Міністерство освіти і науки України
Тернопільський національний економічний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

Зубко Віталіна Сергіївна

Методи пошуку асоціативних правил в базі даних біомедичних зображень / Aassociative rules search methods in a biomedical images database

Спеціальність 8.091501 – Комп'ютерні системи та мережі

Дипломна робота за освітньо-кваліфікаційним рівнем «магістр»

Науковий керівник
д.т.н., професор Березький О.М.
_____

Дипломну роботу допущено до захисту
«___»_____ 20 __ р.
Зав. кафедри КІ
Березький О.М. _____

Тернопіль – 2017

Міністерство освіти і науки України
Тернопільський національний економічний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

Освітньо-кваліфікаційний рівень магістр
Спеціальність 8.05010201 «Комп'ютерні системи та мережі»

«Затверджую»
завідувач кафедри
д.т.н., проф. Березький О.М.
_____
"____"_____20___ р.

ЗАВДАННЯ
НА ДИПЛОМНУ РОБОТУ СТУДЕНТА
Зубко Віталіни Сергіївни

1. Тема дипломної роботи "Методи пошуку асоціативних правил в базі даних біомедичних зображень"
затверджена наказом №484 від "19" жовтня 2016 р.

2. Термін здачі закінченої дипломної роботи "13" лютого 2017 р.

3. Об'єкт дослідження. Біомедичні зображення.

4. Предмет дослідження. Дослідження та пошук асоціативних правил в базі даних біомедичних зображень

5. Перелік задач, які мають бути вирішені:
- здійснити аналіз алгоритмів та методів побудови асоціативних правил в середовища WEKA;
- провести аналіз алгоритмів пошуку асоціативних правил;
- провести експериментальні дослідження по одержанню кількісних ознак мікрооб'єктів на цитологічних та гістологічних зображенях;
- провести класифікацію мікрообєків різними алгоритмами;
- розробити узагальнений алгоритм пошуку асоцітивних правил;
- провести побудову та пошук асоціативних правил різними алгоритмами;
- провести аналіз результатів експерементальних досліджень.

6. Перелік ілюстративного матеріалу:
- тема, мета, завдання, методи досліджень, наукова новизна, практичне значення;
- актуальність;
- об'єкт дослідження;

- предмет дослідження;
- методи дослідження;
- аналіз біомедичних зображень;
- стадії інтелектуального аналізу даних;
- алгоритми інтелектуального аналізу даних;
- основні вікна в середовищі дослідження;
- порівняльні таблиці результатів експериментальних досліджень;
- приклад застосування алгоритмів асоції на експериментальній базі
- приклад асоціативних правил до різних застосованих алгоритмів

7. Консультанти по роботі

Розділ	Консультант	Перевірив
Нормо-контроль	Мельник Г.М.	

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Відмітка про виконання
1	Аналіз бази даних біомедичних зображень та методів інтелектувального аналізу даних	20.10.2015 – 01.01.2016	
2	Алгоритми пошуку асоціативних правил	02.01.2016 – 31.05.2016	
3	Програмна реалізація та тестування алгоритмів пошуку асоціативних правил	01.06.2016 – 19.12.2016	
4	Нормоконтроль, попердній захист	20.12.2016 – 21.12.2016	
5	Захист	13.02.2016	

Завдання прийняв до виконання _____ Зубко В.С.
(підпис) (прізвище та ініціали)

Керівник дипломної роботи _____ Березький О.М.
(підпис) (прізвище та ініціали)

РЕЗЮМЕ

Дипломна робота на тему “Методи пошуку асоціативних правил в базі даних біомедичних зображень” на здобуття освітньо-кваліфікаційного рівня “Магістр” зі спеціальності “Комп’ютерні системи та мережі” написана обсягом 100 сторінок і містить 15 ілюстрації, 7 таблиць, 4 додатки та 65 джерел за переліком посилань.

Метою роботи є пошук асоціативних правил в базі даних цитологічних та гістологічних зображень диспластичних і ракових процесів молочної залози, використовуючи різні алгоритми інтелектуального аналізу даних.

Методи досліджень. Для розв’язання поставлених задач у дипломній роботі було: досліджено та використано методи інтелектуального аналізу даних, алгоритми для здійснення класифікації мікрооб’єктів та побудови моделі, алгоритми для пошуку асоціативних правил в середовиші WEKA.

Результати дослідження: розроблено узагальнений алгоритм пошуку асоціативних правил; проведено порівняльний аналіз по критеріях і показниках даного експерименту; отримано асоціативні правила для діагностики передракових та ракових станів раку молочної залози та відповідні їм лінгвістичні змінні та проведено аналіз отриманих результатів.

Результати роботи можуть бути використані в науковій медицині, впровадження результатів роботи здійснено в Тернопільському обласному онкологічному диспансері.

Орієнтовні напрямки розвитку досліджень: детальний аналіз біомедичних зображень, побудова більших моделей та проведенння класифікації мікрооб’єктів на експериментальній базі цитологічних та гістологічних зображень диспластичних і ракових процесів молочної залози, розробка власного спрощеного алгоритму для пошуку та побудові моделей.

КЛЮЧОВІ СЛОВА: БІОМЕДИЧНЕ ЗОБРАЖЕННЯ, ГІСТОЛОГІЯ, ЦИТОЛОГІЯ, ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ, АСОЦІАТИВНЕ ПРАВИЛО

RESUME

Diploma work: " Methods search of associative rules in the database of biomedical images " to education and qualification of "Master" specialty "Computer systems and networks" written 100 page volume and contains 15 illustrations, 7 tables, 4 applications and 65 sources for references.

The aim is to search associative rules in the database cytological and histological images dysplastic and cancerous breast processes using different data mining algorithms.

Research Methods. To solve the tasks of the thesis work was, studied and used methods of data mining algorithms to perform microscopic grading and construction of models, algorithms for searching associative rules in seredovyshi WEKA.

Results: The generalized search algorithm of associative rules built model experiments; The comparative analysis of criteria and indicators of the experiment; association rules received for the diagnosis of precancerous and cancerous conditions of breast cancer and the corresponding linguistic variables and analyzed the results.

The results can be used in scientific medicine, implementation of the work carried out in the Ternopil regional oncologic dispensary.

The estimated directions of research: a detailed analysis of biomedical images, build models and the classification of microscopic objects on an experimental basis cytological and histological images of dysplastic and cancer of the breast process, developing its own simplified algorithm for searching and building models.

KEY WORDS: BIOMEDICAL IMAGE, HISTOLOGY, CYTOLOGY, DATA MINING, ASSOCIATION RULES.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

БД	–	База даних
САМ	–	Система автоматизованої мікроскопії
ПЗ	–	Програмне забезпечення
ARFF	–	Attribute-Relation File Format
API	–	Application Programming Interface
CSV	–	Comma-Separated Values
ETL	–	Extract Transform and Load
JDBC	–	Java DataBase Connectivity
GUI	–	Graphical User Interface
GPL	–	General Public License
OLAP	–	OnLine Analytical Processing
SVM	–	Support Vector Machines
WEKA	–	Waikato Environment for Knowledge Analysis

ВСТУП

Актуальність теми. Рак грудної залози посідає перше місце в структурі онкозахворюваності жінок. Кожна десята жінка в Україні страждає даним недугом, а згідно прогнозу канцерреєстру незабаром буде хворіти кожна девята. Так, щогодини помирає одна жінка від раку грудної залози, а кожні 30 хвилин фіксується новий випадок. Тому, проблеми ранньої діагностики, лікування та профілактики набувають особливої актуальності.

Рак молочної залози зустрічається, у порівнянні з пухлинами інших локалізацій, досить часто. У світі щорічно реєструється близько одного мільйона нових випадків захворювання на рак молочної залози. В країнах СНД ця патологія посідає перше місце у структурі смертності від онкологічних захворювань у жінок працездатного віку. Особливу актуальність проблема раку молочної залози має в промислово розвинених регіонах України.

На жаль, впровадження сучасних методів діагностики та синтез нових препаратів суттєво не вплинули на кількість пацієнток з уперше виявленими задавненими формами на цю патологію. На сьогодні смертність від злоякісних пухлин молочної залози в світі складає 17,9 на 100 тисяч жіночого населення, а в Україні - 27,0 на 100 тисяч жіночого населення.

Перед онкологом постають задачі не тільки первинної й уточнюючої діагностики та лікування раку молочної залози, але й оцінки ефективності різних методів лікування цієї патології, своєчасного виявлення рецидивів після проведеного лікування.

Сьогодні опрацювання біомедичних зображень є важливим напрямком застосування сучасної медичної техніки. Задачами опрацювання зображень є опис, аналіз та оброблення зображень. Проблеми аналізу біомедичних зображень включаючи класичну задачу розпізнавання фігур заданої форми, важлива також експертна оцінка, яка зараз є дорогою. Виникають проблеми, які зумовлені новими завдання опису зображення та пошуком закономірностей або

наборів закономірностей, що одночасно зустрічаються в багатьох наборах. Оскільки наборів може бути велика кількість необхідно здійснювати цей пошук автоматично. Тому актуальною задачею є розробка алгоритму для пошуку асоціативних правил бази даних цитологічних та гістологічних зображень, що містять кількісні і якісні ознаки мікрооб'єктів.

Мета і завдання дослідження. Метою дослідження є пошук асоціативних правил в базі даних цитологічних та гістологічних зображень диспластичних і ракових процесів молочної залози, використовуючи різні алгоритми інтелектуального аналізу даних.

Об'єкт дослідження. Біомедичні зображення раку молочної залози.

Предмет дослідження. Алгоритми пошуку асоціативних правил.

Методи досліджень. Базуються на використанні методів інтелектуального аналізу даних; алгоритмів класифікації (для класифікації мікроб'єктів на зображеннях та побудові моделей); алгоритмів асоціації (для пошуку асоціативних правил);

Наукова новизна одержаних результатів. 1.Розроблено узагальнений алгоритм для побудови асоціативних правил. 2.Вперше застосовано алгоритми відповідно до наявної тестової бази біомедичних зображень та отримано асоціативні правила**.**

Практичне значення отриманих результатів. Розроблено узагальнений алгоритм для пошуку асоціативних правил для діагностики передракових та ракових станів раку молочної залози.

Публікації та апробація ДР. Основні результати дослідження «Методи пошуку асоціативних правил в базі даних біомедичних зображень» опубліковано на конференції АСІІТ 2016 [3].

Впровадження результатів ДР. Впровадження результатів дипломної роботи здійснено при виконанні держбюджетної теми «КІ-05-2016». «Гібридна інтелектуальна інформаційна технологія діагностування передракових станів молочної залози на оснві аналізу зображень».

У першому розділі буде проведено аналіз біомедичних зображень, описано основні характеритистикаи гістологічних та цитологічних зображень. Буде проведено аналіз методів та алгоритмів інтелектуального аналіз даних, та класифікацію стадій антелектуального аналізу. Також буде проведено аналіз програмних засобів інтелектуального аналізу даних, наведено основні характеристики та практичне використання до певного кола задач.

У другому розділі проведено аналіз усіх існуючих алгоритмів для пошуку асоціативних правил, зокрема, алгоритмів які вбудовані в програмний засіб для побудови моделей та пошуку асоціативних правил.

У третьому розділі буде описано програмно-апартний засіб, тобто робочу станцію-ноутбук та програмне середовище для проведення експерементальних досліджень використовючи наявну базу біомедичних зображень. Також буде розроблено узагальнений алгоритм для пошуку асоціативних правил, який включає в себе кілька основних кроків. Всі кроки будуть здійснені в середовиші WEKA, за початкові дані для опрацювання буде обрана наявна експерементальна база біомедичних зображень. Також буде проведено порівняльну роботу всіх алгоритмів та проаналізовано результати експерементального дослідження. В результаті буде отримано набір асоціативних правил.

В додатках буде представлена модель дерева рішень для класифікованих мікрооб’єктів, вихідний текст файлу для обробки у WEKA, світлокопії виданої публікації та довідка про впровадження результатів дипломної роботи.

1 КЛАСИ БІОМЕДИЧНИХ ЗОБРАЖЕНЬ, МЕТОДИ, АЛГОРИТМИ ТА ПРОГРАМНІ ЗАСОБИ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ

1.1Аналіз біомедичних зображень

Робота з графічною інформацією традиційно є одним з найважливіших напрямів застосування комп'ютера в медицині, який розглядається у спеціальному підрозділі медичної інформатики, що отримав назву аналіз біомедичних зображень.

Біомедичне зображення є одним з важливих засобів отримання візуальної інформації про внутрішні структури й функції людського тіла. Воно може бути отримане радіологічними або не радіологічними методами [1].

Призначення радіологічних методів - зробити доступним для візуального сприйняття інформацію, що не сприймається безпосередньо зором. Така інформація (зображення органів або частин органів) отримується за допомогою випромінювання. Це випромінювання має, як правило, електромагнітну природу. Біомедичні зображення органів отримані засобами радіологічної діагностики є головним джерелом інформації в галузі охорони здоров'я. Всі ці методи для отримання зображень використовують обчислювальні процедури.

Не радіологічними методами отримують зображення, що відзняті відеокамерою (ендоскопія) або сфотографовані (мікроскопічні зображення в гістології, патології, дерматологічні зображення тощо). Ці типи зображень також можуть бути переведені в цифрову форму й згодом оброблені.

Надалі будемо розглядати переважно біомедичні зображення, отримані не радіологічними методами. Саме тому під поняттям «біомедичне зображення» розумітимемо (доступну зоровому сприйняттю) картину просторового розподілу будь-якого виду випромінювання, трансформованого у видиму частину оптичного діапазону.

Після утворення зображення воно має бути інтерпретовано. Засіб інтерпретації та показу може бути носієм вихідного зображення, наприклад,

відеофільм, з якого було сформовано зображення, або інший носій – фотографія чи монітор комп’ютера [2].

Аналіз біомедичних зображень має важливе значення в сучасній медицині. З постійно зростаючою кількістю даних пацієнта, нові виклики та можливості виникають в різних фазах клінічної практики, наприклад, діагностики, лікування та моніторингу.

Об'єкти на медичних зображеннях володіють великою складністю і багатофакторністю, що зумовлює високі вимоги до надійності, точності та достовірності результатів досліджень. Використання обчислювальної техніки та математичних методів у цій галузі дозволяє не тільки прискорити процес обробки матеріалу, але і підвищити точність результатів дослідження [4].

Автоматизація аналізу гістологічних структур прискорює діагностику захворювання, дозволяє розширити межі наукових пошуків у медицині. Автоматичне вимірювання параметрів гістологічних об'єктів дає можливість уточнити лікування і управління терапевтичними процесами. Так, найбільш перспективним методом ранньої діагностики пухлинних захворювань в даний час є автоматизація цітофотометріческого аналізу спеціально приготовлених і забарвлених гістологічних препаратів і поділ їх за принципом норма - патологія.

Однією з головних частин автоматизації вимірювання оптичних і геометричних параметрів є виділення об'єктів на гістологічних препаратах. Це завдання вирішується за допомогою методів і засобів цифрового аналізу зображень.

Основною причиною відсутності автоматизації в гістології є висока варіабельність і слабка контрастність більшості гістологічних структур.

Біомедичні зображення використовуються під час встановлення діагнозу та подальшого лікування [5].

В сучасній медицині відомі такі біомедичні зображення: зображення цифрової мікроскопії (цитологічних, гістологічних зразків), рентгенограми, ультразвукові зображення, зображення магнітно-резонансної томографії та ін.
В даній роботі буде досліджено зображення цитологічних та гістологічних зразків.

Гістологія - наука про будову, розвиток та життєдіяльність тканин людини. Гістологія вивчає не тільки тканини, але й клітини, з яких вони складаються, будову органів і систем організму. Згідно з цим існують такі розділи предмету, як цитологія (наука про клітину), загальна гістологія, або власне гістологія (вивчає тканини), спеціальна гістологія (вивчає будову органів і їх систем) [6].

Предмет гістології — тонка (мікроскопічна) та ультратонка (субмікроскопічна) будова структур організму, яка вивчається методом мікроскопії. Тонкими або мікроскопічними структурами називають такі, що не видні неозброєним оком, а лише під світловим мікроскопом. Ультратонкі або субмікроскопічні структури можна побачити під електронним мікроскопом. Мікроскопічний метод відрізняє гістологію від анатомії, яка теж вивчає будову організму, але на рівні того, що можна бачити неозброєним оком. Ці дві науки, а також топографічну та патологічну анатомії називають морфологічними Отож, можна сказати, що гістологія – це наука про мікроскопічну будову, розвиток і життєдіяльність структур організму.

Гістологія - це окрема наука, яка вивчає будову, функціонування, особливості живих клітин. Ще таку галузь медицини називають клітинною біологією. Клінічна цитологія – це галузь лабораторних досліджень, суть яких полягає в мікроскопічному описовому аналізі цитологічного матеріалу. З допомогою аналізу на цитологію визначають онко захворювання, передракові стани та доброякісні новоутворення, а також запальні процеси. При цьому можна побачити особливості будови різних клітин, а також оцінити клітинний склад яких тканин або біологічних рідин людини. Так можна виявити патологічні зміни, які характерні для певних захворювань, і поставити діагноз.

Цитологія дозволяє виявити деякі фізіологічні зміни в організмі, що виникають. Цитологічне дослідження має схожість з гістологічним аналізом, при якому вивчають шматочки тканин. Але для першого методу діагностики потрібно набагато менше біологічного матеріалу (потрібні тільки окремі клітини або їхні комплекси). За допомогою цитологічного аналізу можна оцінити зміни тільки на обмеженій ділянці тканини або органу. При цьому неможливо вивчити взаєморозташування клітин, як при гістологічному дослідженні [7].

Цитологічний аналіз проводять у тих ситуаціях, коли немає можливості виконати біопсію (прижиттєве вилучення шматочків тканин або органів), якщо необхідно ретельно вивчити будову клітин або швидко одержати результати. Зазвичай цитологічний аналіз проводиться:

- при масових профілактичних оглядах;
- для встановлення або уточнення діагнозу при якому-небудь захворюванні;
- для встановлення або уточнення діагнозу під час оперативного втручання;
- для контролю над ефективністю лікування, як під час його проведення, так і після його завершення;
- для своєчасного виявлення рецидивів (відновлення) будь-яких хвороб.

Об’єктами дослідження сучасної гістології і цитології є нормальні і патологічні клітини і тканини, та їх зображення, отримані в світлових та електронних мікроскопах. Задачу автоматизації аналізу мікроскопічних зображень можуть розв’язати системи автоматизованої мікроскопії (САМ). САМ є програмно-апаратним комплексом для цифрової обробки мікроскопічних зображень. Автоматизований аналіз є об’єктивнішим і дає можливість отримувати не лише якісні, але і кількісні оцінки структурних змін елементів тканини. Одним із важливих етапів автоматизації вимірювань оптичних і геометричних параметрів є виділення об’єктів на гістологічних препаратах. Основними труднощами при аналізі біомедичних зображень є

висока варіабельність параметрів і слабкий контраст більшості об’єктів. Об’єктами на гістологічних зображеннях є зрізи тканин певних органів. Тканина складається із клітин округлої форми розміщених шарами ,розміри яких становлять декілька мікрометрів; найменші з них – від 0,5 до 1,2 мкм. Об’єктами на цитологічних зображеннях є окремі клітини розміщені випадковим чином. Аналіз гістологічного зображення, що виконується засобами САМ, можна розділити на наступні етапи отримання зображення, ручне і автоматичне виділення об’єктів (клітин, ядер, ділянок різного забарвлення або яскравості і т.п.), вимірювання розмірів, форми, положення, оптичних параметрів виділених об’єктів або їх частин, класифікація об’єктів і статистична обробка результатів вимірювань.

Гістологічні зображення володіють наступними особливостями:

- слабкий контраст, що зумовлений використанням камер малої роздільної здатності;
- містять об’єкти, оточені складним за геометричними і оптичними характеристиками фоном;
- нерівномірність фону зумовлена неправильним налаштуванням модуля освітлення мікроскопа при створенні зображення;
- перепади рівнів яскравості об’єктів такі ж, як і у фону, що їх оточує;
- залежно від міри оптичного збільшення зображень одні об’єкти виділяються краще, а інші втрачаються;
- містять області з повторюваною структурою;
- стабільність кольорової палітри для зображень зразків, виготовлених при застосуванні відомих фарбників [8].

Використання спеціалізованої системи координат опису кольору дає можливість поліпшити якість морфологічних операцій і збільшити швидкість в порівнянні з обробкою в традиційних системах координат.

Сегментація зображення призводить до поділу зображення на області із подібними характеристиками. Одні з основних ознак для проведення сегментації – це яскравість для монохромного зображення та кольорова

компонента для кольорового зображення. Також, для процесу сегментації використовуються границі зображення та текстура. Процес сегментації тільки розділяє зображення, а не визначає індивідуальні сегменти та їх взаємозв'язок.

На даний час цифрові зображення були використані для розробки декількох інтернет - атласів цитології, в тому числі відомі NCI Bethesda System Web Atlas. Цей веб - атлас складається з 349 зображень. Отже, в телемедицині практикується використання цифрових зображень. При цьому аналітичний метод здійснюється в одному місці, а необхідні елементи, передаються в електронному вигляді на інший сайт для діагностичної інтерпретації.

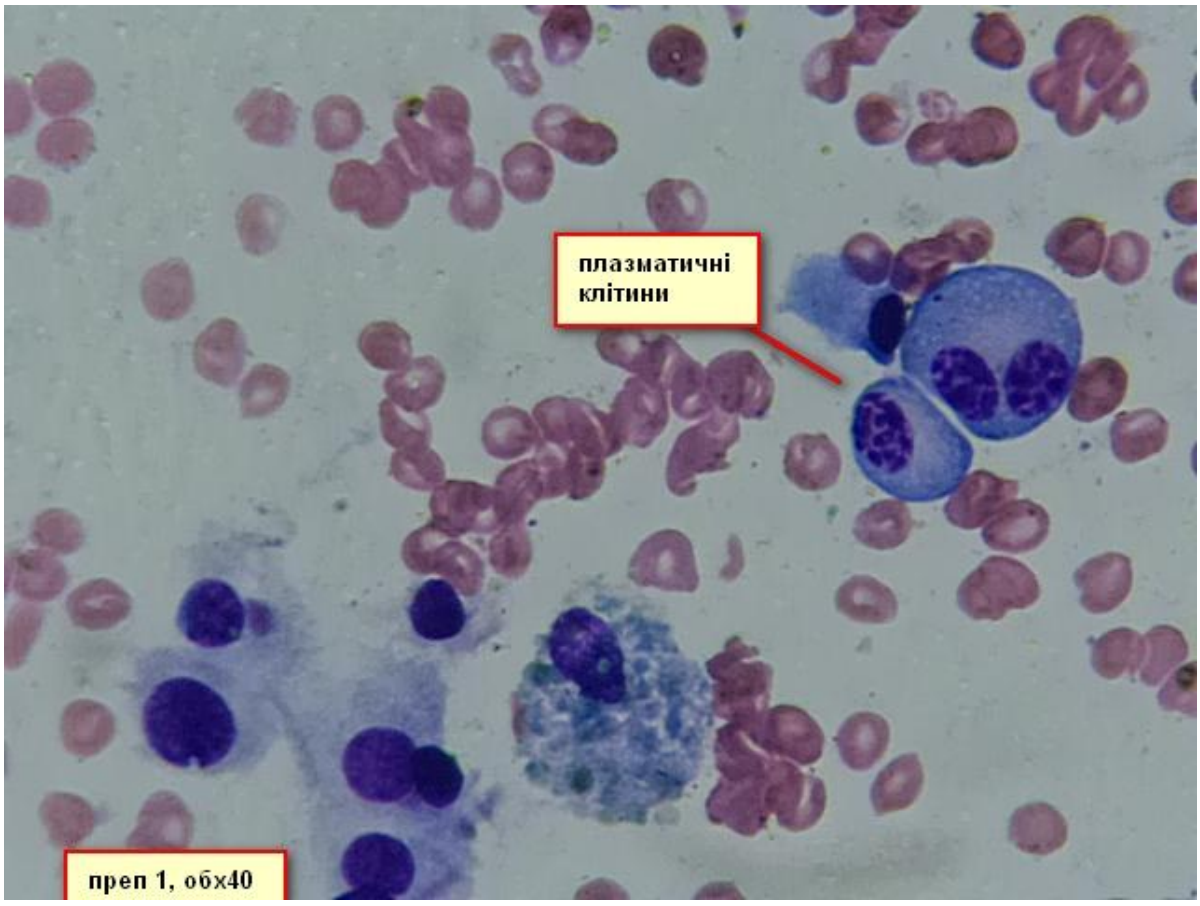

Рисунок 1.1 – Приклад цитологічного зображення

Перевага цифрових зображень полягає в тому, що потенційно усуває необхідність в скельцях (принаймні на момент огляду), а також можливість швидко передавати і віддалено обмінюватися зображеннями в електронному

вигляді для різних цілей (телецитології, конференцій, освіти, забезпечення якості, експертної оцінки).

Візуалізація процесу включає в себе захоплення, збереження, редагування (при необхідності), а також обмін (перегляд, відображення, роздрук) цифрових зображень. Цифрові зображення мають бути якісними [9].

Висока якість слайдів забезпечується при допомозі якісного мікроскопа і цифрової фотокамери. Фотографуються інформативні поля зору з декількох скелець. При цьому, кожне поле зору фотографується з об'єктивом 20 і 40. Проводиться не менше як 15-20 фотознімків з пункції одного вузла. Далі, за допомогою програм слайди редагуються. Якісні фотознімки відбираються (не менше ніж 10 для кожного вузла), маркіруються і з направленням зберігають в базу даних.

Використання телецитології дає можливість пацієнту отримувати цитологічне заключення і визначити кінцевий діагноз.

1.2 Аналіз методів і алгоритмів інтелектуального аналізу даних

Інтелектуальний аналіз даних являє собою відносно нове поле досліджень якого головна мета полягає в придбанні знання від великих обсягів даних.

З одного боку, практикуючі будуть використовувати всі ці дані у своїй роботі, але, в той же час, така велика кількість даних не може бути оброблений людьми протягом короткого часу, щоб зробити діагноз, прогноз і лікування. Основна мета цієї тези є оцінка інтелектуального аналізу даних інструментів в медичних та охорони здоров'я додатків розробити інструмент, який може допомогти зробити своєчасні та точні рішення. Два медичних баз даних, вважаються, по одному для опису різних інструментів та іншої, як у випадку дослідження [10].

Основна особливість Data Mining - це поєднання широкого математичного інструментарію (від класичного статистичного аналізу до нових кібернетичних методів) і останніх досягнень у сфері інформаційних технологій. В технології Data Mining гармонійно об'єдналися строго формалізовані методи і методи неформального аналізу, тобто кількісний і якісний аналіз даних [16].

До методів і алгоритмів Data Mining відносяться наступні: штучні нейроні мережі, дерева рішень, символьні правила, методи найближчого сусіда і к-найближчого сусіда, метод опорних векторів, байсові мережі, лінійна регресія, кореляційно-регресійний аналіз; ієрархічні методи кластерного аналізу, неієрархічні методи кластерного аналізу, у тому числі алгоритми k-середніх і к-медіани; методи пошуку асоціативних правил, у тому числі алгоритм Apriori; метод обмеженого перебору, еволюційне програмування і генетичні алгоритми, різноманітні методи візуалізації даних і безліч інших методів.

Більшість аналітичних методів, що використовуються в технології Data Mining - це відомі математичні алгоритми і методи. Новою в їх застосуванні є можливість їх використовування при рішенні тих або інших конкретних проблем, обумовлена новими можливостями технічних і програмних засобів, що з'явилися. Слід зазначити, що більшість методів Data Mining була розроблена в рамках теорії штучного інтелекту. Єдиної думки щодо того, які задачі слід відносити до Data Mining, немає. Більшість авторитетних джерел перераховує наступні: класифікація, кластеризація, прогнозування, асоціація, візуалізація, аналіз і виявлення відхилень, оцінювання, аналіз зв'язків, підведення підсумків. Розглянемо деякі з них [11].

Класифікація (Classification). Це найпростіша і поширена задача Data Mining. В результаті рішення задачі класифікації виявляються ознаки, які характеризують групи об'єктів досліджуваного набору даних - класи; по цих ознаках новий об'єкт можна віднести до того або іншого класу. Для вирішення задачі класифікації можуть використовуватися методи: найближчого сусіда (Nearest Neighbor); к-ближайшего сусіда (к-Nearest Neighbor); байесові мережі (Bayesian Networks); індукція дерев рішень; нейроні мережі (neural networks).

Кластеризація (Clustering) Кластеризація є логічним продовженням ідеї класифікації. Це задача складніша, особливість кластеризації полягає в тому, що класи об'єктів спочатку не визначені. Результатом кластеризації є розбиття об'єктів на групи. Прикладом методу задачі кластеризації є особливий вид нейроних мереж (карти Кохонена), що само організовуються без вчителя.

Асоціація (Associations). В ході рішення задачі пошуку асоціативних правил відшукуються закономірності між зв'язаними подіями в наборі даних. Відмінність асоціації від двох попередніх задач Data Mining: пошук закономірностей здійснюється не на основі властивостей аналізує мого об'єкту, а між декількома подіями, які відбуваються одночасно.Самий відомий алгоритм рішення задачі пошуку асоціативних правил - алгоритм Apriori.

Послідовність (Sequence), або послідовна асоціація (sequential association) Послідовність дозволяє знайти тимчасові закономірності між транзакціями. Задача послідовності подібна асоціації, але її метою є встановлення закономірностей не між одночасно наступаючими подіями, а між подіями, зв'язаними в часі (тобто що відбуваються з деяким певним інтервалом в часі. Цю задачу Data Mining також називають задачею знаходження послідовних шаблонів (sequential pattern).Правило послідовності: після події X через певний час відбудеться подія У.

Прогнозування (Forecasting). В результаті рішення задачі прогнозування на основі особливостей існуючих даних оцінюються пропущені або ж майбутні значення цільових чисельних показників. Для вирішення таких задач широко застосовуються методи математичної статистики, нейроні мережі і ін. [12].

Візуалізація (Visualization, Graph Mining) В результаті візуалізації створюється графічний образ аналізованих даних. Для вирішення задачі візуалізації використовуються графічні методи, що показують наявність закономірностей в даних. Приклад методів візуалізації - представлення даних в 2-D і 3-D вимірюваннях. Підведення підсумків (Summarization) - задача, мета якої - опис конкретних груп об'єктів з аналізованого набору даних та інші. Задачі Data Mining, залежно від моделей, що використовуються, можуть бути

дескриптивними і прогнозуючими. В результаті рішення описових (descriptive) задач аналітик одержує шаблони, що описують дані, які піддаються інтерпретації. Ці задачі описують загальну концепцію аналізованих даних, визначають інформативні, підсумкові, відмітні особливості даних. Прогнозуючі (predictive) задачі грунтуються на аналізі даних, створенні моделі, прогнозі тенденцій або властивостей нових або невідомих даних [13].

II. Класифікація стадій Data Mining. Data Mining може складатися з двох або трьох стадій.

Стадія 1. Виявлення закономірностей (вільний пошук).

Стадія 2. Використовування виявлених закономірностей для прогнозу невідомих значень (прогностичне моделювання). На додаток до цих стадій іноді вводять стадію оцінювання (валідації) , наступну за стадією вільного пошуку. Мета валідації - перевірка достовірності знайдених закономірностей. Проте, ми вважатимемо валідацію частиною першою стадії, оскільки в реалізації багатьох методів, зокрема, нейроних мереж і дерев рішень, передбачений розподіл загальної множини даних на навчальні і перевірочні, і останні дозволяють перевіряти достовірність отриманих результатів.

Стадія 3. Аналіз виключень - стадія призначена для виявлення і пояснення аномалій, знайдених в закономірностях. Вільний пошук (Discovery). На стадії вільного пошуку здійснюється дослідження набору даних з метою пошуку прихованих закономірностей. Попередні гіпотези щодо виду закономірностей тут не визначаються. Закономірність (law) - істотний і постійно повторюється взаємозв'язок, що визначає етапи і форми процесу становлення, розвитку різних явищ або процесів. Система Data Mining на цій стадії визначає шаблони, для отримання яких в системах OLAP, наприклад, аналітику необхідно обдумувати і створювати множину запитів. Тут же аналітик звільняється від такої роботи - шаблони шукає за нього система [14].

Особливо корисно застосування даного підходу в надвеликих базах даних, де уловити закономірність шляхом створення запитів достатньо складно, для цього вимагається перепробувати безліч різноманітних варіантів.

Вільний пошук представлений такими діями:

- виявлення закономірностей умовної логіки (conditional logic);
- виявлення закономірностей асоціативної логіки (associations and affinities);
- виявлення трендів і коливань (trends and variations);
- індукції правил умовної логіки (задачі класифікації і кластеризації, опис в компактній формі близьких або схожих груп об'єктів);
- індукції правил асоціативної логіки (задачі асоціації і послідовності і витягування при їх допомозі інформація);
- визначення трендів і коливань (початковий етап задачі прогнозування).

На стадії вільного пошуку також повинна здійснюватися валідація закономірностей, тобто перевірка їх достовірності на частини даних, які не брали участь у формуванні закономірностей [20].

Друга стадія Data Mining - прогностичне моделювання - використовує результати роботи першої стадії. Тут знайдені закономірності використовуються безпосередньо для прогнозування.

Прогностичне моделювання включає такі дії:

- прогноз невідомих значень (outcome prediction);
- прогнозування розвитку процесів (forecasting).

В процесі прогностичного моделювання розв'язуються задачі класифікації і прогнозування. При рішенні задачі класифікації результати роботи першої стадії (індукції правил) використовуються для віднесення нового об'єкту з певною упевненістю до одного з відомих, наперед визначених класів на підставі відомих значень. При рішенні задачі прогнозування результати першої стадії (визначення тренда або коливань) використовуються для прогнозу невідомих (пропущених або ж майбутніх) значень цільової змінної (змінних). Порівняємо вільного пошуку і прогностичного моделювання з погляду логіки Вільний пошук розкриває загальні закономірності. Він по своїй природі індуктивний. Закономірності, отримані на цій стадії, формуються від часткового до загального. В результаті ми одержуємо деяке загальне знання про

деякий клас об'єктів на підставі дослідження окремих представників цього класу. Прогностичне моделювання, навпаки, дедуктивне. Закономірності, отримані на цій стадії, формуються від загального до часткового [21].

Тут ми одержуємо нове знання про деякий об'єкт або ж групі об'єктів на підставі:

- знання класу, до якого належать досліджувані об'єкти;
- знання загального правила, діючого в межах даного класу об'єктів.

На третій стадії Data Mining аналізуються виключення або аномалії, виявлені в знайдених закономірностях. Дія, виконувана на цій стадії, - виявлення відхилень (deviation detection). Для виявлення відхилень необхідно визначити норму, яка розраховується на стадії вільного пошуку. Стадія аналізу виключень може бути використана як очищення даних [23].

Є ряд машині інтелектуальні інструменти, які доступні на ринку, але в в той же час не всі інструменти є кращими для всіх проблем в наборі даних. Різні набори дані дають різні результати, засновані на алгоритмах, використовуваних. У роботі ми будемо вивчати деякі алгоритми, засновані на дерева рішень, на основі правил класифікації, ймовірність і м'яких обчислень.

Дерево рішень є одним з більш легкої структури даних, щоб зрозуміти, що є інтелектуальним аналізом даних. Правила від підготовка набору даних спочатку витягуються для формування дерева рішень, який потім використовується для класифікація тестування набору даних. Рішення Дерево обов'язково Дерево з довільним. Ступінь, яка класифікує випадки. Вони є потужним інструментом для класифікації та прогнозування але вимагають великих обчислень. Створення дерева на основі навчальної множини вимагає часу хоча ухвалення рішень, як тільки Дерево зроблено не багато часу. Класифікація дерева рішень: алгоритми можуть бути розділені на дві групи: одна, результат якого бінарне Дерево та інші, які без бінарнjго дерева (також називається багатоходових) [24].

У дереві рішень, листовий вузол являє собою повну класифікацію.

Примірник атрибута приймає рішення вузол визначає, що тест проводиться для отримання листової вузол. Таким чином, з дерева рішень, суб дерева, створеного після будь-якого вузла обов'язково результат тесту, проведеного.

Дерево використовується для класифікації певного примірника від кореня дерева до листової вузол, який забезпечує результат цього примірника. Одним з основних питань при допомоги дерева рішень з'ясувати, наскільки глибоко Дерево має рости, і коли воно повинне зупинитися. Зазвичай, якщо всі атрибути різні і призводять до того ж результату, рішення Дерево не може бути найбільш ефективне у прийнятті рішень і, в той же час, розмір дерева буде великим.

Є також ряд фільтрів, доступних, які застосовуються різні критерії для вибору об'єктів або атрибутів. Це дозволяє відмовитися від частини матриці без маніпулювати вихідним файл даних (який є поганою ідеєю в будь-якому випадку, і трудомістким). Наприклад, можа подивитися на підмножини атрибутів, відкинути перші 20 рядків, нормалізації або атрибути дискретизації і так далі. Щоб застосувати фільтр, необхідно спочатку вибрати, який тип фільтра необхідний, натиснувши на кнопку вибрати прямо під фільтр в основний GUI. Подвійне клацання на папці фільтр, який з'явився розширить вікно, щоб показати дві папки з іменами контрольовані і неконтрольовані, обидва з яких ви можете розширити знову (можна згорнути папки, натиснувши на знак мінус, який з'являється зліва в кожній папці, як тільки це розширені) , Обидва неконтрольовані і контрольовані фільтри можуть бути застосовані до об'єктів і атрибутів. Після того як було вибрано фільтр, обраний варіант буде відображатися в рядку поруч з фільтром, але на даному етапі, нічого не сталося з даними ще. Потім потрібно натиснути застосувати насправді відфільтрувати дані. Існує також кнопка SAVE, яка дозволяє зберегти всі зміни, внесені до даних. Файл журналу використовується, щоб відстежувати те, що було зроблено.

Метод регресійного аналізу є найпростішим і, мабуть, найменш ефективним методом інтелектуального аналізу даних. Найпростіша модель

аналізу використовує один вхідний (незалежний) параметр і один результуючий (залежний) параметр. Безумовно, модель можна ускладнити, додавши кілька десятків вхідних параметрів, але в кожному разі загальний підхід буде один і той же: на підставі кількох незалежних змінних визначається один залежний результат. Таким чином, модель регресійного аналізу використовується для прогнозування значення однієї залежної змінної, виходячи з відомих значень декількох незалежних параметрів.

Тим не менш, для середньостатистичного користувача кластеризація може виявитися найбільш корисним методом інтелектуального аналізу даних. Цей метод дозволить швидко розбити ваші дані на окремі групи і зробити конкретні висновки і припущення щодо кожної групи. Математичні методи, що реалізують кластерний аналіз, досить складні і заплутані, так що в разі кластеризації ми будемо цілком покладатися на обчислювальні можливості WEKA [28].

Далі пропонується стислий загальний опис математичних методів і алгоритмів, що використовуються методом кластеризації.

Кожен атрибут має бути приведений до нормального вигляду. Для цього кожен показник ділиться на різницю між найбільшим і найменшим значенням, які приймає розглянутий атрибут на конкретному наборі даних.

Виходячи з бажаної кількості кластерів, випадковим чином вибирається така ж кількість рядків даних. Ці рядки будуть використовуватися в якості початкових центрів мас кластерів.

Для кожного рядка даних визначається відстань від цього рядка до центру мас кластера (випадковим чином вибраного рядка даних) за допомогою методу найменших квадратів.

Кожен рядок набору даних включається в той кластер, відстань до центру мас якого виявилося найменшим.

У кожному кластері визначається новий центр мас як набір середніх значень по стовпцях на безлічі елементів цього кластера.

Визначається відстань від кожного елемента даних до нового центру мас. Якщо при цьому розподіл елементів по кластерах не змінюється, розбивка даних на кластери закінчено, і всі групи даних визначені. Якщо склад кластерів змінюється, слід повернутися до п.3 і повторювати цей процес до тих пір, поки розбивка на кластери не стане незмінним.

Кінцева мета інтелектуального аналізу даних є створення моделі, модель, яка може поліпшити спосіб читати і інтерпретувати існуючі дані і ваші майбутні дані. Так як існує дуже багато методів з інтелектуального аналізу даних, основним кроком до створення оптимальної моделі, щоб визначити, який тип техніки для використання. Це прийде з практикою і досвідом, а також деякі рекомендації. Звідти, модель повинна бути вдосконалена, щоб зробити його ще більш корисним.

1.3 Програмні засоби інтелектуального аналізу даних

Інтелектуальний аналіз даних не є винятковою прерогативою великих компаній і дорогого програмного забезпечення.

Існує кілька програмних засобів інтелектуального аналізу даних, основні з яких: WEKA, MatLab (Image Processing Toolbox), RapidMiner, KNIME та GMDH Shell. Розглянемо кожен із них детально.

RapidMiner - являє собою програмного забезпечення для інтелектуального аналізу даних. Можна використовувати RapidMiner як автономне додаток для аналізу даних, або інтегрувати його в якості інтелектуального аналізу даних в власні продукти.

Основні особливості RapidMiner:

- Інтеграція даних, аналітична ETL, аналіз даних і звітність в єдиний документ

- Потужний, але інтуїтивно зрозумілий GUI для розробки аналітичних процесів.

- Репозиторій для процесів, даних і управління метаданими.

- Єдине рішення з перетворенням метаданих: методом проб і помилок, перевірка результатів вже під час розробки.

- Єдине рішення , яке підтримує на льоту виявлення помилок і одночасне швидке виправлення.

- Повнотаа і гнучкість: сотні методів інтеграції даних, перетворення даних, моделювання і візуалізації.

KNIME є Java з відкритим вихідним кодом, крос-платформних додатків, назва якого означає "Konstanz Information Miner". Це насправді широко використовується для інтелектуального аналізу даних, аналізу даних і оптимізації. Його можна завантажити як сам (Knime Desktop) основного додатка або весь SDK, який заснований на Eclipse, Helios.

Програмне забезпечення knime також може працювати з різними типами розширень, які вбудовані в "/ завантаження / розширень" вкладки веб-сайту.

GMDH Shell - Потужне програмне забезпечення прогнозування для малого бізнесу, торговців і вчених. GMDH Shell є найпростішим способом точного прогнозування часових рядів, створювати класифікатори і регресійних моделей. На основі штучних нейронних мереж, що дозволяє легко створювати моделі, а також дані Preprocess з мертвої точки.

На відміну від інших інструментів на основі нейронних мереж - GMDH Shell дуже швидкий через запроваджену паралельну обробку і велику оптимізацію ключових алгоритмів [29].

MatLab - Image Processing Toolbox - надає повний набір еталонних стандартних алгоритмів, функцій і додатків для обробки зображень, аналізу, візуалізації та розробки алгоритмів. Можна виконувати аналіз зображення, сегментацію зображення , підвищення якості зображення , зменшення шуму, геометричні перетворення та реєстрації зображення. Багато функцій

підтримуютьнабір інструментів багатоядерних процесорів, графічних процесорів, а також генерацію С-коду.

Image Processing Toolbox підтримує широкий набір типів зображень , в тому числі високий динамічний діапазон, дозвіл гігапікселя. Функції візуалізації і додатки дозволяють досліджувати зображення і відео, вивчити область пікселів, налаштувати колір і контрастность, створювати контури або гістограми. Набір інструментів підтримує робочі процеси для обробки, відображення і навігації великих зображень [30].

WEKA - продукт університету Уайкато (Нова Зеландія), який вперше був випущений в його сучасному вигляді в 1997 році. WEKA розповсюджується за ліцензією GNU General Public License (GPL). Це ПЗ написано на мові Java ™ і забезпечує графічний користувальницький інтерфейс для роботи з файлами даних і генерації візуальних результатів (у вигляді таблиць і графіків). Крім того, ви можете інтегрувати WEKA, як і будь-яку іншу бібліотеку, в свої власні додатки, наприклад, для автоматизації аналізу даних на стороні сервера, використовуючи стандартний API [31].

Weka дозволяє виконувати такі завдання аналізу даних, як підготовку даних (preprocessing), відбір ознак (feature selection), кластеризацію,класифікацію, регресійний аналіз та візуалізацію результатів.

Основним інтерфейсом користувача є Explorer, хоча ті ж функціональні можливості підтримуються з командного рядка та інтерфейсу Knowledge Flow. Для систематичного порівняння різних алгоритмів машинного навчання використовується інтерфейс Experimenter. Він дозволяє порівнювати результати не лише різних алгоритмів на одному наборі даних, а й одного алгоритму на різних наборах даних.

Інтерфейс Explorer містить наступні панелі:

- Панель попереднього опрацювання уможливлює імпорт даних з бази даних, текстових файлів у форматі CSV, а також попереднє опрацювання цих даних за допомогою різноманітних алгоритмів (фільтрів). Ці фільтри

використовуються для трансформування даних, а також для видалення певних атрибутів.

- Панель класифікації надає можливість застосувати алгоритми класифікації та регресійного аналізу до обраного набору даних, візуалізувати та оцінити результати, відобразити ROC криві тощо.

- Панель асоціації надає доступ до методів, які дозволяють оцінити взаємозв'язки між атрибутами.

- Панель кластеризації містить різноманітні методи кластеризації, наприклад метод кластеризації методом k-середніх, EM-алгоритм тощо.

- Панель вибору атрибутів дозволяє ідентифікувати атрибути, які найбільш впливають на якість прогнозування.

- Панель візуалізації відображає точкові діаграми [31].

WEKA це засіб для машинного навчання, алгоритми для аналізу даних завдань. алгоритми можуть бути або наносять безпосередньо на набір даних або викликані з вашого власного Java коду. WEKA містить інструменти для попередньої обробки даних, Класифікація, регресія, кластеризація, асоціативні правила, і візуалізація. Це також добре підходить для розробки нової машини схеми навчання.

Алгоритми навчання в Weka отримані з абстрактного класу: Weka.classifiers.Classifier.

Блок Explorer містить у собі попередню обробку,класифікацію, кластер, асоціативність, вибір атрибутів та візуалізацію [32].

Попередня обробка (Preprocess) включає:

- завантаження даних;
- попередня обробка даних;
- аналіз атрибутів.

Класифікація включає:

- вибірка Опції тесту;
- використання навчальної вибірки;
- класифікатори на виконання;

- перегляд результатів.

Експериментатор. Дозволяє користувачам створювати, запускати, модифікувати і проаналізувати експерименти в більш зручній формі, ніж при індивідуальній обробці.

Експериментатор: Встановлення/налаштування: простий або розширений

Напрям результатів:

- ARFF.

- CSV.

- База даних JDBC.

Є також ряд фільтрів, доступних, які застосовуються різні критерії для вибору яких об'єктів або атрибутів. Це дозволяє відмовитися від частини матриці без маніпулювати вихідним файл даних (який є поганою ідеєю в будь-якому випадку, і трудомістким). Наприклад, можа подивитися на підмножини атрибутів, відкинути перші 20 рядків, нормалізації або атрибути дискретизації і так далі. Щоб застосувати фільтр, необхідно спочатку вибрати, який тип фільтра необхідний, натиснувши на кнопку вибрати прямо під фільтр в основний GUI [33]. Подвійне клацання на папці фільтр, який з'явився розширить вікно, щоб показати дві папки з іменами контрольовані і неконтрольовані, обидва з яких ви можете розширити знову (можна згорнути папки, натиснувши на знак мінус, який з'являється зліва в кожній папці, як тільки це розширені) , Обидва неконтрольовані і контрольовані фільтри можуть бути застосовані до об'єктів і атрибутів. Після того як було вибрано фільтр, обраний варіант буде відображатися в рядку поруч з фільтром, але на даному етапі, нічого не сталося з даними ще. Потім потрібно натиснути застосувати насправді відфільтрувати дані. Існує також кнопка SAVE, яка дозволяє зберегти всі зміни, внесені до даних. Файл журналу використовується, щоб відстежувати те, що було зроблено [34].

Метод регресійного аналізу є найпростішим і, мабуть, найменш ефективним методом інтелектуального аналізу даних. Найпростіша модель аналізу використовує один вхідний (незалежний) параметр і один результуючий

(залежний) параметр (прикладами такої моделі є точкові діаграми Excel і аналогічні їм XYDiagram в OpenOffice.org).Безумовно, модель можна ускладнити, додавши кілька десятків вхідних параметрів, але в кожному разі загальний підхід буде один і той же: на підставі кількох незалежних змінних визначається один залежний результат. Таким чином, модель регресійного аналізу використовується для прогнозування значення однієї залежної змінної, виходячи з відомих значень декількох незалежних параметрів.

Для того щоб завантажити дані в WEKA, їх слід перетворити у формат, зрозумілий цього пакета програмного забезпечення. Найбільш підходящим форматом для завантаження даних в WEKA є формат Attribute-Relation File Format (ARFF), який спочатку визначає тип завантажуваних даних, а потім вказує власне дані. У файлі формату ARFF ви вказуєте назву і тип даних для кожного стовпця таблиці, а потім власне дані по рядках. У моделях регресійного аналізу використовуються лише два типи даних: NUMERIC і DATE. Після того, як ви описали всі стовпці таблиці, ви додаєте дані по рядках, використовуючи як роздільник кому [35].

Кластеризація дозволяє розбити дані на групи, кожна з яких має певні ознаки. Метод кластерного аналізу використовується в тих випадках, коли необхідно виділити деякі правила, взаємозв'язку або тенденції у великих наборах даних. В залежності від потреб бізнесу, ви можете виділити кілька різних груп даних. Одне з явних переваг кластеризації в порівнянні з класифікацією полягає в тому, що для розбиття множини на групи може використовуватися будь-який атрибут (метод класифікації використовує тільки певну підмножину атрибутів). Як основний недолік методу кластеризації слід згадати той факт, що укладач моделі повинен заздалегідь вирішити, на скільки груп слід розбити дані. Для людини, яка не має жодного уявлення про конкретний наборі даних, прийняти таке рішення досить важко. Чи слід нам створити три групи або п'ять груп? А може, нам потрібно визначити десять груп? Може знадобитися декілька ітерацій проб і помилок, для того щоб визначити оптимальну кількість кластерів.

Тим не менш, для середньостатистичного користувача кластеризація може виявитися найбільш корисним методом інтелектуального аналізу даних. Цей метод дозволить швидко розбити ваші дані на окремі групи і зробити конкретні висновки і припущення щодо кожної групи. Математичні методи, що реалізують кластерний аналіз, досить складні і заплутані, так що в разі кластеризації ми будемо цілком покладатися на обчислювальні можливості WEKA.

Далі пропонується стислий загальний опис математичних методів і алгоритмів, що використовуються методом кластеризації.

1. Кожен атрибут має бути приведений до нормального вигляду. Для цього кожен показник ділиться на різницю між найбільшим і найменшим значенням, які приймає розглянутий атрибут на конкретному наборі даних. Наприклад, якщо розглянутий атрибут - вік, і його найбільше значення - 72, а найменше - 16, то значенням 32 буде відповідати нормалізована величина 0.5714.

2. Виходячи з бажаної кількості кластерів, випадковим чином вибирається така ж кількість рядків даних. Ці рядки будуть використовуватися в якості початкових центрів мас кластерів.

3. Для кожного рядка даних визначається відстань від цього рядка до центру мас кластера (випадковим чином вибраного рядка даних) за допомогою методу найменших квадратів.

4. Кожен рядок набору даних включається в той кластер, відстань до центру мас якого виявилося найменшим.

5. У кожному кластері визначається новий центр мас як набір середніх значень по стовпцях на безлічі елементів цього кластера.

6. Визначається відстань від кожного елемента даних до нового центру мас. Якщо при цьому розподіл елементів по кластерах не змінюється, розбивка даних на кластери закінчено, і всі групи даних визначені. Якщо склад кластерів змінюється, слід повернутися до п.3 і повторювати цей процес до тих пір, поки розбивка на кластери не стане незмінним.

В цілому, всі ці обчислення виглядають досить нудно. Для того щоб розбити набір даних з 10 рядків на три кластери за допомогою електронних таблиць, буде потрібно приблизно півгодини напруженої роботи.

Деяка кількість методів видобутку даних реалізовані в програмному забезпеченні WEKA.

Деякі з них засновані на дереві рішень, як J48 дерева рішень, деякі з них на основі правил, як ZeroK і таблиць рішень, і деякі з них засновані на ймовірності та регресії, як наївний алгоритм Байеса [36].

Алгоритм C4.5 є частиною багатостороннього дерева рішень. З 4.5 дає бінарний розподіл, якщо обрана змінна на чисельні, але якщо є інші змінні, що представляють атрибути, це призведе до категоричних змін. Тобто, вузол - буде розділений на вузлах, де C кількість категорій для цього атрибута. J48 Дерево рішень в WEKA заснований на алгоритмі C4.5 дерева рішень.

Метод J48 є модифікацією методу C4.5, а він у свою чергу - вдосконалений варіант алгоритму ID3. Метод працює як з номінальними, так і з числовими змінними. Пропущені дані також не заважають роботі алгоритму, оскільки передбачається, що пропущені значення по змінній імовірнісний розподілений пропорційно частоті появи існуючих значень.

Дерево має кращий вигляд, чим при використанні методу ID3. Це викликано більш ранньою зупинкою алгоритму. Також досягається вища точність.

Алгоритм J4.8 має декілька удосконалень, в порівнянні з ID3:

- Можливість працювати не лише з категоріальними атрибутами, але також з числовими.

- Після побудови дерева відбувається усікання його гілок. Якщо Дерево, що вийшло, занадто велике, виконується або угрупування декількох вузлів в один лист, або заміщення вузла дерева подДеревом, що пролягає нижче.

Перед операцією над Деревом обчислюється помилка правила класифікації, що міститься в даному вузлі. Якщо після заміщення (чи

угрупування) помилка не зростає (і не сильно збільшується ентропія), означає заміну можна провести без збитку для побудованої моделі.

Метод SVM (у середовищі Weka він називається SMO). Для цього методу не вимагається яких-небудь перетворень початкової вибірки.

Цей метод є алгоритмом класифікації з використанням математичних функцій. Метод використовує нелінійні математичні функції, номінальні дані перетворяться в числові. Основна ідея методу опорних векторів - переведення початкових векторів в простір вищої розмірності і пошук максимальної розділяючої гіперплощини в цьому просторі.

На виведенні алгоритму показуються ваги для усіх можливих атрибутів, при цьому помітна затримка його виводу із-за проведення розрахунків. Відсоток вірної класифікації виявляється досить високим , а середня помилка класифікатора навпаки, виявляється мінімальною серед усіх розглянутих методів.

У результаті виведення цього алгоритму представлене у вигляді вектора n -мірного простору. Цифри, вказані у виводі, - коефіцієнти задаючі площину, що розділяє початкові дані на типи.

Алгоритми машинного навчання розроблені, щоб навчити себе на основі шаблонів і правил, витягнутих з навчального набору даних. Таким чином, маючи хороший набір навчання може підвищити ефективність по відношенню до видобутку правил і шаблонів. Є два способи вибору підмножини атрибутів. Перший полягає у використанні "методу фільтра", де атрибути фільтруються, щоб мати кращий набір результатів до процедури навчання, другий складається з «методу оболонки", де метод навчання знаходиться в межах.

Процедура відбору. У таблиці рішення, що використовується в WEKA же атрибут виділення за допомогою метод обгортки. Атрибут заснований на вимірюванні продуктивність перевірки для різних підмножинах атрибутів і виборі найкращої підмножини. Існує також варіант в WEKA де можна встановити близьке до цього примірника, який покращує продуктивність інструменту значно.

Цей метод заснований на ймовірності знань. Цей метод носить назву наївний байєсоівський, тому що він заснований на правилі Байеса і "наївно" припускає незалежність - це дійсні тільки помножити ймовірність, коли події є незалежними. Таким чином, наївно правилом Байєса виводить ймовірностей прогнозованого класу кожного елемента набору тесту екземпляра. Наївний байесівський на основі контрольованого навчання. Мета передбачити клас тести з інформацією класу, надається в навчальних даних.

Класифікація наївний байесовский читає набір прикладів з навчальної множини і використовує теоремі Байеса, щоб оцінити вірогідність всіх класифікацій. Для кожного екземпляра, то класифікація з найбільшою ймовірністю буде обрана як клас прогнозування.

Наївний байєсовський класифікатор традиційно робить припущення, що один гауссовий розподіл генерує числові атрибути. Два типи наявного Байеса

Алгоритми наведені нижче:

- Наївний байесовский (NB).
- Простий наївний байесовский (СНБ).

Різниця між ними полягає в тому, що наївний байесовский має таку ймовірність, атрибути якого розраховується на основі середнього, стандартного відхилення нормального розподілу в сумі, але простий наївний байесовский - тільки на основі середнього значення і стандартного відхилення [37].

Властивості наївної класифікації :

- Використання усіх змінних і визначення усіх залежностей між ними.
- Наявність двох припущень відносно змінних: усі змінні є однаково важливими та усі змінні є статистично незалежними, тобто значення однієї змінної нічого не говорить про значення інший.

1.4 Постановка завдання на дипломну роботу

Аналіз біомедичних зображень має важливе значення в сучасній медицині. Виникають проблеми, які зумовлені новими завдання опису зображення та пошуком закономірностей, що одночасно зустрічаються в багатьох наборах. Оскільки наборів може бути велика кількість необхідно здійснювати цей пошук автоматично. Тому актуальною задачею є розробка алгоритму для пошуку асоціативних правил бази даних цитологічних та гістологічних зображень, що містять кількісні і якісні ознаки мікрооб'єктів.

Метою роботи є пошук асоціативних правил в базі даних цитологічних та гістологічних зображень диспластичних і ракових процесів молочної залози, використовуючи різні алгоритми інтелектуального аналізу даних.

Проведення аналізу алгоритмів пошуку асоціативних правил та аналіз отриманих правил дає змогу встановити певні закономірності, залежності, що у свою чергу спрощує розуміння передракових та ракових процесів молосної залози.

Отже, методи пошуку асоціативних правил в базі даних біомедичних зображень є актуальною задачею та має практичне значення для медицини.

2 АЛГОРИТМИ ПОШУКУ АСОЦІАТИВНИХ ПРАВИЛ

2.1 Алгоритм Apriori та його модифікації

Алгоритм Apriori визначає набори, які часто зустрічаються за кілька етапів. На і -му етапі визначаються всі часто зустрічаються і - елементні набори. Кожен етап складається з двох кроків:

- формування кандидатів (candidate generation),
- підрахунок підтримки кандидатів (candidate counting).

Розглянемо і -й етап. На кроці формування кандидатів алгоритм створює безліч кандидатів з і - елементних наборів, чия підтримка поки не обрахується. На кроці підрахунку кандидатів алгоритм сканує безліч транзакцій, обчислюючи підтримку наборів-кандидатів. Після сканування відкидаються кандидати, підтримка яких менше певного користувачем мінімуму, і зберігаються тільки часто зустрічаються і-елементі набори. Під час 1-го етапу вбрання безліч наборів-кандидатів містить всі 1-елементні часті набори. Алгоритм обчислює їх підтримку під час кроку підрахунку кандидатів.

Для підрахунку підтримки кандидатів потрібно порівняти кожну транзакцію з кожним кандидатом. Очевидно, що кількість кандидатів може бути дуже великою і потрібен ефективний спосіб підрахунку. Набагато швидше і ефективніше використовувати підхід, заснований на зберіганні кандидатів в хеш-дереві. Внутрішні вузли дерева містять хеш-таблиці з покажчиками на нащадків, а листя - на кандидатів. Це Дерево використовується при швидкому підрахунку підтримки для кандидатів.

Хеш-дерево будується кожен раз, коли формуються кандидати. Першого Дерево складається тільки з кореня, який є листом, і не отримує ніяких кандидатів-наборів. Кожен раз, коли формується новий кандидат, він заноситься в корінь дерева, і так до тих пір, поки кількість кандидатів в корені-листі не перевищить певного порогу. Як тільки це відходить, корінь перетворюється в хеш-таблицю. Стає внутрішнім вузлом, і для нього

створюються нащадки-листя. Всі кандидати розподіляються по вузлах-нащадкам згідно хеш-значень елементів, що входять в набір. Кожний новий кандидат хешується на внутрішніх вузлах, поки не досягне першого вузла-листа, де він і буде зберігатися, поки кількість наборів знову ж таки не перевищить порогу.

Після того як хеш-дерево з кандидатами-наборами побудовано, легко підвважати підтримку для кожного кандидата. Для цього потрібно "пропустити" кожну транзакцію через Дерево і збільшити лічильники для тих кандидатів, чиї елементи також містяться і в транзакції, $Ck \cap Ti = Ck$. на кореневому рівні хеш-функція застосовується до кожного об'єкта з транзакції. Далі, на другому рівні, хеш-функція застосовується до других об'єктів і т. д. На k-му рівні хешується k-елемент, і так до тих пір, поки не досягнемо листа. Якщо кандидат, що зберігається в листі, є підмножиною поточної транзакції, збільшуємо лічильник підтримки цього кандидата на одиницю.

Після того як кожна транзакція з вихідного набору даних "пропущена" через Дерево, можна перевірити, чи задовольняють значення підтримки кандадат мінімального порогу. Кандидати, для яких ця умова виконується, переносяться в розряд часто зустрічаються. Крім того, слід запам'ятати і підтримку набору, яка стане в нагоді при добуванні правил. Ці ж дії застосовуються для знаходження ($k + 1$) - елементних наборів і т. д [38].

При використанні методу асоціативних правил, метою аналізу є встановлення залежностей виду: якщо в структурній одиниці даних зустрівся деякий набір елементів X, то на підставі цього можна зробити висновок про те, що інший набір елементів Y також має з’явитися в цій одиниці. Ці правила мають такий вигляд (2.1):

$$X \rightarrow Y. \qquad (2.1)$$

Нехай I – скінченна множина елементів. Нехай D – набір рядків інформаційної таблиці. Кожному такому рядку відповідає транзакція T, що

складається з підмножини елементів *I, T* $\subseteq$ *I.* Елемент входить в транзакцію, якщо у відповідному рядку відповідний атрибут набуває значення „1”.

Транзакція *T* містить *X*, деяку підмножину елементів *I*, якщо *X*$\subseteq$ *T*. Асоціативним правилом називається імплікація *X*$\Rightarrow$*Y*, де *X*$\subset$ *I, Y* $\subset$ *I* та *X*$\cap$*Y*=$\emptyset$.

Кажуть, що правило X$\Rightarrow$Y має підтримку (support) s, якщо s% транзакцій з *D,* містять множину *X*$\cup$*Y*(за формулою 2.2),

$$supp(X \Rightarrow Y) = supp\ (X \cup Y) = \frac{count\ (T: X \cup Y \subseteq T)}{size(D)} - 100\%. \quad (2.2)$$

Достовірність (confidence) правила показує, яка ймовірність того, що з *X* випливає *Y*. Кажуть, що правило *X*$\Rightarrow$*Y* справедливе з достовірністю c, якщо c% транзакцій з *D*, що містять *X,* також містять *Y* (за формулою 2.3),

$$conf\ (X \Rightarrow Y) = \frac{supp\ (X \cup Y)}{supp\ (X)} \cdot 100\%. \quad (2.3)$$

Поліпшення (improvement) - показує, корисніше правило випадкового вгадування. Поліпшення правила є відношенням числа транзакцій, з-тримають набори *X* і *Y*, до добутку кількості транзакцій, набір *X*, і кількості транзакцій, що містять набір *Y* (формула 2.4):

$$impr_{X \to Y} = \frac{|D_{F=X \cup Y}|}{|D_X||D_Y|} = \frac{Supp_{X \cup Y}}{Supp_x \cdot Supp_Y}. \quad (2.4)$$

Якщо поліпшення більше одиниці, то це означає, що за допомогою правила передбачити наявність набору *Y* найімовірніше, ніж випадкове вгадування, якщо менше одиниці, то навпаки.

Алгоритм Apriori, розроблений є великим досягненням в історії правил асоціації. Він на сьогоднішній день є найбільш відомим алгоритмом асоціативних правил [39]. Цей метод використовує властивість, що будь-яка

підмножина велика наборів повинна бути великими наборами. Крім того, передбачається, що елементи в межах наборів зберігаються в лексикографічному порядку. Apriori генерує кандидатів набори яких часто зустрічаються шляхом об'єднання великих з наборами попереднього без урахування транзакцій в базі даних. Лише з урахуванням великих зустрічаються наборів попереднього проходу, число кандидатів великий наборів значно знижується.

Apriori включає в себе управління буфером для обробки того факту, що всі великі *Lk*-набори і кандидата Ск наборів повинні зберігатися в фазі генерації кандидата тому що може не поміститися в пам'яті.

Аналогічна проблема може виникнути на етапі підрахунку, де для зберігання *Ck* і, щонайменше одна сторінка в буфері. Необхідні транзакції бази даних розглядаються два підходи для вирішення цих питань. Спочатку вони передбачається, що L_{k-1} поміщається в пам'яті, але *Ск* не робить. Автори вирішили цю проблему шляхом зміни apriori_gen () так, що він генерує кількість кандидатів встановлює *Ck*, який поміщається в пам'яті. Великі *Lk* набори в результаті *Ck* записуються на диск, в той час як маленькі наборіи будуть видалені. Цей процес триває, поки всі *Ck* була виміряна. Другий сценарій, що *Lk -1* не поміщається в пам'яті. Ця проблема вирішується шляхом сортування *Lk -1* зовні. Блок *Lk-1* вводиться в пам'ять, в якій перші (*K-2*) елементи є те ж саме. Блоки *Lk-1* зчитуються і кандидати генеруються, поки пам'ять не заповниться. Цей процес триває поки все Ск не зарахований [40].

Фрагмент коду для предсталення алгоритму показаний нижче:

```
Function count(C: a set of itemsets, D: database)
begin
for each transaction T D="Di do begin
forall subsets x T do
if x C then
x.count++;
end
```

end

Algorithm . Apriori

Input:

I, D, s

Output:

L

Алгоритм:

//процедура великих наборів

1) C1: = I; //Кандадат 1-наборів

2) генерувати L1 шляхом обходу бази даних і підрахунку кожного входження атрибуту в транзакції;

3) for (k = 2; Lk-1≠ φ; k++) do begin

//генерування наборів кандидатів

//Новий k-кандадат набори згенеровані для (k-1)-large наборів

4) Ck = apriori-gen(Lk-1);

//підрахунок підтримки Ck

5) Count (Ck, D)

6) Lk = {c Ck | c.count ≥ minsup}

7) end

9) L = kLk

PredictiveApriori - клас реалізації інтелектуального алгоритму Apriori для видобування асоціативних правил.

Він виконує пошук зі збільшенням порогу підтримки для кращих 'N' правил, що стосуються скоригованого значення достовірності на основі підтримки [41].

Правило додається, якщо: очікувана прогностична точність цього правила є одним з 'N' найкращих і не поглинена правилом, принаймні тієї ж очікуваної точності прогнозу. Якщо включені асоціативні правила видобуваються замість (загальних) асоціативних правил.

Якщо встановлено значення -1, останній атрибут буде взятий в якості атрибута класу та існує кількість правил для знаходження. Можливі пропущені значення, бінарні атрибути, порожні номінальні атрибути та номінальний клас.

Алгоритм AprioriTid є різновидом алгоритму Apriori. Відмінною рисою даного алгоритму є підрахунок значення підтримки кандидатів не при скануванні безлічі *D*, а за допомогою безлічі *Ck*, які є множиною кандидатів (k-елементних наборів) потенційно частих, у відповідність яким ставиться ідентифікатор *TID* транзакцій, в яких вони містяться.

Кожен член безлічі *Ck* є парою виду <*TID, Fk*>, де кожен *Fk* є потенційно частим k - елементним набором, представленим в транзакції з ідентифікатором *TID*. Безліч *C1 = D* відповідає безлічі транзакцій, хоча кожен об'єкт в транзакції відповідає однооб'єктному набору в безлічі *C1*, що містить цей об'єкт. Для *k> 1* безліч *Ck* генерується відповідно до алгоритму, описаним нижче.

Член безлічі *Ck*, відповідний транзакції *T*, є парою такого вигляду (формула 2.5):

$$< T.TID, \{c \in C_k | c \in C_k \, | c \in T >. \qquad (2.5)$$

Підмножина наборів в *Ck* з однаковими *TID* (містяться в одній і тій же транзакції) називається записом. Якщо транзакція не містить ні одного k - елементного кандидата, то *Ck* не матиме записи для цієї транзакції Тобто кількість записів в *Ck* може бути менше, ніж в *D*, особливо для великих значень *k*. Крім того, для великих значень *k* кожна запис може бути менше, ніж відповідна їй транзакція, т. К. В транзакції буде міститися мало кандидатів. Однак для малих знань *k* кожен запис може бути більше, ніж відповідна транзакція, тобто *Ck* включає всіх кандидатів *k*-елементних наборів, що містяться в транзакції.

Апріорі-TID використовує генерацію кандидата і апріорну функцію для визначення кандидата для наборів перед початком проходу. Основна відмінність від Apriori в тому, що він не використовує базу даних, для підрахунку підтримки після першого проходу. Швидше, він використовує

кодування кандидата наборів елементів, який використовується в попередному прохододі і позначається *Ck*. У Apriori-TID, кандидат в *Ck* наборів зберігаються в масиві індексується TID в *Ck* наборів. Кожен *Ck* зберігається в послідовної структури. У проході k-го, Апріорі-TID потребує обсяг пам'яті для *Lk-1* і *Ck* при генерації кандидатів. Було також встановлено, що Апріорі-TID Перевищує Апріорі, коли є менше число наборів *Ck*, яке може помістится в пам'яті і розподіл з великих наборів має довгий набір . Це означає, що розподіл записів у великих наборів високий на ранній стадії [42].

Іншим різновидом алгоритму Apriori є алгоритм MSAP (Mining Sequential Alarm Patterns), спеціально розроблений для виконання секвенціального аналізу збоїв телекомунікаційної мережі.

Він використовує наступне властивість підтримки послідовностей: для будь-якій послідовності *Lk* її підтримка буде менше, ніж підтримка послідовностей з безлічі *Lk1*.

Алгоритм MSAP для пошуку подій, що слідують один за одним, використовує поняття "термінового вікна" (Urgent Window). Це дозволяє виявляти не просто однакові послідовності подій, а наступні один за одним. В іншому даний алгоритм працює за тим же принципом, що і Apriori.

2.2 Алгоритми без генерування кандидатів

FP-Growth алгоритм, запропонований Ханвом, є ефективним і масштабованим методом для видобутку повних наборів частих шаблонів зростанням фрагмента загальних даних. Він використовує розширену структуру префікса дерева, щоб зберігати стиснуті дані і важливу інформацію про частих моделей, названих частинами дерева (FP-Дерево). У своєму дослідженні, Хан довів, що його метод перевершує інші популярні способи видобутку шаблонів, які часто зустрічаються, наприклад, Apriori алгоритму і Дерево проекція. У

деяких більш пізніх роботах було доведено, що FP-Growth має кращу продуктивність по порівнянні з іншими методами, в тому числі і Relim. Популярність і ефективність FP-Growth алгоритму сприяє багатьм дослідженням, які пропонують варіанти, щоб поліпшити його продуктивність.

FP-Growth алгоритм є альтернативним способом для пошуку наборів які часто зустрічаються без використання кандидатів поколінь, тим самим підвищуючи продуктивність. Він використовує стратегію розділяй і володарюй. Суть цього методу є використання спеціальної структури даних з ім'ям частих моделей дерева (FP-Growth), яка зберігає інформацію про асоціації [43].

Цей алгоритм працює наступним чином: спочатку він стискає базу даних введення створення екземпляра FP-Growth для подання частих предметів. Після цього першого кроку він ділить стислу базу даних в набір умовних баз даних, кожен з яких пов'язаний з одним частим малюнком. Нарешті, кожна така база даних видобувається окремо. За допомогою цієї стратегії, FP-Growth рекурсивно знижує вартість пошуку коротких шаблонів, а потім їх конкатенує в довгі часті моделі, пропонуючи гарну вибірку.

У великих базах даних, це не представляється можливим провести FP-Дерево в основний пам'яті. Стратегія, щоб впоратися з цією проблемою є, по-перше розділити базу даних в набір невеликих баз даних (званих проектуються баз даних), а потім побудувати FP-Growth з кожного з цих невеликих баз даних. Наступні підрозділи описують структуру FP-Дерево і FP-Growth алгоритм, приклад представлений, щоб зробити його простіше для розуміння цих понять.

Дерево-часті моделі (FP-Дерево) являє собою компактну структуру, яка зберігає кількісну інформацію про часті шаблонів в базі даних.

Хан визначає FP-Дерево в вигляді дерева , визначеної нижче:

Один корінь позначені як "нуль" з набором пункт префікс піддерев, як діти, і стіл для часто елемент-заголовка;

Кожен вузол в елемент префікса піддерева складається з трьох полів:

- Пункт ім'я: регістри, який елемент представлений вузлом;

- Кількість: кількість транзакцій, представлених на ділянці шляху, що досягає вузол;

- Вузол-посилання: посилання на наступний вузол в FP-Дерево, що несуть один і той же предмет-ім'я, або нульове значення, якщо його немає.

Кожен запис в таблиці часті елемент-заголовка складається з двох полів:

- Пункт ім'я: як же до вузла.

- Керівник вузла-посилання: покажчик на перший вузол в FP-дерева, що несе ім'я-елемента [44].

Додатково таблиця часті елемент заголовка може мати підтримку рахунки для елемента. На рисунку 2.1 нижче показаний приклад FP-дерева.

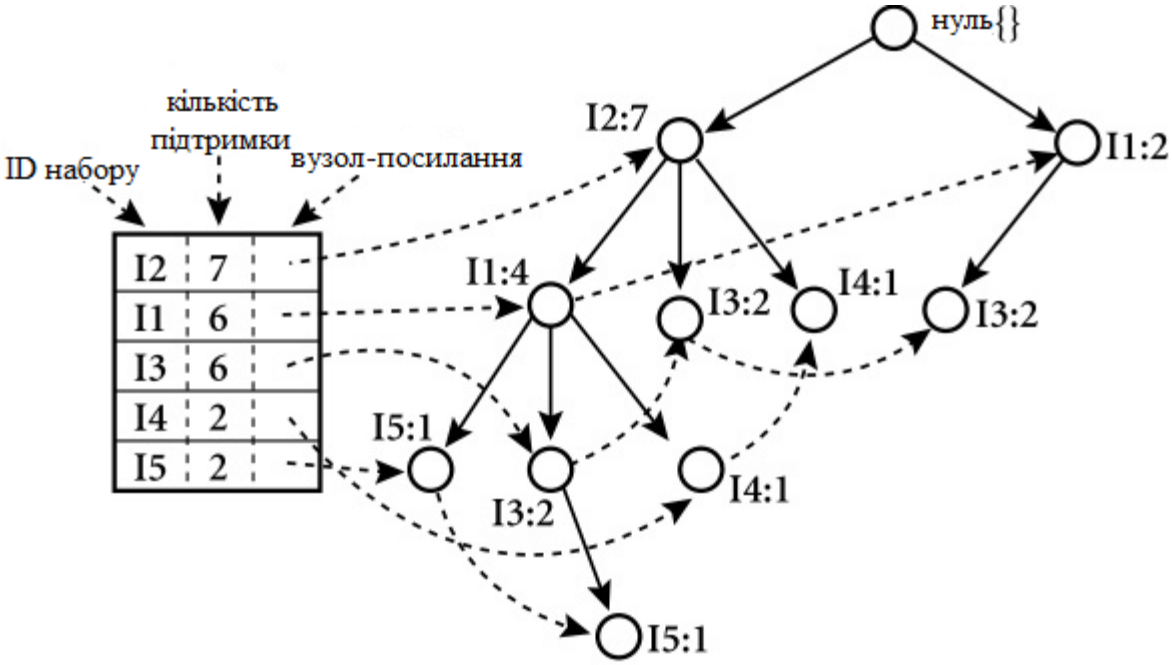

Рисунок 2.1 - Приклад FP-дерева

Оригінальний алгоритм побудови FP-дерева, визначається Ханом представлена нижче в алгоритмі:

- Введення : База даних транзакцій БД і мінімальний поріг підтримки.

- Висновок : FP-Дерево, частий-візерунок дерева БД.

- Метод : FP-Дерево будується таким чином.

Сканування БД бази даних транзакцій відбуваєтьсяодин раз. Відбувається збір F, безліч частих елементів, а також підтримку кожного частого пункту. Сортування F в порядку спаданн допоміжних, як FList, список частих елементів.

Створення коріння з FP-дерева, T, і позначити його як "нуль". Для кожної транзакції в БД Trans потрібно виконати наступні дії:

Використовуючи цей алгоритм, FP-Дерево будується в двох сканувань бази даних. Перше сканування збирає і сортує безліч частих елементів, а другий будує FP-Дерево. Після побудови FP-Дерево це можливо добувати його , щоб знайти повний набір частих шаблонів [45].

Отже, необхідно:

- для кожного елемента побудувати його умовний шаблон-основу, а потім його умовне дерево;
- повторити цей процес на кожній новоствореній умові дерева;
- до тих пір поки результуюче FP-дерево порожнє або містить тільки один шлях;

Основними кроками до FP-дерева є:

1. Побудува умовного шаблону для кожного вузла в FP-дереві;
2. Побудова умовного FP-дерева з кожного умовного шаблону бази;
3. Побудова рекурсивного умовного дерева і використання частих шаблонів, отриманих до певного часу. Тоді якщо умовне FP-Дерево містить єдиний шлях - просто перелічити всі структури

Крок 1: З FP-дерева до умовного шаблону бази (Рисунок 2.2). Тоді маємо умовний шаблон бази:

Елемент	умов_шабл_бази
c	f:3
a	fc:3
b	fca:1, f:1, c:1
m	fca:2, fcab:1
p	fcam:2, cb:1

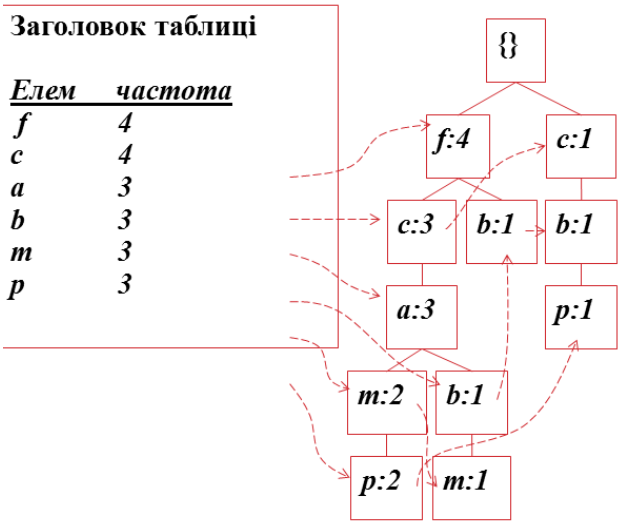

Рисунок 2.2 - Умоний шаблон бази

Наступним кроком є побудува умовного FP-дерева.

Для кожного шаблону-бази необхідно:

- Накопичити лічильник для кожного елемента в базі.
- Побудувати FP-Дерево для частих елементів шаблону бази.

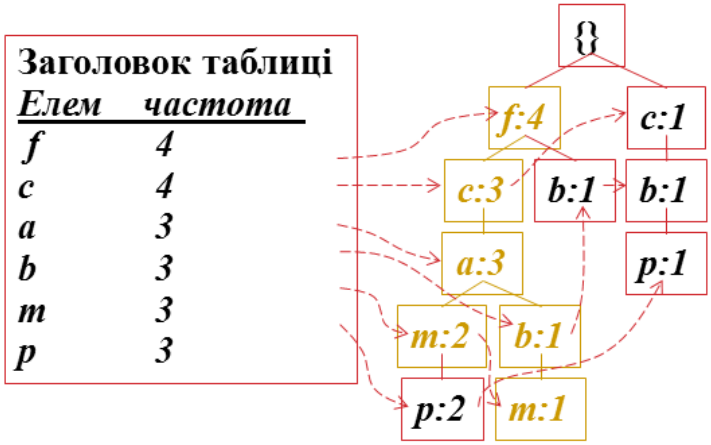

Рисунок 2.3 FP-Дерево для частих елементів

Тоді маємо кінцевий шаблон бази (Рисунок 2.4).

Рисунок 2.4 – Шаблони бази

Отримуємо видобуток шалонів які часто зустрічаються для створення умовного шаблону бази (Рисунок 2.5).

Елем	Умовний шаблон бази	Умовне FP-дерево
p	{(fcam:2), (cb:1)}	{(c:3)}|p
m	{(fca:2), (fcab:1)}	{(f:3, c:3, a:3)}|m
b	{(fca:1), (f:1), (c:1)}	Порожньо
a	{(fc:3)}	{(f:3, c:3)}|a
c	{(f:3)}	{(f:3)}|c
f	Порожньо	Порожньо

Рисунок 2.5 – Умовний шаблон бази

Крок 3: Рекурсивне умовне FP-Дерево.

І одне FP-Дерево шляхом генерації, тоді припустимо, що FP-Дерево T має єдиний шлях P. Повний комплект частою картиною T може бути згенерований шляхом перебору всіх комбінацій під-шляху з P . Отже, всі часті шаблони щодо m є m; fm, cm, am; fcm, fam, cam; fcam.

Отже, FP-growth на порядок швидше, ніж Apriori, і також швидше, ніж Дерево-проекція. Міркування такі:

- Немає генерування кандидатів, немає тестування кандидатів.
- Використовує компактну структуру даних.
- Виключені повторні перевірки , перевірки БД.
- Основні операції підрахунку і побудова FP-Дерево.

На рисунку 2.6 показано співвідношення для порівняння алгоритму FPGrowth і Apriori: Масштабування за підтримки порогу та часу виконання алгоритмів.

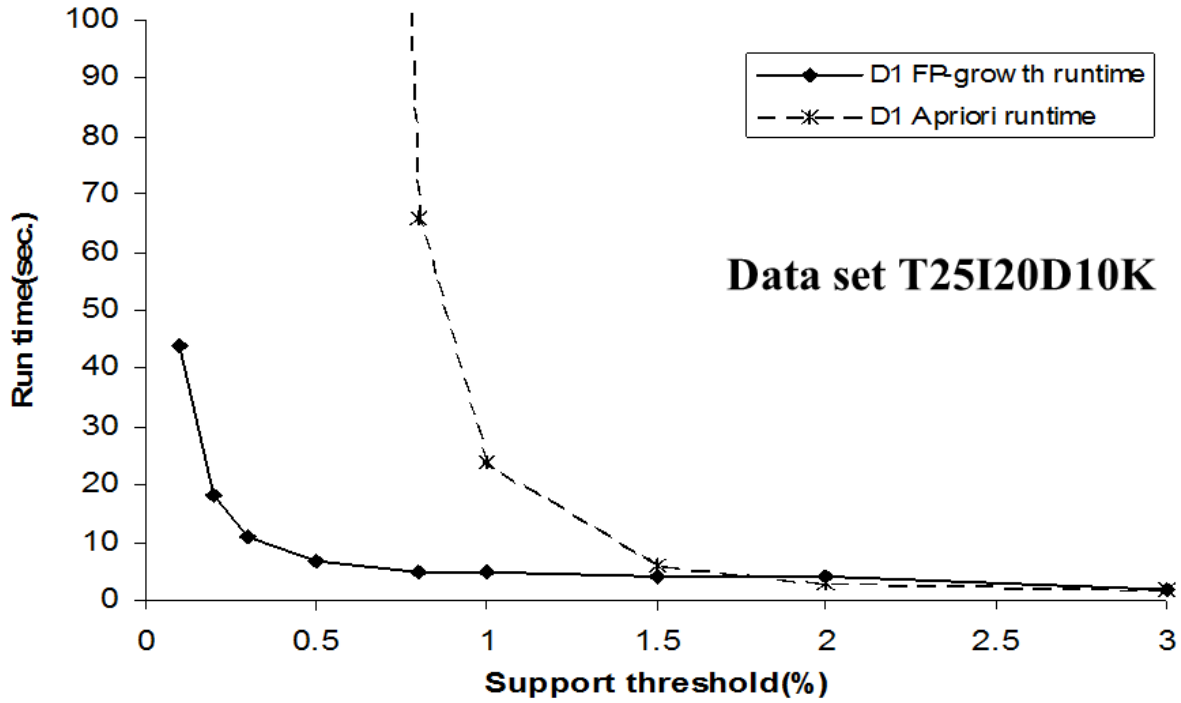

Рисунок 2.6 – Співвідношення алгоритмів FP-growth і Apriori

Отже, FP-growth на порядок швидший та працює ефективніше на всіх порогах підтримки [46].

2.3 Послідовні алгоритми для видобутку асоціативних правил

Алгоритм Generalized Sequential Patterns (GSP) - запропоновано P. Srikant і P. Агравал в IBM Almaden, в 1996 році. Користувачам часто потрібно вказати максимальні або мінімальні тимчасові проміжки між сусідніми елементами послідовної схеми. Наприклад, послідовність має сенс тільки тоді, коли сусідні елементи відбуваються в межах заданого інтервал часу, наприклад, через два місяці.

GSP алгоритм є алгоритмом який використовується для видобутку послідовності. Алгоритми рішення задач інтелектуального аналізу послідовності в основному базуються на апріорному алгоритмі (рівень-навхрест). Один з способів використовувати рівень-навхрест парадигма спочатку виявити всі часті елементи в рівним способом. Це означає , вважаючи входження всіх одноелементних елементів в базі даних.

Потім операції фільтруються шляхом видалення без частих предметів. В кінці цього етапу, кожна транзакція складається тільки з частих елементів. Ця модифікована база даних стає вхідним алгоритму GSP. Цей процес вимагає один прохід по всій базі даних [47].

Алгоритм GSP робить кілька проходів бази даних. При першому проході, все поодинокі елементи (1-послідовності), підраховуються. Через часті елементи, набір кандидатів 2-послідовностей сформовані, і інший прохід виконується, щоб визначити їх частоту. Часті 2-послідовності використовуються для генерації кандидатів на 3-послідовності, і цей процес повторюється до тих пір, не частіше послідовності не буде знайдено. Є два основні кроки в алгоритмі.

- Кандидат покоління. Беручи до уваги безліч частих (K-1) - частих послідовностей F (K-1), кандидати на наступному проході генеруються шляхом приєднання F (K-1) з самим собою. Фазова обрізку виключає будь-яку послідовність, щонайменше, один з яких підпослідовності є не частим [48].

- Підтримка підрахунку голосів. Як правило, хеш - дерево засноване на пошуку і використовується для ефективного підрахунку підтримки. Потім немаксимальні часті послідовності видаляються.

Формування кандидата встановлює Ck (набір кандидатів до-послідовностей). Нижче представлений псевдокод алгоритму.

Для всіх вхідних послідовностей s в базі даних D

do

Збільшення підрахунку всіх a в Ck, якщо s підтримує

Fk = {a ∈ Ck таким чином, що його частота перевищує поріг}

k = k+1;

Результат = Набір всіх частих послідовностей об'єднання всіх Fks

End do

End do

Наведений вище алгоритм виглядає як алгоритм Apriori. Основна відмінність, проте , генерування наборів - кандидатів. Припустимо , що:

$$A \rightarrow B \; i \; A \rightarrow C, \tag{2.6}$$

дві часті 2-послідовності. Елементи, які беруть участь в цих послідовностей *(A, B) i (A, C)* відповідно. Покоління кандидат в звичайному стилі Apriori дав би *(A, B, C)* в якості 3-НІЯКИХ гарантій, але в даному контексті ми отримуємо наступні 3-послідовності в результаті приєднання вище 2 послідовностей

$$A \rightarrow B \rightarrow C, \; A \rightarrow C \rightarrow B \; i \; A \rightarrow BC. \tag{2.7}$$

Алгоритм GSP виявляє часті послідовності, що дозволяє тимчасові обмеження, таких як максимальний проміжок і мінімальний проміжок між елементами послідовності. Крім того, він підтримує поняття ковзаючого вікна, тобто тимчасового інтервалу, протягом якого елементи спостерігаються як що належать одному і події, навіть якщо вони відбуваються з різних подій [49].

Алгоритм Tertius представлений в Flach & Lachiche (2001) - використовується для визначення, наскільки сильна підтримується правилом і новизна правила. Крім того цей підтвердження може бути пов'язане з алгоритмом пошуку генерування цікавих асоціативних правил (контрольовані і неконтрольовані) з наборів даних з номінальним атрибутами.

Правило видобування Tertius раніше була реалізована для машинного навчання бібліотеки Weka (Deltour, 2001). Така реалізація дозволяє легко використовувати Tertius на будь-якому наборі даних, який знаходиться в сумісному форматі Weka (наприклад, ARFF файлів). Проте, існує розрив між алгоритмом Tertius обговорюється в Flach & Lachiche (2001) і реалізації Weka. У орієнтаціє йде на рівняннях і параметри, використовуваних для генерації трьох значень, пов'язаних з кожним правилом об'єднання: (1) підтвердження, (2) істинно-позитивним, і (3) помилково-позитивним.

Алгоритм Tertius будує правила з значень пари атрибутів в навчальних даних і ранжирує їх відповідно до того, як вони надійні - тобто скільки разів правило справедливі і в навчальних даних.

Правило складається з тіла і голови. Тіло містить умови (відомі як літерали), необхідні для правила проведення, і може складатися з будь-якої кількості літер. Голова містить подія, яке відбувається, якщо правила справедливі. Під час навчання правило, Tertius починається з порожнього правила - той, який містить порожній корпус і порожньою головою. Правило потім уточнена шляхом додавання пари атрибут-значення в тому порядку, в якому вони з'являються в наборі даних. Як тільки це буде зроблено, то алгоритм підраховує, скільки разів правило справедливо (як тіло і голова істинні), і часи, коли правило дає помилковий позитивний результат (коли тіло вірно, але голова помилково).

Одним з недоліків Tertius є його відносно тривалий час роботи, яке в значній мірі залежить від кількості літер в правилах. Збільшення допустима кількість літералів збільшує час роботи в геометричній прогресії, тому ми хочемо, щоб зберегти максимум до трьох. Навіть при максимально дозволеного

трьох літер, Виконавча ще досить довго - працює Tertius може зайняти до декількох годин, для деяких з наших великих випробувань.

Реалізація Weka з Tertius виводить три окремі значення з кожним правилом асоціації: (1) значення підтвердження, (2) істинно-позитивним, і (3) помилково-позитивний. З цих трьох значень, значення підтвердження правила є найбільш важливим, оскільки він вимірює, наскільки незвично / цікаві правила. Решта два значення включені для вимірювання значення правил асоціації. Вони будуть пояснені більш детально нижче в цьому розділі.

По-перше, ми дивимося на чотирьох параметрів, спільних для всіх функцій:

- body_counter= кількість точок, де атрибути відповідають правилам;
- head_counter= кількість точок, де головний атрибут не відповідає правилу;
- m_counter= кількість точок, де тіло атрибутів відповідає правилу, але головний атрибут ні;
- instances=кількість точок.

Тоді як, m_counter відноситься до фактичного лічильника. body_counter, відноситься до першої рядку в таблиці спряженості, в той час як head_counter відноситься до другій колони . Для Tertius цілей, всі чотири параметра беруть участь в генерації вихідних значень [50].

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ АЛГОРИТМІВ ПОШУКУ АСОЦІАТИВНИХ ПРАВИЛ

3.1 Програмно-апаратні засоби для проведення експериментів

Експериментальні дослідження були проведені на настільному ноутбуці ACER Travelmate P253. Основні технічні характеристики апаратного комплексу наведені у таблиці 3.1.

Таблиця 3.1 – Технічні характеристики ноутбука ACER Travelmate P253

Технічні характеристики	Опис
Екран	15.6" (1366x768) WXGA HD
Процесор	Двоядерний Intel Core i3-3100M (2.4 ГГц)
Тип оперативної пам'яті	DDR3 1333 МГц , 6 Gb
Чіпсет	Intel HM77
Інтегрований графічний адаптер	Intel HD Graphics 4000
Графічний адаптер	nVidia GeForce 710M, 2 ГБ
Обсяг накопичувача	HDD 500 ГБ
Мережеві адаптери	Wi-Fi 802.11 B / G / N, Bluetooth 4.0, Ethernet (10/100)
Роз'єми і порти введення-виведення	2 порта USB 2.0, 1 порт USB 3.0 / VGA / HDMI / LAN (RJ-45)

Встановлена операційна система Windows 7 Ultimate, розрядністю 64 біти та графічний пристрій із підтримкою DirectX 9 і драйвером WDDM 1.0.

Програмним засобом для проведеня експерименту є WEKA. Це комп'ютерна програма для автоматичного аналізу великих наборів даних та пошуку в них закономірностей. Ця інформація може бути використана для

автоматичного складання прогнозів або щоб допомогти приймати рішення швидше і точніше.

В таблиці 3.2 наведено основну інформації про використовуваний програмний засіб WEKA.

Таблиця 3.2 – Відомості про WEKA

Розробник	Університет Уайкато
Версія продукту	3.6.11
Написано на	Java
Платформа	Багатоплатформеність
Доступні мови	Англійська
Тип	Машинне навчання
Ліцензія	GPL
Веб-сайт	www.cs.waikato.ac.nz/~ml/weka/

Для роботи з WEKA на робочій станції повинно бути інстальовано інтегроване середовище розробки – Java Virtual Machine, версії 1.7 від компанії Oracle.

Це ПЗ написано на мові Java™ і забезпечує графічний користувальницький інтерфейс для роботи з файлами даних і генерації візуальних результатів (у вигляді таблиць і графіків). Крім того, ми можемо інтегрувати WEKA, як і будь-яку іншу бібліотеку, в свої власні додатки, наприклад, для автоматизації аналізу даних на стороні сервера, використовуючи стандартний API.

Weka містить інструменти для попередньої обробки даних, класифікації, регресії, кластеризація, правила асоціації та візуалізації. Він також добре підходить для розробки нових схем машинного навчання.

Weka являє собою набір алгоритмів машинного навчання для інтелектуального аналізу даних завдань. Алгоритми можуть бути застосовані безпосередньо до набору даних або викликані з вашого власного коду Java.

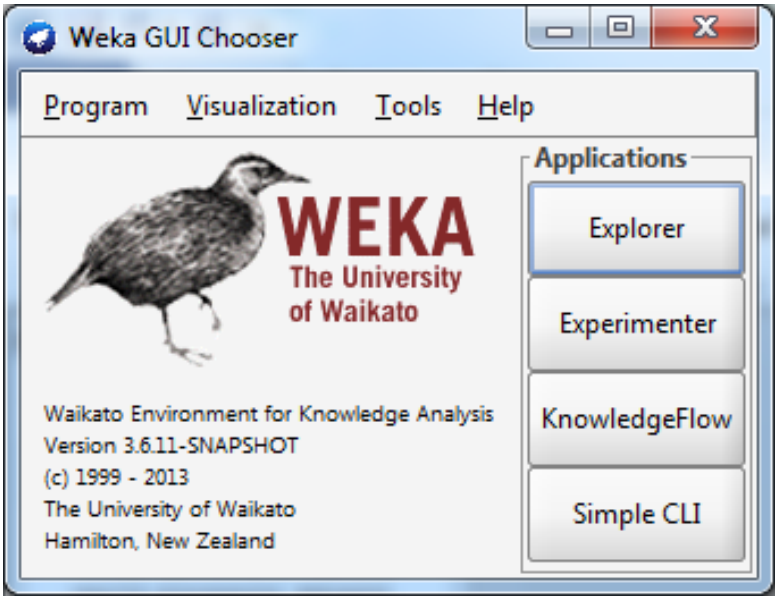

Рисунок 3.1 – Головне вікно програми

Переваги WEKA:

- Ліцензія GPL.
- Портативність завдяки реалізації на Java.
- Простота використання завдяки своєму графічному інтерфейсу.
- Всеосяжний збір даних для попередньої обробки і моделювання.

.

3.2 Узагальнений алгоритм пошуку асоціативних правил

Розроблений узагальнений алгоритм включає в себе:

- підготовку файлу до роботи в середовищі WEKA – формування файлу, завантаження даних;
- попередню обробку даних файлу – застосування фільтру;
- класифікацію даних (побудова моделі) - класифікація мікрооб’єктів алгоритмами класифікації та побудова дерева рішень;
- пошук асоціативних правил використовуючи різні алгоритми – застосування алгоритмів пошуку до наявної тестової бази;

- виділення асоціативних правил – аналіз отриманих результатів.

Для того щоб завантажити дані в WEKA, їх слід перетворити у формат, зрозумілий цього пакета програмного забезпечення. Найбільш підходящим форматом для завантаження даних в WEKA є формат Attribute-Relation File Format (ARFF), який спочатку визначає тип завантажуваних даних, а потім вказує власне дані.

На першому кроці необхідно перевести таблицю csv, що містить дані, у формат arff і модифікувати її.

Модифікація полягає в додаванні полів метаданих: в початок файлу на окремих стрічках назви залежності @relation ім'я, описи атрибутів @attribute ім'я тип і @data перед початком самих даних. Типи даних наступні: чисельні (numeric, real, integer), такі, що перераховують (задаються перерахуванням виду {i1, .., in}), строкові (string), дата (date [date format]).

Приклад файлу файлу на завантаження у WEKA виглядає на рисунку 3.2.

Відповідно до файлу, у нас 12 залежних змінних (атрибутів):

- Діагноз_Гістологія.
- Гістологічні характеристики_Властивість до апокринної секреції.
- Гістологічні характеристики_Множинні вогнища проліферації.
- Гістологічні характеристики_Некрози.
- Гістологічні характеристики_Підвищена проліферативна здатність внутрішньопротокового епітелію.
- Клітина_Форма.
- Клітина_Інтенсивініть апокриннової секреції.
- Ядро_Хроматин.
- Угрупування клітин або клітинні комплекси_Розташування.
- ЯЦВ_Частка ядра.
- Клітина_Вплив на розвиток колагенових волокон.
- Угрупування клітин або клітинні комплекси_Форма.

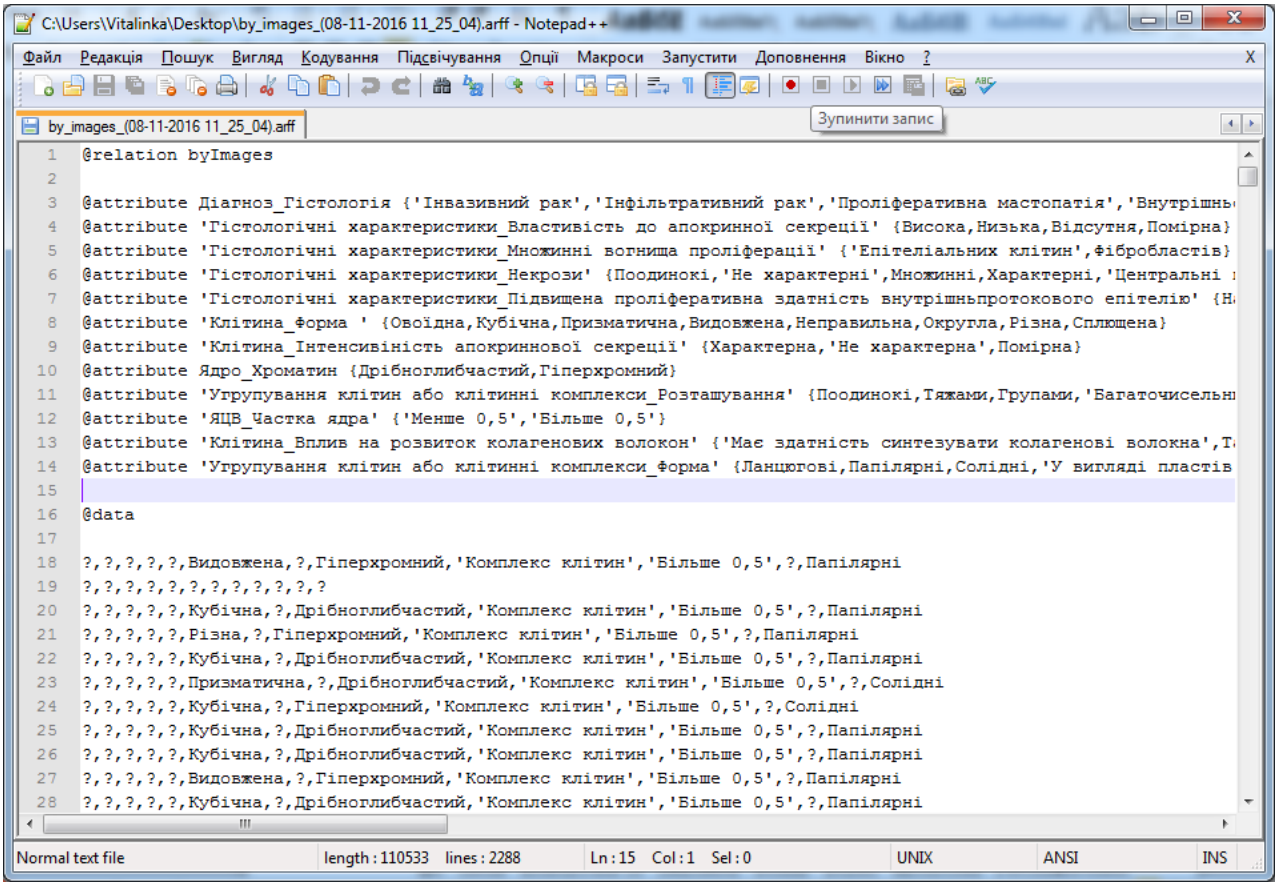

Рисунок 3.2 – Приклад файлу на завантаження у WEKA

При запуску WEKA, пакет пропонує на вибір 4 графічних інтерфейси. Для всіх прикладів використовується опція Explorer. Її функціональних можливостей цілком достатньо для вирішення наших завдань. На другому кроці файл з даними потрібно завантажити в WEKA. В результаті відкриється закладка Preprocess вікна Explorer, потім натискаємо на кнопку Open File і вибраємо ARFF-файл.

На цій вкладці можна завантажити файл в систему, а потім редагувати завантажені дані. Редагування може здійснюватися як вручну.

Для автоматичної обробки даних використовуємо фільтри для очищення і/або трансформації завантажених даних.

Фільтри діляться на два типи - ті, застосування яких до даних може викликати відхилення (тобто фактично ці фільтри вимагають вже наявності якихось знань, отриманих від застосованого алгоритму навчання), і ті, які

можна застосовувати до ще необроблених даних (unsupervised). Для простоти застосовуватимемо фільтри unsupervised.

Нйбільш підходящими фільтрами є:

- RemoveType, Remove - для видалення певних атрибутів, у тому числі і за типом – цей фільтр корисний , оскільки не усі типи можуть бути використані в різних алгоритмах.
- Disctretize - для перетворення числового атрибуту на дискретний;
- RemoveUseless - видаляє атрибути, які не змінюються взагалі або які змінюються занадто часто. Постійні атрибути автоматично видаляються, і ті, що перевищують максимальний відсоток параметра дисперсії. Максимальний параметр дисперсії застосовується тільки до номінальних атрибутів.
- ReplaceMissingValues - для заміни відсутніх (пропущених) значень середніми по атрибуту;

У вхідному наборі алгоритм вимагає тільки номінальні значення змінних, а також, щоб не було пропущених значень. Для цього застосовуємо фільтр RemoveType і видаляємо усі типи даних окрім nominal.

Після застосування фільтру в наборі залишаються тільки дані номінального типу. З ними і продовжує роботу алгоритм. Далі для роботи алгоритму потрібна відсутність порожніх значень. Щоб здійснити це застосовується фільтр ReplaceMissingValues, замінюючий порожні значення середніми.

Якщо видалити усі рядки зі значенням null то можна отримати невеликий набір правил, по яких можна класифікувати об'єкти.

Приклад виведення програми зображений на рисунку 3.3.

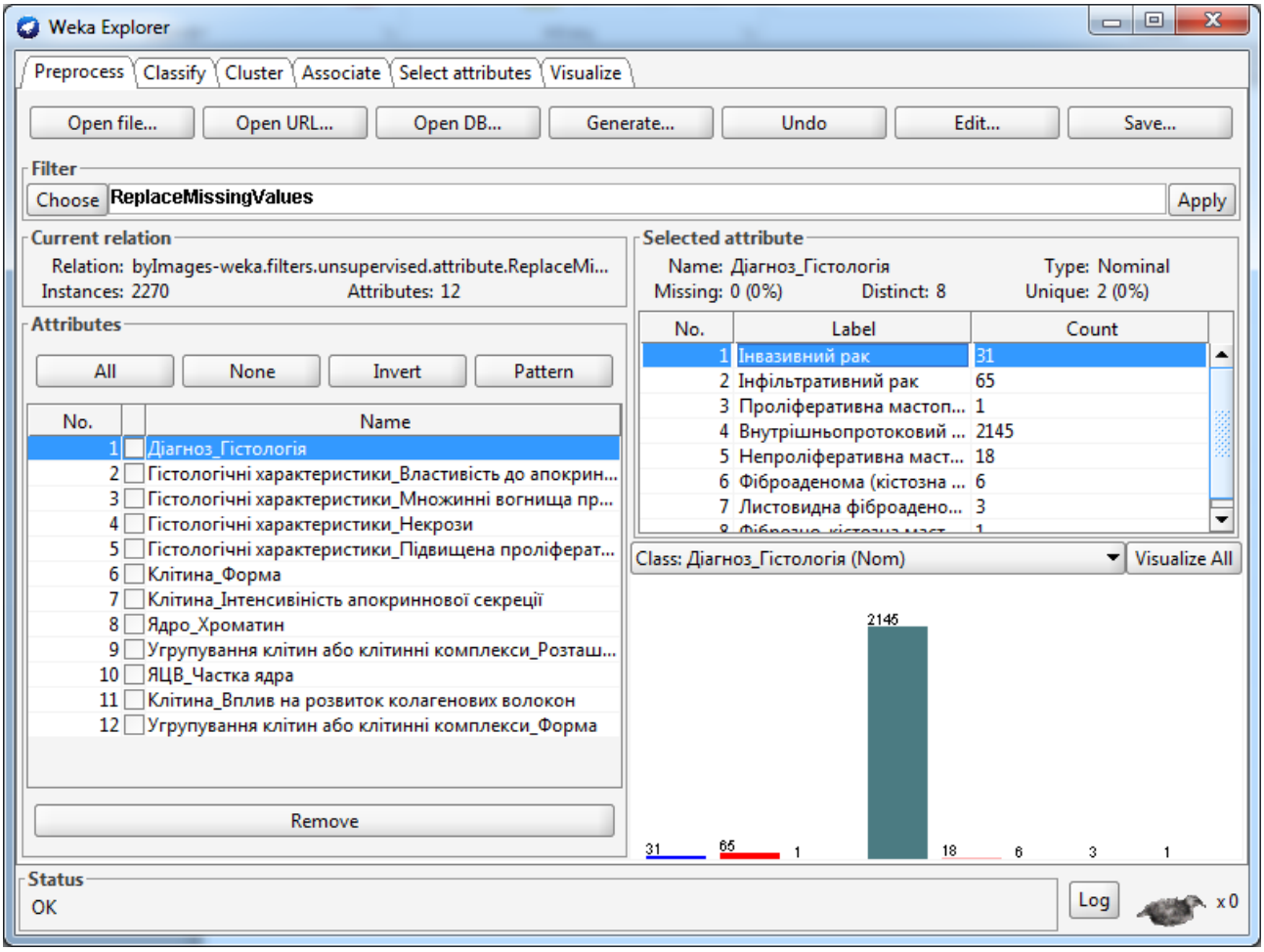

Рисунок 3.3 – Результат застосування фільтру ReplaceMissingValues

Після етапу попередньої обробки даних варто провести класифікацію вхідних екземплярів. Загальна кількість екземплярів на тестовій базі біомедичних зображень становить 257.

У вкладці Classify за допомогою кнопки Choose вибирається метод класифікації. Після вибору методу класифікації (класифікатора, classifier) необхідно вибрати метод перевірки. Основним методом є крос-перевірка (cross - validation), суть її в тому, що початковий набір даних в якій-небудь пропорції розбивається на навчальну і тестову множини (вибірки). Далі на навчальній множині дані класифікуються, а на тестовій перевіряються. Таким чином обчислюється помилка класифікації.

Після цього обирається залежна змінна класифікації. В даному випадку було обрано «Діагноз гістологія».

Потім натискається кнопка Start. Після закінчення аналізу заповниться вікно Output і додасться новий запис у вікно Result.

Для порівняння будемо класифікувати початкові дані наступними алгоритмами:

1.Naive Bayes - наївний байесовским метод, досить прозорий і зрозумілий алгоритм класифікації. "Наївною" вона називається тому, що виходить з припущення про взаємну незалежність ознак. Ідея алгоритму полягає в тому, що формуються правила, в умовних частинах яких порівнюються усі незалежні змінні з відповідними можливими значеннями.

2.Метод ZeroK - Клас для створення і використання 0-R класифікатора. Показує середнє значення (для числового класу) або режим (для номінального класу

3. Метод J48 (модифікація C4.5)- Цей алгоритм також застосовується до початкових даних без їх зміни. Результатом його роботи є Дерево рішень.

4.Метод 1R - один з найпростіших і зрозуміліших методів класифікації. Застосовується як до числових даних, які розбиваються на проміжки, так і до даних типу nominal.

5. Метод SVM (у середовищі Weka він називається SMO). Для цього методу не вимагається яких-небудь перетворень початкової вибірки.

Цей метод є одним з найточніших, завдяки досконалості алгоритму. Також цей алгоритм складний для розуміння, оскільки даними оперується в n - мерном просторі, складному для представлення людиною.

В таблиці 3.2 наведено результати класифікації мікроб’єктів на навчальних та тестових вибірках.

З порівняння методів видно, що з більшою точністю об'єкти класифікують алгоритми NAIVE BAYES, J48 і SVM (SMO). SVM (SMO) точніший, але складний для читання інформації.

Таблиця 3.2 - Порівняльна таблиця результатів різних методів класифікації на двох вибірках

Метод Класифікації	Навчальна вибірка		Тестова вибірка	
	Точність, %	Помилка,%	Точність,%	Помилка, %
NAIVE BAYES	95.3307	4.6693	95.3307	4.6693
J48	98.4436	1.5564	98.4436	1.5564
ZeroK	51.3619	48.6381	51.3619	48.6381
1R(OneR)	78.5992	21.4008	78.5992	21.4008
SVM (SMO)	100	0	100	0

Найкращий результат класифікації показав алгоритм J48 (вибірка – крос-перевірка). Результат класифікації алгоритмом J48 в середовищі WEKA представлений на рисунку 3.4.

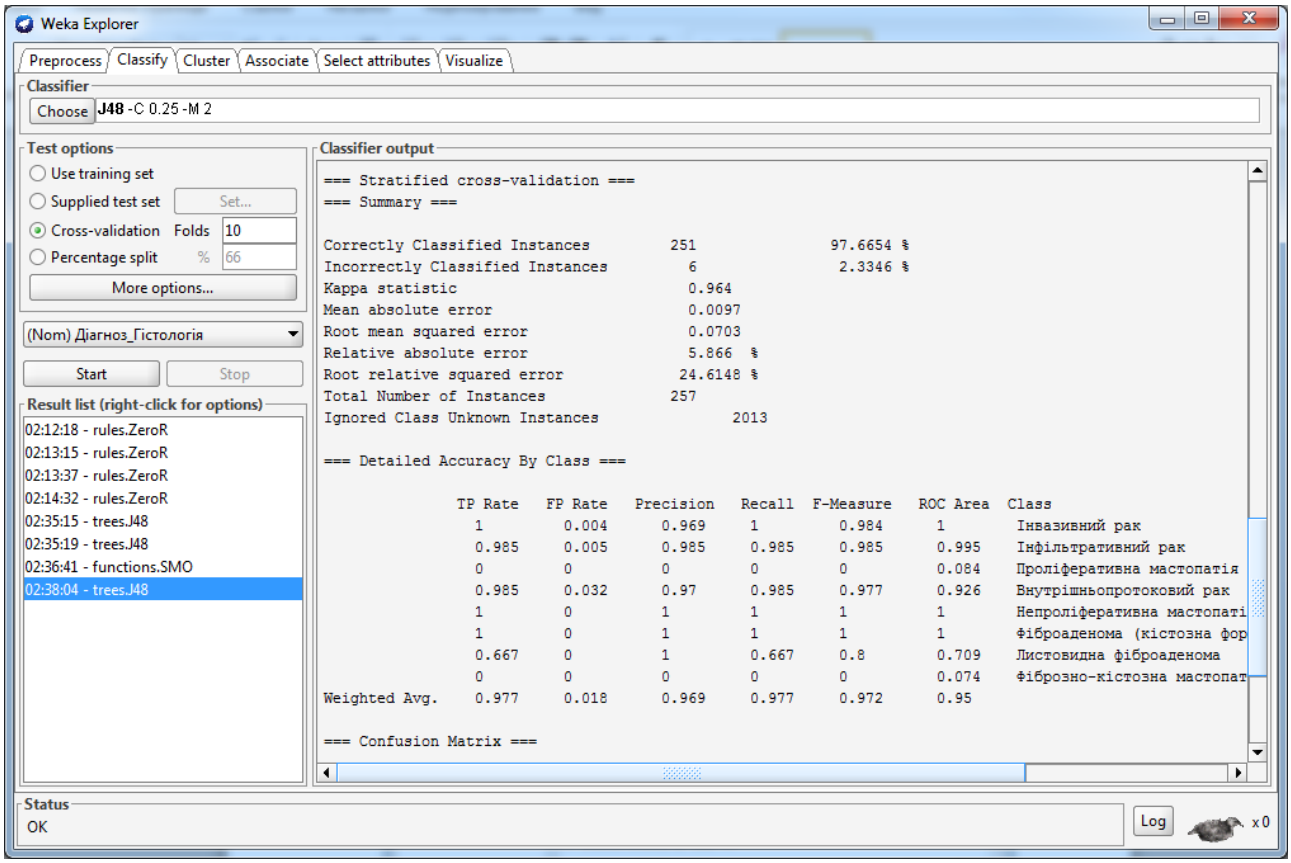

Рисунок 3.4 – Результат класифікації алгоритмом J48

Найбільш суттєві дані - це показники класифікації "Correctly Classified Instances" і "Incorrectly Classified Instances". Крім того, слід звернути увагу на число в першому рядку стовпця ROC Area (1). Оскільки показник точності нашої моделі - 98.4436 %, то в первісному розгляді її не можна назвати досить хорошою.

Варто зазначити, що правильно прокласифіковані екземпляри складають близько 98.4436%, тобто 251 з 257 загальної кількості.

Побудовано модель Дерево рішень до алгоритму J48 для вибраних атрибутів таких як :

- діагноз_гістологія;
- ЯЦВ_частка ядра;
- клінина_вплив на розвиток колагенових волокон;
- угрупуваня клітин або Клітинні комплекси_Форма.

Дане дерево рішень представлено в додатку Б. Аналіз про класифікованих екземплярів представлено в таблиці 3.3.

Таблиця 3.3 - Аналіз прокласифікованих екземплярів

	Оцінка	Швидкість	Точність	Відклик	F-Міра	ROC-область	Клас,події
	1	0,004	0,969	1	0,984	1	Інвазивний рак
	0,985	0,005	0,985	0.985	0,985	0,994	Інфільтративний рак
	0	0	0	0	0	0,084	Проліферативна мастопатія
	0,985	0,032	0,97	0.985	0,977	0,926	Внутрішньопротоковий рак
	1	0	1	1	1	1	Непроліферативна мастопатія
	1	0	1	1	1	1	Фіброаденома (кістозна форма)
	0,667	0	1	0.667	0,8	0,709	Листовидна фіброаденома
	0	0	0	0	0	0,074	Фіброзно-кістозна мастопатія
Середньозважені	0,977	0,018	0,969	0,977	0,972	0,95	

Пошук асоціативних правил. На вкладці Associate вибирається метод знаходження, для нього виставляються параметри кліком на його назві, після чого натискається кнопка Start і аналізується вивід (перед початком використання методу Апріорі необхідно застосувати фільтр RemoveType і видалити numeric -атрибути). У даному випадку асоціативні правила будуються використовуючи всі доступні алгоритми.

Пошук асоціативних правил здійснено на основі алгоритмів Apriori, Filtered Associator, Predective Apriori та Tertius. В початковому файлі 2270 приміриків.

При зміні метрики правила міняються повністю. Метрики це міра, що дозволяє отримати числове значення деяких властивостей програмного забезпечення та його специфікацій. По замовчуванню метрики встановлені програмою, але для випробування можна змінювати ці значення на будь-які тестові. Це дає змогу детально дослідити поведінку і результати експериментів.

Алгоритму Apriori, для якого основними характеристиками є значення мінімальної підтримки та мінімальної достовірності. Кількість примірників становить 341.

У налаштуваннях методу по замовчуванню встановлювалося створення 10 асоціативних правил. Цей алгоритм визначає набори, що часто зустрічаються, відповідно найточнішими є самі набори, що часто зустрічаються. Виводяться набори з мінімальною метрикою. На рисунку 3.5 показані метрики для алгоритму Apriori.

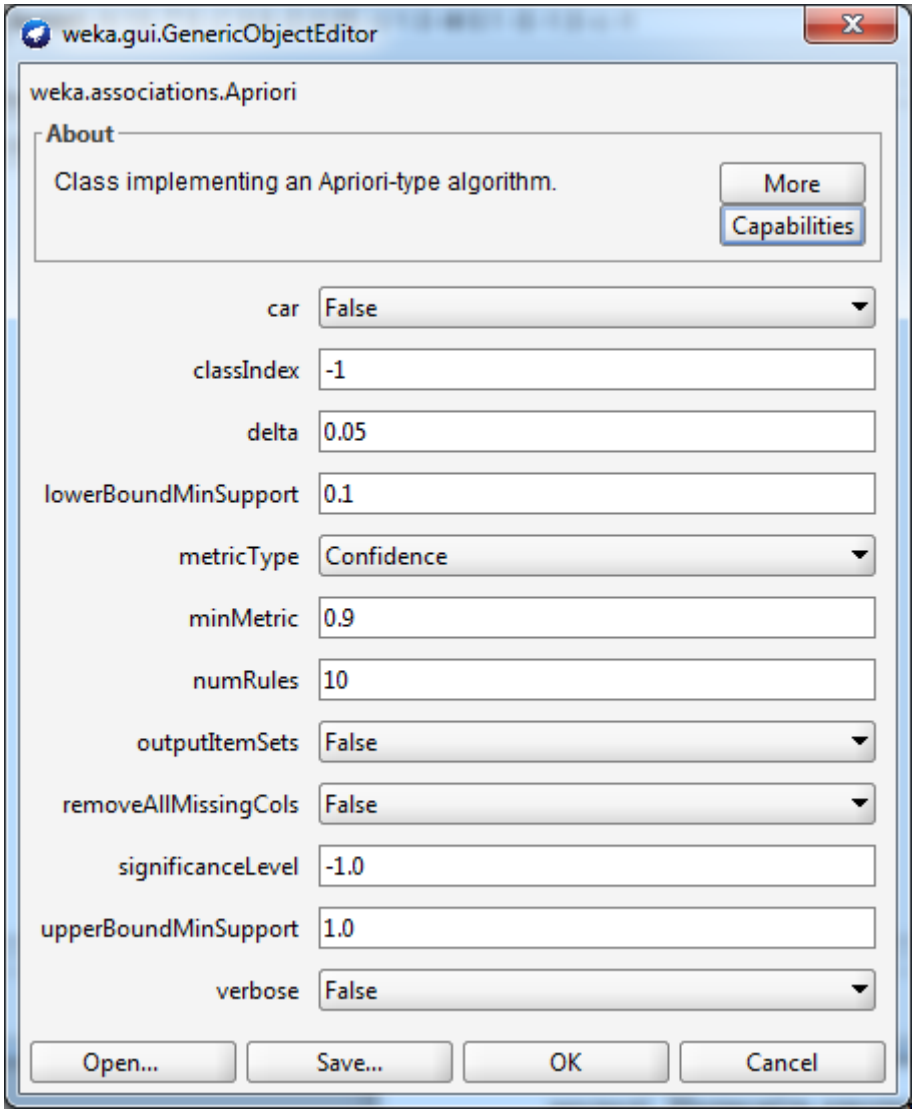

Рисунок 3.5 - Метрики для алгоритму Apriori

Налаштування алгоритму Apriori:

- car – (true/false) - Якщо включений клас асоціативних правил клас здобуті видобуваються замість загальних асоціативних правил.

- classIndex – індекс атрибута класу. Якщо встановлено значення -1, останній атрибут береться в якості атрибута класу.

- delta – ітеративно зменшити підтримку. Знижує підтримку до хв підтримки не буде досягнуто або необхідну кількість правил було згенеровано.

- lowerBoundMinSupport - нижня межа мінімальної підтримки.

- metricType - Встановити тип метрики для ранжирування правил.

- minMetric – Мінімальний бал метрики. Розглянемо тільки правила з оцінки вище цього значення.

- numRules – Кількість правил для знаходження.

- outputItemSets – Якщо включено наборів виводяться так само.

- removeAllMissingCols – видалити колонки з усіха відсутніми значеннями.

- significanceLevel – рівень значущості. Критерій значущості

- upperBoundMinSupport – Верхня межа для мінімальної підтримки.

- verbose – Якщо включений алгоритм буде працювати в розширеному режимі.

Після визначення усіх параметрів можна приступити до створення моделі. Натисніть кнопку Start. Результат представлений на рисунку 3.6.

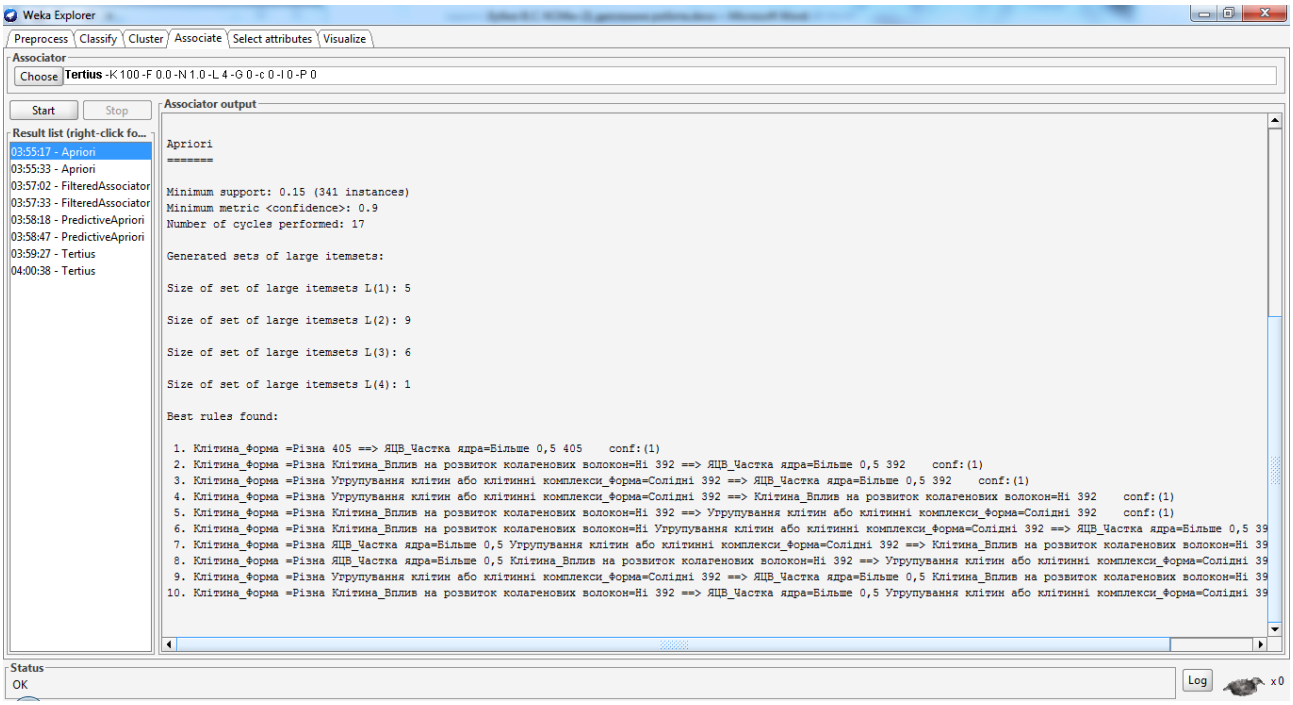

Рисунок 3.6 – Результат виводу алгоритму Apriori

Для порівняння було встановлено 100 правил для виводу. Швидкість для обрахування моделі не змінилась. Повні правила наведені нижче.

Клітина_Форма =Різна Угрупування клітин або клітинні комплекси _Форма=Солідні 392 ==> Клітина_Вплив на розвиток колагенових волокон=Ні 392 conf:(1)

Клітина_Форма =Різна ЯЦВ_Частка ядра=Більше 0,5 Клітина_Вплив на розвиток колагенових волокон=Ні 392 ==> Угрупування клітин або клітинні комплекси_Форма=Солідні 392 conf:(1)

Наступним алгоритмом було вибрано Filtered Associator на 10 та 100 правилах. Додатково було застосовано MultiFilter. В якості асоціатора був вибаний метод FilteredAssociator. Налаштування представлені на рисунку 3.7.

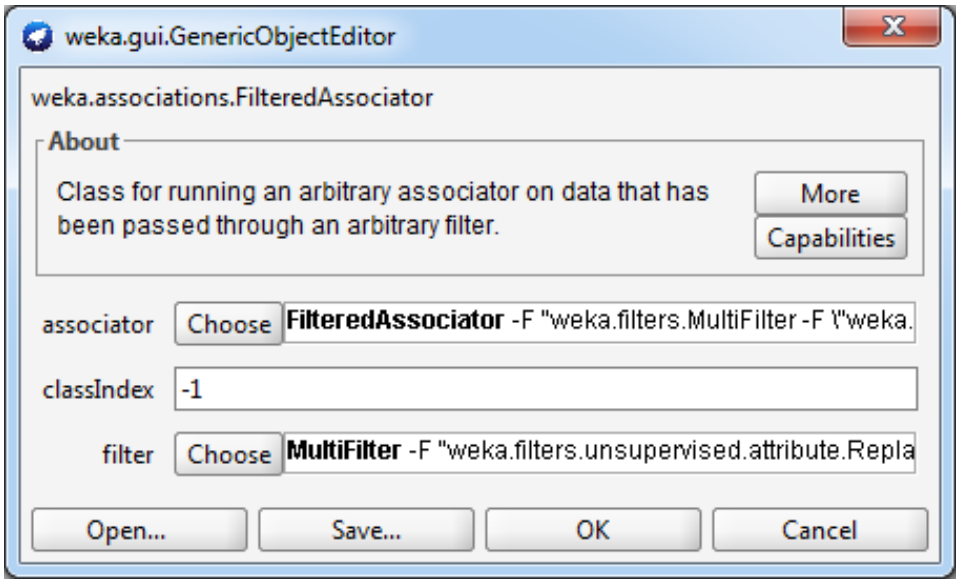

Рисунок 3.7 – Налаштування для алгоритму FilteredAssociator

Основними характеристиками при виведенні моделі є значення мінімальної підтримки та мінімальної достовірності. Кількість примірників значно більша ніж Apriori, і вона становить 2157 .

Отримані асоціативні правила використовуючи алгоритм FilteredAssociator:

Гістологічні характеристики_Підвищена проліферативна здатність внутрішньпротокового епітелію=Відсутня 2260 ==> Гістологічні характеристики_Множинні вогнища проліферації=Фібробластів 2260 conf:(1)

Гістологічні характеристики_Властивість до апокринної секреції=Відсутня 2243 ==> Клітина_Інтенсивініcть апокриннової секреції=Помірна 2243 conf:(1)

Відповідні налаштування для проведення експериментів було здійснено використовуючи алгоритми PredectiveApriori та Tertius. Ці алгоритми займають на порядок більше часу для сформування моделі та асоціативних правил. Основних характеристик при виведенні моделі немає, проводяться налаштування перед запуском обрахунку (використовуються налаштування по замовчуваню).

Приклад отриманих асоціативних правил використовуючи алгоритм PredectiveApriori наведений нижче:

Діагноз_Гістологія=Внутрішньопротоковий рак 132 ==> Гістологічні характеристики_Властивість до апокринної секреції=Відсутня Угрупування клітин або клітинні комплекси_Форма=Солідні 132 acc:(0.99412).

Ядро_Хроматин=Гіперхромний Угрупування клітин або клітинні комплекси_Розташування=Багаточисельними округлими стурктурами 287 ==> ЯЦВ_Частка ядра=Більше 0,5 Угрупування клітин або клітинні комплекси_Форма=Солідні 287 acc:(0.99488).

На основі отриманих результатів можна отримати наступні приклади повних асоціативних правил передракових станів молочної залози, використовуючи алгоритм Tertuis:

/* 0,911736 0,001808 */ Діагноз_Цитологія = Папілярний рак ==> Клітина_Форма = Кубічна or Угрупування клітин або клітинні комплекси_Розташування = Багаточисельними округлими стурктурами or Угрупування клітин або клітинні комплекси_Форма = Папілярні.

/* 0,911736 0,001808 */ Діагноз_Цитологія = Папілярний рак ==> Клітина_Форма = Призматична or Угрупування клітин або клітинні комплекси_Розташування = Багаточисельними округлими стурктурами or Угрупування клітин або клітинні комплекси_Форма = Папілярні.

/*0,911736 0,001808 */ Діагноз_Цитологія = Папілярний рак ==> Угрупування клітин або клітинні комплекси_Розташування = Багаточисельними округлими стурктурами or Угрупування клітин або клітинні комплекси_Форма = Папілярні or Ядерце_Кількість = Одиничні дрібні ядерця.

Приклад неповного правила:

/* 0,930113 0,000000 */ Клітина_Колір = Насичений and Клітина_Цитоплазма = Насичена цитоплазмою ==> Угрупування клітин або клітинні комплекси_Розташування = Багаточисельними округлими стурктурами or Хроматин_Тип = Сітчастий

Аналіз результатів асоціативних правил буде представлено нижче.

3.3 Аналіз результатів проведених експериментальних досліджень

Для порівняння отриманих показників використаємо два алгоритми із відповідними основними характеристиками. Алгоритм Apriori та Filtered Associator мають однакові характеритиски. Порівняльна таблиця отриманих результатів для алгоритмів представлена у вигляді таблиці. Експеремнтальні дослідження проводилися на 10 та 100 правилах.

Таблиця 3.4 – Порівняльна таблиця отриманих показників при побудові моделей для пошуку асоціативних правил

Алгоритм (кількість правил)	Кількість примірників	Мінімальна підтримка	Мінімальна достовірність	Сформовані набори даних	Кількість виконуваних циклів
Apriori (10)	341	0.15	0.9	4	17
Apriori (10)	227	0.1	0.9	6	18
FilteredAssociator (10)	2157	0.95	0.9	5	1
FilteredAssociator (100)	2157	0.95	0.9	5	1

В таблиці 3.5 представлено порівняльну таблицю отриманих показників за критерієм час виконання.

Таблиця 3.5 – Порівняльна таблиця отриманих показників при побудові моделей для пошуку асоціатиних правил

Кількість правил	Час виконання знаходження правил, сек			
	Apriori	FilteredAssociator	PredectiveApriori	Tertius
10 правил	0,64 сек	0,74 сек	8,57 сек	18,5 сек
100 правил	1,16 сек	0,83 сек	13,2 сек	47,2 сек

З порівняльної таблиці видно, що алгоритми Apriori та FilteredAssociator є швидшими, тоді як PredectiveApriori та Tertius набагато довше виконуються, і це займає більше часу на видобуток асоціативних правил.

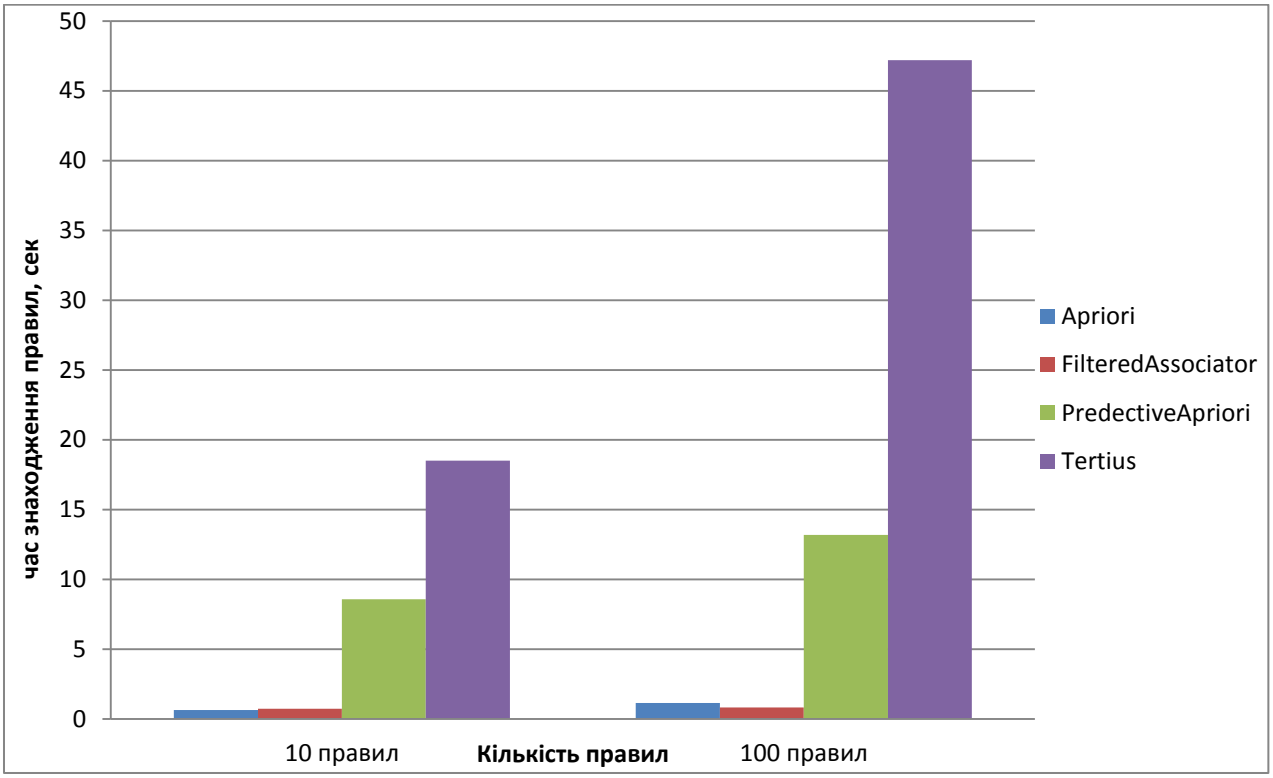

Рисунок 3.8 – Діаграма по кількості та часу знаходження правил

В таблиці 3.6 представлено порівняльну таблицю отриманих показників корисності правил.

Таблиця 3.6 – Порівняльна таблиця отриманих показників корисності правил

Кількість правил	Корисні правила/Неповні правила, кількість			
	Apriori	FilteredAssociator	PredectiveApriori	Tertius
10 правил	6/4	5/5	7/3	8/2
100 правил	83/17	6/4	67/33	77/23

З отриманих результатів можна зробити висновок що алгоритм Apriori та Tertius дають більшу кількість корисних (повних) правил.

В результаті дослідженя було проаналізовано результати експерементів та представлено дані в табличному вигляді.

ВИСНОВКИ

1.Проведено аналіз біомедичних зображень, описано основні характеритистикаи гістологічних та цитологічних зображень. Здійснено аналіз методів та алгоритмів інтелектуального аналіз даних, та класифікацію стадій антелектуального аналізу.

2. Також було проведено аналіз програмних засобів інтелектуального аналізу даних, наведено основні характеристики та практичне використання до певного кола задач.

3. Здійснено аналіз усіх існуючих алгоритмів для пошуку асоціативних правил, зокрема, алгоритмів які вбудовані в програмний засіб для побудови моделей та пошуку асоціативних правил.

4.Описано програмно-апартний засіб, тобто робочу станцію-ноутбук та програмне середовище для проведення експерементальних досліджень використовючи наявну базу біомедичних зображень.

5.Також розроблено узагальнений алгоритм для пошуку асоціативних правил, який включає в себе кілька основних кроків. Всі кроки були здійснені в середовиші WEKA, за початкові дані для опрацювання буде обрана наявна експерементальна база біомедичних зображень.

6.Проведено порівняльну роботу всіх алгоритмів та проаналізовано результати експерементального дослідження. В результаті було отримано набір асоціативних правил для діагностики передракових та ракових станів раку молочної залози та відповідні їм лінгвістичні змінні.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Березький О. Комп’ютерна система аналізу біомедичних зображень / О. Березький, Ю. Батько, Г. Мельник // Вісник Національного університету "Львівська політехніка". Комп'ютерні науки та інформаційні технології. – 2009 – с. 11-18.
2. Березький О.М. Інтелектуальна система для діагностування різних форм раку молочної залози на основі аналізу гістологічних та цитологічних зображень / О.М. Березький, Г.М. Мельник, Ю. М. Батько, Т. В. Дацко // Науковий вісник НЛТУ України - 2013. - № 23.13. - С. 357-367.
3. Вербовий С.О. Методи пошуку асоціативних правил в базі даних біомедичних зображень// Зубко В.С. / Матеріали конференції ACIT, Тернопіль, 2016.
4. Методичні рекомендації до виконання дипломної роботи з освітньо-кваліфікаційного рівня “Магістр”. Спеціальність „Комп’ютерні системи та мережі” / О.М. Березький, Л.О. Дубчак / Під ред. О.М. Березького – Тернопіль: ТНЕУ, 2013.– 47 с.
5. Кафедра комп’ютерної інженерії [Електронний ресурс] –Режим доступу до сайту http://www.tneu.edu.ua/faculty/fkit/department-ki/Опис кафедри комп’ютерної інженерії [Електронний ресурс] –Режим доступу до сайту http://tanet.tneu.org/about/kafedry/ki/opys .
6. Барсегян А.А.: Технологии анализа данных. Data Mining, Visual Mining, Text Mining, OLAP / М.С. Куприянов, В.В. Степаненко, И.И. Холод - БХВ-Петербург, 2007.
7. Дюк В.А., Data Mining: учебный курс. -СПб.:/ Самойленко А.П. Питер, 2001. - 368 с.
8. Bellman R Abstraction and pattern classification. /Kalaba R., Zadeh L. A.— Journal of Mathematical Analysis and Applications, 1996, v. 13,

9. Bezdek J. C. Numerical taxonomy with fuzzy sets.—Journal of Mathematical Biology, 2001, v. 1, p. 57—71.
10. Capocelli R., sets and decision theory. / De Luca A. Fuzzy — Information and Control, 1993, v. 23, p. 446—473.
11. Chang S. K. On the execution of fuzzy programs using finite state machines.— IEEE Transactions on Computers, 2002, v. C-21, p. 241—253.
12. De Luca A., A definition of a non-probabilistic entropy in the setting of fuzzy sets theory. / Termini S. — Information and Control, 1989, v. 20, p. 301—312.
13. De Luca A., Algebraic properties of fuzzy sets. / Termini S. — Journal of Mathematical Analysis and Applications, 1999, v. 40, p. 373—386.
14. De Luca A., Termini S. Entropy of L-fuzzy sets.— Information and Control, 1974, v. 24, p. 55—73.
15. De Luca A.,.On the convergence of entropy measures of fuzzy sets. / Termini S — Kybernetes, 2006, v. 6, p. 219—227.
16. De Luca A., Entropy and energy measures of a fuzzy set. / Termini S. — In: Advances in Fuzzy Set Theory and Applications/Ed. by M.M. Gupta, R.K. Ragade, R.R. Yager. Amsterdam: North-Holland, 1999, p. 321.
17. De Luca A. On some algebraic aspects of the measures of fuzziness./ Termini S. — In: Fuzzy Information and Decision Processes/Ed. by
18. Gupta M.M., Amsterdam: / E. Sanchez.North-Holland, 1992, p. 17—24.
19. DiNo1a A., On the fuzziness measure and negation in totally ordered lattices. / Sessa S. — BUSEFAL, 1981, v. 8, p. 68—77.
20. D i Nola A. Ordering via fuzzy entropy./ Ventre A. G. — In: Fuzzy Information and Decision Processes/Ed. by M. M. Gupta, E. Sanchez. Amsterdam: North-Holland, 1982, p. 25—28.
21. Wolberg W. H. et al. "Computer-derived nuclear features distinguish malignant from benign breast cytology", Human Pathology, 26 - 1995 - p. 792-796,.
22. Knowledge Discovery Through Data Mining: What Is Knowledge Discovery? - Tandem Computers Inc., 1996. Кречетов Н. Продукты для интеллектуального анализа данных. - Рынок программных средств, N14-15_97, с.32-39.

23. Boulding K.E. General Systems Theory - The Skeleton of Science // Management Science, 2, 1996.
24. Дюк В.АData Mining: учебный курс./ Самойленко А.П. -СПб.: Питер, 2001. - 368 с.
25. Барсегян А.А. Методы и модели анализа данных: OLAP и Data Mining / Барсегян А.А., Куприянов М.С., Степаненко В.В., Холод И.И. – СПб.: БХВ-Петербург, 2004. – 336 с.
26. Барсегян А.А. Технологии анализа данных: Data Mining, Visual Mining, Text Mining, OLAP / Барсегян А.А., Куприянов М.С., Степаненко В.В., Холод И.И. – СПб.: БХВ-Петербург, 2010. – 384 с.
27. Чубукова И.А. Data Mining. Учебное пособие. - М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2006.
28. Agrawal R. Fast Algorithms for Mining Association Rules in Large Databases, Proceedings of the 20th International Conference on Very Large Data Bases, September 1994, s. 478-499.
29. Berry M.J.A./ Data Mining Techniques for Marketing, Sales and Customer Support, Wiley Computer Publishing, 1997.
30. Anurag Choubey, "A Survey of Efficient Algorithms and New Approach for Fast Discovery of Frequent itemset for Association Rule Mining", / Ravindra Patel, J.L.Rana IJSCE ,ISSN: 2231-2307, vol. 1, issue 2,May 2011.
31. Dr. Varun Kumar, "Mining Association Rules in Student's Assessment Data", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 5, No 3, September 2012.
32. Du Ping, " A New Improvement of Apriori Algorithm for Mining Association Rules", International Conference on Computer Application and System Modeling (ICCASM 2010), Vol. 2, 529-532.
33. Farah Hanna AL-Zawaidah, "An Improved Algorithm for Mining Association Rules in Large Databases", World of Computer Science and Information Technology Journal (WCSIT), / Yosef Hasan Jbara ISSN: 2221- 0741 Vol. 1, No. 7, 311-316, 2011.

34. Hassan M. Najadat, “An Improved Apriori Algorithm for Association Rules”, / Mohammed Al-Maolegi, Bassam Arkok,/International Research Journal of Computer Science and Application Vol. 1, No. 1, June 2013, PP: 01 – 08.
35. H.Toivonen, “Sampling large databases for association rules”. In Proc. 2006 Int. Conf. Very Large Data Bases(VLDB'06),pages 134-145, Bombay, India, Sep.2006.
36. Hu Ji-ming, Xian Xue-feng. “ The Research and Improvement of Apriori for association rules mining”, Computer Technology and Development 2006 16(4) pp. 99-104.
37. Jiao Yabing, “Research of an Improved Apriori Algorithm in Data Mining Association
38. Chen M.S.: Data Mining: An Overview from a Database Perspective, IEEE Transactions on Knowledge and Data Engineering, / Han J., Yu P.S.8(6): p.866-883, 2001.
39. Weka Machine Learning [Електронний ресурс] How to Run Your First Classifier in Weka – Режим доступу: http://machinelearningmastery.com/how-to-run-your-first-classifier-in-weka/ .
40. Documentation [Електронний ресурс] Software - Режим доступу:http://www.cs.waikato.ac.nz/ml/weka/documentation.html.
41. Data Mining [Електронний ресурс] ZeroR Режим доступу: http://www.saedsayad.com/zeror.htm.
42. IntroductionToWeka [Електронний ресурс] IntroductionToWeka - Режим доступу:https://www.cs.auckland.ac.nz/courses/compsci367s1c/tutorials/IntroductionToWeka.pdf
43. An Introduction to Data Mining [Електронний ресурс] Modeling - Режим доступу:http://www.saedsayad.com/data_mining_map.htm
44. An Introduction to Data Mining [Електронний ресурс] Data Preparation- Режим доступу: http://www.saedsayad.com/data_preparation.htm

45. CISC 333 Weka Tutorial - Part 1 [Електронний ресурс] CISC 333 Weka Tutorial - Part 1 Data Preparation- Режим доступу: http://research.cs.queensu.ca/home/cisc333/tutorial/Weka.html
46. Zaiane, O.R. (1998). Mining multimedia data. Proceedings of the 1998 conference of the Centre for Advanced Studies on Collaborative Research / Han, J., Li, Z., Hou, J, Toronto, Ontario, Canada. Retrieved March 22, 2006 from ACM Digital Library.
47. Benoit, Gerald. (2002). Data Mining [Chapter 6, pps 265-310). In Cronin, B. (Ed.), Annual Review of Information Science and Technology: Vol. 36 (pp. 265-310). Silver Spring, MD: American Society for Information Science and Technology.Fayyad, U.M. (1996, Fall). From data mining to knowledge discovery in databases. / Piatetsky-Shapiro, G., & Smyth, P. AI Magazine, 17(3), pp. 37-54.Two Crows. (1999) About Data Mining [Third Edition].
48. Witten, I.H. (2005). Data mining: practical machine learning tools and techniques(2nd ed, Morgan-Kaufman Series of Data Management Systems). San Francisco: Elsevier.
49. Deltour, A. (2001). *Tertius Extension to Weka* (Technical Report No. CSTR-01-001). United Kingdom: University of Bristol.
50. Data mining [Електронний ресурс] REPORT Data mining – Режим доступу: http://knowledge.allbest.ru/programming/2c0b65625a3ad68b4c43a88521316c36_0.html
51. Agrawal R. Fast Algorithms for Mining Association Rules in Large Databases, / Srikant R.: / Proceedings of the 20th International Conference on Very Large Data Bases, September 2012, s. 478-499.
52. Dr. Varun Kumar, “Mining Association Rules in Student’s Assessment Data”, / Anupama Chadha / IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 5, No 3, September 2012.
57. Du Ping, Gao Yongping, “ A New Improvement of Apriori Algorithm for Mining Association Rules”, International Conference on Computer Application and System Modeling (ICCASM 2010), Vol. 2, 529-532.

58. Hassan M. Najadat, “An Improved Apriori Algorithm for Association Rules”, Mohammed Al-Maolegi, Bassam Arkok International Research Journal of Computer Science and Application Vol. 1, No. 1, June 2013, PP: 01 – 08.

59. Toivonen H., “Sampling large databases for association rules”. In Proc. 2006 Int. Conf. Very Large Data Bases(VLDB'06),pages 134-145, Bombay, India, Sep.2006.

60. Maning, Xian Xue-feng. “ The Research and Improvement of Apriori for association rules mining”, Computer Technology and Development 2006 16(4) pp. 99-104.