

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	<p>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.</p>
Budget	<p>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.</p>
Language	<p>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.</p>
Director	<p>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.</p>

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    imbdId    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	3.045515	3.5
4	5	3.267857	3.5
..	
948	161944	5.000000	5.0
9049	162376	4.500000	4.5
9050	162342	5.000000	5.0
9051	162472	3.000000	3.0
9052	163949	5.000000	5.0

(9053 rows x 3 columns)

- File: **average_ratings_extended.csv**

- Create dummy data for the 4 new columns.
- Create the dummy data DataFrame.
- 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  3.872469636           4.0
1        2  3.401869159           3.5
2        3  3.161016949           3.0
3        4  2.384615385           2.5
4        5  3.267857143           3.5
...       ...
9048  161944  5.0000000           5.0
9049  162376  4.5000000           4.5
9050  162542  5.0000000           5.0
9051  162672  3.0000000           3.0
9052  163949  5.0000000           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.

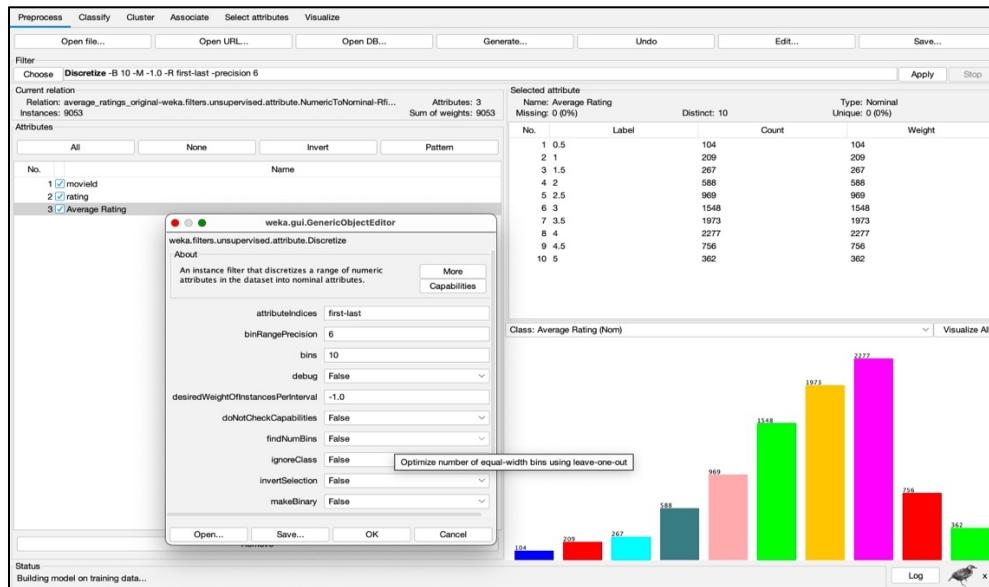


Fig. Pre-processing average ratings original.arff

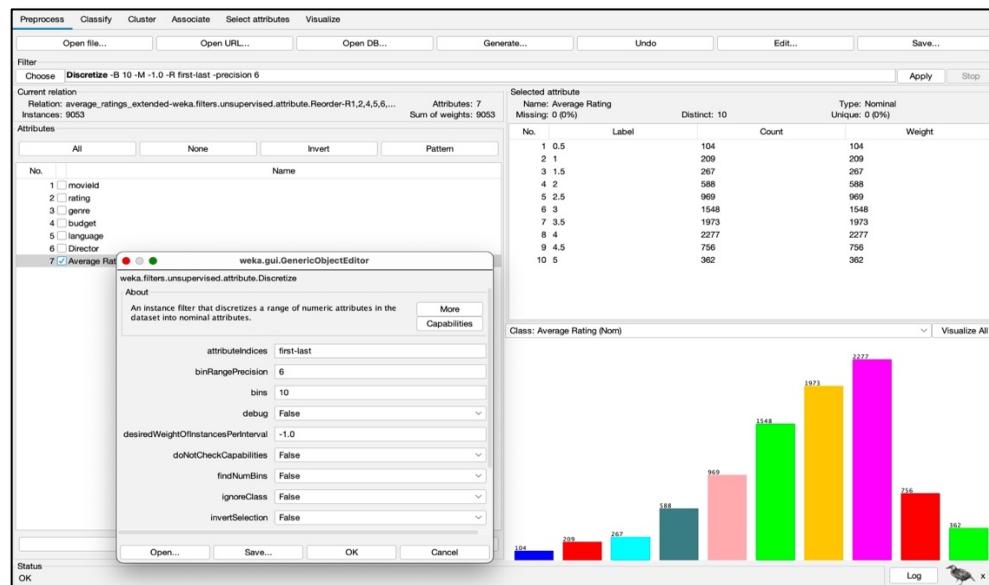
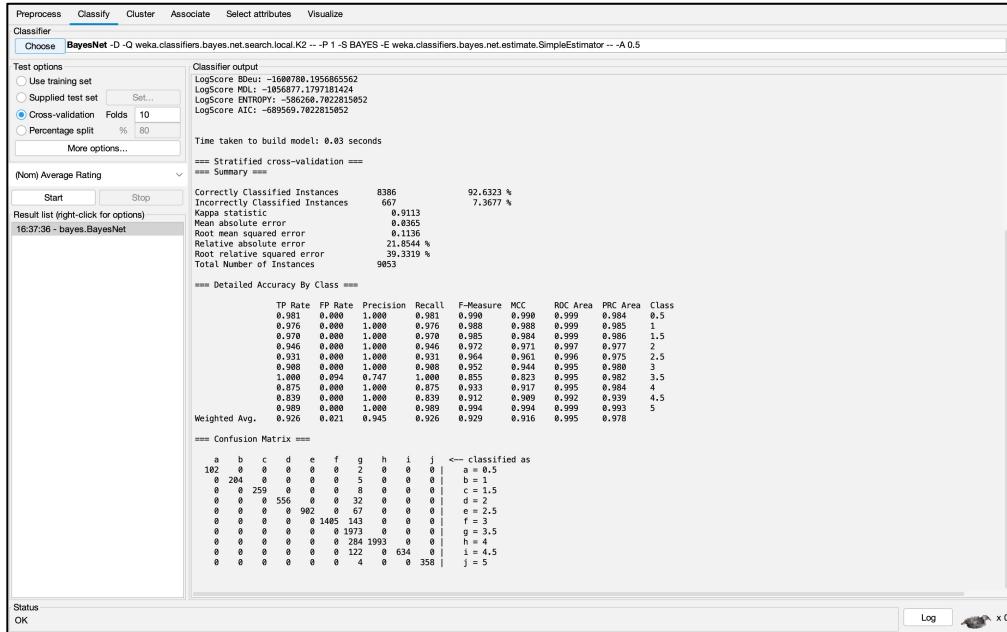


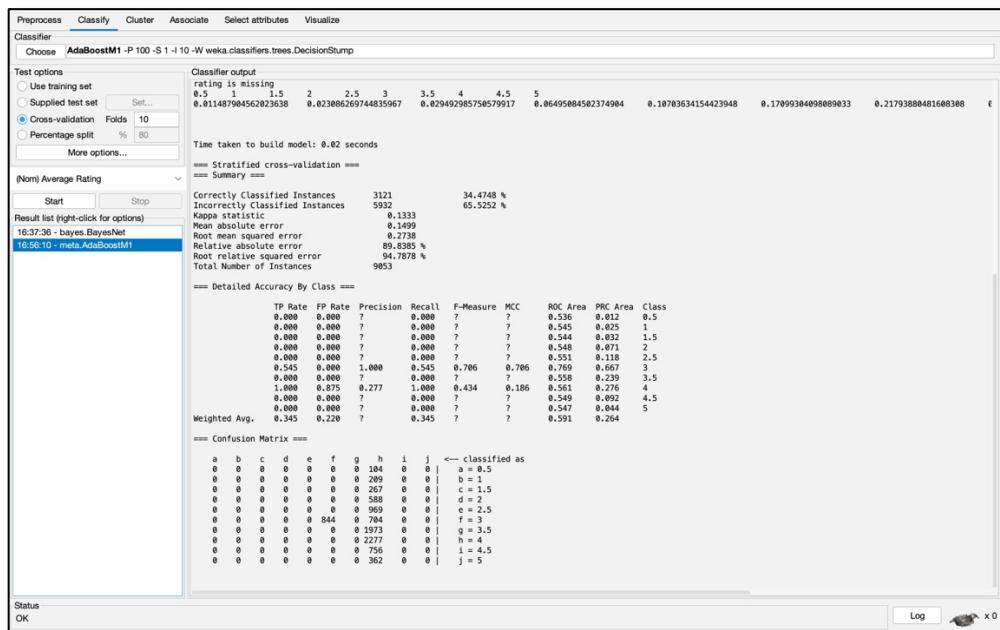
Fig. Pre-Processing average ratings extended.arff

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

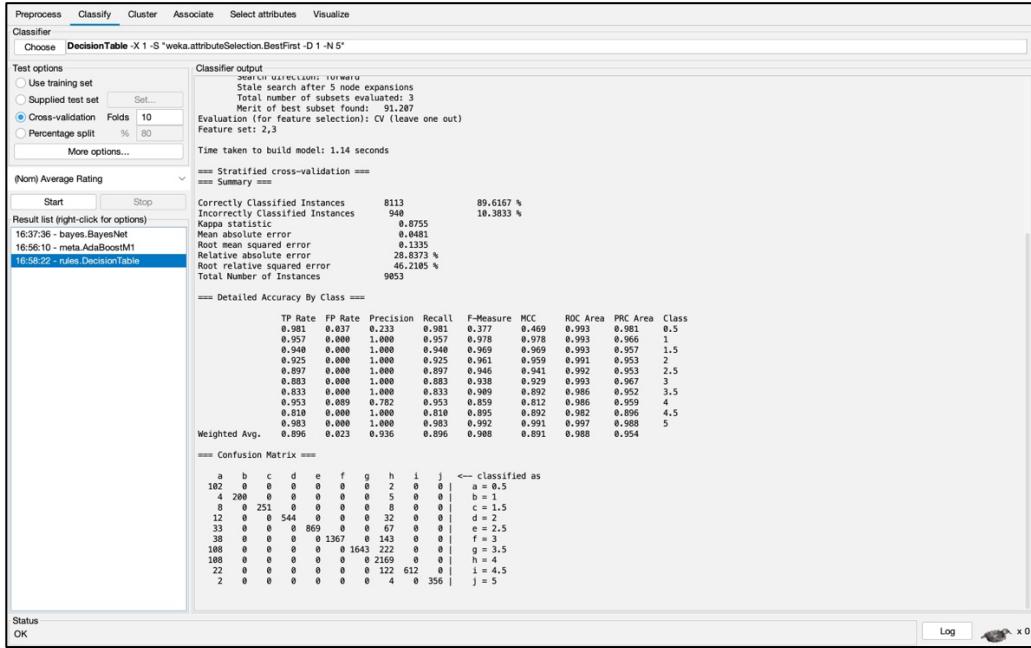
➤ Naïve Bayes (Bayes Net):



➤ Adaboost:



➤ Decision Table:



Classifier output:

```

TAKING = J1-J2-J3-J4-J5
State search after 5 node expansions
Total number of subsets evaluated: 3
Maximal test subset found: 91,207
Evaluation (for feature selection): CV (leave one out)
Feature set: 2, 3, 4, 5
Time taken to build model: 1.14 seconds

```

Summary

	Correctly Classified Instances	8113	89.6167 %
Incorrectly Classified Instances	940	10.3833 %	
Kappa statistic	0.975		
Mean absolute error	0.0481		
Root mean squared error	0.1335		
Relative absolute error	28.8373 %		
Root relative squared error	46.2105 %		
Total Number of Instances	9553		

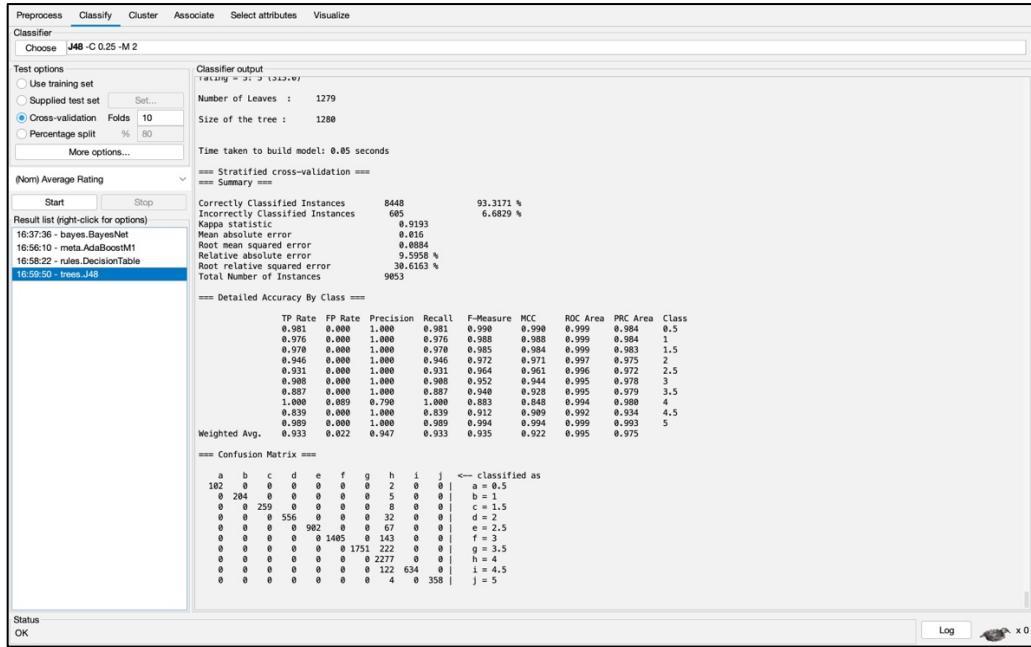
Detailed Accuracy By Class

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.981	0.037	0.233	0.981	0.377	0.469	0.993	0.981	0.5	
0.957	0.000	1.000	0.957	0.978	0.978	0.993	0.966	1	
0.948	0.000	1.000	0.948	0.949	0.969	0.969	0.993	1.5	
0.935	0.000	1.000	0.935	0.935	0.935	0.935	0.932	2	
0.897	0.000	1.000	0.897	0.946	0.941	0.992	0.953	2.5	
0.883	0.000	1.000	0.883	0.938	0.929	0.993	0.967	3	
0.833	0.000	1.000	0.833	0.969	0.899	0.986	0.952	3.5	
0.833	0.000	1.000	0.833	0.969	0.899	0.986	0.952	4	
0.810	0.000	1.000	0.810	0.895	0.892	0.982	0.896	4.5	
0.983	0.000	1.000	0.983	0.992	0.991	0.997	0.988	5	
Weighted Avg.	0.896	0.023	0.936	0.896	0.988	0.981	0.988	0.954	

Confusion Matrix

	a	b	c	d	e	f	g	h	i	j	<-- classified as
182	0	0	0	0	0	0	2	0	0	0	a = 0.5
4	280	0	0	0	0	0	5	0	0	0	b = 1
8	0	251	0	0	0	0	8	0	0	0	c = 1.5
12	0	0	546	0	0	0	2	0	0	0	d = 2
33	0	0	0	869	0	0	67	0	0	0	e = 2.5
38	0	0	0	0	1367	0	143	0	0	0	f = 3
188	0	0	0	0	0	1643	222	0	0	0	g = 3.5
186	0	0	0	0	0	0	2169	0	0	0	h = 4
22	0	0	0	0	0	0	122	612	0	0	i = 4.5
2	0	0	0	0	0	0	4	0	358	0	j = 5

➤ J48:



Classifier output:

```

TAKING = J1-J2-J3-J4-J5
Number of Leaves : 1279
Size of the tree : 1280

```

Summary

	Correctly Classified Instances	8448	93.3171 %
Incorrectly Classified Instances	665	6.6829 %	
Kappa statistic	0.9193		
Mean absolute error	0.016		
Root mean squared error	0.0884		
Relative absolute error	5.958 %		
Root relative squared error	38.6163 %		
Total Number of Instances	9553		

Detailed Accuracy By Class

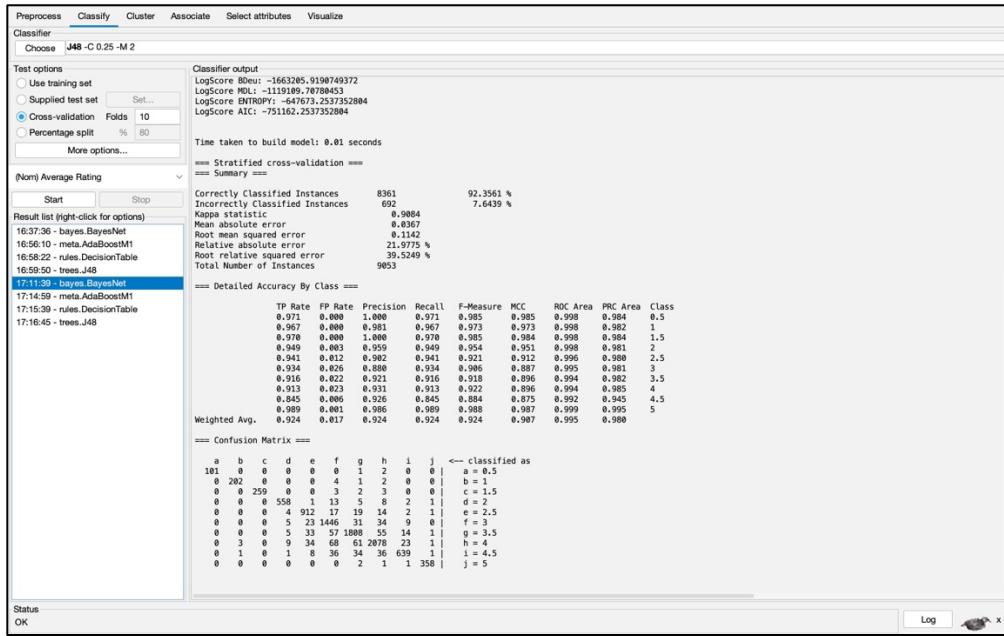
	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.981	0.000	1.000	0.981	0.298	0.998	0.999	0.984	0.984	0.5
0.976	0.000	1.000	0.976	0.988	0.988	0.999	0.999	0.999	1
0.970	0.000	1.000	0.970	0.985	0.984	0.999	0.983	1.5	
0.946	0.000	1.000	0.946	0.972	0.971	0.997	0.975	2	
0.933	0.000	1.000	0.933	0.973	0.944	0.996	0.972	2.5	
0.889	0.000	1.000	0.889	0.988	0.952	0.944	0.995	0.978	3
0.887	0.000	1.000	0.887	0.948	0.928	0.995	0.979	3.5	
1.000	0.089	0.790	1.000	0.883	0.848	0.994	0.988	4	
0.989	0.000	1.000	0.989	0.994	0.994	0.999	0.993	4.5	
Weighted Avg.	0.933	0.022	0.947	0.933	0.935	0.922	0.995	0.975	

Confusion Matrix

	a	b	c	d	e	f	g	h	i	j	<-- classified as
182	0	0	0	0	0	0	2	0	0	0	a = 0.5
4	284	0	0	0	0	0	5	0	0	0	b = 1
0	0	259	0	0	0	0	8	0	0	0	c = 1.5
0	0	0	556	0	0	0	32	0	0	0	d = 2
0	0	0	0	968	0	0	67	0	0	0	e = 2.5
0	0	0	0	0	1485	0	143	0	0	0	f = 3
0	0	0	0	0	0	1751	222	0	0	0	g = 3.5
0	0	0	0	0	0	0	2277	0	0	0	h = 4
0	0	0	0	0	0	0	122	634	0	0	i = 4.5
0	0	0	0	0	0	0	4	0	358	0	j = 5

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

➤ Naïve Bayes (Bayes Net):



Preprocess Classify Cluster Associate Select attributes Visualize

Classifier Choose J48 -C 0.25 -M 2

Test options

- Use training set
- Supplied test set Set...
- Cross-validation Folds 10
- Percentage split % 80

More options...

Time taken to build model: 0.01 seconds

LogScore B000: -1663285.9198749372
 LogScore MDL: -119189.70788453
 LogScore ENTROPY: -647673.2537352804
 LogScore AIC: -751162.2537352804

Result list (right-click for options)

- 16:37:36 - bayes.BayesNet
- 16:56:10 - meta AdaBoostM1
- 16:58:22 - rules.DecisionTable
- 16:59:50 - trees.J48
- 17:11:39 - bayes.BayesNet
- 17:14:59 - meta.AdaBoostM1
- 17:15:39 - rules.DecisionTable
- 17:16:45 - trees.J48

(Nom) Average Rating

Start Stop

Correctly Classified Instances 8361 92.3561 %
 Incorrectly Classified Instances 692 7.6439 %
 Kappa statistic 0.9084
 Mean absolute error 0.0367
 Root mean squared error 0.1142
 Relative absolute error 21.9775 %
 Root relative squared error 39.5249 %
 Total Number of Instances 9853

== Stratified cross-validation ==
 == Summary ==

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.971	0.000	1.000	0.971	0.985	0.985	0.984	0.984	0.984	0.5
0.967	0.000	0.981	0.967	0.973	0.973	0.998	0.982	1	
0.970	0.000	1.000	0.970	0.985	0.984	0.998	0.984	0.984	1.5
0.969	0.000	0.981	0.969	0.973	0.973	0.998	0.982	0.982	2
0.941	0.012	0.982	0.941	0.921	0.912	0.996	0.988	0.988	2.5
0.934	0.026	0.988	0.934	0.906	0.887	0.995	0.981	0.981	3
0.936	0.002	0.921	0.936	0.918	0.896	0.994	0.982	0.982	3.5
0.933	0.027	0.933	0.933	0.913	0.894	0.994	0.980	0.980	4
0.845	0.005	0.926	0.845	0.884	0.875	0.992	0.945	0.945	4.5
0.899	0.001	0.986	0.899	0.988	0.987	0.999	0.995	0.995	5
Weighted Avg.	0.924	0.017	0.924	0.924	0.924	0.998	0.987	0.987	0.988

== Detailed Accuracy By Class ==

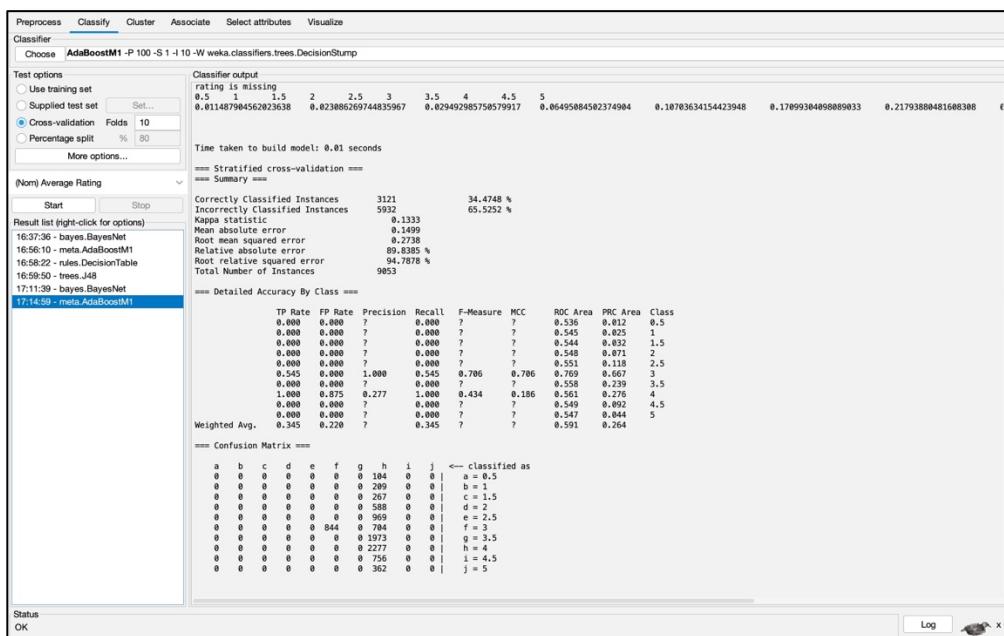
	a	b	c	d	e	f	g	h	i	j	<-- Classified as
101	0	0	0	0	0	1	2	0	0	1	a = 0.5
0	202	0	0	0	4	1	2	0	0	1	b = 1
0	0	250	0	0	3	2	3	0	0	1	c = 1.5
0	0	0	558	0	18	5	0	0	1	2	d = 2
0	0	0	4	912	17	19	14	2	1	1	e = 2.5
0	0	0	5	23	1446	31	34	9	0	1	f = 3
0	0	0	5	53	57	14	24	1	1	1	g = 3.5
0	3	0	0	0	61	2078	23	1	1	1	h = 4
0	1	0	1	8	36	34	36	639	1	1	i = 4.5
0	0	0	0	0	2	1	1	358	1	1	j = 5

== Confusion Matrix ==

	a	b	c	d	e	f	g	h	i	j	<-- Classified as
a	3121	0	0	0	0	0	0	0	0	0	a = 0.5
b	0	5922	0	0	0	0	0	0	0	0	b = 1
c	0	0	5922	0	0	0	0	0	0	0	c = 1.5
d	0	0	0	5922	0	0	0	0	0	0	d = 2
e	0	0	0	0	5922	0	0	0	0	0	e = 2.5
f	0	0	0	0	0	5922	0	0	0	0	f = 3
g	0	0	0	0	0	0	5922	0	0	0	g = 3.5
h	0	0	0	0	0	0	0	5922	0	0	h = 4
i	0	0	0	0	0	0	0	0	5922	0	i = 4.5
j	0	0	0	0	0	0	0	0	0	5922	j = 5

Status OK

➤ Adaboost



Preprocess Classify Cluster Associate Select attributes Visualize

Classifier Choose AdaBoostM1 -P 100 -S 1 -I 10 -W weka.classifiers.trees.DecisionStump

Test options

- Use training set
- Supplied test set Set...
- Cross-validation Folds 10
- Percentage split % 80

More options...

Time taken to build model: 0.01 seconds

rating is missing 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5 0.01148794562693638 0.02388626744835967 0.029492985758579917 0.06495845802374984 0.187083634154423948 0.18709384048889033 0.2179388041688388 €

Result list (right-click for options)

- 16:37:36 - bayes.BayesNet
- 16:56:10 - meta AdaBoostM1
- 16:58:22 - rules.DecisionTable
- 16:59:50 - trees.J48
- 17:11:39 - bayes.BayesNet
- 17:14:59 - meta.AdaBoostM1
- 17:15:39 - rules.DecisionTable
- 17:16:45 - trees.J48

(Nom) Average Rating

Start Stop

Correctly Classified Instances 3121 34.4748 %
 Incorrectly Classified Instances 5922 65.5252 %
 Kappa statistic 0.1323
 Mean absolute error 0.1499
 Root mean squared error 0.2738
 Relative absolute error 89.838 %
 Root relative squared error 94.787 %
 Total Number of Instances 9853

== Stratified cross-validation ==
 == Summary ==

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.800	0.000	?	0.800	?	?	?	0.536	0.800	0.5
0.800	0.000	?	0.800	?	?	?	0.535	0.825	1
0.800	0.000	?	0.800	?	?	?	0.544	0.832	1.5
0.800	0.000	?	0.800	?	?	?	0.548	0.871	2
0.800	0.000	?	0.800	?	?	?	0.548	0.871	2.5
0.545	0.000	1.000	0.545	0.706	0.786	0.769	0.667	0.667	3
0.800	0.000	?	0.800	?	?	?	0.558	0.239	3.5
1.000	0.075	0.277	1.000	0.434	0.186	0.561	0.276	0.276	4
0.800	0.000	?	0.800	?	?	?	0.569	0.892	4.5
0.800	0.000	?	0.800	?	?	?	0.547	0.844	5
0.345	0.220	?	0.345	?	?	?	0.591	0.264	

== Detailed Accuracy By Class ==

	a	b	c	d	e	f	g	h	i	j	<-- Classified as
a	3121	0	0	0	0	0	0	0	0	0	a = 0.5
b	0	5922	0	0	0	0	0	0	0	0	b = 1
c	0	0	5922	0	0	0	0	0	0	0	c = 1.5
d	0	0	0	5922	0	0	0	0	0	0	d = 2
e	0	0	0	0	5922	0	0	0	0	0	e = 2.5
f	0	0	0	0	0	5922	0	0	0	0	f = 3
g	0	0	0	0	0	0	5922	0	0	0	g = 3.5
h	0	0	0	0	0	0	0	5922	0	0	h = 4
i	0	0	0	0	0	0	0	0	5922	0	i = 4.5
j	0	0	0	0	0	0	0	0	0	5922	j = 5

== Confusion Matrix ==

	a	b	c	d	e	f	g	h	i	j	<-- Classified as
a	3121	0	0	0	0	0	0	0	0	0	a = 0.5
b	0	5922	0	0	0	0	0	0	0	0	b = 1
c	0	0	5922	0	0	0	0	0	0	0	c = 1.5
d	0	0	0	5922	0	0	0	0	0	0	d = 2
e	0	0	0	0	5922	0	0	0	0	0	e = 2.5
f	0	0	0	0	0	5922	0	0	0	0	f = 3
g	0	0	0	0	0	0	5922	0	0	0	g = 3.5
h	0	0	0	0	0	0	0	5922	0	0	h = 4
i	0	0	0	0	0	0	0	0	5922	0	i = 4.5
j	0	0	0	0	0	0	0	0	0	5922	j = 5

Status OK

➤ Decision Table

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier Choose **DecisionTable -X 1 -S "weka.attributeSelection.BestFirst -D 1 -N 5"**

Test options Use training set Supplied test set Set... Cross-validation Folds 10 Percentage split % 80

More options...

Classifier output

```

State search after 5 node expansions
Total number of subsets evaluated: 24
Metric of best subset found: 91.207
Evaluation (for feature selection): CV (leave one out)
Feature set: 2,7
Time taken to build model: 1.69 seconds

```

== Stratified cross-validation ==

== Summary ==

	Correctly Classified Instances	8113	89.6167 %
	Incorrectly Classified Instances	948	10.3833 %
Kappa statistic		0.8755	
Mean absolute error		0.0481	
Root mean squared error		0.1335	
Relative absolute error		28.8373 %	
Root relative squared error		46.2105 %	
Total Number of Instances		9053	

== Detailed Accuracy By Class ==

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.981	0.037	0.233	0.981	0.377	0.469	0.993	0.981	0.5	
0.957	0.000	1.000	0.957	0.978	0.978	0.993	0.966	1	
0.949	0.000	0.000	0.949	0.969	0.969	0.991	0.957	1.5	
0.925	0.000	1.000	0.925	0.961	0.959	0.991	0.953	2	
0.897	0.000	1.000	0.897	0.946	0.941	0.992	0.953	2.5	
0.883	0.000	1.000	0.883	0.938	0.929	0.993	0.967	3	
0.833	0.000	1.000	0.833	0.890	0.890	0.996	0.986	3.5	
0.853	0.000	0.000	0.853	0.989	0.989	0.992	0.986	4	
0.810	0.000	1.000	0.810	0.895	0.892	0.982	0.896	4.5	
0.983	0.000	1.000	0.983	0.992	0.991	0.997	0.988	5	
Weighted Avg.	0.896	0.023	0.936	0.896	0.908	0.991	0.988	0.954	

== Confusion Matrix ==

	a	b	c	d	e	f	g	h	i	j	<-- classified as
102	0	0	0	0	0	0	2	0	0	0	a = 0.5
4	200	0	0	0	0	0	5	0	0	0	b = 1
8	0	251	0	0	0	0	8	0	0	0	c = 1.5
12	0	0	544	0	0	0	32	0	0	0	d = 2
33	0	0	0	865	0	0	67	0	0	0	e = 2.5
38	0	0	0	0	150	0	143	0	0	0	f = 3
108	0	0	0	0	0	1643	222	0	0	0	g = 3.5
108	0	0	0	0	0	0	2169	0	0	0	h = 4
22	0	0	0	0	0	0	122	612	0	0	i = 4.5
2	0	0	0	0	0	0	4	0	356	0	j = 5

Status OK Log x 0

➤ J48

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier Choose **J48 -C 0.25 -M 2**

Test options Use training set Supplied test set Set... Cross-validation Folds 10 Percentage split % 80

More options...

Classifier output

```

Number of Leaves : 1279
Size of the tree : 1280

```

Time taken to build model: 0.01 seconds

== Stratified cross-validation ==

== Summary ==

	Correctly Classified Instances	8448	93.3171 %
	Incorrectly Classified Instances	685	6.6829 %
Kappa statistic		0.9193	
Mean absolute error		0.016	
Root mean squared error		0.0884	
Relative absolute error		0.9398 %	
Root relative squared error		38.6163 %	
Total Number of Instances		9053	

== Detailed Accuracy By Class ==

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.981	0.000	1.000	0.981	0.998	0.998	0.999	0.984	0.5	
0.976	0.000	1.000	0.976	0.988	0.988	0.999	0.984	1	
0.946	0.000	1.000	0.946	0.989	0.989	0.999	0.983	1.5	
0.946	0.000	1.000	0.946	0.972	0.972	0.997	0.975	2	
0.931	0.000	1.000	0.931	0.964	0.964	0.996	0.972	2.5	
0.908	0.000	1.000	0.908	0.982	0.952	0.944	0.995	3	
0.887	0.000	1.000	0.887	0.948	0.948	0.995	0.979	3.5	
1.000	0.000	1.000	1.000	0.993	1.000	1.000	1.000	4	
0.839	0.000	1.000	0.839	0.912	0.909	0.992	0.934	4.5	
0.989	0.000	1.000	0.989	0.994	0.994	0.999	0.999	5	
Weighted Avg.	0.933	0.022	0.947	0.933	0.935	0.922	0.995	0.975	

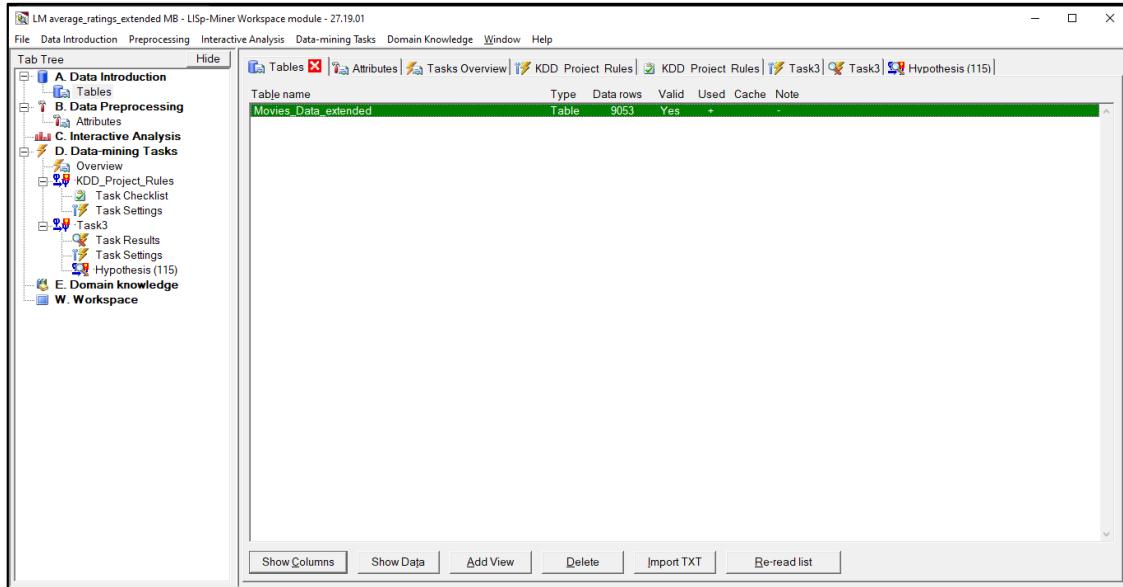
== Confusion Matrix ==

	a	b	c	d	e	f	g	h	i	j	<-- classified as
102	0	0	0	0	0	0	2	0	0	0	a = 0.5
3	204	0	0	0	0	0	5	0	0	0	b = 1
0	0	259	0	0	0	0	8	0	0	0	c = 1.5
0	0	0	556	0	0	0	32	0	0	0	d = 2
0	0	0	0	982	0	0	67	0	0	0	e = 2.5
0	0	0	0	0	1485	0	143	0	0	0	f = 3
0	0	0	0	0	0	178	222	0	0	0	g = 3.5
0	0	0	0	0	0	0	2277	0	0	0	h = 4
0	0	0	0	0	0	0	122	634	0	0	i = 4.5
0	0	0	0	0	0	0	4	0	356	0	j = 5

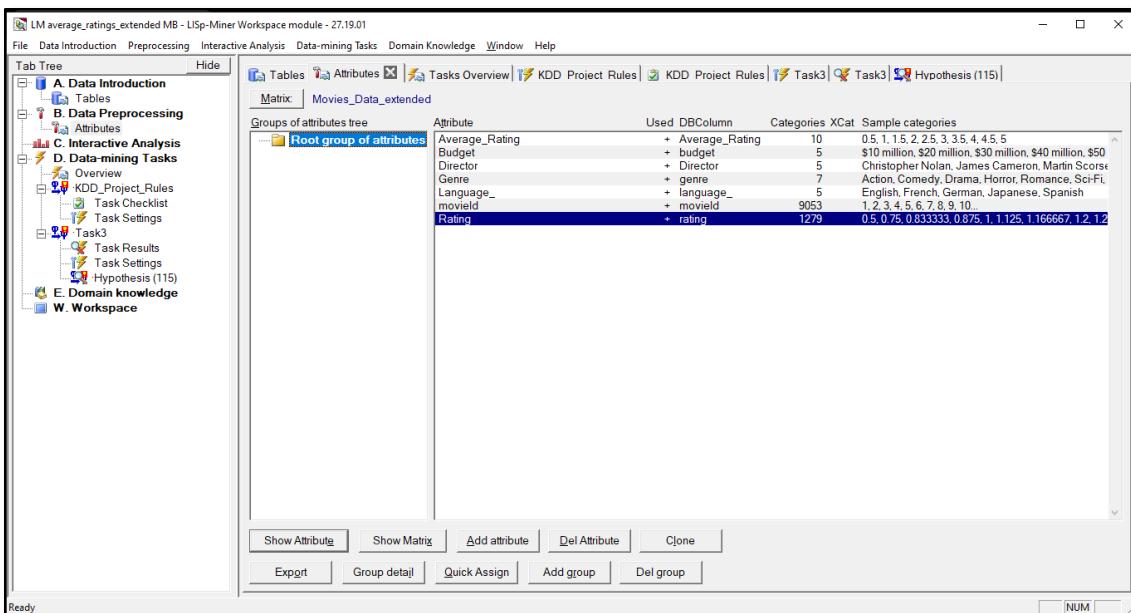
Status OK Log x 0

➤ LISP MINER

➤ Loading the extended dataset in LISP Miner:



➤ Overview of the file attributes:



The screenshot shows the LISP-Miner workspace interface with the 'Attributes' tab active. A table lists attributes for the 'Movies_Data_extended' dataset. The columns include Groups of attributes tree, Attribute, Used DBColumn, Categories, XCat, and Sample categories. Key attributes shown include Average_Rating, Budget, Director, Genre, Language_, movieId, and Rating. The 'Root group of attributes' is selected in the tree view on the left.

Groups of attributes tree	Attribute	Used DBColumn	Categories	XCat	Sample categories
Root group of attributes	Average_Rating	+ Average_Rating	10	0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5	
	Budget	+ budget	5	\$10 million, \$20 million, \$30 million, \$40 million, \$50	
	Director	+ Director	5	Christopher Nolan, James Cameron, Martin Scorsese	
	Genre	+ genre	7	Action, Comedy, Drama, Horror, Romance, Sci-Fi	
	Language_	+ language_	5	English, French, German, Japanese, Spanish	
	movieId	+ movieId	9053	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	
	Rating	+ rating	1279	0.5, 0.75, 0.833333, 0.875, 1, 1.125, 1.166667, 1.2, 1.25	

➤ 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 succendent of our action rules:

Antecedent:

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

Succendent:

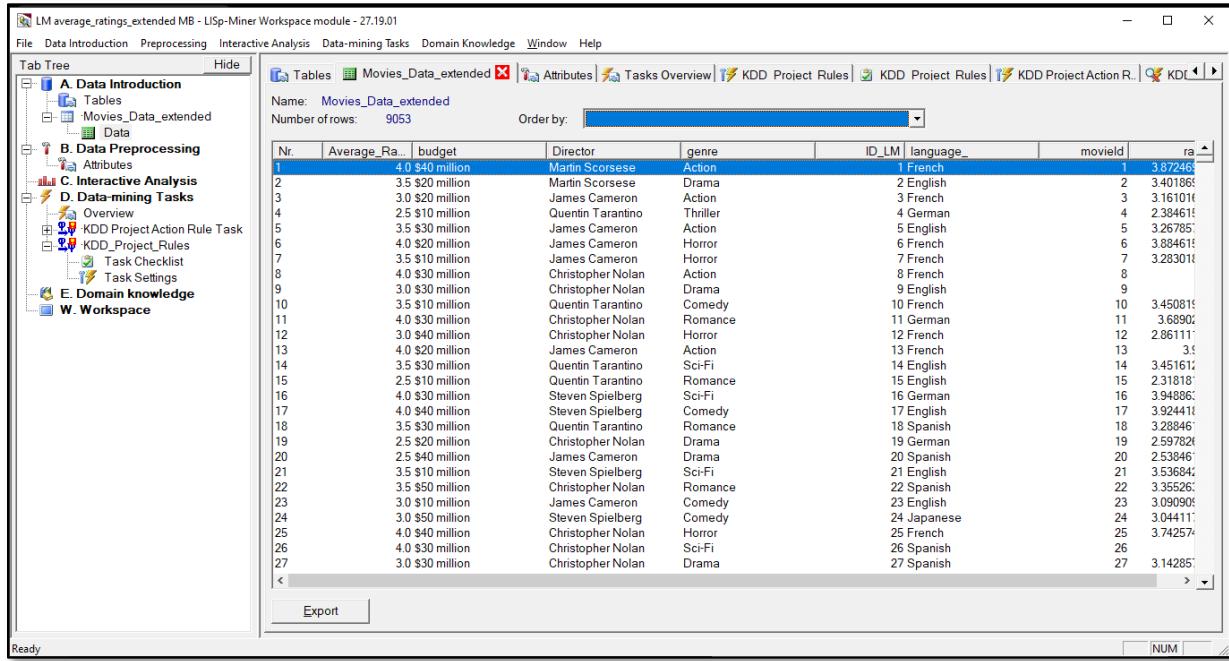
- Director – Variable Attribute (from Christopher Nolan to James Cameron)

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 succendent 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.

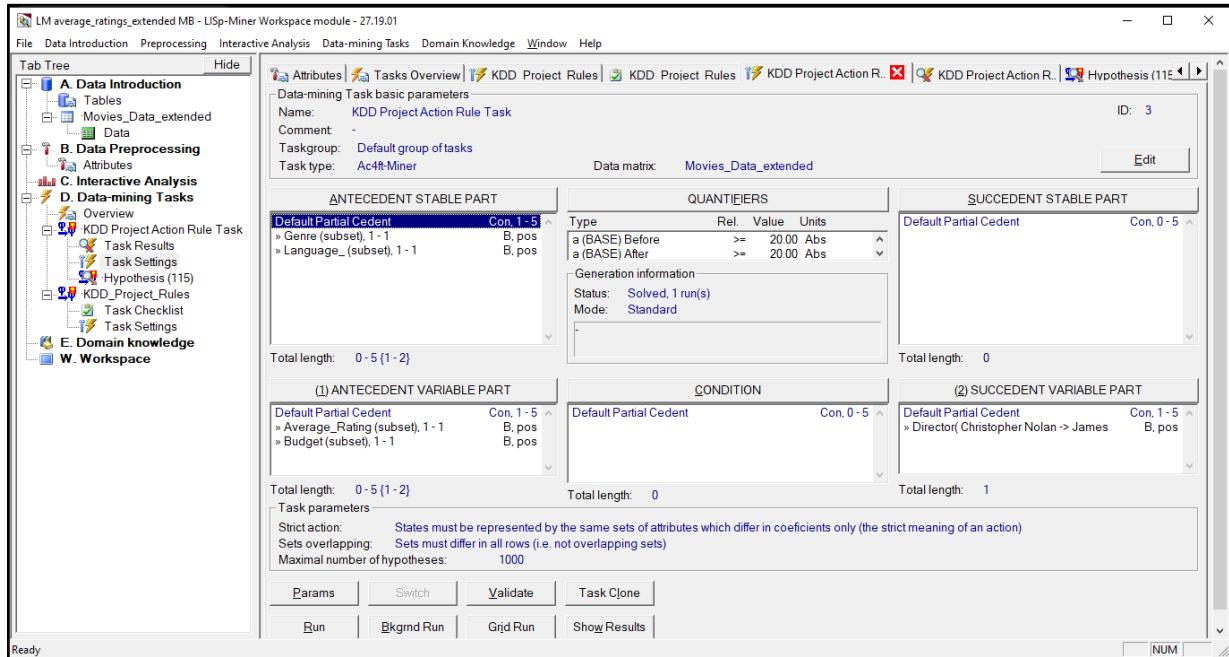
Our analytical task involves exploring the patterns between the antecedent and succendent attributes to identify actionable rules that can improve movie ratings. By defining our analytical task and selecting the relevant attributes, we can extract meaningful and informative patterns that can aid our 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:

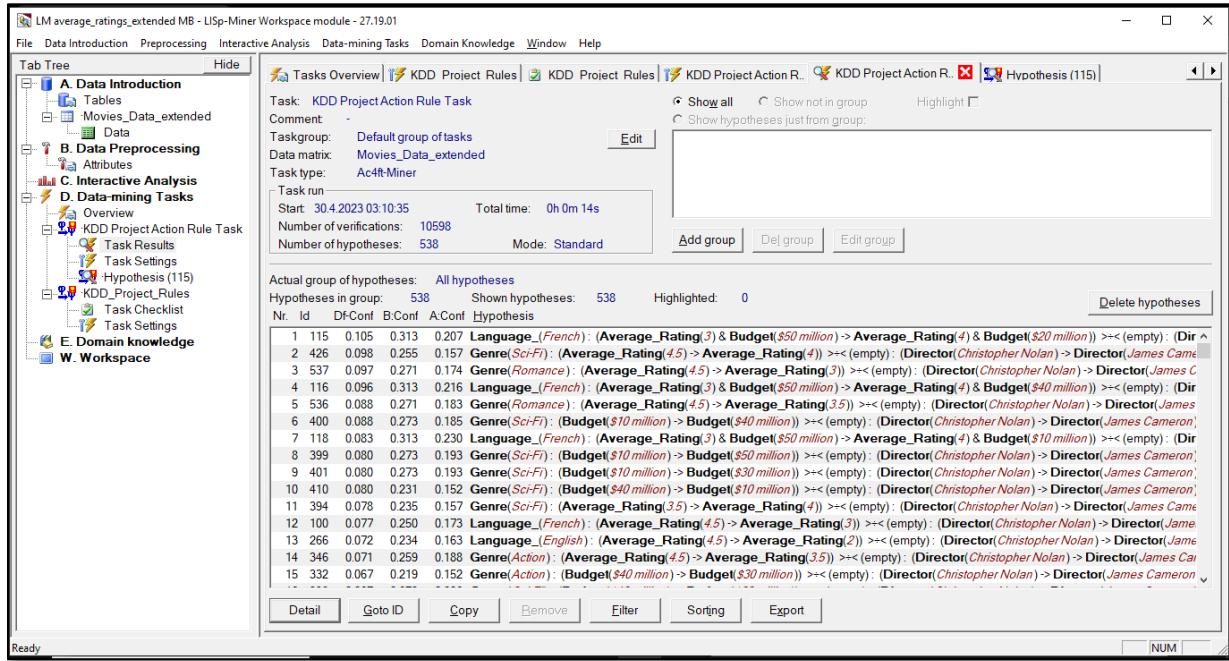


Nr	Average_Ra	budget	Director	genre	ID_LM	language_	movielid	ra
1	4.0	\$40 million	Martin Scorsese	Action	1	French	1	3.87245
2	3.5	\$20 million	Martin Scorsese	Drama	2	English	2	3.40165
3	3.0	\$20 million	James Cameron	Action	3	French	3	3.16101
4	2.5	\$10 million	Quentin Tarantino	Thriller	4	German	4	2.38461
5	3.5	\$30 million	James Cameron	Action	5	English	5	3.26785
6	4.0	\$20 million	James Cameron	Horror	6	French	6	3.88461
7	3.5	\$10 million	James Cameron	Horror	7	French	7	3.28301
8	4.0	\$30 million	Christopher Nolan	Action	8	French	8	
9	3.0	\$30 million	Christopher Nolan	Drama	9	English	9	
10	3.5	\$10 million	Quentin Tarantino	Comedy	10	French	10	3.45081
11	4.0	\$30 million	Christopher Nolan	Romance	11	German	11	3.6890
12	3.0	\$40 million	Christopher Nolan	Horror	12	French	12	2.86111
13	4.0	\$20 million	James Cameron	Action	13	French	13	3.
14	3.5	\$30 million	Quentin Tarantino	Sci-Fi	14	English	14	3.45161
15	2.5	\$10 million	Quentin Tarantino	Romance	15	English	15	2.31818
16	4.0	\$30 million	Steven Spielberg	Sci-Fi	16	German	16	3.94886
17	4.0	\$40 million	Steven Spielberg	Comedy	17	English	17	3.92441
18	3.5	\$30 million	Quentin Tarantino	Romance	18	Spanish	18	3.28846
19	2.5	\$20 million	Christopher Nolan	Drama	19	German	19	2.59782
20	2.5	\$40 million	James Cameron	Drama	20	Spanish	20	2.53846
21	3.5	\$10 million	Steven Spielberg	Sci-Fi	21	English	21	3.53684
22	3.5	\$50 million	Christopher Nolan	Romance	22	Spanish	22	3.35526
23	3.0	\$10 million	James Cameron	Comedy	23	English	23	3.09090
24	3.0	\$50 million	Steven Spielberg	Comedy	24	Japanese	24	3.04411
25	4.0	\$40 million	Christopher Nolan	Horror	25	French	25	3.74257
26	4.0	\$30 million	Christopher Nolan	Sci-Fi	26	Spanish	26	
27	3.0	\$30 million	Christopher Nolan	Drama	27	Spanish	27	3.14285

➤ Parameter tuning for hypothesis generation:



➤ Hypothesis on the dataset:



LM average_ratings_extended MB - LISp-Miner Workspace module - 27.19.01

Task: KDD Project Action Rule Task

Comment: -

Taskgroup: Default group of tasks

Data matrix: Movies_Data_extended

Task type: Ac4rl-Miner

Task run

Start: 30.4.2023 03:10:35 Total time: 0h 0m 14s

Number of verifications: 10598

Number of hypotheses: 538 Mode: Standard

Actual group of hypotheses: All hypotheses

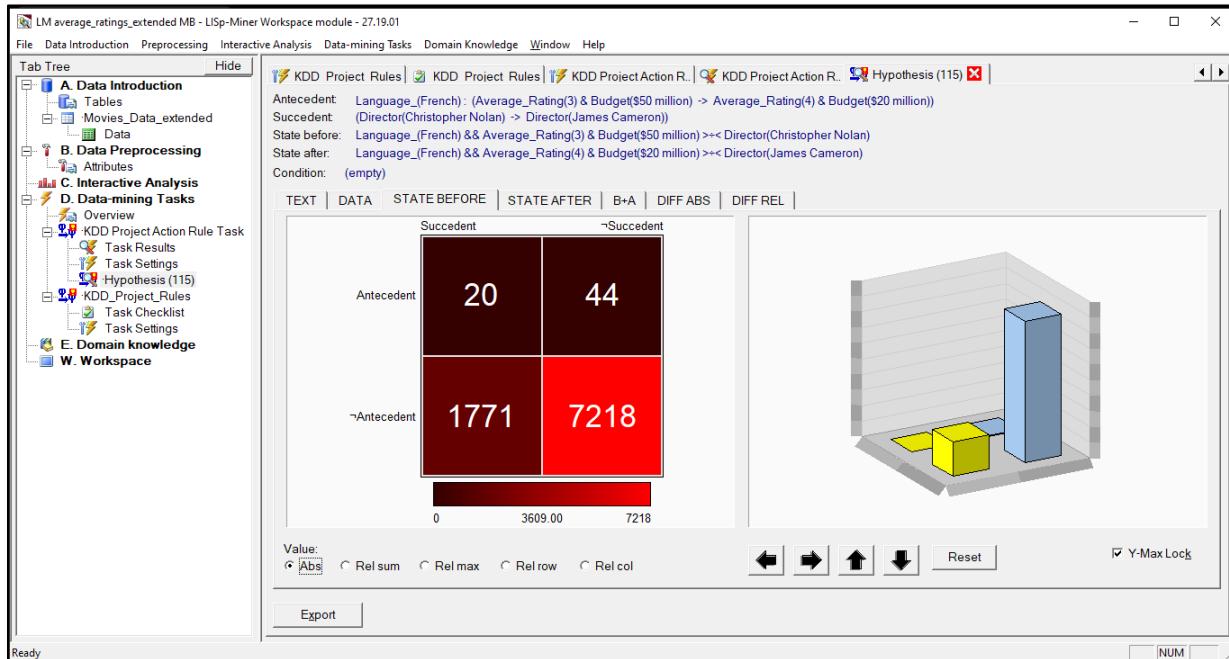
Hypotheses in group: 538 Shown hypotheses: 538 Highlighted: 0

Delete hypotheses

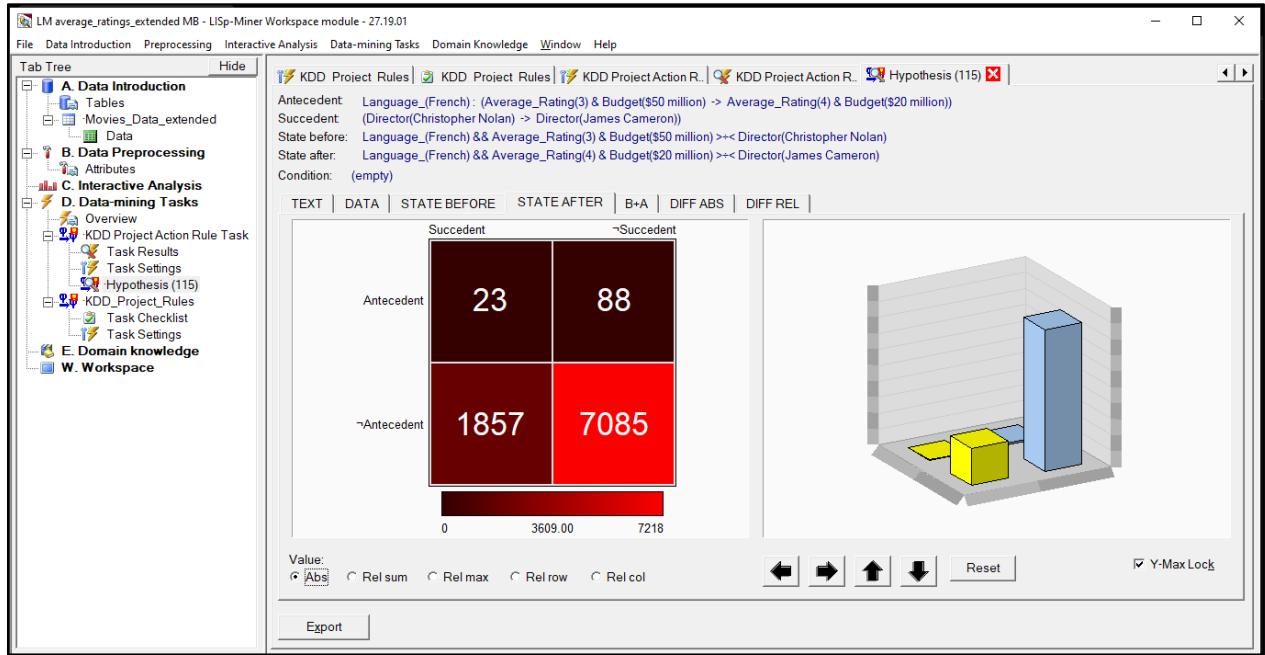
Nr.	Id	D-Conf	B-Conf	A-Conf	Hypothesis
1	115	0.105	0.313	0.207	Language_(French): (Average_Rating(3) & Budget(\$50 million)) >< (empty) : (Dir
2	426	0.098	0.255	0.157	Genre(Sci-Fi): (Average_Rating(4.5) > Average_Rating(4)) >< (empty) : (Director(Christopher Nolan) > Director(James Came
3	537	0.097	0.271	0.174	Genre(Romance): (Average_Rating(4.5) > Average_Rating(3)) >< (empty) : (Director(Christopher Nolan) > Director(James C
4	116	0.098	0.313	0.216	Language_(French): (Average_Rating(3) & Budget(\$50 million)) >< (empty) : (Dir
5	536	0.088	0.271	0.183	Genre(Romance): (Average_Rating(4.5) > Average_Rating(3.5)) >< (empty) : (Director(Christopher Nolan) > Director(James
6	400	0.088	0.273	0.185	Genre(Sci-Fi): (Budget(\$10 million) > Budget(\$40 million)) >< (empty) : (Director(Christopher Nolan) > Director(James Cameron)
7	118	0.083	0.313	0.230	Language_(French): (Average_Rating(3) & Budget(\$50 million)) >< (empty) : (Dir
8	399	0.080	0.273	0.193	Genre(Sci-Fi): (Budget(\$10 million) > Budget(\$50 million)) >< (empty) : (Director(Christopher Nolan) > Director(James Cameron)
9	401	0.080	0.273	0.193	Genre(Sci-Fi): (Budget(\$10 million) > Budget(\$30 million)) >< (empty) : (Director(Christopher Nolan) > Director(James Cameron)
10	410	0.080	0.231	0.152	Genre(Sci-Fi): (Budget(\$40 million) > Budget(\$10 million)) >< (empty) : (Director(Christopher Nolan) > Director(James Cameron)
11	394	0.078	0.235	0.157	Genre(Sci-Fi): (Average_Rating(3.5) > Average_Rating(4)) >< (empty) : (Director(Christopher Nolan) > Director(James Came
12	100	0.077	0.250	0.173	Language_(French): (Average_Rating(4.5) > Average_Rating(3)) >< (empty) : (Director(Christopher Nolan) > Director(Jame
13	266	0.072	0.234	0.163	Language_(English): (Average_Rating(4.5) > Average_Rating(2)) >< (empty) : (Director(Christopher Nolan) > Director(Jame
14	346	0.071	0.259	0.188	Genre(Action): (Average_Rating(4.5) > Average_Rating(3.5)) >< (empty) : (Director(Christopher Nolan) > Director(James Ca
15	332	0.067	0.219	0.152	Genre(Action): (Budget(\$40 million) > Budget(\$30 million)) >< (empty) : (Director(Christopher Nolan) > Director(James Cameron

Detail **Goto ID** **Copy** **Remove** **Filter** **Sorting** **Export**

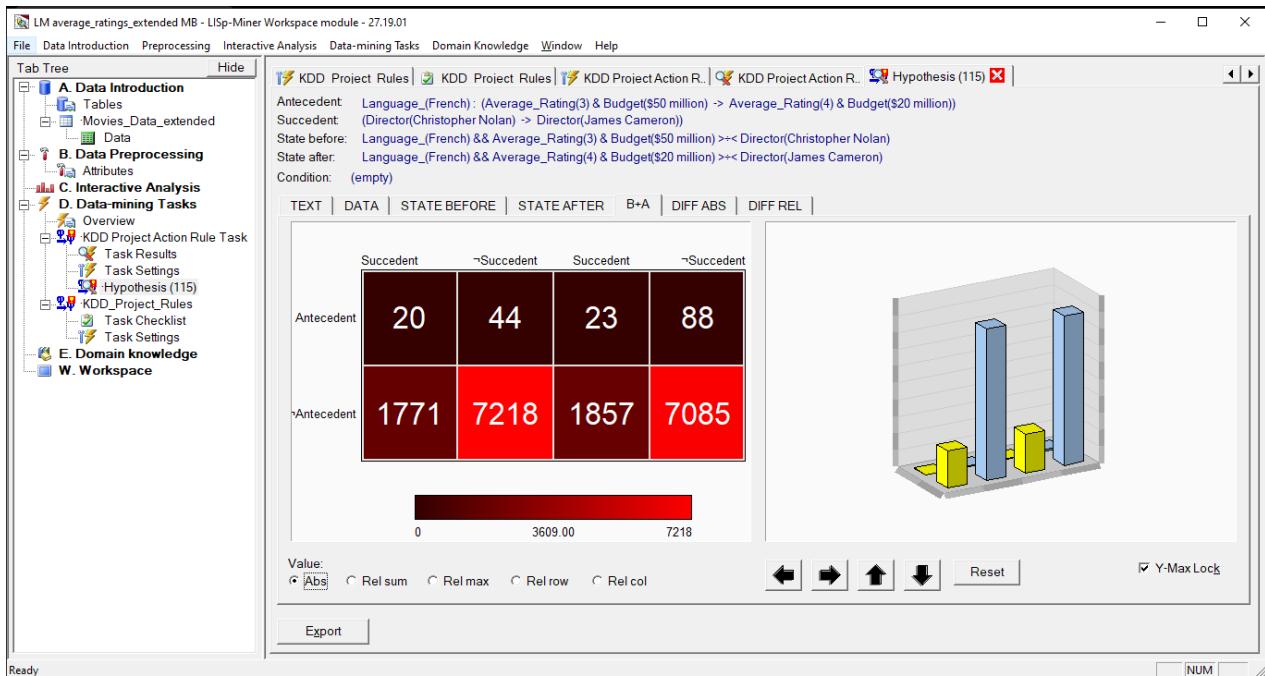
➤ Visualizing the Generated Hypothesis (1):



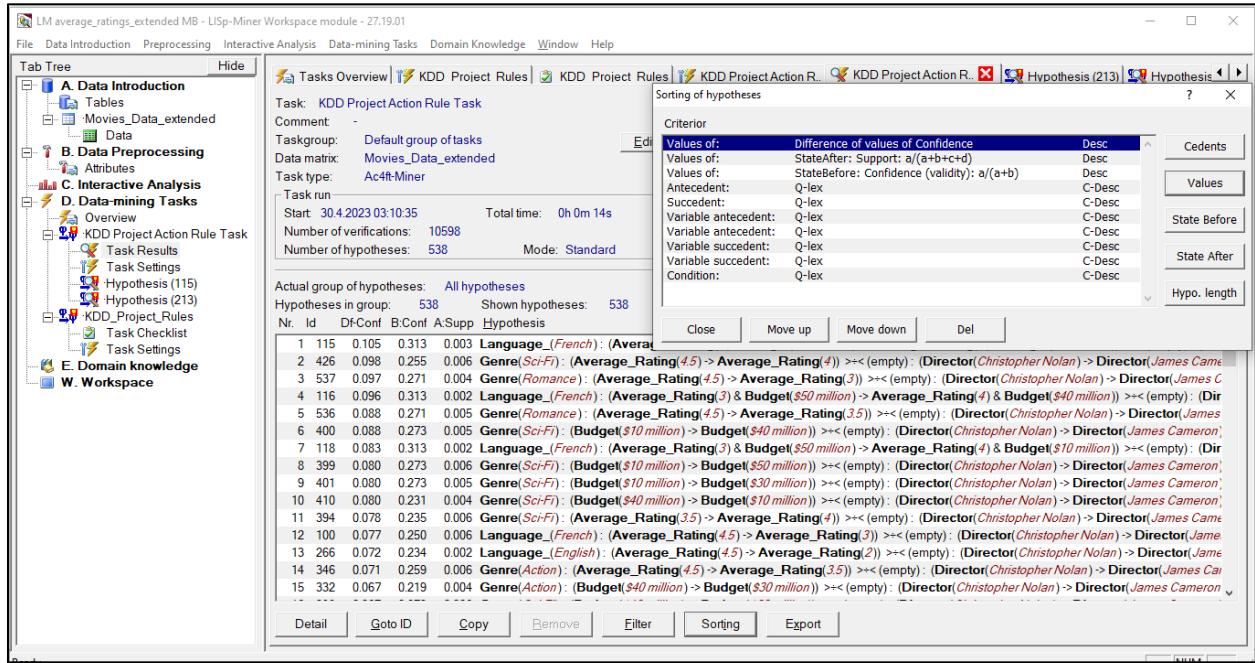
➤ Visualizing the Generated Hypothesis (2):



➤ 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.