

MED

METODY EKSPŁORACJI DANYCH W ODKRYWANIU WIEDZY

Sprawozdanie

z

projektu

Mikołaj Bańkowski 310408

Mateusz Kołacz 336360

Prowadzący

dr inż Robert Bembenik

12 stycznia 2025

Spis treści

1	Wprowadzenie i definicja problemu	2
1.1	Cel projektu	2
1.2	Temat projektu	2
1.3	Interpretacja oraz definicja problemu	2
1.3.1	Współczynnik podniesienia (lift)	3
1.3.2	Wsparcie relatywne (relative support)	3
1.3.3	Certainty Factor	3
1.3.4	Jaccard / coherence	3
1.3.5	Odd Ratio	4
1.3.6	Uwagi	4
2	Opis implementacji	5
2.1	Struktura projektu	5
2.2	Folder główny	6
2.2.1	run.py	6
2.2.2	apriori_comparison.py	6
2.2.3	apriori_eclat_comparison.py	6
2.2.4	metrics_comparison.py	6
2.2.5	requirements.txt	6
2.3	Folder src	6
2.3.1	dataclasses.py	6
2.3.2	metrics.py	6
2.3.3	apriori.py	6
2.3.4	eclat.py	7
2.3.5	apriori_utils.py	7
2.3.6	data_manager.py	7
3	Instrukcja użytkownika (jak uruchomić i korzystać z implementacji)	8
4	Charakterystyka wykorzystywanych zbiorów danych	9
4.1	Komentarz ogólny	9
4.2	Opis zbiorów danych z repozytorium UCIML	9
5	Wyniki eksperymentów pokazujących właściwości proponowanego rozwiązania	10
5.1	Apriori vs Apriori biblioteka	10
5.2	Apriori vs Eclat	11
5.3	Porównanie zachowania się wybranych miar w odniesieniu do wartości współ- czynnika podniesienia	13
5.3.1	Car Evaluation	13
5.3.2	Tic Tac Toe endgame	15
5.3.3	Nursery	16
6	Wnioski	18
7	Literatura	19

1 Wprowadzenie i definicja problemu

1.1 Cel projektu

Celem projektu było zbadanie własności zaimplementowanego przez siebie algorytmu: czas wykonania, uzyskiwane wyniki dla różnych wartości parametrów algorytmu oraz kilku zbiorów wejściowych. Zależnie od tematu projekt mógł być realizowany przez jedną lub dwie osoby. Implementacja algorytmu/rozwiązania miała zostać przygotowana w jednym z popularnych języków programowania.

1.2 Temat projektu

Implementacja algorytmu do odkrywania reguł asocjacyjnych (Apriori, Eclat) wraz z implementacją wyliczania miar: współczynnika podniesienia (lift) oraz 3 wybranych (conviction, cosine, gini, Jaccard / coherence, certainty factor, improvement, mutual Information i odd ratio). Porównanie zachowania się wybranych miar w odniesieniu do wartości współczynnika podniesienia.

Eksperymenty należało przeprowadzić z wykorzystaniem publicznie dostępnych zbiorów danych, np.:

- <https://archive.ics.uci.edu/datasets>
- <http://fimi.uantwerpen.be/data/>
- <https://github.com/deric/clustering-benchmark>
- <http://www.philippe-fournier-viger.com/spmf/index.php?link=datasets.php>

1.3 Interpretacja oraz definicja problemu

Postawione zadanie polegało na implementacji wybranego algorytmu a następnie weryfikacji rozwiązania i analizie zachowania w specyficznych warunkach. Zaimplementowany został algorytm Apriori i Eclat. Językiem użytym do wykorzystania zadania był Python w wersji 3.12.3. Rozwiązanie miało umożliwiać wyznaczanie reguł asocjacyjnych z dyskretnych zbiorów danych a następnie obliczanie wartości wybranych miar:

1.3.1 Współczynnik podniesienia (lift)

Współczynnik podniesienia (lift) to miara, która ocenia siłę reguły asocjacyjnej. Określa, jak bardzo wystąpienie zdarzenia B jest bardziej prawdopodobne, gdy występuje zdarzenie A , w porównaniu do sytuacji, gdyby zdarzenia A i B były niezależne.

$$\text{Lift}(A \rightarrow B) = \frac{\text{support}(A \cap B)}{\text{support}(A) \times \text{support}(B)}$$

Interpretacja:

- **Lift = 1:** Brak zależności między A i B .
- **Lift > 1:** Zdarzenie A zwiększa prawdopodobieństwo wystąpienia zdarzenia B .
- **Lift < 1:** Zdarzenie A zmniejsza prawdopodobieństwo wystąpienia zdarzenia B .

1.3.2 Wsparcie relatywne (relative support)

Wsparcie relatywne (relative support) to miara, która określa, jak często reguła asocjacyjna występuje w zbiorze danych w porównaniu do całkowitej liczby transakcji.

$$\text{Relative Support}(A \rightarrow B) = \text{support}(A \cap B)$$

Interpretacja:

- Wartość wsparcia relatywnego mieści się w przedziale od 0 do 1.
- Wysoka wartość oznacza, że reguła jest często spotykana w zbiorze danych.

1.3.3 Certainty Factor

Certainty factor (CF) to miara, która określa stopień pewności, że zdarzenie B nastąpi, gdy wystąpi zdarzenie A . Jest to miara dodatkowego wsparcia udzielanego przez zdarzenie A zdarzeniu B .

$$\text{CF}(A \rightarrow B) = \frac{\text{support}(A \cap B) - \text{support}(A) \times \text{support}(B)}{1 - \text{support}(B)}$$

Interpretacja:

- **CF = 0:** Zdarzenie A nie wpływa na prawdopodobieństwo zdarzenia B .
- **CF > 0:** Zdarzenie A zwiększa prawdopodobieństwo wystąpienia zdarzenia B .
- **CF < 0:** Zdarzenie A zmniejsza prawdopodobieństwo wystąpienia zdarzenia B .

1.3.4 Jaccard / coherence

Miara Jaccarda (coherence) to współczynnik podobieństwa między dwoma zbiorami, w tym przypadku między zbiorem transakcji zawierających A i B .

$$\text{Jaccard}(A \rightarrow B) = \frac{\text{support}(A \cap B)}{\text{support}(A) + \text{support}(B) - \text{support}(A \cap B)}$$

Interpretacja:

- Wysoka wartość oznacza dużą współzależność między A i B .

1.3.5 Odd Ratio

Miara odd ratio określa stosunek szans wystąpienia zdarzenia A w obecności zdarzenia B do szans wystąpienia zdarzenia A w braku zdarzenia B .

$$\text{Odd Ratio}(A \rightarrow B) = \frac{\text{support}(A \cap B) \times \text{support}(\neg A \cap \neg B)}{\text{support}(A \cap \neg B) \times \text{support}(\neg A \cap B)}$$

Interpretacja:

- **Odd Ratio = 1:** Brak zależności między A i B .
- **Odd Ratio > 1:** Zdarzenie A jest bardziej prawdopodobne, gdy występuje zdarzenie B .
- **Odd Ratio < 1:** Zdarzenie A jest mniej prawdopodobne, gdy występuje zdarzenie B .

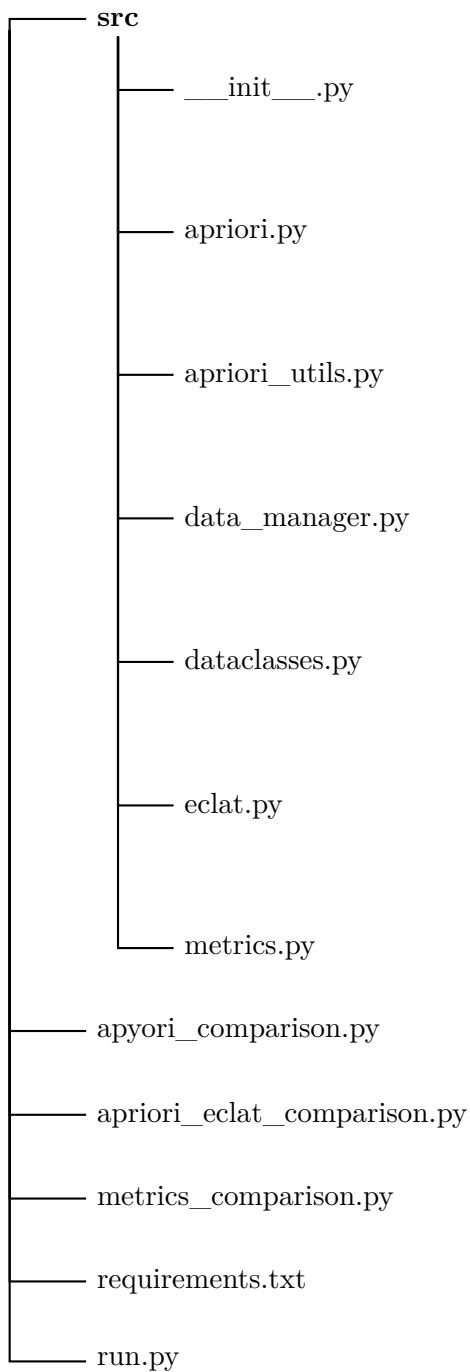
1.3.6 Uwagi

Rozwiązanie działa zarówno z danymi podanymi w formie pliku `txt`, jak i z wybranymi zbiorami dyskretnymi z repozytorium UCIML. W ramach przeprowadzonych testów przeanalizowana została dokładność rozwiązania zestawiona z innymi, bardziej rozbudowanymi rozwiązaniami dostępnymi w formie biblioteki języka Python – `apriori`[5]. Dodatkowo przeprowadzono porównanie algorytmów Apriori i Eclat, które dostarczyło cennych informacji na temat ich wydajności i skuteczności w analizie zbiorów danych o różnej wielkości i gęstości. Następnie analizie poddane zostało zachowanie się wybranych miar w odniesieniu do wartości współczynnika podniesienia.

2 Opis implementacji

W ramach implementacji wydzielone zostały abstrakcje obiektów pełniące określone role i zapewniające spójny interfejs programowania. Struktura kodu została zaprojektowana tak, aby zachować ortogonalność elementów oprogramowania, co oznacza minimalizację zależności między poszczególnymi modułami oraz łatwość wprowadzania modyfikacji i rozbudowy.

2.1 Struktura projektu



2.2 Folder główny

2.2.1 run.py

To główny skrypt aplikacji, łączący w sobie działanie wszystkich zdefiniowanych klas i umożliwiający korzystanie z nich za pomocą interfejsu wiersza poleceń.

2.2.2 apyori_comparison.py

Skrypt służy do porównywania wydajności i wyników własnej implementacji algorytmu Apriori z wersją dostępną w bibliotece apyori. Analizowane są takie aspekty jak zgodność reguł, wartości współczynników pewności (confidence) oraz czas wykonywania algorytmu na różnych zbiorach danych i parametrach analizy.

2.2.3 apriori_eclat_comparison.py

Skrypt służy do porównywania wydajności i wyników własnej implementacji algorytmu Apriori z wynikami własnej implementacji algorytmu Eclat. Analizowane są takie aspekty jak zgodność reguł, wartości współczynników pewności (confidence) oraz czas wykonywania algorytmu na różnych zbiorach danych i parametrach analizy.

2.2.4 metrics_comparison.py

Skrypt odpowiada za analizę oraz porównywanie wybranych miar jakości reguł asocjacyjnych, takich jak Lift, Jaccard, Certainty Factor i Odds Ratio. Jego główną funkcjonalnością jest zbadanie ich zachowania w odniesieniu do wartości współczynnika podniesienia (lift) oraz ocena przydatności poszczególnych miar w interpretacji wyników analizy.

2.2.5 requirements.txt

Plik zawiera listę zależności i bibliotek niezbędnych do poprawnego działania projektu. Znajdują się w nim informacje o wersjach wymaganych bibliotek Python

2.3 Folder src

2.3.1 dataclasses.py

Jest to moduł zawierający abstrakcyjne obiekty reprezentujące pojedynczy zbiór atrybutów, regułę asocjacyjną oraz powiązane z nią metryki. Taki podział jest zgodny z paradygmatem programowania obiektowego i zapewnia możliwość definiowania metod i zachowań obiektów odzwierciedlających logikę biznesową.

2.3.2 metrics.py

Jest to klasa, która enkapsuluje w sobie logikę umożliwiającą wyznaczanie wybranych metryk na podstawie dostarczonych zbiorów reguł i zbiorów atrybutów.

2.3.3 apriori.py

To główna klasa, która skupia w sobie implementacje algorytmu Apriori i zapewnia interfejs zwracający reguły asocjacyjne na podstawie dostarczonego do algorytmu zbioru danych.

2.3.4 eclat.py

To główna klasa, która skupia w sobie implementacje algorytmu Eclat i zapewnia interfejs zwracający reguły asocjacyjne na podstawie dostarczonego do algorytmu zbioru danych.

2.3.5 apriori_utils.py

To klasa pomocnicza, definiująca w sobie metody używane przez algorytm *Apriori* oraz algorytm *Eclat*, ale nie bezpośrednio z nim związane - takie jak łączenie zbiorów czy graficzna prezentacja wyliczonych metryk.

2.3.6 data_manager.py

To klasa, która jest odpowiedzialna za dostarczenie danych do pozostałych obiektów. Umożliwia czerpanie danych zarówno z lokalnego pliku tekstowego oraz zbiorów danych udostępnianych przez repozytorium UCIML.

3 Instrukcja użytkownika (jak uruchomić i korzystać z implementacji)

Poniższy skrypt umożliwia uruchomienie algorytmu, który służy do wykrywania często współwystępujących wzorców w zbiorze danych. Aby korzystać ze skryptu, należy uruchomić go z odpowiednimi flagami w linii poleceń. Poniżej omówione zostały dostępne flagi oraz przykłady ich użycia:

- **-f, --input-file:** Ta opcja pozwala na wskazanie pliku wejściowego zawierającego zbiór danych. Wymagane jest aby plik pod podaną ścieżką istniał. Oto przykład użycia:

```
python script.py -f path/to/dataset.csv
```

- **-a, --algorithm:** Ta opcja określa, który algorytm ma zostać użyty do analizy danych. Użytkownik może wybrać pomiędzy dostępnymi algorytmami, takimi jak **Apriori** lub **Eclat**, w zależności od charakterystyki zbioru danych oraz celu analizy.

```
python script.py -a eclat
```

- **-u, --UCI-dataset:** Ta opcja umożliwia pobranie zadanego zbioru danych z bazy UCI Machine Learning Repository. Domyślnie ustawione jest "car_evaluation". Pozostałe możliwe zbiory danych to "tic_tac_toe_endgame" oraz "nursery". Przykład użycia:

```
python script.py -u nursery
```

- **-s, --min-support:** Ta opcja określa minimalną wartość wsparcia (support), która jest używana do filtrowania rzadkich elementów. Domyślnie jest to 0.15. Przykład użycia:

```
python script.py -s 0.2
```

- **-c, --min-confidence:** Ta opcja określa minimalną wartość ufności (confidence), która jest używana do wyznaczania silnych reguł asocjacyjnych. Domyślnie jest to 0.6. Przykład użycia:

```
python script.py -c 0.7
```

Przykład uruchomienia skryptu z kilkoma flagami jednocześnie:

```
python script.py -a eclat -f path/to/dataset.csv -s 0.2 -c 0.7
```

W przypadku, gdy nie zostanie podany plik wejściowy, skrypt spróbuje pobrać domyślny zbiór danych z UCI (chyba, że zostanie określony inny zbiór przy użyciu flagi -u).

4 Charakterystyka wykorzystywanych zbiorów danych

4.1 Komentarz ogólny

Wykorzystane zostały trzy zbiory danych z repozytorium UCIML o charakterze dyskretnym. W przyszłości możliwa jest implementacja dodatkowej funkcjonalności, która miałaby na celu dyskretyzację zbiorów danych z wartościami rzeczywistymi, chociażby na podstawie kategoryzacji *niska, średnia, wysoka* wartość atrybutu.

4.2 Opis zbiorów danych z repozytorium UCIML

1. [Car Evaluation](#) - Zbiór danych dotyczący oceny samochodów. Zawiera informacje na temat cech samochodu (np. cena, liczba drzwi, pojemność bagażnika, bezpieczeństwo) oraz decyzji dotyczących oceny samochodu (klasyfikacja jako: unacceptable, acceptable, good, very good).

Zbiór ten zawiera 1728 rekordów, z czego każdy opisuje pojedynczy egzemplarz samochodu. Dane są wyraźnie zdefiniowane i sklasyfikowane, co ułatwia analizę i zrozumienie danych.

2. [Tic Tac Toe Endgame](#) - Zbiór danych dotyczący końcówki gry w kółko i krzyżyk. Zawiera informacje o ułożeniu planszy gry (np. pozycje poszczególnych pól: top-left, top-middle, top-right, itd.) oraz decyzji dotyczących oceny końcowego stanu gry (klasyfikacja jako: win, loss, draw).

Zbiór ten zawiera 958 rekordów, z czego każdy opisuje pojedynczą sytuację w końcówce gry w kółko i krzyżyk. Dane są w pełni dyskretne i reprezentują wszystkie możliwe kombinacje ułożenia pól na planszy w końcowej fazie gry.

3. [Nursery](#) - Zbiór danych dotyczący oceny przedszkoli. Zawiera informacje o cechach przedszkola (np. ilość dzieci, ilość osób dorosłych, poziom opieki) oraz decyzji dotyczących oceny przedszkola (klasyfikacja jako: recommend, priority, not recommend, very-recom).

Zbiór ten zawiera 12960 rekordów, z czego każdy opisuje pojedyncze przedszkole. Dane są zróżnicowane i uwzględniają różne aspekty funkcjonowania przedszkoli, co pozwala na wszechstronną analizę.

5 Wyniki eksperymentów pokazujących właściwości proponowanego rozwiązania

5.1 Apriori vs Apriori biblioteka

W ramach tej części testów, stworzona implementacja algorytmu Apriori była porównywana z dostępną w języku python biblioteką - **apyori** możliwą do pobrania przez:

```
pip install apyori
```

Porównane zostały takie własności jak procent odnalezionych reguł asocjacyjnych, procent poprawnie wyznaczonych wartości wsp. confidence (w odniesieniu do wyników zwracanych przez implementację apyori) oraz czasy wykonywania obydwu rozwiązań.

Całość dostępna jest w skrypcie apyori_comparison.py, dostępnym w kodzie źródłowym rozwiązania. Porównanie odbyło się przy uwzględnieniu różnych zbiorów danych i wartości współczynników min_support oraz min_confidence.

Czasy egzekucji były mierzone dla 1 000 000 wywołań funkcji wyznaczającej reguły asocjacyjne dla każdego zestawu parametrów wejściowych.

Dataset	Min Support	Min Confidence	Rules Match	Confidence Match	My Exec Time	Apyori Exec Time
car_evaluation	0.15	0.15	100.00%	100.00%	2.041 ms	0.271 ms
car_evaluation	0.15	0.5	100.00%	100.00%	2.623 ms	0.311 ms
car_evaluation	0.15	0.8	100.00%	100.00%	2.622 ms	0.435 ms
car_evaluation	0.5	0.15	100.00%	100.00%	2.271 ms	0.353 ms
car_evaluation	0.5	0.5	100.00%	100.00%	2.919 ms	0.281 ms
car_evaluation	0.5	0.8	100.00%	100.00%	2.063 ms	0.268 ms
car_evaluation	0.8	0.15	100.00%	100.00%	2.473 ms	0.329 ms
car_evaluation	0.8	0.5	100.00%	100.00%	2.361 ms	0.328 ms
car_evaluation	0.8	0.8	100.00%	100.00%	2.139 ms	0.331 ms
tic_tac_toe_endgame	0.15	0.15	100.00%	100.00%	2.341 ms	0.415 ms
tic_tac_toe_endgame	0.15	0.5	100.00%	100.00%	2.844 ms	0.321 ms
tic_tac_toe_endgame	0.15	0.8	100.00%	100.00%	2.164 ms	0.268 ms
tic_tac_toe_endgame	0.5	0.15	100.00%	100.00%	2.242 ms	0.271 ms
tic_tac_toe_endgame	0.5	0.5	100.00%	100.00%	1.949 ms	0.264 ms
tic_tac_toe_endgame	0.5	0.8	100.00%	100.00%	2.214 ms	0.291 ms
tic_tac_toe_endgame	0.8	0.15	100.00%	100.00%	2.035 ms	0.276 ms
tic_tac_toe_endgame	0.8	0.5	100.00%	100.00%	1.994 ms	0.272 ms
tic_tac_toe_endgame	0.8	0.8	100.00%	100.00%	2.190 ms	0.341 ms
nursery	0.15	0.15	100.00%	100.00%	2.926 ms	0.259 ms
nursery	0.15	0.5	100.00%	100.00%	3.055 ms	0.271 ms
nursery	0.15	0.8	100.00%	100.00%	3.021 ms	0.263 ms
nursery	0.5	0.15	100.00%	100.00%	2.088 ms	0.258 ms
nursery	0.5	0.5	100.00%	100.00%	2.072 ms	0.257 ms
nursery	0.5	0.8	100.00%	100.00%	2.108 ms	0.264 ms
nursery	0.8	0.15	100.00%	100.00%	2.039 ms	0.257 ms
nursery	0.8	0.5	100.00%	100.00%	2.038 ms	0.265 ms
nursery	0.8	0.8	100.00%	100.00%	2.064 ms	0.262 ms

Rysunek 1: Porównanie wyników własnej implementacji i biblioteki apyori

Na powyższym rysunku widać, że w każdym przypadku implementacja wykonana w ramach niniejszego projektu osiągnęła 100% pokrycia odkrytych reguł asocjacyjnych i wsp. confidence

dla różnych parametrów wejściowych. Oznacza to, że nie pomija ona żadnej z reguł i jest całkowicie spójna z rozwiązaniem dostarczonym przez bibliotekę apyori. Mniej korzystnie wyglądają natomiast czasy egzekucji obydwu implementacji.

Własna implementacja algorytmu zajmuje średnio 10 razy dłużej, niż dostarczone rozwiązanie biblioteczne. Jest to jednak akceptowalna różnica - biblioteka apyori była tworzona znacznie dokładniej i ze znacznie większym naciskiem na doprecyzowanie niż niniejszy projekt.

Po osiągnięciu satysfakcjonujących wyników i dokładności działania algorytmu, twórcy mogli skupić się na aspektach wydajnościowych. W ramach dalszego rozwoju niniejszego projektu, również istniałaby możliwość spróbowania poprawienia wyników czasu egzekucji dostarczonego rozwiązania, ale wykracza to poza zakres projektu.

5.2 Apriori vs Eclat

W ramach tej części testów, stworzona implementacja algorytmu Apriori była porównywana z stworzoną implementacją algorytmu Eclat.

Porównane zostały takie własności jak procent odnalezionych reguł asocjacyjnych, procent poprawnie wyznaczonych wartości wsp. confidence (w odniesieniu do wyników zwracanych przez implementację apyori) oraz czasy wykonywania obydwu rozwiązań.

Całość dostępna jest w skrypcie `apriori_eclat_comparison.py`, dostępnym w kodzie źródłowym rozwiązania. Porównanie odbyło się przy uwzględnieniu różnych zbiorów danych i wartości współczynników `min_support` oraz `min_confidence`.

Dataset	Min Support	Min Confidence	Rules Match	Confidence Match	Apriori Exec Time	Eclat Exec Time
car_evaluation	0.15	0.15	100.00%	100.00%	1.742 s	1.182 s
car_evaluation	0.15	0.5	100.00%	100.00%	1.755 s	1.194 s
car_evaluation	0.15	0.8	100.00%	100.00%	1.795 s	1.252 s
car_evaluation	0.5	0.15	100.00%	100.00%	1.766 s	1.208 s
car_evaluation	0.5	0.5	100.00%	100.00%	1.743 s	1.237 s
car_evaluation	0.5	0.8	100.00%	100.00%	1.724 s	1.215 s
car_evaluation	0.8	0.15	100.00%	100.00%	1.767 s	1.211 s
car_evaluation	0.8	0.5	100.00%	100.00%	1.766 s	1.225 s
car_evaluation	0.8	0.8	100.00%	100.00%	1.788 s	1.217 s
tic_tac_toe_endgame	0.15	0.15	100.00%	100.00%	1.752 s	1.214 s
tic_tac_toe_endgame	0.15	0.5	100.00%	100.00%	1.833 s	1.312 s
tic_tac_toe_endgame	0.15	0.8	100.00%	100.00%	1.905 s	1.212 s
tic_tac_toe_endgame	0.5	0.15	100.00%	100.00%	1.728 s	1.194 s
tic_tac_toe_endgame	0.5	0.5	100.00%	100.00%	1.727 s	1.197 s
tic_tac_toe_endgame	0.5	0.8	100.00%	100.00%	1.749 s	1.201 s
tic_tac_toe_endgame	0.8	0.15	100.00%	100.00%	1.723 s	1.198 s
tic_tac_toe_endgame	0.8	0.5	100.00%	100.00%	1.733 s	1.197 s
tic_tac_toe_endgame	0.8	0.8	100.00%	100.00%	1.735 s	1.203 s
nursery	0.15	0.15	100.00%	100.00%	2.738 s	1.410 s
nursery	0.15	0.5	100.00%	100.00%	2.762 s	1.388 s
nursery	0.15	0.8	100.00%	100.00%	2.726 s	1.376 s
nursery	0.5	0.15	100.00%	100.00%	1.889 s	1.253 s
nursery	0.5	0.5	100.00%	100.00%	1.888 s	1.257 s
nursery	0.5	0.8	100.00%	100.00%	1.892 s	1.241 s
nursery	0.8	0.15	100.00%	100.00%	1.860 s	1.257 s
nursery	0.8	0.5	100.00%	100.00%	1.908 s	1.342 s
nursery	0.8	0.8	100.00%	100.00%	1.885 s	1.257 s

Rysunek 2: Porównanie wyników algorytmu Apriori i Eclat własnej implementacji

Oba algorytmy, Apriori i Eclat, wygenerowały identyczne reguły asocjacyjne dla podanych

parametrów minimalnego wsparcia (`min_support`) i minimalnej pewności (`min_confidence`). Zarówno wskaźnik zgodności reguł, jak i wskaźnik zgodności pewności, osiągnął 100% dla wszystkich trzech analizowanych zbiorów danych: `car_evaluation`, `tic_tac_toe_endgame` i `nursery`. Jednakże różnica między algorytmami stała się zauważalna w kontekście czasu wykonania, gdzie Eclat konsekwentnie przewyższał Apriori, zwłaszcza przy większych i bardziej złożonych zbiorach danych, takich jak `nursery`. Stabilność czasów wykonania była również lepsza w przypadku Eclat, podczas gdy czasy dla Apriori wykazywały większe wahania w zależności od parametrów. Można zatem stwierdzić, że Eclat jest bardziej efektywny w przetwarzaniu dużych i gęstych zbiorów danych, co wynika z jego unikalnego podejścia do reprezentacji i operacji na danych.

Przewaga czasowa algorytmu Eclat nad Apriori wynika głównie z różnic w reprezentacji danych i sposobie ich przetwarzania. Apriori analizuje dane poziomo, traktując każdą transakcję jako osobną jednostkę i wielokrotnie skanuje bazę danych w poszukiwaniu częstych wzorców. W miarę wzrostu liczby kombinacji złożoność obliczeniowa rośnie wykładniczo, co znacząco wydłuża czas przetwarzania. Z kolei Eclat korzysta z pionowej reprezentacji danych w postaci zbiorów identyfikatorów transakcji (TID-sets). Dzięki temu algorytm wykonuje operacje przecięć na zbiorach, co minimalizuje konieczność wielokrotnego przeszukiwania bazy danych. Ponadto rekurencyjna natura algorytmu Eclat pozwala na bardziej efektywne przeszukiwanie przestrzeni rozwiązań, co jest szczególnie widoczne w przypadku dużych, gęstych zbiorów danych. Dodatkowo Eclat lepiej radzi sobie z dużą liczbą unikalnych elementów w danych, co dodatkowo przekłada się na przewagę czasową nad Apriori.

Algorytm Apriori jest lepszym wyborem w sytuacjach, gdy mamy do czynienia z mniejszymi i bardziej rozproszonymi zbiorami danych o stosunkowo niewielkiej liczbie unikalnych elementów. Jego interpretowalność i prostota implementacji sprawiają, że jest szczególnie użyteczny w scenariuszach, gdzie analiza wyników i ich przejrzystość są kluczowe. Eclat natomiast jest bardziej efektywny w przypadku dużych i gęstych zbiorów danych, gdzie liczba kombinacji elementów jest znacznie większa. Dzięki operacjom na zbiorach identyfikatorów transakcji (TID-sets) oraz rekurencyjnemu podejściu, Eclat skutecznie redukuje liczbę powtórnych skanów bazy danych, co znacząco skraca czas przetwarzania. W praktyce, dla małych i średnich zbiorów danych, oba algorytmy będą działać efektywnie, ale dla dużych, złożonych zestawów danych, zaleca się użycie algorytmu Eclat.

5.3 Porównanie zachowania się wybranych miar w odniesieniu do wartości współczynnika podniesienia

W ramach niniejszej analizy, własna implementacja algorytmu Apriori została użyta do wyznaczenia reguł asocjacyjnych i powiązanych z nimi metryk na trzech dyskretnych zbiorach danych pochodzących z UCIML repo: "car_evaluation", "tic_tac_toe_endgame" oraz "nursery".

Uzyskane wyniki zostały posortowane na podstawie wsp. podniesienia - lift a następnie zaprezentowane w postaci tabeli ogólnej i tabeli uśrednionych wartości metryk w zależności od wartości wsp. lift.

W związku z nieczytelną i długą listą wyznaczonych reguł asocjacyjnych w dwóch ostatnich zbiorach danych, w niniejszym sprawozdaniu zamieszczono jedynie przykładową pełną tabelę dla zbioru danych Car evaluation (pozostałe można obejrzeć pobierając kod i uruchamiając skrypt metrics_comparison.py). Wszystkie poniższe wyniki zostały uzyskane przy wartości $\text{min_support} = 0.15$ oraz $\text{min_confidence} = 0.3$.

5.3.1 Car Evaluation

Lift Range	Rule	Relative Support	Lift	Certainty	Jaccard	Odds Ratio
0.6-0.7	('safety_high',) ==> ('target_unacc',)	0.16	0.69	1.44	0.87	2.06
0.7-0.8	('persons_4',) ==> ('target_unacc',)	0.18	0.77	1.63	1.10	2.32
0.7-0.8	('persons_more',) ==> ('target_unacc',)	0.19	0.80	1.68	1.18	2.40
0.8-0.9	('safety_med',) ==> ('target_unacc',)	0.21	0.89	1.86	1.50	2.66
0.8-0.9	('maint_med',) ==> ('target_unacc',)	0.16	0.89	2.48	1.88	3.54
0.8-0.9	('maint_low',) ==> ('target_unacc',)	0.16	0.89	2.48	1.88	3.54
0.8-0.9	('buying_med',) ==> ('target_unacc',)	0.16	0.89	2.48	1.88	3.54
0.9-1.0	('doors_4',) ==> ('target_unacc',)	0.17	0.97	2.70	2.46	3.86
0.9-1.0	('doors_3',) ==> ('target_unacc',)	0.17	0.99	2.78	2.71	3.97
0.9-1.0	('target_unacc',) ==> ('lug_boot_big',)	0.21	0.91	0.43	0.42	1.30
0.9-1.0	('lug_boot_big',) ==> ('target_unacc',)	0.21	0.91	1.92	1.62	2.74
0.9-1.0	('lug_boot_med',) ==> ('target_unacc',)	0.23	0.97	2.04	1.93	2.92
0.9-1.0	('target_unacc',) ==> ('lug_boot_med',)	0.23	0.97	0.46	0.46	1.39
0.9-1.0	('doors_5more',) ==> ('target_unacc',)	0.17	0.97	2.70	2.46	3.86
1.0-1.1	('buying_high',) ==> ('target_unacc',)	0.19	1.07	3.00	3.75	4.28
1.0-1.1	('maint_high',) ==> ('target_unacc',)	0.18	1.04	2.91	3.25	4.15
1.0-1.1	('doors_2',) ==> ('target_unacc',)	0.19	1.08	3.02	3.86	4.31
1.1-1.2	('buying_vhigh',) ==> ('target_unacc',)	0.21	1.19	3.33	7.13	4.76
1.1-1.2	('target_unacc',) ==> ('lug_boot_small',)	0.26	1.12	0.53	0.56	1.59
1.1-1.2	('lug_boot_small',) ==> ('target_unacc',)	0.26	1.12	2.34	3.10	3.35
1.1-1.2	('maint_vhigh',) ==> ('target_unacc',)	0.21	1.19	3.33	7.13	4.76
1.4-1.5	('target_unacc',) ==> ('persons_2',)	0.33	1.43	0.68	0.85	2.04
1.4-1.5	('persons_2',) ==> ('target_unacc',)	0.33	1.43	3.00	29.79	4.28
1.4-1.5	('safety_low',) ==> ('target_unacc',)	0.33	1.43	3.00	29.79	4.28
1.4-1.5	('target_unacc',) ==> ('safety_low',)	0.33	1.43	0.68	0.85	2.04

Rysunek 3: Pełna tabela wyników dla zbioru danych Car Evaluation

Lift Range	No. Rules	Relative Support [Avg. / Std. dev.]	Certainty [Avg. / Std. dev.]	Jaccard [Avg. / Std. dev.]	Odds Ratio [Avg. / Std. dev.]
0.6-0.7	1	[0.160 / 0.000]	[1.443 / 0.000]	[0.870 / 0.000]	[2.060 / 0.000]
0.7-0.8	2	[0.183 / 0.000]	[1.651 / 0.001]	[1.140 / 0.001]	[2.358 / 0.001]
0.8-0.9	4	[0.168 / 0.000]	[2.326 / 0.073]	[1.785 / 0.027]	[3.322 / 0.148]
0.9-1.0	7	[0.199 / 0.001]	[1.863 / 0.898]	[1.723 / 0.778]	[2.862 / 1.120]
1.0-1.1	3	[0.186 / 0.000]	[2.975 / 0.002]	[3.619 / 0.069]	[4.249 / 0.005]
1.1-1.2	4	[0.234 / 0.001]	[2.385 / 1.309]	[4.479 / 7.824]	[3.615 / 1.696]
1.4-1.5	4	[0.333 / 0.000]	[1.840 / 1.346]	[15.323 / 209.370]	[3.162 / 1.260]

Rysunek 4: Tabela wyników uśrednionych dla zbioru danych Car Evaluation

UWAGA! Wyliczone miary wartości średnich oraz odchyłeń standardowych mogą wydawać się błędne, w odniesieniu do tabeli z pełnymi wynikami, jednak ma to związek z różnymi precyzjami wyświetlania wyników i zostało zweryfikowane.

Na podstawie powyższej tabeli można stwierdzić, że **współczynnik wsparcia relatywnego (relative support)** jest tym większy, im wyższy współczynnik podniesienia lift. Widać też, że wartości w poszczególnych kubekach nie odbiegają znacząco od siebie (minimalne wartości odchylenia standardowego).

W przypadku współczynnika **Certainty**, nie widać dokładnej zależności pomiędzy wartościami wsp. lift, jednak można dostrzec znacznie wyższą wartość odchylenia standardowego.

Współczynnik Jaccarda rośnie wykładniczo w stosunku do wartości współczynnika podniesienia. To samo dotyczy odchylenia standardowego oraz jego wartości w zadanym przedziale.

W przypadku współczynnika **Odds Ratio** również widać tendencję wzrostową w stosunku do wartości wsp. lift, jednak nie jest to silnie widoczne zarówno w stosunku do wartości średniej jak i odchylenia standardowego.

5.3.2 Tic Tac Toe endgame

Lift Range	Rule	Relative Support	Lift	Certainty	Jaccard	Odds Ratio
0.6-0.7	('middle-middle-square_o',) ==> ('target_positive',)	0.15	0.67	1.23	0.76	1.88
0.8-0.9	('middle-middle-square_x',) ==> ('bottom-left-square_x',)	0.18	0.86	0.79	0.78	1.80
0.8-0.9	('bottom-left-square_x',) ==> ('middle-middle-square_x',)	0.18	0.86	0.94	0.82	1.97
0.8-0.9	('middle-middle-square_x',) ==> ('middle-right-square_x',)	0.17	0.89	0.73	0.67	1.85
0.8-0.9	('middle-right-square_x',) ==> ('middle-middle-square_x',)	0.17	0.89	1.07	0.94	2.24
0.8-0.9	('top-right-square_o',) ==> ('target_positive',)	0.20	0.86	1.61	1.29	2.47
0.8-0.9	('target_positive',) ==> ('top-right-square_o',)	0.20	0.86	0.46	0.43	1.32
0.8-0.9	('middle-middle-square_x',) ==> ('top-middle-square_x',)	0.17	0.89	0.73	0.67	1.85
0.8-0.9	('top-middle-square_x',) ==> ('middle-middle-square_x',)	0.17	0.89	1.07	0.94	2.24
0.8-0.9	('bottom-right-square_x',) ==> ('bottom-left-square_x',)	0.17	0.88	0.88	0.78	2.01
0.8-0.9	('bottom-left-square_x',) ==> ('bottom-right-square_x',)	0.17	0.88	0.88	0.78	2.01
0.8-0.9	('top-right-square_x',) ==> ('bottom-right-square_x',)	0.17	0.88	0.88	0.78	2.01
0.8-0.9	('bottom-right-square_x',) ==> ('top-right-square_x',)	0.17	0.88	0.88	0.78	2.01
0.8-0.9	('middle-middle-square_x',) ==> ('bottom-right-square_x',)	0.18	0.86	0.79	0.70	1.80
0.8-0.9	('bottom-right-square_x',) ==> ('middle-middle-square_x',)	0.18	0.86	0.94	0.82	1.97
0.8-0.9	('middle-middle-square_x',) ==> ('top-left-square_x',)	0.18	0.86	0.79	0.70	1.80
0.8-0.9	('top-left-square_x',) ==> ('middle-middle-square_x',)	0.18	0.86	0.94	0.82	1.97
0.8-0.9	('bottom-right-square_o',) ==> ('target_positive',)	0.20	0.86	1.61	1.29	2.47
0.8-0.9	('target_positive',) ==> ('bottom-right-square_o',)	0.20	0.86	0.46	0.43	1.32
0.8-0.9	('bottom-left-square_x',) ==> ('top-left-square_x',)	0.17	0.88	0.88	0.78	2.01
0.8-0.9	('top-left-square_x',) ==> ('bottom-left-square_x',)	0.17	0.88	0.88	0.78	2.01
0.8-0.9	('bottom-left-square_o',) ==> ('target_positive',)	0.20	0.86	1.61	1.29	2.47
0.8-0.9	('target_positive',) ==> ('bottom-left-square_o',)	0.20	0.86	0.46	0.43	1.32
0.8-0.9	('top-right-square_x',) ==> ('top-left-square_x',)	0.17	0.88	0.88	0.78	2.01
0.8-0.9	('top-left-square_x',) ==> ('top-right-square_x',)	0.17	0.88	0.88	0.78	2.01
0.8-0.9	('middle-middle-square_x',) ==> ('top-right-square_x',)	0.18	0.86	0.79	0.70	1.80
0.8-0.9	('top-right-square_x',) ==> ('middle-middle-square_x',)	0.18	0.86	0.94	0.82	1.97
0.8-0.9	('middle-middle-square_x',) ==> ('middle-left-square_x',)	0.17	0.89	0.73	0.67	1.85
0.8-0.9	('middle-left-square_x',) ==> ('middle-middle-square_x',)	0.17	0.89	1.07	0.94	2.24
0.8-0.9	('middle-middle-square_x',) ==> ('bottom-middle-square_x',)	0.17	0.89	0.73	0.67	1.85
0.8-0.9	('bottom-middle-square_x',) ==> ('middle-middle-square_x',)	0.17	0.89	1.07	0.94	2.24
0.8-0.9	('top-left-square_o',) ==> ('target_positive',)	0.20	0.86	1.61	1.29	2.47
0.8-0.9	('target_positive',) ==> ('top-left-square_o',)	0.20	0.86	0.46	0.43	1.32
0.9-1.0	('middle-right-square_x',) ==> ('target_positive',)	0.23	0.91	1.51	1.31	2.31
0.9-1.0	('target_positive',) ==> ('middle-right-square_x',)	0.23	0.91	0.55	0.52	1.39
0.9-1.0	('top-right-square_x',) ==> ('bottom-left-square_x',)	0.18	0.95	0.95	0.91	2.19

Rysunek 5: Fragment tabeli wyników dla zbioru danych Tic Tac Toe endgame

Lift Range	No. Rules	Relative Support [Avg. / Std. dev.]	Certainty [Avg. / Std. dev.]	Jaccard [Avg. / Std. dev.]	Odds Ratio [Avg. / Std. dev.]
0.6-0.7	1	[0.154 / 0.000]	[1.227 / 0.000]	[0.760 / 0.000]	[1.877 / 0.000]
0.8-0.9	32	[0.178 / 0.000]	[0.920 / 0.096]	[0.800 / 0.053]	[1.960 / 0.099]
0.9-1.0	36	[0.181 / 0.001]	[0.976 / 0.060]	[0.920 / 0.044]	[2.200 / 0.098]
1.0-1.1	56	[0.192 / 0.003]	[1.234 / 0.317]	[1.334 / 0.445]	[2.658 / 0.392]
1.1-1.2	56	[0.173 / 0.000]	[1.164 / 0.069]	[1.360 / 0.120]	[2.927 / 0.127]
1.2-1.3	46	[0.183 / 0.002]	[1.578 / 0.958]	[-15.597 / 3176.050]	[3.688 / 1.279]
1.3-1.4	16	[0.185 / 0.000]	[1.349 / 0.026]	[2.072 / 0.141]	[3.612 / 0.047]
1.6-1.7	2	[0.200 / 0.000]	[1.630 / 0.002]	[4.413 / 0.080]	[4.647 / 0.003]

Rysunek 6: Tabela wyników uśrednionych dla zbioru danych Tic Tac Toe endgame

Na podstawie powyższej tabeli można stwierdzić, że **współczynnik wsparcia relatywnego (relative support)** jest również tym większy, im wyższy współczynnik podniesienia lift. Ponownie widać również, że wartości w poszczególnych kolumnach nie odbiegają znacząco od siebie (minimalne wartości odchylenia standardowego).

W przypadku współczynnika **Certainty**, podobnie jak w przypadku poprzedniego zbioru danych - nie widać dokładnej zależności w stosunku do wartości wsp. lift.

Współczynnik Jaccarda ponownie zdaje się rosnąć wykładniczo w stosunku do wsp. lift.

5.3.3 Nursery

Lift Range	Rule	Relative Support	Lift	Certainty	Jaccard	Odds Ratio
0.9-1.0	('health_priority',) ==> ('finance_inconv',)	0.17	1.00	1.50	1.50	3.00
0.9-1.0	('finance_inconv',) ==> ('health_priority',)	0.17	1.00	0.67	0.67	2.00
0.9-1.0	('housing_less_conv',) ==> ('finance_inconv',)	0.17	1.00	1.50	1.50	3.00
0.9-1.0	('finance_inconv',) ==> ('housing_less_conv',)	0.17	1.00	0.67	0.67	2.00
0.9-1.0	('parents_pretentious',) ==> ('finance_inconv',)	0.17	1.00	1.50	1.50	3.00
0.9-1.0	('finance_inconv',) ==> ('parents_pretentious',)	0.17	1.00	0.67	0.67	2.00
0.9-1.0	('parents_great_pret',) ==> ('finance_inconv',)	0.17	1.00	1.50	1.50	3.00
0.9-1.0	('finance_inconv',) ==> ('parents_great_pret',)	0.17	1.00	0.67	0.67	2.00
0.9-1.0	('finance_convenient',) ==> ('health_priority',)	0.17	1.00	0.67	0.67	2.00
0.9-1.0	('health_priority',) ==> ('finance_convenient',)	0.17	1.00	1.50	1.50	3.00
0.9-1.0	('health_recommended',) ==> ('finance_inconv',)	0.17	1.00	1.50	1.50	3.00
0.9-1.0	('finance_inconv',) ==> ('health_recommended',)	0.17	1.00	0.67	0.67	2.00
0.9-1.0	('finance_inconv',) ==> ('housing_convenient',)	0.17	1.00	0.67	0.67	2.00
0.9-1.0	('housing_convenient',) ==> ('finance_inconv',)	0.17	1.00	1.50	1.50	3.00
0.9-1.0	('finance_convenient',) ==> ('health_not_recom',)	0.17	1.00	0.67	0.67	2.00
0.9-1.0	('health_not_recom',) ==> ('finance_convenient',)	0.17	1.00	1.50	1.50	3.00
0.9-1.0	('target_priority',) ==> ('finance_inconv',)	0.16	0.95	1.44	1.33	2.88
0.9-1.0	('finance_inconv',) ==> ('target_priority',)	0.16	0.95	0.62	0.60	1.90
0.9-1.0	('target_not_recom',) ==> ('finance_inconv',)	0.17	1.00	1.50	1.50	3.00
0.9-1.0	('finance_inconv',) ==> ('target_not_recom',)	0.17	1.00	0.67	0.67	2.00
0.9-1.0	('finance_convenient',) ==> ('social_problematic',)	0.17	1.00	0.67	0.67	2.00
0.9-1.0	('social_problematic',) ==> ('finance_convenient',)	0.17	1.00	1.50	1.50	3.00
0.9-1.0	('finance_convenient',) ==> ('housing_critical',)	0.17	1.00	0.67	0.67	2.00
0.9-1.0	('housing_critical',) ==> ('finance_convenient',)	0.17	1.00	1.50	1.50	3.00
0.9-1.0	('finance_convenient',) ==> ('housing_convenient',)	0.17	1.00	0.67	0.67	2.00
0.9-1.0	('housing_convenient',) ==> ('finance_convenient',)	0.17	1.00	1.50	1.50	3.00
0.9-1.0	('finance_convenient',) ==> ('health_recommended',)	0.17	1.00	0.67	0.67	2.00
0.9-1.0	('health_recommended',) ==> ('finance_convenient',)	0.17	1.00	1.50	1.50	3.00
0.9-1.0	('finance_inconv',) ==> ('health_not_recom',)	0.17	1.00	0.67	0.67	2.00
0.9-1.0	('health_not_recom',) ==> ('finance_inconv',)	0.17	1.00	1.50	1.50	3.00
0.9-1.0	('finance_inconv',) ==> ('social_nonprob',)	0.17	1.00	0.67	0.67	2.00
0.9-1.0	('social_nonprob',) ==> ('finance_inconv',)	0.17	1.00	1.50	1.50	3.00
0.9-1.0	('finance_convenient',) ==> ('target_not_recom',)	0.17	1.00	0.67	0.67	2.00
0.9-1.0	('target_not_recom',) ==> ('finance_convenient',)	0.17	1.00	1.50	1.50	3.00
0.9-1.0	('social_problematic',) ==> ('finance_inconv',)	0.17	1.00	1.50	1.50	3.00
0.9-1.0	('finance_inconv',) ==> ('social_problematic',)	0.17	1.00	0.67	0.67	2.00
0.9-1.0	('housing_critical',) ==> ('finance_inconv',)	0.17	1.00	1.50	1.50	3.00
0.9-1.0	('finance_inconv',) ==> ('housing_critical',)	0.17	1.00	0.67	0.67	2.00

Rysunek 7: Fragment tabeli wyników dla zbioru danych Nursery

Lift Range	No. Rules	Relative Support [Avg. / Std. dev.]	Certainty [Avg. / Std. dev.]	Jaccard [Avg. / Std. dev.]	Odds Ratio [Avg. / Std. dev.]
0.9-1.0	58	[0.166 / 0.000]	[1.082 / 0.173]	[1.079 / 0.173]	[2.496 / 0.250]
1.0-1.1	4	[0.171 / 0.000]	[1.175 / 0.243]	[1.290 / 0.340]	[2.733 / 0.368]
1.4-1.5	2	[0.156 / 0.000]	[1.503 / 0.010]	[3.040 / 0.160]	[4.654 / 0.024]
1.6-1.7	2	[0.186 / 0.000]	[1.696 / 0.000]	[5.591 / 0.054]	[5.121 / 0.001]
1.8-1.9	2	[0.190 / 0.000]	[1.833 / 0.015]	[12.399 / 22.473]	[5.675 / 0.035]
2.9-3.0	10	[0.200 / 0.004]	[3.600 / 4.140]	[-1.400 / 1.440]	[12.600 / 19.440]

Rysunek 8: Tabela wyników uśrednionych dla zbioru danych Nursery

Na podstawie powyższej tabeli można ponownie stwierdzić, że **współczynnik wsparcia relatywnego (relative support)** jest tym większy, im wyższy współczynnik podniesienia lift. Lekkie wahania można zignorować, przez niewielką ilość reguł asocjacyjnych w środkowych zakresach wartości wsp. lift.

W przypadku współczynnika **Certainty**, podobnie jak w przypadku dwóch poprzednich zbiorów, nie widać dokładnej zależności pomiędzy wartościami wsp. lift. Można jednak dostrzec znacznie wyższą wartość odchylenia standardowego.

Współczynnik Jaccarda ponownie zdaje się rosnać wykładniczo w stosunku do wsp. lift. Nie jest to tak dobrze widoczne jak w przypadku poprzednich zbiorów danych przez niezbalansowany rozkład reguł w poszczególnych przedziałach wartości wsp. lift.

W przypadku współczynnika **Odds Ratio** również widać tendencje wzrostową w stosunku do wartości wsp. lift, która zdaje się być lekko zaburzona przez duże odchylenia wartości w ostatnim przedziale.

6 Wnioski

Na podstawie niniejszego raportu można wysnuć kilka istotnych wniosków:

1. Implementacja algorytmu **Apriori** w ramach projektu osiągnęła pełne pokrycie z wynikami uzyskanymi za pomocą popularnej biblioteki **apyori**, co świadczy o poprawności jej działania.
2. Pomimo pełnego pokrycia, własna implementacja algorytmu **Apriori** działa średnio **10 razy wolniej** niż biblioteka **apyori**. W przyszłości istnieje możliwość optymalizacji kodu w celu poprawy wydajności.
3. Analiza zachowania się wybranych miar w odniesieniu do wartości współczynnika podniesienia (*lift*) wykazała, że większość miar (*współczynnik Jaccarda*, *Certainty Factor*, *Odds Ratio*) wykazuje tendencję do wzrostu wraz ze wzrostem wartości współczynnika podniesienia.
4. Dokładność i wiarygodność wyników uzyskanych przez algorytm **Apriori** zależy od odpowiedniego dobrania parametrów, takich jak *minimalne wsparcie* (*min_support*) i *minimalne zaufanie* (*min_confidence*). Warto przeprowadzić analizę wrażliwości, aby określić optymalne wartości tych parametrów dla konkretnego zestawu danych.
5. Analiza wybranych miar asocjacyjnych może dostarczyć cennych informacji o relacjach między zbiorami elementów w danych. Niemniej jednak, interpretacja wyników wymaga ostrożności i głębszej analizy uzyskanych wartości.
6. W kontekście porównania algorytmów **Apriori** i **Eclat** wykazano, że oba algorytmy generują identyczne reguły asocjacyjne, co potwierdza ich spójność i poprawność działania. Niemniej jednak:
 - Algorytm **Eclat** jest **znacznie szybszy** niż algorytm **Apriori**, szczególnie dla dużych i gęstych zbiorów danych, takich jak *nursery*.
 - Przewaga czasowa algorytmu **Eclat** wynika z wykorzystania pionowej reprezentacji danych (*TID-sets*) oraz bardziej efektywnego przeszukiwania przestrzeni kombinacji poprzez operacje na zbiorach identyfikatorów transakcji.
 - Algorytm **Apriori** natomiast jest bardziej intuicyjny i lepiej nadaje się do analizy mniejszych oraz mniej złożonych zbiorów danych, gdzie interpretowalność wyników jest kluczowa.
7. W praktyce dla dużych zbiorów danych, takich jak *nursery*, rekomenduje się użycie algorytmu **Eclat**, natomiast dla mniejszych zbiorów, takich jak *tic_tac_toe_endgame*, oba algorytmy działają efektywnie i ich wybór może zależeć od innych czynników, takich jak dostępne zasoby obliczeniowe czy potrzeba interpretowalności wyników.

Ogólnie rzecz biorąc, projekt dostarcza solidnej implementacji algorytmów **Apriori** oraz **Eclat** i umożliwia przeprowadzenie analizy skuteczności reguł asocjacyjnych oraz wybranych miar w kontekście różnych zbiorów danych. Wyniki wskazują, że algorytm **Eclat** jest lepszym wyborem dla dużych i gęstych zbiorów danych, natomiast algorytm **Apriori** pozostaje skutecznym narzędziem do analizy mniejszych zestawów. Istnieje również potencjał do dalszego rozwoju projektu w celu optymalizacji wydajności i rozszerzenia funkcjonalności.

7 Literatura

- [1] Roberto J. Bayardo, Rakesh Agrawal, and Dimitrios Gunopulos. Constraint-based rule mining in large, dense databases. In *Proceedings of the 15th International Conference on Data Engineering (ICDE)*, pages 188–197. IEEE, 2000.
- [2] Fernando Berzal, Ignacio Blanco, Daniel Sanchez, and Maria-Amparo Vila. Measuring the accuracy and interest of association rules: A new framework. *Intelligent Data Analysis*, 6(3):221–235, 2002.
- [3] Sergey Brin, Rajeev Motwani, Jeffrey D. Ullman, and Shalom Tsur. Dynamic itemset counting and implication rules for market basket data. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 255–264. ACM, 1997.
- [4] Pang-Ning Tan, Vipin Kumar, and Jaideep Srivastava. Selecting the right objective measure for association analysis. *Information Systems*, 29(4):293–313, 2004.
- [5] Kiyoto Yabe. apyori: A simple implementation of apriori algorithm. <https://github.com/ymoch/apyori>, 2015. Accessed: 2024-05-30.