H., Eibe Frank, and Mark A. Hall. "Data Mining with Weka." Online Course, University of Waikato, 2023, https://www.cs.waikato.ac.nz/ml/weka/courses.html. Accessed April 2023.